www.westermo.com

User Guide

for WeOS 5.31.0

© 2026 Westermo Network Technologies AB

Table of Contents

1. Introduction 7
1.1. WeOS User Guide 7
1.2. Getting Started 8
2. Configuration Guides 15
2.1. Management Interfaces 15
2.1.1. Switch management interfaces 15
2.1.2. Web Interface 16
2.1.3. Command Line Interface (CLI) 21
2.1.4. Serial Console 22
2.1.5. Secure Shell (SSH) 25
2.1.6. Telnet 36
2.1.7. PROFINET 38
2.1.8. Limit number of simultaneous remote management sessions 42
2.1.9. Simple Network Management Protocol 43
2.1.10. Metrics 53
2.2. AAA and User Management 56
2.2.1. Authentication, Authorisation and Accounting 56
2.2.2. AAA - Restricted interface 80
2.3. Ethernet/LAN Ports 87
2.3.1. Ethernet Ports 87
2.3.2. Link Aggregation 100
2.3.3. Transparent Bridge 104
2.3.4. Link Layer Discovery Protocol 106
2.3.5. Port Monitor 108
2.3.6. Power over Ethernet (PoE) 110
2.3.7. Jumbo Frame Switching 115
2.3.8. Flow Control 116
2.4. Bridging/Switching 118
2.4.1. Bridge Functionality 118
2.4.2. VLAN Support 121
2.4.3. Layer-2 QoS Config Guide 132
2.4.4. IGMP/MLD Snooping 139
2.4.5. Port Access Control 153
2.4.6. Fast Reconfiguration of Network Topology (FRNT) 185
2.4.7. Rapid Spanning Tree Protocol 195
2.4.8. Media Redundancy Protocol 199
2.4.9. Ring Coupling 204
2.4.10. Ring Chaining 211
2.4.11. High-Availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP) 216
2.5. IP Host Services 221
2.5.1. Interface 221
2.5.2. DDNS (Dynamic DNS) 236
2.6. Generic Maintenance 241
2.6.1. Auxiliary Ports 241
2.6.2. System Bootstrap 243
2.6.3. Encrypted secrets 256

© 2026 Westermo Network Technologies AB

2.6.4. Factory Reset 258
2.6.5. Filesystem 260
2.6.6. Media Tools 269
2.6.7. Packages Support 283
2.6.8. Port Provisioning 288
2.6.9. System Settings 291
2.6.10. Firmware Upgrade 297
2.6.11. Firmware Checksum 301
2.6.12. Watchdog 304
2.7. Alarm, LEDs and Logging 309
2.7.1. Alarm 309
2.7.2. Auditable Events 322
2.7.3. Logging Support 364
2.7.4. Front Panel LEDs 428
2.8. Network Services 430
2.8.1. DHCP Relay Agent 430
2.8.2. DHCP Server 438
2.8.3. Domain Name Service – DNS 452
2.8.4. Network Time Protocol – NTP 456
2.8.5. PTP / IEEE 1588 462
2.9. PKI 472
2.9.1. Certificate and Key Management and Usage 472
2.9.2. Simple Certificate Enrollment Protocol 480
2.9.3. Certificate Revocation 489
2.10. Routing 495
2.10.1. Static Routing 495
2.10.2. Equal-Cost Multi-Path Routing (ECMP) 502
2.10.3. Virtual Router Redundancy Protocol 506
2.10.4. Open Shortest Path First Protocol 515
2.10.5. Routing Information Protocol 532
2.10.6. Static Multicast Routing 539
2.10.7. Protocol Independent Multicast - Sparse Mode (PIM-SM) 542
2.10.8. Hardware offloading for Routing 567
2.10.9. Policy-Based Routing (PBR) 569
2.11. Tunnels and VPN 576
2.11.1. GRE Tunnelling Guide 576
2.11.2. OpenVPN Guide 580
2.11.3. IPsec Tunnelling Guide 592
2.11.4. WireGuard Tunnelling Guide 606
2.12. Firewall and NAT 612
2.12.1. Firewall Guide 612
2.12.2. Stateless NAT Guide 624
2.13. Train 627
2.13.1. TTDP (IEC 61375-2-5) 627
2.13.2. Status 642
2.13.3. Custom multicast routes 642
2.13.4. IEC 61375-2-3 (Communication Profile) 643
2.13.5. TRDP custom telegrams 648
2.13.6. “Custom telegram” data items list 660

© 2026 Westermo Network Technologies AB

2.14. Policy Filtering 680
2.14.1. Policy Traffic Filtering 680
2.15. Configuration Integrity 687
2.15.1. Boot Configuration Integrity 687
2.15.2. Configuration-integrity 688
2.16. Application Hosting 689
2.16.1. Application Hosting - Introduction 689
2.16.2. Application Hosting - Environment 691
2.16.3. Application Hosting - Getting started 694
2.16.4. Application Hosting - Configuration 696
2.16.5. Application Hosting - Management 704
2.16.6. Application Hosting - Common Problems 707
3. Hardening Guides 709
3.1. Introduction 709
3.1.1. Overview 709
3.1.2. Security Principles 710
3.1.3. Security Architecture 712
3.2. Hardening Practices 714
3.2.1. General Information 714
3.2.2. Fundamental Hardening 717
3.2.3. Enhanced Hardening 727
3.3. Checklists 733
3.3.1. IEC/ISA 62443-4-2 checklist 733
4. HowTos 736
4.1. AAA and User Management 736
4.1.1. AAA - Login with Builtin User Accounts HowTo 736
4.1.2. AAA - Login with Local Database HowTo 739
4.1.3. AAA - Login with Centralized Authentication HowTo 740
4.2. Ethernet/LAN Ports 747
4.2.1. Ethernet Port Speed/Duplex Howto 747
4.2.2. Port Monitor Basic HowTo 749
4.2.3. Storm Control HowTo 751
4.2.4. Health Monitor HowTo 755
4.3. Bridging/Switching 758
4.3.1. VLAN HowTo 758
4.3.2. Layer-2 QoS HowTo 769
4.3.3. Multicast Distribution HowTo 777
4.3.4. IEEE 802.1X Basic Setup HowTo 781
4.3.5. MAC authentication using accept list HowTo 784
4.3.6. MAC authentication using RADIUS (MAB) HowTo 787
4.3.7. Dynamic VLAN Assignment with 802.1X HowTo 790
4.3.8. Dynamic VLAN Assignment with MAC Authentication Bypass (MAB) HowTo 800
4.3.9. FRNT HowTo 808
4.3.10. MRP HowTo 810
4.3.11. MRP Over Layer 2 Tunnel 812
4.3.12. IEEE 802.1Q Tunneling 818
4.3.13. HSR and PRP HowTo guide 822
4.3.14. Ring with two nodes HowTo 826

© 2026 Westermo Network Technologies AB

4.3.15. A ring with six nodes HowTo 828
4.4. IP Host Services 830
4.4.1. Dynamic DNS (DDNS) Basic HowTo 830
4.4.2. IPv6 Basic HowTo 834
4.5. Generic Maintenance 848
4.5.1. Access External Media Devices Howto 848
4.5.2. Backup and Restore 856
4.5.3. File Handling Howto 859
4.5.4. Bootstrap System Configuration Over Network 861
4.5.5. Bootstrap System Configuration with DHCP 864
4.5.6. External Media HowTo 868
4.5.7. Secure Boot unlock using tokens 873
4.6. Alarm, LEDs and Logging 875
4.6.1. Logging to Remote Machines HowTo 875
4.6.2. Logging and DHCP Option 7 882
4.6.3. Logging Relay HowTo 887
4.6.4. Logging and Receiving From Remote Clients 894
4.6.5. Transmitting Auditable Events to Remote Machines HowTo 900
4.7. Network Services 903
4.7.1. DHCP Server HowTo 903
4.7.2. DHCP Client Relay Server HowTo 907
4.7.3. DHCP Relay/Server in Ring HowTo 909
4.7.4. DHCP IP Per-Port HowTo 912
4.7.5. PTP HowTo 914
4.8. PKI 917
4.8.1. PKI - Enroll Certificates HowTo 917
4.9. Routing 925
4.9.1. VRRP HowTo 925
4.9.2. VRRP Load Balancing HowTo 927
4.9.3. VRRP Synchronisation Groups HowTo 930
4.9.4. OSPF HowTo 934
4.9.5. OSPF Areas HowTo 938
4.9.6. OSPF Route Summarization and Filtering HowTo 946
4.9.7. OSPF Route Redistribution HowTo 950
4.9.8. Policy-Based Routing (PBR) 956
4.9.9. Equal-Cost Multi-Path Routing (ECMP) HowTo 964
4.9.10. Multiple Upstream Failover HowTo 974
4.9.11. Protocol Independent Multicast - Sparse Mode (PIM-SM) HowTo 982
4.10. Tunnels and VPN 989
4.10.1. Host to LAN Tunnel HowTo 989
4.10.2. LAN Tunnel Bridging HowTo 994
4.10.3. Internal Routes HowTo 998
4.10.4. IPsec Site-To-Site 1002
4.10.5. IPsec Host-To-Host 1005
4.10.6. IPsec in Legacy Mode (IKEv1) 1007
4.10.7. IPsec with Certificates 1010
4.10.8. IPsec Through a Firewall 1014
4.10.9. WireGuard Site-To-Site 1018
4.11. Firewall and NAT 1023

© 2026 Westermo Network Technologies AB

4.11.1. Firewall HowTo 1023
4.11.2. Stateless NAT HowTo 1028
4.12. Train 1032
4.12.1. One Consist Network with One ETB Node HowTo 1032
4.12.2. One Consist Network with Two Redundant ETB Nodes HowTo 1035
4.12.3. Two Consist Networks with One ETB Node HowTo 1041
4.12.4. Two Consist Networks with Two Redundant ETB Nodes HowTo 1044
4.12.5. Multicast Between End Devices in Two Consists HowTo 1050
4.12.6. Port Settings for ETBN Backbone Ports HowTo 1054
4.12.7. Redundant ECSP Services HowTo 1057
4.12.8. Non Redundant ECSP Services HowTo 1065
4.12.9. Split ECN over ETB HowTo (advanced) 1071
4.13. Policy 1077
4.13.1. Policy Traffic Filtering HowTo 1077
4.14. Application Hosting 1079
4.14.1. Application Hosting HowTo 1079
4.14.2. Use Case: Firewall 1086
5. Appendix 1092
5.1. Application Hosting 1092
5.1.1. AppArmor Profiles 1092
5.1.2. Seccomp Profiles 1095
5.2. Train 1119
5.2.1. ECN Subnetting 1119
6. File Appendix 1122

© 2026 Westermo Network Technologies AB

1. Introduction

1.1. WeOS User Guide

About

Westermo has an extensive set of network products for industrial data communication, managed as well as unmanaged. The products can be found in a diverse set of harsh environment applications, where robustness and reliability are vital properties.

This guide describes the functionality of managed Westermo products running the Westermo network operating system, WeOS.

We recommend newcomers to first browse through the Getting Started guide, before diving into other topics of interest.

Legal Information

The contents of this document are provided ”as is”. Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are made in relation to the accuracy and reliability or contents of this document. Westermo reserves the right to revise this document or withdraw it at any time without prior notice.

Under no circumstances shall Westermo be responsible for any loss of data or income or any special, incidental, and consequential or indirect damages howsoever caused. More information about Westermo can be found at the following Internet address: https://www.westermo.com[1]

[1] https://www.westermo.com

1.2. Getting Started

Windows

Prerequisites

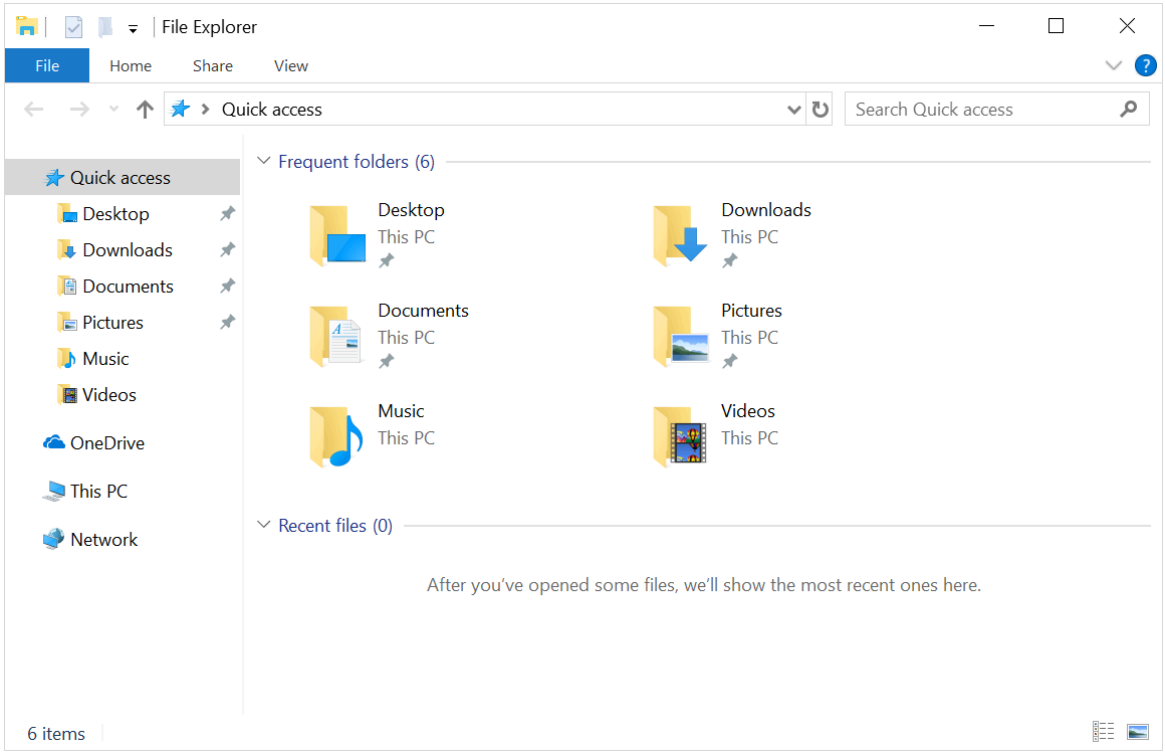
- This guide assumes you have *administrator rights* on your Windows PC
- The PC has an Ethernet NIC configured to acquire its IP address via DHCP

Steps to Access the Device

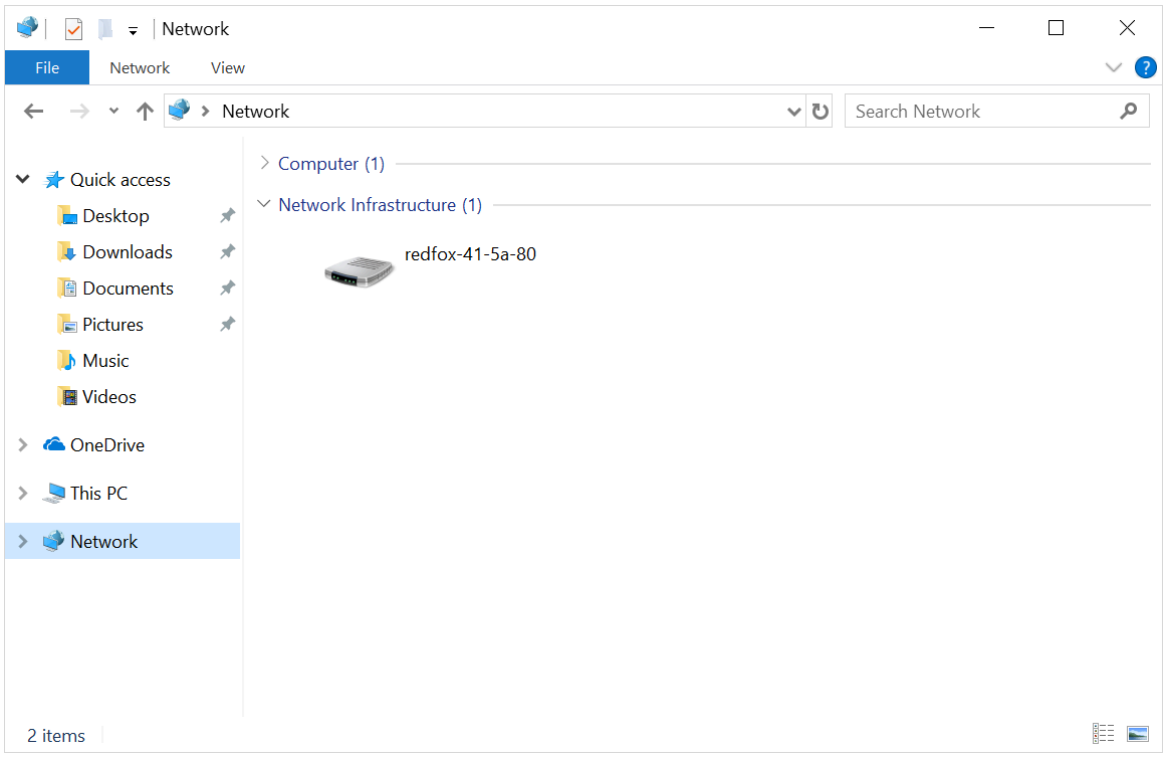
1. Make sure nothing is connected to the WeOS device
2. Make sure your PC is not connected to any network
3. Power on the switch (insert the power cable)
4. Use an Ethernet cable to connect your Windows PC to an Ethernet port on the device

```
.-------.
|       |
| WeOS  |
|       |
'---+---'
    |
    | Ethernet cable
    |
.---+---.
|       |
| WinPC |
|       |
'-------'
```

5. Open File Explorer (a.k.a Windows Explorer), shortcut *Win-E*, on your PC, and click on the *Network* item in the left pane in Windows 10. (In Windows 7 you find the *Network* item after expanding the *Desktop* item.)

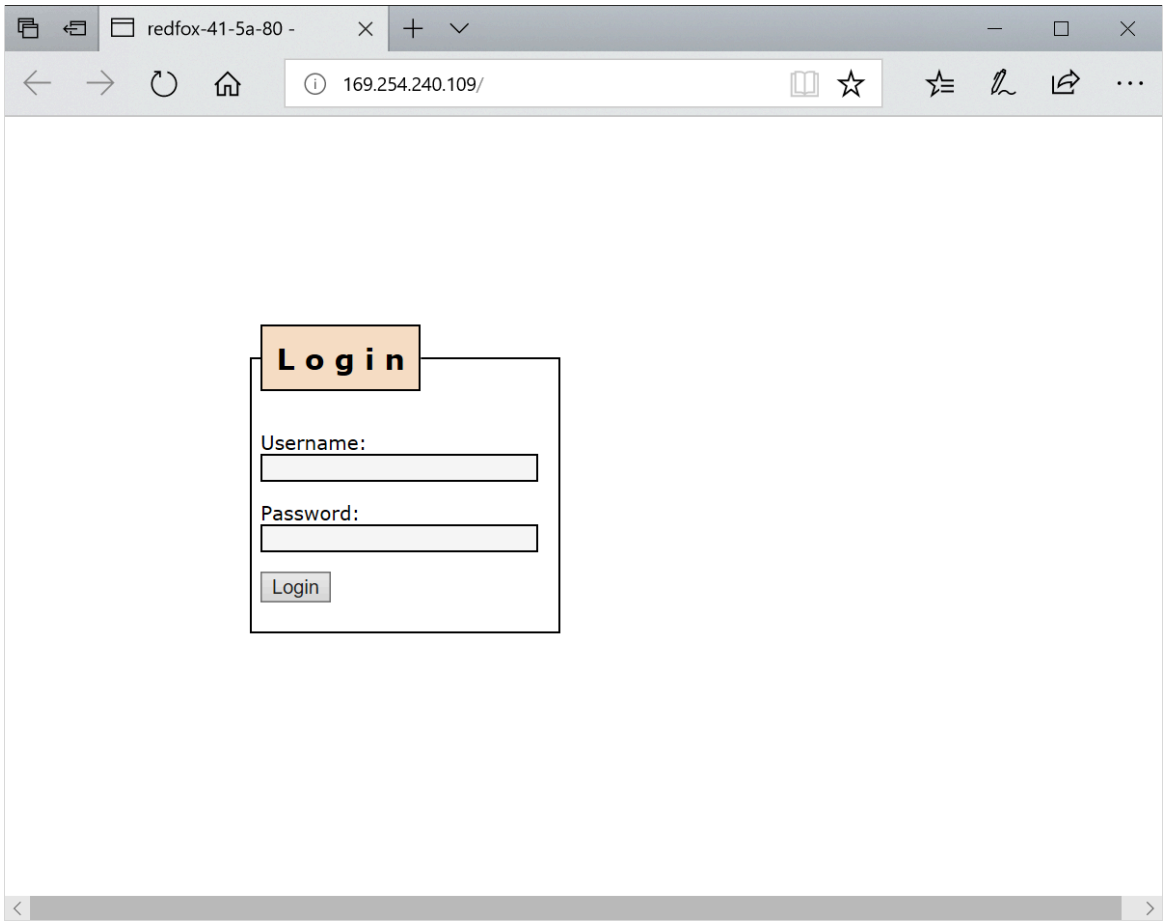


6. The WeOS switch should appear as *Infrastructure Equipment* in the right pane. The switch will be named as a concatenation of its name and the last digits of its *MAC address*, e.g., **redfox-41-5a-80**.

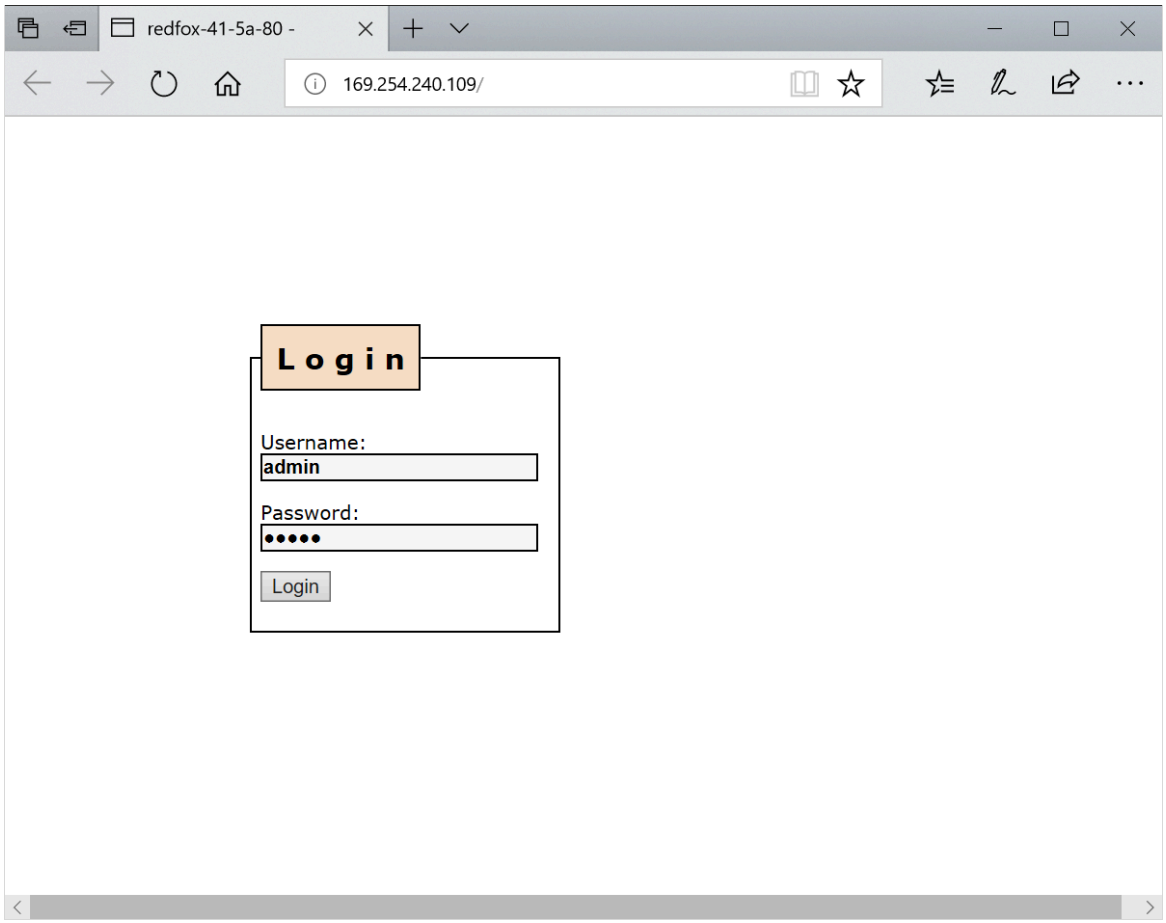


See the Troubleshooting section below in case of problems.

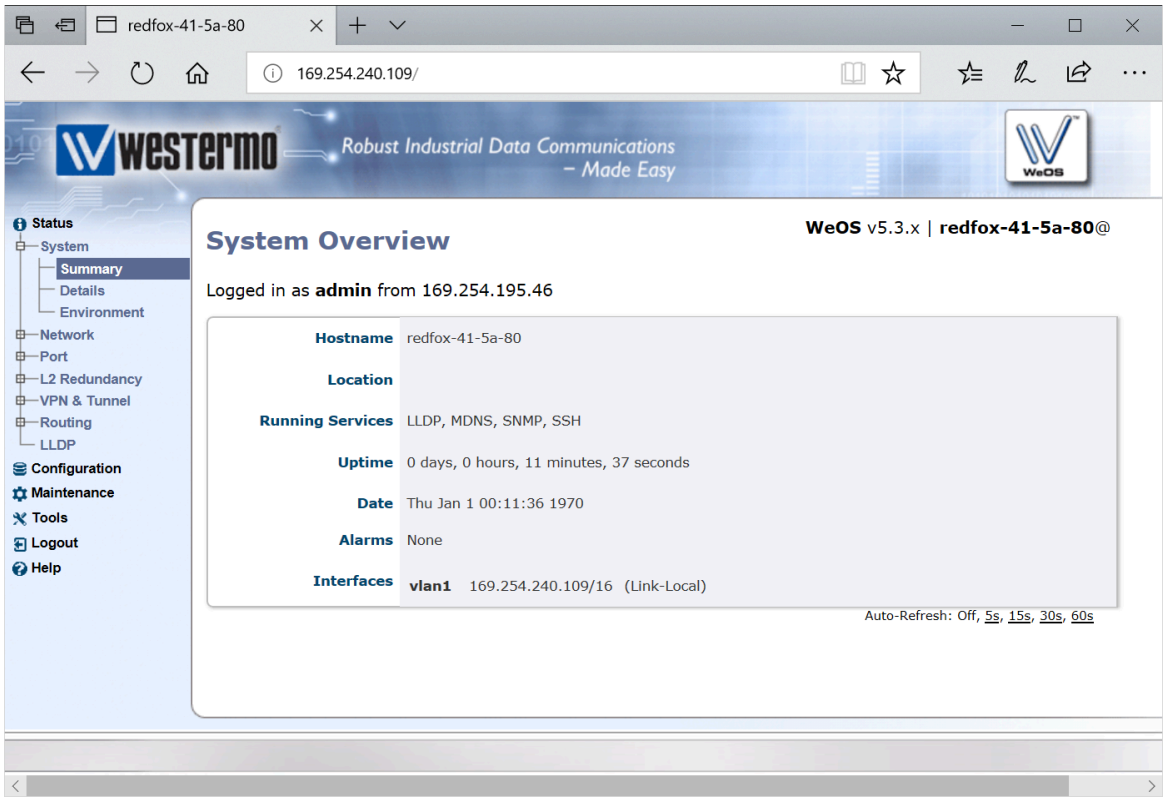
7. Double-click on the icon of the device to access the device via your web-browser.



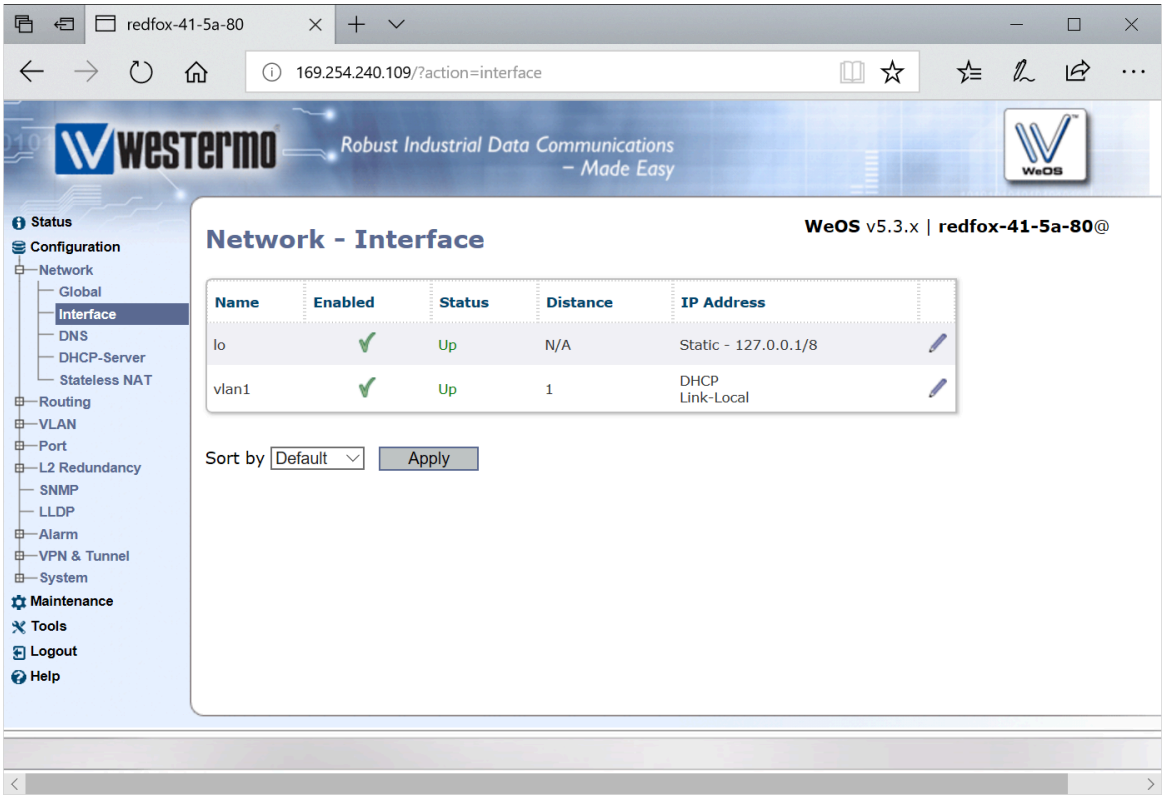
8. Login using the default login name (**admin**) and password (**admin**).



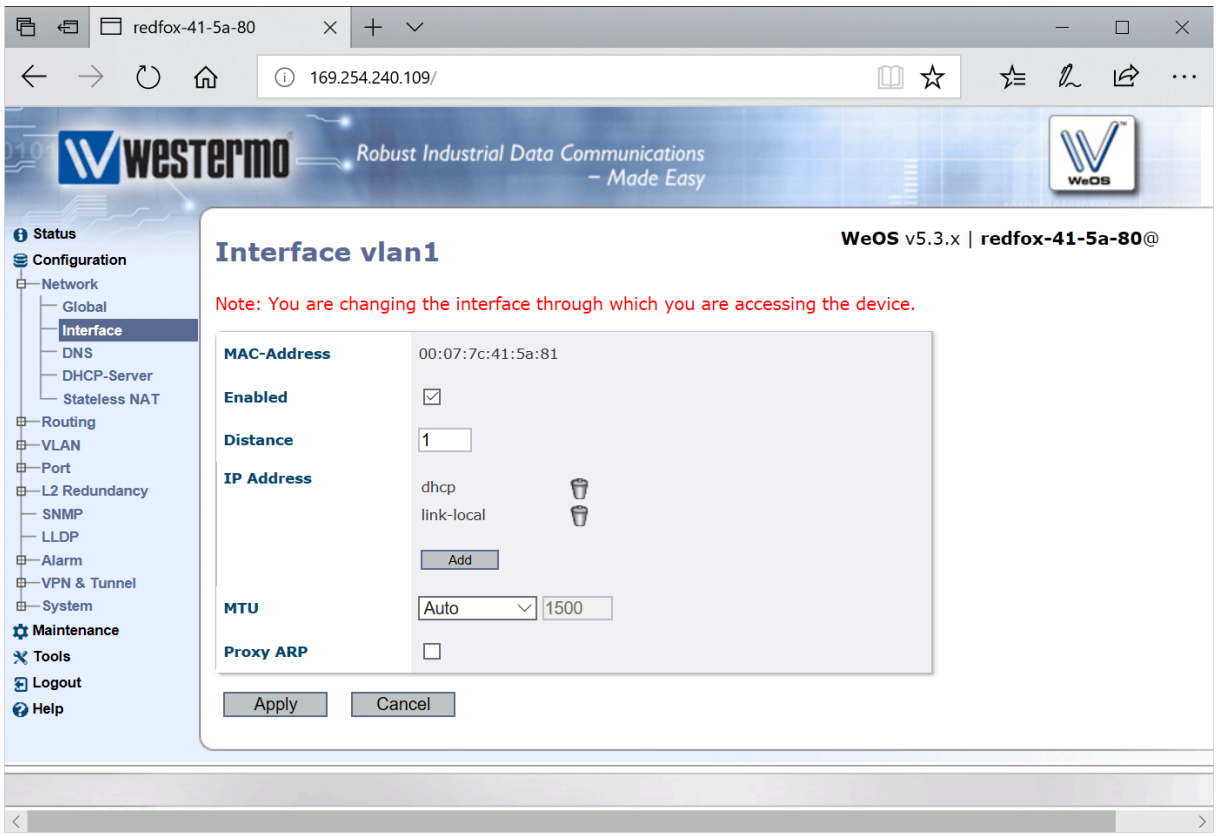
9. When logged in, you are presented with the default screen.



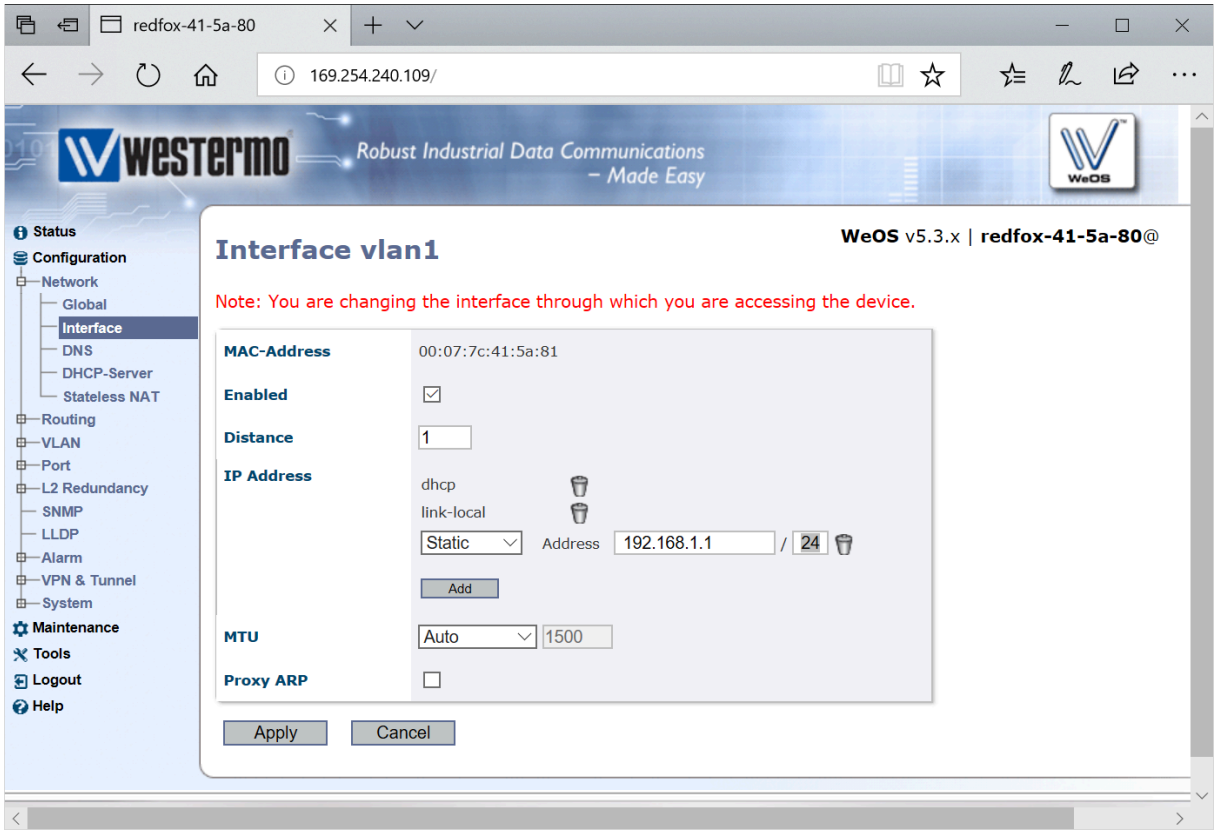
10. To assign a static IP address to your switch, use menu path *Configuration*, *Network* and *Interface* to list the existing interfaces.



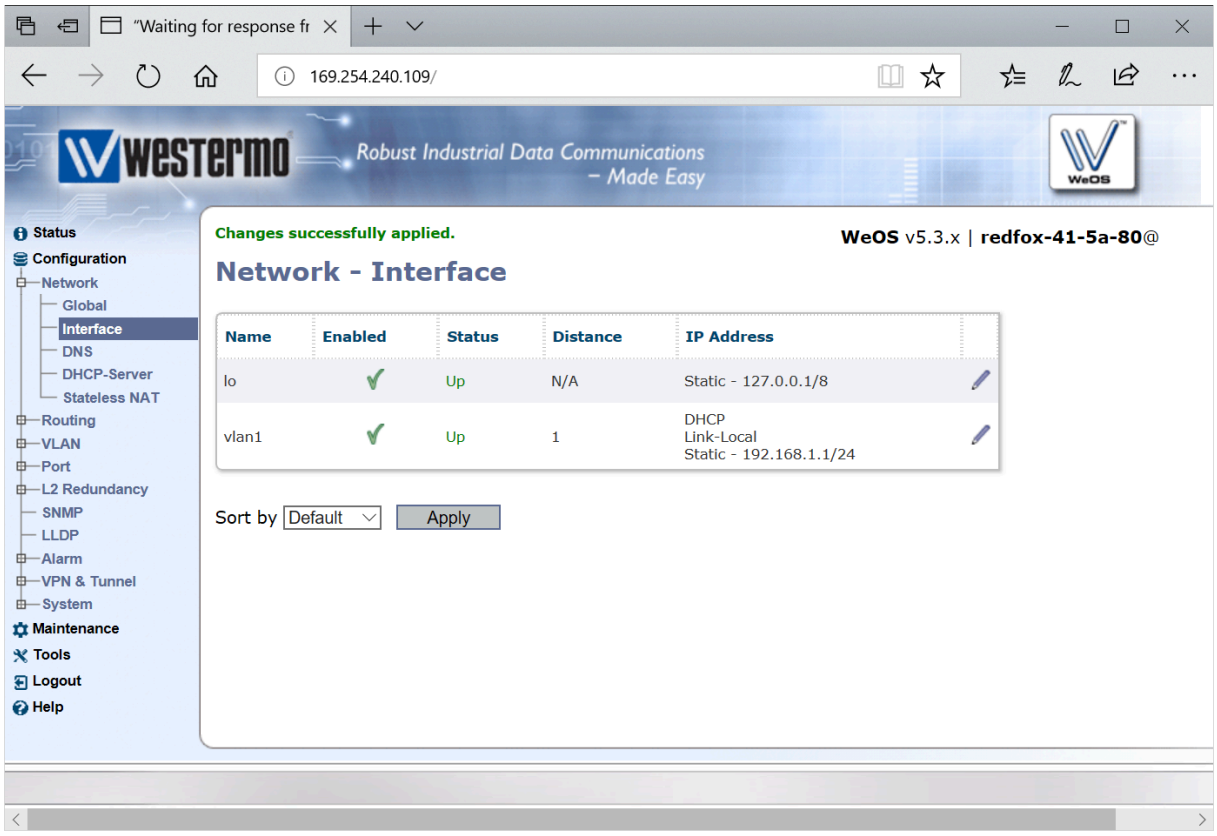
11. Enter Interface Edit Configuration page for the appropriate interface (here *vlan1*) by selecting the Edit icon for that interface.



12. Click the **Add** button and enter the appropriate IP address as prefix length (here *192.168.1.1/24*).



13. Click **Apply** button, which applies the new setting and takes you back to the Interface Overview Configuration Page.

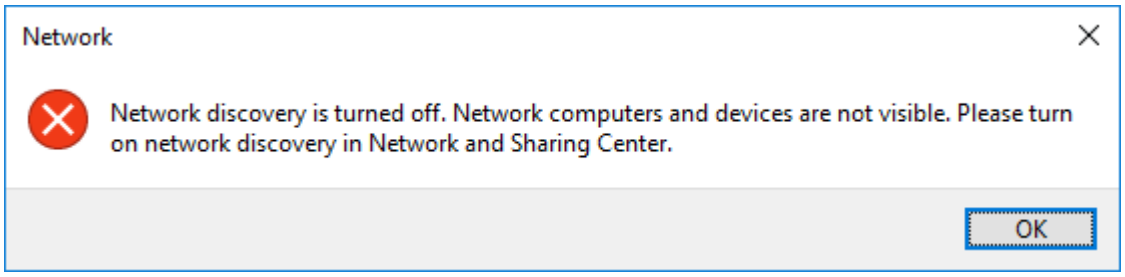


To remove the other address methods used (here *DHCP* and *Link-local*), please enter the Interface Edit Configuration page for the interface (here *vlan1*) and delete those methods if desired.

Note: deleting the *Link-local* method will break the connection to the web service once the change is applied. To access the web page again, you need to login again using the newly assigned IP address (here *https://192.168.1.1*[1]).

Troubleshooting

If *Network discovery* is not enabled on your Windows PC this dialog, or a similar screen, may appear.



Then enable *Network discovery* and try again. Please contact your systems administrator if you need assistance to enable *Network discovery*.

If *Network discovery* is enabled, but no WeOS device appears in the infrastructure list:

- Verify that the switch has power by inspecting that (at least one) power LED indicate GREEN. If not, ensure the power cable is inserted correctly.
- Ensure that the Ethernet cable is properly connected between the PC and the WeOS switch, and verify that the port LED on the switch shows GREEN (indicates link-up). If not, contact your system administrator for assistance.

[1] https://192.168.1.1

- Verify that your Ethernet NIC is configured to acquire its IP address via DHCP. If not, change the setting to use DHCP.
- Contact your system administrator for further assistance. Alternatively, you login to your unit using a console cable (described below).

2. Configuration Guides

2.1. Management Interfaces

2.1.1. Switch management interfaces

Introduction

The switch main configuration is stored in JSON format, however to manage the switch the following options are supported:

- Command line interface (CLI): The CLI can be accessed via Console, SSH and Telnet.
- Web interface: Web interface can be accessed via HTTP/HTTPS.
- SNMP (v1/v2c/v3): Currently limited to read-only operations and trap support; no general SNMP write support. See MIB conformance information in release zip for details.

Configuration

Management services can be configured in the *management* configuration context.

```
example:/#> configure
example:/config/#> management
example:/config/management/#>
```

From the management configuration context it is possible to enter specific sub-contexts to manage settings of individual services:

- `web`: Manage settings for the Web interface.
- `cli`: Manage general settings for the CLI. These settings apply to CLI, irrespective if accessed via Console, SSH or Telnet.
- `ssh`: Manage settings for the SSH service.
- `telnet`: Manage settings for the Telnet service.
- `snmp`: Manage SNMP settings.

2.1.2. Web Interface

Introduction

The switch provides a Web interface for management and status overview. Both HTTP and HTTPS access are supported and can be configured individually. A single Web session is possible at a time.

Configuration of Web services is done in the Web configuration context.

```
example:/#> configure
example:/config/#> management
example:/config/management/#> web
example:/config/management/web/#>
```

HTTP and HTTPS are managed in individual sub-contexts. The example below disable HTTP access, and changes the default port for HTTPS to
43434.

```
example:/config/management/web/#> http
example:/config/management/web/http/#> no enable
example:/config/management/web/http/#> end
example:/config/management/web/#> https
example:/config/management/web/https/#> port 43434
example:/config/management/web/https/#>
```

Session timeout

If enabled it sets the Web session timeout including the upload timeout for file upload. Value is in minutes. If not enabled the value 0 is stored in web session timeout and then the file upload will use a default timeout of 7200 sec, 120 min.

The example below shows how to set the Web session inactivity timeout to 5 minutes (default 10 minutes).

```
example:/#> configure
example:/config/#> management
example:/config/management/#> web
example:/config/management/web/#> session-timeout 5
```

Syntax

`[no] session-timeout <1..1440>`

Description

Set auto logout delay in minutes, default 10 min.

Pre-Login Banner

If configured this will ensure that a pre-login banner is displayed on the login page of the Web interface. The message can be written and formatted free-form at the discretion of the administrator. Generally, these types of messages are composed to inform any potential unauthorized users that they are attempting to access a restricted area.

Utilizing this could help dissuade any potential intruders by providing a direct warning that accessing the device may be an illegal or prohibited activity. If a login banner has been present

during a possible breach, the potential perpetrator will have an increased difficulty if attempting to claim any ignorance, since the individual was clearly warned that access was prohibited.

When configured, the login page will also display an acknowledgement checkbox that must be checked before the user is allowed to attempt a login. This is so that the user should not be able to accidentally overlook the message displayed in the banner.

> **WARNING**
>
> Any pre-login banner message that is intended to be used for any potential legal matters should possibly be approved for use by the organisations legal team.
>
> Westermo only provide the possibility to compose such a message, and is never involved and/or responsible for any of the contents and/or actions that may want to be pursued as a result.

The following is an example how this can be configured using the CLI. When configured using the CLI escape characters can be used to create line-breaks for instance. Remember to enclose the message like this `"MESSAGE"` or `'MESSAGE'`.

```
example:/#> configure
example:/config/#> management
example:/config/management/#> web
example:/config/management/web/#>
login-banner "This is an example message.\n\nThis is on a new line.\n\nRemember,
this is just an example text, not to actually be used in a production environment."
example:/config/management/web/#>
```

If the overview `show` command is used the banner will be attempted to be displayed as it would appear format wise on the Web login:

```
example:/config/management/web/#> show
[ ... ]

Login Banner
| This is an example message.                                                  |
|                                                                              |
| This is on a new line.                                                       |
|                                                                              |
| Remember, this is just an example text, not to actually be used in a         |
| production environment.                                                      |

[ ... ]
```

In order to see how the message was composed, including the escape characters, use the specific `show login-banner` command:

```
example:/config/management/web/#> show login-banner
"This is an example message.\n\nThis is on a new line.\n\nRemember, this is just an
example text, not to actually be used in a production environment."
```

Syntax

`[no] login-banner [MESSAGE]`

Description

Configure a banner to be displayed on the login page of the WEB.

This is a banner that will be displayed on the login page, before the user has logged in to the device. When this is configured the user will be forced to acknowledge the message before being allowed to log in.

Default: `no login-banner` (Disabled)

Options

Option	Description
`no`	Remove any configured login-banner and disable it.
`MESSAGE`	Free-form message, remember to enclose the message like this `"MESSAGE"` or `'MESSAGE'`. When writing the message certain escape characters can also be included: • `\b` => Backspace • `\t` => Horizontal TAB • `\n` => New line • `\r` => Carriage Return

Other General Web Settings

These are accessed in the main Web configuration context:

```
example:/#> configure
example:/config/#> management
example:/config/management/#> web
example:/config/management/web/#>
```

`[no] suppress-login-error-messages`

Description

Suppress login authentication error messages on the Web login page.

If enabled, the Web login page will not display any error messages when a user fails to log in. Instead, the page will simply reload and no error message will be shown.

This can be useful in environments where you want to avoid providing any information about the login attempts what so ever.

Default: `no suppress-login-error-messages` (Disabled)

Options

Option	Description
`no`	Disable suppression of login error messages, i.e. enable error messages.

HTTP Settings

```
example:/#> configure
example:/config/#> management
example:/config/management/#> web
example:/config/management/web/#> http
example:/config/management/web/http/#>
```

Syntax of HTTP settings

`[no] enable`

Description

Activate or deactivate Web access via HTTP

`[no] port PORT`

Description

Port to run HTTP daemon (default 80)

HTTPS Settings

```
example:/#> configure
example:/config/#> management
example:/config/management/#> web
example:/config/management/web/#> https
example:/config/management/web/https/#>
```

For HTTPS, the switch auto-generates an individual Web server certificate upon first boot, after a factory reset, or if the Web server certificate has been deleted. In addition, it is possible to upload a custom certificate and specify the Web server to use it instead.

Syntax of HTTPS settings

`[no] enable`

Description

Activate or deactivate Web access via HTTPS

`[no] port PORT`

Description

Port to run HTTPS daemon (default 443)

`[no] certificate LABEL`

Description

Set custom certificate to use, if not specified, the default certificate provided by the firmware will be used.

© 2026 Westermo Network Technologies AB

`[no] fallback`

Description

If enabled, the device will use a default certificate if the specified certificate is pending enrollment or is removed.

`[no] tls13`

Description

Enable or disable TLS 1.3 support. When enabled, the Web server auto-selects the appropriate protocol, supporting both TLS 1.2 and TLS 1.3. The default value depends on the hardware platform.

The example below shows how to specify a custom Web certificate, assuming the certificate has already been uploaded to the certificate repository (see label 'server1' below).

```
example:/config/management/web/https/#> do show cert full
TYPE     HASH     EXPIRES     NAME                LABEL
CA-local e0145f8c Sep  4 2032 ACME CA             server1
Pub      eed11dc2 Dec 10 2024 server1             server1
Pub      4a23103e Jan 19 2038 zero-12-34-50.local web-default
Key      N/A      N/A         server1             server1
Key      N/A      N/A         web-default         web-default
example:/config/management/web/https/#> certificate server1
example:/config/management/web/https/#>
```

Access to the Web

Up to WeOS 5.26, Administrators had exclusive access to the Web, with both read and write permissions, allowing them to not only view the status of the system, but to make configuration changes.

As of WeOS 5.26, Operators have been granted read-only access to the Web, where they can view the status of the system, perform firmware upgrade and restart the system. Operators are not able to make configuration changes to the system.

Operator is not able to log in through the web if Administrator is already logged in, but the opposite works well. An Administrator is able to login regardless if another user is already logged in, logging out the previous logged in user. This behaviour is valid only for the web interface.

2.1.3. Command Line Interface (CLI)

Introduction

The Command Line Interface (CLI) can be access via the console port or remotely via SSH or Telnet.

Timeout

It is possible to configure a console CLI timeout (seconds). When enabling CLI timeout the user is automatically logged out after the configured number of seconds of inactivity.

```
example:/#> configure
example:/config/#> management
example:/config/management/#> cli
example:/config/management/cli/#> timeout 60
```

The example above sets a CLI timeout of 60 seconds. After one minute of inactivity the user will be automatically logged out of the CLI. This timeout is valid for users logging in to the device over console, SSH or Telnet.

Note: The CLI timeout is set independently from the Web *session timeout*. There is also a separate SSH idle timeout, which automatically disconnects an SSH connection when the connecting user has not been active for a while.

`[no] timeout SEC`

Description

This setting is used to automatically logs out a user from CLI when user has not been active for a while.

Options

Option	Description
`no`	Disable CLI idle timeout completely
`SEC`	Number of seconds (1..86400) of idle activity until a user is automatically logged out from CLI.

2.1.4. Serial Console

Introduction

For devices equipped with a serial console port the device can be managed through the console, without need for a working network connection. The serial console uses the settings *115200/8-N-1*.

The console port can also be used to display logging information (if console logging is enabled) and to conduct a factory reset.

Authentication

The console credentials are the same as for connecting through SSH or logging in to the Web interface.

Console login

It is possible to disable console access using either the CLI or the Web interface. After disabling the console it is still possible to manage the device through SSH or Web.

```
example:/#> configure
example:/config/#> management
example:/config/management/#> console
example:/config/management/console/#> no login
```

Note: Factory reset is *only* allowed over the console. This still works if console login is disabled.

Configuration

The console service is available from the management configuration context in the CLI:

```
example:/#> configure
example:/config/#> management
example:/config/management/#> console
example:/config/management/console/#>
```

`[no] login`

Description

Enable or disable the console login.

Default: Enabled

Options

Option	Description
`no`	Disable the console login.

`[no] login-banner [MESSAGE]`

Description

Configure a banner to be displayed on the local console login.

This is a banner that will be displayed on the local console login, before the user has logged in to the device.

continues on next page

continued

`[no] login-banner [MESSAGE]`

> **TIP**
>
> For additional general information regarding pre-login banner, refer to the section on Pre-Login Banner on the Management Web configuration page.

Default: `no login-banner` (Disabled)

Configuration Example

A pre-login banner message can be configured in the following manner:

```
example:/config/management/console/#> login-banner "This is a message.\n\nThis is displayed on a new line."
```

The overview `show` command will try and display the message as it would be when considering the escape characters:

```
example:/config/management/console/#> show
[ ... ]

Login Banner
| This is a message.                                                |
|                                                                   |
| This is displayed on a new line.                                  |
```

To see how the raw message that was configured the specific `show login-banner` command can be used:

```
example:/config/management/console/#> show login-banner
"This is a message.\n\nThis is displayed on a new line."
```

Login Example

This is how the message would be displayed when trying to access the device over a console connection:

```
Please press Enter to activate this console.

This is a message.

This is displayed on a new line.
example login:
```

Options

Option	Description
`no`	Remove any configured login-banner and disable it.
`MESSAGE`	Free-form message, remember to enclose the message like this `"MESSAGE"` or `'MESSAGE'`. When writing the message certain escape characters can also be included:

continues on next page

continued

`[no] login-banner [MESSAGE]`

Option	Description
	• `\a` => BELL • `\b` => Backspace • `\t` => Horizontal TAB • `\n` => New line • `\v` => Vertical TAB • `\f` => Form Feed • `\r` => Carriage Return • `\\` => \

2.1.5. Secure Shell (SSH)

Introduction

Secure Shell (SSH) is a widely used protocol for establishing a secure channel over a public network. An SSH service is enabled by default in all products to allow safe remote login and access to the command line interface. The service defaults to listening on Internet port 22 for incoming connections, on all interfaces.

All products also come with an SSH client, which can be used to log in to other devices. Useful when debugging remote systems which do not have routing set up or cannot be connected to otherwise. This is sometimes referred to as using a device as a “hop server”.

Connecting to a Device

To connect to the SSH service you need an SSH client. There are many options available, one of the most useful is the freely available PuTTY[1] client on Windows, which also supports serial console connection. But like Linux and other UNIX systems, Windows 10 today ships with the OpenSSH[2] client **ssh**, which is a command line program. In the examples below a Linux host with the **ssh** client is used.

Connecting can be done to the assigned DNS name, the advertised mDNS name, or using the IP address of the remote device:

```
operator@example:~$ ssh admin@weos.local
admin@weos.local's password: *******
```

Enter the password associated with the *admin* account and the next text displayed is the remote system’s message of the day (MOTD) and the CLI prompt. Use `logout`, `exit`, or `Ctrl-D` to log out and disconnect.

Note: The first time connecting to a device from a specific host, the SSH client will respond with a warning (similar to):

```
The authenticity of host 'weos.local (192.168.122.89)' can't be established.
ECDSA key fingerprint is SHA256:lhQjYbBcWVZY2Zqmm1HizFx0M2opgTsBFNXmja161KE.
Are you sure you want to continue connecting (yes/no)?
```

Answering **yes** saves the public key of the remote device into a local database (known_hosts) and associates it with the IP address and port (TCP port 22 unless otherwise specified) used to connect. This key is verified every time you connect to the same address and port.

The warning text above comes from OpenSSH. Other SSH implementations may have differently worded warning messages. You may also receive a warning about a possible “man-in-the-middle” attack, which usually occurs when the same address is being used with another device. See the Troubleshooting section for more details.

Connect from a Device

The built-in SSH client can be used to connect to other devices in a network. A typical use case is to access devices in a private network where only a gateway device is publicly accessible. An operator can SSH into the gateway device and from there initiate another SSH connection to the private part of the topology.

[1] https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
[2] https://www.openssh.com/

To initiate an SSH connection from the CLI, use the `ssh` command from the top-level admin-exec context:

```
admin@example:/#> ssh admin@192.168.122.1:222
```

The `admin@` part can be changed to a different login name or left out completely if the username on the target device is the same as then one logged in on the originating device. The `:222` part designates the port number, can also be separated by a space, e.g. `admin@192.168.122.1 222`. It can be left out if the standard port (22) is used, this depends on the configuration of the SSH service of the target device. The IP address can also be replaced with a domain name.

The warning messages regarding an new untrusted host is similar to that displayed by OpenSSH:

```
    Host '192.168.122.1' is not in the trusted hosts file.
    (ecdsa-sha2-nistp256 fingerprint md5 d7:17:86:c7:69:fc:dd:16:f4:4a:62:55:eb:47:6f:43)
    Do you want to continue connecting? (y/n)
```

The warning message for a changed host key (possible eavesdropping) is also similar to that displayed by OpenSSH:

```
    WARNING: Host key mismatch for 192.168.122.1!
    Fingerprint is md5 d7:17:86:c7:69:fc:dd:16:f4:4a:62:55:eb:47:6f:43
    Expected       md5 bd:3d:e5:f3:49:24:98:14:22:28:9d:92:0a:12:48:3d
    Continue anyway (y/N)?
```

Typing **y** here allows the connection to proceed.

Syntax

As mentioned above, the `ssh` command is used directly in the top-level admin-exec context.

```
admin@example:/#>
```

Two main formats are supported for the command, where they essentially just differ in how the user is specified.

`ssh [user USER] [force ipv4|ipv6] HOST[:PORT| PORT]`

Description

Secure shell login to another host.

This command can be used to log in to another host using the SSH protocol.

As mentioned, the only difference in this format is that the user is provided using the `user` option, which is a separate option instead of being part of the host specification using the `USER@HOST` format.

> **ONLY ONE USER SPECIFICATION**
>
> A user cannot be specified both by the `user` option and by using the `USER@` syntax. This will result in an error.

continues on next page

continued

`ssh [user USER] [force ipv4|ipv6] HOST[:PORT| PORT]`

Example

Connecting to a device as user `admin`:

```
admin@example:/#> ssh user admin 192.168.1.1
```

```
admin@example:/#> ssh user admin 2001:db8::1
```

```
admin@example:/#> ssh user admin myhost
```

Special case when you want to connect to a device on an IPv6 link-local address, you need to specify the interface as well, e.g. `vlan1`:

```
admin@example:/#> ssh user admin fe80::1%vlan1
```

Connecting to a device as user `admin` on a specific port, e.g. port 2222:

```
admin@example:/#> ssh user admin 192.168.1.1:2222
```

```
admin@example:/#> ssh user admin [2001:db8::1]:2222
```

```
admin@example:/#> ssh user admin myhost:2222
```

Using the `force` option to force the use of either IPv4 or IPv6 addresses only:

```
admin@example:/#> ssh force ipv4 user admin myhost
```

```
admin@example:/#> ssh force ipv6 user admin myhost
```

> **USER SPECIFICATION**
>
> Again, all of these example commands can also be executed using the `USER@HOST` format, while omitting the `user` option.

Options

Option	Description
`user`	The user to log in as. If not specified, the current user is used, e.g. if the user is `admin` on the device, then the user `admin` is used to log in to the remote host.
`force ipv4|ipv6`	Force the use of either IPv4 or IPv6 addresses only. This is useful when the host has both IPv4 and IPv6 addresses, but you want to connect using only one of them.

continues on next page

continued

`ssh [user USER] [force ipv4|ipv6] HOST[:PORT| PORT]`

Option	Description
`HOST`	Should be a valid IPv4 or IPv6 address, or a valid FQDN (Fully Qualified Domain Name).
`PORT`	Optional port number, defaults to 22 if not specified. If a port is specified, it can be separated by a colon `:` or a space.

`ssh [force ipv4|ipv6] [USER@]HOST[:PORT| PORT]`

Description

As mentioned, the only difference when using this format, i.e. using the `user@host` format, is how the user is specified. Other than that, the command behaves the same as the previous format.

Therefore, see the previous section for more details on how to use the command.

Example

Connecting to a device as user `admin`:

```
admin@example:/#> ssh admin@192.168.1.1
```

```
admin@example:/#> ssh admin@2001:db8::1
```

```
admin@example:/#> ssh admin@myhost
```

Managing SSH host keys

Two types of SSH host keys are generated the first time a system starts. One or more of these keys can persistently be re-generated.

Supported key types:
- rsa
- ecdsa

NOTE: TPM based devices only support one type of key at a time. The automatically generated key type on first system boot will be ecdsa. The key type can be changed, but only one key type can be active at a time.

Example of how to generate a new 4096-bit RSA ssh key.

```
example:/#> ssh-host-key generate rsa size 4096
```

Configuration

The SSH service can be configured in several ways:
- It can be deactivated
- It can be restricted to a subset of available interfaces
- The port number can be changed

- Keepalive can be tuned or disabled
- Idle timeout can be configured or disabled
- Fail delay can be changed.
- An optional pre-login banner can be configured.

> **WARNING**
>
> Changing the SSH configuration while connected to the device via SSH is risky. If possible, use an alternative management front-end, e.g. the WebUI.

The SSH service is available from the management configuration context in the CLI:

```
example:/#> configure
example:/config/#> management
example:/config/management/#> ssh
example:/config/management/ssh/#>
```

`[no] enable`

Description

Enable/disable the SSH server. The rest of the configuration is however, saved for when you re-enable the service again.

The ssh service configuration can be completely reset from the parent `management` configuration context using `no ssh`.

> **WARNING**
>
> Secure copy (SCP) relies on the SSH service. If SSH is deactivated, it will not be possible for a remote host or device to use scp to copy files to or from this device.

Options

Option	Description
`no`	Disable SSH service completely

`[no] listen IFNAME[,IFNAME]`

Description

Interfaces on which to allow SSH connections.

To restrict SSH to only listen on interfaces vlan1 and vlan2:

```
listen vlan1 vlan2
```

To listen on all interfaces again (default):

```
no listen
```

`[no] port PORT`

Description

By default, SSH listens on TCP port 22. SSH clients connect to this port unless another port is specified.

> **NOTE**
>
> To connect to an SSH server listening to a non-standard port, the port has to be specified when connecting. For example:
>
> ```
> foo@localhost$ *ssh admin@192.168.122.89 -p 222*
> ```

`[no] keepalive SEC`

Description

By default, a 60 second keepalive is enabled for SSH connections. Meaning, the server and client exchange control messages every 60 seconds, which has two purposes:

- Prevent firewalls and routers from closing the connection if there is no data sent over it for a long time
- Detect a loss of connection

Options

Option	Description
`no`	Disable keepalive messages
`SEC`	Interval in seconds between keepalive probes, default: 60 sec

`[no] idle-timeout SEC`

Description

This setting is used to automatically disconnect an SSH connection when the connecting user has not been active for a while.

Keepalive probes should not be counted as user activity, however this is a non-standardised area, subject to implementation differences in both the SSH server and client. It is possible the two mechanisms (keepalive and idle timeout) will interfere with each other if both are enabled.

Options

Option	Description
`no`	Disable idle-timeout completely
`SEC`	Number of seconds (of idle activity) until connection times out

© 2026 Westermo Network Technologies AB

`[no] fail-delay SEC`

Description

Set a delay to SEC seconds occurring after a failed login attempt.

The effective delay is not constant but each time randomly generated from a pseudo Gaussian distribution with mean = SEC + 300ms and standard deviation = SEC/6. The effective delay does not exceed +-50% of SEC around the mean. Example: When selected delay SEC = 3s then mean = 3.3s, std. deviation = 0.5s, min = 1.8s, max = 4.8s.

Options

Option	Description
`no`	Set default value.
`SEC`	Fail delay in seconds

`[no] external-access`

Description

Configure access to secure shell copy (scp) to the device initiated from a remote host.

If enabled, it is possible to transfer file(s) between a remote host and WeOS device, when the transfer is initiated from the remote host.

It is always possible to transfer files when the transfer initiated from WeOS device.

Default: Disabled

Options

Option	Description
`no`	Disable external access.

WARNING

It is highly discouraged to have this setting persistently enabled. This function allows very permissive file access to the device remotely.

`[no] login-banner [MESSAGE]`

Description

Configure a banner to be displayed on the ssh console login.

This is a banner that will be displayed on the ssh console login, before the user has logged in to the device.

continues on next page

continued

`[no] login-banner [MESSAGE]`

> **TIP**
>
> For additional general information regarding pre-login banner, refer to the section on Pre-Login Banner on the Management Web configuration page.

Default: `no login-banner` (Disabled)

Configuration Example

A pre-login banner message can be configured in the following manner:

```
example:/config/management/ssh/#> login-banner "This is a message.\n\nThis is displayed on a new line."
```

The overview `show` command will try and display the message as it would be when considering the escape characters:

```
example:/config/management/ssh/#> show
[ ... ]

Login Banner
| This is a message.                                                    |
|                                                                       |
| This is displayed on a new line.                                      |
```

To see how the raw message that was configured the specific `show login-banner` command can be used:

```
example:/config/management/ssh/#> show login-banner
"This is a message.\n\nThis is displayed on a new line."
```

Login Example

This is how the message would be displayed when trying to access the device over a SSH connection:

```
operator@example:~$ ssh admin@192.168.1.1
This is a message.

This is displayed on a new line.
admin@192.168.1.1's password:
```

Options

Option	Description
`no`	Remove any configured login-banner and disable it.
`MESSAGE`	Free-form message, remember to enclose the message like this `"MESSAGE"` or `'MESSAGE'`. When writing the message certain escape characters can also be included:

continues on next page

continued

`[no] login-banner [MESSAGE]`

Option	Description
	• `\a` => BELL • `\b` => Backspace • `\t` => Horizontal TAB • `\n` => New line • `\v` => Vertical TAB • `\f` => Form Feed • `\r` => Carriage Return • `\\` => \

NOTE

The following options can only be configured via a Console session. It is highly discouraged to have this setting persistently enabled. This function should only be used when specific troubleshooting is needed.

`[no] shell-access`

Description

Manage remote shell access configuration.

This command will open a sub-context where the remote shell access configuration is managed.

Options

Option	Description
`no`	Disable remote shell access

```
example:/#> configure
example:/config/#> management
example:/config/management/#> ssh
example:/config/management/ssh/#> shell-access
example:/config/management/ssh/shell-access#>
```

`[no] enable`

Description

Enable or disable ‘shell’ access from remote SSH connections

Options

Option	Description
`no`	disable ‘shell’ access from remote SSH connections (default)

`[no] audit-log [interval [VALUE] [seconds|minutes|hours|days]]`

Description

Enable or disable continuous audit log event when remote shell access is enabled. The continuous audit log event will log a message to the audit log every set interval.

Default: Enabled with 30 minutes interval.

Options

Option	Description
`interval [VALUE] seconds|minutes|hours|days`	Configure the time interval for logging audit-log messages.
`no`	disable continuous audit log

`[no] warning`

Description

Enable or disable the persistent warning when remote shell access is enabled.

Options

Option	Description
`no`	disable persistent warning

Troubleshooting

Occasionally when initiating an SSH connection, you may get a warning like this:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:lhQjYbBcWVZY2Zqmm1HizFx0M2opgTsBFNXmja161KE.
Please contact your system administrator.
Add correct host key in /home/foo/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/foo/.ssh/known_hosts:58
remove with:
  ssh-keygen -f "/home/foo/.ssh/known_hosts" -R 192.168.122.89

ECDSA host key for 192.168.122.89 has changed and you have requested strict checking.
Host key verification failed.
```

The warning is displayed when the local SSH client tries to look up the SSH server's public key in the `known_hosts` database and finds a different key associated with the same IP address + port combination.

This happens for example when:

- The device running WeOS is replaced with another device while the same IP address is retained.

- The IP address is changed to one which was previously associated with a different physical device.
- A factory reset is performed on the WeOS device. This will cause a new key to be generated when the device is restarted.

In any of these valid cases, the solution is to remove the invalid database entry causing the warning. The command to run was printed as part of the the warning message:

```
operator@example:~$ ssh-keygen -f "/home/foo/.ssh/known_hosts" -R 192.168.122.89
```

If none of the above valid reasons for a mismatch in the “IP address to host key” relation applies, it is possible you are the target of a man-in-the-middle attack where someone is trying to intercept the communication with the WeOS device.

If so, do not invalidate the database entry as explained above. Instead take appropriate action to report the incident and secure communication.

2.1.6. Telnet

Introduction

Telnet is a protocol to establish remote access to a unit over an IP network. As Telnet does not provide adequate security primitives it is disabled by default and only provided for legacy purposes. For remote access, it *recommended* to instead use SSH.

If Telnet is enabled on a switch, Telnet can be used to connect to the switch CLI remotely.

Connecting to a Device

To connect to a switch using Telnet, the Telnet service must first be enabled on the switch. This can be done via another management interface, e.g., via the console.

```
example:/#> configure
example:/config/#> management
example:/config/management/#> telnet
Activating Telnet with default settings, type 'abort' to cancel.
example:/config/management/telnet/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

To connect to the Telnet service you need a Telnet client. One example is the PuTTY[1] client on Windows, which also supports SSH. As stated earlier, use of SSH is recommended over Telnet for remote access. In the examples below a Linux host with its telnet client is used.

Connecting can be done to the assigned DNS name, the advertised mDNS name, or using the IP address of the remote device:

```
user@linux-pc:~$ telnet weos.local
Trying 192.168.2.33 ...
Connected to weos.local.
Escape character is '^]'.

example login : admin
Password: *******
```

Once logged in, the remote system's message of the day (MOTD) and the CLI prompt is displayed. Use `logout`, `exit`, or `Ctrl-D` to log out and disconnect.

Configuration

The Telnet service can be configured in several ways:

- It can be activated/deactivated
- It can be restricted to a subset of available interfaces
- The port number can be changed

There is currently no specific setting for Telnet idle timeout. Please set the general CLI inactivity timeout when enabling Telnet, to ensure Telnet users are logged out automatically if not active.

The Telnet service settings are available from the management configuration context in the CLI:

```
example:/#> configure
example:/config/#> management
```

[1] https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

```
example:/config/management/#> telnet
example:/config/management/telnet/#>
```

`[no] enable`

Description

Enable/disable the Telnet server. If disabled, the rest of the configuration is saved for when you re-enable the service again.

The telnet service configuration can be completely reset from the parent `management` configuration context using `no telnet`.

Options

Option	Description
`no`	Disable Telnet service completely (default)

`[no] listen IFNAME[,IFNAME]`

Description

Interfaces on which to allow Telnet connections.

To restrict Telnet to only listen on interfaces vlan1 and vlan2:

```
listen vlan1 vlan2
```

To listen on all interfaces again (default):

```
no listen
```

`[no] port PORT`

Description

By default, Telnet listens on TCP port 23. Telnet clients connect to this port unless another port is specified.

> **NOTE**
>
> To connect to a Telnet server listening to a non-standard port, the port has to be specified when connecting. For example:
>
> ```
> user@linux-pc:~$ telnet weos.local 1234
> ```

2.1.7. PROFINET

> **NOTE**
>
> See product pages at www.westermo.com[1] for information on PROFINET certified WeOS versions and products. Furthermore, PROFINET support requires an add-on to be installed. Contact your regular Westermo sales channel for more information.

WeOS provides PROFINET™ according to conformance class B (CC-B). When PROFINET support is enabled in WeOS, it is assumed that all unit management is done via PROFINET (TIA portal), i.e., no configuration via CLI, Web, etc.

PROFINET Adaptions of Default WeOS Behaviour

Factory Default Configuration

The factory default settings are adapted for units with PROFINET enabled to allow them to be discovered via PROFINET and for the PROFINET factory reset functionality to work properly.

- PROFINET settings:
  - PROFINET is enabled by default
  - The PROFINET interface is set to ‘vlan1’
- Alarm system:
  - A PROFINET trigger is defined in the alarm system
- Interface settings:
  - Interface ‘vlan1’ has address method ‘inet link-local’ (only). That is, no ‘inet dhcp’ for vlan1 in the PROFINET factory default.
- VLAN settings:
  - IGMP snooping is disabled by default on VLAN 1
- MRP:
  - MRP is enabled in client mode (MRC) and MRP profile ‘200’.
  - The lowest ports (e.g., ethX1 and ethX2 on Viper-212A) are defined as ring ports.

PROFINET for Discovery, Configuration and Status Retrieval

PROFINET can be used to discover and manage WeOS devices. The added management interfaces include PROFINET DCP (link-layer protocol) as well as IP/UDP PROFINET DCE access.

In particular, if the WeOS firewall is enabled, please ensure to add adequate *input allow* rules for PROFINET DCE (UDP RPC). Default UDP port is 34964 (0x8894), and additional UDP ports are opened starting from 49152 (0xC000).

In the factory default setting, PROFINET interface is ‘vlan1’ and IGMP is disabled on the associated VLAN (VLAN 1). If IGMP snooping is enabled on the VLAN of the PROFINET interface, PROFINET group MAC addresses need to be added to FDB.

IP Settings

WeOS allows PROFINET DCP to set a (static) IPv4 address/netmask on the defined PROFINET interface (default ‘vlan1’). When the address is set via PROFINET DCP, any existing static IP address(es) on the PROFINET interface will be removed.

PROFINET DCP can also be used to the unit’s default IP gateway.

[1] https://www.westermo.com

MRP Settings

The factory default setting holds an MRP instance. PROFINET DCE can update the MRP mode (MRM or MRC) and MRP ring ports. It can also create a second MRP instance and its associated settings.

Ethernet Port Settings

PROFINET DCE can update the admin status (enabled/disabled) of Ethernet ports. It can also update the speed/duplex and auto-negotiation settings.

LLDP Behaviour and TLV Content

The content of some of WeOS regular LLDP TLVs are adapted for PROFINET capable WeOS units:

- Chassis Id TLV Subtype (and Value): Locally Assigned Subtype
- Port Id TLV Subtype (and Value): Locally Assigned Subtype
- TTL TLV Value: 20 sec
- Port Description TLV Value: Port Name, e.g., ethX1

In addition, the LLDP packets hold PROFINET specific TLVs.

The LLDP transmission interval is 5 seconds for PROFINET capable units.

Use of IEC 61375 TTDP Not Supported

IEC 61375 is not supported on PROFINET capable WeOS units. That is, the WeOS unit cannot run PROFINET and also act as IEC 61375 ETBN. Although it may be possible to enable TTDP on PROFINET units, this is not supported and the behaviour is undefined.

Managing PROFINET

PROFINET is activated on the device by simple toggling it on, and providing it with a VLAN interface to operate on.

Configuration

In order to enable PROFINET from the CLI access the configuration context in the following manner:

```
viper-26-d0-80:/#> configure
viper-26-d0-80:/config/#> profinet
Activating PROFINET with default settings, type 'abort' to cancel.
profinet: Invalid settings: No interface specified.

viper-26-d0-80:/config/profinet/#>
```

As can be seen by the prompt, we need to specify the VLAN interface that profinet will serve. By default the interface **vlan1** will be available on the device. It is specified as follows:

```
viper-26-d0-80:/config/profinet/#> iface vlan1
viper-26-d0-80:/config/profinet/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
viper-26-d0-80:/#>
```

PROFINET should now be running, this can be verified by typing the following command:

```
viper-26-d0-80:/#> show profinet
PROFINET Enabled, running as PID 6805
Ready for AR connection: No, missing station name, missing IP address
No Active Application Relationships
```

```
Profinet configuration
MAC Address   : 00:11:b4:26:d0:80
Station-name  :
IP Address    : 0.0.0.0
Netmask       : 0.0.0.0
Gateway       : 0.0.0.0

Device-vendor: Viper-212A
Vendor ID     : 0×05c5
Device ID     : 0×5300
```

> **NOTE**
>
> Optionally the **station name** can be configured directly on the device. It is also possible to set the IP address of the provided VLAN interface directly, if it is not set using DCP later on.

Status Tools on the Device

The device also provide a number of CLI commands that can be used to gauge specific profinet information from the device during operation. These can be useful for debugging purposes, or simply if a user wants to check the operational status of profinet from a non-engineering tool.

The profinet CLI commands can be found in the following context:

```
viper-26-d0-80:/#> profinet
viper-26-d0-80:/profinet/#> help
show config          Show basic profinet configuration info
show ar-connections  Show AR Connection(s) information
show states          Show AR Connection(s) state information
show submodules      Show submodules information
show scheduler       Show profinet scheduler information
show diagnosis       Show profinet diagnosis
show data-check      Show profinet data check configuration
show data-adjust     Show profinet data adjust configuration
show im-records      Show profinet Identification and Maintenance (I&M) Records
show                 Show a summary of the profinet settings
dcp                  Performing a DCP Identify scan and list
_____________________________________________________________________________
See "help all" or "help COMMAND" for more online help.
viper-26-d0-80:/profinet/#>
```

GSDML File Download From the Device

The GSDML file for the associated device can be directly obtained from the device itself. In order to download the file use the Web interface and navigate to **Configuration** -> **System** -> **Profinet** and click the download button.

PROFINET PKG File

In order for the device to know that it should be able to run PROFINET, a special PKG file is applied on the device. By default the provided device should come with such a PKG file already present on the device.

As mentioned, the PKG instructs the device that it is capable of running PROFINET. Further, it also provides the device with its specific PROFINET factory configuration file. In addition, it also includes the correct GSDML file for the device.

> **NOTE**
>
> This is not a software PKG, it just includes configuration files.

Updating the PROFINET PKG File

It may be necessary to upgrade the PROFINET PKG file. This is something that can easily be done from the Web Interface. Navigate to **Maintenance** -> **PKG**.

> **NOTE**
>
> Important, ensure to remove/purge the current PROFINET PKG file, if it exist, before applying the new one. To do this in the web click the *Trash can* icon.

Click **Import**, a new window will show, select **Browse** and navigate to the location of the PKG file, then press **apply**.

> **NOTE**
>
> It is best to do a factory reset (see below) of the device after a new PROFINET PKG file has been applied.

Factory Reset

A factory reset can be performed in a number of different ways as described in the Factory Reset section. PROFINET provides an additional method: sending a PROFINET *DCP Set* with *Option 5* (Control) and *suboption 6* (Factory reset). This will perform a full factory reset of the device and reboot, in the same manner as the CLI and Web methods.

2.1.8. Limit number of simultaneous remote management sessions

It is possible to configure the number of allowed simultaneous remote sessions (SSH, Telnet and Web). By default, the limit will be *disabled*. Login via console or SNMP access are not considered in this limit. If a limit is set, it is strongly recommended to set inactivity timeouts for all enabled remote login methods, to avoid locking out users.

In the example below, simultaneous remote sessions are limited to three, and inactivity timeouts are set for Web (5 minutes) and SSH (300 seconds), assuming Telnet is disabled.

```
example:/#> configure
example:/config/#> management
example:/config/management/#> max-remote-sessions 3
example:/config/management/#> web
example:/config/management/web/#> session-timeout 5
example:/config/management/web/#> end
example:/config/management/#> ssh
example:/config/management/ssh/#> idle-timeout 300
example:/config/management/ssh/#>
```

With this setting and three active sessions (e.g., one Web access and two SSH session), an attempt to open another SSH session will be rejected.

Syntax

The limitation of remote sessions service is available from the management configuration context in the CLI:

`[no] max-remote-sessions <1..31>`

Description

Set or disable the limitation of simultaneous remote sessions.

2.1.9. Simple Network Management Protocol

The Simple Network Management Protocol is an application layer protocol for collecting and handling information about managed devices in IP networks. SNMP is used for network management and network monitoring. The data in SNMP is exposed in variables organised in management information bases, called MIB:s.

Overview

In WeOS 5, SNMP act as an agent if enabled and supports version SNMPv2c and SNMPv3 of the protocol. WeOS 5 also supports up to 8 users and 8 trap/inform hosts.

To use SNMPv3 traps/informs, the engine id must be enabled first and at least one SNMPv3 user created before adding a traphost.

Write Support

Currently WeOS does not have full write support, only a few very select OIDs are writable. So in general it is best to consider most of the MIBs as read-only.

Supported MIBs

A number of standard MIBs are supported along with a few private MIBs.

> **DETAILED INFORMATION**
>
> For more detailed information about the MIB support, refer to the included MIB and conformance files in any downloaded WeOS Release's zip-archive.
>
> The conformance files lists the deviations and variations from the MIB files if any.

Standard MIBs

MIB and Base OID(s)	Description	Note
BRIDGE-MIB `.1.3.6.1.2.1.17`	MIB for devices supporting IEEE 802.1D. Note: In dot1dTpFdb, if more than one entry is found for a mac address, the one with the lowest ifindex will be the one present in the table.	Supports *dot1dBase*, *dot1dStp* and *dot1dTp*.
ENTITY-MIB `.1.3.6.1.2.1.47`	Physical entity MIB. Used for managing and monitoring physical components of network devices, such as chassis, modules, and sensors. Can be used to read serial number, firmware version, etc…	Supports *entityPhysical*.
ENTITY-SENSOR-MIB `.1.3.6.1.2.1.99`	MIB for monitoring physical sensors in network devices, such as temperature, voltage, and fan speed.	-

MIB and Base OID(s)	Description	Note
EtherLike-MIB `.1.3.6.1.2.1.10.7`	MIB for managing and monitoring Ethernet-like interfaces.	Supports *dot3StatsTable*.
HOST-RESOURCES-MIB `.1.3.6.1.2.1.25`	MIB for managing and monitoring host resources, such as CPU, memory, and storage.	Supports *hrSystem*.
IEC-62439-2-MIB `.1.0.62439.1`	MIB for managing and monitoring the Media Redundancy Protocol (MRP) as defined by the IEC 62439-2 standard.	-
IEC-62439-3-MIB `.1.0.62439.2`	MIB for managing and monitoring the Parallel Redundancy Protocol (PRP) as defined by the IEC 62439-3 standard.	Supports *lreConfigurationGeneralGroup*, *lreConfigurationPortsGroup*, *lreStatisticsGeneralGroup*
IEEE8023-LAG-MIB `.1.2.840.10006.300.43`	MIB for managing and monitoring Link Aggregation Groups (LAGs) as defined by the IEEE 802.3ad standard.	Supports *dot3adAgg* nad *dot3adAggPort*.
IF-MIB `.1.3.6.1.2.1.2`, `.1.3.6.1.2.1.31`	MIB for managing and monitoring network interfaces.	Supports *ifTable*, *ifXTable*.
LLDP-MIB `.1.0.8802.1.1.2`	MIB for managing and monitoring the Link Layer Discovery Protocol (LLDP).	Partial Support for *lldpConfiguration*, *lldpPortConfigTable*, *lldpStatistics*, *lldpLocalSystemData* and *lldpExtensions*.
MAU-MIB `.1.3.6.1.2.1.26`	MIB for managing and monitoring Media Attachment Units (MAUs).	Supports *dot3IfMauBasicGroup* and *dot3IfMauAutoNegGroup*
POWER-ETHERNET-MIB `.1.3.6.1.2.1.105`	MIB for managing and monitoring Power over Ethernet (PoE) devices.	Supports *pethPsePortGroup* and *pethMainPseGroup* for applicable devices.
Q-BRIDGE-MIB `.1.3.6.1.2.1.17.7`	MIB for managing and monitoring VLANs in bridged networks.	Supports *dot1qBase*, *dot1qFdbTable*, *dot1qTpFdbTable*, *dot1qVlanCurrentTable* and *dot1qVlanStaticTable*
RFC1213-MIB `1.3.6.1.2.1`	The original MIB-II standard for network management. Provides basic management information for TCP/IP-based	Supports *system*, *snmp*, *interfaces*, *at*, *ip*, *icmp*, *tcp* and *udp*.

MIB and Base OID(s)	Description	Note
	networks, including system, interface, and protocol statistics.	
RMON-MIB `.1.3.6.1.2.1.16`	MIB for Remote Network Monitoring (RMON).	Supports *etherStatsTable*.
RSTP-MIB `.1.3.6.1.2.1.17.2.16`, `.1.3.6.1.2.1.17.2.17`, `.1.3.6.1.2.1.17.2.19`	MIB for managing and monitoring Rapid Spanning Tree Protocol (RSTP).	Supports *rstpBridgeGroup* and *rstpPortGroup*
SNMP-FRAMEWORK-MIB `.1.3.6.1.6.3.10`	MIB for managing and monitoring the SNMP framework itself.	Supports *snmpEngine*.
SNMP-MPD-MIB `.1.3.6.1.6.3.11`	MIB for managing and monitoring the SNMP Message Processing and Dispatching (MPD) subsystem.	Supports *snmpMPDStats*.
SNMP-USER-BASED-SM-MIB `.1.3.6.1.6.3.15`	MIB for managing and monitoring SNMPv3 user-based security models.	Supports *usmStats* and *usmUser*.
SNMPv2-MIB `.1.3.6.1.2.1.1` `.1.3.6.1.2.1.11`	MIB for managing and monitoring SNMPv2 entities.	Supports *system* and *snmp*.
SNMP-VIEW-BASED-ACM-MIB `.1.3.6.1.6.3.16`	MIB for managing and monitoring SNMP view-based access control models.	Supports *vacmContextTable*, *vacmSecurityToGroupTable*, *vacmAccessTable* and *vacmMIBViews*
TRDP-MIB `.1.0.61375.2.1.1`	Train Real-Time Data Protocol (TRDP) MIB, part of the IEC 61375.	Supports *trdpGenGroup*, *trdpMemGroup*, *trdpPdGroup*, *trdpMduGroup*, *trdpMdtGroup* and *trdpRedGroup*
TTDB-MIB `.1.0.61375.2.3`	Train Topology DataBase (TTDB) MIB, part of the IEC 61375.	Supports *ttdbBasicGroup* and *ttdbOpVehListGroup*
TTDP-MIB `.1.0.61375.2.5`	Train Topology Discovery Protocol (TTDP) MIB, part of the IEC 61375.	Supports *ttdpBasicGroup* and *ttdpStatsGroup*
UCD-SNMP-MIB `.1.3.6.1.4.1.2021`	MIB for managing and monitoring system performance and resource utilisation.	Supports *laTable* and *memory*
VRRP-MIB `.1.3.6.1.2.1.68`	MIB for managing and monitoring Virtual Router Redundancy Protocol (VRRP).	Supports *vrrpOperGroup* and *vrrpStatsGroup*

MIB and Base OID(s)	Description	Note
VRRPV3-MIB `.1.3.6.1.2.1.207`	MIB for managing and monitoring Virtual Router Redundancy Protocol version 3 (VRRPv3).	Supports *vrrpv3OperationsGroup* and *vrrpv3StatisticsGroup*

Private MIBs

These MIBs are specific to Westermo devices. You can find the MIB files in the zip archive of the WeOS release you have downloaded. Afterwards you could load them into your SNMP management software.

MIB and Base OID(s)	Description
WESTERMO-COMMAND-MIB `.1.3.6.1.4.1.16177.2.7`	MIB for managing and monitoring Westermo specific commands over SNMP, such as reboot and factory reset.
WESTERMO-DDM-MIB `.1.3.6.1.4.1.16177.2.1`	MIB for managing and monitoring Westermo's Digital Diagnostic Monitoring (DDM) features.
WESTERMO-EVENT-MIB `.1.3.6.1.4.1.16177.2.3`	MIB for managing and monitoring Westermo specific events and alarms.
WESTERMO-FRNT-MIB `.1.3.6.1.4.1.16177.2.5`	MIB for managing and monitoring Westermo's FRNT (Fast Reconfiguration of Network Topology) feature.
WESTERMO-INTERFACE-MIB `.1.3.6.1.4.1.16177.2.4`	The MIB lists all ports and interfaces and their reference index. The purpose of this reference index is to have a predictable index for ports and interfaces.
WESTERMO-RICH-MIB `.1.3.6.1.4.1.16177.2.10`	MIB for managing and monitoring Westermo's Ring Chaining (RiCh) protocol feature.
WESTERMO-RICO-MIB `.1.3.6.1.4.1.16177.2.8`	MIB for managing and monitoring Westermo's Ring Coupling Protocol (RiCo) feature.
WESTERMO-SYSTEM-MIB `.1.3.6.1.4.1.16177.2.9`	MIB for managing and monitoring Westermo specific system information.

SNMP Notifications (Traps/Informs) Support

Trap/Inform	Included data	Description
`Summary`	System location, system name, date, summary status (OK or warning)	Sent when the overall alarm summary changes between OK and warning.
`Link`	Interface index, administrative status, operational status, interface-change uptime, message	Sent when an interface goes up or down.
`Power`	Event ID, event type, event status, message, system information, date, power sensor name and value	Sent when a power sensor changes between OK and warning.

Trap/Inform	Included data	Description
`Temp`	Event ID, event type, event status, message, system information, date, temperature sensor name, value, scale, precision	Sent when a temperature sensor changes between OK and warning.
`Ring`	Event ID, event type, event status, message, system information, date	Sent when the FRNT or MRP ring status changes between OK and warning.
`Ping`	Event ID, event type, event status, message, system information, date	Sent when a ping monitor changes between OK and warning.
`Storm`	Event ID, event type, event status, message, system information, date	Sent when storm detection changes between OK and warning.
`Health monitor`	Event ID, event type, event status, message, system information, date	Sent when a health monitor changes between OK and warning.
`RiCH`	RiCh topology count, topology-count status	Sent when the RiCh topology count becomes valid or invalid.
`SNMPd coldstart`	Standard SNMP coldStart notification data	Sent when the SNMP agent starts.
`SNMPd shutdown`	Standard SNMP shutdown notification data	Sent when the SNMP agent shuts down.
`lldpRemTablesChange`	Insert, delete, age-out, drop counters, local port, remote system name and port description (when available)	Sent when a remote LLDP neighbor is added, updated, or deleted.
`arpInfoNewStation`	IP address, newly observed MAC address, interface name	Sent when a new station is detected.
`arpInfoFlipFlop`	IP address, current and previous MAC addresses, interface name	Sent when an IP address changes between MAC addresses.
`arpInfoChanged`	IP address, current and previous MAC addresses, interface name	Sent when an ARP mapping changes.
`arpInfoEthernetBroadcast`	IP address, Ethernet MAC address, interface name	Sent when an Ethernet broadcast is detected.
`arpInfoBogon`	IP address, MAC address, interface name	Sent when a bogon address is detected.
`arpInfoEthernetMismatch`	IP address, Ethernet MAC address, ARP MAC address, interface name	Sent when the Ethernet and ARP MAC addresses do not match.
`arpInfoReusedAddress`	IP address, MAC address, interface name	Sent when an address is reused.

Trap/Inform	Included data	Description
`arpInfoIPBroadcast`	IP address, MAC address, interface name	Sent when an IP broadcast is detected.
`arpInfoAddressConflict`	IP address, MAC address, interface name	Sent when an address conflict is detected.

Configuration

Enter SNMP context

```
example:/#> configure
example:/config/#> management
example:/config/management/#> snmp
example:/config/management/snmp/#>
```

Syntax

`[no] enable`

Description

This will enable or disable SNMP.

Options

Option	Description
`no`	Disable SNMP.

`[no] rocommunity <STR>`

Description

This setting controls the read-only community for SNMPv2.

Options

Option	Description
`no`	Disables the read-only community for SNMPv2.

`[no] rwcommunity <STR>`

Description

This setting controls the read-writable community for SNMPv2.

Options

Option	Description
`no`	Disables the read-writable community for SNMPv2.

`[no] trapcommunity <STR>`

Description

This setting controls the trap community for SNMPv2.

Options

Option	Description
`no`	Set the trap community for SNMPv2 back to default `trap`.

`[no] ifindex-persistence`

Description

This setting controls if the ifindex persistence is enabled or disabled.

Options

Option	Description
`no`	Disables the ifindex persistence.

`[no] engine-id [auto|custom <HEXSTR>]`

Description

This setting controls the engine-id, which is needed for sending SNMPv3 traps. This engine-id is only used for sending v3 traps. Get, set, walk etc. uses an engine-id that starts with the internal Net-SNMP prefix 80001f88 instead of 80003f31 followed by 03 and the MAC from br0.

Options

Option	Description
`no`	Disables the engine-id.
`auto`	Sets the engine-id value based on the base MAC address and Westermo's enterprise number.
`custom`	Sets the engine-id to a custom chosen hexadecimal string which can be between 10 (5 octets) and 64 (32 octets) hex digits long.

`[no] listen <IFACE> [<IFACE>...]]`

Description

This setting controls which interface(s) the SNMP daemon will listen to, all, some interfaces or none.

Options

Option	Description
`no`	SNMP daemon will not listen to any interface.

`[no] commands-allowed [reboot, factory-reset]`

Description

This setting enables or disables the one or both of the two commands that are possible to use over SNMP. This means that it's possible to reboot or factory reset a device on distance over SNMP if allowed here. The commands have a fixed order in the SNMP commands table where the reboot command have the index 1 and factory reset have the index 2.

Options

Option	Description
`no`	Disables one or all of the commands.
`reboot`	Enables or disables reboot of the device over SNMP.
`factory-reset`	Enables or disables factory-reset of the device over SNMP.

```
[no] host <IP-ADDR[:PORT]|[IPv6-ADDR]:PORT|FQDN[:PORT]> [version <version> [username
<USERNAME>]] [type trap|inform [resends <NUM>]]
```

Description

Manage SNMP trap sink host definitions.

Use this command to add or remove trap host definitions. These host definitions dictate how the system generates SNMP notifications (traps or informs) when such events occur, specifically where and how to send them. As such, this command only configures where and how any SNMP notifications will be sent; it does not configure which events will be generated.

For instance, an alarm configured with an SNMP action will attempt to generate a SNMP notification event, but if no host has been configured with this command, no SNMP notification can be sent anywhere.

Therefore, any part of the system that may support generating SNMP notifications for a given event needs to have at least one trap host configured for any SNMP notifications to actually be sent.

> **NOTE**
>
> Up to eight trap hosts can be configured.

Default values:

- If no version is specified, the default is 2.
- If no type is specified, the default is trap.
- If no resend value is specified for informs, the default is 5.
  - Resends are only relevant for informs.

Example

```
example:/config/management/snmp/#> host 192.168.1.1
example:/config/management/snmp/#> host 192.168.1.2:1620 version 2 type trap
example:/config/management/snmp/#> host 192.168.1.3 version 2 type inform resends 3
```

continues on next page

continued

```
[no] host <IP-ADDR[:PORT]|[IPv6-ADDR]:PORT|FQDN[:PORT]> [version <version> [username
<USERNAME>]] [type trap|inform [resends <NUM>]]
```

```
example:/config/management/snmp/#> host 192.168.1.4 version 3 type inform user JohnDoe
example:/config/management/snmp/#> host snmp.example.com version 2 type trap
```

Options

Option	Description
`no`	Disables one or all traphosts.
`version`	SNMP version as a figure 2 or 3. Default is 2.
`username`	In case of version 3, a username of a user present in the SNMP configuration.
`type`	The type of message to send to the traphost, trap or inform. Default is trap.
`resends`	Number of times to resend the message in case of no ack received from the recipient. 0-60 number of resends is supported. Only available for informs. Default number of resends is 5.

```
[no] rwuser <USERNAME> [auth <md5|sha1|sha224|sha256|sha384|sha512> <PASSPHRASE>
[crypto <des|aes128|aes192|aes256|aes192c|aes256c> <PASSPHRASE>]] [OIDTREE]
```

Description

This setting manages read-writable users.

Options

Option	Description
`no`	Disables one or all read-writable users.
`auth`	Set the authentication encryption type and required password for the authentication.
`crypto`	Set the encryption type and required password for the encryption.
`oidtree`	Restricts access for the user to the subtree rooted at the given OID. Default is full access.

```
[no] rouser <USERNAME> [auth <md5|sha1|sha224|sha256|sha384|sha512> <PASSPHRASE>
[crypto <des|aes128|aes192|aes256|aes192c|aes256c> <PASSPHRASE>]] [OIDTREE]
```

Description

This setting manages read-only users.

Options

Option	Description
`no`	Disables one or all read-only users.
`auth`	Set the authentication encryption type and required password for the authentication.
`crypto`	Set the encryption type and required password for the encryption.

continues on next page

continued

```
[no] rouser <USERNAME> [auth <md5|sha1|sha224|sha256|sha384|sha512> <PASSPHRASE>
[crypto <des|aes128|aes192|aes256|aes192c|aes256c> <PASSPHRASE>]] [OIDTREE]
```

Option	Description
`oidtree`	Restricts access for the user to the subtree rooted at the given OID. Default is full access.

Examples

Add a read-writable user with authentication and crypto

```
example:/#> configure
example:/config/#> management
example:/config/management/#> snmp
example:/config/management/snmp/#> rwuser JohnDoe auth sha384 Secret1 crypto aes192 Secret2
```

Add a host for version 3 informs with the user created above

```
example:/#> configure
example:/config/#> management
example:/config/management/#> snmp
example:/config/management/snmp/#> host 1.2.3.4 version 3 type informs user JohnDoe
```

Add a host for version 2 traps

```
example:/#> configure
example:/config/#> management
example:/config/management/#> snmp
example:/config/management/snmp/#> host snmp.example.com version 2 type trap
```

2.1.10. Metrics

Introduction

The metrics provides the time sensitive data about the system cpu(s), memory and port(s) at regular interval.

Sample rate

The sample rate in seconds determines the interval at which the metrics is collected and updated. The default sample rate is 20 seconds

CPU

The system provides the current **cpu utilization** (load) for each cpu core and summary of utilisation in total.

Memory

The system provides the following memory metrics

- Memory Metrics:
  - Total Memory (kB)
  - Used Memory (kB) and in percentage of total memory
  - Free Memory (kB) and in percentage of total memory
  - Available Memory (kB) and in percentage of total memory

Ports

The system provides the following memory metrics

- Port Metrics:
  - Speed (Mbit/s) and duplex full or half
  - Inbound bitrate per second
  - Outbound bitrate per second
  - Utilisation

Port-utilisation

By default the port metrics are not enabled and it needs to be enabled to collect and present port metrics

Web

The metrics can be viewed in open metrics format by enabling the web option. The default port is 9100. It is possible to configure ports between 1024 and 65535.

> **WARNING**
>
> Metrics data available on the Web works only if https is enabled in the management

SNMP

Enables metrics data to be available via SNMP request

Configuration

```
example:/#> configure
example:/config/#> metrics
```

Syntax

`[no] enable`

Description

Enable/Disable metrics.

Options

Option	Description
`no`	Disable.

`[no] port-utilization PORT`

Description

Enable collection and presentation metrics on the PORT

Options

Option	Description
`no`	disable metrics on the PORT
`PORT`	The port that metrics data will be collected and presented.

`[no] sample-rate SAMPLE-RATE`

Description

Set the interval at which metrics data are collected and updated

Options

Option	Description
`no`	set the sample rate to default
`SAMPLE-RATE`	The interval in seconds

`[no] web PORT-NO`

Description

Enable metrics data to be available via PORT-NO

Options

Option	Description
`no`	Disable metrics data on web
`PORT-NO`	The port number the metrics data is available

© 2026 Westermo Network Technologies AB

`[no] SNMP`

Description

Enable metrics data to be available via SNMP request

Options

Option	Description
`no`	Metrics data will not be available via SNMP request

Status Cpu Utilisation

```
example:/#> show metrics cpu-utilization
CPU utilisation
  cpu      0.28 %
  cpu0     0.28 %
```

Status Memory Utilisation

```
example:/#> show metrics memory-utilization
Memory
  MemoryTotal        103192.00 kB
  MemoryUsed          95168.00 kB  (92.00 %)
  MemoryFree           8024.00 kB  (7.00 %)
  MemoryAvailable     37452.00 kB  (36.00 %)
```

Status Port Utilisation

```
example:/#> show metrics port-utilization
Port     Speed        Inbound          Outbound        Utilisation
Eth1     100M-Full    n/a              n/a             n/a
Eth2     ---------    n/a              n/a             n/a
Eth3     100M-Full    36.3 Mbit/s      2697.6 kbit/s   18%
Eth4     100M-Full    6051.0 kbit/s    2697.6 kbit/s   3%
```

2.2. AAA and User Management

2.2.1. Authentication, Authorisation and Accounting

Introduction

Authentication, Authorisation and Accounting (AAA) is an umbrella term referring to protocols and methods that can be used to handle and verify user access.

The three individual terms refer to who is allowed to access what device (Authentication), if access is granted what are the limitations of their available actions (authorisation), and to recount what actions they performed (accounting).

For some use-cases utilising the systems AAA options, refer to the following:

- Built-in Accounts
- Local Databases HowTo
- Remote Servers HowTo

The SSL security level is configured globally under `/config/system/ssl-security-level`, since it applies to the system's LDAP connection rather than to an individual remote-server entry. See System Settings.

Overview

AAA in the system is separated into two different parts that are combined together. Firstly the system allows for setting up different types of AAA methods. Afterwards, the configured AAA method can be bound to various services in the system that have support for AAA methods. The best example of this is the login service.

Authentication Chain

An authentication chain is a configurable sequence of authentication methods that the system uses to verify user credentials during login attempts. Each method in the chain is evaluated in order, and the process continues until a method successfully authenticates the user or all methods have been exhausted. The chain can include remote servers, server groups, and built-in accounts, with options to control behaviour on authentication rejection and fallback to local accounts. This approach provides flexibility and redundancy, ensuring reliable access control even if some authentication sources are unavailable. Below is an example of an authentication chain that includes two remote servers, a server group, and the built-in accounts as fallback. The chain is configured to try the next method on rejection and to fallback to local accounts if all methods are unavailable.

```
+-----------------+     +-----------------+     +----------------+     +-------------------+
| Remote Server 0 | --> | Remote Server 1 | --> | Server Group 0 | --> | Built-in Accounts |
+-----------------+     +-----------------+     +----------------+     +-------------------+
     |                   |                       |
     v                   v                       v
 [on-reject: next]       [on-reject: next]    [local-fallback: always]
```

Figure 1: Example AAA authentication chain showing Remote Server, Server Group, and Built-in Accounts with fallback and reject handling.

Authentication Methods

The AAA framework in the system have support for a number of different AAA methods that are presented below:

- **Built-in Accounts**: The only account present in a factory default system. As an example, the *admin* user is one such an account. Built-in accounts allow for defining per account authorisation

levels. For more information see Built-in Accounts HowTo. The built-in login methods are implicit and are always last in the authentication chain. For more info see `local-fallback`.

- **Remote Servers**: Centralized authentication on remote servers via RADIUS, TACACS+, and LDAP. For additional information see Remote Servers HowTo. **NOTE:** The device can be configured to use multiple remote servers, but none will be used unless they are added to the authentication chain. Only one LDAP server entry can be added to the chain, but it can have a primary address and an optional secondary address for a redundant failover server. LDAP credentials are cached during login and hence allows login using cached credentials. The cache is removed at reboot.
- **Server Groups**: Allows grouping of multiple remote servers for authentication. This can be useful for load balancing and failover scenarios. For additional information see Remote Servers HowTo.
- **Local Database (Deprecated)**: Allows storing of credentials locally on the device, with no need for any external infrastructure.

  This functionality is considered **deprecated** and no longer supported for newly configured systems. Existing configurations will continue to work. For more information see Local Databases HowTo.

Services

The system currently supports AAA methods to be used for the following services:

Service	Remote Server	Local Database	Built-in Accounts
Login	X		X
OpenVPN	X	X	

OpenVPN

The systems AAA methods can be used to authenticate OpenVPN clients. The OpenVPN client will authenticate itself using a certificate, and optionally also via username and password. When using this option, a system acting as an OpenVPN Server, can either use a local database or a central server (or server group) to verify the client's username and password.

For an example of how this can be done refer to this HowTo:

- OpenVPN Host to LAN

Authorisation levels

A user should be granted the least possible privilege that allows performing that user's task. This is realised by a light-weight Role Base Access Control (RBAC) that is available for built-in accounts.

Modules

The functionality available to the users is split into multiple areas called RBAC modules. For CLI, each command will belong to one of these modules. The RBAC modules are:

- **Device status** for accessing the status of the device (show commands will mostly go here).
- **Audit trail** for accessing the audit log functionality.
- **PKI** for accessing Public Key Infrastructure functionality.
- **AAA** for accessing Authentication, Authorisation, and Accounting functionality.
- **Device config** for accessing device configuration context.
- **Utils/tools** for accessing utilities and tools (like ping, ssh, telnet etc).

- **Maintenance** for accessing maintenance tools (like copy, export, diff etc).
- **Firmware management** for accessing firmware management.
- **Factory reset** for performing factory reset operation.
- **Shell access** for accessing the Linux shell.

Permissions

For each RBAC module, the following **permissions** are defined:

Permission	Acronym	Definition
view	v	View on screen
read	r	Download reports where available
write	w	Update data
executing	e	Execute a command
update	u	Change firmware
shell	X	Enter Linux shell

Roles

Currently six roles are available:

Role	Description	Responsibility
Administrator	Full access to all device functions and configurations	• Configure security policies • Manage users and roles • Perform device configuration • Manage firmware upgrades • Manage backup and restore
Engineer	Permission to configure most functionality except security features like AAA, PKI and audit logs	• Identify and solve complex functional issues • Perform device configuration • Manage firmware upgrades • Manage backup and restore
Operator	Limited access to operational tasks, such as monitoring and basic configuration changes	• Identify and solve functional issues • Perform device control actions • Manage firmware upgrades
Auditor	Restricted access, primarily read-only, for viewing audit logs	• Monitor audit-log to identify potential security issues
Viewer	Restricted access, primarily read-only, for monitoring	• Identify and solve simple functional issues
Guest	**Deprecated**: Replaced by Viewer. Has read-only access to a restricted CLI	• Identify and solve simple functional issues

Built-in accounts and LDAP remote server can have multiple roles.

Role	LDAP	RADIUS	TACACS+	Local Database	Built-in Accounts
Administrator	X	X	X	X	X
Guest	X				X

Role	LDAP	RADIUS	TACACS+	Local Database	Built-in Accounts
Viewer	X				X
Operator	X				X
Engineer	X				X
Auditor	X				X

Roles and Permissions per Module

Each role will have a set of permissions on each RBAC module according to the following table:

Module	Administrator	Guest	Viewer	Operator	Engineer	Auditor
Device status	vr	v	v	vr	vr	v
Audit trail	vrwe	-	-	vr	vr	vr
PKI	vrwe	v	v	vr	v	vr
AAA	vrwe	v	v	vr	vr	vr
Device config	vrwe	-	-	vr	vrwe	-
Utils/tools	vrwe	-	-	vre	vrwe	-
Maintenance	vrwe	-	-	vre	vrwe	-
Firmware management	vrweu	-	-	vrweu	vrweu	-
Factory reset	vrwe	-	-	-	-	-
Shell access	Xvw	-	-	-	-	-

Configuration

The AAA configuration consists of a number of different areas spread over a number of different context levels.

General Settings

The AAA can be configured from the configuration context in the CLI. This context contains the overall AAA configuration and access to all other AAA sub-contexts.

```
example:/#> configure
example:/config/#> aaa
example:/config/aaa/#>
```

Syntax

`[no] user USERNAME`

Description

Create and mange a built-in account.

Options

Option	Description
`USERNAME`	Name of a new or already existing built-in user account
`no`	Removes specified user

`username NAME [password | hash] PASSWORD`

Description

Set the password for specific allowed users, like the **admin** user.

> ⚠ WARNING
>
> **DEPRECATED**: use `user admin` to enter *user-admin* context and `password` from within that context.

Options

Option	Description
`NAME`	The user to change the password for.
`PASSWORD`	Password with the following allowed characters: ASCII 33-126 (octal 041-176), max length 64 chars.

`[no] password-policy`

Description

Set password policy. For details read here.

`[no] lockout-policy`

Description

The lockout policy is a security feature that prevents unauthorized access to the system by locking out users who enter incorrect login credentials too many times. You can activate and configure the lockout policy in this sub-context.

Options

Option	Description
`enable`	Enable/disable account lockouts
`deny`	Number of consecutive authentication failures until lockout of the user. Default number is 3 times. Allowed values are 3 - 10 times.
`deny-admin`	If this option is selected even the built-in *admin* ackount can be locked out. Default is yes (enabled).
`unlock-time`	Duration of the lockout (s). Default is 600 seconds (10 minutes). Allowed values are 10 - 1800 seconds.
`[no] exclude <console>`	Always allow logins over the selected interface. Currently only console is selectable.

continues on next page

continued

`[no] lockout-policy`

> **NOTE**
>
> Lockout only happens if `deny` number of failed login attempts happens within 900 seconds. This value is currently hard-coded and not configurable.

> **NOTE**
>
> You can view the locked out users with the command *show lockout* in the *admin-exec* context and unlock a locked out user with the command *lockout <username> reset* in the same context.

`[no] auth-chain`

Description

Manage the authentication chain.

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`no`	Reset the authentication chain to defaults.

`[no] local-db ID [plain]`

Description

Create and manage local databases. Default type is ‘hash’ but the keyword ‘plain’ can be used to create a database that contains plain text passwords. The type can only be defined once when the local database is created, therefore the keyword is not available in the sub-configuration context.

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`no`	Remove a specific local database if an `ID` is provided, otherwise remove all local databases.
`ID`	Id in the range 0-4, any other id will be invalid.

`[no] remote-server ID [type <radius | tacacs | ldap>]`

Description

Create and manage remote authentication servers.

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`no`	Remove a specific remote server if an `ID` is provided, otherwise remove all servers.
`ID`	Id in the range 0-5, any other id will be invalid.
`type`	The type of remote server to create: `radius`, `tacacs`, or `ldap`. This setting can still be changed in the server configuration. Only one LDAP server entry can be added to the chain, but it can have a primary address and an optional secondary address for a failover server.
`encryption-type / connection-security`	The encryption type (RADIUS/TACACS) or connection security (LDAP) of the remote server. Available options depend on server type. For `radius`: `none`, `peap-mschapv2`. For `tacacs`: `none`. For `ldap`: `none`, `ldaps`, `ldap-starttls`.

`[no] server-group ID [type <radius | tacacs>]`

Description

Create and manage server groups.

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`no`	Remove a specific server group if an `ID` is provided, otherwise remove all server groups.
`ID`	Id in the range 0-1, any other id will be invalid.
`type`	The type of server group to create, either `radius` or `tacacs`. This setting can still be changed in the server group configuration.

`[no] mac-auth ID`

Description

Manage a MAC authentication group.

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`no`	Remove a specific MAC authentication group if an `ID` is provided, otherwise remove all MAC authentication groups.
`ID`	Id in the range 0-99.

Built-in Account Settings

This section describes configuration of built-in accounts. Up to 16 built-in accounts are allowed. Only built-in accounts can have guest / operator / auditor roles. An example of a built-in account that is always present in factory configured system is *admin*.

```
example:/config/aaa/#> user admin
example:/config/aaa/user-admin/#>
```

`[no] enable`

Description

Enable or disable this user in the system. A user cannot disable themselves.

> **NOTE**
>
> SSH Authorised Keys of a disabled user are removed and must be imported again upon enabling the user.

`[no] password [expire] [alg ALG] secret PASSWORD`

Description

Set password in clear text and hash it using given algorithm.

Options

Option	Description
`expire`	Expire password, forcing user to change it at next login.
`ALG`	`md5`, `sha256` or `sha512`. By default md5crypt (`md5`) is used if no algorithm is given. If the password-policy option is set the password must meet the policy.

continues on next page

continued

`[no] password [expire] [alg ALG] secret PASSWORD`

Option	Description
`PASSWORD`	Password with the following allowed characters: ASCII 33-126 (octal 041-176), max length 64 chars.
`no`	Disable password for the user. When issuing 'no password' command password-based authentication is disabled, making it impossible to log in using a username and password. However, if SSH keys exist, you can still log in using them. **NOTE** Please exercise caution when using 'no password', as it could potentially lock you out from the device if no other administrator accounts exist and no SSH keys are defined.

`[no] hash HASH-STRING`

Description

Set the user password in hash format.

> **NOTE**
>
> When the password-policy option is enabled this option is disabled (See here).

Options

Option	Description
`HASH-STRING`	An arbitrary hash-string. It should be used with caution. A cryptographically weak hash-string will compromise the system's security.
`no`	disable password for the user

`[no] role viewer | guest | operator | administrator | engineer | auditor`

Description

Set role of the user. Default role is viewer. There must be at least one administrator in the system.

Options

Option	Description
`administrator`	Has all privileges in the system.
`viewer`	Has permissions to view objects with-in the device and read values where applicable. [Default]

continues on next page

continued

`[no] role viewer | guest | operator | administrator | engineer | auditor`

Option	Description
`guest`	Has only access to a restricted interface with limited read-only functionality. Deprecated, replaced by viewer.
`operator`	Has permissions to read and view objects and values with-in the device and where applicable perform control actions (e.g. reboot) and update firmware.
`auditor`	Can read audit logs and other audit relevant information.
`engineer`	Can perform most of the configuration, except security related parts like AAA, PKI, audit logs.

`[no] ssh-user-key import`

Description

SSH user key management

Options

Option	Description

`import`	Import authorised public keys. Enter an interactive mode for pasting user's authorised public keys. One key per line, each key having format as used in OpenSSH i.e.:
	`<type-name> <base64-encoded-ssh-public-key> [comment]`
	Example:
	`ssh-rsa AAAAB3NzaC...G6KCGOnv user@host`
	Notes:
	• This command overwrites any existing keys.
	• Imported key(s) are not checked for validity.

`no`	Removes imported keys.

Examples:

For examples on how to create and manage built-in accounts see Built-in Accounts

Local Database Settings

Configure a local database and enter its configuration context. When creating a local database the `ID` must be a number from 0 to 4.

> **WARNING**
>
> This functionality is considered **deprecated** and no longer supported for newly configured systems. Existing configurations will continue to work.

```
example:/config/aaa/#> local-db 0
Creating new local db 0
example:/config/aaa/local-db-0/#>
```

Syntax

`[no] username NAME [password | hash] PASSWORD`

Description

Add a user to this local database.

Options

Option	Description
`no`	Remove a specific user based on the provided `NAME`.
`NAME`	The user to change the password for.
`PASSWORD`	Password with the following allowed characters: ASCII 33-126 (octal 041-176), max length 64 chars.

`[no] description STRING`

Description

Free form description of this local database.

Options

Option	Description
`no`	Remove any description.
`STRING`	Free form text with a maximum length of 15 characters.

Remote Server Settings

Configure a remote server and enter its configuration context. When creating a remote server the `ID` must be a number from 0 to 5.

> **NOTE**
>
> To use a remote server for authentication it must be added to the authentication chain.

```
example:/config/aaa/#> remote-server 0
Creating new remote server 0
example:/config/aaa/remote-server-0/#>
```

Syntax

`[no] type <radius | tacacs | ldap>`

Description

Set the remote server type to be `radius`, `tacacs`, or `ldap`.

continues on next page

continued

`[no] type <radius | tacacs | ldap>`

Options

Option	Description
`no`	Reset the type to its default setting, `radius`.

> **NOTE**
>
> Only one LDAP server entry can be added to the chain, but it can have a primary address and an optional secondary address for a failover server.

`[no] encryption-type <none | peap-mschapv2>`

Description

Set the encryption type for RADIUS/TACACS remote servers.

For `radius`: `none` (default), `peap-mschapv2`.

For `tacacs`: `none` (default).

`[no] connection-security <none | ldaps | ldap-starttls>`

Description

Set the connection security for LDAP remote servers.

For `ldap`: `none` (default), `ldaps` (LDAP over SSL/TLS, default port 636), `ldap-starttls` (LDAP with STARTTLS upgrade, default port 389).

> **WARNING**
>
> Using `none` for `ldap` will transmit all data — including user passwords — in plain text over the network. Use `ldaps` or `ldap-starttls` in production environments.

Options

Option	Description
`no`	Reset to the default encryption type (`none`).

`[no] description STRING`

Description

Free form description of this remote server.

Options

Option	Description
`no`	Remove any description.

continues on next page

continued

`[no] description STRING`

Option	Description
`STRING`	Free form text with a maximum length of 15 characters.

`[no] address <IP | FQDN>`

Description

Address of the remote server. Not available for `ldap` type; use `primary-address` instead.

Options

Option	Description
`no`	Remove and configured address.
`IP`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.
`FQDN`	A fully qualified domain name.

`[no] password PASSWORD`

Description

Password for remote server. For `ldap` type use `bind-password` instead.

> **NOTE**
>
> May not be empty.

Options

Option	Description
`no`	Remove the current configured password.
`PASSWORD`	Password with the following allowed characters: ASCII 33-126 (octal 041-176), max length 64 chars.

`[no] auth-port PORT`

Description

Authentication port to connect to on remote server.

Options

Option	Description
`no`	Use the default port for the selected type of server. For `radius` it is 1812, for `tacacs` it is 49, and for `ldap` it is 389 (or 636 when using `ldaps` encryption).
`PORT`	The port number to use.

`[no] timeout SECONDS`

Description

Abort trying to connect to the server after a timeout. Default is 5 seconds. Valid range is 3 to 60 seconds.

Options

Option	Description
`no`	Reset to the default timeout (5 seconds).
`SECONDS`	Timeout in seconds.

`[no] ca-cert STRING`

Description

Public CA certificate label used to validate the remote server. Available for `radius` with `peap-mschapv2` or `eap-tls` encryption, and for `ldap` with `ldaps` or `ldap-starttls` encryption.

:::warning Important When using a certificate, the certificate SAN must contain the primary/secondary address that is configured. :::

Options

Option	Description
`no`	Remove the configured CA certificate.
`STRING`	Label of a CA certificate imported into the PKI store.

`[no] client-cert STRING`

Description

Client certificate label for TLS authentication. Available for `radius` with `eap-tls` encryption, and for `ldap` with `ldaps` or `ldap-starttls` encryption.

Options

Option	Description
`no`	Remove the configured client certificate.
`STRING`	Label of a client certificate (P12/PFX package) imported into the PKI store.

LDAP-specific Settings

The following settings are only available when the server type is `ldap`.

`[no] ciphers-tls1.2 <CIPHER[,CIPHER ...]>`

Description

List of TLS 1.2 cipher suites (at selected security level) to include in cipher negotiation between clients and servers.

continues on next page

continued

`[no] ciphers-tls1.2 <CIPHER[,CIPHER ...]>`

Options

Option	Description
`no`	Disables TLS1.2
`no <CIPHER[,CIPHER ...]>`	Removes the specified cipher suite from list.
`CIPHER[,CIPHER ...]`	Adds cipher to list to be included in negotiation.
`no argument`	Enables TLS1.2 with DEFAULT cipher suites.

`[no] ciphers-tls1.3 <CIPHER[,CIPHER ...]>`

Description

List of TLS 1.3 cipher suites (at selected security level) to include in cipher negotiation between clients and servers.

Options

Option	Description
`no`	Disables TLS1.3
`no <CIPHER[,CIPHER ...]>`	Removes the specified cipher suite from list.
`CIPHER[,CIPHER ...]`	Adds cipher to list to be included in negotiation.
`no argument`	Enables TLS1.3 with DEFAULT cipher suites.

`[no] elliptic-curve-only <on | off>`

Description

Enable or disable the limitation to only use elliptic curve groups. When enabled, the system will only use elliptic curve groups when negotiating key exchange for supported cipher suites as descriped in RFC 8422 for TLS 1.2 and RFC 8446 for TLS 1.3. This improves the security. Default is off.

Options

Option	Description
`no`	Sets value to default value (off)
`on`	Only use following elliptic curve ciphers when negotiating key. secp256r1:secp384r1:secp521r1:x25519:x448:brainpoolP256r1:brainpoolP384r1:brainpoolP512r1
`off`	Allow use off all curves.

`[no] revocation`

Description

Enable TLS certificate revocation checking for LDAP connections. When enabled, the certificate revocation status is checked during TLS handshake with the LDAP server.

continues on next page

continued

`[no] revocation`

Options

Option	Description
`no`	Disables revocation checking.

`[no] primary-address <IP | FQDN>`

Description

Primary address of the LDAP server.

Options

Option	Description
`no`	Remove the configured address.
`IP`	IP address in standard quad-dotted notation, e.g. 192.168.1.1
`FQDN`	A fully qualified domain name.

`[no] secondary-address <IP | FQDN>`

Description

Secondary address of the LDAP server, used only if the primary address is unreachable. If no server responds, the device will iterate between the primary and secondary address for each authentication attempt until the configured timeout is reached.

Options

Option	Description
`no`	Remove the configured secondary address.
`IP`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.
`FQDN`	A fully qualified domain name.

`[no] credential-cache <on | off>`

Description

Enable or disable caching of credentials for this server. When enabled, the system will cache credentials for users authenticated by this server. This can improve availability but may have security implications. With cache on it is possible to log in even if the server is temporarily unreachable if a successful authentication has been done previously. Cached credentials are cleared at reboot and when the configuration is changed. Default is off.

Options

Option	Description
`no`	Reset to the default value (`off`).

`[no] user-id-mapping UID`

Description

Attribute name used to identify users in the LDAP directory. Default is `sAMAccountName`.

Options

Option	Description
`no`	Reset to the default value (`sAMAccountName`).
`UID`	The LDAP attribute name that contains the username, for example: • `uid` for OpenLDAP • `sAMAccountName` for Active Directory®

`[no] base-dn BASE_DN`

Description

Base Distinguished Name (DN) of the directory subtree to search for user entries.

Options

Option	Description
`no`	Clear the configured base DN.
`BASE_DN`	A DN string, for example: • `ou=users,dc=example,dc=com` • `dc=corp,dc=com`

`[no] bind-dn BIND_DN`

Description

Bind DN of the service account used for LDAP user lookups.

Options

Option	Description
`no`	Clear the configured bind DN.
`BIND_DN`	A DN string, for example: • `cn=ldap-reader,ou=service-accounts,dc=corp,dc=com`

`[no] role-base-dn ROLE_BASE_DN`

Description

Role base DN of the service account used for LDAP user lookups. If this is filled in the role in the role-map is checked against this DN.

continues on next page

continued

`[no] role-base-dn ROLE_BASE_DN`

Options

Option	Description
`no`	Clear the configured role base DN.
`ROLE_BASE_DN`	A DN string without CN, for example: • `ou=groups,dc=corp,dc=com`

`[no] bind-password PASSWORD`

Description

Password for the bind DN service account.

Options

Option	Description
`no`	Remove the configured bind password.
`PASSWORD`	Password with the following allowed characters: ASCII 33-126 (octal 041-176), max length 64 chars.

`[no] search-filter FILTER`

Description

LDAP search filter applied when looking up user entries. Parentheses are added automatically if omitted.

Options

Option	Description
`no`	Clear the configured search filter.
`FILTER`	An LDAP filter expression, for example: • `(objectClass=inetOrgPerson)`

`[no] role-map LOCAL_ROLE REMOTE_LDAP_ROLE`

Description

Map a local system role to a remote LDAP group or role attribute value. Multiple role mappings can be configured.

:::warning Important It is not recommended using the same username on the local user account as used with LDAP since it might result in unknown authentication behavior. :::

continues on next page

continued

`[no] role-map LOCAL_ROLE REMOTE_LDAP_ROLE`

Options

Option	Description
`no`	Remove the mapping for the specified `LOCAL_ROLE`, or remove all mappings if no role is provided.
`LOCAL_ROLE`	A valid local role: `administrator`, `engineer`, `operator`, `auditor`, `viewer`.
`REMOTE_LDAP_ROLE`	The corresponding LDAP group DN or role value, for example: • `CN=NetworkAdmins,OU=Groups,DC=corp,DC=com`

Server Group Settings

Configure a server group and enter its configuration context. When creating a server group the `ID` must be a number from 0 to 1.

> **NOTE**
>
> To use a server group for authentication it must be added to the authentication chain.

```
example:/config/aaa/#> server-group 0
Creating new server group 0
example:/config/aaa/server-group-0/#>
```

Syntax

`[no] type <radius | tacacs>`

Description

Set the remote server type to be either `radius` or `tacacs`.

Options

Option	Description
`no`	Reset the type to its default setting, `radius`.

`[no] description STRING`

Description

Free form description of this server group.

Options

Option	Description
`no`	Remove any description.
`STRING`	Free form text with a maximum length of 15 characters.

`[no] server <ID|ID,ID,...>`

Description

Remote server(s) to be part of the server group.

> **NOTE**
>
> When using a server group for authentication each server will be tested in order. If a response is received from any of the servers the remaining ones in the group will not be tested.

Options

Option	Description
`no`	Remove any configured server IDs.
`ID`	Id of a `remote-server` to be part of the server group.

Authentication Chain Settings

The Authentication chains provides a way to specify how the device priorities different authentication methods. It also allows to specify how the device should handle a reject, and the conditions for local fallback at the end of the chain. Each link in the chain may either be a remote server, or a server group..

```
example:/config/aaa/#> auth-chain
example:/config/aaa/auth-chain/#>
```

Syntax

Syntax: [no] order <server-id|group-id> [..] Usage: Authentication chain order

`[no] order [server-<SID>|group-<GID>] [..]`

Description

Set zero or more authentication methods to be used for login requests on the device. NOTE: Order matters

Options

Option	Description
`no`	Remove any configured authentication methods.
`server-<SID>`	Use an existing `remote-server` with the provided `SID` in the chain.
`group-<GID>`	Use an existing `server-group` with the provided `GID` in the chain.

`[no] on-reject next|deny`

Description

Dictates how the authentication chain behaves when an authentication method rejects the user credentials. NOTE: Admin accounts are always available over the serial console, regardless of `on-reject` and `local-fallback` settings.

Options

Option	Description
`no`	Reset `on-reject` to the default value of `next`
`next`	When an authentication method rejects the users credentials: Try the next authentication method in order, or local-fallback if there are no more methods.
`deny`	When an authentication method rejects the users credentials: Deny the login attempt

`[no] local-fallback never|serial|always`

Description

Acts as a last resort after all other methods fail to give a authorative answer. NOTE: Admin accounts are always available over the serial console, regardless of `on-reject` and `local-fallback` settings.

Options

Option	Description
`no`	Reset `local-fallback` to the default value of `always`
`never`	Only admins accounts over the serial console work.
`serial`	Local accounts are used as a fallback when logging in over the serial console
`always`	Local accounts are used as a fallback

Password Compliance Policy Settings

Password compliance policy provides a way to check a password at creation time against a set of policies. If enabling the password compliance policy the password will be checked for compliance when changing an existing users password or adding a new user in the cli or webgui.

Configure the password policy and enter its configuration context:

```
example:/config/aaa/#> password-policy
Activating password policy with default settings, type 'abort' to cancel.
example:/config/aaa/password-policy/#>
```

Syntax

`[no] enable`

Description

This setting controls if the password compliance policy is enabled or disabled. Default is Disabled.

Options

Option	Description
`no`	Disable password compliance policy

`min-length`

Description

Set password policy min length <8-64>. Default is 14.

Options

Option	Description
`value`	Minimum length of password

`uppercases`

Description

Specify number of uppercase letters <0-64>. Default is 0.

Options

Option	Description
`value`	Number of uppercase letters

`lowercases`

Description

Specify number of lowercase letters <0-64>. Default is 0.

Options

Option	Description
`value`	Number of lowercase letters

`digits`

Description

Specify number of digits <0-64>. Default is 0.

continues on next page

continued

`digits`

Options

Option	Description
`value`	Number of digits

`specials`

Description

Specify number of special characters <0-64>. Default is 0.

Options

Option	Description
`value`	Number of special characters

`reject-username`

Description

Enable/Disable reject username in password. Default is Enabled. If enabled the username, in any form, must not be part of the password.

Options

Option	Description
`0`	Disabled
`1`	Enabled

MAC auth group Settings

Configure a MAC auth group and enter its configuration context.

```
example:/config/#> aaa
example:/config/aaa/#> mac-auth 0
Creating new mac-auth group 0.
Remember, you need to reference it from a Port Access instance as well!
example:/config/aaa/mac-auth-0/#> mac match aa:bb:cc:dd:ee:ff description "My MAC match"
example:/config/aaa/mac-auth-0/#> leave
example:/config/#> leave
```

Syntax

`[no] description STRING`

Description

Free form description of this MAC auth group.

continues on next page

continued

`[no] description STRING`

Options

Option	Description
`no`	Remove any description.
`STRING`	Free form text with a maximum length of 15 characters.

`[no] mac match <MAC-PATTERN> [description <STRING>]`

Description

Create or delete a Ethernet MAC address pattern.

A single MAC address can be specified in the format: hh:hh:hh:hh:hh:hh

Options

Option	Description
`no`	Remove MAC-PATTERN

Status

Show any user currently connected to the device, and the authentication method used when connecting to the device.

```
example:/#> show users
TYPE    USERNAME            REMOTE HOST          LOGIN
Console admin                                    Wed Aug 14 08:47:02 2019
```

Show AAA status, same information as when in config mode.

```
example:/#> show aaa
```

Show local-db status, same information as when in config mode.

```
example:/#> show aaa local-db 1
```

Show local-db status for all databases.

```
example:/#> show aaa local-db
```

2.2.2. AAA - Restricted interface

About

If the user does not have administrator privileges the number of available commands are restricted.

Viewer

Below is presented a list of commands available to a built-in user having a *viewer* role.

```
hostname:/$> whoami
gertrude (viewer)
hostname:/$> help
show aaa              Show Status of AAA
show aggregates       Status of aggregated ports
show alarm [log]      Alarm status, or show alarm log
show arp              Show CPU ARP cache
batch                 Non-interactive mode
clear                 Clear screen.

pki                   Manage PKI settings, and enrollment of certificates.
show custom-trdp      Show custom telegrams.
show dhcp-clients     Show active DHCP clients. ...
dhcp-server [flush]   Manage DHCP Server.
show sensor           Display environment status: temperature, power, digital in and SFP ...
show fdb              Forwarding database (ATU/MAC)
show partitions       Show partition table
show flash-table      Show active flash partition table. ...
frnt <..>             Show/manage status/stats of FRNT rings
show ring [..]        Status of ring protocols
show mrp              Status of MRP rings
show coupling         Status of Coupling instance(s).
show chaining         Status of Chaining instance.
show history          Show command history
show hsr-prp          Status of HSR/PRP Redundancy instance(s).
ip <igmp|ospf|..>     System IP mode commands
ipv6 <mld|..>         System IPv6 mode commands
show iface [..]       Interface overview
show ifaces           Interfaces overview
interactive           Interactive mode
show lldp             LLDP/CDP neighbours
lockout-policy <..>   Account lockout policy ...
show ttdp             TTDP neighbours
show monitor [ID]     Port Monitor Overview
port [..]             Manage or show port status. ...
ptp                   System PTP commands.
show policy           Policy Filtering
rmon                  Enter RMON mode
show spanning-tree    Status of Spanning Tree (RSTP). ...
show vlan [vid]       IEEE 802.1Q VLAN
show vrrp             Status of VRRP daemon.
show vrrp-legacy      Status of VRRP daemon.

show ipconfig <..>    Show legacy neighbour devices. ...
show <..>             Show summary/status of COMMAND, or display contents of FILE. ...
ntp <restart>         Show/Manage NTP clients ...
ssh-host-key [..]     Ssh host key management ...


logout                Logout, or quit the CLI.
show memory           Display memory usage
show processes        Display running processes
system-information    Show system information
tutorial              Brief introduction to the CLI
uptime                Display system uptime
version               Show currently running firmware build details
```

```
watchdog              Show watchdog status, raw
users                 Show which users are online right now and from where.
whoami                Show user ID.
ledflash [s]          Flash all yellow port leds
show port-access      Show status of port access control (IEEE 802.1X and MAC ...
metrics <..>          Metrics status commands

id                    List Unix user id and group id(s).
status [..]           Show the list of warnings and errors
_____________________________________________________________________________
See "help all" or "help COMMAND" for more online help.
hostname:/$>
```

Guest

The *guest* role is deprecated and replaced by the *viewer* role (see above).

Operator

This is a list of commands available to a built-in user having an *operator* role:

```
hostname:/$> whoami
oswald (operator)
hostname:/$> help
show aaa                    Show Status of AAA
show aggregates             Status of aggregated ports
show alarm [log]            Alarm status, or show alarm log
show arp                    Show CPU ARP cache
audit                       System audit commands
batch                       Non-interactive mode
clear                       Clear screen.

pki                         Manage PKI settings, and enrollment of certificates.
show custom-trdp            Show custom telegrams.
show dhcp-clients           Show active DHCP clients. ...
dhcp-server [flush]         Manage DHCP Server.
show sensor                 Display environment status: temperature, power, digital in and SFP ...
show fdb                    Forwarding database (ATU/MAC)
show partitions             Show partition table
show flash-table            Show active flash partition table. ...
frnt <..>                   Show/manage status/stats of FRNT rings
show ring [..]              Status of ring protocols
show mrp                    Status of MRP rings
show coupling               Status of Coupling instance(s).
show chaining               Status of Chaining instance.
show history                Show command history
show hsr-prp                Status of HSR/PRP Redundancy instance(s).
ip <igmp|ospf|..>           System IP mode commands
ipv6 <mld|..>               System IPv6 mode commands
show iface [..]             Interface overview
show ifaces                 Interfaces overview
interactive                 Interactive mode
show lldp                   LLDP/CDP neighbours
lockout-policy <..>         Account lockout policy ...
show ttdp                   TTDP neighbours
show monitor [ID]           Port Monitor Overview
port [..]                   Manage or show port status. ...
ptp                         System PTP commands.
package                     Manage packages ...
show policy                 Policy Filtering
rmon                        Enter RMON mode
show spanning-tree          Status of Spanning Tree (RSTP). ...
tunnel                      Tunnel handling
show vlan [vid]             IEEE 802.1Q VLAN
show vrrp                   Status of VRRP daemon.
show vrrp-legacy            Status of VRRP daemon.

show ipconfig <..>          Show legacy neighbour devices. ...
ipcalc [..]                 An IP subnet calculator ...
ping <..>                   Ping a network host or group. ...
```

```
show <subcommand> [ARGS]  Show summary/status of COMMAND, or display contents of FILE. ...
show factory-config [..]  Factory configuration, default settings ...
show logfile [..]         Show contents of a log file ...
show running-config [..]  Currently running configuration, RAM only ...
show startup-config [..]  Startup configuration, from built-in flash, USB, or SD card ...
show safe-config [..]     Safe configuration, from built-in flash, USB, or SD card ...
show no-config [..]       No configuration, from built-in flash, USB, or SD card ...
ntp <restart>             Show/Manage NTP clients ...
ssh [..]                  Secure shell login to another host ...
ssh-host-key [..]         Ssh host key management ...
telnet <..>               Telnet login to another host
tcpdump <..>              Capture traffic on a network interface. ...
traceroute <..>           Trace the route packets take to a network host. ...
nslookup <HOST> [SERVER]  Query the nameserver for the IP address of the given HOST optionally using  ...


media [..]                Media related operations. ...
date [..]                 System date & time
diff <FILE1> <FILE2>      Compare two configuration files, unidiff. ...
follow <FILE>             Continuously monitor a file, like UNIX 'tail -F'
logout                    Logout, or quit the CLI.
show memory               Display memory usage
show processes            Display running processes
reboot [cold]             Reboot device with startup-config
system-information        Show system information
tutorial                  Brief introduction to the CLI
uptime                    Display system uptime
upgrade <..>              Upgrade all, primary (main), secondary (backup) or bootloader firmware. ...
firmware-checksum [..]    Check the CRC of a firmware image on the device. ...
version                   Show currently running firmware build details
watchdog                  Show watchdog status, raw
users                     Show which users are online right now and from where.
whoami                    Show user ID.
boot                      Device specific bootstrap settings.
ledflash [s]              Flash all yellow port leds
show port-access          Show status of port access control (IEEE 802.1X and MAC ...
config-hash               Shows the SHA256 hash of the running configuration ...
logging <..>              Logging status commands
metrics <..>              Metrics status commands

provisioning              Port provisioning. ...
file                      File operations
id                        List Unix user id and group id(s).
status [..]               Show the list of warnings and errors
_____________________________________________________________________________
See "help all" or "help COMMAND" for more online help.
hostname:/$>
```

Auditor

This is a list of commands available to a built-in user having an *auditor* role:

```
hostname:/$> whoami
fredrik (auditor)
hostname:/$> help
show aaa              Show Status of AAA
show aggregates       Status of aggregated ports
show alarm [log]      Alarm status, or show alarm log
show arp              Show CPU ARP cache
audit                 System audit commands
batch                 Non-interactive mode
clear                 Clear screen.

pki                   Manage PKI settings, and enrollment of certificates.
show custom-trdp      Show custom telegrams.
show dhcp-clients     Show active DHCP clients. ...
dhcp-server [flush]   Manage DHCP Server.
show sensor           Display environment status: temperature, power, digital in and SFP ...
show fdb              Forwarding database (ATU/MAC)
show partitions       Show partition table
```

```
show flash-table     Show active flash partition table. ...
frnt <..>            Show/manage status/stats of FRNT rings
show ring [..]       Status of ring protocols
show mrp             Status of MRP rings
show coupling        Status of Coupling instance(s).
show chaining        Status of Chaining instance.
show history         Show command history
show hsr-prp         Status of HSR/PRP Redundancy instance(s).
ip <igmp|ospf|..>    System IP mode commands
ipv6 <mld|..>        System IPv6 mode commands
show iface [..]      Interface overview
show ifaces          Interfaces overview
interactive          Interactive mode
show lldp            LLDP/CDP neighbours
lockout-policy <..>  Account lockout policy ...
show ttdp            TTDP neighbours
show monitor [ID]    Port Monitor Overview
port [..]            Manage or show port status. ...
ptp                  System PTP commands.
show policy          Policy Filtering
rmon                 Enter RMON mode
show spanning-tree   Status of Spanning Tree (RSTP). ...
show vlan [vid]      IEEE 802.1Q VLAN
show vrrp            Status of VRRP daemon.
show vrrp-legacy     Status of VRRP daemon.

show ipconfig <..>   Show legacy neighbour devices. ...
show <..>            Show summary/status of COMMAND, or display contents of FILE. ...
ntp <restart>        Show/Manage NTP clients ...
ssh-host-key [..]    Ssh host key management ...


logout               Logout, or quit the CLI.
show memory          Display memory usage
show processes       Display running processes
system-information   Show system information
tutorial             Brief introduction to the CLI
uptime               Display system uptime
version              Show currently running firmware build details
watchdog             Show watchdog status, raw
users                Show which users are online right now and from where.
whoami               Show user ID.
ledflash [s]         Flash all yellow port leds
show port-access     Show status of port access control (IEEE 802.1X and MAC ...
metrics <..>         Metrics status commands

id                   List Unix user id and group id(s).
status [..]          Show the list of warnings and errors
_____________________________________________________________________________
See "help all" or "help COMMAND" for more online help.
hostname:/$>
```

The *auditor* role has read-only access to the audit logs:

```
hostname:/$> whoami
fredrik (auditor)
hostname:/$> audit
hostname:/audit/$> help
list [..]          List the audit Categories and EventIDs that exist in the ...
raw-category [..]  Show the audit log file by category.
raw                Show the audit log file.
show               Show the audit ring buffer.
status             Show the audit ring buffer status.
last NUMBER        Show the audit ring buffer last [N] entries.
range [START] NUM  Show the audit ring buffer in range [[START] NUM] entries.
```

```
time [..]          Show the audit ring buffer in time range [[YYYY-MM[-DD]] [hh:mm[:ss]]] [[YY ...
from [..]          Show the audit ring buffer from time [[YYYY-MM[-DD]] ...
to [..]            Show the audit ring buffer to time [[YYYY-MM[-DD]] ...
search [..]        Show the audit ring buffer entries containing [STRING]. ...
category CATEGORY  Show the audit ring buffer entries from category [CATEGORY].
_______________________________________________________________________________
See "help all" or "help COMMAND" for more online help.
hostname:/audit/$>
```

Engineer

Similarly this is a list of commands available to a built-in user having an *engineer* role.

```
hostname:/$> whoami
jonas (engineer)
hostname:/$> help
show aaa                    Show Status of AAA
show aggregates             Status of aggregated ports
show alarm [log]            Alarm status, or show alarm log
show arp                    Show CPU ARP cache
audit                       System audit commands
batch                       Non-interactive mode
clear                       Clear screen.

pki                         Manage PKI settings, and enrollment of certificates.
show custom-trdp            Show custom telegrams.
dhcp <..>                   Manage DHCP client. ...
show dhcp-clients           Show active DHCP clients. ...
dhcp-server [flush]         Manage DHCP Server.
show sensor                 Display environment status: temperature, power, digital in and SFP ...
show fdb                    Forwarding database (ATU/MAC)
show partitions             Show partition table
show flash-table            Show active flash partition table. ...
frnt <..>                   Show/manage status/stats of FRNT rings
show ring [..]              Status of ring protocols
show mrp                    Status of MRP rings
show coupling               Status of Coupling instance(s).
show chaining               Status of Chaining instance.
show history                Show command history
show hsr-prp                Status of HSR/PRP Redundancy instance(s).
ip <igmp|ospf|..>           System IP mode commands
ipv6 <mld|..>               System IPv6 mode commands
show iface [..]             Interface overview
show ifaces                 Interfaces overview
interactive                 Interactive mode
show lldp                   LLDP/CDP neighbours
lockout-policy <..>         Account lockout policy ...
show ttdp                   TTDP neighbours
show monitor [ID]           Port Monitor Overview
port [..]                   Manage or show port status. ...
ptp                         System PTP commands.
package                     Manage packages ...
show policy                 Policy Filtering
rmon                        Enter RMON mode
show spanning-tree          Status of Spanning Tree (RSTP). ...
tunnel                      Tunnel handling
show vlan [vid]             IEEE 802.1Q VLAN
show vrrp                   Status of VRRP daemon.
show vrrp-legacy            Status of VRRP daemon.

show ipconfig <..>          Show legacy neighbour devices. ...
ipcalc [..]                 An IP subnet calculator ...
ping <..>                   Ping a network host or group. ...
show <subcommand> [ARGS]    Show summary/status of COMMAND, or display contents of FILE. ...
show factory-config [..]    Factory configuration, default settings ...
show logfile [..]           Show contents of a log file ...
show running-config [..]    Currently running configuration, RAM only ...
show startup-config [..]    Startup configuration, from built-in flash, USB, or SD card ...
show safe-config [..]       Safe configuration, from built-in flash, USB, or SD card ...
show no-config [..]         No configuration, from built-in flash, USB, or SD card ...
ntp <restart>               Show/Manage NTP clients ...
ssh [..]                    Secure shell login to another host ...
```

```
ssh-host-key [..]          Ssh host key management  ...
telnet <..>                Telnet login to another host
tcpdump <..>               Capture traffic on a network interface.  ...
traceroute <..>            Trace the route packets take to a network host.  ...
nslookup <HOST> [SERVER]   Query the nameserver for the IP address of the given HOST optionally using   ...

configure [terminal]       Enter system configuration mode (running config).  ...

copy [force] <from> <to>   Copy configuration files, save config.  ...
media [..]                 Media related operations.  ...
date [..]                  System date & time
diff <FILE1> <FILE2>       Compare two configuration files, unidiff.  ...
follow <FILE>              Continuously monitor a file, like UNIX 'tail -F'
logout                     Logout, or quit the CLI.
show memory                Display memory usage
show processes             Display running processes
reboot [cold]              Reboot device with startup-config
system-information         Show system information
tutorial                   Brief introduction to the CLI
uptime                     Display system uptime
upgrade <..>               Upgrade all, primary (main), secondary (backup) or bootloader firmware.  ...
firmware-checksum [..]     Check the CRC of a firmware image on the device.  ...
version                    Show currently running firmware build details
watchdog                   Show watchdog status, raw
users                      Show which users are online right now and from where.
whoami                     Show user ID.
boot                       Device specific bootstrap settings.
ledflash [s]               Flash all yellow port leds
show port-access           Show status of port access control (IEEE 802.1X and MAC  ...
config-hash                Shows the SHA256 hash of the running configuration  ...
logging <..>               Logging status commands
metrics <..>               Metrics status commands

provisioning               Port provisioning.  ...
encrypt-password <..>      Sets a password to be used in encrypted-secrets.
file                       File operations
id                         List Unix user id and group id(s).
status [..]                Show the list of warnings and errors
_____________________________________________________________________________
See "help all" or "help COMMAND" for more online help.
hostname:/$>
```

The *engineer* role can configure most of the features, except some security related ones like AAA, PKI, audit log:

```
hostname:/$> whoami
jonas (engineer)
hostname:/$> config
hostname:/config/$> help
alarm              Alarm & Event settings

lag <ID>           Manage link aggregates.
port <..>          Physical port settings.  ...
vlan <VID>         IEEE 802.1Q VLAN settings
vlans              Common settings for IEEE 802.1Q VLANs
iface <IFNAME>     Interface IP settings.  ...
ip                 System IP level settings
ipv6               System IPv6 specific settings
fdb                Forwarding database (MAC) settings
ring [..]          Manage Ring Protocols.
router <..>        Dynamic routing protocols
spanning-tree      Manage Spanning Tree (RSTP) settings
ntp                Network time settings
ptp                Precision Time Protocol settings
system             Common system settings
logging            Local and remote logging
```

```
monitor <ID>       Port Monitoring/Mirroring
dhcp-server <..>   DHCP server settings
dhcp-relay         DHCP relay agent settings
dns                DNS server/client settings
tunnel <..>        VPN tunnel settings
discover           Device discovery services, mDNS, SSDP
management         Device management services, HTTP, HTTPS, SSH, Telnet, SNMP
lldp               LLDP daemon settings
ttdp               TTDP settings
metrics            metrics settings
hostname <..>      System hostname string (RFC1123)
location <..>      System location string
contact <email>    System contact string
watchdog           Watchdog daemon settings. ...
port-access        Port access control settings (IEEE 802.1X and MAC Auth)
policy             Traffic filtering policy settings. Traffic filtering policies ...
hsr-prp <ID>       Manage HSR/PRP settings
_____________________________________________________________________________
See "help all" or "help COMMAND" for more online help.
hostname:/config/$>
```

2.3. Ethernet/LAN Ports

2.3.1. Ethernet Ports

Introduction

This page describes settings related to Ethernet ports. Ethernet ports enable connectivity between units using LAN technology. By default all Ethernet ports on a WeOS unit are bridged and associated with VLAN 1. Thus, by default a WeOS unit acts as a manageable switch. It is possible to move ports to different VLANs. It is also possible let an Ethernet port become a standalone network interface, see VLAN and Interface configuration pages.

For details on port settings related to priority handling, see the Layer-2 QoS Configuration page.

Port Naming and Listing

Ethernet ports are named *eth1*, *eth2*, etc. The exception is products with *M12 connectors*, e.g., the Westermo Viper products, where Ethernet ports are named *ethX1*, *ethX2*, etc.

```
example:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Up     1000-T  1000M-Full  Forwarding      vlan U:1
eth2             Up     1000-T   100M-Full  Forwarding      vlan U:1
eth3             Down   1000-T  ----------  No-Link         vlan U:1
eth4             Down   1000-T  ----------  No-Link         vlan U:1
eth5             Down   1000-T  ----------  No-Link         vlan U:1
eth6             Down   1000-T  ----------  No-Link         vlan U:1
eth7             Down   1000-T  ----------  No-Link         vlan U:1
eth8             Down   1000-T  ----------  No-Link         vlan U:1
eth9             Down   No-SFP  ----------  No-Link         vlan U:1
eth10            Up     100-FX   100M-Full  Forwarding      vlan U:1
eth11            Up    1000-LX  1000M-Full  Forwarding      vlan U:1
eth12            Down   No-SFP  ----------  No-Link         vlan U:1
Speed formatted as, e.g. 100-Full denotes static speed-duplex links.
example:/#>
```

To specify a range of ports in CLI, use *double dots* (**eth1..eth3**), and comma to form a list (**eth1,eth5,eth9**).

```
example:/config/#> vlan 2
example:/config/vlan-2/#> untagged eth1..eth3,eth5
Moving untagged port eth1 from vid 1 to vid 2.
Moving untagged port eth2 from vid 1 to vid 2.
Moving untagged port eth3 from vid 1 to vid 2.
Moving untagged port eth5 from vid 1 to vid 2.
example:/config/vlan-2/#>
```

Types of Ethernet ports

Ethernet ports can be (fixed) copper ports or SFP ports. SFP ports can be used with Westermo SFP ports, fibre or copper SFPs.

Port speeds vary for the different port types and SFP types. For details and canonical reference, see the data sheet of your WeOS product.

- Fixed Copper Ports: These ports are commonly referred to as Fast Ethernet (FE) or Gbit Ethernet copper ports.
  - 10/100: Ports supporting 10BASE-T and 100BASE-TX.
  - 10/100/1000: Ports supporting 10BASE-T, 100BASE-TX and 1000BASE-T.

- SFP Ports:
  - 100/1000: Ports supporting Westermo 'Gbit Fibre SFPs', 'Fast Ethernet Fibre SFPs' and 'Gbit Copper SFPs'.
  - 100/1000 (no PHY)[1]: Ports supporting Westermo 'Gbit Fibre SFPs', 'Fast Ethernet Fibre SFPs'[2] and 'Gbit Copper SFPs'.
  - 1000/10000: Ports supporting Westermo '10 Gbit Fibre SFPs', 'Gbit Fibre SFPs' and 'Gbit Copper SFPs'.

SFPs are not hot pluggable, i.e., inserting a new type of SFP will need a reboot of the device.

Configuration

Configuration of network ports is done in the port context. An example of how to configure port eth1:

```
example:/#> configure
example:/config/#> port eth1
example:/config/port-eth1/#>
```

Configuring port speed/duplex settings

For examples, see the Port Speed/Duplex Howto.

There are two settings related to configuration of port speed and duplex mode. Auto-negotiation is enabled by default, and speed-duplex which will advertise all supported modes for the port type. *The default settings are sufficient for almost all use cases.*

```
example:/config/port-eth1/#> show auto-negotiate
Enabled
example:/config/port-eth1/#> show speed-duplex
Auto
example:/config/port-eth1/#>
```

Tweaking speed-duplex settings be done in two ways: either to disable auto-negotiation and use a fixed speed-duplex mode, or to control the *advertisement mask*.

- Enable/disable auto-negotiation: Auto-negotiation is enabled by default, but can be enabled/disabled[3] per port via the `[no] auto-negotiate` setting.
- Advertisement mask: The second setting is used to control what speed/duplex mode(s) to offer during the auto-negotiation (fine-tuning the advertisement mask). The command is `[no] speed-duplex <LIST>`. The default is **no speed-duplex** (same as **speed-duplex auto**), which means that "all" applicable modes are advertised.

Applicable mode depends on the PHY/port type

On fixed FE copper ports, speed/duplex can be *10-half*, *10-full*, *100-half* and *100-full*. Thus, **no speed-duplex** means that these four modes will be advertised in the auto-negotiation. Any attempt to configure any other more mode (say **speed-duplex 1000-full**) will be rejected. Fixed Gbit copper ports support the same modes, but also *1000-half* and *1000-full*.

[1] Some products with 100/1000 SFP ports need a special type of Westermo 100 Mbit/s Fibre SFP to operate at 100 Mbit. See the data sheet of your Westermo product.

[2] Some products with 100/1000 SFP ports need a special type of Westermo 100 Mbit/s Fibre SFP to operate at 100 Mbit. See the data sheet of your Westermo product.

[3] More precisely, the system will use the configured `[no] auto-negotiate` setting if applicable. In particular, it is OK to enable auto-negotiation in an SFP port with a FE Fibre SFP - the system will ignore the setting as the FE Fibre standard prohibits auto-negotiation.

When configuring SFP ports, all modes of the internal SFP PHY apply. For example, if **speed-duplex 100-full, 1000-full** is configured, then only "1000-full" effectively applies if a Gbit Fibre SFP is inserted. If the system cannot apply the configured setting the port will not come up, e.g., if setting **speed-duplex 1000-full** and inserting a FE Fibre SFP. Again, using the default settings (auto-negotiation with all applicable speed-duplex modes enabled) will be sufficient in almost all use cases.

Configuring a fixed speed-duplex

To limit a port to use a specific speed-duplex mode you should configure the `speed-duplex` setting to that mode (e.g., *100-full*). Optionally, you could also disable auto-negotiation.

Disabling auto-negotiation is preferred when the peer does not use auto-negotiation. However, keeping *auto-negotiation* enabled is generally recommended, as it improves interoperability with other auto-negotiating devices (§37.1.4.4 of[1]).

```
example:/config/#> port eth1
example:/config/port-eth1/#> speed-duplex 100-full
example:/config/port-eth1/#> no auto-negotiate
example:/config/port-eth1/#>
```

> **NOTE**
>
> It is not strictly necessary to select a single speed-duplex mode. If auto-negotiation is disabled when the *speed-duplex* setting specifies multiple modes, the *best* applicable mode will be used as fixed mode. This is always the case for FE fibre SFPs, as auto-negotiation does not apply if enabled.

CLI Configuration Command Syntax

Configuration commands are available in the Port Configuration Context.

```
example:/#> config
example:/config/#> port eth1
example:/config/port-eth1/#>
```

`[no] enable`

Description

This setting controls if the port is enabled or disabled. If the port is disabled the network link will go down and no network activity is possible.

Options

Option	Description
`no`	Disable port

[1] IEEE 802.3-2012 Clause 37: "Auto-Negotiation function, type 1000BASE-X"

`[no] auto-negotiate`

Description

This setting controls the port's ability to transmit auto-negotiate frames to a link partner. It is used in combination with speed-duplex when a single speed-duplex setting is given, this setting controls if static mode or auto-neg for a single setting is to be transmitted.

Options

Option	Description
`no`	Disable auto-negotiation No auto-negotiation will be performed
`auto-negotiate`	Use auto-negotiation to resolve speed and duplex

`[no] speed-duplex <auto | SPEED_DUPLEX[,SPEED_DUPLEX]>`

Description

This setting controls the speed and duplex of the port or the advertisement mask if auto-negotiation is used.

Options

Option	Description
`auto`	Default use all possible speed duplex combinations that is possible for this port.
`SPEED_DUPLEX`	Speed and duplex combination such as *100-half*.

`[no] priority <0..7>`

Description

This setting controls the port priority. See Layer-2 QoS for details. Default: 0

Options

Option	Description
`value`	Priority value (0..7) according to 802.1p where 7 means the highest priority.
`no`	Reset port priority to default (0).

`[no] priority-mode <tag|ip|port|prefer[ip|tag]>`

Description

This setting configures the port's priority mode. See Layer-2 QoS for details. Default: prefer tag

Options

Option	Description
`no`	Reset to this port's priority mode to default (tag).

continues on next page

continued

`[no] priority-mode <tag|ip|port|prefer[ip|tag]>`

Option	Description
`tag`	Use the VLAN tag (PCP) priority value of incoming frame.
`ip`	Use the IP QoS/DSCP priority field of incoming packet.
`prefer`	For 'prefer tag', use VLAN tag (PCP) priority for an incoming VLAN tagged IP packet. For untagged IP packet, use DSCP priority. For VLAN tagged non-IP packet, use VLAN tag priority. For 'prefer ip', use IP (DSCP) priority for an incoming VLAN tagged IP packet or an untagged IP packet. For VLAN tagged non-IP packets, use VLAN tag priority. Use priority mode 'port' if none of the above matches.
`port`	Use this port's priority value, see the `[no] priority <0..7>` setting above.

`[no] mdix-mode <auto|mdi|mdix>`

Description

Specify the ports MDI/MDIX crossover settings. By default, the port is able to sense the which pin to use to get link, auto MDI/MDIX (Media Dependent Interface/crossover).

Options

Option	Description
`no`	Sets the default auto MDI/MDIX mode on the port.
`auto`	See `no` above
`mdi`	Fixed MDI/MDIX setting, use the tx pair for transmit.
`mdix`	Fixed MDI/MDIX setting, use the tx pair to receive.

`[no] default-vid VID`

Description

Specify the *fallback* default VID for the untagged frames on the port. This only applies if the port is *not* associated untagged with any VLAN.

Options

Option	Description
`no`	Sets the default-vid to 1 on the port.
`VID`	default-vid for the port <1-4094>

`[no] description <STRING>`

Description

Specify the ports description string.

continues on next page

continued

`[no] description <STRING>`

Options

Option	Description
`no`	Disable the ports description string.
`STRING`	Free form description of this port, with a maximum length of 32 characters, The string can be up to 32 characters long and any combination of printable ASCII characters in range [32-126]

`[no] learning`

Description

Set learning (switch/hub) mode, default enabled.

Options

Option	Description
`no`	Set the port to no learning, i.e. hub mode.

`[no] gbit-cu-master [auto]`

Description

This setting enables manual configuration of the master/follower for the clock sync on Gbit Ethernet. Default is *auto*.

Options

Option	Description
`gbit-cu-master auto`	This sets the master/follower for the clock sync on Gbit Ethernet to automatic mode. This is the default.
`gbit-cu-master`	This sets the master/follower for the clock sync on Gbit Ethernet to manual *master* mode.
`no gbit-cu-master`	This sets the master/follower for the clock sync on Gbit Ethernet to manual *follower* mode.

`[no] fastlink [1000-t-up]`

Description

This settings enables the fastlink functionality if it's supported on the port. Fastlink minimises the time for the port to detect link loss on Gbit ports.

Options

Option	Description
`no`	Disable fastlink.

continues on next page

continued

`[no] fastlink [1000-t-up]`

Option	Description
`1000-t-up`	Enables fastlink up for Gbit copper Ethernet. This may only be set in combination with speed-duplex 1000-full, master/follower manual config and manual mdi/mdix config. **Note:** This option adds 10ms to fastlink down and may thereby not be used in combination with FRNT.

`rate-limit`

Description

Enter context to configure rate-limit for ingress and egress. For supported products it is possible to configure storm-control from this configuration context.

`health-monitor`

Description

Enter context to configure link health-monitor. Through health-monitor context, it is possible to configure monitoring of link state-changes.

`show`

Description

Show the configuration summary for the port

`show detailed [PORT[,PORT...]]`

Description

Show detailed port information.

If PORT(s) is specified, detailed information for the specified port(s) is shown. If no PORT is specified, information for all ports is shown.

> **AVAILABILITY**
>
> This command will only be present if the port context is accessed with more than one port in the current context's port range. If the context is accessed with a single port the regular `show` command will show detailed information for that port.

> **SELECTABLE PORTS**
>
> Can only show detailed information for ports that are part of the current context's port range.

continues on next page

continued

`show detailed [PORT[,PORT...]]`

Example

```
example:/config/port-eth1..eth4/#> show detailed
[ ... ]
example:/config/port-eth1..eth4/#> show detailed eth1
[ ... ]
example:/config/port-eth1..eth4/#> show detailed eth1,eth2
[ ... ]
```

Options

Option	Description
`PORT`	The name of a port, such as `eth1`.

Port Rate Limit

Configuration commands for rate-limit can be found in port rate-limit context found under the port configuration context.

> **NOTE**
>
> Enabling rate-limit in your network may affect the performance of protocols and redundancy services dependent on the flow of management traffic, such as FRNT, MRP, RSTP, and other services. If management traffic connected to these services start getting rate-limited their performance can not be guaranteed.

In the rate-limit configuration context it is possible to configure storm control. For examples of how storm control can be configured and used, refer to the Storm Control Howto.

```
example:/#> config
example:/config/#> port eth1
example:/config/port-eth1/#> rate-limit
example:/config/port-eth1/rate-limit/#>
```

`[no] ingress <bps|fps> <8..1000000>[k|M|G] [match <TYPE> [<TYPES>]]`

Description

Specify the ports ingress traffic rate limit in bits-per-second (bps) or frames-per-second (fps). It is possible to use ISO modifiers k/M/G, *e.g.*, 256k or 10M as specifiers for kbps and Mbps. TYPE can be one or more of: all, broadcast, multicast, and/or unknown-unicast. Set values are rounded off to the nearest possible HW setting.

Examples:

- `ingress bps 64k match bc mc`
- `ingress fps 8k`

continues on next page

continued

`[no] ingress <bps|fps> <8..1000000>[k|M|G] [match <TYPE> [<TYPES>]]`

Options

Option	Description
`no`	Disable ingress-rate-limit on the port
`RATE`	An integer rate value in the range: 8-1000000. Minimum for bps is 64k. Minimum for fps is 8k.
`TYPE`	Type of traffic. Can be one or more of: all, bc, mc, and u-uni.

`[no] egress <64..1000000>[k|M|G]`

Description

Specify the ports egress traffic rate limit in kbps. It is possible to use ISO modifiers k/M/G, e.g., 256k or 10M as specifiers for kbps and Mbps. Set values are rounded off to the nearest possible HW setting.

Options

Option	Description
`no`	Disable the egress-rate-limit on the port.
`RATE`	An integer rate value in the range: 64-1000000.

`[no] storm-control [timeout <seconds>]`

Description

When ingress rate limit threshold is reached, the port will be brought down and placed into an error state.

If a timeout is set, the function will clear the error state and bring the port up when the timer runs out. The timeout can be set between 5-300 seconds, and by default, no timeout is set.

If no timeout is set, port error state needs to be manually cleared by the user. This can either be done by using the `port clear-error PORT` or `port up PORT` commands.

Options

Option	Description
`no`	Disable storm-control for this port.
`TIMEOUT`	Timeout for port if storm-control is triggered, can be set to 5-300 seconds.

Health Monitor

Configuration commands for Health Monitor can be found in the `health-monitor` context under the port configuration context. Health Monitor offers the ability to monitor link state-changes. If the number of state-changes in a configured time window exceeds a configured threshold, Health Monitor places the port in `Err-Disabled` state. For examples of how to configure Health Monitor, refer to the Health Monitor Howto.

```
example:/#> config
example:/config/#> port eth1
example:/config/port-eth1/#> health-monitor
example:/config/port-eth1/health-monitor/#>
```

`[no] state-changes threshold <THRESHOLD> window <WINDOW> [timeout <TIMEOUT>]`

Description

Specify the threshold and window that should trigger a registration of `Err-Disabled`. An optional timeout may be given to automatically clear the error state.

If a timeout is set, the function will clear the error state when the timer runs out. The timeout can be set between 5-3600 minutes, and by default, no timeout is set.

If no timeout is set, port error state needs to be manually cleared by the user. This can either be done by using the `port clear-error PORT` or `port up PORT` commands.

Options

Option	Description
`no`	Disable Health Monitor state-changes for the port(s).
`THRESHOLD`	An integer value in the range of 5-20.
`WINDOW`	An integer value in seconds, range is 30-3600.
`TIMEOUT`	An integer value in minutes, range is 5-3600.

For increased flexibility, Health Monitor provides a optional timeout parameter that enables it to automatically clear the error state. However, in some environments it may be preferable to keep the degraded port disabled until an operator has investigated the issue. This is done by configuring Health Monitor `state-changes` without a timeout.

Additionally the CLI offers the flexibility of configuring Health Monitor either on a single port or on a range of ports.

Port Status

In order to verify a ports setting the `show` command can be used in *admin-exec* context. Note that this output can differ for different port types. For example, an SFP fibre port can contain a Digital Diagnostics Monitoring (DDM) section if supported.

`[show] port [detail|up|down|clear-error] [PORTS]`

Description

Manage or show port status.

This command can be used to display the status of all physical ports, or manage runtime (operative) state of one or more ports for debugging.

Examples:

- `show port` Displays a brief overview of port status
- `show port detail` Displays a more detailed view
- `show port eth2` Displays detailed status for port Ethernet port 2
- `port up eth2` Force port eth2 operatively up

continues on next page

continued

`[show] port [detail|up|down|clear-error] [PORTS]`

- `port down eth2` Force port eth2 down, regardless of admin state
- `port clear-error eth2` Clear all errors registered on port eth2, sets operating state equal to admin state.

```
Host-1:/#> show port eth7
eth7
Link            : Up
Oper. status    : Enabled
Admin status    : Enabled
Description     : eth7
Type            : 1000-T
State           : Forwarding
Alarm           : N/A
VLAN            : U:1
MAC             : 00:07:7c:53:df:87
Autoneg         : Disabled
Speed           : 100M
Duplex          : Full
MDI/MDIX        : Auto
MDI/MDIX status: Invalid
Polarity status: UNKNOWN
Gbit-cu-master : auto
Fastlink        : Disabled
Flow Control    : Disabled
Prio mode       : tag
Prio            : 0
Ingress limit   : None
Egress limit    : None
Storm Control   : Disabled
Health Monitor : state-changes disabled
Shielded mode   : Unshielded
PHY power mode : Low power (CLASS B)
Bridge group    : Disabled

Error State: None (effective)
```

This shows that the port is a Gigabit copper port (1000-T) and that it is enabled and has a link. It is only running at 100 Mbit/s speed and it is not using auto-negotiation. This is also indicated by the underline under the speed and duplex entries.

Error states

Protection services such as Storm Control and Health Monitor, can place a port into an error state when they detect problematic network conditions. When a port enters an error state it is operatively disabled (shown as *Err-Disabled* in port status) and no traffic will be forwarded on the port.

> **NOTE**
>
> Error states are not part of the device configuration. They only affect runtime (operative) state and are cleared automatically on reboot.

Viewing port error state

The `show port` command indicates error-disabled ports in the STATE column:

```
example:/#> show port
PORT              LINK      TYPE        SPEED  STATE           OWNER/VLAN
eth1              Down       N/A  ————————  No-Link         vlan U:1
eth2              Down       N/A  ————————  No-Link         vlan U:1
eth3              Down     1000-T  ————————  Err-Disabled    vlan U:1
eth4              Up       1000-T  1000M-Full  Forwarding     vlan U:1
eth5              Up       1000-T  1000M-Full  Forwarding     vlan U:1
eth6              Up       1000-T  1000M-Full  Forwarding     vlan U:1
eth7              Down     1000-T  ————————  No-Link         vlan U:1
eth8              Down     1000-T  ————————  No-Link         vlan U:1
eth9              Down     1000-T  ————————  No-Link         vlan U:1
eth10             Down     1000-T  ————————  No-Link         vlan U:1
```

To see which service(s) caused the error state, use `show port PORT` on the specific port:

```
example:/#> show port eth3
eth3
Link            : Down
Oper. status    : Disabled
Admin status    : Enabled
[ ... ]

Error State: Err-Disabled (effective)
STATE              SERVICE          REASON
Err-Disabled       Storm Control    Storm detected
```

Clearing port error states

There are several ways to clear error states and return a port to normal operation:

`port clear-error <PORT>`

Description

Clear all error states on the specified port.

`port clear-error all`

Description

Clear all error states on all ports.

`port up <PORT>`

Description

Force a port operatively up, which also clears any error states on that port.

```
example:/#> port clear-error eth3
example:/#>
```

> **NOTE**
>
> Some services, such as Storm Control with a configured timeout, will automatically clear their error state after the timeout expires.

Interaction with other operations

The following operations will also clear port error states:

- **Provisioning** — all error states are cleared for ports participating in provisioning when the provisioning service starts.
- **Configuration restore** — using the `copy` command, `import` command, or backup restore clears all error states.

SNMP support

The IF-MIB can be used to view and manage Ethernet port status. Ethernet ports are represented in IF-MIB as *ifType ethernetCsmacd(6)*. The MAU-MIB can be used to view and manage port speed-duplex settings.

See release zip for details on MIB conformance.

2.3.2. Link Aggregation

Introduction

Link aggregation is the act of combining multiple physical links to act as a single logical link. The system supports layer 2 link aggregation over Ethernet. The physical links are referred to here as link members.

Link aggregation primarily provides redundancy for the connection in case one of its member links fail. When this occurs data transmission continues on the remaining member links.

Link aggregation can be used to increase data throughput, but it should be noted that the data distribution algorithm distributes “flows” based per member link.

Overview

The figures below illustrate how link aggregation represents one logical link containing multiple individual links.

```
.-------------------.                .--.              .-------------------.
|                 1 +----------------|--|--------------+ 1                 |
|                   |                |  |              |                   |
|     Switch A    2 +----------------|--|--------------+ 2    Switch B     |
|                   |                |  |              |                   |
|                 3 +----------------|--|--------------+ 3                 |
'-------------------'                '--'              '-------------------'
                                 Aggregated link
```

Figure 1: Physical view

```
.-------------------.                                  .-------------------.
|                   |                                  |                   |
|                   |                                  |                   |
|    Switch A  lag1 +----------------------------------+ lag1  Switch B    |
|                   |                                  |                   |
|                   |                                  |                   |
'-------------------'                                  '-------------------'
```

Figure 2: Logical View.

The group of combined ports that make up an aggregated link is referred to as a **LAG** (Link Aggregation Group). Some other similar terms like **bond** or **team** is also generally used to describe the same thing.

Generally, two types of paradigms are considered when it comes to handling link aggregation, those are static and dynamic. Handling link aggregation, in simplistic terms, boils down to the determination process of what ports should be present in the LAG for the configured aggregate.

LACP

The most common protocol for handling dynamic link aggregation is the vendor independent standard LACP (Link Aggregation Control Protocol), defined in IEEE 802.1AX.

LACP works by continuously monitor all the links in the LAG, by sending frames (LACPDUs) on all links in the LAG. This is the methodology LACP uses to find other possible devices with LACP enabled in order to establish active link aggregates between them.

Any interfaces at the end of the links will exchange information in order to verify that they can be added to the aggregation. Further, if any active link in the LAG is found to be inaccessible it will timeout. This happens when the two interfaces on a link fails to receive LACPDUs from each other after three of them have been sent.

IEEE 802.1AX requires all links in an aggregate to be of the same speed. If they are not, the lower speed links will become non-active and detached. However, if the last high speed link becomes unavailable, automatic fail-over to the fastest lower speed link group will occur.

Static

The other option is to define a static link aggregation. In this case there is no standardised protocol, therefore most solutions here are proprietary. But most static link aggregation methods revolve around by defining what ports should constitute the aggregate. Afterwards the active links in the LAG are defined by their linkup status.

TTDP

The system also supports the Train Topology Discovery Protocol (TTDP), part of the IEC 61375 standard. It makes use of link aggregation as part of the train backbone. In order to understand how link aggregation is used and configured for use with TTDP refer to that part of the documentation.

Configuration

Link aggregates can be created from the top-level configuration context in the CLI. Notice the requirement for an identifier argument.

```
example:/#> configure
example:/config/#> lag 1
example:/config/lag-1/#>
```

The identifier can be either a number (recommended) or a name:

```
example:/config/#> lag primary
example:/config/lag-primary/#>
```

Syntax

`[no] type static|lacp|ttdp`

Description

Set the mode of operation for the aggregate.

Options

Option	Description
`no`	Reset to the default: LACP
`static`	Use static LAG handling.
`lacp`	Use LACP for LAG handling.
`ttdp`	Let TTDP handle the LAG.

`[no] ports PORT|PORT,PORT|PORT..PORT`

Description

Configure the ports that should be part of the aggregate.

continues on next page

continued

`[no] ports PORT|PORT,PORT|PORT..PORT`

Options

Option	Description
`no`	Remove all configured ports from the aggregate.
`PORT`	The port(s) to be included in the aggregate.

`[no] mode [active|passive]`

Description

Set the operational mode of an LACP type aggregate.

Options

Option	Description
`no`	Reset the setting to its default: active
`active`	Set the mode to active participation. In this mode the device will always generate LACPDUs and it actively wants to participate in the LACP protocol, regardless of its partner's mode configuration.
`passive`	Set the mode to passive participation. In this mode the device will not generate any LACPDUs by its own initiative. It will only do so if its partner is configured to be an active participant.

`[no] timeout [short|long]`

Description

Configure the LACP timeout. This dictates the how often LACPDUs are sent between hosts on the active links in the LAG, before a timeout is generated.

Options

Option	Description
`no`	Reset the setting to its default: short
`short`	Set the timeout to three seconds, LACPDUs sent every second.
`long`	Set the timeout to 90 seconds, LACPDUs sent every 30 seconds.

Status

The current status of any configured link aggregate on the device can be observed in the following manner:

```
example:/#> show aggregate
Aggregate lag1  MAC: 0c:03:fd:88:57:00  Type: LACP
-------------------------------------------------------------------------
Port     Link  Active   LACP State    Partner ID        Port  Key
-------------------------------------------------------------------------
```

```
Eth 0      UP  Yes      DISTRIBUTING  0c:03:fd:05:21:00    3     0
Eth 1      UP  Yes      DISTRIBUTING  0c:03:fd:05:21:00    4     0
```

2.3.3. Transparent Bridge

Introduction

A transparent bridge is used to configure multiple ports in a group that can only forward traffic between those ports. The ports configured in a transparent bridge is isolated from other ports, including the CPU.

Overview

The figure below shows a configured switch where ports 1 through 4 is configured as normal switched ports. Ports 5 and 6 is configured in a transparent bridge, thus traffic passing trough those ports are isolated from the other ports.

```
                    Switch A
             .--------------------.
-------------+ 1 -----\ /------ 2 +-------------
             |         X          |
-------------+ 3 -----/ \------ 4 +-------------
             |                    |
-------------+ 5 -------------- 6 +-------------
             '--------------------'
```

Figure 1: Transparent bridge for ports 5 and 6

Configuration

Transparent bridges is configured from the port contex, setting the bridge id for port(s).

```
example:/#> configure
example:/config/#> port ethX5,ethX6
example:/config/port-ethX5..ethX6/#> bridge-group 1
example:/config/port-ethX5..ethX6/#> leave
```

> **NOTE**
>
> Make sure you are not configuring the port that you are accessing the unit from in a transparent bridge as you will loose access to the unit.

Syntax

`[no] bridge-group [id]`

Description

Set the id for the transparent bridge.

Options

Option	Description
`no`	Disable the transparent bridge.
`id`	Use id (number) for the transparent bridge.

Status

The current configured transparent bridge(s) can be observed with `show port`

```
example:/#> show port
[ ... ]
ethX5            Down    100-T  ──────────  No-Link        br-grp-1
ethX6            Down    100-T  ──────────  No-Link        br-grp-1
[ ... ]
```

2.3.4. Link Layer Discovery Protocol

The Link Layer Discovery Protocol (LLDP) is a standardised layer 2 protocol (IEEE 802.1AB) which advertises information about the device itself and its capabilities to other devices within a LAN.

WeOS aims to be inline with 2005 version of the LLDP standard. An update of the standard (LLDPv2) was published in 2009, but is currently out of scope. The LLDP protocol also advertises from which port the LLDP packet was sent. This enables the unit to build up a local view of the remote ports on neighbour devices it is connected to for each local port. This information is then stored in an SNMP MIB (LLDP MIB), which can be used by NMS-systems to draw a topology map of the network.

LLDP frames in WeOS are by default advertised every 30 seconds. If an interface stops receiving frames, the neighbour information is expired after 4 advertisement intervals (configurable on sender side). LLDP can be enabled/disabled globally as well as on a per port basis.

Configuration

```
example:/#> configure
example:/config/#> lldp
example:/config/lldp/#>
```

Syntax

`[no] enable`

Description

This setting controls if LLDP is enabled or disabled.

Options

Option	Description
`no`	Disable LLDP

`[no] tx-interval INTERVAL`

Description

This setting controls how often LLDP packets are sent, e.g., `tx-interval 10` sets the interval to 10 seconds.

Options

Option	Description
`no`	Use default tx-interval 30 seconds.
`INTERVAL`	An integer interval value in the range: 5-300.

`[no] tx-hold-multiplier MULTIPLIER`

Description

The `tx-hold-multiplier` controls how many times the `tx-interval` should be multiplied to calculate the TTL (time to live) value in the LLDP packet, e.g. `tx-hold-multiplier 3` and `tx-interval 5` results in a TTL of 15 (seconds).

continues on next page

continued

`[no] tx-hold-multiplier MULTIPLIER`

Options

Option	Description
`no`	Use default tx-hold-multiplier 4.
`MULTIPLIER`	An integer multiplier value in the range: 2-10.

Port Settings

LLDP frames can be enabled/disabled per port basis. When enabled the transmission direction can also be set to either transmission only, reception only or to both (default); transmission and reception of LLDP frames.

```
example:/#> configure
example:/config/#> lldp
example:/config/lldp/#> port eth8
example:/config/lldp/port-eth8#>
```

Syntax

`[no] enable`

Description

This setting controls if LLDP is enabled or disabled on the port(s).

Options

Option	Description
`no`	Disable LLDP on the port(s).

`[no] direction <both|rx-only|tx-only>`

Description

This setting sets the transmission direction.

Options

Option	Description
`no`	Use default direction both(RX and TX).
`both`	enables both transmission and reception of LLDP frames.
`rx-only`	enables only reception of LLDP frames.
`tx-only`	enables only transmission of LLDP frames.

2.3.5. Port Monitor

Introduction

Port monitoring or mirroring, is the act of observing traffic on a specified source port, or multiple source ports, and duplicating the traffic to another provided destination port.

For an example use-case, refer to the following:

- Basic Port Monitor

Overview

It is possible to configure multiple different monitor instances that can be active at the same time. Each monitor instance can have one specified destination, and multiple different sources.

Sources

A source port, or a monitored port, is a port that will have its traffic copied/mirrored to a provided destination port. The following is relevant for monitor source ports:

- A source port cannot be configured as a destination port at the same time, in any monitor instance.
- It can only be specified as a source for one monitor instance at a time.
- Each individual source port can have its monitoring direction (ingress, egress, both) specified.
- A source port can be any physical ethernet interface on the device.

Destinations

A destination port, or a monitor port, is the port where all traffic will be received from the defined sources. The following is relevant for a destination port:

- Each monitor instance can have one destination specified.
- A single destination port can be used in multiple different monitor instances.
- A destination cannot also be a source, in any monitor instance.
- A destination port can be any physical ethernet interface on the device.

Configuration

Monitor instances can be created from the top-level configuration context in the CLI. Notice the requirement for an identifier argument.

```
example:/#> configure
example:/config/#> monitor 1
example:/config/monitor-1/#>
```

The identifier can be either a number or a name:

```
example:/config/#> monitor debug
example:/config/monitor-debug/#>
```

Syntax

`[no] enable`

Description

Enable/Disable port monitoring.

continues on next page

continued

`[no] enable`

Options

Option	Description
`no`	Disable.

`[no] destination PORT`

Description

Set monitor destination.

Options

Option	Description
`no`	Remove the specified destination port(s), or all ports if none specified.
`PORT`	The destination port to be used for the monitor instance.

`[no] source port PORT[,PORT ...] | PORT..PORT [ingress] [egress]`

Description

Set the monitoring source(s). If not specified, direction will be set to both ingress and egress.

Options

Option	Description
`no`	Remove the specified source port(s), or all ports if none specified.
`PORT`	The source port(s) to be used for the monitor instance.
`ingress`	Capture traffic that ingresses on the source(s).
`egress`	Capture traffic that egresses on the source(s).

Status

The current configured port monitor instances can be observed in the following manner.

```
example:/#> show monitor
Instance: 1
Source(s)      : eth1 inbound | outbound
Destination(s) : eth2

Instance: debug
Source(s)      : eth5 inbound | outbound
Destination(s) : eth7
```

2.3.6. Power over Ethernet (PoE)

Introduction

Some products have Ethernet ports with support for Power Over Ethernet (PoE). The purpose of PoE is to transmit power over Ethernet cabling from one device to another. The specification is published as the IEEE 802.3-2018 standard clause 33, where a power supplying device is defined as Power Sourcing Equipment (PSE) and the device receiving the power is defined a Powered Device (PD). The PoE switch products are PSEs, able to supply power to PDs.

Configuration

Configuration of PoE ports is done in the PoE context. An example of how to configure port ethX1:

```
example:/#> configure
example:/config/#> poe
example:/config/poe#> port ethX1
example:/config/poe/port-ethX1/#>
```

Syntax

`[no] enable`

Description

Enable/disable PoE on this port. If disabled, no PoE detection and classification will occur and the network port will behave as a non-PoE port. Default: Enabled

Options

Option	Description
`no`	Disable PoE port

`[no] priority <low|high|critical>`

Description

Configure PoE allocation priority setting ”priority low” is the lowest priority, while ”priority critical” is the highest. Default: Low

Options

Option	Description
`no`	Resets priority to default ‘low’.

`[no] limit <1-30>`

Description

Configure specific PoE Power limit (in Watts) on this port, e.g., ‘limit 20’ to limit the delivered power to 20 Watts. The ‘limit’ setting behaviour depends on the related limit-mode setting. Default: Disabled (no limit)

continues on next page

continued

`[no] limit <1-30>`

Options

Option	Description
`no`	Removes port specific power limits.

`[no] limit-mode <class|forced>`

Description

The limit setting consists of two modes; class and forced. The ‘limit-mode’ is only applicable for ports with a power ‘limit’ set. Default: Class

Options

Option	Description
`no`	Resets limit-mode to default ‘class’.

`show`

Description

Shows the port PoE settings.

Status Overview

To get an overview of all port statuses as well as for example the allocated and consumed power, the ‘show poe’ command can be used.

```
example:/#> show poe
PoE System Power Status
Maximum Power   :   80.0 W
Allocated Power :     76 W
Consumed Power  :     65 W
Power Usage     :     81 %
PORT    ENA  PRIORITY  LIMIT    MODE   DETECT    CLASS  POWER
ethX1   YES      High     No   Class     Good   Class3   12 W
ethX2    NO      High     No   Class  Unknown  Unknown    0 W
ethX3   YES      High     No   Class     Good   Class3   13 W
ethX4   N/A       ---    ---     ---      ---      ---     ---
ethX5    NO      High     No   Class  Unknown  Unknown    0 W
ethX6   YES      High     No   Class     Good   Class3   12 W
ethX7   YES      High     No   Class     Open  Unknown    0 W
ethX8   N/A       ---    ---     ---      ---      ---     ---
ethX9   YES      High     No   Class     Good   Class4   28 W
ethX10   NO      High     No   Class  Unknown  Unknown    0 W
ethX11  N/A       ---    ---     ---      ---      ---     ---
ethX12  N/A       ---    ---     ---      ---      ---     ---
```

Products with 160 W total will list the power status for the total as well as the power status for each group.

```
example:/#> show poe
PoE System Power Status
```

```
Maximum Power   :  160.0 W
Allocated Power :     46 W
Consumed Power  :     45 W
Power Usage     :     28 %

Power Group 1
Maximum Power   :   80.0 W
Allocated Power :     15 W
Consumed Power  :     15 W
Power Usage     :     18 %

Power Group 2
Maximum Power   :   80.0 W
Allocated Power :     31 W
Consumed Power  :     30 W
Power Usage     :     37 %

PORT     ENA  PRIORITY  LIMIT    MODE   DETECT    CLASS  POWER  PowerGroup
ethX1    N/A       ---    ---     ---      ---      ---    ---
ethX2    N/A       ---    ---     ---      ---      ---    ---
ethX3    YES       Low     No   Class     Open  Unknown    0 W           1
ethX4    YES       Low     No   Class     Good   Class3   15 W           1
ethX5    YES       Low     No   Class     Open  Unknown    0 W           2
ethX6    YES       Low     No   Class     Open  Unknown    0 W           2
ethX7    YES       Low     No   Class     Open  Unknown    0 W           2
ethX8    N/A       ---    ---     ---      ---      ---    ---
ethX9    YES       Low     No   Class     Open  Unknown    0 W           1
ethX10   YES       Low     No   Class     Open  Unknown    0 W           1
ethX11   YES       Low     No   Class     Good   Class3   15 W           2
ethX12   YES       Low     No   Class     Open  Unknown    0 W           2
ethX13   YES       Low     No   Class     Open  Unknown    0 W           2
ethX14   N/A       ---    ---     ---      ---      ---    ---
ethX15   N/A       ---    ---     ---      ---      ---    ---
ethX16   N/A       ---    ---     ---      ---      ---    ---
ethX17   YES       Low     No   Class     Open  Unknown    0 W           2
ethX18   YES       Low     No   Class     Good   Class3   15 W           2
ethX19   N/A       ---    ---     ---      ---      ---    ---
ethX20   N/A       ---    ---     ---      ---      ---    ---
```

In order to inspect an individual port's PoE status, the `show poe PORT` command can be used.

```
example:/#> show poe ethX1
PoE port ethX1
Enabled        : YES
Priority       : High
Power Limit    : No
Detection      : Good
Class          : Class3
Power Status   : OK
Consumed power : 12 W
Current        : 245 mA
Voltage        : 53477 mV
Power remaining: 3800 mW
```

In this case the port has detected a class 3 device, and is driving enough current and voltage to power it with 12 W.

PoE Power Classes

When plugging in a PD unit to a PoE port on the switch, the switch will detect the class of the connected PD, depending on the unit's resistance and thereby its maximum power consumption.

The table below lists the maximum power consumption for PDs of the different PoE classes, as well as the (somewhat higher) power actually allocated by the switch, which considers cable losses. Thus, when admitting a class 0 unit, the switch allocates 15.4 W to ensure 12.94 W reach the PoE unit.

PoE Class	Max Unit Power Consumption (W)	Allocated Power (W)
0	12.94	15.4
1	3.84	4.0
2	6.49	7.0
3	12.95	15.4
4	25.50	30.0

It is also possible to configure a maximum power limit on each individual PoE port. When setting a power limit, the default behaviour is that the port is allocated the minimum of the (a) configured power limit, and (b) the power allocated for the attached unit's class (as listed in the table above). For example, if a port's power limit is set to 12 W, then 7.0 W is allocated when connecting a class 2 PD, while 12 W is allocated if a class 3 PD is connected. This mode is referred to class based limit mode.

There is an additional limit mode: forced limit mode. The forced mode enables the switch to supply power to non-conforming PDs or even non-PoE devices. This mode goes beyond specification, but can be convenient in certain system setups. The forced mode drives power on the port even when no classification is done, assuming a power 'limit' is set for the port.

The following additional classification is made for the connected unit depending on resistance:

- **Good:** Ok. A PD is connected (Resistance within specification of PoE class 0-4).
- **Open:** Ok. Port not connected. ("Infinite" resistance, i.e., open circuit).
- **Short:** Ok, when non-PoE unit is connected (Resistance determined as short circuit).
- **Low:** The connected unit is detected as a PD and served, although its the resistance is too low to meet the PoE specification (and too high to be determined as short circuit, i.e. non-PoE unit connected).
- **High:** The connected unit is detected as a PD unit and served, although its the resistance is too high to meet the PoE specification (and too low to be determined as unconnected, i.e. open circuit).

Allocation of PoE Power

The maximum power a PoE switch can deliver is limited. When more power is requested than available, the switch will stop/refuse delivering power on the port(s) with lowest priority.

To compensate for limited accuracy in measured power consumption, your PoE switch may allow the measured and allocated power to raise somewhat above the stated maximum power of the product, before power delivery is stopped/refused on some port. The user should still ensure that PoE equipment attached to the PoE switch do not use more than the maximum power of the product in total.

© 2026 Westermo Network Technologies AB

Calculating available power and per-port power limitation

PoE power is allocated to handle max consumption by all admitted devices. For each port, the max consumption is calculated as the minimum value of:

- The power allocated to PDs of the attached class.
- The power limit configured for the port (if set), given that limit mode is class based (default).

The exception is when a power limit is set with limit mode forced; then the maximum allowed consumption is always equal to the configured power limit.

The available power is calculated as max output power of the switch, minus the sum of max power for all (admitted) ports. If a new device is attached, its allocated power will be compared to the available power:

- If there is enough power available to serve the new device, it will be admitted.
- If there is not enough power available to serve the new device, the switch will deliver power to the ports with highest priority. Thus, to admit the new device, one (or more) of the already admitted devices will be declined power.

PoE Port Priority

There are three levels of PoE priority (low, high, critical), which can be configured per port. If there is not enough power to serve all attached PDs, preference will be given to ports with higher priority.

In 160 W capable PoE products there exists two power groups, each group capable of driving 80 W. The PoE port priority is handled within each group along the same principles as with products with one power group.

As situations can occur where the switch must chose between two ports of the same level of configured priority, there is a need for a second level ”tie-break” priority. This tie-break priority is allocated in ascending port order, giving the lowest PoE port the highest tie-break priority. The ”tie-break” priority is as follows (starting with the highest tie-break priority):

- *Lynx-3510 PoE Switch*: eth3(**1**), eth4(**2**), eth5(**3**), eth6(**4**), eth7(**5**), eth8(**6**), eth9(**7**), eth10(**8**)
- *Viper-x12A PoE Switch*: ethX1(**1**), ethX2(**2**), ethX3(**3**), ethX5(**4**), ethX6(**5**), ethX7(**6**), ethX9(**7**), ethX10(**8**)
- *Viper-x20A PoE Switch*: ethX5(**1**), ethX6(**2**), ethX7(**3**), ethX11(**4**), ethX12(**5**), ethX13(**6**), ethX17(**7**), ethX18(**8**)
- *20-port 160 W capable PoE units*: Power group 1: ethX3(**1**), ethX4(**2**), ethX9(**3**), ethX10(**4**) Power group 2: ethX5(**1**), ethX6(**2**), ethX7(**3**), ethX11(**4**), ethX12(**5**), ethX13(**6**), ethX17(**7**), ethX18(**8**)

© 2026 Westermo Network Technologies AB

2.3.7. Jumbo Frame Switching

Introduction

On certain platforms it is possible to enable switching (but only switching) of Jumbo Frames. Enabling this option sets the MTU discard policy for any ports associated with a VLAN to 9000 bytes.

```
example:/#> configure
example:/config/#> system
example:/config/system/#> jumbo-frame-switching
```

The example above enables jumbo frame switching. It can be disable again by using `no jumbo-frame-switching` under `config/system`

`[no] jumbo-frame-switching`

Description

Enables switching of jumbo frames (up to 9000 bytes) on any ports associated with a VLAN.

Options

Option	Description
`no`	Disables jumbo frame switching

2.3.8. Flow Control

This configuration option is only available on devices that has support for this functionality.

Introduction

Flow control is a mechanism for stopping transmission of data on ports. The goal is to avoid packet loss when network is congested. This is accomplished by the MAC sending pause frames to the sender. Only full-duplex ports may send pause frames. For flow control to function both the ingress port (that sends the pause frames) and egress port (that is congested) on a switch needs to be configured with this option, and of course the port that should pause sending.

Configuration

```
example:/#> configure
example:/config/#> port ethX
example:/config/port-ethX/#> help flow-control
Syntax:
        [no] flow-control <autoneg|forced>
Usage:
        Set flow control, sending and reacting to pause frames.

        off      : Disable        (autoneg = 0, rx = 0, tx = 0)
        autoneg  : Autonegotiate  (autoneg = 1, rx = 1, tx = 1)
        forced   : Forced on      (autoneg = 0, rx = 1, tx = 1)
example:/config/port-ethX/#> flow-control autoneg
example:/config/port-ethX/#> show flow-control
autoneg
```

Syntax

`no flow-control OR flow-control off`

Description

Disables flow control.

Options

Option	Description
`no`	Disable.

`flow-control <autoneg|forced>`

Description

Enables flow control.

Options

Option	Description
`autoneg`	Set the port to autonegotiate flow control. Both sides of a link needs to set this for feature to be enabled.
`forced`	Set the port to force sending of pause frames.

Note

Because of differences in hardware implementation, how flow control is configured can differ.

For some platforms, it’s enough to enable flow control on the receiving port. On others, flow control must be enabled on both the receiving and transmitting port relevant for the flow of frames.

2.4. Bridging/Switching

2.4.1. Bridge Functionality

Configuration

```
example:/#> configure
example:/config/#> fdb
example:/config/fdb/#> show aging-timeout
300
example:/config/fdb/#>
```

`[no] aging-timeout <0|10..3825>`

Description

Manage aging timeout of dynamically learnt MAC entries. Default: 300 sec.

Dynamically learnt (unicast) MAC entries are refreshed whenever a new packet with the same source MAC is received on the port. Entries are removed if not refreshed within the aging-timeout value. A value of zero (0) disables aging entirely.

`[no|show] group <ADDR> [vlan VID] port <PORTS|all>`

Description

Manage static IPv4 multicast entries

- ADDR: IPv4 Multicast address, e.g., *225.1.2.3*. Note: It is not possible to add static entries for IP multicast in range 224.0.0.X or for other addresses mapping to the corresponding MAC range as 224.0.0.X, see reserved multicast groups.
- vlan VID: Setting to limit the static multicast entry to a specific VLAN. If left out, the entry applies to all VLANs.
- port <PORTS|all>: Port range of various port types:
  - Physical: E.g., *eth1,eth3..eth5*.
  - Link aggregates: E.g., *lag1*.
  - Virtual: E.g., *ssl1*
  - all: All ports. This keyword expands to all currently existing ports.

```
example:/config/fdb/#> group 226.1.1.1 port eth3,ssl1
example:/config/fdb/#> group 227.2.2.2 vlan 1 port all
example:/config/fdb/#> show group
IP Multicast Groups    VLAN  Port(s)
226.1.1.1               ALL  eth3, ssl1
227.2.2.2                 1  eth1..eth12, ssl1
example:/config/fdb/#>
```

`[no|show] mac <ADDR> [vlan VID] port <PORTS,cpu|all,cpu>`

Description

Manage static MAC multicast entries.

continues on next page

continued

`[no|show] mac <ADDR> [vlan VID] port <PORTS,cpu|all,cpu>`

- ADDR: MAC multicast address. Can *not* be used for MAC addresses within the IPv4 multicast range of MAC addresses. That is, MAC addresses in range 01:00:5e:00:00:00 – 01:00:5e:7f:ff:ff should instead be managed via the `group` command, using IP multicast addresses.
- vlan VID: Setting to limit the static multicast entry to a specific VLAN. If left out, the entry applies to all VLANs.
- port <PORTS,cpu|all,cpu>: Port range of various port types:
  - Physical: E.g., *eth1,eth3..eth5*
  - Link aggregates: E.g., *lag1*
  - Virtual: E.g., *ssl1*
  - all: All ports (excluding CPU). This keyword expands to all currently existing ports.
  - cpu: Send to internal CPU port.

```
example:/config/fdb/#> mac 01:23:45:01:01:01 port eth1,eth3..eth5
example:/config/fdb/#> mac 01:23:45:01:01:02 vlan 1 port all
example:/config/fdb/#> show mac
Static MAC Entries     VLAN  Port(s)
01:23:45:01:01:01       ALL  eth1, eth3..eth5
01:23:45:01:01:02         1  eth1..eth12, ssl1
example:/config/fdb/#>
```

NOTE

When using link aggregation, the link aggregate (e.g., *lag1*) represents all its individual member ports for the `group` or `mac` commands. Thus, physical ports, such as *eth1* and *eth2*, should **not** be added to the port list if they are member of a link aggregate.

FDB/MDB Status

```
example:/#> show fdb
GROUP                VLAN          PORT(S)
239.255.255.250         1
225.1.2.3               2          eth5..eth6, lag1

MAC                  VLAN  STATE    PORT(S)
00:07:7c:61:e5:01       1  296 s    eth1
00:07:7c:61:e5:00       1  277 s    eth1
00:80:c8:3c:a7:ff       2  300 s    eth5
00:07:7c:61:e5:41       2  298 s    eth5
00:07:7c:61:e5:40       2  277 s    eth5
01:80:c2:00:00:0e       1  mcast    eth1..eth4, ssl1
01:80:c2:00:00:0e       2  mcast    eth5..eth6, lag1
01:80:c2:00:00:03       1  mcast    eth1..eth4, ssl1
01:80:c2:00:00:03       2  mcast    eth5..eth6, lag1
01:80:c2:00:00:00       1  mcast    eth1..eth4, ssl1
01:80:c2:00:00:00       2  mcast    eth5..eth6, lag1
01:80:c2:00:00:02       1  mcast    eth1..eth4, ssl1
01:80:c2:00:00:02       2  mcast    eth5..eth6, lag1
example:/#>
```

Flood unknown traffic to CPU

```
example:/#> configure
example:/config/#> system
example:/config/system/#> flooding
example:/config/system/flooding/#> multicast auto
example:/#> show
Unicast   : Enable
Multicast: Auto
example:/#>
```

`[no] multicast <auto>`

Description

Set flooding of unknown multicast traffic to CPU.
Default: Enable

Enable or disable flooding of unknown multicast packets to the CPU.
Set ‘auto’ to automatically decide the appropriate setting for multicast flooding to the CPU.

The ‘auto’ setting will enable flooding of multicast to the CPU when either tunnels or multicast forwarding is configured.

`[no] unicast`

Description

Set flooding of unknown unicast traffic to CPU. **Default**: Enable

2.4.2. VLAN Support

Introduction

This document describes how to configure Virtual LANs (VLANs). The system supports port based VLANs and VLAN tagging according to IEEE 802.1Q. Each VLAN has a MAC filtering database, often referred to as *Independent VLAN Learning*. Up to 64 simultaneous VLANs are fully supported, and up to 255 VLANs can be used for switching.

With VLAN you can segment your LAN infrastructure into multiple LANs (broadcast domains). In its simplest form you can segment a (physical) switch into multiple logical switches. We refer to this as a *port based VLAN*.

```
      Physical View                       Logical View
.---------------------.            .---------.   .---------.
|                     |            |         |   |         |
|        Switch       |            | Switch1 |   | Switch2 |
|                     |            |         |   |         |
|  1 2 3       4 5 6  |            |  1 2 3  |   |  4 5 6  |
'--+-+-+-------+-+-+--'            '--+-+-+--'   '--+-+-+--'
   | | |       | | |                | | |         | | |
   | | |       | | |                | | |         | | |
  VLAN 1      VLAN 2                VLAN 1        VLAN 2
```

Figure 1: Using VLANs to split a physical switch into two logical switches

It is possible to extend the VLAN to span multiple switches. In that case, VLANs typically share the cables connecting the switches. We refer to these cables as VLAN trunks, or trunk links.

```
                      Physical View
.---------------------.            .---------------------.
|                     |            |                     |
|      Switch A       |            |        Switch B     |
|  1 2 3    4 5 6   7 |            | 1   2 3 4    5 6 7  |
'--+-+-+----+-+-+---+-'            '-+---+-+-+----+-+-+--'
   | | |    | | |   |   VLAN Trunk   |   | | |    | | |
   | | |    | | |   '----------------'   | | |    | | |
   VLAN 1   VLAN 2                        VLAN 1   VLAN 2


                      Logical View
.-----------. .-----------.   .-----------. .-----------.
|           | |           |   |           | |           |
| Switch A1 | | Switch B1 |   | Switch A2 | | Switch B2 |
|           | |           |   |           | |           |
|  1 2 3 7  | |  1 2 3 4  |   |  4 5 6 7  | |  1 5 6 7  |
'--+-+-+-+--' '--+-+-+-+--'   '--+-+-+-+--' '--+-+-+-+--'
   | | | '-------' | | |         | | | '-------' | | |
   | | |           | | |         | | |           | | |
  VLAN 1          VLAN 1        VLAN 2          VLAN 2
```

Figure 2: VLANs sharing both switches and cable(s)

In order to multiplex the traffic over the VLAN trunks, a *VLAN tag* is inserted into the packet. The tag holds information about VLAN Identifier (here ‘1’ or ‘2’) along with some additional signalling information. See figure below for a detailed description.

```
  VLAN Tag Control Information
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PCP |C|        VID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

PCP:   Priority Code Point (0-7)
C/CFI: Canonical Format Indicator
VID:   VLAN Identifier (0-4095)
```

Figure 3: VLAN Tag Control Information (IEEE 802.1Q)

Segmenting Your Switch

```
.---------------.
|               |
|     Switch    |
|               |
|  1 2 3 4 5 6  |
'--+-+-+-+-+-+--'
 | | | | | |
 | | | | | |
  VLAN 1
```

Figure 4a: Default VLAN setup

By default, all Ethernet ports on a switch are associated with VLAN 1 untagged, as shown in Figure 4a.

```
.---------------------.
|                     |
|        Switch       |
|                     |
|  1 2 3       4 5 6  |
'--+-+-+-------+-+-+--'
 | | |       | | |
 | | |       | | |
 VLAN 1      VLAN 2
```

Figure 4b: Intended VLAN Setup

If we wish to allocate some ports to another VLAN as in Figure 4b, we create VLAN 2 and assign it the ports untagged. Implicitly, these ports will be removed from VLAN 1, as a port *can only be associated untagged with one VLAN at a time*.

```
Switch:/#> configure
Switch:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
Switch:/config/vlan-2/#> untagged eth4..eth6
Moving untagged port eth4 from vid 1 to vid 2.
Moving untagged port eth5 from vid 1 to vid 2.
Moving untagged port eth6 from vid 1 to vid 2.
Switch:/config/vlan-2/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Switch:/#>
```

> **WARNING**
>
> Be careful when configuring VLANs on your switch remotely, as the configuration change may affect the connectivity. For example, if you connect via port 5 (Figure 4A), you may lose connectivity when the new VLAN configuration takes effect.

Now consider the case when we wish the VLAN to span multiple switches, see Figure 5 below.

```
                        Physical View
.---------------------.            .---------------------.
|                     |            |                     |
|      Switch A       |            |        Switch B     |
|  1 2 3    4 5 6   7 |            | 1   2 3 4    5 6 7  |
'--+-+-+----+-+-+---+-'            '-+---+-+-+----+-+-+--'
   | | |    | | |   |   VLAN Trunk   |   | | |    | | |
   | | |    | | |   '----------------'   | | |    | | |
  VLAN 1   VLAN 2                       VLAN 1   VLAN 2
```

Figure 5: VLANs spanning multiple switches

Ports 1-6 on switch A are configured as the example above (Figure 4). Port 7 on switch A is a VLAN trunk port, and can be configured as follows.

```
SwitchA:/#> configure
SwitchA:/config/#> vlan 2
SwitchA:/config/vlan-2/#> tagged eth7
SwitchA:/config/vlan-2/#> end
SwitchA:/config/#> vlan 1
SwitchA:/config/vlan-1/#> tagged eth7
SwitchA:/config/vlan-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
SwitchA:/#>
```

Port 1 on Switch B needs to be configured in the same way.

Concepts

Ports, VLANs, and Interfaces

Figure 6 gives a simplified view of how Ethernet ports, VLANs and network interfaces are related on your switch. It shows the factory default port assignments on the left, and a sample configuration with multiple VLANS on the right, where port 7 is used as a VLAN trunk port:

```
.--------------------.                    .-------------------------.
|   Layer 2/3 Switch |                    |     Layer 2/3 Switch    |
| .----------------. |                    | .---------------------. |
| |     Routing    | | Network Interfaces | |       Routing       | |
| |                | | (routing/mgmt/...) | |                     | |
| |      vlan1     | |                    | |vlan1   vlan2    vlan3| |
| '--------+-------' |                    | '--+-------+--------+--' |
|          |         |                    |    |       |        |    |
| .--------+-------. |       VLANs        | .--+--..---+---. .--+--. |
| |      VLAN1     | |    (switching)     | |VLAN1|| VLAN2 | |VLAN3| |
| '+-+-+-+-+-+-+-+-' |                    | '+-+-+''+-+-+-+' '+-+-+' |
|  | | | | | | | | |  |                    |  | | |  | | |  \ /  | |  |
|  1 2 3 4 5 6 7 8 9  |   Ethernet Ports   |  1 2 3  4 5 6   7   8 9  |
'--+-+-+-+-+-+-+-+--'                    '--+-+-+--+-+-+---+---+-+--'
   | | | | | | | | |                       | | |  | | |   |   | |
   | | | | | | | | |                       | | |  | | |   |   | |
```

Figure 6: Relationship between Ethernet ports, VLANs and Network Interfaces

At factory default, all ports are associated with VLAN 1. There is connectivity between all ports, using regular layer-2 switching. For every VLAN, there is also a network interface, here *vlan1*. The maximum numbers of interfaces are 64, if more than 64 VLANs are configured the additional VLANs will not have an interface. It can be assigned one or more IP addresses, enabling management of the switch via SSH, HTTP/HTTPS, and the switch can act as DHCP server to the LAN, etc.

By creating additional VLANs 2 and 3, we segment the ports into three broadcast domains. For each new VLAN, there is also a network interface, *vlan2* and *vlan3*. Switching is possible within each LAN, but to communicate across VLANs, packets need to be routed at layer 3 (IP forwarding). IP forwarding is available for products of software level *Extended*.

LAN Port Types

Two main categories of LAN port types are available:

- **Physical LAN ports:** This includes Ethernet ports. In the future, this will also include *link aggregate* ports (IEEE 802.1AX).
- **Virtual LAN ports:** This includes Layer-2 SSL VPN ports, available for products of software level *Extended*.

Virtual LAN ports have not been covered in the examples above, thus a brief introduction to virtual LAN ports and how to use them with VLANs is shown below.

```
      Alice             Internet             Bob
    .------.            .--.-.            .------.
    | L2/3 |           ( (    )__         | L2/3 |
    |Switch|          (_,  \ ) ,_)        |Switch|
    '-+--+-'          / '-'--`--'\        '-+--+-'
      |  '-----------'             '---------'  |
      |  <----------------------------------->  |
      |                L2 SSL VPN               |
  --+--+--+--+--                       --+--+--+--+--
    |  |  |  |                         |  |  |  |
    LAN (A-side)                        LAN (B-side)
```

Figure 7: Extending (V)LANs using L2 VPN bridging

Alice and Bob in Figure 7 can be referred to as *half-bridges*. They extend the LAN over some kind of backbone, in this case the Internet. Conceptually, you can think of Alice and Bob as a single switch (bridge) split into two halves (half-bridges), connected by some Layer-2 tunnelling technique, in this case an L2 SSL VPN.

```
            Alice
.---------------------------.
|     Layer 2/3 Switch      |
|  .--------------------.   |
|  | Routing/Tunnelling |   |
|  |                    |   |  Network Interface
|  |  L2VPN----------.  |   |  (routing/tunneling/mgmt)
|  |  ssl0           |  |   |
|  |   |             |  |   |
|  |   |  vlan1    vlan2|   |
|  '---|----+--------+--'   |
|  .---'    |        |      |
|  |  .-----+----..--+--.   |  VLANs
|  |  |  VLAN1   ||VLAN2|   |  (switching)
|  |  '+-+-+-+-+-'.--+--'   |
|  |   | | | | |     |      |
|  '---' 1 2 3 4     5      |  Ethernet Ports
'-------+-+-+-+------+-----'
        | | | |      |
        | | | |      '-----> (towards Internet)
      LAN (A-side)
```

Figure 8: Switch internal view of ports, VLANs and network interfaces for a half-bridge (simplified)

Creating L2 SSL VPNs is not covered here. Instead we focus on the VLAN configuration needed, assuming that the virtual L2 port *ssl0* already exists. Furthermore, we assume that all ports reside on VLAN 1 at the start, thus VLAN 2 needs to be created too.

```
alice:/#> configure
alice:/config/#> vlan 1
alice:/config/vlan-1/#> untagged ssl0
alice:/config/vlan-1/#> show untagged
U:eth1..eth5, ssl0
alice:/config/vlan-1/#> end
alice:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
alice:/config/vlan-2/#> untagged eth5
Moving untagged port eth5 from vid 1 to vid 2.
alice:/config/vlan-2/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

Access Ports & Trunk Ports

```
.---------------------.            .---------------------.
|                     |            |                     |
|      Switch A       |            |        Switch B     |
|  1 2 3    4 5 6   7 |            | 1   2 3 4    5 6 7  |
'--+-+-+----+-+-+---+-'            '-+---+-+-+----+-+-+--'
   | | |    | | |   |   VLAN Trunk   |   | | |    | | |
   | | |    | | |   '----------------'   | | |    | | |
  VLAN 1   VLAN 2                       VLAN 1   VLAN 2
    Access Ports                          Access Ports
```

Figure 9: Access ports and VLAN trunk ports

We refer to ports where end devices connect as *access ports*. On the switch, access ports are associated with a single VLAN *untagged*.

VLAN trunk ports are used to interconnect VLAN capable switches. Trunk ports are (typically) associated *tagged* with all VLANs it carries traffic for. In the example in Figure 9, port 7 on Switch A would be associated tagged with VLAN 1 and VLAN 2.

To distinguish between VLANs over a VLAN trunk, it is not strictly required to associate all VLANs tagged; it is possible to have one VLAN untagged, and all the others tagged. Still, the general *recommendation is to configure the VLAN trunk ports tagged on all VLANs it carries*. Using VLAN tagging enables packets to carry priority (see Figure 3 above), and avoids unintended VLAN leakage if the switches at each side of the trunk happens to configure different VLANs untagged.

As an example of VLAN tagging, consider a host (H1) connected to port 4 on Switch A:
1. H1 sends a broadcast packet
2. Switch A will forward the packet untagged to ports 5 and 6, and tagged (VID 2) on ports 7
3. Switch B receives the packet on its port 1 and determines it is associated with VLAN 2
4. Switch B forwards the packet untagged on ports 5-7

The distinction between access ports and VLAN trunk ports should not be taken too strictly. For example, there may be VLAN aware hosts, capable of sending/receiving tagged packets. A special case is when host sends *priority tagged* packets, i.e., a tagged packet with VID 0 (zero). Hosts can do that to signal the (wished) priority of its traffic on a packet by packet basis. For example, if a host on port 4 of switch A sends a priority tagged packet (VID 0), switch A will still associate it with VID 2. Thus, switch A will tag it with VID 2 when forwarding the packet over the trunk. Depending on the L2 QoS configuration on switch A, it may determine the packet's priority based on the information in the tag.

Isolated Port Interfaces

The systems allows for isolating ports by removing them from any VLAN membership. This has the added benefit of making the available for use as a layer-3 interface.

The below figure show the factory default port assignments on the left, and an isolated port, or simply port interface, on the right:

```
.--------------------.                    .-----------------------.
|   Layer 2/3 Switch |                    |    Layer 2/3 Switch   |
| .----------------. |                    | .-------------------. |
| |     Routing    | | Network Interfaces | |      Routing      | |
| |                | | (routing/mgmt/...) | |                   | |
| |      vlan1     | |                    | |     vlan1     eth9| |
| '--------+-------' |                    | '-------+---------+-' |
|          |         |                    |         |         |   |
| .--------+-------. |       VLANs        | .-------+-------. |   |
| |      Bridge    | |    (switching)     | |     Bridge    | |   |
| '+-+-+-+-+-+-+-+-' |                    | '+-+-+-+-+-+-+-+' |   |
|  | | | | | | | | |  |                    |  | | | | | | | |  |   |
|  1 2 3 4 5 6 7 8 9  |   Ethernet ports   |  1 2 3 4 5 6 7 8  9   |
'--+-+-+-+-+-+-+-+--'                    '--+-+-+-+-+-+-+-+--+---'
   | | | | | | | | |                       | | | | | | | |  |
   | | | | | | | | |                       | | | | | | | |  |
```

Figure 10: Removing port 9 from VLAN 1 turns it into a port interface

> **NOTE**
>
> The *bridge* shown in the image exists both in the switching fabric and in the software stack. When a port is isolated it is removed from the software bridge, and in the switching fabric it is isolated from all other ports using device specific mechanisms similar to that used for implementing IEEE 802.1X port authentication.

An example of the port interface concept is shown in **Figure 10**. The steps to achieve this are as follows:

```
Switch:/#> configure
Switch:/config/#> vlan 1
Switch:/config/vlan-1/#> no untagged eth9
Switch:/config/vlan-1/#> show untagged
U:eth1..eth8
Switch:/config/vlan-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Switch:/#>
```

Port *eth9* can then be used as a regular network interface, e.g., one can assign an IP address to it.

```
Switch:/#> configure
Switch:/config/#> iface eth9
Switch:/config/iface-eth9/#> inet static 192.168.2.1/24
Switch:/config/iface-eth9/inet-static-192.168.2.1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Switch:/#>
```

With port interfaces, the half-bridge configuration for L2 SSL VPNs above could be simplified. Port 5 could be turned into a port interface, removing the need to create a second VLAN for the connection to the Internet.

```
          Alice
.--------------------------.
|    Layer 2/3 Switch      |
|  .--------------------.  |
|  | Routing/Tunnelling |  |
|  |                    |  |  Network Interface
|  | L2VPN----------.   |  |  (routing/tunneling/mgmt)
|  | ssl0           |   |  |
|  |  |             |   |  |
|  |  |  vlan1    eth5  |  |
|  '--|----+--------+---'  |
| .---'    |        |      |
| |  .-----+----.   |      |  VLANs
| |  |   VLAN   |   |      |  (switching)
| |  '+-+-+-+-+-'   |      |
| |   | | | | |     |      |
| '---' 1 2 3 4     5      |  Ethernet Ports
'-------+-+-+-+-----+------'
        | | | |     |
        | | | |     '-----> (towards Internet)
    LAN (A-side)
```

Figure 11: Use of port interface removes the need for a second VLAN

Compare this with **Figure 8**, above.

Configuration

Individual VLANs

Individual VLAN configuration is accessed from the global configuration context. When creating a VLAN the VID must be in the range `1-4095`. For example to configure VLAN 10:

```
example:/#> configure
example:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
example:/config/vlan-10/#>
```

To remove a configured VLAN the following command can be executed:

```
example:/#> configure
example:/config/#> no vlan 10
```

Syntax

`[no] name <STRING>`

Description

Set a name, or description, of the VLAN.

Options

Option	Description
`no`	Remove the configured name.
`STRING`	Free from string representation of the name.

`[no] tagged [PORT | PORT..PORT | PORT,PORT | all]`

Description

Set tagged ports for the VLAN.

> **NOTE**
>
> A port can be configured tagged in multiple VLANs. However, a port cannot be associated both tagged and untagged in the same VLAN. Configuring a port as tagged automatically remove any untagged configuration for the port on that VLAN.

> **WARNING**
>
> Be extra careful when configuring VLANs remotely!

Example:

```
example:/config/vlan-10/#> tagged eth1, eth10..eth12
```

Options

Option	Description
`no`	Remove all configured ports.
`PORT`	Name of a port, example: eth1, ssl1, lag1.

`[no] untagged [PORT | PORT..PORT | PORT,PORT | all]`

Description

Set untagged ports for the VLAN.

> **NOTE**
>
> A port can only be configured untagged for one specific VLAN at a time. When configuring a port as untagged on a VLAN the system automatically moves it from any other VLAN it was untagged in. Configuring a port as untagged automatically removes any tagged configuration for the port in that VLAN.

> **WARNING**
>
> Be extra careful when configuring VLANs remotely!

Example:

```
example:/config/vlan-10/#> untagged eth1, eth10..eth12
```

continues on next page

continued

`[no] untagged [PORT | PORT..PORT | PORT,PORT | all]`

Options

Option	Description
`no`	Remove all configured ports.
`PORT`	Name of a port, example: eth1, ssl1, lag1.

`[no] forbid [PORT | PORT..PORT | PORT,PORT | all]`

Description

Set forbidden ports, only applicable when dynamic VLANs are active.

Options

Option	Description
`no`	Remove all configured ports.
`PORT`	Name of a port, example: eth1, ssl1, lag1.

`[no] multicast-snooping`

Description

Enable, or disable IGMP/MLD snooping for the specific VLAN interface.

Options

Option	Description
`no`	Disable IGMP/MLD snooping.

IEEE 802.1Q Tunneling

Q-in-Q enables traffic tagged with customer vlan VLAN ID (CVID) to be tunneled through provider service VLAN ID (SVID). Customer connect to provider access port, traffic that ingress access port will be encapsulated with a second VLAN tag, provider SVID, and this tag is removed upon egressing through an access port. This allows the customer to use any CVIDs on their internal network.

VLAN Q-in-Q Configuration is similar to creating Vlan, the only difference lies in configuring an access port on the provider switch. The port that encapsulate the customer vlan into provider vlan configured as nest port.

VLAN Q-in-Q provider network

Alice and Bob are connected to their respective providers providedA and providedB access port (eth1) and the providers connected to each other on eth2

VLAN Q-in-Q Customer network

```
Alice:/#> configure
Alice:/config/#> vlan 2
Alice:/config/vlan-2/#> tagged eth1
Alice:/config/#>
```

```
Bob:/#> configure
Bob:/config/#> vlan 2
Bob:/config/vlan-2/#> tagged eth1
Bob:/config/#>
```

VLAN Q-in-Q provider network

```
providedA:/#> configure
providedA:/config/#> vlan 10
providedA:/config/vlan-10/#> untagged eth1
providedA:/config/vlan-10/#> nest eth1
providedA:/config/vlan-10/#> tagged eth2
providedA:/config/#>
```

Syntax

`[no] nest <PORT|PORTS>`

Description

Set untagged port or range of ports nest policy

When nest policy is enabled on a port it serves as an access port by disabling IEEE 802.1Q and allowing customer VLAN ID (CVID) tagged frame to ingress in untagged port.

When a CVID tagged frame enters the port it is encapsulated with the port's own VLAN ID (SVID).

This encapsulation ensures that the frame can be correctly forwarded within the network

Traffic egressing on access port are restored to original CVID, SVID information is stripped off

Example:

```
example:/config/vlan-10/#> nest eth1,eth10..eth12
```

Options

Option	Description
`no`	The no command will set the ports policy to default 'dot1q' With 'no nest' all ports are set to policy 'dot1q'.

A range of ports is denoted as: PORTA..PORTN

```
providedB:/#> configure
providedB:/config/#> vlan 10
providedB:/config/vlan-10/#> untagged eth1
providedB:/config/vlan-10/#> nest eth1
providedB:/config/vlan-10/#> tagged eth2
providedB:/config/#>
```

General VLAN settings

General VLAN configuration is accessed from the global configuration context, and are global to all individual VLANs. It can be accessed in the following manner:

```
example:/#> configure
example:/config/#> vlans
example:/config/vlans/#>
```

Syntax

`[no] vlan-iface-mac-mode [base-mac | port-inherit-lbit]`

Description

Set the default mac assignment mode for all VLAN interfaces. This command allows changing the default behaviour how MAC addresses are assigned to VLAN interfaces. This setting is inherited to all VLANs on the device.

Default: `base-mac`

Options

Option	Description
`no`	Revert to the default setting.
`base-mac`	The MAC address for each VLAN will be the same as the Base MAC address on the unit.
`port-inherit-lbit`	Generate a locally administered address based on the ports that constitute the VLAN. Addresses are considered in the following order: 1. A globally administered address is preferred over a locally administered address. 2. An address belonging to an untagged member of the VLAN is peferred over one belonging to a tagged member. 3. A lower address is preferred over a higher one. This assignment mode is useful when interoperating with networking equipment that does not support independent VLAN learning (IVL) as it typically generates a distinct address for each VLAN.

Status Overview

In order to see all VLANs that have been configured on the system, the following command can be executed from the exec context in the CLI:

```
example:/#> show vlan
VID  NAME              UNTAGGED/TAGGED
   1 vlan1             U:eth0..eth6
                       T:
                       N:
   2 vlan10            U:eth7
                       T:
                       N:
```

2.4.3. Layer-2 QoS Config Guide

About

This document describes the Layer-2 QoS functionality in switched LANs.

Layer-2 QoS Priority

When Ethernet operates in *full-duplex* mode there is no risk for collisions. However, there is a risk for *congestion* if the traffic destined for a port exceeds its capacity.

Consider the example below, showing a switch with 4 Fast Ethernet ports (100 Mbit/s). Ports 1 and 2 both receive data flows of 100 Mbit/s load. All traffic from port 2, and half of the traffic port 1 are heading for port 4, exceeding the capacity of port 4. The egress queue (output queue) at port 4 will build up, and if the situation sustains, packets will eventually have to be dropped.

```
           .-------------------------.
100 Mbit/s |1             50 Mbit/s 3| 50 Mbit/s
---------->+----------+------------->+---------->
           |           \             |
           |            \ 50 Mbit/s  |
100 Mbit/s |2 100 Mbit/s \          4| 150 Mbit/s > 100 Mbit/s!!
---------->+--------------+--------->+---------->
           |                         |
           '-------------------------'
```

Figure 1: Load from multiple ports exceeds egress/output capacity.

Layer-2 priority address two issues in this scenario:

- It ensures that low priority packets are dropped before high priority packets. For example, assume packets entering port 1 have higher priority than packets from port 2. Then we would not expect any loss in packets from port 1 destined for port 4, while 50% of the ‘’port 2‘‘ packets would be lost, once the port 4 egress queue becomes full.
- When the is a queue on an egress port, layer-2 priority ensures that the high priority packets are transmitted first, assuming the switch implements *strict priority scheduling*.

The example above illustrates congestion occurring when a traffic load from multiple ports exceeds the capacity of the egress port where it is heading. The same could occur with traffic from a single port, if the egress port has lower capacity; e.g., a Gbit/s ingress port and a Fast Ethernet egress port, or if ingress and egress ports are of the same type, but egress rate limiting is applied to reduce the speed of the egress port.

Traffic classes, priority and congestion

Each port has multiple priority queues for egress traffic. We refer to *egress queues* and *traffic classes* interchangeably. Switches described here have 8 egress queues per port, thus can prioritise between 8 traffic classes. (Switches with 4 egress queues per port are also common, providing somewhat lower prioritization granularity.)

The example below illustrates the use of output queues to manage priority. To make the example more compact, these ports only support 4 traffic classes.

```
- Step 1: Packet W ingress P2, heads for P5
                   .---------------------.
                   |            --Q3--.  |
     ------------->+P1            | | |  |
                   |            ------'  |
            [W]->  |            --Q2--.  |
     ------------->+P2            | |T|  |
                   |            ------'  |
                   |            --Q1--.P5+-----------
     ------------->+P3            |S|R|  |
                   |            ------'  |
                   |            --Q0--.  |
     ------------->+P4            | |U|  |
                   |            ------'  |
                   '---------------------'

- Step 2: Packet W mapped to Queue 1 (last in FIFO)
                   .---------------------.
                   |            --Q3--.  |
     ------------->+P1            | | |  |
                   |            ------'  |
                   |            --Q2--.  |
     ------------->+P2            | |T|  |
                   |            ------'  |
                   |            --Q1--.P5+-----------
     ------------->+P3           W|S|R|  |
                   |            ------'  |
                   |            --Q0--.  |
     ------------->+P4            | |U|  |
                   |            ------'  |
                   '---------------------'

- Step 3: High priority queues are served first
                   .---------------------.
                   |            --Q3--.  |
     ------------->+P1            | | |  |
                   |            ------'  |
                   |            --Q2--.  |
     ------------->+P2            | | |  |
                   |            ------'  |  [R][T]->
                   |            --Q1--.P5+--------------------
     ------------->+P3            |W|S|  |
                   |            ------'  |
                   |            --Q0--.  |
     ------------->+P4            | |U|  |
                   |            ------'  |
                   '---------------------'
```

Figure 2: Example with 4 Queues/Traffic Classes.

In step 2, packet W is mapped to traffic class 1 (queue 1). This is just an example. See below for information on determining packet priority.

Additional information on traffic classes and queue management:

- Ethernet ports support 8 traffic classes (8 egress queues per port)
- Virtual LAN ports, e.g., layer-2 VPN SSL ports, also support 8 traffic classes
- High priority queues are emptied first (strict priority scheduling)
- If a high priority packet arrives while a packet from a low priority queue is being transmitted, the transmission will finish before serving the high priority queue (no *pre-emption*)

Determining a Packet's Priority

A packet's priority is determined on ingress, i.e., when the packet enters the switch. Three *priority modes* are available for per-prot configuration.

- Port Priority mode
- VLAN tag mode

- IP DSCP/ToS mode

The below sections detail the behaviour of each mode.

Port Priority mode

If priority mode ‘port’ is configured on the port, then all ingress packets will be assigned the ingress port’s *port priority*. The *port priority* can be configured in range 0-7 (default ‘0’). Thus, a packet entering a port with port priority ‘2’ and priority mode ‘port’ will is determined to have (IEEE) priority ‘2’.

VLAN tag mode

If priority mode VLAN ‘tag’ is configured on the port, then all ingress packets arriving with an 802.1Q tag will get priority specified in the PCP field of the VLAN tag (IEEE 802.1p/802.1Q).

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PCP |C|        VID            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

PCP:   Priority Code Point (0-7)
C/CFI: Canonical Format Indicator
VID:   VLAN Identifier (0-4095)
```

Figure 3: VLAN Tag Control Information (IEEE 802.1Q).

Thus, a packet with PCP value ‘3’ in its VLAN tag, entering a port configured with priority mode ‘tag’ is determined to have (IEEE) priority ‘3’.

If no VLAN tag is present in the arriving packet, priority is based on the *port priority* of the ingress port, see above.

IP DSCP/ToS mode

If priority mode ‘ip’ is configured, then all ingress IPv4 packets will be assigned priority based on the DSCP value (formerly ToS) in the packets IP header.

```
+-+-+-+-+-+-+-+-+
|   DSCP    |CU |
+-+-+-+-+-+-+-+-+

DSCP: Differentiated Services Codepoint (0-63)
CU:   Currently Unused
```

Figure 4: Differentiated Services Field (RFC 2474).

Thus, a packet with DSCP value ‘42’ in its IP header, entering a port configured with priority mode ‘ip’ is determined to have (IP) priority ‘42’.

If the incoming packet is not an IP packet (e.g., an ARP packet), priority is based on the *port priority* of the ingress port, see above.

By *default*, ports are configured with:
- priority mode ‘VLAN tag’
- port priority ‘0’ (zero)

Below is an example where *priority mode* is changed to ‘port’ and the *port priority* is changed to 5 for port 2 (eth2).

```
example:/#> configure
example:/config/#> port eth2
example:/config/port-eth2/#> priority-mode port
example:/config/port-eth2/#> priority 5
example:/config/port-eth2/#> leave
```

```
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

> **NOTE**
>
> Packets relating to layer-2 network control (RSTP, FRNT, etc.) are typically assigned highest (IEEE) priority ('7'), irrespective of the priority mode configured for the port.

Mapping Priority to Traffic Class

The scheme to map priority to traffic class depends on the priority mode used to determine the packet's priority as described in the following two subsections.

IEEE Priority to Traffic Class

When priority is based on VLAN tag or port priority, a priority value in range 0-7 is determined for the packet. This priority is referred to as *IEEE priority*. For ports with 8 traffic classes, the IEEE Priority to Traffic Class mapping scheme is a simple 1-to-1 mapping.

IEEE Priority	Traffic Class
7 (111)	7 (111)
6 (110)	6 (110)
5 (101)	5 (101)
4 (100)	4 (100)
3 (011)	3 (011)
2 (010)	2 (010)
1 (001)	1 (001)
0 (000)	0 (000)

IP (DSCP) Priority to Traffic Class

When priority is based on the IP DSCP field, a priority value in range 0-63 is determined for the packet. This priority is referred to as *IP priority*. For ports with 8 traffic classes, the IP Priority to Traffic Class mapping scheme looks at the three most significant bits (MSBs) of the DSCP field as shown below.

IP (DSCP) Priority	Traffic Class
56-63 (111xxx)	7 (111)
48-55 (110xxx)	6 (110)
40-47 (101xxx)	5 (101)
32-39 (100xxx)	4 (100)
24-31 (011xxx)	3 (011)
16-23 (010xxx)	2 (010)
8-15 (001xxx)	1 (001)
0-7 (000xxx)	0 (000)

VLAN tag priority upon egress

A packet leaving the switch with a VLAN tag, will have its VLAN tag priority (PCP bits) set to a value corresponding to its traffic class (0-7).

Rate Limiting

Ingress Rate Limiting

The switch can be configured to limit the rate of a port's incoming traffic, this is called *ingress rate limiting*. By default a port accepts packets at a rate up to the link speed, with ingress rate limiting activated the switch starts dropping packets when data arrives above the given rate threshold. This feature can be useful as a complement to layer-2 priority handling when congestion within the network is to be avoided.

The rate may be specified in bits per second or frames per second, such as 64 kbit/s, or 250 kfps/s. By default, a configured rate limit will apply to all ingress traffic, however, it is possible to restrain the rate limit to specific (layer-2) traffic types: *broadcast*, *multicast* and/or *unknown unicast*.

Unknown unicast traffic is traffic with a unicast destination MAC address not present in the switch forwarding database. Unknown unicast traffic is flooded onto all ports within the VLAN.

> **NOTE**
>
> It is only possible to configure rate limit on physical ethernet ports.

```
example:/#> configure
example:/config/#> port eth1
example:/config/port-eth1/#> rate-limit
example:/config/port-eth1/rate-limit/#> ingress bps 2M
example:/config/port-eth1/rate-limit/#> end
example:/config/port-eth1/#> end
example:/config/#> port eth2
example:/config/port-eth2/#> rate-limit
example:/config/port-eth2/rate-limit/#> ingress bps 1M match bc mc
example:/config/port-eth2/rate-limit/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

Egress Rate Limiting

The switch can be configured to limit the outbound data rate on a port egress rate limiting. By default each port will send at the maximum speed of the link, with egress rate limiting activated the switch limits the outbound rate to a given threshold.

The rate may be specified in bits per second, such as 350 kbit/s, or 500 Mbit/s. If egress rate limit is set, *all* outgoing traffic is limited to the specified threshold.

```
example:/#> configure
example:/config/#> port eth3
example:/config/port-eth3/#> rate-limit
example:/config/port-eth3/rate-limit/#> egress 8M
example:/config/port-eth3/rate-limit/#> show egress
8000 kbps
example:/config/port-eth3/rate-limit/#> leave
```

```
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

Configuration

Per-port priority settings are located in the port configuration context in the CLI. One or more *physical* LAN ports can be configured at the same time. The range can be given as: `port eth3,eth5..eth10`

```
example:/#> configure
example:/config/#> port eth3
example:/config/port-eth3/#>
```

When you are done, use `leave` to activate settings and return to *Admin Exec* context.

Syntax

`[no|show] priority-mode [tag|ip|port]`

Description

Select the classification mode used at ingress; `tag` for VLAN tag (PCP), `ip` for DSCP/ToS, and `port` for port priority. The `no` form of the command resets the setting to its default value, `tag`.

`[no|show] priority [0-7]`

Description

If priority mode is set to `port` this setting defines the priority assigned to ingressing packets. The `no` form of the command resets the setting to its default value, `0` (zero).

Commands for ingress- and egress-rate-limit are found under the rate-limit configuration context.

`[no|show] ingress <bps|fps> <64-1000000>`

Description

Set ingress limit in kbps or kfps. The value can also be given with ISO modifiers k/M/G as specifiers for kbps and Mbps (or kfps and Mfps), e.g., 256k or 10M. The no form of the command disables rate limiting, which is the default.

`[no|show] ingress <bps|fps> <64-1000000> match <bc mc u-uni|all>`

Description

By default, the rate limit applies to all traffic types. To limit a specific type of traffic, use the optional `match [bc mc u-uni]` setting to limit MAC broadcast (bc), MAC multicast (mc) or unknown MAC unicast (u-uni). E.g., `ingress bps 2M match bc mc` limits the total MAC broadcast *and* multicast to 2 Mbps.

`[no|show] egress-rate-limit <64-1000000>`

Description

Set egress rate limit in kbps. The value can also be given with ISO modifiers k/M/G as specifiers for kbps and Mbps, e.g., 256k or 10M. The no form of the command disables rate limiting, which is the default.

> **NOTE**
>
> Ingress and egress rate values are always rounded off to the nearest possible setting in the underlying switch hardware.

2.4.4. IGMP/MLD Snooping

About

This document gives a brief overview of multicast within a LAN, with a focus on IP multicast, and how it can be controlled using IGMP/MLD snooping. It also covers non-IGMP/MLD capable devices and how they can be integrated into a network with IGMP/MLD snooping enabled.

Introduction

Multicast and broadcast, as opposed to unicast, can be used to communicate to more than one receiver. The main difference between multicast and broadcast is that multicast can be controlled. When disabling its control mechanisms, multicast behaves like broadcast.

To control distribution of multicast within a LAN the following main primitives are available:

- *Flood or block per default*: It is possible to control per port if traffic to *unknown multicast* should be flooded or blocked. The default is to *flood*.
- *Dynamically limit IP multicast*: By enabling *IGMP/MLD snooping*, a switch can learn on which ports subscribers of IP multicast addresses reside. Traffic to these *known multicast* addresses is only forwarded to ports leading to subscribers (and towards multicast routers/IGMP or, MLD Queriers).
- *Static filters*: It is possible to manually specify on which ports to forward a multicast address.

Settings for the first two primities are covered here, while static filters is documented in the transparent bridge configuration guide. With these primitives, switches can distribute IP multicast data within a LAN in the following ways:

1. treat multicast traffic as broadcast – always flood on all ports
2. flood until subscribers appear, then limiting specific flows
3. (always) limit forwarding of multicast only to subscribers

Cases 2 and 3 assumes IGMP/MLD snooping is enabled.

To limit IP multicast requires switches to inspect Internet Group Management Protocol (IGMP), or Multicast Listener Discovery (MLD) control messages, which exchanged by hosts and routers to learn which ports lead to subscribers – this mechanism is referred to as *IGMP/MLD snooping*, RFC 4541[1].

```
                  Switch
                 .------.    MC
                 |      | =======> .---.
                 |  .---+----------+ H |
          MC     | /    |    MC    '---'
.---. =========> |/     | =======> .---.
| H +------------+------+----------+ H |
'---'            |\     |    MC    '---'
                 | \    | =======> .---.
                 |  `---+----------+ H |
                 '------'          '---'
```

Figure 1: Multicast flow flooded like broadcast traffic. This is the case when flooding of unknown traffic is enabled and when IGMP/MLD snooping is disabled, or it is enabled, but no subscriber has joined.

[1] https://tools.ietf.org/html/rfc4541

```
                  Switch
                 .------.
                 |      |          .---.
                 |      +----------+ H |
          MC     |      |          '---'
.---. =========> |      |          .---.
| H +------------+      +----------+ H |
'---'            |      |          '---'
                 |      |          .---.
                 |      +----------+ H |
                 '------'          '---'
```

Figure 2: Flow of multicast blocked by switch. IGMP/MLD snooping is enabled, but no subscriber has joined. Flooding of unknown multicast is disabled.

```
                  Switch
                 .------.    MC
                 |      | =======> .---.
                 |  .---+----------+ H |
          MC     | /    | <---Join '---'
.---. =========> |/     |          .---.
| H +------------+      +----------+ H |
'---'            |      |          '---'
                 |      |          .---.
                 |      +----------+ H |
                 '------'          '---'
```

Figure 3: Multicast flow forwarded to subscriber (only). IGMP/MLD snooping is enabled and a single subscriber has joined. The switch learns the location of subscribers by sniffing IGMP/MLD join messages.

The IGMP/MLD snooping support also features support for acting as an *IGMP/MLD querier* – a role which is usually handled by a *multicast router*. Having switches with IGMP/MLD querier capabilities enables efficient distribution of IP multicast in networks without multicast routers.

> **NOTE**
>
> Most switches only limit the broadcast effects of multicast on layer-2, i.e. MAC level. It is therefore important to design IP multicast networks so that groups do not overlap. E.g. 225.1.2.3 and 226.1.2.3 map to the same multicast MAC address and will effectively be treated as the same group in the switch fabric. Meaning, both groups will be forwarded by switches and may potentially overload the intended receiver. See RFC 1112[1] and RFC 6085[2] for details on how IPv4 and IPv6 multicast groups, respectively, map to MAC multicast addresses.

IGMP/MLD Snooping

With IGMP Snooping enabled, switches continuously track subscribed multicast groups and limit their reach to the abilities of the underlying switch fabric. Devices support limiting up to 2048 groups, but ability to limit is also dependent on the distribution of learnt MAC addresses.

[1] https://tools.ietf.org/html/rfc1112
[2] https://tools.ietf.org/html/rfc6085

> **NOTE**
>
> Both unicast and multicast MAC addresses are stored hashed in the switch's MAC database. When addresses hash to the same location they are stored in one of four "bins", when all these are occupied the database is exhausted. The fifth MAC address cannot be stored, and an *ATU full* warning message will appear in syslog. The MAC address continues to be *unknown* and is thereby subject to flooding policies.

Multicast is either *known* or *unknown*. I.e., the switch either has a matching multicast group entry in its MAC database or not.

```
   Switch               Switch               Switch
.---------.   MC X   .---------.   MC X   .----------.
|      .--| -------> |---------|  ------> |--.       |
|     /   +----------+         +----------+   \      |
|    /   .| .......> |.........| .......> |.   \     |
|   /   . |   MC Y   | .       |   MC Y   | .   \    |
'----+----'          '----+----'          '----+-----'
   ^ | ^                . |                  . | | ^
   | | . MC          MC . |              MC  . | | | Join X
   | | . X,Y          Y . |             X,Y  . | | |
   | | .                v |                  v | v |
 .---+---.            .---+---.             .--+----.
 |   H   |            |   H   |             |   H   |
 '-------'            '-------'             '-------'
```

Figure 4: Flooding of unknown multicast (Y, dotted) to all, known multicast (X, dashed) only to subscribers.

In WeOS IGMP/MLD snooping allows for the paradigm to flood multicast until the network knows more. Here IP multicast is broadcast until an IGMP/MLD *Join*[1] message from an end device is received, IGMP/MLD snooping in the switch detects this, dynamically installs a MAC entry to reclassify the group as known and then forwards the IGMP/MLD report towards the IGMP/MLD Querier on the LAN. In the event of MAC db exhaustion (see note on *ATU full* above) in this setup, the multicast will still be forwarded.

Flooding of multicast applies to both IP and MAC multicast alike, IGMP/MLD can only limit IP multicast. For the most part, users will likely opt for either flood on all ports (the safe default), or none:

- With flooding of unknown multicast *enabled* on all ports, all multicast traffic is forwarded on all ports in the same VLAN, i.e., like broadcast traffic. Only IP multicast may be limited in this mode, and only while at least one subscriber of an IP multicast group is active. The default is to flood on all ports (excluding CPU).
- With flooding of unknown multicast *disabled* on all ports, multicast packets are only forwarded on ports leading to a subscriber of an IP multicast group, ports leading to an IP multicast router (i.e., the elected querier or a statically configured multicast router port.), and ports in static FDB filters.

Enabling and disabling of IGMP/MLD snooping is both controlled globally and per VLAN. By default, IGMP/MLD snooping is enabled both globally and per VLAN.

- When IGMP/MLD Snooping is disabled globally, the per-VLAN settings are ignored and IGMP/MLD Snooping is completely disabled on the device.

[1] In IGMPv2 the terms are *join* and *leave*, with the collective name IGMP *report*. In IGMPv3 only *report* is used.

- When IGMP/MLD Snooping is enabled globally, the per-VLAN settings control multicast filtering per VLAN. E.g., it is possible to disable IGMP/MLD snooping on individual VLANs as shown in the example below.

```
example:/config/ip/#> show multicast-snooping
Enabled, can be enabled/disabled per VLAN
example:/config/ip/#> end
example:/config/#> vlan 1
example:/config/vlan-1/#> no multicast-snooping
example:/config/vlan-1/#> show multicast-snooping
Disabled
example:/config/vlan-1/
```

With IGMP/MLD snooping *enabled* on a VLAN, IP multicast packets are only forwarded to ports leading to a subscriber of that IP multicast group, and to ports leading to an IP multicast router,

With IGMP/MLD snooping *disabled* on a VLAN, the switch will not inspect IGMP/MLD packets to learn the location of subscribers or Queriers. Thus, IP multicast traffic will treated as *unknown* multicast traffic.

Ports shared between multiple VLANs may have different IGMP/MLD snooping settings on different VLANs, i.e., one VLAN may have IGMP/MLD snooping *enabled* and another may have it *disabled*. As WeOS5 learns MAC address per VLAN, IP multicast traffic can be limited on that port for the VLAN with IGMP/MLD snooping enabled, while multicast traffic on another VLAN is flooded.

Settings for controlling flooding of unknown multicast, and for globally enabling/disabling IGMP/MLD snooping are located in the *IP configuration* context:

```
example:/#> config
example:/config/#> ip
example:/config/ip/#>
```

`[no] multicast-snooping`

Description

Allow per VLAN IGMP/MLD snooping, or disable it entirely. Default: Enabled

When enabled the per-VLAN ‘multicast-snooping’ setting controls if IGMP/MLD snooping should be enabled. When this setting is disabled all IGMP/MLD snooping is disabled, regardless of the per-VLAN setting

`[no] multicast-flood-unknown [PORTS]`

Description

Control flooding of unknown multicast.

This setting controls the behaviour of unknown multicast. Also called unregistered multicast. When this setting is enabled, by default all ports, both unknown MAC and IP multicast from other ports in the same VLAN is flooded on these ports, in the same VLAN.

When disabled, unknown multicast is blocked, unless the port is also configured as a static multicast-router-port, or IGMP/MLD snooping has registered the port a dynamic multicast router port due to an upstream IGMP/MLD querier or PIM router. To forward IPv4 multicast, or

continues on next page

continued

`[no] multicast-flood-unknown [PORTS]`

MAC multicast, a static FDB entry must be set or an end-device on that port must send an IGMP membership report for the IPv4 group.

When IGMP snooping is enabled and an end-device reports/joins a multicast group, only that group is filtered. The filtering is per group and per device/VLAN, depending on the underlying HW.

> **NOTE**
>
> Certain overlapping multicast groups may be hashed to the same “bucket” in the underlying hardware. When the bucket is full, filtering cannot be done. When this happens, the system generates a warning message to the system log and the IGMP status for the given group has a ‘!’ flag. It is recommended to leave this setting at its default value.

The setting to enable/disable IGMP snooping per VLAN is available in the VLAN configuration context:

```
example:/#> config
example:/config/#> vlan 1
example:/config/vlan-1/#>
```

`[no] multicast-snooping`

Description

Enable, or disable IGMP/MLD snooping

IGMP/MLD Querier Settings

As part of the IGMP/MLD snooping functionality, the switch can also act as Querier - a role usually handled by *multicast routers*. Switches with querier capabilities allow for distribution of IP multicast in networks without multicast routers, using IGMP/MLD to limit the broadcast effects of multicast.

IGMP/MLD Querier settings are located in the IP configuration context.

`[no] multicast-querier <auto|proxy>`

Description

Auto mode is the default querier mode. It implements the IGMP/MLD standard to elect[1] a designated IGMP/MLD querier on each LAN. In this mode all multicast, both known and unknown, is flooded to the elected querier, which acts as a distribution point for the LAN. This is an important aspect when designing networks and calculating the required bandwidth of individual links.

continues on next page

[1] The querier with the lowest IP address on each LAN is elected. Usually the gateway or multicast router.

continued

`[no] multicast-querier <auto|proxy>`

Proxy mode is the alternative querier mode. Useful in networks where the switch should never take part in the querier election and only act as a silent forwarder of IGMP/MLD queries (and reports). However, to prevent loss of multicast distribution on LANs where there is no elected IGMP/MLD querier, the switch by default send *proxy queries*[2].

On VLANs where the network interface is not assigned an IP address, the switch will automatically fall back to *proxy mode* on that VLAN, regardless of the global querier mode setting.

The proxy query feature can confuse IGMP implementations from some vendors. This may result in other switches forwarding **all** multicast towards the switch originating the proxy query, severely impacting performance by saturating links. Therefore, this feature can be disabled, leaving the switch to only forward any IGMP reports (join/leave messages) and IGMP queries, acting as a pure proxy.

`[no] multicast-proxy-query`

Description

Allow sending IGMP query with source IP 0.0.0.0 in proxy mode. Default: Enabled

When IGMP/MLD snooping is enabled and the multicast-querier mode is set to 'proxy', or the VLAN interface has no address, and no elected IGMP/MLD querier on the LAN exists, then the device initiates IGMP proxy queries using source IP address 0.0.0.0.

In a network with no multicast router this feature speeds up network convergence for multicast. Disabling it means the device will never initiate an IGMP query.

`[no] multicast-querier-interval <5-300>`

Description

Query interval (QI). Default: 125 seconds

When elected as IGMP/MLD querier, the switch can emit queries in the range 5-300 sec. This interval is also used when timing out multicast to end devices that for some reason stop answering the queries.

`[no] multicast-querier-robustness <VALUE>`

Description

Control IGMP/MLD querier robustness value, QRV. Default: 2

This setting controls the querier/query robustness value. It in turn controls how multicast routers/queriers and group memberships time out. It is intended to be used sparingly but can be useful in lossy networks to allow for losing query or membership reports. When the QRV is set to a value of 2 (default) max 1 (QRV - 1) query may be lost by an end device, or one

continues on next page

[2] Proxy queries use source IP address 0.0.0.0, which is reserved and must never take part in the IGMP querier election process, as clearly stated in RFC 4541[3].

[3] https://tools.ietf.org/html/rfc4541

continued

`[no] multicast-querier-robustness <VALUE>`

membership report lost by the querier before they are considered timed out. A certain jitter is allowed, see for instance the description for the multicast-router-timeout. The membership timeout is calculated according to the following formula:

timeout = QRV * QI + QRI

Where QI is the querier interval and QRI is the hard coded Querier Response Interval (10 sec).

`[no] multicast-router-timeout <0|1-2147483647>`

Description

Timeout for learned multicast routers. Default: Auto (0)

A functional IGMP/MLD querier, usually a multicast router, on a LAN is critical to successful multicast operation. If the querier dies, or suddenly gets a new IP address, another device must take over. This setting controls the time in seconds before this device can take over.

By default this value is automatically calculated using the currently configured query interval (QI), the robustness value (QRV) and the Query Response Interval (QRI) according to the following formula:

timeout = QRV * QI + QRI/2

Where the Querier Response Interval (QRI) is hard coded to 10 sec.

When a multicast receiver attached to a switch port leaves a multicast group (i.e., stops subscribing to an IP multicast address or is simply disconnected from the port), the IGMP/MLD snooping leave latency (the time until the switch stops forwarding the associated multicast data) is within 2-3 times the configured *Query Interval*. See IGMP Fast Leave for information on optimising this behaviour on *access ports*.

Multicast Router Ports

With IGMP/MLD snooping enabled, the switch learns on which ports there are interested receivers of multicast. It accomplishes this by listening to IGMP report messages sent by all subscribers. Thus, the switch only forwards IP multicast on ports that are members of multicast groups.

The switch also forwards *all* multicast traffic, both subscribed (known) and *unknown*, on ports leading to multicast routers. The following ports are considered as *multicast router ports*:

- Ports configured as multicast router ports
- Ports where IGMP/MLD Queries are received, usually queries are sent by multicast routers, but also by IGMP/MLD snooping aware switches like WeOS.
- Ports on which Multicast Router Discovery (MRDISC) packets are received, see RFC 4286[1]
- RiCo uplink ports : To provide fast fail-over of multicast traffic, *FRNT Ring Coupling* (RiCo) uplinks are added to the list of multicast router ports. This is both done at the Ring Coupling nodes, as well as on switches on the remote side of the uplink.

Ports can be explicitly configured as multicast router ports in the IP configuration context:

[1] https://tools.ietf.org/html/rfc4286

```
example:/config/#> ip
example:/config/ip/#> multicast-router-ports eth7
example:/config/ip/#>
```

`[no] multicast-router-ports <PORTS>`

Description

Set static multicast router ports. Default: Disabled

FRNT ring ports are not considered multicast router ports *per se*. The *Fast Reconnect* feature of FRNT is instead handled per multicast group: if a multicast address is learnt on a ring port, the other ring port is automatically added. Similarly, if an IGMP/MLD Query is received on a ring port, both ring ports are considered *multicast router ports*. WeOS applies this multicast *Fast Reconnect* feature for MRP too. In case of ring breakage this practice ensures next to zero reconfiguration time for multicast over FRNT and MRP.

IGMP/MLD Fast Leave

IGMP/MLD snooping supports *IGMP/MLD Leave* by default and *IGMP/MLD Fast Leave* can be enabled on a per-port basis. The CLI `multicast-fast-leave` setting allows the keyword `all`. The Fast Leave feature is recommended only for access ports.

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> no multicast-fast-leave
example:/config/ip/#> multicast-fast-leave eth3, eth6
example:/config/ip/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
example:/#> show ip igmp
 Static Multicast ports
 ------------------------------------------------------------------------------
 IGMP Fast Leave ports      : eth3, eth6
 Static router ports        : eth5
 Discovered router ports    : ---
 Dual Homing/Coupling ports : ---
 FRNT ring ports            : ---
 Multicast flooded on ports : eth1, eth2, eth3, eth4, eth5, eth6, eth7, eth8

 VID  Querier IP       Querier MAC        Port              Interval  Timeout
 ------------------------------------------------------------------------------
   1  10.0.0.1         LOCAL

 VID  Multicast Group  Filtered MAC Addr  Active ports
 ------------------------------------------------------------------------------
   1  225.1.2.3        01:00:5E:01:02:03  eth5

 ------------------------------------------------------------------------------
Total: 1 filters, max 2048, in 1 VLANs.
```

When an IGMP Leave is received on a port configured with Fast Leave it issues a group specific query for the group being left and then immediately cuts the multicast stream for that group. With Fast Leave *disabled*, a standard grace period is honored for the benefit of any multicast receivers attached on downstream port splitters (hubs or unmanaged switches). When no membership report/reply is received the multicast group will time-out within three query intervals.

The setting to control multicast fast leave is available in the *IP configuration* context:

`[no] multicast-fast-leave <PORTS>`

Description

IGMP/MLD Fast Leave (access) ports. Default: None

Multicast Snooping Boundary

The device supports the ability to specify specific ports as a boundary for IGMP/MLD snooping, referred to as a *multicast snooping boundary*. This is a setting that is applied on a per-port basis and can be used to limit the scope of IGMP/MLD snooping to a specific set of ports.

When a port is specified to act as a multicast snooping boundary, it will drop any IGMP/MLD control traffic on ingress, egress, or both, controlled by the `multicast-snooping-boundary-direction` setting. The control traffic that will be dropped includes IGMP/MLD queries, reports, and leave messages.

A few examples where the multicast snooping boundary can be useful:

- Define multiple IGMP/MLD snooping domains in the same layer 2 network domain, i.e., to have multiple queriers in the same layer 2 network.
- Isolate sections of the network where IGMP/MLD snooping is not desired.

> **NOTE**
>
> The multicast snooping boundary setting only affects IGMP/MLD control traffic. Regular multicast traffic can still pass through the port.

Configuration

The configuration of the multicast snooping boundary is done in the IP configuration context:

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#>
```

`[no] multicast-snooping-boundary [PORTS]`

Description

Specify port(s) to act as a boundary for IGMP/MLD signalling.

Using this option it is possible to define IGMP/MLD boundaries in the same layer 2 domain. This can allow for multiple queriers to exist at the same time, or to limit the scope of IGMP/MLD snooping to a specific set of ports.

When a port is defined as a boundary, it will by default drop IGMP/MLD control traffic on both ingress and egress. This means that the port will not forward IGMP/MLD queries, reports or leave messages.

If for any reason IGMP/MLD control traffic should only be dropped on either ingress or egress, this can be specified using the `multicast-snooping-boundary-direction` option.

continues on next page

continued

`[no] multicast-snooping-boundary [PORTS]`

> **NOTE**
>
> When a port is defined as a boundary, it is still possible for regular multicast traffic to pass through the port. The boundary setting only affects IGMP/MLD control traffic.

Default: Disabled

Example

Enable the multicast snooping boundary for a single port:

```
example:/config/ip/#> multicast-snooping-boundary eth1
```

Enable the multicast snooping boundary for a range of ports:

```
example:/config/ip/#> multicast-snooping-boundary eth1, eth2, eth3
```

Enable the multicast snooping boundary for a range of ports:

```
example:/config/ip/#> multicast-snooping-boundary eth1..eth3
```

Enable the multicast snooping boundary for all ports:

```
example:/config/ip/#> multicast-snooping-boundary all
```

All the above examples can be prefixed with the `no` keyword to disable the multicast snooping boundary for the specified ports in the same way.

Options

Option	Description
`no`	Disable the multicast snooping boundary for all ports or any specific port(s).
`PORTS`	Specific physical port or range of ports. The port(s) can be specified as a single port, a comma separated list of ports, or a range of ports using a `..`. It is also possible to specify the `all` keyword to apply the setting to all ports.

`[no] multicast-snooping-boundary-direction [ingress][egress]`

Description

Specify the direction of the boundary for IGMP/MLD signalling.

This option will specify which directions, either ingress, egress or both, that the boundary will drop IGMP/MLD control traffic.

continues on next page

continued

`[no] multicast-snooping-boundary-direction [ingress][egress]`

> **NOTE**
>
> The `multicast-snooping-boundary-direction` setting is only applicable when the `multicast-snooping-boundary` setting is enabled for at least one port.

Default: Both Ingress and Egress

Example

Specify the boundary to drop IGMP/MLD control traffic on ingress only:

```
example:/config/ip/#> multicast-snooping-boundary-direction ingress
```

Specify the boundary to drop IGMP/MLD control traffic on egress only:

```
example:/config/ip/#> multicast-snooping-boundary-direction egress
```

Specify the boundary to drop IGMP/MLD control traffic on both ingress and egress:

```
example:/config/ip/#> multicast-snooping-boundary-direction ingress egress
```

The `no` keyword can be prefixed to the command to disable the setting.

Options

Option	Description
`no`	Disable the multicast snooping boundary direction setting for the specified direction. **NOTE** At least one direction must always be enabled.
`ingress`	Drop IGMP/MLD control traffic on ingress, i.e., traffic entering the switch.
`egress`	Drop IGMP/MLD control traffic on egress, i.e., traffic leaving the switch.

Example: Basic scenario

The following example just intend to illustrate the basic usage of the multicast snooping boundary. Consider the simplified network topology in **Figure 5** below:

```
                                IGMP Domain 1 : IGMP Domain 2
  .--.-.                                      :                          .--.-.
 ( (    )__                                   :                         ( (    )__
(_,  \ ) ,_)                                   :                        (_,  \ ) ,_)
  '-'--`--'                                    :                          '-'--`--'
     |                                         :                                |
     | 10.0.0.0/24                             :                   10.0.0.0/24 |
     |                                         :                                |
.----+----.           .---------.              :     .---------.           .----+----.
|         |       eth2|         |eth1 : eth1|         |eth2       |         |
|   S1    +-----------+   S2    +-----------+   S3    +-----------+   S4    |
|         |           |         |              :     |         |           |         |
'---------'           '---------'              :     '---------'           '---------'
 10.0.0.10             10.0.0.11               :      10.0.0.21             10.0.0.20
                                               :
                                               :
```

Figure 5: Basic network topology, depicting four switches (S1-S4) connected together to a larger layer 2 network. Where a multicast snooping boundary is configured on ports eth1 on switches S2 and S3. In this scenario it results in two separate IGMP domains, IGMP Domain 1 and IGMP Domain 2, located on the same layer 2 network.

Given the above example, we assume that everything is already configured and working as expected, except for the multicast snooping boundary. Before the multicast snooping boundary is configured, the switch **S1** should be the elected IGMP querier for the entire layer 2 network, since it has the lowest IP address. In this case, if we check the reported querier on switch **S3**, for example, it should be reporting **S1** (10.0.0.10) as the IGMP querier:

```
S3:/#> show ip igmp
S3:/#> show ip igmp
Multicast Overview
Query Interval           : 12 sec
Robustness Value         : 2
Router Timeout           : 29
Fast Leave Ports         :
Router Ports             : eth1
Flood Ports              : eth0, eth1, eth2, eth3, eth4, eth5, eth6, eth7

Interface          State      Querier                Timeout  Ver
vlan1              Up         10.0.0.10                   23    3

 VID  Multicast MAC          Multicast Group        Ports
   1  01:00:5E:7F:FF:FA      239.255.255.250        br0, eth1, eth2
```

In order to create two separate IGMP domains, IGMP Domain 1 and IGMP Domain 2, we will configure both switch **S2** and **S3** to have a multicast snooping boundary enabled on port eth1:

> **NOTE**
>
> It could be enough to configure the boundary on only one of the switches, but since we control both switches in this example, we will configure both of them.

```
S2:/#> configure
S2:/config/#> ip
S2:/config/ip/#> multicast-snooping-boundary eth1
S2:/config/ip/#> leave
S2:/#>
```

```
S3:/#> configure
S3:/config/#> ip
S3:/config/ip/#> multicast-snooping-boundary eth1
S3:/config/ip/#> leave
S3:/#>
```

If we now check the reported querier on switch **S3**, it should no longer report **S1** (10.0.0.10) as the IGMP querier, but it should now report switch **S4** (10.0.0.20) as the IGMP querier:

> **NOTE**
>
> It may be necessary to wait up to the entire multicast router timeout period before the querier election process is completed.

```
S3:/#> show ip igmp
S3:/#> show ip igmp
Multicast Overview
Query Interval          : 12 sec
Robustness Value        : 2
Router Timeout          : 29
Fast Leave Ports        :
Router Ports            : eth2
Flood Ports             : eth0, eth1, eth2, eth3, eth4, eth5, eth6, eth7

Interface         State     Querier               Timeout  Ver
vlan1             Up        10.0.0.20                  23    3

 VID  Multicast MAC         Multicast Group       Ports
   1  01:00:5E:7F:FF:FA     239.255.255.250       br0, eth1, eth2
```

However, if we check the reported querier on switch **S2**, it should still report **S1** (10.0.0.10) as the IGMP querier:

```
S2:/#> show ip igmp
S2:/#> show ip igmp
Multicast Overview
Query Interval          : 12 sec
Robustness Value        : 2
Router Timeout          : 29
Fast Leave Ports        :
Router Ports            : eth2
Flood Ports             : eth0, eth1, eth2, eth3, eth4, eth5, eth6, eth7

Interface         State     Querier               Timeout  Ver
vlan1             Up        10.0.0.10                  18    3

 VID  Multicast MAC         Multicast Group       Ports
   1  01:00:5E:7F:FF:FA     239.255.255.250       br0, eth2, eth1
```

Effectively, we have now created two separate IGMP domains: IGMP Domain 1 and IGMP Domain 2, located on the same layer 2 network. All the switches to the “left” of and including **S2** will belong to IGMP Domain 1, while all the switches to the “right” of and including **S3** will belong to IGMP Domain 2. Where switch **S1** is the IGMP querier for IGMP Domain 1, and switch **S4** is the IGMP querier for IGMP Domain 2.

> **NOTE**
>
> This example used IGMP snooping to illustrate the concept of a multicast snooping boundary. However, the same principles apply to MLD snooping and MLD snooping boundaries. It is the same setting.

In addition to query frames being dropped on the boundary port, the boundary port will also not forward any IGMP/MLD reports or leave messages. So, if we have a specific group that we want to be able to traverse the boundary port, static FDB entries could be used.

> **NOTE**
>
> The specific group we want to be able to traverse the boundary must also reach that device in that domain. Depending on the placement of the querier, static FDB entries may be needed on several devices in order to reach the boundary device.
>
> Alternatively, multicast router ports could be configured in the direction leading to the boundary device. However, this could result in more multicast traffic being forwarded to the boundary device than intended.

Reserved Multicast Groups

IP multicast in the Local Network Control Block, 224.0.0.X, is reserved for protocols like RIP (224.0.0.9), OSPF (224.0.0.5 and 224.0.0.6), and VRRP (224.0.0.18). Traffic in this range should not be filtered by IGMP Snooping switches RFC 4541[1]. Furthermore, WeOS units filter multicast on the MAC level, and since multiple IP multicast addresses map to the same MAC address, see RFC 1112[2], this means there is a larger set of “well-known” reserved groups in consecutive ranges that IGMP does not filter. Any packets received for these groups are flooded on all ports in the same VLAN on which they were received, except the port on which the packets entered the switch.

The table below lists the IP multicast address groups excluded from IGMP filtering, i.e., the 32 well-known address groups (8192 total addresses) that IGMP does not filter.

Reserved MC			
224.0.0.x	232.0.0.x	224.128.0.x	232.128.0.x
225.0.0.x	233.0.0.x	225.128.0.x	233.128.0.x
226.0.0.x	234.0.0.x	226.128.0.x	234.128.0.x
227.0.0.x	235.0.0.x	227.128.0.x	235.128.0.x
228.0.0.x	236.0.0.x	228.128.0.x	236.128.0.x
229.0.0.x	237.0.0.x	229.128.0.x	237.128.0.x
230.0.0.x	238.0.0.x	230.128.0.x	238.128.0.x
231.0.0.x	239.0.0.x	231.128.0.x	239.128.0.x

Table 1: Reserved RFC 1112[3] groups. X is any value from 0 to 255.

[1] https://tools.ietf.org/html/rfc4541
[2] https://tools.ietf.org/html/rfc1112
[3] https://tools.ietf.org/html/rfc1112

2.4.5. Port Access Control

Table of contents

- Introduction
- 802.1X
  - Specifying Authentication Server
  - Authenticator Initiation
  - Re-authentication Interval
- MAC Authentication
- 802.1X and MAC Authentication co-existence
- Port Access and Trunk ports
- CLI Command Syntax
  - Port Access Control Context
  - 802.1X Configuration Context
  - MAC Auth Configuration Context
  - Port Access Control Status

Related HowTo's:
- 802.1X Basic Setup HowTo
- MAC-Auth using RADIUS (MAB)
- MAC-Auth using accept list

Introduction

Port access control is used to stop unauthorised hosts from accessing the network. Authentication is required to gain access.

Port access provides two authentication mechanisms for hosts that wish to access a network. These two are:
- IEEE 802.1X authentication: Only clients (supplicants) able to prove the identity using 802.1X cryptographic handshake are allowed to access the network.
- MAC authentication: Only clients with granted MAC addresses are allowed to access the network.

```
                                |
Authentication          .--.-.
   Server             ( (    )__
  .------.           (_,  \ ) ,_)
  |RADIUS|             '-'--`--'
  '--+---'                |Internet/
     |                    |Intranet
  .--+------+------+------+--.
  |  5      6      7      8  |  Authenticator
  |           WeOS           | (Switch/Router)
  |  1      2      3      4  |
  '--+------+------+------+--'
     |      |      X      X  <== Controlled Ports
     |      |      |
   .-+--. .-+--. .-+--.
   | H1 | | H2 | | H3 |       Hosts/Supplicants
   '----' '----' '----'
Authorised Hosts | Blocked Ports
```

Figure 1: WeOS Switch/Router with ports eth1-eth4 subject to access control.

Switch ports configured for port access control will by default be blocked for incoming traffic. Only when a connected device has successfully authenticated itself, it will be allowed to send data

through the port. Packets from unauthorised devices are still dropped, i.e., only packets with a source MAC address of devices authorised via 802.1X or MAC authentication are allowed. Unknown multicast and broadcast will be blocked from egress on a controlled port until there is at least one MAC address authenticated on this port. For unknown multicast to egress a port at all, the 'Multicast Flood Unknown' setting must be enabled on this port.

> **NOTE**
>
> The behavior described above applies to the default **MAC Based** port method. See Port Method for information about alternative port methods.

Port access configuration is available from the general configuration context. The example below shows how to define the set of ports subject to access control.

```
example:/#> config
example:/config/#> port-access
example:/config/port-access/#> port eth1..eth4
example:/config/port-access/#> leave
example:/#>
```

You also need to specify which control mechanism to use (802.1X or MAC-Auth, or both), and which RADIUS server or local MAC list (Only for MAC-Auth) to use for authorisation. This is described in the sections below.

> **SEE ALSO**
>
> Related Examples and Use-Cases
> - 802.1X Basic Setup HowTo
> - MAC-Auth using RADIUS (MAB)
> - MAC-Auth using accept list
> - Dynamic VLAN Assignment with 802.1X
> - Dynamic VLAN Assignment with MAC Authentication Bypass (MAB)

IEEE 802.1X

WeOS units are able to act as IEEE 802.1X authenticators. WeOS uses the RADIUS protocol with extensions for Extensible Authentication Protocol, EAP (RFC 2869[1]), to communicate to a back-end authentication server. WeOS neither includes a RADIUS server nor a local authentication server mechanism for 802.1X. Instead the 802.1X authentication server must be provided externally as depicted in Figure 1.

Assuming you have already setup a RADIUS server instance, configuring port access control with 802.1X can be done as follows (the 802.1X Basic Setup HowTo provides a more extensive example).

```
example:/#> config
example:/config/#> port-access
example:/config/port-access/#> port eth1..eth4
example:/config/port-access/#> dot1x
example:/config/port-access/dot1x/#> authentication-server 1
```

[1] https://tools.ietf.org/html/rfc2869

```
example:/config/port-access/dot1x/#> leave
example:/#>
```

Figure 2 and Figure 3 illustrate the principles of a successful authentication with IEEE 802.1X. In reality the protocol exchanges several messages between the supplicant, the authenticator and the RADIUS back-end server (see the standard documents for details). The WeOS unit acts as an IEEE 802.1X authenticator, relaying the EAP messages to the RADIUS server.

```
                 Authentication            |
                    Server              .--.-.
                  .------.            ( (    )__
                  |RADIUS|           (_,  \ ) ,_)
                  '--+---'             '-'--`--'
RADIUS             ^ |                    |Internet/
EAP Request        | |                    |Intranet
                  .--+------+------+------+--.
                  |  5      6      7      8  |  Authenticator
                  |           WeOS           | (Switch/Router)
                  |  1      2      3      4  |
                  '--+------+------+------+--'
                     X      X      X      X  <== Controlled Ports
           EAPoL    ^ |                           (All Blocked by
           Request  | |                            Authenticator)
                  .--+-.
                  | H1 |
                  '----'
```

Figure 2: H1 (Supplicant) initiates EAP authentication handshake (EAPoL Request). Authenticator relays EAP message onto back-end RADIUS server.

```
                 Authentication            |
                    Server              .--.-.
                  .------.            ( (    )__
                  |RADIUS|           (_,  \ ) ,_)
                  '--+---'             '-'--`--'
RADIUS             | |                    |Internet/
EAP Success        V |                    |Intranet
                  .--+------+------+------+--.
                  |  5      6      7      8  |  Authenticator
                  |           WeOS           | (Switch/Router)
                  |  1      2      3      4  |
                  '--+------+------+------+--'
                   | |      X      X      X  <== Controlled Ports
           EAPoL   | |                           (Port 1 unblocked
           Reply   V |                            by Authenticator)
                  .--+-.
                  | H1 |
                  '----'
```

Figure 3: At the end of a successful EAP handshake, the RADIUS server sends an EAP Success message. The Authenticator unblocks the port.

> **NOTE**
>
> When using the **MAC Based** port method (default), a WeOS unit acting as Authenticator will only *unlock* the port for the authorised MAC address. Other MAC addresses are still blocked on that port. This mode is stricter than defined in the IEEE 802.1X[1] standard. With the **Port Based** method, the entire port is unlocked after successful authentication.

Specifying Authentication Server

WeOS requires an external RADIUS server to authenticate the 802.1X supplicant. It is possible to specify two servers for 802.1x where the secondary specified server will be used if the first fail to answer the radius authentication request.

> **NOTE**
>
> For best performance, it is recommended to use IP address (rather than domain name) when defining the RADIUS server.

```
example:/#> config
example:/config/#> aaa
example:/config/aaa/#> remote-server 1
Creating new remote server 1
example:/config/aaa/remote-server-1/#> address 192.168.2.1
example:/config/aaa/remote-server-1/#> password MyS3cret
example:/config/aaa/remote-server-1/#> end
example:/config/aaa/#> remote-server 2
example:/config/aaa/remote-server-2/#> address 192.168.3.1
example:/config/aaa/remote-server-2/#> password MyS3cret
example:/config/aaa/remote-server-2/#> end
example:/config/aaa/#> end
example:/config/#> port-access
example:/config/port-access/#> dot1x
example:/config/port-access/dot1x/#> authentication-server 1,2
example:/config/port-access/dot1x/#> leave
example:/#>
```

See AAA configuration for more information on how to define RADIUS servers in WeOS.

Authenticator Initiation

WeOS supports Authenticator initiated authentication as defined by §8.1.2.1 in IEEE 802.1X[2]. This is enabled by default with interval 30 seconds.

When enabled, authenticator actively sends EAP Identity Requests every 30 seconds by default until a supplicant successfully authenticates. The interval is configurable in range 10-300 seconds). The example below shows how to set the interval to 60 seconds.

```
example:/#> config
example:/config/port-access/#> dot1x
example:/config/port-access/dot1x/#> active-authentication 60
```

[1] IEEE 802.1X-2004: Port-Based Network Access Control, IEEE Standard for Local and metropolitan area networks
[2] IEEE 802.1X-2004: Port-Based Network Access Control, IEEE Standard for Local and metropolitan area networks

```
example:/config/port-access/dot1x/#> leave
example:/#>
```

`no active-authentication` disables authenticator initiated authentication. If disabled the 802.1X client (supplicant) must be capable to initiate the authentication procedure to gain access.

Re-authentication Interval

The authenticator can periodically request the supplicant to re-authenticate itself. By default, period re-authentication is disabled. If enabled, the default request interval is 3600 seconds, see the example below. The interval can be configured in range 10 to 352800 seconds. Note that a value lower than 600 is not recommended and only possible to configure via cli.

```
example:/#> config
example:/config/port-access/#> dot1x
example:/config/port-access/dot1x/#> re-authenticate
example:/config/port-access/dot1x/#> show re-authenticate
3600
example:/config/port-access/dot1x/#> leave
example:/#>
```

MAC Authentication

MAC based authentication is not as secure as IEEE 802.1X. Devices are granted access based on the MAC address without any cryptographic authentication exchange, and it is fairly easy to modify the MAC address on a PC and most other equipment.

MAC authentication is useful for devices that do not support 802.1X, such as printers or legacy equipment. These devices cannot act as 802.1X supplicants, so MAC authentication provides a way for them to gain access to the network based on their MAC address alone.

To define which MAC address(es) to grant, it is possible to configure a local *accept list* with MAC addresses to grant access. It is also possible to let the WeOS use a backend RADIUS to control what MAC addresses to accept - the latter method is also referred to as MAC authentication bypass (MAB).

```
                 Authentication            |
                    Server              .--.-.
                  .------.            ( (    )__
                  |RADIUS|           (_,  \ ) ,_)
                  '--+---'             '-'--`--'
RADIUS             ^ |                    |Internet/
Calling-ID         | |                    |Intranet
(MAC-H1)          .--+------+------+------+--.
                  |  5      6      7      8  |  Authenticator
                  |           WeOS           | (Switch/Router)
                  |  1      2      3      4  |
                  '--+------+------+------+--'
                     X      X      X      X  <== Controlled Ports
                   ^ |                           (All Blocked by
 Data Pkt          | |                            Authenticator)
 Src MAC          .--+-.
(MAC-H1)          | H1 |
                  '----'
```

Figure 4: H1 (Host/Supplicant) connects to port controlled by MAC Authentication. Switch looks in local 'Accept-list' or (as here) forwards H1's MAC address to a back-end RADIUS server.

```
                    Authentication             |
                        Server              .--.-.
                       .------.            ( (    )__
                       |RADIUS|           (_,  \ ) ,_)
                       '--+---'             '-'--`--'
            RADIUS      | |                    |Internet/
            Accept      V |                    |Intranet
                       .--+------+------+------+--.
                       |  5      6      7      8  |  Authenticator
                       |           WeOS           | (Switch/Router)
                       |  1      2      3      4  |
                       '--+------+------+------+--'
                          |      X      X      X
                          |
                          |
                       .--+-.
                       | H1 |
                       '----'
```

Figure 5: After successful authentication, the RADIUS server sends an Accept message back to the Authenticator. The Authenticator unblocks the port for the MAC address of H1.

- When defining a local accept list, the list can consist of one or more MAC address patterns. MAC patterns may contain a wild-card at the end to match a whole range of addresses. Examples: The pattern 00:11:22:33:44:55 matches exactly one address, while the pattern 00:aa:bb:* matches all addresses beginning with 00:aa:bb.
- When using a RADIUS server (MAB), it is currently only possible to specify a single server. WeOS passes the MAC address of the connecting host in a RADIUS *Calling-ID* attribute.
- It is possible to use both accept list and RADIUS server.

```
example:/#> config
example:/config/#> aaa
example:/config/aaa/#> mac-auth 1
example:/config/aaa/mac-auth-1/#> mac match 00:11:22:33:44:55
example:/config/aaa/mac-auth-1/#> mac match 00:aa:7c:*
example:/config/aaa/mac-auth-1/#> end
example:/config/aaa/#> end
example:/config/#> port-access
example:/config/port-access/#> mac-authentication
example:/config/port-access/mac-auth/#> accept-list 1
example:/config/port-access/mac-auth/#> leave
example:/#>
```

See AAA configuration for more information on how to define RADIUS servers in WeOS.

MAC-Auth using RADIUS (MAB) and MAC-Auth using accept list HowTos provide more extensive examples.

802.1X and MAC Authentication co-existence

It is possible to enable both 802.1X and MAC Authentication for a single port access control instance. The reason for using both methods could be if you have a mix of 802.1X capable devices and legacy (non-802.X capable) devices in your network.

If both authentication methods are enabled, the first one that does a successful authentication will take precedence. If a successful 802.1X authentication attempt has been made, subsequent MAC Auth attempts are cancelled, and vice versa.

Port Access and Trunk ports

If a port is configured to belong to several VLANs, a successful authentication triggered on any of the VLANs will result in access being granted on all the VLANs the port is a member of (for MAC Based method, the specific MAC address is unlocked; for Port Based method, the entire port is unlocked). A constraint for 802.1X is that the authentication must be performed on an untagged VLAN - this may change in future releases.

> **WHEN USING VLAN ASSIGNMENT**
>
> When using VLAN assignment it will not be possible to have any of the port access ports be configured tagged in any VLANs, this will result in a configuration error.

Port Method

Each port access control instance can select one of two different Port Methods that will determine how the selected ports will behave in terms of providing access to the network. The two methods are:

- **MAC Based (Default):** Each individual client must be authenticated, where its MAC address is unlocked on the port.
- **Port Based:** The entire port is unlocked for all traffic when a single client has been successfully authenticated on a given port. When the port is unlocked, all traffic will be allowed to access the port, essentially the port will now behave like a regular port without access control. The unlocked state is “controlled” by the client that was authenticated, meaning that if that client is deauthenticated the port will be locked again.

```
                Authentication            |
                   Server              .--.-.
                .------.            ( (    )__
                |RADIUS|           (_,  \ ) ,_)
                '--+---'             '-'--`--'
RADIUS           ^ |                    |Internet/
Calling-ID       | |                    |Intranet
(MAC-H1)        .--+------+------+------+--.
                |  5      6      7      8  |  Authenticator
                |           WeOS           | (Switch/Router)
                |  1      2      3      4  |
                '--+------+------+------+--'
                   X      X       X      X
                          |
                          |
                     .-----+-----.
            .----.   |           |   .----.
            | H1 +---+  Switch   +---+ H4 |
            '----'   |           |   '----'
                     '--+-----+--'
                        |     |
                     .--+-. .-+--.
                     | H2 | | H3 |
                     '----' '----'
```

Figure 6: With Port Based method, the entire port is unlocked for all traffic when a single client has been successfully authenticated on a given port. In this example, when H1 successfully authenticates, the entire port is unlocked and thus H2, H3 and H4 will also have access to the network. With MAC Based method, only H1 would have access to the network after successful

authentication, and H2, H3 and H4 would still be blocked until they have successfully authenticated themselves.

This setting may be more important to consider if what is behind the port access port is not a single end device but rather a switch or an access point which could have multiple clients behind it, as in the example above. If only one end device will be behind a given port access port, both methods will, for all practical purposes, behave the same.

BE AWARE OF THE NETWORK TOPOLOGY WHEN USING PORT BASED METHOD

Just to be clear, be mindful when selecting the Port Based method, so that you will not inadvertently provide access for anyone, as mentioned above, this is especially important if multiple clients could be behind the port access port.

BEFORE WEOS 5.28

Support for configuring a specific Port Method did not exist before WeOS 5.28. Before that the implicit port method was what is now referred to at **MAC Based** and the concept of **Port Based** was not supported.

VLAN Assignment

Dynamic VLAN assignment allows the network administrator to centrally manage which VLAN a port is placed in upon successful authentication, using RADIUS policies instead of static per-port VLAN configuration on the switch. This is useful for segmenting traffic based on role, department, or security level.

Dynamic VLAN assignment is supported for both 802.1X and MAC authentication (MAB). When enabled, the switch can receive VLAN assignment information from the RADIUS server and dynamically assign the port to the specified VLAN upon successful authentication.

The following is an overview of the constraints and behaviors that apply when using dynamic VLAN assignment:

- Ports can only be dynamically assigned **untagged** to a VLAN.
- Ports can only be assigned to VLANs that:
  - Already exist on the device.
  - Are configured as allowed VLANs in the port access configuration.
  - Have an associated VLAN interface created for it on the device.
- Port access ports cannot be configured as tagged members of any VLAN when VLAN assignment is enabled for the instance.
- Authentication will fail if the RADIUS server specifies a VLAN ID that is not in the configured allowed VLANs list.
- VLAN information from the RADIUS server must be provided in the `Tunnel-Private-Group-ID` attribute as a VLAN ID (VID) only. VLAN names are not supported.
- Multiple clients on the same port can only be authenticated if they all request assignment to the same VLAN. The first authenticated client will determine the VLAN assignment for the port, and subsequent clients must request the same VLAN assignment to be able to authenticate successfully.
- Dynamic VLAN assignment is **not pre-emptive**. If a port is already assigned to a VLAN, it cannot be reassigned to another VLAN until the last client that is authenticated on that port is

deauthenticated, and the port is moved back to its statically configured untagged VLAN, i.e. the starting VLAN.
- When using **MAC Based** port method: Multiple clients on the same port can only be authenticated if they all request assignment to the same VLAN.
- When VLAN assignment is disabled, ports always remain in their statically configured untagged VLAN, regardless of RADIUS server responses.

> **SEE ALSO**
>
> Basic examples and use-cases related to VLAN assignment
> - Dynamic VLAN Assignment with 802.1X HowTo
> - Dynamic VLAN Assignment with MAC Authentication Bypass (MAB) HowTo

Assignable VLANs

When dynamic VLAN assignment is enabled, VLANs are not automatically created. Only VLANs that already exist on the device can be assigned to ports. This provides better control and security over which VLANs authenticated clients can access.

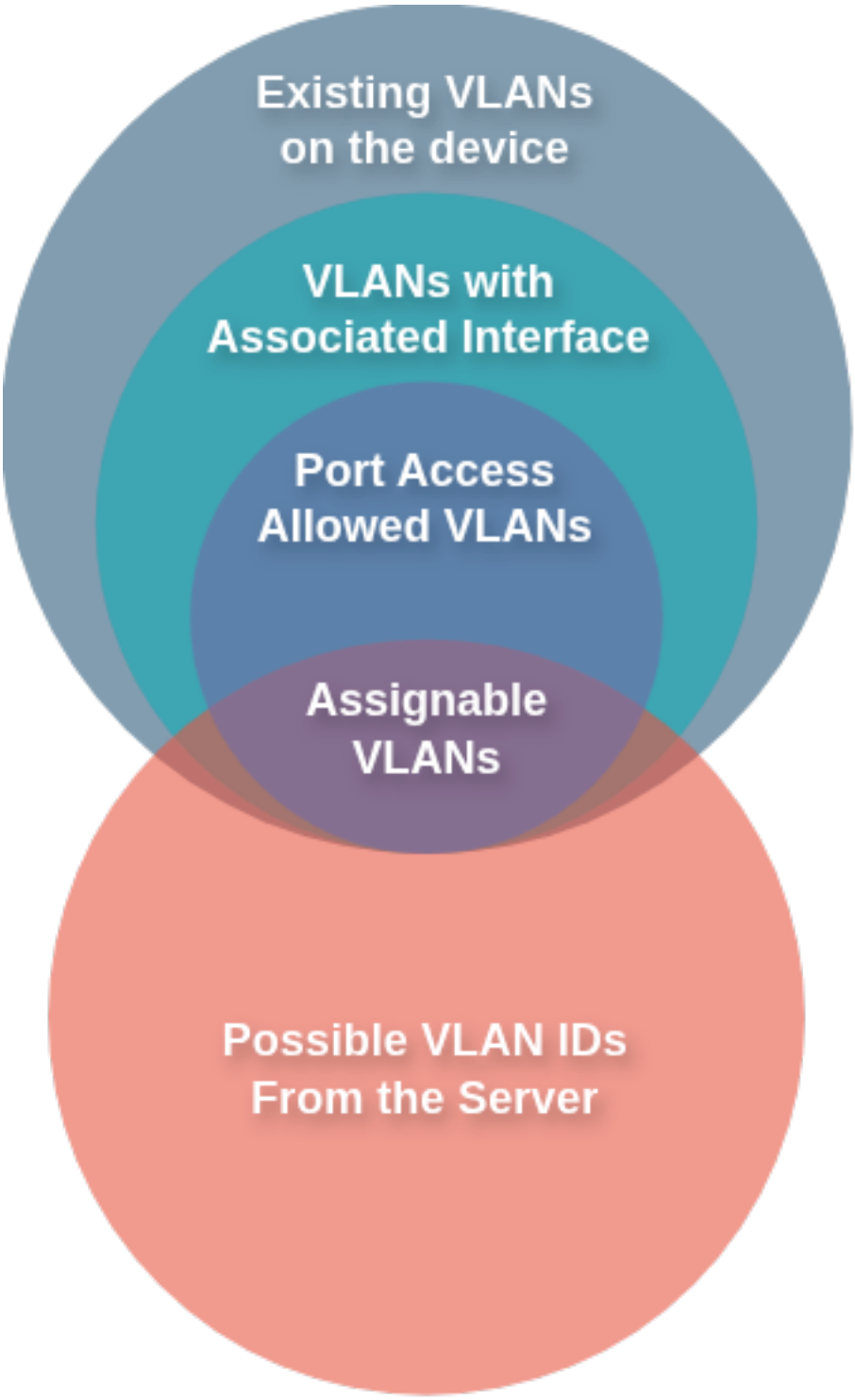


Figure 7: Illustration showing the subset of VLANs that are assignable through port access.

For a VLAN to be assignable through port access, it must meet all of the following requirements, as illustrated in Figure 7:

- The VLAN must already exist on the device
- The VLAN must have an associated VLAN interface created on the device
- The VLAN must be configured as an allowed VLAN in the port access configuration

For example, to make VLANs 10, 20, and 30 available for dynamic assignment, first create the VLANs:

```
example:/#> config
example:/config/#> vlan 10
example:/config/vlan-10/#> end
example:/config/#> vlan 20
example:/config/vlan-20/#> end
example:/config/#> vlan 30
example:/config/vlan-30/#> end
example:/config/#>
```

Then configure these VLANs as allowed in the port access VLAN assignment configuration. If the RADIUS server specifies a VLAN ID that is not in the allowed VLANs list, authentication will fail.

> **VLAN INTERFACES**
>
> Only VLANs that have an associated VLAN interface can be used for dynamic assignment. When a VLAN is created, a VLAN interface is automatically created for it, it currently cannot be controlled explicitly.
>
> If the maximum number of VLAN interfaces allowed on the device is reached, the next VLAN that is created will not have an associated VLAN interface created. It should be possible to have up to 64 VLAN interfaces.

Authentication Success/Failure Scenarios When Using VLAN Assignment

The following table shows when authentication succeeds or fails based on VLAN assignment configuration, RADIUS server response, and current port state. When **Allowed** the client is authenticated and the port is assigned to the specified VLAN, or it remains in the VLAN it is currently assigned to. When **Denied** the authentication fails and the port remains in its current VLAN and the client is not granted access to the network.

> **ABOUT THESE SCENARIOS**
>
> The table below assumes that basic port access authentication requirements are met. Without VLAN assignment enabled, the authentication would succeed. The scenarios describe only the additional checks performed for VLAN assignment that determine whether authentication is granted or denied and what the resulting VLAN assignment for the port will be when authentication is granted.

A	B	C	D	E	F	Status	Details
No	No	-	-	-	-	✅ Allowed	Port remains in statically configured VLAN
No	Yes	No	-	-	-	❌ Denied	No VID and no default VLAN configured
No	Yes	Yes	No	-	-	✅ Allowed	Port assigned to Default VLAN
No	Yes	Yes	Yes	Yes	-	✅ Allowed	Default VLAN matches current assignment
No	Yes	Yes	Yes	No	-	❌ Denied	No preemption, Default VLAN differs from current assignment
Yes	No	-	-	-	-	✅ Allowed	Port remains in statically configured VLAN (VLAN assignment disabled, VID ignored)
Yes	Yes	-	-	-	No	❌ Denied	VID not in allowed VLANs list

A	B	C	D	E	F	Status	Details
Yes	Yes	-	No	-	Yes	✅ Allowed	Port assigned to server-provided VID
Yes	Yes	-	Yes	Yes	Yes	✅ Allowed	VID matches current assignment
Yes	Yes	-	Yes	No	Yes	❌ Denied	No preemption, VID differs from current assignment

- **A:** RADIUS server provides a VLAN ID (VID) in the Tunnel-Private-Group-ID attribute as part of the authentication response.
- **B:** VLAN assignment is enabled in the port access configuration.
- **C:** Default VLAN is enabled in the port access configuration.
  - When default VLAN is enabled it will currently be implicitly set to be the VLAN that the port is statically configured as an untagged member of. So it happens to be the same as the starting VLAN for the port, but it is a different concept from the starting VLAN. (It is possible that the default VLAN could be configurable with a different VLAN than the starting VLAN in the future.)
- **D:** There are other clients authenticated on the same port.
- **E:** The VID to be assigned matches the currently assigned VLAN for the port.
- **F:** The VID to be assigned is in the configured allowed VLANs list for the port access instance.

RADIUS Server Communication Attribute Value Pairs

This section describes the RADIUS Attribute Value Pairs (AVPs) that WeOS provides in the RADIUS communication for port access authentication. These attributes are included in the authentication requests for both 802.1X and MAC authentication.

Understanding these attributes can be helpful when configuring your RADIUS server to work with WeOS, or when troubleshooting authentication issues.

MAC Address Format in RADIUS Attributes

When MAC addresses are provided in RADIUS attributes, they use the following format by default, as specified by RFC 3580[1]:

- Hyphen-separated hexadecimal octets in uppercase
- Example: `00-A1-B2-C3-D4-E5`

This format is used in attributes such as `Calling-Station-ID` and `Called-Station-ID`. For MAC authentication (MAB), it also applies to `User-Name` and `User-Password` attributes when **Service-Type** is set to **Call-Check**.

RADIUS Attributes for 802.1X

When performing 802.1X authentication, WeOS sends the following RADIUS attributes in the authentication request:

Attribute Name	ID	Value	Description
`Message-Authenticator`	80	Calculated value	Used to protect the integrity of the RADIUS message
`User-Name`	1	Username from client	The username provided by the client in the EAPOL request

[1] https://tools.ietf.org/html/rfc3580

Attribute Name	ID	Value	Description
`Called-Station-ID`	30	Port MAC address	The MAC address of the port the client is authenticating on (format: `XX-XX-XX-XX-XX-XX`)
`NAS-Port-Type`	61	15 (Ethernet)	Indicates the physical port type is Ethernet
`Service-Type`	6	2 (Framed-User)	Indicates 802.1X authentication
`NAS-Port`	5	Port ifindex	The interface index of the authenticating port
`NAS-Port-Id`	87	Port name	Human-readable port name (e.g., `eth1`)
`Calling-Station-ID`	31	Client MAC address	The MAC address of the client trying to authenticate (format: `XX-XX-XX-XX-XX-XX`)
`Connect-Info`	77	Connection details	Information about the client's connection, such as link speed
`Framed-MTU`	12	1500	Standard MTU size for Ethernet
`EAP-Message`	79	EAP message	The EAP message from the client's EAPOL request
`NAS-IP-Address`	4	IP address	The IP address used to communicate with the RADIUS server

RADIUS Attributes for MAC Authentication

For MAC authentication (MAB), different RADIUS attributes are sent depending on the configured **Service-Type**. The Service-Type can be configured as either **Call-Check** (default) or **Authenticate-Only**.

Attributes with Service-Type Call-Check (Default)

When **Service-Type** is set to **Call-Check**, which is the default and recommended setting for MAB, the following attributes are sent:

Attribute Name	ID	Value	Description
`Message-Authenticator`	80	Calculated value	Used to protect the integrity of the RADIUS message
`Service-Type`	6	10 (Call-Check)	Indicates MAC authentication bypass (MAB)
`Calling-Station-ID`	31	Device MAC address	The MAC address of the device trying to authenticate (format: `XX-XX-XX-XX-XX-XX`)
`NAS-Port-Type`	61	15 (Ethernet)	Indicates the physical port type is Ethernet
`Framed-MTU`	12	1500	Standard MTU size for Ethernet
`User-Name`	1	Device MAC address	Same MAC address as `Calling-Station-ID`
`User-Password`	2	Device MAC address (encrypted)	Same MAC address as `Calling-Station-ID`, encrypted
`NAS-Port`	5	Port ifindex	The interface index of the authenticating port

Attribute Name	ID	Value	Description
`NAS-IP-Address`	4	IP address	The IP address used to communicate with the RADIUS server

> **NOTE**
>
> Many RADIUS servers may expect the **Call-Check** service type for MAB, with the device's MAC address provided in both the `User-Name` and `User-Password` attributes. Using this service type is a common approach to indicate to the RADIUS server that MAC authentication bypass (MAB) is being performed.

Attributes with Service-Type Authenticate-Only

When **Service-Type** is set to **Authenticate-Only**, the following attributes are sent:

Attribute Name	ID	Value	Description
`Message-Authenticator`	80	Calculated value	Used to protect the integrity of the RADIUS message
`Service-Type`	6	8 (Authenticate-Only)	Alternative service type for MAC authentication
`Calling-Station-ID`	31	Device MAC address	The MAC address of the device trying to authenticate (format: `XX-XX-XX-XX-XX-XX`)
`NAS-Port-Type`	61	15 (Ethernet)	Indicates the physical port type is Ethernet
`Framed-MTU`	12	1500	Standard MTU size for Ethernet
`NAS-Port`	5	Port ifindex	The interface index of the authenticating port
`NAS-IP-Address`	4	IP address	The IP address used to communicate with the RADIUS server

> **NOTE**
>
> The **Authenticate-Only** service type is provided for backwards compatibility. Note that when using this service type, the `User-Name` and `User-Password` attributes are not included in the request. Many RADIUS servers commonly expect the **Call-Check** service type for MAB.

CLI Command Syntax

Port Access Control Context

Port Access Control context is reached via the General Configuration context. Multiple port access control instances can be configured, if no instance ID is provided it will create a new instance with the next available instance ID:

```
example:/#> config
example:/config/#> port-access
Creating new Port Access instance 1.
example:/config/port-access-1/#>
```

In order to configure a specific instance, the instance ID must be provided (can be tab-completed):

```
example:/#> config
example:/config/#> port-access 1
example:/config/port-access-1/#>
```

`[no|show] port-access [INSTANCE-NUM]`

Description

Port access control settings (IEEE 802.1X and MAC Auth)

Options

Option	Description
`no`	Remove port access control.
`INSTANCE-NUM`	Port access instance number. If not provided, the next available instance number will be used to create a new instance.

`[no] enable`

Description

Activate or deactivate port access control. Deactivating does not remove port access control configuration.

Options

Option	Description
`no`	Disable/deactivate port access control.

`[no] <PORT[,PORT ...] | PORT..PORT>`

Description

Port(s) to be managed by 802.1X and/or MAC authentication.

The selected ports will be managed by the port access control instance. All the selected ports will be locked when this port access instance is enabled. The only way to gain access on any of these ports will be to authenticate using either 802.1X or MAC authentication.

A Port is valid for selection if it fulfills the following criteria:

- Is a physical ethernet port.
- Is a bridge port, i.e. it is configured as part of a VLAN.
  - If VLAN assignment is enabled the port cannot be tagged in any VLAN, it is only allowed to be untagged.
- The port is not configured as part of another port access instance.

continues on next page

continued

`[no] <PORT[,PORT ...] | PORT..PORT>`

> **WARNING**
>
> Be careful when selecting ports, as it is possible to lock yourself out if the device is accessed remotely, e.g. via SSH or Web interface, and one of the selected ports is the one the device is accessed through.

Examples

```
example:/config/port-access-1/#> port eth1
example:/config/port-access-1/#> port eth1 eth2 eth3
example:/config/port-access-1/#> port eth1..eth5
example:/config/port-access-1/#> no port eth1
```

Options

Option	Description
`no`	Disable port access control of PORT(S)
`PORT`	A single port, e.g. eth1.

`[no] description [STRING]`

Description

Free form string describing current port access control configuration.

Options

Option	Description
`no`	Remove description.
`STRING`	Free form string. If the string contains spaces, it must be enclosed in quotes `"Example description"`.

`[no] port-method [mac-based|port-based]`

Description

Set port access control method.

The port-method determines how authentication behaves for clients on the managed ports. The behavior of the different methods is:

- **mac-based**: Clients are authenticated based on their MAC address. In other words, this means that all clients must be individually authenticated to be granted access.
- **port-based**: If a single client is authenticated on the port, the entire port is opened, granting access to all other clients on the network.

continues on next page

continued

`[no] port-method [mac-based|port-based]`

BE AWARE OF THE NETWORK TOPOLOGY WHEN USING PORT BASED METHOD

Be mindful of the fact that any device connected to a port with port-based authentication will be able to access the network once a client is authenticated. Therefore, when using this mode, consider carefully what is located behind the port(s). For instance if there will only be one single end device connected to the port, such as a printer or a camera, port-based authentication might be sufficient. But if there is a possibility that there will be multiple end devices behind the port(s) consider who is responsible for authenticating and which other devices may be granted access afterwards.

Default: `mac-based`

Examples

```
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> port-method port-based
example:/config/port-access-1/#> no port-method
```

Options

Option	Description
`no`	Reset to default (mac-based)
`mac-based`	Keyword to select MAC Based port method.
`port-based`	Keyword to select Port Based port method.

`[no] dot1x`

Description

Enable (and enter) or disable the 802.1X sub-context.

When enabled, this option enters the 802.1X configuration context. See 802.1X configuration context below.

Options

Option	Description
`no`	Disable 802.1X.

`[no] mac-auth`

Description

Enable (and enter) or disable the MAC authentication sub-context.

When enabled, this option enters the MAC authentication configuration context. See MAC authentication configuration context below.

continues on next page

continued

`[no] mac-auth`

Options

Option	Description
`no`	Disable MAC authentication.

`[no] vlan-assignment`

Description

Enable (and enter) or disable the VLAN assignment sub-context.

When enabled, this option enters the VLAN assignment configuration context. See VLAN assignment configuration context below.

Options

Option	Description
`no`	Disable VLAN assignment.

802.1X Configuration Context

802.1X configuration is a sub-context of port access control context.

```
example:/#> config
example:/config/#> port-access
example:/config/port-access/#> dot1x
example:/config/port-access/dot1x/#>
```

`[no] enable`

Description

Enable or disable 802.1X authentication.

Options

Option	Description
`no`	Disable 802.1X authentication

`[no] authentication-server <SERVER-ID|SERVER-ID,SERVER-ID>`

Description

Select one or two remote authentication servers to use.

If a secondary remote authentication server is also configured, it will be tried if the first remote server has failed retransmitting according to 802.1x standard behaviour. A retry connecting to the first remote authentication server will be done 600 seconds after the first connection attempt.

continues on next page

continued

`[no] authentication-server <SERVER-ID|SERVER-ID,SERVER-ID>`

The **SERVER_ID** represents a `remote-server` configured under the `aaa` configuration context. Only remote servers of type **radius** can be used as authentication servers for 802.1X.

> **TAB COMPLETION**
>
> The command supports tab-completion of configured remote servers.

Examples

```
example:/config/port-access-1/dot1x/#> authentication-server 1
example:/config/port-access-1/dot1x/#> authentication-server 1,2
example:/config/port-access-1/dot1x/#> no authentication-server
```

Options

Option	Description
`no`	Disable the use of the given authentication server.
`SERVER-ID`	The ID of the (RADIUS) authentication server to use. If no authentication server with the specified ID exists or is misconfigured, a warning will be shown. For information on configuring authentication server, see AAA configuration.

`[no] active-authentication [INTERVAL]`

Description

Enable or disable active-authentication.

When enabled, the device will periodically send EAP Identity Request frames to attempt to initiate authentication with potential supplicants located behind the configured port access ports. If any supplicant is authenticated successfully on a port, the periodic EAP Identity Request will stop for that port. The periodic EAP Identity Request will resume if all authenticated supplicants on the port are de-authenticated.

The interval is specified in seconds and can be configured to a value in the range 10..300 seconds.

Default: Enabled (30 seconds)

Examples

```
example:/config/port-access-1/dot1x/#> active-authentication
example:/config/port-access-1/dot1x/#> active-authentication 30
example:/config/port-access-1/dot1x/#> no active-authentication
```

Options

Option	Description
`no`	Disables active-authentication

continues on next page

continued

`[no] active-authentication [INTERVAL]`

Option	Description
`INTERVAL`	INTERVAL in seconds (range 10-300). Argument is optional with default 30 seconds.

`[no] re-authenticate [INTERVAL]`

Description

Enable or disable periodic re-authentication.

When enabled, the device will periodically request re-authentication for already authenticated clients. If an authenticated client fails to re-authenticate, the client will be de-authenticated.

The interval is specified in seconds and can be configured to a value in the range 10..352800 seconds.

Default: Disabled (when enabled, default is 3600 seconds)

Examples

```
example:/config/port-access-1/dot1x/#> re-authenticate
example:/config/port-access-1/dot1x/#> re-authenticate 600
example:/config/port-access-1/dot1x/#> no re-authenticate
```

Options

Option	Description
`no`	Disables periodic re-authentication
`INTERVAL`	INTERVAL in seconds (range 10-352800 seconds, i.e., 10 seconds to 1 week). Argument is optional with default 3600 seconds.

`[no] quiet-period [INTERVAL]`

Description

Enable or disable quiet period.

The quiet period is a time interval, after a failed authentication attempt, during which any new authentication requests from the same client will not be processed.

The quiet period is useful to prevent immediate repeated authentication attempts from a client that is unable to authenticate.

The interval is specified in seconds and can be configured to a value in the range 3..3600 seconds.

Default: Enabled (10 seconds)

continues on next page

continued

`[no] quiet-period [INTERVAL]`

Examples

```
example:/config/port-access-1/dot1x/#> quiet-period
example:/config/port-access-1/dot1x/#> quiet-period 10
example:/config/port-access-1/dot1x/#> no quiet-period
```

Options

Option	Description
`no`	Disables quiet period
`INTERVAL`	INTERVAL in seconds (range 3-3600).

MAC Auth Configuration Context

MAC authentication configuration is a sub-context of port access context.

```
example:/#> config
example:/config/#> port-access
example:/config/port-access/#> mac-authentication
example:/config/port-access/mac-auth/#>
```

`[no] enable`

Description

Enable or disable MAC authentication.

Options

Option	Description
`no`	Disable MAC authentication.

`[no] authentication-server <SERVER-ID>`

Description

Select a remote authentication server to use for MAB.

The **SERVER_ID** represents a `remote-server` configured under the `aaa` configuration context. Only remote servers of type **radius** can be used as authentication servers for MAC Authentication Bypass (MAB).

> **TAB COMPLETION**
>
> The command supports tab-completion of configured remote servers.

continues on next page

continued

`[no] authentication-server <SERVER-ID>`

Examples

```
example:/config/port-access-1/mac-auth/#> authentication-server 1
example:/config/port-access-1/mac-auth/#> no authentication-server
```

Options

Option	Description
`no`	Disable the use of the given authentication server.
`SERVER-ID`	The ID of the (RADIUS) authentication server to use. If no authentication server with the specified ID exists or is misconfigured, a warning will be shown. For information on configuring authentication server, see AAA configuration.

`[no] service-type [call-check|authenticate-only]`

Description

Select the service type used for the MAB RADIUS requests.

Configure the RADIUS Service-Type attribute for MAC Authentication Bypass (MAB) requests.

The Service-Type attribute determines how authentication credentials are sent to the RADIUS server during MAB authentication:

- **call-check**: Uses Service-Type 10 (Call Check). The device will use the client's MAC address as both username and password in the authentication request to the RADIUS server.
- **authenticate-only**: Uses Service-Type 8 (Authenticate Only). The device will not send any username or password in the authentication request to the RADIUS server.

Default: `call-check`

Examples

```
example:/config/port-access-1/mac-auth/#> service-type authenticate-only
example:/config/port-access-1/mac-auth/#> service-type call-check
example:/config/port-access-1/mac-auth/#> no service-type
```

Options

Option	Description
`no`	Reset to default (call-check)

`[no] accept-list <MAC-LIST-ID>`

Description

Specify a local MAC address whitelist to use for authentication.

The **MAC_LIST_ID** represents a `mac-auth` list configured under the `aaa` configuration context. This allows local authentication of specific MAC addresses without requiring a RADIUS server.

continues on next page

continued

`[no] accept-list <MAC-LIST-ID>`

TAB COMPLETION

The command supports tab-completion of configured MAC lists.

Examples

```
example:/config/port-access-1/mac-auth/#> accept-list 1
example:/config/port-access-1/mac-auth/#> no accept-list
```

Options

Option	Description
`no`	Disable the use of the given MAC list.
`MAC-LIST-ID`	The ID of the MAC List to use. If the specified MAC-LIST-ID does not exist or is misconfigured, a warning will be shown. For information on creating MAC lists, see [AAA configuration][mac-list].

VLAN Assignment Configuration Context

VLAN assignment configuration is a sub-context of port access context.

```
example:/#> config
example:/config/#> port-access
example:/config/port-access/#> vlan-assignment
example:/config/port-access/vlan-assignment/#>
```

`[no] enable`

Description

Enable or disable VLAN assignment.

When enabled, the authenticated ports will be assigned to VLANs based on information received from the authentication server.

When enabled the server must provide VLAN information in the authentication response, otherwise the port will not be authenticated. The exception is when `default-vlan` is enabled.

In addition, the VLAN provided by the server must be in the list of `allowed-vlans` configured for the instance, otherwise the port will not be authenticated. For a VLAN to be allowed, it must be explicitly listed in the `allowed-vlans` configuration.

NO TAGGED VLANS ALLOWED

When VLAN assignment is enabled, the configured port access ports cannot be tagged in any VLANs. The ports must be untagged members of one VLAN (the PVID), as soon as VLAN assignment is enabled.

Default: Disabled

continues on next page

continued

`[no] enable`

Options

Option	Description
`no`	Disable VLAN assignment.

`[no] allowed-vlans [VLAN [VLAN...]]`

Description

Specify which VLANs are allowed for assignment.

The list of allowed VLANs specifies which provided VLANs from the authentication server that are permitted for assignment to the authenticated ports. If the server provides a VLAN that is not in this list, the port will not be authenticated.

For a VLAN to be selectable, it must be manually configured in the VLAN configuration. In other words, the VLAN will not be automatically created by the VLAN assignment handling, it will only assign ports to existing VLANs.

> **SAME AS ALLOWED-VIDS**
>
> The `allowed-vlans` and `allowed-vids` commands configure the same setting, just with different input formats. These two commands are simply aliases for the same configuration, so they have the same effect and constraints.

> **TAB COMPLETION**
>
> The command supports tab-completion of existing VLANs on the device.

Default: No allowed VLANs are configured

Examples

The VLANs are specified by their entire interface name, this supports tab completion for existing VLANs.

```
example:/config/port-access-1/vlan-assignment/#> allowed-vlans vlan10
example:/config/port-access-1/vlan-assignment/#> allowed-vlans vlan10 vlan20
example:/config/port-access-1/vlan-assignment/#> no allowed-vlans
```

Options

Option	Description
`no`	Clear the list of allowed VLANs.
`VLAN`	The VLAN to allow for assignment, specified by its interface name, e.g. **vlan10**. The VLAN must already exist on the device to be selectable.

`[no] allowed-vids [VID [VID...]]`

Description

Specify which VLAN IDs (VIDs) are allowed for assignment.

The list of allowed VLAN IDs specifies which provided VLAN IDs from the authentication server that are permitted for assignment to the authenticated ports. If the server provides a VLAN ID that is not in this list, the port will not be authenticated.

For a VID to be selectable, the VLAN corresponding to that VID must be created manually in the VLAN configuration. The VLAN for the VID will not be created automatically by the VLAN assignment handling; it will only assign ports to existing VLANs.

> **SAME AS ALLOWED-VLANS**
>
> The `allowed-vids` and `allowed-vlans` commands configure the same setting, just with different input formats. These two commands are simply aliases for the same configuration, so they have the same effect and constraints.

> **TAB COMPLETION**
>
> The command supports tab-completion of existing VLANs on the device, by their VID in the format `10`, e.g. `vlan10` for VID 10.

Default: No allowed VLAN IDs are configured

Examples

The VLAN IDs are specified by their numeric VID value.

```
example:/config/port-access-1/vlan-assignment/#> allowed-vids 10
example:/config/port-access-1/vlan-assignment/#> allowed-vids 10 20 30
example:/config/port-access-1/vlan-assignment/#> no allowed-vids
```

Options

Option	Description
`no`	Clear the list of allowed VLAN IDs.
`VID`	The numeric VLAN ID (VID) to allow for assignment, e.g. **10**. The VLAN corresponding to the VID must already exist on the device to be selectable.

`[no] default-vlan`

Description

Enable or disable default VLAN assignment.

When enabled, assignment to a default VLAN for an authenticated port is allowed, if the server does not provide any VLAN information in the authentication response.

continues on next page

continued

`[no] default-vlan`

When disabled, the server must provide VLAN information in the authentication response, otherwise the port will not be authenticated.

The default-vlan for a port is the configured PVID of the port, i.e. the VLAN the port is an untagged member of. Every selected port access port, for the instance, is required to be configured untagged in a VLAN, as soon as VLAN assignment is enabled.

> **NO TAGGED VLANS ALLOWED**
>
> When VLAN assignment is enabled, the selected ports cannot be tagged in any VLANs, only untagged VLANs are allowed.

Default: Disabled

Options

Option	Description
`no`	Disable default VLAN assignment.

Port Access Control Status

Port access control status can be viewed and monitored through several different commands and views. The status information is organized into four main sections, each focusing on different aspects of port access:

- **802.1X Authentications** - Currently authenticated 802.1X clients (supplicants), showing MAC address, supplicant username, port, authentication server, and date added.
- **MAC Auth Authentications** - Currently authenticated MAC addresses, showing MAC address, port, authentication method (MAB or accept-list), and date added.
- **Ports** - Port access port status, showing port name, port method (MAC Based or Port Based), state (unlocked/locked or number of authentications), and list of authenticated MAC addresses.
- **VLAN Assignment** - VLAN assignment status for each port, showing whether VLAN assignment is enabled, current assignment state, assignment source (server or default), assigning client MAC, and assigned VLAN.

The status can be viewed in two main ways:

1. **Direct command execution** - Run status commands directly from the admin-exec context using `show port-access [options]`.
2. **Status context** - Enter the `port-access` status context to access status commands and enable additional display options.

Viewing Status from Admin-Exec Context

Summary View

To view a summary of all port access status information:

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled

╒ 802.1X Authentications ═══════════════════════════════════════════════
# MAC-ADDRESS        SUPPLICANT    PORT    SERVER          DATE-ADDED
1  76:03:0e:ff:ba:ab  supplicant1   eth1    198.19.10.1     Feb 26 16:26:26

╒ MAC Auth Authentications ═══════════════════════════════════════════
# MAC-ADDRESS        PORT    METHOD                          DATE-ADDED
1  76:03:0e:eb:11:68  eth3    MAB (198.19.10.1:4000)          Feb 26 16:19:39

╒ Ports ═══════════════════════════════════════════════════════════════
PORT    METHOD      STATE                           AUTHENTICATED-MAC
eth1    MAC Based   1 Authentication                76:03:0e:ff:ba:ab
eth2    MAC Based   0 Authentications               -
eth3    MAC Based   1 Authentication                76:03:0e:eb:11:68

╒ VLAN Assignment ═══════════════════════════════════════════════════
PORT    ENABLED  STATE       FROM     ASSIGNING-CLIENT   VLAN
eth1    No       Unassigned  -        -                  -
eth2    No       Unassigned  -        -                  -
eth3    Yes      Assigned    Server   76:03:0e:eb:11:68  VID 30 (vlan30)

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

Individual Status Tables

Individual status tables can also be displayed separately, as shown in the examples below.

802.1X Authentications

```
example:/#> show port-access dot1x-authentications
#  MAC-ADDRESS        SUPPLICANT      PORT    SERVER          DATE-ADDED
1  76:03:0e:ff:ba:ab  supplicant1     eth1    198.19.10.1     Feb 26 16:26:26
example:/#>
```

Column	Description
#	Sequential entry number
MAC-ADDRESS	MAC address of the authenticated client
SUPPLICANT	Supplicant username provided during authentication (if available)
PORT	Port name where the client is authenticated (e.g., eth1, eth2)
SERVER	IP address of the RADIUS server that authenticated the client
DATE-ADDED	Date and time when the client was authenticated

MAC Auth Authentications

```
example:/#> show port-access mac-auth-authentications
#  MAC-ADDRESS        PORT    METHOD                          DATE-ADDED
1  76:03:0e:eb:11:68  eth3    MAB (198.19.10.1:4000)          Feb 26 16:19:39
example:/#>
```

Column	Description
#	Sequential entry number
MAC-ADDRESS	MAC address of the authenticated client
PORT	Port name where the client is authenticated
METHOD	Authentication method: “MAB (server:port)” for RADIUS-based authentication, or “accept-list” for local list authentication
DATE-ADDED	Date and time when the client was authenticated

Ports

```
example:/#> show port-access ports
PORT    METHOD      STATE                          AUTHENTICATED-MAC
eth1    MAC Based   1 Authentication               76:03:0e:ff:ba:ab
eth2    MAC Based   0 Authentications              -
eth3    MAC Based   1 Authentication               76:03:0e:eb:11:68
example:/#>
```

Column	Description
PORT	Port name (e.g., eth1, eth2)
METHOD	Port method: “MAC Based” or “Port Based”
STATE	For **MAC Based**: number of authenticated clients (e.g., “2 Authentications”). For **Port Based**: “Locked” or “Unlocked”
AUTHENTICATED-MAC	Comma-separated list of MAC addresses currently authenticated on the port

VLAN Assignment

```
example:/#> show port-access vlan-assignment
PORT    ENABLED  STATE       FROM     ASSIGNING-CLIENT     VLAN
eth1    No       Unassigned  -        -                    -
eth2    No       Unassigned  -        -                    -
eth3    Yes      Assigned    Server   76:03:0e:eb:11:68    VID 30 (vlan30)
example:/#>
```

Column	Description
PORT	Port name
ENABLED	Whether VLAN assignment is enabled for the port (Yes/No)
STATE	Assignment state: `Assigned` or `Unassigned`
FROM	Source of VLAN assignment: `Server` for RADIUS-provided, `Default` for default VLAN, or `-` if unassigned
ASSIGNING-CLIENT	MAC address of the client that triggered the VLAN assignment
VLAN	Assigned VLAN in the format `VID <number> (<name>)`, or `-` if unassigned

Instance-Specific Status

To view status for a specific instance, provide the instance ID:

```
example:/#> show port-access 1
[Instance 1 status displayed ... ]
example:/#>
```

> **TAB COMPLETION**
>
> The command supports tab-completion for the existing instance IDs.

Audit Log

To view port access related audit log messages:

```
example:/#> show port-access audit
[Audit log messages related to port access ... ]
example:/#>
```

> **AUDIT TRAIL ACCESS REQUIRED**
>
> Access to the audit log is restricted to user roles with **Audit trail view** permissions. See the Authorization levels section in the AAA documentation for more information about roles and permissions.

JSON Format

To view status in JSON format:

```
example:/#> show port-access json
{
        "instance_count": 1,
        "enabled_instance_count": 1,
        "instances": [
                {
                        "id": 1,
                        "dot1x": {
                                "enabled": true,
                                "pid": 1234,
                                "authentications": [ ... ]
                        },
                        "mac_auth": {
                                "enabled": true,
                                "pid": 1364,
                                "authentications": [ ... ]
                        },
                        "ports": [ ... ]
                }
        ]
}
example:/#>
```

Port Access Status Context

For more convenient access to status commands and additional display options, enter the port-access status context:

```
example:/#> port-access
example:/port-access/#>
```

Or for a specific instance:

```
example:/#> port-access 1
example:/port-access-1/#>
```

Within this context, the following commands are available:

`show`

Description

Show a summary of all port access status information.

This command displays all four status tables: 802.1X Authentications, MAC Auth Authentications, Ports, and VLAN Assignment. When executed from the general port-access context (without a specific instance), it shows information for all configured instances. When executed from a specific instance context, it shows only that instance's information.

The same output can be obtained in JSON format by using `show json`.

`show dot1x-authentications`

Description

Show currently authenticated 802.1X clients.

Displays a table of all currently authenticated 802.1X clients (supplicants).

`show mac-auth-authentications`

Description

Show currently authenticated MAC addresses.

Displays a table of all currently authenticated MAC addresses using MAC authentication (MAB or accept-list).

`show ports`

Description

Show port access port status.

Displays a table showing the status of all ports configured for port access.

`show vlan-assignment`

Description

Show port access VLAN assignment status.

Displays a table showing the VLAN assignment status for each port access port.

> **NOTE**
>
> A port is considered "Assigned" only when port access has dynamically assigned it to a VLAN (either from RADIUS server or default VLAN), even if the assigned VLAN is the same as the port's starting VLAN.

`show audit`

Description

Show port access related log messages from the audit log.

This command serves as a quick way to access relevant audit log messages related to port access directly from this context. The messages displayed are the same as those that would appear when viewing the full audit log, but filtered to show only port access related events.

> **AUDIT TRAIL ACCESS REQUIRED**
>
> Access to the audit log is restricted to user roles with **Audit trail view** permissions. See the Authorization levels section in the AAA documentation for more information about roles and permissions.

`show log`

Description

Alias for `audit`. Show port access related log messages from the audit log.

`[no] instance-dividers`

Description

Enable or disable dividers between instances in the output of show commands.

This is a non-persistent setting that only affects the display of show commands in this context. When enabled, show commands will include dividers between instances to make it easier to distinguish which information belongs to which instance.

Default: Disabled

Example with dividers enabled

```
example:/port-access/#> instance-dividers
example:/port-access/#> show
```

continues on next page

continued

`[no] instance-dividers`

```
Number of Port Access Instances: 2
Table Instance Dividers        : Enabled


 802.1X Authentications
#  MAC-ADDRESS         SUPPLICANT     PORT     SERVER           DATE-ADDED
Instance 1
1  76:03:0e:ff:ba:ab   supplicant1    eth1     198.19.10.1      Feb 26 16:26:26
Instance 2
No 802.1X Authentications for Port Access Instance 2


 MAC Auth Authentications
#  MAC-ADDRESS         PORT     METHOD                           DATE-ADDED
Instance 1
No MAC Auth Authentications for Port Access Instance 1
Instance 2
1  76:03:0e:eb:11:68   eth3     MAB (198.19.10.1:4000)           Feb 26 16:19:39


 Ports
PORT     METHOD       STATE                            AUTHENTICATED-MAC
Instance 1
eth1     MAC Based    1 Authentication                 76:03:0e:ff:ba:ab
eth2     MAC Based    0 Authentications                -
Instance 2
eth3     MAC Based    1 Authentication                 76:03:0e:eb:11:68


 VLAN Assignment
PORT     ENABLED  STATE        FROM     ASSIGNING-CLIENT   VLAN
Instance 1
eth1     No       Unassigned   -        -                  -
eth2     No       Unassigned   -        -                  -
Instance 2
eth3     Yes      Assigned     Server   76:03:0e:eb:11:68  VID 30 (vlan30)
```

2.4.6. Fast Reconfiguration of Network Topology (FRNT)

The Fast Reconfiguration of Network Topology (FRNT) protocol handles fast reconfiguration in switched ring and sub-ring topologies. When rapid convergence in case of link or switch failure is required, FRNT becomes the protocol of choice when it comes to layer-2 resilience and robustness.

Two versions of FRNT exist, FRNTv0 and FRNTv2. FRNTv2 natively supports topologies with regular rings and with sub-ring extensions, while FRNTv0 is limited to a single ring.

Introduction

The overall functionality of FRNTv0 and FRNTv2 is the same when configured in a ring topology. The FRNT protocol handles fast reconfiguration in switched ring topologies. One of the switches has the role of FRNT *focal point* while the other switches are referred to as FRNT *members*. When the switches are connected in a ring, it is the responsibility of the focal point to break the loop by putting one of its ports in blocking state, as shown in Figure 1.

```
.-----.                .-----.
|     |M             N|     |
| SW1 +-||------------+ SW2 |
|   fp|               |     |
'--+--'               '--+--'
  N|                     |M
   |        FRNT         |
  M|                     |N
.--+--.               .--+--.
|     |               |     |
| SW4 +---------------+ SW3 |
|     |N             M|     |
'-----'               '-----'
```

Figure 1: Basic FRNT ring configuration, SW1 is focal point (fp).

> **NOTE**
>
> In an FRNT ring or a sub-ring, only one of the switches can be configured as focal point. The other switches should be configured as member switches. For FRNT to operate properly, all devices in the ring must be running FRNT.

Fail-Over Handling

Once a link failure is detected somewhere along the ring, the focal point puts its blocking port in forwarding state to establish full connectivity between the switches. FRNT is event based, thus switches that detect a link down event will immediately send a link down FRNT message towards the focal point. Intermediate switches forward all FRNT messages, with highest priority, and the focal point opens its blocking port upon receiving the link down message.

```
.-----.                .-----.
|     |M             N|     |
| SW1 +---------------+ SW2 |
|   fp|               |     |
'--+--'               '--+--'
  N|                     |M
   |        FRNT         |
  M|                     |N
.--+--.               .--+--.
|     |               |     |
| SW4 +------ X ------+ SW3 |
|     |N             M|     |
'-----'               '-----'
```

Figure 2: Upon a link break, the focal point (SW1) puts its blocked port in forwarding state.

Similarly, when a broken link comes back up again and the ring is fully connected, the focal point will react and put its blocking port back to blocking state.

Link Monitoring and Qualification

An FRNT link must be qualified before it can be used as a active link in an FRNT ring. The link qualification is done with control traffic, i.e, FRNT *Hello* packets. The qualification with Hello packets verifies that there is two-way connectivity over the link. The primary reason for this is to ensure adequate operation of FRNT in both directions of a link, before taking it into operation. This feature provides robustness against one-way link or transmission failures, and is specifically important over a fibre optic link pairs.

> **NOTE**
>
> In FRNTv2 it is possible to disable Hello packets. The port will become qualified on link up and no further tests of the link quality is not done. The FRNTv2 switches on each side of the link must use the same qualification procedure.

Guidelines on Port selection and configuration

When enabling FRNT on a switch, two ports are set as ring ports. Below are some recommendations and rules when selecting and configuring the FRNT ports:

- Avoid using copper SFPs as FRNT ports: When using Ethernet ports as FRNT ports, choose fixed Ethernet ports or fibre SFPs. Copper SFPs may be used as FRNT ports, but will generally imply non-negligible degradation of fail-over performance.
- When activating FRNT on gigabit copper ports the setting ‘fastlink’‘ should be enabled to reduce the link down time. Per the IEEE 802.3 Clause 40 standard, a 1000BASE-T PHY is required to wait 750 milliseconds or more to report that link is down after detecting a problem with the link. It is not possible to meet the failover time for FRNT if the PHY follows the 750 wait time. By activating fastlink on a port the IEEE standard is intentionally violated.

For example, if port eth1 is a gigabit copper port used as FRNT ring port, enabling fastlink can be done as follows.

```
example:/#> configure
example:/config/#> port eth1
example:/config/port-eth1/#> fastlink
example:/config/port-eth1/#> leave
example:/#>
```

Multiple FRNT rings and sub-rings (FRNTv2)

FRNTv2 supports sub-ring topologies. Sub-rings can be thought of as a partial ring in the shape of a “C” that is not fully closed (a ”horse-shoe”). Sub-rings can be hooked into a regular FRNTv0 or FRNTv2 super-ring and to other FRNTv2 sub-rings as well. This allows for complex ring topologies to be built as shown in Figure 3.

```
 .----------.      .----------.       .----------.       .------------.
 |  Super-  |      |  Super-  |       |  Sub-    |       |  Super-    |
 |  Ring    |      |  Ring    |       |  Ring    |       |   Ring     |
 |          |      |          |      .+---.  .---+.      |            |
.+---.  .---+.    .+---.  .---+.     |    |  |    |     .+---.    .---+.
|    +--+    |    |    +--+    |     |    +--+    |     |    +----+    |
|    |  |    |    |    |  |    |     '+---'  '---+'     |    |    |    |
'+---'  '---+'    '+---'  '---+'      |  Super-  |      '+--+'    '+--+'
 |  Sub-    |      |  Sub-    |       |  Ring    |       |  | Sub- |  |
 |  Ring    |      |  Ring    |       |          |       |  | Ring |  |
 '----------'     .+---.  .---+.     .+---.  .---+.      |  '------'  |
                  |    +--+    |     |    +--+    |      |            |
                  |    |  |    |     |    |  |    |      |    Sub-    |
                  '+---'  '---+'     '+---'  '---+'      |    Ring    |
                   |  Sub-    |       |  Sub-    |       '------------'
                   |  Ring    |       |  Ring    |
                   '----------'       '----------'
```

Figure 3: Examples of sub-ring topologies

The switches in each end of the sub-ring will have only one FRNT ring port configured and one of those switches needs to be the focal point. The member switches between the sub-ring *end switches* are configured as normal member switches with two ring ports. When the sub-ring is intact the focal point will block its sub-ring port to prevent a loop. If a link break is detected in the sub-ring the focal point will open its blocked ring port to restore connectivity in the network.

> **NOTE**
>
> All FRNTv2 nodes in a ring (super-ring or sub-ring) must be configured with the same ring ID. The ring ID used in a super-ring and in a sub-ring must differ.

Below is a configuration example for a member FRNTv2 sub-ring end switch, i.e., only a single FRNT port is configured (here port eth4). It assumes there is already an FRNT super-ring configured (here an FRNTv2 ring with ring ID '1' and ring-ports eth1 and eth2).

```
example:/#> configure
example:/#> configure
example:/config/#> ring
example:/config/ring/#> frnt 2
example:/config/ring/frnt-2/#> port eth4
example:/config/ring/frnt-2/port-eth4/#> end
example:/config/ring/frnt-2/#> no focal-point
example:/config/ring/frnt-2/#> leave
example:/#> show frnt
                      Top
 Rid  Ver  Status     Cnt  Mode    Port 1                 Port 2
   1    2  OK           5  Focal   eth1    Qualified      eth2    Forwarding
   2    2  OK           3  Member  eth4    Forwarding     None
example:/#>
```

Reserved VLANs

Reserved VLANs for FRNT are:

- FRNTv0 uses VLAN IDs 4020-4022 and 4032-4033.
- FRNTv2 uses VLAN IDs 4021-4022.

Thus, when FRNT is enabled on a switch, these VLANs are implicitly reserved and cannot be configured by the user.

Configuration

FRNT is a top-level configuration context in the CLI:

```
example:/#> configure
example:/config/#> ring
example:/config/ring/#> frnt 1
example:/config/ring/frnt-1/#>
```

`[no] frnt [ID]`

Description

Enable/Disable FRNT and enter FRNT configuration context of given instance.

Options

Option	Description
`ID`	FRNT instance ID (range 0-255). At most 10 instances can be created. The selection of FRNT ID affects default values for FRNT *version* and FRNTv2 *Ring-ID*. • ID 0: Implies version '0'. Also implies ring ID '0', although FRNTv0 does not consider ring IDs. • ID 1-20: Implies version '2'. FRNTv2 ring ID defaults to the specified FRNT ID, but can be changed. Currently ID defaults '0'
`no`	Remove the given FRNT instance (or all if no ID is specified).

The available commands differs depending on FRNT version. Both are listed below, starting with FRNTv2 settings.

Syntax FRNTv2 Configuration

`version <0,2>`

Description

Set FRNT version '0' (FRNTv0) or '2' (FRNTv2).

Default:
- For FRNT Instance '0', the default version becomes '0'.
- For FRNT Instance '1' or higher, the default version becomes '2'.

Changing version affects ring ID. To change from FRNTv0 to FRNTv2, it is better to remove the FRNT instance and create a new instance with ID
0.

`ring-id <ID>`

Description

Set FRNTv2 Ring ID.

ID

Description

Ring ID in range 1-20. Defaults to be same as FRNT instance ID. All FRNT nodes in a ring needs to have same ring ID configured.

`[no] focal-point`

Description

Set FRNT mode: focal-point or member. Default: focal-point

Options

Option	Description
`no`	Set switch to act as member

`[no] port <PORT>`

Description

Set PORT as FRNT ring port, and enter configuration context for the given port. Typically the 'port PORT' command is run twice, one for each ring port. The exception are nodes which constitute the end of a *sub-ring*, when only a single port is specified.

The first configured port will by default be *blocking port* if the node is *focal-point*. See the 'blocked-port' setting below.

Options

Option	Description
`PORT`	Name of ring port, typically an Ethernet port such as *eth1*. Logical layer-2 ports (e.g., *ssl1* or *lag1*) can be FRNT ports as long as they are associated to the same *bridge*.

`[no] blocked-port <PORT>`

Description

(Setting only available in 'focal-point' mode.) Blocked port on focal point when the ring is OK. When no blocking port is specified the first configured port becomes the blocking port.

Options

Option	Description
`PORT`	The ring port to be blocked by focal point when ring is intact. PORT must be defined as ring port, see 'port' setting above.
`no`	Auto (let first configured port act as blocking port)

`[no] ring-interval <INTERVAL>`

Description

Expert setting. Default: 500 ms (Setting only available in 'focal-point' mode.)

Set interval of FRNT *ring* packets in milliseconds. The default is fine for all regular use cases.

Options

Option	Description
`INTERVAL`	Interval in milliseconds. Range 50-10000 (ms).
`no`	Reset to default (500)

`[no] guarded-recovery`

Description

Expert setting. Default: Enabled (Setting only available in 'focal-point' mode.)

This expert setting should be enabled in all regular use cases. When enabled, FRNT will use all ring control events to prevent a logical loop in the ring. When guarded-recovery is disabled, only ring packets are used when deciding if a ring port on focal point should be in blocked or forwarding state.

`[no] response-delay <INTERVAL>`

Description

Expert setting. Default: 1500 ms (Setting only available in 'focal-point' mode.)

A delay from ring closed until the focal point change state to ring ok. During the response delay the focal point waits for other link up events in the ring to prevent unnessasary changes in ring state.

Options

Option	Description
`INTERVAL`	Interval in milliseconds. Range 500-10000 (ms)
`no`	Reset to default (1500)

`[no] hello-time <INTERVAL>`

Description

Setting within FRNT *port configuration* context. FRNT Hello/Health packet interval, default: 500 msec

This expert level setting is primarily intended for improving recovery times on Gigabit links with slow link-down detection, i.e., where PHYs do not support fastlink. Hello time can be disabled and then then FRNT port is qualified on link up.

continues on next page

continued

`[no] hello-time <INTERVAL>`

> **WARNING**
>
> Although the lowest possible setting provides the quickest recovery time (if slow link-down) it also severely affects the CPU load of the system. It is therefore strongly recommended to experiment to figure out the best setting for the particular system at hand. E.g., routing or remote monitoring with SNMP share the same CPU resources. Use 'repeat show processes' to see system load at run-time.

Options

Option	Description
`INTERVAL`	Interval in milliseconds. Range 20-10000 (ms). Default: 500 (ms)
`no`	Disable use of Hello packets.

`[no] fwd-delay <DELAY>`

Description

Setting within FRNT *port configuration* context, and only available in *member* mode. Let port shift to forwarding state after specified forward delay (seconds). Default: Disabled

This expert setting sets the *forward delay* in seconds for the port. If forward delay is set, the port will change state to forwarding when the delay time has passed without receiving FRNT Hello packets. For normal ring operation this setting should be disabled.

Options

Option	Description
`DELAY`	Forward delay in seconds, range 3-60 (s).
`no`	Disable forward delay, i.e., await link qualification before consider shifting to forwarding state.

Syntax FRNTv0 Configuration

`version <0,2>`

Description

Set FRNT version '0' (FRNTv0) or '2' (FRNTv2).

Default:

- For FRNT Instance '0', the default version becomes '0'.
- For FRNT Instance '1' or higher, the default version becomes '2'.

Changing version affects ring ID. To change from FRNTv0 to FRNTv2, it is better to remove the FRNT instance and create a new instance with ID
0.

`[no] focal-point`

Description

Set FRNT mode: focal-point or member. Default: focal-point

Options

Option	Description
`no`	Set switch to act as member

`[no] ports [portM,portN]`

Description

Set ring ports, a.k.a. port M, port N.

In normal operation, the focal point selects port M to be blocked. If you have a preference, place that port first in the pair.

Options

Option	Description
`no`	Remove all or one port

`[no] hello-time [20-1000,20-1000]`

Description

Health packet interval (M,N), default: 100 msec.

This expert level setting is primarily intended for improving recovery times on Gigabit links with slow link-down detection.

FRNT Status Commands and Explanations

In order to see the current status of FRNT for an individual device the following command can be used:

`[show] frnt [clear-stats] [ID|all]`

Description

Show FRNT status and statistics, or clear FRNT statistics.

Use `show frnt [ID|all]` to show FRNT status and statistics for one or all FRNT instances. If specifying a specific ID, a detailed view of that instance is shown.

Use `frnt clear-stats [ID|all]` to reset topology change count for one or all FRNT instances.

Showing FRNT status overview

```
example:/#> show frnt
                      Top
Rid   Ver   Status     Cnt   Mode     Port 1                 Port 2
```

```
   0    0  OK          5  Focal   eth1    Blocking       eth2    Forwarding
   1    2  OK          3  Member  eth4    Forwarding     None
example:/#>
```

For a detailed explanation of various FRNT states, see further below.

Detailed FRNTv0 status

```
example:/#> show frnt 0
Ring ID                    : 0
Version                    : FRNTv0
Enabled                    : Yes, running as PID 2668
Ring Status                : OK
Topology Change Count      : 5
Time Since Last Change     : 0 Days 0 Hours 0 Mins 41 Secs
Mode                       : Focal Point
Port M                     : eth1     UP  Blocking
Port N                     : eth2     UP  Forwarding
example:/#>
```

Detailed FRNTv2 status (here with only a single ring port)

```
example:/#> show frnt 1
Ring id                    : 1
Version                    : FRNTv2
Enabled                    : Yes, running as PID 1826
Ring Status                : OK
Topology Change Count      : 3
Time Since Last Change     : 0 Days 0 Hours 0 Mins 28 Secs
Mode                       : Member
Port                       : eth4     UP  Forwarding
  Hello time               : 500
  Forward Delay            : Disabled

example:/#>
```

Port Status in FRNTv0 and FRNTv2 are summarised in table below.

Link state	Port State	FRNTv0 Port Status	FRNTv2 Port Status
Down	Down	Down	Down
Up	Blocking	Down	Not Qualified
Up	Blocking	Up	Qualified
Up	Forwarding	Forwarding	Forwarding

- **FRNTv0 Port status**
  - *Down*: FRNT does not receive hello packets from its attached FRNT node.
  - *Up*: FRNT receives hello packets from its attached FRNT node. If the port is in forwarding state it is active in the ring and it is forwarding traffic.
- **FRNTv2 Port status:**
  - *Down*: Link down on port.
  - *Not Qualified*: The port has link up but do not get response on transmitted hello packets from the attached FRNT node.

- *Qualified*: The port receives hello packets from its attached FRNT node. It is ready to be used in the ring if needed but is still blocked. Focal point will have one of its port in qualified state when the ring is intact.
- *Forwarding*: An active port in the ring and it is forwarding traffic.

2.4.7. Rapid Spanning Tree Protocol

Introduction

This document gives an overview of RSTP, a redundancy protocol for layer-2.

The spanning tree protocol (STP) and its successor rapid spanning tree protocol (RSTP) are the standard protocols to support redundancy while avoiding broadcast storms in switched networks. RSTP supports a fall-back to STP when connecting the switch to another device only capable of STP.

The purpose of the spanning tree protocol is to ensure that an arbitrary physical LAN topology is turned into a logical tree topology (i.e., loop free) in such a way that all links in the network are still connected (i.e., a spanning tree). This is accomplished by having the switches put some of their ports in blocking state.

Overview

The RSTP is always starting with all ports in blocking state. After the init phase one switch, only one, will be elected to root switch. The election is based of the mac addresses on the device and the bridge priority ID. The root switch is responsible to send out the RSTP frames, called Bridge Protocol Data Unit (BPDU) packet. The BPDU packet is an IEEE standard multicast packet and can go through a blocked port.

The RSTP service supports version IEEE 802.1D-2004.

```
RSTP network after the election is done

     .-----.               .-----.               .-----.
     |     |               |     |               |     |
     | SW1 +---------------+ SW2 |---------------+ SW3 |
     | Root|               |     |               |     |
     '--+--'               '--+--'               '--+--'
        |                     |                     \
        |                     |                     |
        |                     \                     |
     .--+--.               .--+--.               .--+--.
     |     |               |     |               |     |
     | SW4 +---------------+ SW5 |---------------+ SW6 |
     |     |               |     |               |     |
     '-----'               '-----'               '-----'

\ = port is in blocking mode
```

Figure 1: RSTP network, SW1 has the lowest bridge ID and is therefore elected to root switch. The bridge ID is combination of both bridge priority and the MAC address. The switch can be configured to act as a root switch with the bridge priority configuration.

Configuration

```
example:/#> configure
example:/config/#> spanning-tree
example:/config/spanning-tree/#>
```

`[no] spanning-tree`

Description

Configure spanning-tree on the device. The spanning-tree settings is global on the switch. It is only possible to have one spanning-tree instance running at the same time.

continues on next page

continued

`[no] spanning-tree`

Options

Option	Description
`no`	Disable the spanning-tree.

`[no] priority PRIO`

Description

Set bridge priority, where a low value means high priority, which increase the probability of being elected as root bridge.

Options

Option	Description
`no`	Use default bridge priority 8 (32768).
`PRIO`	An integer to set the bridge priority in range 0-15 (0-65536).

`[no] max-age-time AGE`

Description

Set spanning-tree max age timeout. Since bridges use the max age configured at the root bridge, this parameter setting only matters if this bridge becomes the root bridge.

Options

Option	Description
`no`	Use default bridge max-age-time. The default time is 20 seconds.
`AGE`	An integer to set the bridge maximum age timeout in the range of 6-40 seconds.

`[no] hello-time HELLO`

Description

Since bridges use the hello time configured at the root bridge, this parameter setting only matters if this bridge becomes the root bridge. This is the time between the hello BPDU packets.

Options

Option	Description
`no`	Use default bridge hello-time. The default time is 2 seconds.
`HELLO`	An integer to set the bridge hello-time interval in range of 1-10 seconds.

© 2026 Westermo Network Technologies AB

`[no] forward-delay FWD`

Description

Since bridges use the forward delay configured at the root bridge, this parameter setting only matters if this bridge becomes the root bridge. Used when operating in STP mode (i.e., not RSTP). Defines the time period by which the protocol can be sure that STP information on a topology change has propagated from one side of the network to the other. The STP convergence time is limited by twice the forwarding delay (plus the time it takes to detect the topology change).

Options

Option	Description
`no`	Use default bridge forward-delay. The default time is 15 seconds.
`FWD`	An integer to set the bridge forward-delay timeout in the range of 4-30 seconds.

Spanning Tree Port Configuration

Enter Spanning Tree Port Configuration context to manage per port spanning- tree settings for one or more ports.

```
example:/config/spanning-tree/#> port eth1
example:/config/spanning-tree/eth1/#>
```

`[no] enable`

Description

Enable/disable RSTP on this/these port(s).

Options

Option	Description
`no`	Disable RSTP on this/these port(s).

`[no] admin-edge`

Description

Ports where only end nodes connect are referred to as edge ports. If a port is only used for connecting hosts (i.e., no risk for loops), it can be configured as an admin edge port.

Options

Option	Description
`no`	Disable edge function on this/these port(s).

`[no] path-cost <COST>`

Description

Each port is assigned a ”cost”. This is used by each bridge to find the least cost path to the root bridge as part of the tree establishment.

Options

Option	Description
`no`	Disable the path-cost on this/these port(s). No path-cost resets to default, i.e., based on speed-duplex.
`COST`	Set path-cost to COST. This will override the path-cost based on speed-duplex. This can control which port that will be designated port. The cost is in range 0-20000000. The different path-cost due to the port speed : Speed in Mbps: path-cost value 10: 2000000 100: 200000 1000: 20000

2.4.8. Media Redundancy Protocol

Introduction

The Media Redundancy Protocol (MRP) is a protocol designed to handle redundancy in switched ring layer-2 topologies. Other available ring protocols that can handle the same situations and use-cases include FRNT.

For an example use-case see:

- MRP Basic
- MRP Over Layer 2 Tunnel

Overview

MRP is used to provide redundancy in ring topologies and it operates on layer-2 of the OSI model. A basic MRP ring consist of one Media Redundancy Manager (MRM) and one or more Media Redundancy Clients (MRC). The goal is that all devices in the ring should be reachable even if there is a link break somewhere in the ring.

```
.-------.               .-------.
|       |S              |       |
|  MRM  +-||------------+ MRC-1 |
|       |               |       |
'---+---'               '---+---'
   P|                       |
    |        MRP RING       |
    |                       |
.---+---.               .---+---.
|       |               |       |
| MRC-3 +---------------+ MRC-2 |
|       |               |       |
'-------'               '-------'
```

Figure 1: Basic MRP ring configuration where the ring is in the CLOSED state, the ring is intact.

The state of an operating MRP ring can be either *Closed* or *Open*. When the ring is in the *closed* state the ring is intact (no link breaks), the MRM will put its secondary ring port in **blocking** mode to avoid a loop in the topology. An example of this can be seen in Figure 1.

```
.-------.               .-------.
|       |S              |       |
|  MRM  +---------------+ MRC-1 |
|       |               |       |
'---+---'               '---+---'
   P|                       |
    |        MRP RING       |
    |                       |
.---+---.               .---+---.
|       |               |       |
| MRC-3 +------ X ------+ MRC-2 |
|       |               |       |
'-------'               '-------'
```

Figure 2: Basic MRP ring configuration where the ring is in the OPEN state, the ring is broken.

If a link or node in the ring goes down, the ring will effectively broken and MRP will transition to its *open* state. This will cause the MRM to put its secondary port in **forwarding** mode. An example of this can be seen in Figure
2.

If the primary port goes down, the other MRM port will become primary, i.e., the roles can change dynamically depending on the link state.

> **NOTE**
>
> Currently its assumed that all MRCs that is part of the MRP ring are capable of blocking their ring ports. Therefore, utilising devices in the ring that are incapable of blocking their ports is not supported.

Ring roles

Each MRP device needs to be configured to a specific role.

- **Media Redundancy Manager (MRM):** Responsible for managing the ring and ensuring that its second port is blocked when the ring is *closed* to avoid loops, and that it is put in forwarding when the ring is open. A single MRP ring can only have one MRM in the topology.
- **Media Redundancy Automanager (MRA):** Currently **NOT supported**. A MRP node designated as a MRA will be made available for an election process to select the manager of the ring automatically, hence Automanager.
- **Media Redundancy Client (MRC):** A client node in the MRP ring.

Multiple instances

Max two instances of MRP are supported but with some restrictions. Two instances can be configured if both are MRM but only one instance is supported when configured as an MRC. Each MRM is configured in the same way as an independent instance but with the restriction that the ring ports can not be shared with other MRM instances.

MRP Signalling

MRP operates by sending frames called *MRP Test* frames from the MRM. The frames will be sent in both directions on the ring from the both of the ring ports on the MRM. If the *MRP Test* frames from one of the ring ports reaches the other ring port, i.e. the frame has travelled around the entirety of the ring, the ring is intact (Ring status closed).

In addition to detecting link failure by observing if the *MRP Test* frames travel around the ring, the MRCs can also send *MRP Link Change* frames towards the MRM.

When the MRM notices, in any way, that the ring is broken, it will send out a *MRP Topology Change* frame towards all of the MRCs in the ring. When a MRC receives a *MRP Topology Change* frame it informs the client that there has been a topology change in the ring and that the FDBs should be flushed.

The following MAC multicast addresses are used by MRP:

OUI	Extension Identifier	Meaning
01:15:4E	00:00:01	Used for *MRP Test* frames.
01:15:4E	00:00:02	Used for *MRP Link Change*, *MRP Topology Change* and *MRP Option* frames

Recovery profiles

Multiple different recovery profiles can configured for MRP, that defines what the max reconfiguration time of the topology should be. The profiles are specified in milliseconds. Below the currently supported recovery profiles are presented, along with *some* of the differences:

Parameter \ Max Recovery Time Profile	200 ms Profile	30 ms Profile
MRP Test Frames **default** interval	20 ms	3.5ms
MRP Test Frames **short** interval	10 ms	1 ms

The lower the profile selected, based on the ms, the more traffic will sent on the network. The profile changes the timing on the *MRP Test* frames that are sent from the MRM. In order to support the lower profiles reconfiguration times the *MRP Test* frames will be sent more frequently, which will increase the load on the network.

However, as long as the devices that are part of the MRP ring can detect link-down on their own ring ports and they can send *MRP Link Change* frames, the reconfiguration times will essentially be the same regardless of the selected profile. This is because the MRM will react instantly on the *MRP Link Change* frames and will not need to wait until it detects a loss of *MRP Test* frames.

Configuration

MRP is configured from the ring configuration context in the CLI:

```
example:/#> configure
example:/config/#> ring
example:/config/ring/#> mrp 1
example:/config/ring/mrp-1/#>
```

`[no] enable`

Description

Enable or disable this MRP ring instance.

Options

Option	Description
`no`	Disable the MRP ring instance.

`[no] ports [portP,portS]`

Description

Set ring ports, a.k.a. primary port , secondary port.

In normal operation, the MRP toggle between primary or secondary port and keep secondary blocked. I.e if primary goes down then secondary becomes primary.

Options

Option	Description
`no`	Remove configured ring ports.

`[no] mode [manager,client]`

Description

Set the mode of the MRP ring instance.

continues on next page

continued

`[no] mode [manager,client]`

Default: manager.

Options

Option	Description
`no`	Reset to the default.
`manager`	Manager (MRM): Media Redundancy Manager.
`client`	Client (MRC) : Media Redundancy Client.

`[no] profile [200,30]`

Description

Set MRP ring recovery profile mode.

The following failover times are supported:

- 200 msec
- 30 msec

Default: 200

Options

Option	Description
`no`	Reset to the default.

`[no] vid [1-4094]`

Description

Enable/disable VLAN encapsulation of MRP signalling with given VLAN id.

Default: (0) Disabled.

Options

Option	Description
`no`	Disable VLAN encapsulation.

Manager Specific Settings

The following settings are only configurable on the Media Redundancy Manager (MRM).

```
example:/config/ring/mrp-1/#> mode manager
```

`[no] link-react`

Description

Enable/disable React on link change.

continues on next page

continued

`[no] link-react`

When enabled the manager will respond directly on a link change detected by clients.

Default: Enabled.

Options

Option	Description
`no`	Disable react on link change.

Status Overview

The status of MRP can be observed in both a summary view and a more detailed MRP instance view. The summary view is intended to provide a more detailed view of multiple rings, when support for multiple rings have been added.

Summary View

In order to get a current status summary overview of MRP, use the following command:

```
example:/#> show mrp
MRP Enabled : Yes, running as PID 2233
RID  MODE  PROFILE  STATUS  T-COUNT  PORT 1              PORT 2
1    MRM   200      Closed  1        eth1 FORWARDING     eth2 BLOCKED
```

Instance View

In order to get a more detailed status overview for an individual instance, use the following command:

```
example:/#> show mrp 1
Ring ID              : 1
Enabled              : Yes running as PID 2233
Mode                 : MRM
Status               : Closed (CHK_RC)
Profile              : 200
Port 1               : eth1 FORWARDING (UP)
Port 2               : eth2 BLOCKED (UP)
VLAN ID              : 0
React on link change : Enabled
Transitions          : 1
Last topology change : 0 Days 0 Hours 16 Mins 31 Secs
MRM Src Addr         : 74:69:6e:67:20:6d
```

2.4.9. Ring Coupling

Westermo's third generation Ring Coupling Protocol (RiCo), which is further development of RiCo version 2. RiCo enables multiple connections between network topologies to create further redundancy. The networks are connected at layer-2, and the RiCo protocol ensures loop-free connectivity by letting one of the connections forward data, while the redundant coupling links are blocked.

> **NOTE**
>
> RiCo version 3 is not backwards compatible with version 2. RiCo is currently limited to only FRNTv2 as its redundancy domain protocol.

This page will only explain functionality and configuration of RiCo version 3. When only RiCo is written in this page it mean RiCo version 3.

Introduction

The RiCo protocol manages connections between redundancy domains. The connections that interlink redundancy domains are referred to as coupling links and require RiCo configuration at both ends. The RiCo nodes will advertise their best coupling link within their respective domains. Each RiCo node will build a list of all the coupling links in the domain. Among these links, the one with the lowest priority, best priority, will be elected as the Master link, while the others will serve as backup links. The Master link is responsible for forwarding traffic, while the backup links are designed to block traffic to prevent loops, as shown in Figure 1.

```
.-----.    .-----.            .-----.    .-----.
|     |    |     |   Master   |     |    |     |
| SW1 +----+ SW2 +------------+ SW5 +----+ SW6 |
|     |    |     |            |     |    |     |
'--+--'    '--+--'            '--+--'    '--+--'
   |          |                  |          |
   |  FRNTv2  |                  |  FRNTv2  |
   |          |                  |          |
.--+--.    .--+--.            .--+--.    .--+--.
|     |    |     |   Backup   |     |    |     |
| SW3 +----+ SW4 +-||------||-+ SW7 +----+ SW8 |
|     |    |     |            |     |    |     |
'-----'    '-----'            '-----'    '-----'
```

Figure 1: Two FRNTv2 ring connected with RiCo version 3.

The RiCo protocol is a Ring Coupling protocol and connects rings, not individual switches, which can be a misunderstanding. Each ring needs to provide full connectivity by itself and cannot rely on the coupling links for this. FRNTv2 can provide full connectivity in the ring with one fault, but with more than one fault in the ring, RiCo nodes may be isolated even though the switch could be reached over the coupling link. See Figure 2

```
.-----.    .-----.            .-----.    .-----.
|     |    |     |   Backup   |     |    |     |
| SW1 +--| + SW2 +---------||-+ SW5 +----+ SW6 |
|     |    |     |            |     |    |     |
'--+--'    '--+--'            '--+--'    '--+--'
   |         ---                |          |
   |  FRNTv2  |                  |  FRNTv2  |
   |          |                  |          |
.--+--.    .--+--.            .--+--.    .--+--.
|     |    |     |   Master   |     |    |     |
| SW3 +----+ SW4 +------------+ SW7 +----+ SW8 |
|     |    |     |            |     |    |     |
'-----'    '-----'            '-----'    '-----'
```

Figure 2: Sw2 becomes isolated because of two breaks in left ring.

Fail-Over Handling

Once a RiCo node detects a coupling link failure it will inform the other RiCo nodes in the same domain. The RiCo node with the new best coupling link priority will now take over the Master status and open its blocking port.

```
.-----.    .-----.           .-----.    .-----.
|     |    |     |           |     |    |     |
| SW1 +----+ SW2 +-----xx----+ SW5 +----+ SW6 |
|     |    |     |           |     |    |     |
'--+--'    '--+--'           '--+--'    '--+--'
   |          |                 |          |
   |  FRNTv2  |                 |  FRNTv2  |
   |          |                 |          |
.--+--.    .--+--.           .--+--.    .--+--.
|     |    |     |   Master  |     |    |     |
| SW3 +----+ SW4 +-----------+ SW7 +----+ SW8 |
|     |    |     |           |     |    |     |
'-----'    '-----'           '-----'    '-----'
```

Figure 3: Upon a link break the backup link will become Master and put its blocked port in forwarding state.

When the broken link comes back up again it has the best link priority and SW2 and SW5 will advertise this link priority in their redundancy domain. The Master switches SW4 and SW7 will block their ports and advertise that they are in backup state. SW2 and SW5 can now change its port states to forwarding and take over as Master.

To achieve faster failover on Gigabit copper links, fastlink should be enabled on the ports, see Fastlink for more information.

Qualification of coupling links

A coupling link must first be qualified before it can be used as an active link in the election process between RiCo nodes. To qualify a coupling link both sides must agree on the link priority. When the link is qualified it will be advertised in each redundancy domain. The priority vector is constructed of priority:mac-address:port-id. The priority part can be changed to increase the probability of being elected as the Master coupling link.
If default priority is left unchanged or configured to the same value on all switches the switch with the lowest mac-address will have the best link priority.

> **NOTE**
>
> It is not supported to use media converters on the coupling link.

Multiple RiCo instances

Multiple RiCo instances can be configured in the same redundancy domain. In figure Figure 4 FRNTv2 rings are connected together in a ladder structure. Each ring have two RiCo instances configured, one to the left and one to the right with different instance id. The RiCo instance id for each direction of the connecting coupling links do not have to match. However, only RiCo coupling-links with the same Instance id will perform the Master election within its domain.

```
      .-----.  .-----.        .-----.  .-----.
    --+ Sw1 +--+ Sw2 +--------+ Sw5 +--+ Sw6 +--
      |     |  |     |        |     |  |     |
RiCo '+----'  '----+'  RiCo  '+----'  '----+' RiCo
 id:2 |   FRNTv2   | 1 :id: 2 |   FRNTv2   | 3:id
      |            |          |            |
      .+----.  .----+.        .+----.  .----+.
      | Sw3 |  | Sw4 |        | Sw7 |  | Sw8 |
  -||+     +--+     +||----||+     +--+     +||-
      '-----'  '-----'        '-----'  '-----'
```

Figure 4: Examples of RiCo topologies

```
Sw6:/#> configure
Sw6:/config/#> ring
Sw6:/config/ring/#> frntv2 1
Sw6:/config/ring/frnt-1/#> port eth1
Sw6:/config/ring/frnt-1/port-eth1/#> end
Sw6:/config/ring/frnt-1/#> port eth2
Sw6:/config/ring/frnt-1/port-eth2/#> end
Sw6:/config/ring/frnt-1/#> end
Sw6:/config/ring/#>
Sw6:/config/ring/#> coupling 3
Sw6:/config/ring/coupling-3/#> coupling-port eth4
Sw6:/config/ring/coupling-3/coupling-port-eth4/#> priority 100
Sw6:/config/ring/coupling-3/coupling-port-eth4/#> end
Sw6:/config/ring/coupling-3#> leave
Sw6:/#> show coupling

Coupling instance id 3
Advertisement time   : 1000 ms
Domain status        : OK
Coupling ports       : eth4
Redundancy protocol  : FRNTv2
Advertisement ports  : eth1 eth2
═══════════════════════════════════════════════════════════════════════════
Coupling links
Link id                Priority   State        Port     Preempt   Transitions
* 00-11-b4-13-61-c0    100        Master       eth4     Yes       4
  00-07-7c-7c-c8-20    200        Backup       Global   Yes       N/A

* The elected coupling link
Sw6:/#>
```

Configuration

RiCo is configured from the ring configuration context in the CLI:

```
example:/#> configure
example:/config/#> ring
example:/config/ring/#> coupling 1
example:/config/ring/coupling-1/#>
```

`[no] coupling [ID]`

Description

Enable/Disable RiCo and enter RiCo configuration context of given instance.

continues on next page

continued

`[no] coupling [ID]`

Options

Option	Description
`ID`	RiCo instance ID (range 1-255). Election of coupling links is performed on links in the sames instance. Default is '1'
`no`	Remove the given RiCo instance.

Syntax RiCo Configuration

`[no] adv-interval <INTERVAL>`

Description

Set RiCo advertisement interval. Default 1000 ms.
The RiCo protocol is event based and all RiCo nodes will be updated on link change.

Options

Option	Description
`no`	Reset to default (1000 ms)
`INTERVAL`	Interval in milliseconds. Range 50-10000 (ms)

> **NOTE**
>
> Changing this interval will not improve fail over times so only change this if you are advised to.

`[no] nodes <NUMBER>`

Description

The number of RiCo nodes in the redundancy domain with the same instance number. **Nodes** must be configured to show **Domain status**, see RiCo Status.

This parameter is optional and is used to show summary status for the RiCo coupling domain.

Options

Option	Description
`no`	Set to unconfigured.
`NUMBER`	How many RiCo switches exists in this instance in the redundancy domain.

`[no|show] coupling-port <PORT,PORT..PORT>`

Description

Set PORT as coupling port and enter configuration context for the given port.

continues on next page

continued

`[no|show] coupling-port <PORT,PORT..PORT>`

Options

Option	Description
`PORT`	Name of coupling port, typically an Ethernet port such as *eth1*.

`[no] priority <PRIO>`

Description

Setting within *coupling-port configuration* context. Default: 16384
Set coupling link priority, where a low value means better priority, which increase the probability of being elected as Master coupling link.

Options

Option	Description
`no`	Use default priority 16384.
`PRIO`	An integer to set the priority in range 1-32766.

`[no] preempt`

Description

Setting within *coupling-port configuration* context. Default: Enabled
When preempt is enabled the priority of the link will decide which link will be elected as the Master link. If preempt is disabled the link will get a sticky behaviour. After it has been elected as a Master link it will continue to be the Master link until the link is down.

Options

Option	Description
`no`	Disable preempt and the coupling link will get a sticky behaviour.

If the configuration of preempt differs on each side of the link 'no preempt' will have precedence.

`[no] hello-interval <INTERVAL>`

Description

Setting within *coupling-port configuration* context. Default: 1000 ms
RiCo hello packet interval on the coupling ports.

Options

Option	Description
`no`	Use default interval value 1000 ms.
`INTERVAL`	Interval in range 50-5000 ms.

continues on next page

continued

`[no] hello-interval <INTERVAL>`

> **NOTE**
>
> Changing this interval will not improve fail over times so only change this if you are advised to.

RiCo Status Commands and Explanations

In order to see the current status of RiCo for an individual device the following command can be used:

`show coupling [ID]`

Description

Show RiCo status and statistics.

Options

Option	Description
`ID`	Get status for the specified instance

Showing RiCo configuration and status of the link qualification and election.

```
example:/#> show coupling

Coupling instance id 1
Advertisement time  : 1000 ms
Domain status       : OK
Coupling ports      : eth5
Redundancy protocol : FRNTv2
Advertisement ports : eth3 eth4
================================================================================
Coupling links
Link id               Priority   State        Port     Preempt   Transitions
  00-11-b4-39-8b-c0   16384      Backup       eth5     Yes       3
* 00-07-7c-00-14-50   16384      Master       Global   Yes       N/A

* The elected coupling link
example:/#>
```

- **Advertisement time**: The configured value for advertisement time.
- **Domain status**: To show this status the number of **nodes** in this RiCo instance domain must be configured.
  - *OK*: At least the configured number of **nodes** in the instance domain have link up, as backup or master.
  - *Degraded*: At least one or more coupling links are up in the instance redundancy domain.
  - *Broken*: All RiCo coupling links in this instance redundancy domain are down.
  - *N/A*: **Nodes** is not configured.
- **Coupling ports**: Configured coupling ports.

- **Redundancy protocol**: Which ring protocol RiCov3 is running on. The only supported ring protocol is FRNTv2.
- **Advertisement ports**: On which ports the advertisement packets are sent on.

Coupling links:

- **Link id:** This link id is the mac address derived from qualification process.
- **Priority**: The configured priority of the coupling link used in the link election.
- **RiCo State:**
  - *Down*: Link down on port.
  - *Backup*: Port is up and in blocking state. A port with better priority exists.
  - *Master*: This port has the best priority in the instance and are forwarding traffic.
    Only one Master port can exist in an instance.
- **Port:**
  - *port name*: When it is a port name such as *eth1* This is a local link on this switch.
  - *Global*: This link is an advertised link from another switch in the instance.
- **Preempt:**
  On link down the configuration of preempt is shown.
  When a link is elected as Master:
  - *No*: This link has preempt disabled and will remain Master until it gets link down.
  - *Yes*: If this link is Master the priority was used in the selection.
    When the link is in backup state it will always show Yes.
- **Transitions:** The number of link changes, only shown for local links.

2.4.10. Ring Chaining

Glossary

- **UUID** - Universally Unique IDentifier[1]

Introduction

In short, the Ring Chaining (RiCh) protocol is designed to construct a chain of networks, and to distribute the information about that chain, across all of the constituent networks. The protocol is designed to work in a “chain” topology, most commonly achieved by connecting multiple ring networks into a one-dimensional chain. The chain is constituted by an ordered list of networks (termed rings, see below), where each ring has an orientation.

Overview

RiCh is a distributed protocol running on at least one coupling node in a ring network as well as an extension of RiCo. A coupling node is part of the ring and has a ring coupling that can connect to a coupling node in another ring.

Ring couplings

RiCh rings are built on FRNTv2 rings that are connected using the ring coupling protocol RiCov3. Each end-node runs two RiCo instances that are used to connect to other rings and to distribute connection information within the ring.

Discovery of topology

With RiCh, nodes will discover other nodes present on the network and exchange information. This RiCh signalling in turn allows each node to gather information about:

- **UUID:** The identity of present ring(Universally Unique Identifier, or simply UUID)
- **CRC:** A checksum calculated over the chain of rings that a node has collected
- **Inauguration state:** indicates if all ring information collected have the same CRC value

Redundancy

RiCh (mode active) has to run on at least one node in the ring. Additional RiCh instances (mode active) can be configured to run on other node in the network to work as *backup* instances. At start up, all active instances will send out topology messages. The node with the numerically smallest IP will be selected as *master* to continue sending topology messages. The remaining nodes will act as *backup* nodes. The *backup* nodes stop sending topology messages but continue to collect the incoming message and build the connectivity table so that they are prepared to quickly takeover as *master* in case the master node fails.

```
        .-------.    .-------.
        |       |    |       |
<-dir1--+ RICH1 +----+ RICO1 +--dir2->
        |       |    |       |
        '---+---'    '---+---'
            |            |
        .---+---.    .---+---.
        |       |    |       |
<-dir1--+ RICO2 +----+ RICH2 +--dir2->
        |       |    |       |
        '-------'    '-------'
```

Figure 1: A ring with four nodes. All nodes are running FRNTv2 and RiCov3. In this example there is a redundant setup of active RiCh running on the nodes RICH1 and RICH2.

[1] See RFC 4122

Configuration

RiCh can be activated from the top-level configuration context in the CLI.

```
example:/#> configure
example:/config/#> ring chaining
Activating RiCh with default settings.
example:/config/ring/chaining/#>
```

> **NOTE**
>
> RiCh will be enabled by default when entering the configuration context.

For configuration examples, see
- Single ring with two nodes
- Single ring with six nodes

Syntax

`[no] enable`

Description

Enable RiCh.

> **NOTE**
>
> Simply entering the RiCh config context will enable RiCh when leaving the RiCh context or config context.

Options

Option	Description
`no`	Disable RiCh. The current configuration is saved, for when you re-enable the service again. **NOTE** The RiCh configuration can be completely removed from the **ring config context** using **no chaining**.

`[no] uuid <UUID-STRING>`

Description

Set the ring UUID. Each node belonging to the same ring must use the exact same UUID, and no two rings may have the same UUIDs. Thus, the UUID is what defines individual rings.

continues on next page

continued

`[no] uuid <UUID-STRING>`

Options

Option	Description
`UUID-STRING`	A literal UUID string, in the RFC 4122 format. `example:/config/ring/chaining/#> uuid 11111111-1111-1111-1111-111111111111`
`no`	Reset UUID to all zeros.

`[no] iface <INTERFACE>`

Description

Specify the interface on which RiCh should distribute topology messages.

Options

Option	Description
`INTERFACE`	Name of the interface that RiCh will send messages on.
`no`	Reset to default (not set).

`[no] address <ADDRESS>`

Description

Specify the multicast group address on which RiCh should distribute topology messages.

Options

Option	Description
`ADDRESS`	Multicast group address that RiCh will distribute topology messages on.
`no`	Reset to default (not set).

`[no] trigger <always|master>`

Description

Specify under which conditions RiCh should trigger an SNMP trap at coupling/decoupling.

Options

Option	Description
`always`	The RiCh instance will always generate a SNMP trap at coupling/decoupling.
`master`	The RiCh instance will generate a SNMP trap at coupling/decoupling if it is acting as master.
`no`	Reset to default (master).

`[no] mode <active|passive>`

Description

Specify the mode for running RiCh.

Options

Option	Description
`active`	Run the full RiCh features with the ability to become the RiCh master node.
`passive`	Run only the Rico extension for RiCh.
`no`	Reset to default (active).

`[no] dir1 <RICO-ID>`

Description

Specify which RiCo instance is connected to direction 1.

Options

Option	Description
`RICO-ID`	Id of the RiCo instance associated with direction 1.
`no`	Resets to (not set).

`[no] dir2 <RICO-ID>`

Description

Specify which RiCo instance is connected to direction 2.

Options

Option	Description
`RICO-ID`	Id of the RiCo instance associated with direction 2.
`no`	Resets to (not set).

`show`

Description

Show a summary of RiCh settings.

Status

To see the current status of RiCh, run the `show chaining` command in `Admin context`.

```
example:/#> show chaining
RICH NODE
Own UUID      : 11111111-1111-1111-1111-111111111111
Chain state   : Inaugurated
Chain CRC     : 32cca232
```

```
Node state   : Master (sending)

UUID                                    ORIENTATION
22222222-2222-2222-2222-222222222222  DIFFERENT
11111111-1111-1111-1111-111111111111  DIFFERENT
33333333-3333-3333-3333-333333333333  SAME

example:/#>
```

2.4.11. High-Availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP)

High-Availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP), defined in IEC 62439-3, provides seamless failover on link failure. Both build on the same idea of duplicating traffic and can be coupled with each other.

The purpose of this document is to cover the basics of HSR/PRP.

For example use-cases, see:
- HSR and PRP configuration guide

Introduction

The general idea of HSR and PRP is duplicating traffic and sending over different paths in the network. At the point where they meet up again, only the first frame to arrive will be forwarded. A duplicate discard algorithm eliminates the second copy to ensure no frame is delivered twice.

Each frame is uniquely identified by a combination of a sequence number (a 16-bit unsigned integer) and its source MAC address. The sequence number wraps around after reaching the maximum value. To avoid identifying different frames as the same, **EntryForgetTime** specifies how long (milliseconds) it remembers the identity of a frame. When the entry expires, a new frame with the same identity counts as a new frame, not as a duplicate of the earlier one.

In the event of a link failure, one copy will still reach its destination.

Within HSR and PRP networks different types of devices are called by specific names according to how they can be attached to a network.

Doubly Attached Nodes or **DANs** are called as such as they are devices that support HSR/PRP through at least two Ethernet connections. Usually DANs are referred to by the protocol they run, **Doubly Attached Node HSR (DANH)** or **Doubly Attached Node PRP (DANP)**.

Singly Attached Nodes or **SANs** are devices that are restricted to a single Ethernet connection. In this case, a RedBox can connect the SAN to an HSR/PRP network, allowing the SAN to appear as a DANH/DANP to the rest of the network.

Redundancy Box (RedBox)

A RedBox has at least 3 Ethernet ports. Two of these are assigned as redundancy ports, called port A and port B, that create and discard duplicate traffic. The third port is the interlink. It is possible to have multiple ports as interlinks provided they are all configured the same. The redundant ports operate either in HSR mode or PRP mode. The interlink operates in one of three modes: SAN, HSR, or PRP. The modes a RedBox as a whole can support are HSR-SAN, PRP-SAN, HSR-PRP, and HSR-HSR.
- HSR-SAN: Connects a SAN to an HSR ring.
- PRP-SAN: Connects a SAN to two PRP LANs.
- HSR-PRP: Connects a PRP LAN to an HSR ring.
- HSR-HSR: Connects another HSR network to an HSR ring (can be used to make a QuadBox)

High-Availability Seamless Redundancy (HSR)

HSR is a ring protocol. All network traffic in the ring has an HSR tag with the Ethertype `0x892f`. If a frame contains a VLAN tag, then the HSR tag comes after the VLAN tag.

The HSR tag contains:
- 4-bit path identifier (**PathId**)
- 12-bit frame size (**LSDUsize**)

- 16-bit sequence number

The device injecting a frame into the ring gives it an HSR tag with a sequence number based on the source MAC and sends the frame to both ring ports. Upon reaching a DANH that's the unique destination (unicast), or a RedBox with the destination behind it, it stops traversing the ring. Multicast and broadcast always traverse the entire ring, and the device that initially injected the frame into the ring discards it after a full lap.

The **PathId** and **LSDUsize** fields aren't relevant in the context of a setup with only HSR. **PathId** is 0. **LSDUsize** is the size of the frame, excluding the DMAC, SMAC, and VLAN tags. They only become relevant in PRP and when coupling HSR with PRP.

Figure 1 shows a minimal HSR setup.

```
                +---------+
                |         |
                |   SAN   |
                |         |
                +----+----+
                     |
                     |
                +----I----+
                |         |
                | RedBox  |
        +-----A HSR-SAN B------+
        |       |         |      |
        |       +---------+      |
        |                        |
        |                        |
        |       +---------+      |
        |       |         |      |
        +-----A RedBox  B------+
                | HSR-SAN |
                |         |
                +----I----+
                     |
                     |
                +----+----+
                |         |
                |   SAN   |
                |         |
                +---------+
```

Figure 1: Two SAN devices connected redundantly via two HSR-SAN RedBoxes.

Parallel Redundancy Protocol (PRP)

Unlike HSR, PRP isn't a ring protocol. Instead, it duplicates the traffic onto two different LANs, named LAN A and LAN B. These two LANs must match up with the A and B ports of the DANP/RedBox on each side. The PRP tag looks similar to the HSR tag but in the place of **PathId** there is instead the 4-bit field **LanId**. It has the value `0xA` for the frame sent on LAN A, and `0xB` for LAN B. A PRP device accepts frames even if the **LanId** field doesn't match with the receiving port, but it will increment an error counter. It also accepts frames without a PRP tag, allowing for many other SANs within each LAN.

Instead of a header like HSR, the PRP tag is a trailer on the frame, placed right before the Ethernet checksum. To not confuse the payload of a non-PRP frame with the trailer the **LSDUsize** acts as an extra checksum to verify that it's a PRP frame. There may still be cases of a correctly formed PRP tag appearing in the payload of a frame, but the checksum makes it less likely. For this reason, frames are still delivered even if the **LanId** doesn't match the port. An error counter is incremented to indicate that it happened. The A and B ports have no significance in an HSR ring.

The topology in Figure 1 can also be used for PRP-SAN mode, assuming both A ports are connected to each other (LAN A), and same for B ports.

HSR-PRP coupling

The mode HSR-PRP connects a PRP LAN to an HSR ring. The interlink port connects to the PRP LAN and is set to A or B depending on which LAN it is. Along with the **LanId**, it has a **NetId** to indicate which PRP network it came from. Together these two fields form the **PathId** field of the HSR tag. The least significant bit is the **LanId** and the remaining three bits are the **NetId**. The value of **NetId** must be in the range 1-6. A RedBox receiving a PRP frame on the interlink forwards it to the HSR ring and sets the **PathId** field based on the settings on the interlink port to indicate where it came from. This prevents reinjection into the other PRP LAN.

Supervision frames and ring status

Every HSR/PRP node sends out supervision frames (Ethertype `0x88fb`) at **LifeCheckInterval** intervals. These frames serve two purposes.

1. Ensuring that the redundancy is ok by checking that it gets the frames on both ports.
2. For RedBoxes, inform other nodes what nodes it has connected on the interlink.

Alarms

HSR/PRP alarms trigger when the redundancy is broken. Configure alarms in the alarm context as a **ring** trigger. See Alarm. If the devices stops seeing supervision traffic from a node on both links, the alarm triggers.

HSR/PRP with PTP

Precision Time Protocol (PTP) will behave differently when used in conjunction with HSR/PTP.

PTP messages are still duplicated over HSR/PRP networks, except peer delay messages. They are link-local and use the destination MAC address `01:80:C2:00:00:0E`. They measure the time delay to the link partner to allow compensation for ingress, egress, and cable delay.

Despite this, they still carry an HSR tag. All PTP synchronisation messages are defined to have the same length, as to not introduce differences in travel time due to frame length. Since the other PTP messages need the HSR tag to traverse the ring it makes sense that peer delays also have it.

Unlike normal HSR/PRP traffic, the PTP receivers do not use the first copy to arrive. It selects one port to listen on, and swaps over to the other when messages cease to arrive on the selected side. Always taking the first to arrive may result in additional jitter, as the residence time will vary between the two paths.

The redundancy ports' PTP port states are coupled. For a TC they are either both MASTER, transmitting Sync information received on the interlink, or they are SLAVE/PASSIVE_SLAVE (one each), forwarding Sync information from the SLAVE port to interlink. In an HSR ring, regardless of port state and mode (TC/BC), all Sync messages are forwarded within the ring according to TC operation and are discarded by the device that injected it.

Syntax HSR/PRP Configuration

`type <hsr|prp|hsr-prp>`

Description

Select type of HSR/PRP instance (HSR-SAN, PRP-SAN, HSR-PRP). Must be set before any other options.

`[no] enable`

Description

Enable/Disable the instance.

`[no] supervision-frame`

Description

Enable/Disable supervision frames

`mode <H|N>`

Description

Select the operating mode for HSR. Only applicable for `type hsr`.

Options

Option	Description
`H`	[HSR-tagged mode] default operational mode where the unit inserts HSR tag and forwards traffic.
`N`	[No forwarding] operational mode, just like mode H, except that the unit does not forward traffic from port to port. Intended for testing purposes only.

`prp-id <1..6>`

Description

NetId of the PRP network attached to the interlink. Only applicable for `type hsr-prp`.

RedBoxes connected to the same PRP network should have the same NetId. The HSR tag carries this value, which prevents frames from LAN A from being reinjected into its corresponding LAN B, and vice versa.

`lan-id <A|B>`

Description

LanId of the PRP network attached to the interlink. Only applicable for `type hsr-prp`.

The HSR tag of frames originating from the interlink contains which LAN it is.

When both HSR/PRP and PTP are configured, PTP automatically assumes HSR/PRP mode for the redundancy ports.

HSR/PRP Status Commands and Explanations

```
redbox:/#> show hsr-prp
HSR instance id 1
Enabled                      : YES
Type                         : HSR
HSR mode                     : H
```

```
Port A                      : ethA
Port B                      : ethB
Supervision frames          : Enabled
Life check interval         : 2 s
Entry forget time           : 6 (640 ms)
Node reboot interval        : 500 ms
Node forget time            : 64 s
Proxy node table forget time: 64 s
Multicast octet             : 00
Redundancy status           : Ok
CntError A                  : 0
CntError B                  : 0
CntReceived A               : 0
CntReceived B               : 0
CntSent A                   : 0
CntSent B                   : 0
CntWrongLan A               : 0
CntWrongLan B               : 0
CntDuplicate A              : 0
CntDuplicate B              : 0

STATUS
Node Table          Type      Age (s):  A  |  B  |  I
50:00:84:65:9d:2a   REDBOXH             0  |  0  |  -
00:11:b4:53:2f:80   REDBOXP             1  |  1  |  1
38:f3:ab:96:ec:12   VDANP               1  |  1  |  1
00:07:7c:6e:34:ca   VDANP               1  |  1  |  1
00:11:b4:53:2d:00   REDBOXP             0  |  0  |  -
00:07:7c:6e:34:c8   VDANP               0  |  0  |  -
Proxy Nodes
22:11:b4:02:03:01
00:07:7c:6e:34:c7
00:07:7c:6e:34:c0
```

This view contains the settings and statistics of the HSR/PRP instance.

`Redundancy status` indicates if the redundancy is working or not. Possible output values are `Ok` and `Broken`.

`CntError` represents frames dropped due to, for example, bad Ethernet checksum. `CntReceived` is the amount of valid frames received. `CntSent` is the amount of frames sent. `CntWrongLan` is only valid for PRP and HSR-PRP and shows the number of frames received that contain the wrong LAN in the PRP tag. If the network is misconfigured this counter will rapidly increase. Occasional increases could mean a non-PRP frame where the payload happened to look like a PRP tag. `CntDuplicate` is the amount of frames detected as copies of frames received earlier.

`Proxy Nodes` shows the nodes present on the interlink port. It exists for advertising to other nodes which ones are reachable through the HSR/PRP network. In HSR-PRP mode, the Proxy Nodes list only contain nodes that are not advertised through supervision frames (i.e. if a node is in the Node Table it will not also show up as a proxy node).

`Node Table` shows the nodes advertised by other RedBoxes/DANH/DANP in the network through HSR/PRP supervision frames. The proxy nodes of other devices appears in this list.

2.5. IP Host Services

2.5.1. Interface

Introduction

A network interface handles configuration of the network layer in a computer network. It can be a representation of a hardware unit but it also be a pure software construction in case of a SSL interface or a loopback interface. It typically has a network address.

> **SEE ALSO**
>
> Related Examples and Use-Cases
> - IPv6 Basic

Overview

In its simplest case a interface is a higher layer representation of a network port in a 1-1 mapping. A more complex example is a VLAN interface that can contain several network ports. It handles the network layer which means it typically has a network address such as an IPV4 address. It also resolves mapping between network addresses and MAC addresses through ARP functionality.

For the image below two hosts are connected via ports eth1 on each unit. Host-1 have configured interface vlan1 with address 192.168.0.10/24 and Host-2 have configured vlan2 with address 192.168.0.20/24.

```
 Host-1                                          Host-2
.-------.                                       .-------.
|       |                                       |       |
|       |                                       |       |
|   1   |vlan1                            vlan2|   1   |
'---+---'                                       '---+---'
    |.10                192.168.0.0/24            .20|
    +----------------------------------------------+
```

Configuration

Configuration of network interfaces is done in the iface context. Note that there are several configurable options which may not be relevant for each interface type. A custom MAC address and configuration options with respect to ARP are only relevant for interfaces which have a MAC address (LAN interfaces).

Example of how to configure interface vlan1:

```
example:/#> configure
example:/config/#> iface vlan1
example:/config/iface-vlan1/#>
```

On VLAN interfaces, it is possible to circumvent the link status propagation property by configuring an interface as always up:

```
example:/config/iface-vlan1/#> enable always
```

However, disabling link status propagation may significantly impact layer-3 protocols such as RIP, OSPF, VRRP, and more. The protocols will have to fall-back to other methods to detect link-down, e.g. hello message timeout and similar.

Do not use the ***enable always*** setting unless you really know what you are doing.

Syntax

`[no|show] inet [static IP/LEN|dhcp|ssl|link-local|ttdp]`

Description

This setting controls the handling of network addresses. It can be set and/or acquired in several ways.

> **NOTE**
>
> Enters a sub-configuration context for any inet specific settings.
>
> See Inet DHCP Configuration below for more details on DHCP client configuration.

Options

Option	Description
`no`	No network address will be set
`static IP/LEN`	Configure a static [IP] address with network mask with length [LEN]
`dhcp`	Acquire a network address through the DHCP protocol
`ssl`	Acquire a network address through the SSL protocol
`link-local`	Acquire a network address through the link-local protocol
`ttdp`	Acquire a network address through the TTDP protocol

`[no|show] inet6 [static IP/LEN [eui-64] | link-local]`

Description

This settings controls how IPv6 addresses can be configured on the interface. Currently only static and link-local address methods are supported for IPv6 address configuration.

> **NOTE**
>
> For the IPv6 address to be added to the interface, global IPv6 functionality must be enabled on the device. This is done by enabling the global IPv6 functionality in the IPv6 configuration context. The below example is how it is done in the CLI:
>
> ```
> example:/#> configure
> example:/config/#> ipv6
> example:/config/ipv6/#> enable
> ```

> **NOTE**
>
> Enters a sub-configuration context for any inet6 settings that may be specific to the selected method.

continues on next page

continued

`[no|show] inet6 [static IP/LEN [eui-64] | link-local]`

Examples

Configuring a static IPv6 address:

```
example:/config/iface-vlan1/#> inet6 static 2001:db8::1234:5678/64
```

Configuring a link-local IPv6 address:

```
example:/config/iface-vlan1/#> inet6 link-local
```

Options

Option	Description
`no`	Remove configured IPv6 address method. If no specific inet6 method is specified, all configured IPv6 address methods will be removed.
`static IP/LEN [eui-64]`	Configure a static `IP` address with network mask with length `LEN`. If no `LEN` is provided, the default length of **/64** will be used. The optional `eui-64` keyword can be used to enable EUI-64 address generation for the interface. The eui-64 address generation is based on the MAC address of the interface. The MAC address is split in half, and the 7th bit is flipped. The resulting half MAC address is then inserted in the middle of the IPv6 address. The initial part of the IPv6 address is taken from the provided static IP address. As en example, if the initial part of the provided IPv6 address is **2001:db8:8888:9999** and the MAC address of the interface is **00:11:22:33:44:55**, the resulting IPv6 address will be **2001:db8:8888:9999:211:22ff:fe33:4455/64**. This is how it would be configured in the CLI: `example:/config/iface-vlan1/#> inet6 static 2001:db8:8888:9999::/64 eui-64`
`link-local`	Specify that the interface should acquire a link-local IPv6 address. The link-local address is based on the MAC address of the interface. If the MAC address of the interface is **00:11:22:33:44:55**, the resulting link-local address will be **fe80::211:22ff:fe33:4455/64**.

continues on next page

continued

`[no|show] inet6 [static IP/LEN [eui-64] | link-local]`

Option	Description
	NOTE Regardless of the selected IPv6 address method, the link-local address will always be enabled, even if it is not explicitly configured. The reason for this is that the link-local address is always required for IPv6 communication.

`[no] enable [always]`

Description

This setting controls the administrative state for the interface, if it should be UP or DOWN.

> **NOTE**
>
> If the interface is a VLAN interface and it is disabled, the CPU connection to that VLAN is also completely severed. Only layer-2 switching will work on this VLAN, while all layer-3 services such as management, routing, firewall, etc. will be disabled.
>
> If the intention is to disable the entire VLAN itself, that action is performed in the VLAN configuration context.

> **WARNING**
>
> Disabling an interface may cause loss of connectivity to the device, which may require physical access to the device to restore connectivity. Be mindful of which interface, you may potentially be accessing the device over.

Options

Option	Description
`no`	Administratively disable the interface.
`always`	When set, the interface will remain up even if the link is down. **NOTE** This setting only available for VLAN interfaces.

`[no] shutdown`

Description

Shutdown the interface, same as “no up”

continues on next page

continued

`[no] shutdown`

Options

Option	Description
`no`	Enable interface

`[no] mac [MAC]`

Description

Change or restore MAC address of interface.

Options

Option	Description
`no`	Use default MAC address

`[no] mtu [68-1500]`

Description

Set maximum transfer unit, MTU, of interface.

Options

Option	Description
`no`	Use default MTU size

`[no] redirect`

Description

Enable sending of ICMP redirect messages.

Options

Option	Description
`no`	Disable sending of ICMP redirect messages

`[no] proxy-arp`

Description

Enable or disable Proxy ARP on this interface.

When enabled the system will respond to ARP requests on behalf of other devices. The system will answer ARP requests for IP addresses that are not on this interface's subnet, provided it has a route to reach those addresses via another interface.

The system responds with its own MAC address, allowing it to act as a proxy for forwarding traffic between different subnets or networks.

continues on next page

continued

`[no] proxy-arp`

Default: Disabled

Options

Option	Description
`no`	Disable proxy ARP on this interface

`[no] arp-notify`

Description

Enable or disable sending gratuitous ARP on this interface.

When enabled the system will send gratuitous ARP frames when:

- The interface is administratively brought up, i.e. it is enabled in the configuration.
- The interface is operationally brought up.
- The MAC address of the interface is changed.
- The IP address of the interface is changed.

Default: Enabled

Options

Option	Description
`no`	Disable gratuitous ARP on this interface

`[no] arp-accept`

Description

Enable or disable accepting gratuitous ARP on this interface.

When enabled the system will accept gratuitous ARP packets. If the ARP frame contain an IP address that is not present in the ARP table, the system will add the IP address to the ARP table. If the ARP frame contains an IP address that is already present in the ARP table, the ARP table will be updated with the new MAC address.

Default: Enabled

Options

Option	Description
`no`	Disable receiving gratuitous ARP on this interface

`[no] arp-watch`

Description

Enable ARP watch on this interface.

continues on next page

continued

`[no] arp-watch`

When enabled the system will monitor ARP activity on this interface. These ARP events can be configured to be reported via:

- **audit**: Log ARP events to the system audit log.
- **snmp**: Generate SNMP Notifications (Traps/Informs) for ARP events.

The following ARP events can be configured to be reported when/if they are detected on this interface:

- **new-station**: The Ethernet address has not been seen before.
- **flip-flop**: The Ethernet address changed from the most recently seen address to the second most recently seen address.
- **changed**: The host switched to a new Ethernet address.
- **ethernet-broadcast**: The MAC Ethernet address is a broadcast address.
- **ip-broadcast**: The IP address is a broadcast address.
- **bogon**: The source IP address is not local to the local subnet.
- **ethernet-mismatch**: The source MAC Ethernet address didn't match the address inside the ARP packet.
- **reused-address**: The Ethernet address changed from the most recently seen address to the third (or greater) least recently seen address.
- **address-conflict**: Indicates a possible IP address conflict based on MAC address changes observed in ARP traffic. Generated when any conflict related change event occurs (changed, flip-flop, or reused-address), providing a convenient way to detect potential address conflicts. Detection is based on stateless ARP frame observation and represents a potential conflict situation. This event is generated independently and will be reported even if the individual change events are disabled.

When arp-watch is initially enabled, all events and all reporting methods are enabled by default, except **address-conflict**.

> **NOTE**
>
> The `arp-watch` command enters a sub-context that allows for configuration of which ARP events should be reported and how they should be reported.
>
> See ARP Watch Configuration below.

> **NOTE**
>
> For SNMP notifications to be generated, SNMP must be configured to send notifications. Enabling SNMP for ARP watch allows ARP watch events to generate SNMP notifications. However, if SNMP is not configured to send notifications, no notifications will be sent.
>
> See SNMP Configuration for more details on SNMP configuration.

Default: Disabled

continues on next page

continued

`[no] arp-watch`

> **POSSIBLE ROUTING PERFORMANCE IMPACT**
>
> Enabling this setting may have impact on routing performance, since it needs to inspect certain frames on this interface.

Example:

```
example:/config/iface-vlan1/#> arp-watch
example:/config/iface-vlan1/#> no arp-watch
example:/config/iface-vlan1/#> arp-watch report snmp audit
example:/config/iface-vlan1/#> arp-watch no report snmp
example:/config/iface-vlan1/#> arp-watch event bogon
example:/config/iface-vlan1/#> no arp-watch event new-station
```

Options

Option	Description
`no`	Disable ARP watch on this interface

`[no] ip-watch`

Description

Enable IP watch on this interface.

When enabled the system will run IPwatchD tool to detect duplicate MAC/IP addresses on this interface. These events will be logged to the systems audit log. The following events are logged:

- Duplicate IP Address: A remote device has the same IP as this interface.
- Duplicate MAC Address: A remote device has the same MAC as this interface.

Default: Disabled

> **POSSIBLE ROUTING PERFORMANCE IMPACT**
>
> Enabling this setting may have impact on routing performance, since in needs to inspect certain frames on this interface.

Options

Option	Description
`no`	Disable IP watch on this interface

`[no] name`

Description

This setting controls the interface name.

continues on next page

continued

`[no] name`

Options

Option	Description
`no`	Default name used for this interface

`[no] distance [1-255] [track TRIGGER-ID]`

Description

Administrative distance for routes learned on this interface.

Static routes learned dynamically, e.g. via DHCP, will be installed in the routing table with this administrative distance.

Possible values are 1-255, where 1 is the best and 255 is infinity, it will be visible in the routing table but will never be activated.

Use the form 'no distance' to reset the value to its default value, 16. Use 'distance 255' to prevent routes from ever being activated.

A ping trigger can be associated with the interface distance setting. In case of connectivity loss, the distance of the associated route (s) are raised to infinity (255)

Options

Option	Description
`no`	Use default distance for this interface (16)

`show`

Description

Show the summary for the configuration of the interface, an example is shown below.

```
Name          : vlan1
Admin Mode    : Up
MAC Address   : AUTO
IP Addresses  :
  Inet link-local
  Inet static
    192.168.2.10/24
MTU           : AUTO (1500)
ICMP Redirect : Sending
Proxy ARP     : Disabled
ARP Notify    : Enabled
ARP Accept    : Enabled
Distance      : 1
```

Inet DHCP Configuration

DHCP client configuration is performed in a sub-context under the interface configuration:

```
example:/config/iface-vlan1/#> inet dhcp
example:/config/iface-vlan1/inet-dhcp/#>
```

`[no] clientid <hex|string> <STRING>`

Description

Configure the Client ID that is sent in DHCP requests.

Default: Use mac address.

Options

Option	Description
`no`	Remove the configured client-id and use the default value.
`hex`	Indicates that the subsequent value is provided in hexadecimal form. The following syntax forms are valid: `0x1a2b`, `1a:2b`, `1a2b`.
`string`	Indicates that the subsequent value is provided in string format.

`[no] arping`

Description

Enable ARP-ping option in DHCP client.

Options

Option	Description
`no`	Disable ARP-ping option in DHCP client.

Default: Enabled

`[no] option <OPT> [<OPT>...]`

Description

Configure the DHCP Parameter Request List (DHCP option 55) that the client provides in its communication with the server.

The system supports the following options to be included in the DHCP Parameter Request List:

Option Number	Description	Requested by Default
1	Subnet	Yes
3	Router	Yes
6	DNS Server	Yes
7	Log Server	Yes
12	Hostname	Yes
15	Domain	Yes
42	NTP Server	Yes
66	TFTP Server	-

continues on next page

continued

`[no] option <OPT> [<OPT>...]`

Option Number	Description	Requested by Default
67	Bootfile	-
121	Classless Route	Yes
249	Microsoft Classless Route	Yes

> **NOTE**
>
> Only the options that are explicitly requested will be used and applied on the device. In other words, if a DHCP server provides any of these options anyway, when it was not included as part of the client's Parameter Request List, the client will ignore those options.

WeOS handles DHCP options 12 and 15 in a special way. Currently, they are only used to add the hostname (option 12) along with any provided domain name (option 15) to the local DNS resolver configuration, which includes DNS host entries and search domains. They are not used in any other part of the system. To set the system hostname and domain name for other purposes, see the system settings documentation.

DHCP Option 249 is a Microsoft specific variant of option 121. If both 121 and 249 are received as part of a lease, and their contents are identical, only one copy of the relevant routes will be installed on the client. If their content differ, their union will be installed. System administrators should ensure routes advertised by options 121 and 249 are identical.

If DHCP Classless static route option (121 and/or 249) is configured on the WeOS DHCP client, then option 3 should also be configured. If the server provides option 121/249 in its response, it will be used by the client and any response to option 3 will be ignored.

Options

Option	Description
`no`	Use `no option` to reset the option request to the default options. Use `no option OPT` to not request that specific option. **NOTE** If all specific option requests are removed no Parameter Request List will be included as part of the client's requests, i.e. there is no implicit Parameter Request List (option 55) included from the client. In this scenario the DHCP client will apply any option it may support, if it is provided by the lease from the DHCP server.
`OPT`	DHCP option number.

ARP Watch Configuration

ARP watch configuration is performed in a sub-context under the interface configuration:

```
example:/config/iface-vlan1/#> arp-watch
example:/config/iface-vlan1/arp-watch/#>
```

`[no] report [METHOD [ METHOD ...]]`

Description

Select how ARP watch events should be reported. Multiple reporting methods can be specified.

> **NOTE**
>
> If no specific method is specified, it is treated as `all`, which means that all methods are enabled or disabled together depending on whether 'no' is used or not.

Default: All reporting methods are enabled when ARP watch is enabled.

Example:

```
example:/config/iface-vlan1/arp-watch/#> report snmp audit
example:/config/iface-vlan1/arp-watch/#> no report snmp
example:/config/iface-vlan1/arp-watch/#> no report
example:/config/iface-vlan1/arp-watch/#> report all
```

Options

Option	Description
`no`	Disable the specified reporting method(s), or all if none specified.
`METHOD`	The reporting method(s) to enable: • `audit`: Log ARP events to the system audit log. • `snmp`: Generate SNMP Notifications (Traps/Informs) for ARP events. • `all`: Enable all reporting methods. **NOTE** For SNMP notifications to be generated, SNMP must be configured to send notifications. Enabling SNMP for ARP watch allows ARP watch events to generate SNMP notifications. However, if SNMP is not configured to send notifications, no notifications will be sent. See SNMP Configuration for more details on SNMP configuration.

`[no] event [EVENT [ EVENT ...]]`

Description

Configure which ARP events should be monitored and reported. Multiple events can be specified.

continues on next page

continued

`[no] event [EVENT [ EVENT ...]]`

> **NOTE**
>
> If no specific event is specified, it is treated as `all`, which means that all events are enabled or disabled together depending on whether 'no' is used or not.

Default: All events are enabled when ARP watch is enabled.

Example:

```
example:/config/iface-vlan1/arp-watch/#> event new-station flip-flop changed
example:/config/iface-vlan1/arp-watch/#> no event flip-flop
example:/config/iface-vlan1/arp-watch/#> event flip-flop
example:/config/iface-vlan1/arp-watch/#> no event
example:/config/iface-vlan1/arp-watch/#> event all
```

Options

Option	Description
`no`	Disable the specified event(s), or all if none specified.
`EVENT`	The event type(s) to enable: • `new-station`: The Ethernet address has not been seen before. • `flip-flop`: The Ethernet address changed from the most recently seen address to the second most recently seen address. • `changed`: The host switched to a new Ethernet address. • `ethernet-broadcast`: The MAC Ethernet address is a broadcast address. • `ip-broadcast`: The IP address is a broadcast address. • `bogon`: The source IP address is not local to the local subnet. • `ethernet-mismatch`: The source MAC Ethernet address didn't match the address inside the ARP packet. • `reused-address`: The Ethernet address changed from the most recently seen address to the third (or greater) least recently seen address. • `address-conflict`: Indicates a possible IP address conflict based on MAC address changes observed in ARP traffic. Generated when any conflict- related change event occurs (changed, flip-flop, or reused-address), providing a convenient way to detect potential address conflicts. Detection is based on stateless ARP frame observation and represents a potential conflict situation. This event is generated independently and will be reported even if the individual change events are disabled.

Examples

The factory configuration for a standard interface will be configured to acquire network address through DHCP and link-local address protocols. For the example below, we will configure the two host with static network addresses and link-local addresses also.

Use case with static address

Two network units are connected via a network cable on port 1 on both hosts. Per default when a DHCP server isn't present on the network the hosts will only acquire a link link address in the

169.254.0.0/16 range. In this case we want to set fixed addresses on both host in the 192.168.0.0/24 range. We'll also use link local addresses on both hosts.

```
 Host-1                                          Host-2
.-------.                                       .-------.
|       |                                       |       |
|       |                                       |       |
|   1   |vlan1                            vlan1|   1   |
'---+---'                                       '---+---'
    |.10                192.168.0.0/24            .20|
    +-----------------------------------------------+
```

`Configuration for Host-1`

Description

```
Host-1:/#> configure
Host-1:/config/#> iface vlan1
Host-1:/config/iface-vlan1/#> inet static 192.168.2.10/24
Host-1:/config/iface-vlan1/inet-static-192.168.2.10/#> exit
Host-1:/config/iface-vlan1/#> inet link-local
Host-1:/config/iface-vlan1/inet-link-local#> leave
```

`Configuration for Host-2`

Description

```
Host-2:/#> configure
Host-2:/config/#> iface vlan1
Host-2:/config/iface-vlan1/#> inet static 192.168.2.20/24
Host-2:/config/iface-vlan1/inet-static-192.168.2.20/#> exit
Host-2:/config/iface-vlan1/#> inet link-local
Host-2:/config/iface-vlan1/inet-link-local#> leave
```

Use case in DHCP Client mode

Two network units are connected via a network cable on port 1 on both hosts. The Host-2 runs a DHCP server with subnet range 192.168.1.0. The DHCP-Server has in this case an address pool in the range of 192.168.1.100 and 192.168.1.199. The Host-1 is configured to use inet dhcp on vlan1. We'll also use link local address.

```
 Host-1                                          Host-2
.-------.                                       .-------.
| DHCP  |                                       | DHCP  |
| Client|                                       | Server|
|   1   |vlan1                            vlan1|   1   |
'---+---'                                       '---+---'
    | dhcp .100             192.168.1.0/24        .1|
    +-----------------------------------------------+
                            pool 192.168.1.100-199
```

`Configuration for Host-1`

Description

```
Host-1:/#> configure
Host-1:/config/#> iface vlan1
```

continues on next page

continued

Configuration for Host-1

```
Host-1:/config/iface-vlan1/#> inet dhcp
Host-1:/config/iface-vlan1/inet-dhcp/#> exit
Host-1:/config/iface-vlan1/#> inet link-local
Host-1:/config/iface-vlan1/inet-link-local#> leave
```

Configuration for Host-2

Description

```
example:/#> configure
example:/config/#> dhcp-server
example:/config/dhcp-server/#> subnet 192.168.1.0/24
Created new DHCP server for subnet 192.168.1.0.
example:/config/dhcp-server/subnet-192.168.1.0/#>
example:/config/dhcp-server/subnet-192.168.1.0/#>pool 192.168.1.50   192.168.1.99
example:/config/dhcp-server/subnet-192.168.1.0/#>leave
```

for more information see

- DHCP-Server/Configuration

Status Overview

In order to verify an interface's setting the 'show' command can be used:

```
Host-1:/#> show iface vlan1
Name         : vlan1
Admin. State : Up
Oper. Status : UP
MAC Address  : 52:54:00:12:34:56
IP Addresses : 192.168.2.10/24    (static)
               169.254.81.131/16  (link-local)
MTU          : 1500
Statistics   :
               RX: Packets    Bytes        Errors CsumErrs OutOfSeq Mcasts
                   57         4621         0      0        0        15
               TX: Packets    Bytes        Errors DeadLoop NoRoute  NoBufs
                   164        29003        0      0        0        0
```

This show that the interface vlan1 on Host-1 is up. It has two network addresses, 192.168.2.10/24 which has been configured as static and also network address 169.254.81.131/16 which has been resolved through link-local addressing.

2.5.2. DDNS (Dynamic DNS)

Introduction

DDNS is a service offered by many Internet Service Providers (ISPs) to allow end-users to register a dynamic (DHCP/PPPoE/PPPoA) address with a DNS record.

When the switch acquires its IP address dynamically (via DHCP or PPP) from an ISP, maintaining the DNS server entry is cumbersome. To that end the system includes support for dynamic DNS (DDNS). With DDNS enabled, the switch updates its DNS server entry automatically when acquiring a new IP address.

Below is a non-exhaustive list of supported DDNS providers:

- DYN: http://www.dyndns.org[1]
- FreeDNS: http://freedns.afraid.org[2]
- no-ip: http://www.noip.com[3]
- LOOPIA: http://www.loopia.com[4]

Basic DDNS configuration

DDNS configuration context can be found under IP context in the CLI:

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> ddns
example:/config/ip/ddns/#>
```

Syntax

`[no] enable`

Description

Enable/disable DDNS. Useful to disable a fully configured DDNS before deployment, the configuration will remain dormant while disabled.

Options

Option	Description
`no`	Disable DDNS

`[no] interval [1-864000]`

Description

This setting controls the IP address check interval, and is in range 1 - 864000. Default value is 43200 seconds (12 hours)

continues on next page

[1] http://www.dyndns.org
[2] http://freedns.afraid.org
[3] http://www.noip.com
[4] http://www.loopia.com

continued

`[no] interval [1-864000]`

Options

Option	Description
`no`	Reset interval to its default value (43200 seconds)

`[no] ca-store [auto|local]`

Description

This setting controls the Root CA certificate store. Default value is 'auto'.

Options

Option	Description
`no`	Reset CA certificate store to its default value: auto
`auto`	WeOS' builtin certificate store. This includes all certificates shipped directly with WeOS.
`local`	Local store includes certificates imported by user and overrides WeOS' builtin certificate store (auto).

`[no] provider-id [PROVIDER_UID]`

Description

Create/remove DDNS provider instance with specified UID. UIDs are unique numbers in range 1 - 999. Maximum 7 simultaneous provider instances may be configured.

Options

Option	Description
`no`	Delete provider with specified UID
`PROVIDER_UID`	Unique identifier for DDNS provider. This simply represents DDNS provider instance.

DDNS Provider configuration

In order to configure DDNS provider, a provider-instance must be created first by specifying UID.

```
example:/config/ip/ddns/#> provider-id 1
example:/config/ip/ddns/provider-1/#>
```

Syntax

`[no] provider [freedns|dyn|cloudflare|loopia|...]`

Description

Add/remove builtin DDNS provider.

continues on next page

continued

`[no] provider [freedns|dyn|cloudflare|loopia|...]`

Options

Option	Description
`no`	Remove builtin DDNS provider
`freedns`	Freedns DDNS provider.
`dyn`	Dyndns DDNS provider.
`cloudflare`	Cloudflare DDNS provider.
`loopia`	Loopia DDNS provider.
`noip`	NoIP DDNS provider.
`yandex`	Yandex DDNS provider.
`duiadns`	Duiadns DDNS provider.
`dnspod`	Dnspod DDNS provider.

`[no] custom [NAME]`

Description

Set custom DDNS provider name. This setting is mutually exclusive with `provider` setting.

Options

Option	Description
`no`	Remove custom DDNS provider
`NAME`	Custom DDNS provider name.

`[no] ssl`

Description

Enable/disable HTTPS. Enabled by default.

Options

Option	Description
`no`	Disable HTTPS. HTTP will be enabled instead.

`[no] username [USERNAME] [hash] [PASSWORD]`

Description

Set username and password for DDNS provider. Password can be omitted when setting username.

Options

Option	Description
`no`	Remove username and password

continues on next page

continued

`[no] username [USERNAME] [hash] [PASSWORD]`

Option	Description
`USERNAME`	username for DDNS provider account.
`PASSWORD`	password for DDNS provider account.
`hash`	specify password as a hash.

`[no] password [hash] [PASSWORD]`

Description

Set password for DDNS provider. Password may be set either in plain text or as a hash.

Options

Option	Description
`no`	Remove password
`PASSWORD`	password for DDNS provider account.
`hash`	specify password as a hash.

`[no] hostname [HOSTNAME[,HOSTNAME,...]]`

Description

Set hostname(s) for DDNS provider.

Options

Option	Description
`no`	Remove hostnames
`HOSTNAME`	name of the host.

`[no] server [SERVER] [PATH]`

Description

Set server and url for custom DDNS provider.

Options

Option	Description
`no`	Remove custom DDNS server and url
`SERVER`	DDNS server string (example: twodns.de).
`PATH`	path that follows ddns server string, (example: /)

`[no] checkip-server [SERVER] [PATH]`

Description

Set checkip server and url for custom DDNS provider.

Options

Option	Description
`no`	Remove checkip server and url
`SERVER`	Checkip server string (example: updatemyip.org).
`PATH`	path that follows checkip server string, (example: /update/now)

`[no] checkip-ssl`

Description

Enable/disable HTTPS for checkip serever.

Options

Option	Description
`no`	Disable HTTPS, HTTP will be used instead

2.6. Generic Maintenance

2.6.1. Auxiliary Ports

Introduction

For various security reasons it might be useful at times to completely inactivate auxiliary ports on devices, and by auxiliary ports we mean `USB` and `SD`.

While it is possible to temporarily disable auxiliary ports in software, these may be easily re-enabled later on.

What is meant to be achieved instead is complete inactivation of such ports with ability to bring them back to life when needed. Such procedures will require device to be restarted.

Auxiliary Port Management

```
example:/#> boot
example:/boot/#> show aux-port
```

`[no] aux-port [PORT]`

Description

Activate or inactivate auxiliary port

For various security reasons it might be useful at times to completely inactivate auxiliary ports.

Depending on the system, following auxiliary ports may be present:

[USB] - a USB port [SD] - an SD-card port/slot

NOTE: Device must be restarted whenever configuration changes are introduced.

`show aux-port` command lists all available auxiliary ports together with their status.

- `ACTUAL STATUS` mirrors current status of auxiliary port;
- `CONFIGURED STATUS` represents latest user-configured status;

If `ACTUAL STATUS` is at variance with `CONFIGURED STATUS` - system must be restarted for desired effect to take place.

`no aux-port usb` command will disable usb port after restart of the OS. The same command without parameter will disable all available auxiliary ports on the system.

```
example:/boot/#> show aux-port

EXPANSION PORT      ACTUAL STATE      CONFIGURED STATE
usb                 ENABLED           ENABLED
```

```
example:/boot/#> no aux-port usb

EXPANSION PORT      ACTUAL STATE      CONFIGURED STATE
usb                 ENABLED           DISABLED (reboot required)
```

Running `show aux-port` after reboot presents user with updated status.

```
example:/boot/#> show aux-port

EXPANSION PORT     ACTUAL STATE     CONFIGURED STATE
usb                DISABLED         DISABLED
```

2.6.2. System Bootstrap

Introduction

How a system boots is important to how it will actively operate. This document presents available options that govern system bootstrap.

The main aspects that are configured in this context is where the system will obtain its configuration file from and where it will access various certificates. Any external media device that should be usable on the device needs to be defined in this context.

In addition, this context provides information about the secure boot status. Secure boot is a mechanism that verifies the integrity of code before allowing it to execute during system startup. It prevents unauthorized or malicious code from compromising the boot process. The chain of trust is a sequence of steps that collectively establish trust in the system. It ensures the integrity of the software. The trust chain starts from a foundational element called the root of trust. Throughout this documentation, the term hardware anchor is used. In simple terms, the hardware anchor indicates that trusted and immutable code residing within the processor validates signatures of the subsequent code layers. Essentially, the root of trust resides within the hardware itself.

For additional information and example use-cases see:

- Config From External Media
- Access External Media Devices
- Bootstrap System Configuration Over the Network
- Bootstrap System Configuration Using DHCP
- Secure Boot Configuration using unlocking tokens

Usage of External Media Devices

SD-card and USB memory sticks are supported external storage types. Availability depends on the product make and model, not all products support all types of media.

The SD-card has higher priority than USB memory sticks. Meaning, if both an SD-card and a USB memory stick is present, the SD-card will be used.

> **NOTE**
>
> SD-cards must to be inserted before boot to be detected. Hot-plugging is not supported. USB memory sticks have hot-plug support and will be made available to the system, provided no SD-card is present.

The system will also handle partitions that may be present on an external media device. In the case of partitions, each individual one will be treated as a separate media definition in the system.

Any external media device that is intended to be used, for any purpose, on the device needs to be specified in the boot configuration context. These definitions dictate how the physical media should be matched and represented on the device. If this is not done the external media device will not be mounted and accessible to the user.

Configuration

This is the top level configuration in the boot context.

```
example:/#> boot
example:/boot/#>
```

`[no] config-sync [FROM-MEDIA]`

Description

Enable automatic sync of configuration and certificates from external media to built-in flash before system bootstrap is completed.

With this feature the system can be set up to copy files from an external media device (e.g. USB, SD-Card) to built-in flash (internal) before booting up the system with built-in flash. I.e., the external media device would only need to be inserted once at deployment.

The target media cannot be changed, it will always be the internal media definition.

Default: Disabled.

Options

Option	Description
`no`	Disable the automatic sync.
`FROM-MEDIA`	The media from which to sync from. Medias are configured and set up using the `media` command.

`[no] config-order [MEDIA:TYPE] [, MEDIA:TYPE [, ...]]`

Description

Define the system boot configuration order. At startup the `config-order` is traversed in order, the first config option to be successfully applied will be used.

The `config-order` is specified with a media and file/mode to use. Multiple options can be specified. The specified media is what the system will attempt to mount when bootstrapping the system.

> **NOTE**
>
> If an external media is specified (i.e. not `internal`), the configuration files must be located on the external media. It will not use any builtin default files in the same manner as an `internal` configuration would.

> **NOTE**
>
> If configuration is loaded from external media or netboot, the incoming configuration can bypass input validation.

If no more options exist, the `config-fallback` will be used.

Default: `internal:startup`.

```
example:/boot/#> config-order external:startup, internal:startup
```

continues on next page

continued

`[no] config-order [MEDIA:TYPE] [, MEDIA:TYPE [, ...]]`

Options

Option	Description
`no`	Reset to the default `config-order` : `internal:startup`.
`MEDIA`	Dynamically created using the **`media`** command. Use the **`show`** command in the current context or **`show media`** to list all existing media definitions. In addition, two default media definitions exist: • **`internal:`** The representation of the built-in flash. This cannot be changed or removed. • **`external:`** The default media definition for any connected external media device (e.g. USB, SD-Card). It is configured to always match the first partition on said external media device. This definition is not static or locked, it can be freely adjusted by the user, even removed.
`TYPE`	The specific type of configuration file to be used or a specific mode of obtaining said file. The following options are valid selections: • **`startup:`** This option will cause the system to boot using the standard startup configuration *file*. • **`factory:`** This option will cause the system to boot using the factory default configuration *file*. • **`none:`** This option will cause the system to boot in a *blank* state, or a no configuration mode, using the special configuration file *no-config.cfg*. In this state essentially no services will be enabled on the device and all ports will be disabled. The only access to the device will be using a console connection. • **`safe:`** This option will cause the system to boot in a *safe* state, using the special configuration file *safe-config.cfg*. In this state basic services will be enabled so that the device can easily be made accessible. In addition, all port interfaces on the device will be separated from each other and be configured as a DHCP client. This will allow a user to easily provide the device with an IP address so that the device can be made accessible with, for instance, WEB and SSH. • **`net:`** This option will cause the system to boot in net configuration *mode*. In this mode the device will attempt to obtain a startup configuration file over the net (e.g. DHCP). **TIP** For a more detailed description on these configuration files that the system uses to achieve this, and how to potentially override them with custom behaviour, refer to this page and section.

> **NOTE**
>
> 'cert-order' External should only be used to increase the available space for certificates. Any private keys that are uploaded to the device will be no be readable by other devices if the Media is transferred.

`[no] cert-order [MEDIA] [, MEDIA [, ...]]`

Description

Define the system boot certificate order.

The certificate order is specified by a media indicating where to locate the certificates. Multiple options can be specified, if one fails the next in the order will be used.

> **NOTE**
>
> A "hidden" option will always exist as the last resort, and that will be the internal flash definition (**`internal`**).

Default: `internal`.

Options

Option	Description
`no`	Reset to the default `cert-order` : `internal`.
`MEDIA`	Dynamically created using the **`media`** command. Use the **`show`** command in the current context or **`show media`** to list all existing media definitions. In addition, two default media definitions exist: • **`internal:`** The representation of the built-in flash. This cannot be changed or removed. • **`external:`** The default media definition for any connected external media device (e.g. USB, SD-Card). It is configured to always match the first partition on said external media device. This definition is not static or locked, it can be freely adjusted by the user, even removed.

`[no] config-fallback [reboot|failsafe|factory|none]`

Description

Specify the fallback option to `config-order`. The configured action is taken only if the above `config-order` fails. Should this happen an alarm is also triggered in the system.

Default: `failsafe`.

Options

Option	Description
`no`	Reset to the default fallback, safe.

continues on next page

continued

`[no] config-fallback [reboot|failsafe|factory|none]`

Option	Description
`reboot`	If this option is used the device will simply restart if the fallback is reached.
`failsafe`	This option will cause the system to boot in a *safe* state, using the special configuration file *safe-config.cfg*. In this state basic services will be enabled so that the device can easily be made accessible. In addition, all port interfaces on the device will be separated from each other and be configured as a DHCP client. This will allow a user to easily provide the device with an IP address so that the device can be made accessible with, for instance, WEB and SSH. The major difference between this and simply using the **`safe`** mode in a **`config-order`** entry, is that this is an error state. In addition, an alarm will also be triggered, indicating that the system booted on its fallback. **TIP** For a more detailed description on the special configuration file that the system uses the achieve this, and how to potentially override it with custom behaviour, refer to this page and section.
`factory`	This option will cause the system to boot using the factory default configuration file *factory-config.cfg*. The major difference between this and simply using the **`factory`** mode in a **`config-order`** entry, is that this is an error state. In addition, an alarm will also be triggered, indicating that the system booted on its fallback. **TIP** For a more detailed description on the special configuration file that the system uses the achieve this, and how to potentially override it with custom behaviour, refer to this page and section.
`none`	This option will cause the system to boot in a *blank* state, or a no configuration mode, using the special configuration file *no-config.cfg*. In this state essentially no services will be enabled on the device and all ports will be disabled. The only access to the device will be using a console connection. The major difference between this and simply using the **`none`** mode in a **`config-order`** entry, is that this is an error state. In addition, an alarm will also be triggered, indicating that the system booted on its fallback. **TIP** For a more detailed description on the special configuration file that the system uses the achieve this, and how to potentially override it with custom behaviour, refer to this page and section.

`[no] media NAME`

Description

Create and manage media mapping definitions.

Specify the media devices that should be recognised by the system and be made available to the user. The media definitions that are defined here dictate how any external media device (e.g. USB, SD-Card) will be mapped and accessible on the rest of the system.

Each individual media definition require a specific 'match' rule that specify how a physical media will be linked to the media definition. This allows the user to match not only a specific external media device as a whole, but individual partitions that may exist on it as well. A media definition can be matched based on the following information:

- **Partition number:** Used to indicate a match based on a specific partition number on the physical media device. If no partitions exist on the device, that will count as a match for partition 1 (The first).
- **Label:** Used to indicate a match based on a specific label possibly present on the physical media device.

Currently it is only possible to create media definitions for external media devices that are connected. Hence, it is not possible to create any media definitions for internal memory on the built-in flash. The built-in flash is currently entirely represented by the special media definition `internal`. This is a definition that *cannot* be changed or removed by the user.

An external media device will only be accessible on the system if a matching media definition exist. In other words, if no match exist the device will not be mounted on the system whatsoever.

If a match cannot be made, this media will simply not be present, *no* error will occur because a specific match cannot be made.

Default: By default two different media definitions will be present:

- `internal:` Special definition and representation of the built-in flash, this media definition *cannot* be changed or removed.
- `external:` Base definition aimed to match the first partition located on any connected external physical media. If no partitions exist on the device this will also be a match in that case.

Options

Option	Description
`no`	Resets all media definitions back to the defaults.
`NAME`	Media definition name. Maximum length is 32 characters. Only alphanumeric characters and the special characters underscore (`_`) and dash (`-`) are allowed.

`[no|show] net`

Description

Enter a sub-configuration context handling `net` boot configuration options.

continues on next page

continued

`[no|show] net`

Options

Option	Description
`no`	Resets all `net` specific settings to its defaults.

`[no|show] loader`

Description

Enter the configuration context for the boot-loader, reset boot-loader settings to default, or show current settings.

Options

Option	Description
`no`	Resets all boot-loader settings.

Media Settings

The media configuration is a sub-context of the boot configuration context.

> **NOTE**
>
> It is currently not possible to perform any changes to internal media definition. Any other media definition can be freely changed in any way the user wants.

```
example:/#> boot
example:/boot/#> media external
example:/boot/media-external/#>
```

`[no] match [label NAME] | [partition NUMB]`

Description

Set a match rule specifying how the media will be identified.

The match rule specify how a physical device (e.g. USB, SD-Card) will be linked to the media definition.

> **NOTE**
>
> Currently the match rule cannot specify a specific external media connection port, it will operate based on the device that is connected. If a device has the possibility to connect more than one external media device (Both a USB device and a SD-Card) it cannot be guaranteed which one will be selected, but in general the SD-Card will be preferred.
>
> However, it is safest to currently only have a maximum of one external media device connected, in order to ensure that the behaviour will be deterministic.

continues on next page

continued

`[no] match [label NAME] | [partition NUMB]`

```
example:/boot/media-external/#> match partition 1
example:/boot/media-external/#> match label my-device-label
```

Options

Option	Description
`no`	A match rule must always exist for a media definition. Therefore, it cannot be removed, it can only be changed.
`partition`	Used to indicate a match based on a specific partition number on the physical media device. If no partitions exist on the device, that will count as a match for partition 1 (The first).
`label`	Used to indicate a match based on a specific label possibly present on the physical media device.

`[no] read-only`

Description

Enable, or disable read-only mode.

Default: Disabled.

Options

Option	Description
`no`	Disables read-only.

`[no] timeout [1-60]`

Description

Set the media timeout, how long to wait for external media to “wake up” at power-on. Some USB stick controllers can take up to 30 sec. to start up.

Default: 30 seconds.

Options

Option	Description
`no`	Resets the timeout to its default, 30 seconds.

Net Settings

Net configuration is a sub-context in the boot configuration.

```
example:/#> boot
example:/boot/#> net
example:/boot/net/#>
```

`[no] sync`

Description

Sync any configuration file, obtained with the net boot mode, to the startup configuration file of the selected boot media.

The media that the obtained configuration file will be synced to is determined by the media selected in association with the net mode in the **`config-order`**. A few examples of this:

- **`internal:net`**: Sync the obtained configuration to the startup config located on the built-in flash.
- **`external:net`**: Sync the obtained configuration to the startup config located on the external media device specified by the media definition **`external`**.

Default: Disabled.

Options

Option	Description
`no`	Disable sync.

`[no] mode [dhcp]`

Description

Net boot mode to use if configured in the `config-order`.

Default: `dhcp`.

Options

Option	Description
`no`	Resets to its default.
`dhcp`	Use DHCP option 66 and 67 to retrieve server location and the name of the configuration file to download and apply.

`[no] timeout [0|1-2147483647]`

Description

Timeout, in seconds, for how long to attempt a net boot. When the timer expires, the next option in the `config-order` will be attempted.

If the `timeout` is set to 0, the timeout will be infinite.

Default: 300.

Options

Option	Description
`no`	Reset to the default value.

`[no] clientid [STRING]`

Description

Client ID to be used in the DHCP Discover message. This can be used to specify to the DHCP server any specific configuration file that this device wants.

Default: None.

Options

Option	Description
`no`	Remove any configured Client ID.
`STRING`	A free form string, with a maximum length of 64 characters.

`[no] accept-first-lease`

Description

Enable to always accept the first DHCP lease received.

When this setting is enabled the DHCP boot handling will accept the first lease that it receives. What accept entails depends on the contents of the received DHCP lease, the following two cases exist:

1. If the DHCP lease contains boot information (option 66/67), an attempt to download the config will be performed as normal.
2. If the DHCP lease contains *NO* boot information (option 66/67), the bootstrap process will directly continue to the next option specified in the boot **`config-order`**, or if no other option exist go to the **`config-fallback`**.

Default: Disabled.

Options

Option	Description
`no`	Disable sync.

`[no] password [STRING]`

Description

A password may be used to secure login during net boot when the console is accessible.

If left blank the net boot password will be the factory defaults until a the user configuration is received.

Default: None.

Options

Option	Description
`no`	Remove any configured Password

continues on next page

continued

`[no] password [STRING]`

Option	Description
`STRING`	A free form string, with a maximum length of 64 characters.

Boot-loader Settings

Boot-loader configuration is a sub-context in the boot configuration.

```
example:/#> boot
example:/boot/#> loader
example:/boot/loader/#>
```

`[no] login password|hash STRING`

Description

Set the bootloader password.

Options

Option	Description
`no`	Disable password.
`STRING`	The string representation of the password.

`[no] rescue-address IPADDR`

Description

Set the rescue mode (netconsole) IP address.

Parameters

Options

Option	Description
`no`	Reset the setting to its default value, 192.168.2.200.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] rescue-netmask NETMASK`

Description

Set the rescue mode (netconsole) netmask.

Options

Option	Description
`no`	Reset the setting to its default value, 225.255.255.0.
`NETMASK`	Size of the net in quad-dotted format, e.g. 255.255.255.0.

© 2026 Westermo Network Technologies AB

`[no] rescue-port PORT`

Description

Set the rescue mode (netconsole) UDP port.

Options

Option	Description
`no`	Reset the setting to its default value, 6000.
`PORT`	Port number in range 1-65535.

`[no] rescue-peer IPADDR`

Description

Set the rescue mode (netconsole) peer IP.

Options

Option	Description
`no`	Reset the setting to its default value, 192.168.2.1.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] allow-untrusted`

Description

Set allow untrusted for non hardware-anchored machines. This disables the OS signature check. It's recommended to leave this at the default value.

Options

Option	Description
`no`	Reset the setting to its default value, re-enabling the OS signature check for non hardware-anchored machines. This setting has no effect on hardware-anchored machines.

`[no] unlock-license [VFS-uri]`

Description

Set a cryptographic unlock license for hardware anchored machines. This disables the OS signature check. It's recommended to leave this at the default value.

Options

Option	Description
`no`	Reset the setting to its default value, re-enabling the OS signature check for hardware-anchored machines. This setting has no effect on non hardware-anchored machines.

`boot-integrity restore`

Description

Restores a boot configuration integrity violation. This feature protects the bootloader configuration. Note: If downgrading to a version before this feature, and then under '/boot' menu modifies configuration this will result in a integrity violation after an upgrade.

Status

The currently configured boot settings can be viewed form the top level in the CLI:

```
example:/#> show boot
Bootstrap Configuration
Config-sync     : Disabled
Config-fallback : failsafe

Config order
#  MEDIA                      MODE
1  internal                   startup


Certificate order
#  MEDIA
1  internal


Media Configuration
#  NAME                       R/O  TIMEOUT  MATCH
1  internal                   No   -------  System Internal
2  external                   No   30       Partition number: 1

Net Configuration
Sync            : Disabled
Timeout         : 300
Mode            : dhcp
Client ID       :

Bootloader Configuration
Login Password  : Disabled

Rescue Mode
    Address     : None
    Netmask     : None
    Peer        : None
    Port        : -2

Secure boot status
    Hardware anchor : Disabled
    Secure OS boot  : Enforce signed

Secure boot control
    Allow untrusted : Disabled

Boot configuration integrity
    Status          : OK
```

2.6.3. Encrypted secrets

Introduction

The Encrypted Secrets feature ensures that no passwords/secrets are stored in plain text within the WeOS configuration, when enabled.

Encryption sources

The secrets are encrypted using the AES-256 encryption algorithm and the encryption key depends on the chosen encryption source:

- **portable**: The same encryption key for all devices, useful for sharing configurations across multiple systems
- **hwuid**: Device-specific encryption key, enhancing security for individual devices (not portable)
- **password**: Password-derived encryption key, enabling portability if the same password is used on all devices

Configuration

The encryption source can be set using the `encrypt-secrets` command under the *configure system* menu. The command takes one argument, the encryption source.

To use the *password* encryption source, first set the password using the `encrypt-password` command in the *admin-exec* context.

```
example:/#> encrypt-password s3cretPassword
example:/#> configure
example:/config/#> system
example:/config/system#> encrypt-secrets password
example:/config/system#> leave
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

In this case the password *s3cretPassword* is used to derive the encryption key since the *password* encryption source is chosen.

Changing the encryption password

Steps to safely change the password:

1. Change the password using the `encrypt-password` command in *admin-exec* context
2. Save the updated running configuration to the startup configuration using the `cp running startup` command

> **WARNING**
>
> Changing the encryption password can cause encrypted secrets in the startup configuration to become inaccessible. To avoid this, ensure the running configuration is saved.

> **NOTE**
>
> This problem only occurs when encryption source is set to *password* and the password is changed.

Moving configuration between devices

If the encryption source is set to *hwuid*, the encrypted secrets are not portable. This means that the configuration file cannot be moved to another device. The encrypted secrets will be unreadable on another device.

To move configurations between devices, export them using the *portable* encryption source. For added security, the source could be changed to *hwuid* after loading the configuration on the new device.

The middle ground between portability and protection from information disclosure could be to use the provided password option.

Example secrets

- **IPsec**: PSK secret
- **RIP**: authentication password
- **OSPF**: authentication password
- **DDNS**: password
- **SNMP**: community strings
- **RADIUS**: shared secret
- **TACACS+**: shared secret
- **SCEP**: challenge password

Verify that the secrets are encrypted by looking at the configuration. (*show running-config*). The encrypted secrets are displayed as base64 encoded strings.

> **NOTE**
>
> The encryption password is set independently of the encryption source and not saved in the configuration file, instead it is stored in the device's secure storage.

`[no] encrypt-secrets <source>`

Description

Select encryption source.

Options

Option	Description
`no`	Store secrets in plain text.
`hwuid`	Use device unique encryption key. Not portable.
`password`	Use password to derive encryption key.
`portable`	Use the same encryption key for all devices (derived from firmware).

2.6.4. Factory Reset

Introduction

The system configuration contains of several parts:

- WeOS configuration
- Boot configuration
- Certificate and key repository
- SSH host keys

This section describes the various methods to conduct a factory reset, which

- resets the WeOS configuration to the factory configuration of the running WeOS version
- resets the boot configuration to its default values
- clears the certificate and key repository
- clears the SSH host keys

As part of the factory reset, the unit will reboot.

> **NOTE**
>
> Upon reboot, the system will create a default Web server certificate (self-signed), and also create new SSH host keys. The WeOS configuration and certificate/key repository are typically stored on internal flash. See Media Tools for details and alternatives.

If you only wish to reset the WeOS startup configuration, the CLI provides the following method:

```
example:/#> copy factory-config startup-config
example:/#>
```

Factory Reset via Web Interface

To perform a factory reset of the device using the Web interface, navigate to **Maintenance** -> **Factory Reset**, then press the **Reset** button.

Factory Reset via the CLI

Factory reset can be done via the `factory-reset` command when logged into the unit.

```
example:/#> factory-reset
Factory Reset -- Restore factory default settings.
=> Are you sure (y/N)? y
Erasing barebox configuration
Backing up theme catalog.
Erasing all certificate and config files.
Restoring branding.
All done, rebooting ...
```

Factory Reset via the Console

Factory reset can be done via the console port login prompt. This uses the special user and password **factory/reset**.

```
example login: factory
Password: reset (typed, but not echoed back to screen)

Factory Reset -- Restore factory default settings.
```

```
⇒ Are you sure (y/N)? y
Erasing barebox configuration
Backing up theme catalog.
Erasing all certificate and config files.
Restoring branding.
All done, rebooting ...
```

2.6.5. Filesystem

Overview

> ⚠ **WARNING**
>
> This context is used for direct file-access and should therefore be used with caution. For example, erase will not keep track if the file is used by any other device subsystem. Therefore, performing this action on used files may cause unforeseen behaviour.
>
> The file context provides a number of commands to handle files in certain predefined directories located on internal and external media.

The commands are:
- **diff** - Compare two files
- **dir** - List files
- **copy** - Copy file from/to internal or external media
- **show** - Show file
- **erase** - Erase configuration
- **import** - Import file from remote source onto filesystem
- **export** - Export file to remote destination

The following subcontexts can be selected in the file context. Each subcontext has its own set of implicit directories and set of suitable commands for the applicable type of file.
- **cfg** Configuration file
- **log** Log files

File system commands are accessed from the admin execution context in the following manner:

```
example:/#> file
example:/file/#>
```

Then selecting subcontext:

```
example:/file#> log
example:/file/log#>
```

Or one can move to the subcontext directly:

```
example:/#> file log
example:/file/log#>
```

Media selector

Throughout this series of commands the keyword media can be used to select between internal media and available external media. External media is a connected USB-stick or SD-card, internal media is the builtin storage.

For additional information on the Media tools and commands, refer to this page.

For additional information on how to access external media, refer to this page

Cfg

Handling of configuration files.

`diff [media <label>] <FILE1> [media <label>] <FILE2>`

Description

Compare two configuration files. Shows the differences between two files

Example:

```
example:/file/#> cfg
example:/file/cfg/#> diff config0.cfg media internal config1.cfg
--- config0.cfg
+++ config1.cfg
@@ -1,6 +1,6 @@
{
  "meta": {
- "id": "factory-config.cfg",
+ "id": "Example",
  "vendor": "Westermo",
  "os": "WeOS",
  "release": "5.3"
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted configured boot media is used.

`dir [media <label>] [subdir]`

Description

List files on internal or external media.

Example:

```
example:/file/#> cfg
example:/file/cfg/#> dir
/cfg/                   MOD. DATE/LINK    SIZE  MD5
config0.cfg             2025-01-14 13:44  2724
config99.cfg            2025-02-04 14:16  9739  a38550e2507d12fc61bf119df2c9481c
startup-config.cfg → config0.cfg
startup-config.lnk      2024-12-04 11:28    12
Free: 45.0 MiB │ Total: 45.3 MiB
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted configured boot media is used.
`subdir`	Optional sub directory to list contents of.

`copy [media <label>] <src> [media <label>] <dst>`

Description

Copy file from from/to internal or external media. If media keyword is omitted configured boot media is used.

Example:

```
example:/file/#> cfg
example:/file/cfg/#> copy config0.cfg media external config_ex.cfg
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted configured boot media is used.
`src`	Source filename
`dst`	Destination filename

`erase [media <label>] <file>`

Description

Erase file

Example:

```
example:/file/#> cfg
example:/file/cfg/#> erase example.cfg
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted configured boot media is used.
`file`	File to erase.

`show [media <label>] <file> [KEYWORD[ KEYWORD ...]`

Description

Show configuration

If no keyword is given, the command shows the entire configuration file.

If keywords are given that matches keys within the JSON file, only those sections are shown. The keywords can be nested, for example: show config0.cfg bridge 0 vlan

Example:

continues on next page

continued

`show [media <label>] <file> [KEYWORD[ KEYWORD ...]`

```
example:/file/#> cfg
example:/file/cfg/#> show config0.cfg bridge 0 vlan
[
{
  "vid": 1,
  "id": "vlan1",
  "enable": true,
  "untagged": [
  "all"
  ],
  "multicast-snooping": true,
  "priority": 0
}
]
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted configured boot media is used.
`file`	File to show.

`import [SOURCE] [DESTINATION]`

Description

Import configuration from remote source onto filesystem.

Example:

```
import proto ftp user user1 password pw1 192.0.2.1 remote_path local_file
```

`SOURCE`

Description

`[proto <ftp|tftp|scp|sftp|http|https>] [user <username>] [password <password>] <address[:port]> <remote_path>`

Options

Option	Description
`proto`	Protocol to use. ftp, tftp, scp, sftp, http or https.
`username`	Username if used by protocol.
`password`	Password if used by protocol.
`remote_path`	Path on the remote side.

continues on next page

continued

`import [SOURCE] [DESTINATION]`

`DESTINATION`

Description

`[media <label>] <local_file>`

Options

Option	Description
`media`	Media selector. Select any available media.
`local_file`	File to use as destination

`export [SOURCE] [DESTINATION]`

Description

Export configuration file to remote destination.

If the remote filename is omitted the local filename is used as the destination filename.

Example:

```
example:/file/#> cfg
example:/file/cfg/#> export local_file proto ftp user user1 password pw1 192.168.0.2.1 remote_path
```

`SOURCE`

Description

`[media <label>] <local_file>`

Options

Option	Description
`media`	Media selector. Select any available media.
`local_file`	File to use as source

`DESTINATION`

Description

```
[proto <ftp|tftp|scp|sftp|http|https>] [user <username>] [password <password>]
<address[:port]> <remote_path>
```

Options

Option	Description
`proto`	Protocol to use. ftp, tftp, scp, sftp, http or https.
`username`	Username if used by protocol.

continues on next page
continues on next page

continued

`export [SOURCE] [DESTINATION]`

continued

`DESTINATION`

Option	Description
`password`	Password if used by protocol.
`remote_path`	Path to be used on the remote side.

Log

Handling of log files.

`dir [media <label>] [subdir]`

Description

List files on internal or external media

Example:

```
example:/file/#> log
example:/file/log/#> dir
/log/                   MOD. DATE/LINK    SIZE  MD5
bootsettings.log        2025-02-24 12:20    20
btmp                    2025-02-24 12:20     0
kern.log                2025-02-24 12:24 51539
lastlog                 2025-02-24 12:20     0
messages                2025-02-24 12:20  7533
mount.log               2025-02-24 12:20  1908
ntp.log                 2025-02-24 12:20   240
syslog                  2025-02-24 12:20 59072
wtmp                    2025-02-24 12:20  2304
Free: 45.0 MiB | Total: 45.3 MiB
```

Options

Option	Description
`media`	Optional selector for external or internal media. If media keyword is omitted internal media is used.
`subdir`	Optional sub directory to list contents of.

`follow [media <label>] <file>`

Description

Continuously monitor a (log) file, like UNIX ‘tail -F’.

Example:

```
example:/file/#> log
example:/file/log/#> follow messages
```

continues on next page

continued

`follow [media <label>] <file>`

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted internal media is used.
`file`	File to follow.

`copy [media <label>] <src> [media <label>] <dst>`

Description

Copy file from/to internal or external media.

Example:

```
example:/file/#> log
example:/file/log/#> copy messages media external messages
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted internal media is used.
`src`	Source filename
`dst`	Destination filename

`erase [media <label>] <file>`

Description

Erase file

Example:

```
example:/file/#> log
example:/file/log/#> erase example.log
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted internal media is used.
`file`	File to erase.

© 2026 Westermo Network Technologies AB

`show [media <label>] <file>`

Description

Show content of file on internal or external media.

Example:

```
example:/file/#> log
example:/file/log/#> show example.log
Test content
```

Options

Option	Description
`media`	Media selector. Select any available media. If media keyword is omitted internal media is used.
`file`	File to show.

`export [SOURCE] [DESTINATION]`

Description

Export log to remote destination

If the remote filename is omitted the local filename is used as the destination filename.

Example:

```
example:/file/#> log
example:/file/log/#> export local_file proto ftp user user1 password pw1 192.168.0.2.1 remote_path
```

`SOURCE`

Description

`[media <label>] <local_file>`

Options

Option	Description
`media`	Media selector. Select any available media.
`local_file`	File to use as source

`DESTINATION`

Description

`[proto <ftp|tftp|scp|sftp|http|https>] [user <username>] [password <password>] <address[:port]> <remote_path>`

continues on next page
continues on next page

continued

`export [SOURCE] [DESTINATION]`

continued

`DESTINATION`

Options

Option	Description
`proto`	Protocol to use. ftp, tftp, scp, sftp, http or https.
`username`	Username if used by protocol.
`password`	Password if used by protocol.
`remote_path`	Path to be used on the remote side.

2.6.6. Media Tools

Introduction

The system provides a number of simple tools to handle and maintain external media devices (e.g. USB, SD-Cards). External media devices can for instance be used for boot configuration and logging.

These media tools allows for simple partitioning and formatting of connected external media devices. Tools also exist for easy population of relevant files on external media along with synchronisation of files between medias.

For other related information see the following pages:

- System Bootstrap
- Config From External Media
- Access External Media Devices

Overview

The media tools provided are quite limited in the uses that they provide. They aim to provide the possibility to perform the most basic type of operations directly on the device. For more extensive or advanced media operations it is recommended to use a PC. However, the provided tools should be able to handle most of the general use-cases that exist.

The following functionality is provided by the system, that acts directly on the connected external media device(s):

- **Partition:** Allows the user to partition and format a connected external media device in one simple step. It is possible to create up to four partitions. Each of these individual partitions will require a file system to be selected for the formatting that is performed at the same time. The limits to the formatting is the same as the stand-alone **format** option allows.
- **Format:** Allows the user to format a specific external media device with a file system and optional label. Currently one of the following file systems can be selected:
  - **ext2**
  - **ext3**
  - **ext4**
  - **vfat**, deprecated, see section filesystem.

The upcoming functions do not operate directly on the connected external media device itself. Instead they take a media definition as the selection variable. The operation can also only be used when the specified media definition is *Active*, i.e. the media definition matches a physical media device. The available functionality is as follows:

- **Populate:** Used to provide a factory configuration and default external media file structure to the specified media.
- **Sync:** Used to sync configuration files and certificates between two specified medias.

> **NOTE**
>
> Only one representation of the built-in flash currently exist named *internal*. All other media definitions that can be used are matched against externally connected media devices.

Configuration

Media configuration tools are accessed from the admin execution context in the following manner:

```
example:/#> media
example:/media/#>
```

`format [DEV]`

Description

Format any connected physical media device (USB/SD-Card).

Simple formatting of connected media device or partition. Allows to select the wanted filesystem and specification of an optional label.

> **NOTE**
>
> The command will enter a sub-context, see options in this context below.

> **WARNING**
>
> Any files present on the selected device will be removed, once the formatting has been initiated. Therefore, make sure to backup any important files beforehand.
>
> Do not worry, the formatting will not begin until the `start-formatting` command is invoked in the actual format context, that is entered using this command.

Example

```
example:/media/#> format sdb1
example:/media/format-sdb1/#> filesystem ext3
example:/media/format-sdb1/#> label default
example:/media/format-sdb1/#> show
Format settings for device sdb1
Filesystem : ext3
Label      : default
example:/media/format-sdb1/#> start-formatting

Warning: Formatting this dev will remove any files currently stored on it!
         Backup any data you want to keep before formatting the dev.

Do you still want to proceed (y/N)?y

Formatting: 100% - [==================================================================]

Format successful!
example:/media/#>
```

Options

Option	Description
`DEV`	The physical media device to be formatted. This command is tab-completed for the available devices. Further, the available devices can be listed with the `show devices` command inside the media context.

`partition [DEV]`

Description

Partition any connected physical media device (USB/SD-Card).

Easy setup of the partitioning of a connected media device. This includes defining each partition with a partition number, a size of the partition, an optional label and a fs to format the device with.

> **NOTE**
>
> The command will enter a sub-context, see options in this context below.

> **WARNING**
>
> Any files present on the selected device will be removed, once the partitioning and formatting has been initiated. Therefore, make sure to backup any important files beforehand.
>
> Do not worry, the formatting will not begin until the `start-partitioning` command is invoked in the actual partition context, that is entered using this command.

Example

```
example:/media/#> partition sdb
Entering partitioning mode with dev sdb as target ...

No changes will be performed until the start-partition command is
invoked. When invoked the entire device sdb will be fully re-partitioned
and re-formated, any files on the device will be LOST/REMOVED, keep
this in mind before proceeding.

example:/media/partition-sdb/#> delete all
example:/media/partition-sdb/#> add label default size 50% format ext3
example:/media/partition-sdb/#> add label my-files size 20% format ext3
example:/media/partition-sdb/#> add size 20% format ext2
example:/media/partition-sdb/#> add label example-folder size 10% format ext4
example:/media/partition-sdb/#>
example:/media/partition-sdb/#> show
Partition table for dev: sdb
#  DEV-NAME      PART-NUMB  LABEL                          SIZE      FS
1  /dev/sdb1     1          default                        50%       ext3
2  /dev/sdb2     2          my-files                       20%       ext3
3  /dev/sdb3     3          -                              20%       ext2
4  /dev/sdb4     4          example-folder                 10%       ext4
example:/media/partition-sdb/#>
example:/media/partition-sdb/#> start-partitioning

Understand that the content on the device will be lost during the
partitioning and formatting of the device. Make sure that the device
does not contain any files that needs to be saved elsewhere.

Start partitioning and formatting of the device (y/N)? y
Attempt to delete existing partitions on device /dev/sdb:
  1: /dev/sdb1 [Partition numb: 1]
```

continues on next page

continued

`partition [DEV]`

```
  2: /dev/sdb2 [Partition numb: 2]
  3: /dev/sdb3 [Partition numb: 3]
  4: /dev/sdb4 [Partition numb: 4]

Attempt to partition device /dev/sdb in the following manner:
  1: /dev/sdb1 [Partition numb: 1], [Size: 50%]
  2: /dev/sdb2 [Partition numb: 2], [Size: 20%]
  3: /dev/sdb3 [Partition numb: 3], [Size: 20%]
  4: /dev/sdb4 [Partition numb: 4], [Size: 10%]

Initiate formatting of the partitions, this may take a while:
  1: /dev/sdb1 [Partition numb: 1] [Label: default], format with ext3 FS ...
Formatting: 100% - [==================================================================]

  2: /dev/sdb2 [Partition numb: 2] [Label: my-files], format with ext3 FS ...
Formatting: 100% \ [==================================================================]

  3: /dev/sdb3 [Partition numb: 3] [Label: -], format with ext2 FS ...
Formatting: 100% \ [==================================================================]

  4: /dev/sdb4 [Partition numb: 4] [Label: example-folder], format with ext4 FS ...
Formatting: 100% \ [==================================================================]

Device sdb successfully partitioned and formatted!
example:/media/#>
```

Options

Option	Description
`DEV`	The physical media device to be formatted. This command is tab-completed for the available devices. Further, the available devices can be listed with the **`show devices`** command inside the media context.

`populate [MEDIA]`

Description

Populates external media with factory config and directories for certificates.

Example

```
example:/media/#> populate external
```

Options

Option	Description
`MEDIA`	Dynamically created using the **`media`** command in the boot configuration context. Use the **`show`** command in the current context to list all existing media definitions. All usable media definitions will be listed under *Active*. In addition, two default media definitions exist if no changes have been made to medias in the boot context:

continues on next page

continued

`populate [MEDIA]`

Option	Description
	• **`internal:`** The representation of the built-in flash. This cannot be changed or removed. • **`external:`** The default media definition for any connected external media device (e.g. USB, SD-Card). It is configured to always match the first partition on said external media device. This definition is not static or locked, it can be freely adjusted by the user, even removed.

`sync [FROM]<MEDIA> [TO]<MEDIA>`

Description

Sync configuration and certificates between media.

Copies configuration and certificates from the **`FROM`**, media if any, to the **`TO`** media. If no **`FROM`** media is specified the currently used media based on the active boot config is chosen.

Example

```
example:/media/#> sync internal external
Syncing may remove .cfg and certificate files from destination media.
Are you sure you want to continue (y/N)? y
example:/media/#>
```

Options

Option	Description
`MEDIA`	Dynamically created using the **`media`** command in the boot configuration context. Use the **`show`** command in the current context to list all existing media definitions. All usable media definitions will be listed under *Active*. In addition, two default media definitions exist if no changes have been made to medias in the boot context: • **`internal:`** The representation of the built-in flash. This cannot be changed or removed. • **`external:`** The default media definition for any connected external media device (e.g. USB, SD-Card). It is configured to always match the first partition on said external media device. This definition is not static or locked, it can be freely adjusted by the user, even removed.

`eject [force] [MEDIA]`

Description

Eject (unmount) a connected external media device.

Try to safely unmount and eject the specified media device. If no specific media is specified the currently active media(s) will be ejected (except the internal media).

continues on next page

continued

`eject [force] [MEDIA]`

> ⚠ **NOT EJECTABLE**
>
> The internal media cannot be ejected.

Example

Eject a specific media:

```
example:/media/#> eject external
Are you sure you want to eject media 'external' (y/N)? y
Media 'external' ejected successfully.
example:/media/#>
```

Eject all active media:

```
example:/media/#> eject
Are you sure you want to eject all active media (y/N)? y
Media 'external' ejected successfully.
example:/media/#>
```

Options

Option	Description
`force`	If specified the media will try to be unmounted regardless if it is busy or not. This may lead to data loss on the media.
`MEDIA`	Dynamically created using the **`media`** command in the boot configuration context. Use the **`show`** command in the current context to list all existing media definitions. All usable media definitions will be listed under *Active*. In addition, two default media definitions exist if no changes have been made to medias in the boot context: • **`internal:`** The representation of the built-in flash. This cannot be changed or removed. • **`external:`** The default media definition for any connected external media device (e.g. USB, SD-Card). It is configured to always match the first partition on said external media device. This definition is not static or locked, it can be freely adjusted by the user, even removed.

`mount-inactive`

Description

Try to mount all inactive media defined in the boot configuration.

When a media is **inactive** it has been defined as a media in the boot configuration but is not currently mounted. This command will try to mount all such media.

Generally an inactive media should be mounted on a hotplug event of the media device. This command is provided as a way to manually mount any inactive media if needed.

continues on next page

continued

`mount-inactive`

> **MOUNTING INTERNAL MEDIA**
>
> The internal media is always mounted and cannot be mounted again.

Example

```
example:/media/#> mount-inactive
Attempting to mount 1 inactive media ...
Media 'external' mounted successfully.
example:/media/#>
```

`mount-available [DEV]`

Description

Try to mount available media device(s).

When a media device is **available** it means that there is a physical device connected to the system. This command will try and mount all or a specified device to a media defined in the boot configuration. In other words, a defined media that is listed as **inactive** must be present in order for there to be a possibility to mount it. Hence, it must still be defined in the boot configuration, this command simply tries to mount the device based on the match rules defined for a given media definition.

Generally an available media should be mounted on boot or on a hotplug event of the media device. This command is provided as a way to try manually and mount any available media if needed.

Example

Try to mount any available media:

```
example:/media/#> mount-available
Device 'sda1' mounted successfully on media 'external'.
example:/media/#>
```

Try to mount a specific device:

```
example:/media/#> mount-available sda1
Device 'sda1' mounted successfully on media 'external'.
example:/media/#>
```

Options

Option	Description
`DEV`	The physical media device to be mounted. This command is tab-completed for the available devices. Further, the available devices can be listed with the `show devices` command inside the media context.

Format Settings

The format configuration tool is a sub-context of the media context:

```
example:/media/#> format sdb1
example:/media/format-sdb1/#>
```

> **NOTE**
>
> In the format context you create the intended formatting setup for the device. None of the commands will perform any direct changes on the device. Only when the `start-formatting` command is called will any actual changes be performed on the selected device.

`[no] filesystem [FS]`

Description

Select desired filesystem to format on the selected device.

Options

Option	Description
`no`	A file system must be selected, it can only be changed not removed.
`FS`	The wanted filesystem selectable from the following types: • `ext2` • `ext3` • `ext4` • `vfat`, deprecated, see below.

> **NOTE**
>
> VFAT is deprecated and is not fully supported. Certificates cannot be imported on external media formatted with VFAT.

`[no] label [LABEL]`

Description

Set optional label to be used on the formatted device.

Options

Option	Description
`no`	Remove any configured label. It is fine to have a file system that has no configured label.
`LABEL`	Free-form string of the wanted label for the file system. The allowed max length of the label may depend on the file system selected. If it is to long for a particular file system the name may be automatically trimmed.

continues on next page

continued

`[no] label [LABEL]`

Option	Description
	NOTE A label is not required, it is valid to not have a label.

`start-formatting`

Description

Start formatting the device according to current configuration.

Before this command is invoked no changes will be made to the selected device. If the formatting context is exited and this command has not been called, the device will retain its previous formatting configuration.

> **WARNING**
>
> When this command is invoked and the formatting of the device commence, all files currently present on the device will be removed. Ensure that any file that should be saved have been so.

Partition Settings

The partition configuration tool is a sub-context of the media context:

```
example:/media/#> partition sdb
example:/media/partition-sdb/#>
```

> **NOTE**
>
> In the partition context you create the intended structure of the wanted partitions. None of the commands will perform any direct changes on the device. Only when the **`start-partitioning`** command is called will any actual changes be performed on the selected device.

`add [part-numb NUMB] [label LABEL] [size SIZE] [format FS]`

Description

Add a new partition definition.

Options

Option	Description
`part-numb`	The wanted partition number for the partition, specified by **`NUMB`** and must be between 1-4.

continues on next page

continued

`add [part-numb NUMB] [label LABEL] [size SIZE] [format FS]`

Option	Description
	NOTE If not specified the lowest available partition number will be used. **NOTE** Multiple partition definitions cannot have the same partition number.

`label`

Description

The wanted label for the filesystem to be formatted on the partition.

Options

Option	Description
`LABEL`	Free-form string of the wanted label for the file system. The allowed max length of the label may depend on the file system selected. If it is to long for a particular file system the name may be automatically trimmed. **NOTE** A label is not required, it is valid to not have a label.

`size`

Description

The wanted size of the partition, specified by **`SIZE`**.

Options

Option	Description
`SIZE`	The size of the partition, specified by a numerical value and a value indicator: K, M, G or %. Examples: 1000K, 20M, 1G or 50%

`format`

Description

The wanted filesystem to be formatted on the partition.

continues on next page
continues on next page

continued

`add [part-numb NUMB] [label LABEL] [size SIZE] [format FS]`

continued

`format`

The format can be selected from the values specified by **FS**.

Options

Option	Description
FS	The wanted filesystem selectable from the following types: • ext2 • ext3 • ext4 • vfat, deprecated, see section filesystem.

`modify INDEX [part-numb NUMB] [label LABEL] [size SIZE] [format FS]`

Description

Modify an existing partition definition.

Options

Option	Description
INDEX	The index of the partition definition to modify.
part-numb	The wanted partition number for the partition, specified by **NUMB** and must be between 1-4. **NOTE** If not specified the lowest available partition number will be used. **NOTE** Multiple partition definitions cannot have the same partition number.

`label`

Description

The wanted label for the filesystem to be formatted on the partition.

Options

Option	Description
LABEL	Free-form string of the wanted label for the file system. The allowed max length of the label may depend on the file system selected. If it is to long for a particular file system the name may be automatically trimmed.

continues on next page
continues on next page

continued

`modify INDEX [part-numb NUMB] [label LABEL] [size SIZE] [format FS]`

continued

`label`

Option	Description
	NOTE A label is not required, it is valid to not have a label.

`size`

Description

The wanted size of the partition, specified by **`SIZE`**.

Options

Option	Description
`SIZE`	The size of the partition, specified by a numerical value and a value indicator: K, M, G or %. Examples: 1000K, 20M, 1G or 50%

`format`

Description

The wanted filesystem to be formatted on the partition.

The format can be selected from the values specified by **`FS`**.

Options

Option	Description
`FS`	The wanted filesystem selectable from the following types: • `ext2` • `ext3` • `ext4` • `vfat`, deprecated, see section filesystem.

`delete [INDEX] [all]`

Description

Delete a specifc or all existing partition definitions.

continues on next page

continued

`delete [INDEX] [all]`

Options

Option	Description
`INDEX`	The index of the partition definition to delete.
`all`	If specified all current partition definitions will be removed.

`start-partitioning`

Description

Start partitioning and formatting the device according to current configuration.

Before this command is invoked no changes will be made to the selected device. If the partitioning context is exited and this command has not been called, the device will retain its previous partitioning configuration and formatting.

> **WARNING**
>
> Understand that the content on the device will be lost during the partitioning and formatting of the device. Make sure that the device does not contain any files that needs to be saved elsewhere.

Status Overview

Using the **`show`** command in media context will show an overview of our media devices:

```
example:/#> show media
MEDIA       DEVICE      LABEL          FORMAT   RW     SIZE  USED  DESCRIPTION
Active
internal    mtd:Config  -              jffs2    Yes   15.0M    5%  config,cert
external    sdb1        default        ext3     Yes    1.8G    0%  logging
Inactive
my-logs     -           -              -        -         -     -
Available
-           sdb2        my-files       ext3     -    764.0M     -
-           sdb3        -              ext2     -    764.0M     -
-           sdb4        my-label       ext2     -    100.0M     -
```

This view allows the user to immediately gain an understanding of how any connected external media relates to the media definitions. For any external media to be made available on the system, a media definition has to be created to match a specific physical media device or a partition on the device. For more information how this is achieved refer to the Access External Media Devices page.

However, the basic meaning for each of the sub-headers in the output are as follows:

- **`Active`**: A media definition that has a matching physical media device, and is as such *Active* and available to the user in the system. The **`internal`** media definition is a special case that represents the built-in flash media. All other created media definitions intend to match against a connected external media device.

- `Inactive`: Media definitions, that have been created but currently have no matching external media device connected, will be listed here. For instance, any media definition would be listed here in case no external media device have been physically connected to the device. Hence, in this case they are *Inactive* since no physical device exist that match the media definition.
- `Available`: Any physical connected external media that is not matched with a media definition will be listed here. They are *available* and need to be matched against a media definition to become *Active* and available in the system.

It is also possible to list information strictly related to the status of the connected external media device(s), using the `show devices` command in the media context:

```
example:/#> show media devices

Connected Media Devices
NAME       TYPE  NUM  LABEL                    FORMAT   SIZE     MOUNT-MEDIA
sdb        disk  -    -                        -        3.7G     -
  sdb1     part  1    default                  ext3     1912M    external
  sdb2     part  2    my-files                 ext3     764M     -
  sdb3     part  3    -                        ext2     764M     -
  sdb4     part  4    my-label                 ext2     100M     -
```

2.6.7. Packages Support

Introduction

The Packages feature allows you to extend or modify the behaviour of your device by installing special modification packages. Some examples of what packages can do include:

- Applying different factory configurations
- Enabling or configuring specific services (e.g., Profinet)
- Changing device branding
- Solving specific issues or applying workarounds (May only be possible for specific issues. Most problems would still require a firmware update to fix.)

These are just some examples of what can be done. However, there also limitations to what can actually be done or solved using a package.

Packages are intended for advanced configuration and are typically provided by Westermo or your local sales office for particular needs. If you have questions about what can be achieved with packages, or if you require a package for a specific use-case, please contact your local sales office.

> **NOTE**
>
> Currently, there are no packages available publicly for download. If you have issues with a package that you have been provided, please contact Westermo support.
>
> Again, if you have an enquiry about packages, please contact your local sales office. You can find information here[1].

Package Format and Verification

All packages must be in the `.wmopkg` format. This format supports cryptographic signing to ensure authenticity and integrity. By default, only packages signed by Westermo are accepted and can be installed and activated on the device.

If you attempt to install an unsigned or unverifiable package, the system will warn you and, by default, will not activate the package. To allow installation and activation of unsigned packages, you must enable the “Allow Untrusted” option in the bootloader configuration. On devices with a hardware anchor (secure boot), an unlock license may be required instead. For more information, see the Secure Boot unlock using tokens guide and the Boot Configuration documentation.

> **SECURITY WARNING**
>
> For security reasons, it is recommended to only install signed packages, unless you are certain of the package’s origin and purpose. If you are unsure about a package’s origin or purpose, please contact Westermo support for assistance.

Package Activation and Verification

Any package that is installed will not take effect until after a reboot has taken place. On every boot, all installed packages are individually verified to ensure they are correct and have not been tampered with. If a package fails verification, it will not be activated, unless the system is configured to allow untrusted packages.

[1] https://www.westermo.com/contact/where-to-buy

Managing Packages

Packages are managed from the `package` context, accessible from the top-level CLI exec context:

```
example:/> package
example:/package/#>
```

`install [persistent] [media <label>] [proto <ftp|tftp|scp|sftp|http|https>] [user <username>] [password <password>] address[:port] PKGNAME.wmopkg`

Description

Install a `.wmopkg` package on the device. The package can be installed from a remote source using various protocols. It is also possible to install a package from a connected external media device, such as a USB drive or SD card.

If the `persistent` keyword is used, the package will survive a factory reset and can only be removed using the `remove` command. This can be useful when a package, for instance, adjusts the factory configuration or branding of the device.

> **NO PROTOCOL OR MEDIA SPECIFIED**
>
> If no protocol or media is specified, the default behaviour will be to try and use `ftp` with `tftp` as a fallback.

> **PROVIDING CREDENTIALS**
>
> If `password` is not specified on the command line, you will be prompted for it interactively (when required by the protocol). The `user` (username) must be provided; if omitted, the system will use the name of the user currently logged in to the CLI. This avoids the need to write the password in cleartext in the command.

> **PACKAGE ACTIVATION AND VERIFICATION**
>
> Any package that is installed will not take effect until after a reboot has taken place. On every boot, all installed packages are individually verified to ensure they are correct and have not been tampered with. If a package fails verification, it will not be activated, unless the system is configured to allow untrusted packages.

Examples

```
example:/package/#> install 192.0.2.1 pkgname.wmopkg
example:/package/#> install persistent 192.0.2.1 pkgname.wmopkg
example:/package/#> install proto ftp 192.0.2.1 pkgname.wmopkg
example:/package/#> install proto tftp 192.0.2.1 pkgname.wmopkg
example:/package/#> install media external path/on/media/pkgname.wmopkg
```

continues on next page

continued

```
install [persistent] [media <label>] [proto <ftp|tftp|scp|sftp|http|https>] [user
<username>] [password <password>] address[:port] PKGNAME.wmopkg
```

Options

Option	Description
`persistent`	Make the package persistent across factory resets. Package should only be removable if the `remove` command is used.
`media <label>`	Selects the media device to source the package from. The file path is always relative to the root of the chosen media. The `<label>` specify a media device. Media(s) must be configured in the boot context (see Boot Configuration). The default `external` label matches the first partition of any connected external device (USB, SD-Card, etc.), but you can define your own. **Tips:** • To see all active media, run `do show media` in the CLI. • For setup help, see Accessing External Media Devices. **Examples** `example:/package/#>` **`install media external pkgname.wmopkg`** `example:/package/#>` **`install media my-usb relative/path/pkgname.wmopkg`**
`proto <ftp|tftp|scp|sftp|http|https>`	Select the protocol to use for downloading the package from a remote source.
`user <username>`	Username for authentication if required by the selected protocol.
`password <password>`	Password for authentication if required by the selected protocol.
`address[:port]`	The address (and optional port) of the remote server or source.
`PKGNAME.wmopkg`	The package file to install. This can be a full path on the media or remote server, but only the actual filename (not the path) will be used as the package name on the device. The file must have the `.wmopkg` extension.

`remove [PKGNAME]|[all]`

Description

Remove installed package(s) from the device.

> **DEFAULTS TO REMOVE ALL**
>
> If no option is provided after `remove`, it defaults to `all`, meaning all installed packages will be removed.

> **PACKAGE REMOVAL AND ACTIVATION**
>
> Any package that is removed will not be guaranteed to be deactivated until after a reboot has taken place.

Examples

Remove specific package:

```
example:/package/#> remove pkgname
Remove package pkgname (y/N)?y
Removed ... Changes are applied after reboot.
example:/package/#>
```

Remove all packages:

```
example:/package/#> remove all
Remove all packages (y/N)?y
Removing all packages ... Changes are applied after reboot.
example:/package/#>
```

Options

Option	Description
`PKGNAME`	The name of the package to remove. **TAB COMPLETION** You can use tab completion to help fill in the package name.
`all`	Remove all installed packages from the device.

`show`

Description

Show the list of installed packages on the device.

continues on next page

continued

`show`

Example

```
example:/package/#> show
 ┌ Packages ═══════════════════════════════════════════════════════
 │NAME             VERSION      PERSISTENT  VERIFICATION  DESCRIPTION
 │pkgname          5.26.0-rev0  False       Signed        Example
 └─────────────────────────────────────────────────────────────────

example:/package/#>
```

`list`

Description

Alias for the `show` command. Lists installed packages on the device.

Troubleshooting and Support

If you encounter issues with packages, or if you have questions about using or obtaining packages for your device, please contact your local sales office or Westermo support for assistance.

2.6.8. Port Provisioning

Provisioning defines a time frame, during which site engineer must ensure that all cables are plugged in, according to network design. When the provisioning timer has run out, ports without cables plugged in will enter disabled state and this change will be reflected in startup-config.

Introduction

Port provisioning has two stages:

1. System configurator configures provisioning in the lab and powers off the device.
2. Site engineer installs all devices on site, plugs in cables, and powers them on. After powering on, device will immediately enter provisioning state, with pre-configured timeout. When the timer runs out, ports that have no link (one or both ends of the cable are unplugged) will enter disbled state.

Provisioning ensures that no access to devices can be gained by plugging in cables to unused ports.

During operation, ports specified for provisioning will blink the leds, in order to visually indicate their involvement in the process. Additionally, each step of the provisioning is logged into a syslog.

By design, provisioning will bring all specified ports up, in order to ensure correct operation and prevent cases where cable is plugged in into a disabled port, resulting in port being disabled by provisioning, upon timeout.

Configuration

Unlike most features of WeOS, port provisioning is configured outside of the *config* block.

```
example:/#> provisioning
example:/provisioning/#>
example:/provisioning/#> enable
example:/provisioning/#> ports eth1..eth8
example:/provisioning/#> timeout 180
example:/provisioning/#> leave
```

`[no] enable`

Description

Activate provisioning. Note that the actual provisioning process will start upon next boot.

Options

Option	Description
`no`	Disable provisioning.

`[no] ports <PORT|PORT,PORT|PORT..PORT|ALL>`

Description

Ports that are to take part in provisioning.

continues on next page

continued

`[no] ports <PORT|PORT,PORT|PORT..PORT|ALL>`

Options

Option	Description
`no`	Remove configured ports from provisioning. Removes all ports if none are specified.
`PORT`	The port(s) to be included in or removed from provisioning.

`[no] mode <MODE>`

Description

Operating modes:

- one-shot : run only once, on the next device bootup
- always : run on every device bootup

Options

Option	Description
`no`	Reset the mode to the default (one-shot).
`MODE`	‘one-shot’ or ‘always’.

`[no] timeout <INTERVAL>`

Description

Period of time in seconds (upon boot, unless started explicitly) during which provisioning is active (default: 900 seconds).

Options

Option	Description
`no`	Reset the timeout to the default.
`INTERVAL`	INTERVAL in seconds (range 30-3600).

`start`

Description

Start provisioning immediately using current settings.

`stop`

Description

Stop provisioning immediately.

`show`

Description

Show summary of provisioning settings

Recommendations

Authorised individual performing initial setup should ensure that, during provisioning, no unauthorized individuals have access to devices.

2.6.9. System Settings

Introduction

The system settings under the configuration context can be used to set general information about the device, such as hostname, timezone, and other system-wide settings.

To enter the system settings context:

```
example:/#> configure
example:/config/#> system
example:/config/system/#>
```

Configuration

Hostname

The system hostname is used to identify the device in a network. It can be configured in the CLI as shown in the example below.

```
example:/#> configure
example:/config/#> system
example:/config/system/#> hostname bob
example:/config/system/#> leave
bob:/#>
```

The set of allowed characters follows the specification defined in RFC 1123. This includes letters (a-z, A-Z), digits (0-9) and hyphens (-). Additionally, underscores (_) are also allowed. The hostname cannot start or end with a hyphen or underscore, and cannot exceed 63 characters.

`[no] hostname [HOSTNAME]`

Description

System hostname.

Options

Option	Description
`no`	Reset to the default hostname.
`HOSTNAME`	Hostname string (1..63 characters). RFC 1123 compliant. Allowed characters are: • Letters (A-Z, a-z), numbers (0-9), dash (-), and underscore (_). • Hostname may not start or end with dash (-) or underscore (_).

Domain Name

The system domain name is used in conjunction with the hostname to form the fully qualified domain name (FQDN) of the device. The domain name can be configured as shown in the example below.

```
example:/#> configure
example:/config/#> system
```

```
example:/config/system/#> domain example.com
example:/config/system/#> leave
example:/#>
```

Setting the system domain name has the following practical effects:

- The domain name is added to the default search domain for DNS resolution. This means that when a user tries to resolve a hostname without a domain, the system will automatically append the configured domain name and attempt to resolve it. For example, if the domain is set to `example.com` and a user tries to resolve `host1`, the system will additionally attempt to resolve `host1.example.com`. This is true for both local name resolution, as well as if the system is configured as a DNS resolver for other devices on the network.
- DNS records for the local device will include the domain. Because of this, reverse resolution of the device's IP address will return the FQDN. As an implementation note, the device FQDN is added to the record for the IP address 127.0.1.1 in the local DNS resolver configuration.
- If the system is configured to use the FQDN for syslog messages, the domain name will be included in the hostname part of the syslog messages, as described in RFC 5424. See the logging configuration documentation for more details.
- The host and domain names are configured in the underlying Linux system. For instance, if applications use the `getaddrinfo()` system call, or the `dnsdomainname` command in the shell, they will return the configured hostname and domain name.

> **NOTE**
>
> The above is specific to the static configuration of the system domain name. If the device is configured to obtain its hostname and/or domain name dynamically via one or more DHCP leases, only the first and second points will apply. Host and domain names obtained via DHCP will be added to the local DNS configuration, but they will not be used in syslog messages.

`[no] domain [DOMAIN]`

Description

System domain name.

Options

Option	Description
`no`	Remove any configured system domain name.
`DOMAIN`	Domain name string (1..63 characters). Allowed characters are letters (A-Z, a-z), numbers (0-9), dash (-), and dot (.). Domain name must not start or end with a dash (-) or dot (.).

Description

The system description provides a brief text describing the device or its role in the network.

```
example:/config/system/#> description "Main office router"
```

`[no] description [DESCRIPTION]`

Description

System description.

Options

Option	Description
`no`	Reset to the default description.
`DESCRIPTION`	Description text.

Location

The system location describes where the device is physically located.

```
example:/config/system/#> location "Rack 5, Room 101, Building A"
```

`[no] location [LOCATION]`

Description

Location of the system.

Options

Option	Description
`no`	Remove the configured location.
`LOCATION`	Location text.

Contact

The contact information for the system. It is typically used to specified an email address or the name of the person responsible for maintaining the device.

```
example:/config/system/#> contact "admin@example.com"
```

`[no] contact [EMAIL]`

Description

Contact information for the system.

Options

Option	Description
`no`	Remove the configured contact information.
`EMAIL`	Contact text (e.g., email address).

Message of the Day (MOTD)

The message of the day (MOTD) is a banner displayed when a user logs in. This can be used to display policy information, warnings, or any other message for users accessing the device.

```
example:/config/system/#> motd "Welcome to the network gateway.\nContact admin@example.com for support."
```

For flexible formatting, certain control characters are allowed:

Control Character	Description
`\a`	BELL
`\b`	Backspace
`\t`	Horizontal TAB
`\n`	New line
`\v`	Vertical TAB
`\f`	Form Feed
`\r`	Carriage Return
`\e`	ANSI escape
`\\`	Backslash

`[no] motd [MESSAGE]`

Description

System postlogin message.

Options

Option	Description
`no`	Reset to the default message.
`MESSAGE`	Message text, enclosed in quotes.

Timezone

The timezone setting controls the local time displayed by the device. Daylight saving time (DST) adjustments are applied automatically based on the selected timezone.

The timezone can be configured as below:

```
example:/config/system/#> show timezone Europe
Europe/Amsterdam
Europe/Berlin
Europe/London
Europe/Stockholm
 ...
example:/config/system/#> timezone Europe/Stockholm
```

`[no] timezone <timezone> [no-dst]`

Description

System timezone.

continues on next page

continued

`[no] timezone <timezone> [no-dst]`

Options

Option	Description
`no`	Reset to the default timezone (UTC).
`timezone`	A valid timezone identifier (e.g., Europe/Stockholm)
`no-dst`	Disable daylight saving time adjustments

`list-timezones`

Description

List all available timezones

SSL Security Level

The SSL security level sets a system-wide floor for the cryptographic strength accepted in TLS connections. It is applied by OpenSSL and restricts key sizes, signature algorithms, protocol versions and cipher suites. Raising the level therefore reduces the set of cipher suites that can be selected, for example for an LDAP remote server.

```
example:/#> configure
example:/config/#> system
example:/config/system/#> ssl-security-level 3
example:/config/system/#> leave
```

:::warning Important Raise the level with care. From level 3 and upwards RSA, DSA and DH keys shorter than 3072 bits are rejected, and level 4 and 5 raise that bar to 7680 and 15360 bits respectively. Certificates that WeOS generates itself, such as the self-signed certificate the Web server uses for HTTPS by default, are RSA 2048 and are therefore not accepted at those levels. Before selecting level 3 or higher, make sure every certificate involved, on the device as well as on the peer, is strong enough, otherwise the affected connections will fail to establish. :::

`[no] ssl-security-level <NUMBER>`

Description

System-wide SSL security level. Default is 2.

Options

Option	Description
`no`	Reset to the default security level (2).
`NUMBER`	Security level, 1 to 5.
`Level 1 (weak)`	The security level corresponds to a minimum of 80 bits of security. Any parameters offering below 80 bits of security are excluded. As a result RSA, DSA and DH keys shorter than 1024 bits and ECC keys shorter than 160 bits are prohibited. Any cipher suite using MD5 for the MAC is also

continues on next page

continued

`[no] ssl-security-level <NUMBER>`

Option	Description
	prohibited. Any cipher suites using CCM with a 64 bit authentication tag are prohibited. Note that signatures using SHA1 and MD5 are also forbidden at this level as they have less than 80 security bits. Additionally, SSLv3, TLS 1.0, TLS 1.1 and DTLS 1.0 are all disabled at this level.
`Level 2 (default)`	Security level set to 112 bits of security. As a result RSA, DSA and DH keys shorter than 2048 bits and ECC keys shorter than 224 bits are prohibited. In addition to the level 1 exclusions any cipher suite using RC4 is also prohibited. Compression is disabled.
`Level 3 (strong)`	Security level set to 128 bits of security. As a result RSA, DSA and DH keys shorter than 3072 bits and ECC keys shorter than 256 bits are prohibited. In addition to the level 2 exclusions cipher suites not offering forward secrecy are prohibited. Session tickets are disabled.
`Level 4 (highly restrictive)`	Security level set to 192 bits of security. As a result RSA, DSA and DH keys shorter than 7680 bits and ECC keys shorter than 384 bits are prohibited. Cipher suites using SHA1 for the MAC are prohibited.
`Level 5 (extreme hardening)`	Security level set to 256 bits of security. As a result RSA, DSA and DH keys shorter than 15360 bits and ECC keys shorter than 512 bits are prohibited.

2.6.10. Firmware Upgrade

A WeOS unit holds two types of firmware:

- System firmware: The system firmware holds the operating system, which is what we usually refers to when we say WeOS. For robustness purposes, a WeOS unit typically holds two separate system firmware images.
  - Primary firmware image: The primary firmware image (or primary image) contains the system firmware loaded by default by the bootloader.
  - Backup firmware image: The backup firmware image (also known as backup image or secondary image) contains the system firmware image loaded in case an error is encountered while loading the primary image.
- Bootloader: The bootloader firmware (or simply ”bootloader”) is the basic firmware run to bootstrap the system. The bootloader will in turn load the system firmware (trying the primary image first).

It is possible to upgrade both the system firmware (primary and secondary image) and the bootloader firmware. There is no general guarantee that an older system firmware can be loaded into the switch, i.e., downgrade is not generally guaranteed to work. However, if the firmware is downgraded it is recommended to reboot the switch once the old firmware has been installed. When the switch comes up with the old firmware, copy the factory default configuration to the running configuration.

> **SEE ALSO**
>
> Related Documentation
> - Firmware Checksum

Upgrading firmware and bootloader

WeOS firmware and bootloader can be downloaded from www.westermo.com[1]. The WeOS firmware and bootloader can be upgraded using a common ”pkg” file or using individual firmware and bootloader files per WeOS product.

> **NOTE**
>
> The release package (`.pkg`) is the supported upgrade artifact. The internal firmware image inside the package uses a CPU-architecture name (for example `WeOS-armv8-<release>.itb`) but this does not affect the `Platform` value shown by `show sys`, which continues to display the hardware platform name (for example `Envoy`). Use `upgrade validate` to confirm a package is compatible with the device before uploading.

Web Interface Access During CLI Upgrade

While a firmware upgrade is running from the CLI, access to the entire web interface is blocked. TThis measure exists to prevent unsafe user interactions during the Upgrade process. During this time, the following message is displayed on the web interface: “Access not allowed. This can be due to an ongoing update. The page will refresh once the update is complete.” Once the upgrade is finished, the page will automatically refresh to restore access.

[1] http://www.westermo.com

Commands

```
upgrade [force] [no-reboot] [validate] <all|pri|sec|boot> [media <label>] [proto
<ftp|tftp|scp|sftp|http|https>] [user <username>] [password <password>]
address[:port] FILE
```

Description

Starts an upgrade process based on the parameters given.

Options

Option	Description
`force`	A keyword can be used to bypass the image signature check and certain version checks. It also makes it possible to upgrade individual images not using 'all' in this release. This should only be used as a last resort and MAY BRICK THE DEVICE if the wrong imaged is used. Please verify carefully that you are using the correct image for the device being upgraded.
`no-reboot`	If the running image is upgraded it will result in a reboot after upgrade completes. This option can be used to prevent that behaviour. Using the "all" target will always result in a reboot after upgrade, meaning this option has no effect.
`validate`	Used to validate the given upgrade image can be used to perform an upgrade. This option will not perform an upgrade, only download and validate the image.
`pri`	Setting the target to be the primary (main) flash area.
`sec`	Setting the target to be the secondary (backup) flash area.
`boot`	Setting the target to be the bootloader firmware flash area.
`all`	The "all" target is only available for packages and is used in specific cases where many actions must be done at once. Not all upgrades or targets support the all keyword. Device will always reboot after flashing.
`media`	Label of the media (USB) to import from. Alternative to specifying a file transfer protocol and address.
`proto`	Protocol to use. ftp, tftp, scp, sftp, http or https.
`user`	Username if used by protocol.
`password`	Password if used by protocol.
`address[:port]`	IP-address and optional port of the server from which the image/package should be retrieved.
`FILE`	The name of the image or package (pkg) to be used during upgrade.

Firmware Upgrade using FTP or TFTP

WeOS firmware upgrade can fetch the upgrade image from an external FTP or TFTP server. The protocol variant can be explicitly specified through the use of a URI, for example (from WeOS CLI):

```
example:/#> upgrade sec tftp://192.168.122.1/firmware.pkg
```

Or it can be determined automatically. In this case WeOS first tries FTP and if it fails to connect, falls back to TFTP:

```
example:/#> upgrade sec 192.168.122.1 firmware.pkg
```

If username and password are not supplied for an FTP connection, WeOS will use anonymous FTP. Username and password can be supplied using the syntax:

```
example:/#> upgrade sec ftp://username:password@host/path/to/firmware.pkg
```

Firmware upgrade can also be initiated from the WeOS Web interface. In that case selecting the "FTP/TFTP" option will make WeOS try anonymous FTP first, then if that fails, TFTP.

!!! Note: The WeOS CLI upgrade command also accepts URI prefixes `http://` and `usb://`, but not `scp://`.

Firmware Upgrade using target 'all'

In order to move all firmwares in a consistent way, the target 'all' can be used. It will upgrade bootloader and system images in a determined sequence. As usual, power must not be interrupted.

```
example:/#> upgrade all 192.168.122.1 firmware.pkg
```

Product Model Compatibility

During the upgrade process, WeOS automatically verifies whether the product model is supported by the version being upgraded to. Incompatibility between the product model and the WeOS version will result in the upgrade being aborted. Such cases usually arise when downgrading to a WeOS version that predates the launch of the product model.

Deprecated Command Syntax

The upgrade command previously supported a different way of specifying the address of the upgrade image. This method is now deprecated and support may be removed in future releases. The deprecated syntax is shown below for reference.

```
upgrade [force] [no-reboot] <all|pri|sec|boot> <[URI://][ADDRESS[:PORT]/]FILE |
ADDRESS FILE]
```

Description

Starts an upgrade process based on the parameters given.

Options

Option	Description
`force`	A keyword can be used to bypass the image signature check and certain version checks. It also makes it possible to upgrade individual images not using 'all' in this release. This should only be used as a last resort and MAY BRICK THE DEVICE if the wrong imaged is used. Please verify carefully that you are using the correct image for the device being upgraded.
`no-reboot`	If the running image is upgraded it will result in a reboot after upgrade completes. This option can be used to prevent that behaviour. Using the "all" target will always result in a reboot after upgrade, meaning this option has no effect.

continues on next page

continued

```
upgrade [force] [no-reboot] <all|pri|sec|boot> <[URI://][ADDRESS[:PORT]/]FILE |
ADDRESS FILE]
```

Option	Description
`validate`	Used to validate the given upgrade image can be used to perform an upgrade. This option will not perform an upgrade, only download and validate the image.
`pri`	Setting the target to be the primary (main) flash area.
`sec`	Setting the target to be the secondary (backup) flash area.
`boot`	Setting the target to be the bootloader firmware flash area.
`all`	The “all” target is only available for packages and is used in specific cases where many actions must be done at once. Not all upgrades or targets support the all keyword. Device will always reboot after flashing.
`URI`	WeOS VFS URIs can be used to retrieve the firmware file.
`ADDRESS`	IP-address of the server from which the image/package should be retrieved.
`FILE`	The name of the image or package (pkg) to be used during upgrade.

2.6.11. Firmware Checksum

WeOS has two firmware images that exist on the device, the primary and the secondary. Both of these images have an associated checksum that is used to verify the integrity of the firmware.

> **SEE ALSO**
>
> Related Documentation
> - Firmware Upgrade

Overview

By default, both of these firmware images have an associated background task started on boot that is responsible for calculation of their individual checksums. The amount of time it takes to finalize the calculation may vary depending on the specific platform.

> **SYSTEM LOAD**
>
> These calculations may be somewhat CPU intensive while they are running. It is possible that this extra load can cause slight latency in specific system operations during startup, until both of the individual calculations have been finalized.

The calculated checksum for each of the firmware images, will be compared against the checksum that is present in the header of the firmware image. If the calculated checksum does not match the checksum in the header, the firmware image is considered invalid.

Viewing the Checksum Status

The checksum status for each of the individual firmware images can easily be observed in the `system-information` status output, which is viewable from the CLI in the following manner:

```
example:/#> show system-information
[ ... ]

BOOT INFORMATION
Boot loader ver.    : 2024.03.0-1       Active firmware     : RAM
Primary ver.        : 5.24.0            Primary CRC         : OK ←
Secondary ver.      : 5.24.0            Secondary CRC       : OK ←
Config media        : internal          Config file         : startup
Certificate media   : internal

[ ... ]
example:/#>
```

In the above example, we can see how the output will be presented when the checksum for both the primary (**Primary CRC**) and secondary firmware (**Secondary CRC**) images have calculated and successfully validated.

The following states are possible for the **CRC** for each of the individual firmware images:

State	Description
`Unchecked`	The checksum has not yet been calculated.
`Checking ...`	The calculation of the checksum is in progress.
`OK`	The checksum has been calculated and has been correctly validated.

State	Description
`Invalid`	The checksum has been calculated and is invalid.

Table 1: The possible states for the firmware checksum calculation.

Configuring the Automatic Checksum Calculation

As mentioned above, by default the Automatic Checksum Calculation is enabled. Meaning that as soon as the device is started, the background tasks for the checksum calculations will be started.

However, it is possible to disable this if desired. The only real reason why this would be necessary is if the slight possible latency that may be introduced, directly after startup, is causing some issues in the system during that period of time.

> **MANUAL START POSSIBLE**
>
> If the automatic checksum calculation is disabled, it can still be manually started when it may be needed. The main function in the system that will require the checksum to be present is is the firmware upgrade. However, it is still possible to initiate a firmware upgrade even if the checksum calculation is not present, but an additional confirmation check must be acknowledged in this case.

Configuration

The setting controlling the automatic checksum calculation is available in the system context from the top level configuration context:

```
example:/#> configure
example:/config/#> system
example:/config/system/#>
```

`[no] automatic-firmware-checksum`

Description

Enable/Disable automatic firmware checksum calculation.

By default the device calculates the firmware checksum automatically. This setting can be used to disable that behaviour.

Even if the automatic calculation is disabled, it can still be initiated manually using the 'firmware-checksum' command in the exec context.

> **NOTE**
>
> The firmware checksum calculation can be somewhat heavy during startup, which is why it is possible to disable it, if desired.

Default: Enabled

Example

```
example:/config/system/#> automatic-firmware-checksum
example:/config/system/#> no automatic-firmware-checksum
```

continues on next page

continued

`[no] automatic-firmware-checksum`

Options

Option	Description
`no`	Disable the automatic checksum calculation.

Manually Starting the Checksum Calculation

If the automatic checksum calculation is disabled, the checksum calculation can still be initiated on demand. When the command is invoked, the system will start a background task that will calculate the checksum.

The command is available from the top level exec context:

```
example:/#>
```

`firmware-checksum [pri|sec]`

Description

Check the CRC of a firmware image on the device.

This command checks the CRC of a firmware image on the device. The image must be in the primary or secondary firmware slot. The command will check the CRC of the image and compare it to the CRC stored in the image header.

The command can be used with the optional parameters `pri` or `sec` to specify which image to check. If no parameters are given, the command will check the CRC of both images.

The calculation of the CRC will be started in the background and the command will return immediately. If the CRC calculation is already done, the status for the image will be shown.

> **ALREADY CALCULATED**
>
> If the CRC has already been calculated for the given firmware image, the status for the image will be shown.

Example

```
example:/#> firmware-checksum
example:/#> firmware-checksum pri
example:/#> firmware-checksum sec
```

Options

Option	Description
`pri`	Check the CRC of the primary firmware image.
`sec`	Check the CRC of the secondary firmware image.

2.6.12. Watchdog

Introduction

The watchdog is a functionality that can help the system recover from various possible malfunctions that could cause the system to enter a state where it is not able to operate correctly. Anything from a hardware fault or program errors may cause the watchdog to trigger.

The watchdog is essentially just a timer that will cause the system to reset if it runs out. In order to prevent the watchdog from triggering, the system will periodically “kick” the watchdog in order to inform it that the system is still operating within established parameters.

Overview

WeOS allows users to perform adjustments to the watchdog timeout along with the kick interval. The watchdog timeout is not allowed to be set to a value equal or smaller than the kick interval.

In addition, the system also allows for disabling the watchdog. However, this is not something that is recommended. It is suggested that if adjustments to the watchdog is deemed necessary for any reason, tweak the timers instead.

Reset Information

Further, the watchdog in the system can also provide some useful information to the user. One of the more prominent of these is information on the last reset of the system. The information will be provided with a reset reason and cause. The following are the possible reset reasons and causes that the watchdog is able to provide:

Reset Cause	Reset Reason
0	None
1	System OK
2	Failed subscription
3	Failed kick
4	Failed unsubscription
5	Failed to meet deadline
6	Forced reset
8	Descriptor leak
9	Memory leak
10	CPU overload

Monitors

The watchdog also provides three different types of monitors, that are enabled by default as well. These monitors are configured to log information at provided intervals. The following are the three monitors supported by the watchdog:

- **loadavg:** Monitoring the average load on the system.
- **meminfo:** Monitoring the memory in order to detect memory leaks.
- **filenr:** Monitoring file descriptors in order to detect file descriptor leaks.

Currently only these three default monitors are configurable for the watchdog.

Configuration

Watchdog Settings

The watchdog can be configured from the top-level configuration context in the CLI.

```
example:/#> configure
example:/config/#> watchdog
example:/config/watchdog/#>
```

Syntax

`[no] enable`

Description

Enable or disable watchdog daemon.

Default: enabled.

Options

Option	Description
`no`	Disable the watchdog.

`[no] interval [SEC]`

Description

Set watchdog kick interval, seconds.

The time between kicks from watchdogd to the kernel wdt driver. Recommended to set to 1/3 of the kernel wdt timeout.

Min: 1 sec, **Max:** 128 sec, **Default:** 20 sec.

Options

Option	Description
`no`	Reset to the default value: **20**.
`SEC`	Time in seconds within the allowed range: **1-128**.

`[no] timeout [SEC]`

Description

Set kernel wdt timeout, seconds.

The timeout before the kernel watchdog timer elapses. This can be an external hardware watchdog, depending on the device. Recommended to set to 10 seconds or greater.

Min: 1 sec, **Max:** 128 sec, **Default:** 60 sec.

continues on next page

continued

`[no] timeout [SEC]`

Options

Option	Description
`no`	Reset to the default value: **60**.
`SEC`	Time in seconds within the allowed range: **1-128**.

`[show] monitor [loadavg | meminfo | filenr]`

Description

Handle watchdog monitors.

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`show`	Show the configured monitors.
`loadavg`	Monitor for average CPU load.
`meminfo`	Monitor for memory usage.
`filenr`	Monitor for file descriptor count.

Monitor Settings

The monitors can be configured from the watchdog configuration context in the CLI. When entering the monitor context a existing monitor name must be provided.

```
example:/#> configure
example:/config/#> watchdog
example:/config/watchdog/#> monitor NAME
example:/config/watchdog/monitor-NAME/#>
```

Syntax

`[no] enable`

Description

Enable or disable the monitor.

Default: Enabled.

Options

Option	Description
`no`	Disable the monitor.

`[no] interval [SEC]`

Description

Set the monitor interval.

Default: 3600 seconds.

Options

Option	Description
`no`	Reset to the default value..
`SEC`	Time value in seconds.

`[no] description [STRING]`

Description

Free form description of this watchdog monitor.

Options

Option	Description
`no`	Remove any defined description.
`STRING`	Free from string.

`[no] logmark`

Description

Enable or disable logging to at the configured `interval`.

Default: Enabled.

Options

Option	Description
`no`	Disable logging.

`[no] warning [PERCENTAGE]`

Description

Set the percentage when to trigger a warning.

If this threshold is exceeded, a warning will be logged at the next `interval` invocation.

Options

Option	Description
`no`	Disable the warning.
`PERCENTAGE`	Percentage value in the range `1` - `100`.

continues on next page

continued

`[no] warning [PERCENTAGE]`

Option	Description
	NOTE For a `loadavg` monitor, the maximum percentage value is dependent on the number of CPU cores in the product. As an example, two cores would produce a max value of 200%.

Status

The current status of the watchdog can be observed from the top level in the CLI:

```
example:/#> show watchdog
Status              : Enabled
Timeout (sec)       : 60
Kick interval       : 20
Watchdog ID         : 0
Boot counter        : 821
Reset counter       : 1
Reset date          : 2019-06-18T09:05:12Z
Reset reason        : 6 - Forced reset
PID                 : 1
Label               : init
```

The boot counter will be incremented on all boots, both Power on Reset (cold boot) and after a watchdog reset. The reset counter is only incremented on watchdog reset. And set to 0 on a Power on Reset. A reboot from cli will also increment the reset counter since it uses the watchdog for power cycling. The reset reason will differ though. A reboot will be “6 - Forced reset” and real watchdog reset will show as “7 - Unknown failure”. See Reset Information for reset reason for watchdogd.

2.7. Alarm, LEDs and Logging

2.7.1. Alarm

Introduction

Alarms can be used in order to specify specific actions to take when a particular event occurs in the system.

Overview

```
Alarm Triggers <----------------------> Alarm Actions
      ^                                       ^
      |                                       |
      |                                       |
      v                                       v
Alarm Sources                           Alarm Targets
```

Figure 1: The image describes the relations between the different aspects that together constitute the alarm handling functionality in the system. Alarm triggers monitor the state of the provided source and defines conditions that should invoke an associated alarm action. The invoked alarm action specifies what alarm target(s) to use in order to notify the operator.

The alarm handling functionality in the system makes use of the following terminology:

- *Alarm Sources:* An *alarm source* is an object being monitored by an *alarm trigger*. This could be, for instance, the link status (up/down) of an Ethernet port or the temperature value of a temperature sensor. Alarm sources are described further in a section below.
- *Alarm Triggers:* An *alarm trigger* will monitor a provided *alarm source* and define the conditions for when specific alarm events occur, in other words, when the trigger becomes *active* (alarm situation) or *inactive* (normal operation).
- *Alarm Actions and Alarm Targets:* When an alarm event occurs, the operator can be informed in a number of ways. This is defined by a specific *alarm action* that is associated with any give *alarm trigger*. The *alarm action* in turn specify one or more *alarm targets* that will be used to notify the operator. As an example this could be SNMP traps, logging, or a front panel status LED. Alarm actions and targets are described in more detail below.

Alarm Triggers and Alarm Sources

The *alarm trigger* defines rules for when a monitored source should generate an alarm event. The alarm trigger also needs to specify an *alarm action* to invoke when an alarm event occurs.

The system supports the following *alarm sources* to be associated with any given *alarm trigger*:

- *Link Alarm:* It is possible to configure link alarm triggers to react when the status of a link changes (goes up or down). A link alarm trigger can be associated with one or more Ethernet ports, and will be set to active when any of the associated ports matches the alarm condition (link up or down).
- *Temperature:* Temperature alarm triggers can be configured to react when the temperature rises above (or falls below) some defined threshold.
- *Power:* A power trigger can monitor one or more power feed sensors. Most WeOS products have two power feeds (single power supply), with a sensor for each power feed. Typically a single power trigger is used to monitor both power feed sensors.
- *Digital-In:* Alarms can be triggered depending on the presence of input voltage/current on the Digital-In pins of the Digital I/O connector.

- *Ping:* A connectivity checker, sends an ICMP ping in a configurable interval.
- *HW-Failure:* Hardware alarms triggers notifies that the unit has detected a hardware failure (typically if an unsupported SFP is inserted).
- *RING:* The RING status trigger will react when an ring is open or closed (intact). Used for FRNT and MRP ring status.
- *PoE Power Usage:* On units supporting Power Over Ethernet (PoE), alarms can be triggered when the total power usage raises above (or falls below) a configured threshold.
- *Media:* On units supporting external media (USB or SDcard) alarms can be triggered if the external device is available or unavailable.
- *Media-threshold:* On units supporting external media (USB or SDcard) alarms can be triggered if the used space on the media exceeds the configured threshold.
- *Storm Detection:* The storm detection trigger will react if a storm has been detected on a configured port that has been placed into an error-disabled state.
- *Health Monitor:* The health monitor trigger will react if health monitor places the port in the error-disabled state because of excessive link state changes.

Alarm thresholds

For the trigger to know when an alarm event has occurred, threshold values for the monitored alarm sources must be configured.

Alarm sources which are 'binary' to their nature (link up/down, power up/down, digital-in high/low, used space above / below etc.) have thresholds defined implicitly.

For sources which can take values in a wider range (temperature, PoE usage, etc) the alarm thresholds should be configured. Two thresholds are used, a *rising* threshold and a *falling* threshold. Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.

However, once a rising alarm event has occurred, a new rising alarm event cannot be generated (for that alarm source) before the value has fallen down to the falling threshold (and vice versa). Thus, the use of separate rising and falling thresholds creates a *hysteresis* mechanism, which avoids generating multiple alarm events when a monitored value fluctuates around the alarm threshold.

Additional details on threshold settings and properties:

- The rising threshold cannot be set lower than the falling threshold.
- It is possible to use the same value for the rising and falling thresholds.
- Rising alarm events occur if the current sample value is equal or above the rising threshold, and the previously sampled value was below the rising threshold. A rising alarm event will also occur if the first sampled value is equal or above this threshold, and the condition variable is configured as rising (or any of its equivalents: high or up).
- Falling alarm events occur if the current sample value is equal or below the falling threshold, and the previously sampled value was above the falling threshold. A falling alarm event will also occur if the first sampled value is equal or below this threshold, and the condition variable is configured as falling (or any of its equivalents: low or down).

© 2026 Westermo Network Technologies AB

Alarm Actions and Alarm Targets

All *alarm actions* in the system needs to be associated with one or more *alarm targets*, these dictate how the activated alarm will be presented to the operator:

- *LED:* here are front panel LEDs which can indicate status of specific ports or protocols. There is also a general status LED, which shows a green light when the unit is operating 'OK', but shows a red light as soon as any of the associated alarm triggers becomes active. Thus, the 'ON' LED provides a summary alarm function.
- *Log:* Log status change to syslog.
- *SNMP:* Alarms can be configured to generate SNMP traps. Similiar to the 'ON' LED, a summary alarm trap can be enabled to send a trap with a warning OID when there's any active alarm and send a trap with an OK OID if all alarms are inactive.
- *Digout:* On units equipped with a Digital I/O contact, the status relay pins (Digital-Out) can be used as an alarm target. Similar to the 'ON' LED, the status relay provides a summary alarm function, where the state of the 'gate' (closed/open) depends if the switch is operating 'OK', or if there is an issue (any alarm trigger active or when the unit has no power). The product may have multiple status relay pinouts, as described in the product user guide. These are referred to as *normally closed* (NC) or *normally open* (NO), where the word *normally* refers to the behaviour when a unit has no power (rather than if it operating OK nor not).
  - Normally Closed (NC): The NC gate will be *closed* when the unit is unpowered, pre-operational (booting), or when one or more alarms are active. The NC gate will be *open* when the unit is operational and no alarm is active.
  - Normally Open (NO): The NO gate will be *open* when the unit is unpowered, pre-operational (booting), or when one or more alarms are active. The NO gate will be *closed* when the unit is operational and no alarm is active.
- *Port:* Control the admin status of a port.
- *Iface:* Control the admin status of an interface.

Configuration

The alarm settings can be found at the top level in the configuration context.

```
example:/#> configure
example:/config/#> alarm
example:/config/alarm/#>
```

`[no] action [INDEX]`

Description

Create, update or remove an alarm action (profile). Use `action INDEX` to enter the Alarm Action Configuration context and create a new or update an existing action.

> **NOTE**
>
> Will enter a sub-context. Information on this context can be found in a section below.

continues on next page

continued

`[no] action [INDEX]`

Options

Option	Description
`no`	Remove the specified action.
`INDEX`	Numerical index value of the action to access, create, or remove.

`[no] trigger [INDEX] [TYPE]`

Description

Create, update or remove an alarm trigger configuration.

> **NOTE**
>
> Will enter a sub-context. Information on this context can be found in a section below.

Options

Option	Description
`no`	Remove the specified trigger.
`INDEX`	Numerical index value of the trigger to access, create, or remove.
`TYPE`	Type of trigger to create. Specified in the sections above. The available trigger types can also be seen with the command `show types`.

`[no] types`

Description

Show available trigger types.

Options

Option	Description
`no`	Not applicable for this command.

`[no] summary`

Description

Summary alarm trap will be sent through SNMP.

Options

Option	Description
`no`	Will disable summary alarm trap.

Trigger Configuration

Trigger configuration is accessed from the top level of the alarm context:

```
example:/config/alarm/#> trigger TYPE
example:/config/alarm/trigger-1/#>
```

Where TYPE is the type of trigger that should be created. The types are described above when talking about *Alarm Sources*. The available types can also be seen with the *types* command in the cli:

```
example:/config/alarm/#> types
link-alarm
temperature
power
digin
ping
ring
poe
media
media-threshold
Storm-Detection
Health-Monitor
```

> **NOTE**
>
> Depending on the type of trigger selected, additional configuration options may appear in the *Trigger* configuration context.

Common Settings

`[no] enable`

Description

Enable or disable the trigger.

Options

Option	Description
`no`	Disable the trigger.

`[no] action [INDEX]`

Description

Specify the action (profile) to be invoked when this trigger detects an alarm event.

Default: Index 1 (default action).

Options

Option	Description
`no`	Disable the mapping to an alarm action.
`INDEX`	Index of a specific action to associate with the trigger.

`[no] condition [high|low] | [unavailable|available]`

Description

Define whether the `high | unavailable` or `low | available` trigger state should be considered the alarm state, while the other is considered the normal state.

> **NOTE**
>
> Some triggers, such as power triggers, have a static (predefined) alarm condition setting. (Power triggers have condition set to low). For other triggers, the alarm condition setting is configurable.

Default: Differs between different trigger types.

Options

Option	Description
`no`	Reset option to default.

`[no] severity [LEVEL] | [active <LEVEL>] [inactive <LEVEL>]`

Description

Specify the severity level of active and inactive alarm events detected by this trigger.

Active and inactive severity levels can be configured together or independently.

Default: `active` to `warning` and `inactive` to `notice`.

Options

Option	Description
`no`	Set severity to level `none`. Alarm events with severity `none` will not cause SNMP traps to be sent or events to be logged, however, such events can still affect the status relay (digital-out) and ON LED targets.
`LEVEL`	The levels use standard syslog levels: • `none` : No severity. • `debug` : Debug-level message. • `info` : Informational message. • `notice` : Normal but significant condition. • `warning`: Warning condition. • `err` : Error condition. • `crit` : Critical condition. • `alert` : Immediate action required. • `emerg` : System is unusable.

Depending on the type of alarm trigger, the way to specify the *alarm source* will differ. For example, for a link-alarm trigger, the associated *port(s)* must be defined.

`[no] threshold <NUM> | [rising <NUM>] [falling <NUM>] [percent|KB|MB|GB]`

Description

(For triggers with non-binary values for alarm source, such as temperature or PoE power utilisation triggers.) Two thresholds are used; a rising threshold and a falling threshold. Alarm events will be generated when reaching the rising threshold on the way up, and the falling threshold on the way down.

Options

Option	Description
`no`	Reset thresholds to default.

Threshold units:

- `threshold <NUM> [percent]` — Set the threshold as a percentage of total space.
- `threshold <NUM> [KB|MB|GB]` — Set the threshold in bytes.

Link alarm

`[no] port <PORT|PORTS>`

Description

(For 'link-alarm' trigger) Set which port(s) to trigger alarm on.

The alarm will be set to active if any of the specified ports are triggered by the condition (link up or down).

Options

Option	Description
`no`	Remove port(s).

```
example:/config/alarm/#> trigger link-alarm
example:/config/alarm/trigger-1/#> port eth2
example:/config/alarm/trigger-1/#> show
ID              : 1
Type            : link-alarm
Status          : Enabled
Port            : eth2
Severity        : active warning, inactive notice
Condition       : low
Action          : 1
example:/config/alarm/trigger-1/#>
```

Temperature power or digin

`[no] sensor <NUM>`

Description

(For 'temperature', 'power' and 'digin' triggers) Set which sensor to trigger alarm on.

continues on next page

continued

`[no] sensor <NUM>`

Options

Option	Description
`no`	Remove sensor(s).

Ring Trigger

`[no] protocol <frnt|mrp|hsr|prp>`

Description

(For 'ring' triggers) Set which ring protocol to trigger alarm on. Used with 'ring-id' setting to define the ring protocol and instance. Default: FRNT

Options

Option	Description
`no`	Reset protocol to default.

`[no] ring-id <NUM>`

Description

(For 'ring' triggers) Set which ring-id to trigger alarm on. Used with 'protocol' setting to define the ring protocol and ID. For FRNTv0 rings, use ring-id '0' (zero).

Options

Option	Description
`no`	Remove ring-id.

Ping Settings

`[no] peer <FQDN|IPADDR>`

Description

Specify the name/ address of the target used to ping

Options

Option	Description
`no`	Remove the configured peer.

`[no] outbound <Interface|IPADDR>`

Description

Specify the interface or IP address to use as the source for outbound traffic.

continues on next page

continued

`[no] outbound <Interface|IPADDR>`

Options

Option	Description
`no`	Reset the source to the default gateway for the peer

> **NOTE**
>
> During testing, it was observed that when there is no clear route to the peer, the ping command works more reliably if the source address is specified as an explicit IP address rather than using the interface name.

`interval <SEC>`

Description

Set the frequency (in seconds) between ping samples. **Default:** 10

`number <NUM>`

Description

Set the number of packets to send/receive before evaluating the alarm condition. **Default:** 3

`initial-state <ok|warning>`

Description

Set the initial alarm state for the trigger.

PoE Trigger

`sensor <NUM>`

Description

Specify which PoE sensor(s) to monitor.

`threshold <NUM> | [rising <NUM>] [falling <NUM>] [percent|KB|MB|GB]`

Description

Set the power usage threshold for alarm activation.

Media Trigger

`media-name <NAME>`

Description

Specify the name of the external media (USB/SD card) to monitor.

`condition <available|unavailable>`

Description

Set the alarm condition based on media availability.

`interval <SEC>`

Description

Set the frequency between samples. **Default:** 10

Media-Threshold Trigger

`media-name <NAME>`

Description

Specify the name of the external media to monitor.

`threshold <NUM> [percent|KB|MB|GB]`

Description

Set the used space threshold for alarm activation. If used space exceeds the threshold, an alarm will be triggered.

- `threshold <NUM> [percent]` — Set the threshold as a percentage of total space.
- `threshold <NUM> [KB|MB|GB]` — Set the threshold in bytes.

`interval <SEC>`

Description

Set the frequency between samples. **Default:** 10

Storm Detection

`[no] port <PORT|PORTS>`

Description

Set which port(s) to trigger alarm on.

The alarm will be set to active if any of the specified ports are triggered by storm control.

Options

Option	Description
`no`	Remove port(s).

Health Monitor

`[no] port <PORT|PORTS>`

Description

Set which port(s) to trigger alarm on.

The alarm will be set to active if any of the specified ports are triggered by Health Monitor.

Options

Option	Description
`no`	Remove port(s).

Action Configuration

Action configuration is accessed from the top level of the alarm context. If no specific index is provided, a new action with an unused index will be created. In order to access a existing action the index must be provided.

```
example:/config/alarm/#> action
example:/config/alarm/action-2/#>
```

`[no] target [TYPE]`

Description

Add or remove alarm target to an alarm action (profile).

> **NOTE**
>
> Depending on the selected target type, additional configuration options that need to be set can appear. For instance, if the target type port is selected, specific port interfaces needs to be specified.

Options

Option	Description
`no`	Remove any configured targets.
`TYPE`	`led` : ON/Status LED. `log` : Log status change to syslog. `snmp` : Generate an SNMP trap. `digout` : Control the status relay. `port` : Control the admin status of a port. `iface` : Control the admin status of an interface.

The default action (action 1) has SNMP, Log, LED and Digout (status relay) as targets. For new actions, the default target is 'Log'.

```
example:/config/alarm/#> show action 1
ID             : 1
Targets        : snmp log led digout
example:/config/alarm/#> action 2
example:/config/alarm/action-2/#> target led
example:/config/alarm/action-2/#> show
ID             : 2
Targets        : log led
example:/config/alarm/action-2/#>
```

For ‘port’ and ‘iface’ targets, the associated port(s) or network interface(s) to control must be configured.

```
example:/config/alarm/#> action 3
example:/config/alarm/action-3/#> target port
example:/config/alarm/action-3/#> port eth6
example:/config/alarm/action-3/#> show
ID             : 3
Targets        : log port
Ports          : eth6
example:/config/alarm/action-3/#>
```

`[no] port <PORT|PORTS>`

Description

(For ‘target port’) Set which port(s) to control administrative status on.

Options

Option	Description
`no`	Remove configured port.

`[no] iface <IFACE|IFACES>`

Description

(For ‘target iface’) Set which network interface(s) to control administrative status on.

Options

Option	Description
`no`	Remove configured interface.

Status

In order to see all the currently configured alarms on the system, the following command can be invoked from the top level in the CLI:

```
example:/#> show alarm
NO TRIGGER          ENA ACT REASON
 1 temperature      YES  NO
```

If one of the alarms have been triggered, the same command will also display that the alarm is active along with the reason why:

```
example:/#> show alarm
NO TRIGGER          ENA ACT REASON
 1 temperature      YES YES Temperature is 40.75 degrees Celsius
```

2.7.2. Auditable Events

Introduction

This document provides a overview of auditable events within the system. Auditable events are categorised into different groups based on the type of event, making it easier to manage and understand the events. Each individual event that is produced by the system is represented by a unique Event ID.

Auditable events play a crucial role in maintaining the security and integrity of the system. They provide a detailed record of significant actions and changes, which can be used for various purposes, including:

- **Security Monitoring:** Tracking login attempts, configuration changes, and other critical actions to detect and respond to potential security threats.
- **Compliance:** Ensuring that the system adheres to regulatory requirements by maintaining a detailed audit trail of all significant events.
- **Troubleshooting:** Providing a historical record of events that can be used to diagnose and resolve issues within the system.
- **Accounting:** Keeping track of user activities and system changes to ensure accountability and transparency.

This document also details the format of syslog messages generated for audit events, the configuration options available for managing audit event logging, and the operational commands for viewing and managing the audit log.

> **SEE ALSO**
>
> Related Examples and Use-Cases
> - Logging
> - Transmitting Auditable Events to Remote Machines HowTo

Overview

Auditable Event Categories

The auditable events are categorised into different categories, based on the type of the event. The categories are used to group the events into logical groups, making it easier to manage and understand the events. Each individual Event ID will always be associated with a specific category, based on how the Event ID is constructed.

The entire Event ID is a 64-bit value, where the upper 16 bits are used to specify the category of the event, and the lower 48 bits are used to specify the specific event within that category. The following categories are available:

Category ID	Category Name	Description
`0x1000`	Access Control	Events related to access control, such as login, logout, authentication, etc.
`0x2000`	Request Error	Events related to errors in requests, such as invalid requests, unauthorized requests, etc.
`0x3000`	Control System Event	Events related to the control system, such as link up, link down, system reboot, etc.

Category ID	Category Name	Description
`0x4000`	Backup Restore Event	Events related to backup and restore operations, such as storing of configuration, etc.
`0x5000`	Configuration Change	Events related to configuration changes, such as specific configuration changes to the system.
`0x5100`	Configuration Transaction	Events related to configuration transactions, such as configuration commit, rollback, etc.
`0x6000`	Audit Log Event	Events related to the audit log, such as audit log display, audit log clear, etc.
`0x7000`	File System Event	Events related to the file system, such as file creation, deletion, modification, etc.
`0x8000`	Configuration Integrity	Events related to configuration integrity, such as configuration integrity check, etc.
`0x9000`	Boot Process Event	Events related to the boot process

Table 1: Auditable Event Categories

POSSIBLY SUBJECT TO CHANGE

Since this is a very new addition to the system, it is possible that changes will be made to the Categories in the upcoming releases. The intention is to keep the Categories as stable as possible, but because more changes may be needed early on, it is possible that they will be changed.

Auditable Event Types

Each auditable event is represented by a unique Event ID. The Event ID is a 64-bit value, where the upper 16 bits are used to specify the category of the event, and the lower 48 bits are used to specify the specific event within that category.

The Event ID is constructed in the following manner:

```
+-----------------+-----------------+-----------------+
| Category        | Main ID         | Sub ID          |
+-----------------+-----------------+-----------------+
| 0xffff          | 0xffffffff      | 0xffff          |
+-----------------+-----------------+-----------------+
```

As can be seen, the Event ID is divided into three parts:

- **Category:** The upper 16 bits are used to specify the category of the event. This is the exact values defined in the Auditable Event Categories section. Therefore, no Event ID can exist that does not have a valid category, i.e. the initial part of the Event ID will always be a valid category. Therefore, it is always possible to determine the category of an Event ID by looking at the upper 16 bits.
- **Main ID:** The next 32 bits are used to specify the main ID of the event.
- **Sub ID:** The lower 16 bits are used to specify the sub ID of the event. This can be used to further specify an event that is part of a larger category.

As an example, if we take the following Event ID `0x1000000000010010` which represents a successful console login event, we can break it down as follows:

- **Category:** `0x1000` which corresponds to the Access Control category.
- **Main ID:** `0x00000001` which is the main ID of the event.

- **Sub ID:** `0x0010` which is the sub ID of the event.

> **POSSIBLY SUBJECT TO CHANGE**
>
> Since this is a very new addition to the system, it is possible that changes will be made to the Event IDs in the upcoming releases. The intention is to keep the Event IDs as stable as possible, but because more changes may be needed early on, it is possible that they will be changed.
>
> However, if changes are made to existing events in the future, the aim is to retire any existing Event IDs, so that they will not be reused for new events. This way, the Event IDs will always be unique, and will not be reused for different events.

Access Control Events

The following Event IDs are available for the Access Control category:

Event ID	Event Name	Description	Note
`0x1000000000010010`	Console login Success	Successful console login event.	-
`0x1000000000010011`	Console login Exit	Console login session ended.	-
`0x1000000000010012`	Console login Failure	Failed console login attempt.	-
`0x1000000000010013`	Suspicious Console login Success	Console login succeeded with suspicious credentials or context.	-
`0x1000000000010014`	Suspicious Console login Failure	Failed console login attempt flagged as suspicious.	-
`0x1000000000010015`	Console login Success secureTTY	Successful console login via secure TTY (trusted terminal).	-
`0x1000000000010020`	SSH login Success	Successful SSH login event.	-
`0x1000000000010021`	SSH login Exit	SSH login session ended.	-
`0x1000000000010022`	SSH login Failure	Failed SSH login attempt.	-
`0x1000000000010023`	Suspicious SSH login Success	SSH login succeeded with suspicious credentials or context.	-
`0x1000000000010024`	Suspicious SSH login Failure	Failed SSH login attempt flagged as suspicious.	-
`0x1000000000010030`	Web login Success	Successful web interface login event.	-
`0x1000000000010031`	Web login Exit	Web login session ended.	-
`0x1000000000010032`	Web login Failure	Failed web login attempt.	-
`0x1000000000010033`	Suspicious Web login Success	Web login succeeded with suspicious credentials or context.	-

Event ID	Event Name	Description	Note
`0x1000000000010034`	Suspicious Web login Failure	Failed web login attempt flagged as suspicious.	-
`0x1000000000010040`	Telnet login Success	Successful Telnet login event.	-
`0x1000000000010041`	Telnet login Exit	Telnet login session ended.	-
`0x1000000000010042`	Telnet login Failure	Failed Telnet login attempt.	-
`0x1000000000010043`	Suspicious Telnet login Success	Telnet login succeeded with suspicious credentials or context.	-
`0x1000000000010044`	Suspicious Telnet login Failure	Failed Telnet login attempt flagged as suspicious.	-
`0x1000000000010050`	Other login Success	Successful login via other (non-standard) method.	-
`0x1000000000010051`	Other login Exit	Other login session ended.	-
`0x1000000000010052`	Other login Failure	Failed login attempt via other method.	-
`0x1000000000010053`	Suspicious Other login Success	Other login succeeded with suspicious credentials or context.	-
`0x1000000000010054`	Suspicious Other login Failure	Failed other login attempt flagged as suspicious.	-
`0x1000000000020000`	SNMPv3 Access Success	Successful SNMPv3 access.	-
`0x1000000000020001`	SNMPv3 Access Failure	Failed SNMPv3 access attempt.	-
`0x1000000000020002`	SNMPv3 Engine ID Failure	SNMPv3 engine ID verification failed.	-
`0x1000000000020003`	SNMPv2 Access Success	Successful SNMPv2 access.	-
`0x1000000000020004`	SNMPv2 Access Failure	Failed SNMPv2 access attempt.	-
`0x1000000000020005`	SNMP Command Success	SNMP command executed successfully.	-
`0x1000000000020006`	SNMP Command Failure	SNMP command execution failed.	-
`0x1000000000030000`	Enter CLI Shell	Entered CLI shell session.	-
`0x1000000000030001`	Exit CLI Shell	Exited CLI shell session.	-
~~`0x1000000000040001`~~	~~Authorised MAC address through mac-authentication~~	~~MAC address authorised via MAC authentication.~~	**Deprecated**
~~`0x1000000000040002`~~	~~De-authorised MAC address through mac-authentication~~	~~MAC address de-authorised via MAC authentication .~~	**Deprecated**
~~`0x1000000000040003`~~	~~Authentication through mac-authentication failed~~	~~MAC authentication attempt failed .~~	**Deprecated**

Event ID	Event Name	Description	Note
`0x1000000000040008`	Authorised MAC address through MAB	Authorised MAC address through MAB.	-
`0x1000000000040009`	De-authorised MAC address through MAB	De-authorised MAC address through MAB.	-
`0x100000000004000A`	Authentication through MAB failed	Authentication through MAB failed.	-
`0x100000000004000B`	Assign port to VID from radius server through MAB	Assign port to VID from radius server through MAB.	-
`0x100000000004000C`	Assign port to VID from ports default through MAB	Assign port to VID from ports default through MAB.	-
`0x100000000004000D`	Assign port to start VID, nothing authenticated through MAB	Assign port to start VID, nothing authenticated through MAB.	-
`0x100000000004000E`	No allowed VID provided in radius server response for MAB	No allowed VID provided in radius server response for MAB.	-
`0x100000000004000F`	Default VLAN disabled for MAB	Default VLAN disabled for MAB.	-
`0x1000000000040010`	Port is already assigned to a different VID, cannot be assigned by MAB	Port is already assigned to a different VID, cannot be assigned by MAB.	-
`0x1000000000040011`	Port unlocked MAB Port Mode	Port unlocked MAB Port Mode.	-
`0x1000000000040012`	Port locked MAB Port Mode	Port locked MAB Port Mode.	-
`0x1000000000040013`	Authorised MAC address through MAC Whitelist	Authorised MAC address through MAC Whitelist.	-
`0x1000000000040014`	De-authorised MAC address through MAC Whitelist	De-authorised MAC address through MAC Whitelist.	-
`0x1000000000040015`	Authentication through MAC Whitelist failed	Authentication through MAC Whitelist failed.	-
`0x1000000000040016`	Port unlocked MAC Whitelist Port Mode	Port unlocked MAC Whitelist Port Mode.	-
`0x1000000000040017`	Port locked MAC Whitelist Port Mode	Port locked MAC Whitelist Port Mode.	-
~~`0x1000000000040004`~~	~~Authorised MAC address through 802.1X~~	~~MAC address authorised via IEEE 802.1X authentication.~~	**Deprecated**
~~`0x1000000000040005`~~	~~De-authorised MAC address through 802.1X~~	~~MAC address de-authorised via IEEE 802.1X authentication.~~	**Deprecated**
~~`0x1000000000040006`~~	~~Authentication through 802.1X failed~~	~~IEEE 802.1X authentication failed.~~	**Deprecated**
~~`0x1000000000040007`~~	~~IPsec Authentication failed~~	~~IPsec Authentication failed .~~	**Deprecated**

Event ID	Event Name	Description	Note
`0x1000000000050001`	Authorised MAC address through 802.1X	MAC address authorised via IEEE 802.1X authentication.	-
`0x1000000000050002`	De-authorised MAC address through 802.1X	MAC address de-authorised via IEEE 802.1X authentication.	-
`0x1000000000050003`	Authentication through 802.1X failed	IEEE 802.1X authentication failed.	-
`0x1000000000050004`	Assign port to VID from radius server through 802.1X	Assign port to VID from radius server through 802.1X.	-
`0x1000000000050005`	Assign port to VID from ports default through 802.1X	Assign port to VID from ports default through 802.1X.	-
`0x1000000000050006`	Assign port to start VID, nothing authenticated through 802.1X	Assign port to start VID, nothing authenticated through 802.1X.	-
`0x1000000000050007`	No allowed VID provided in radius server response for 802.1X	No allowed VID provided in radius server response for 802.1X.	-
`0x1000000000050008`	Default VLAN disabled for 802.1X	Default VLAN disabled for 802.1X.	-
`0x1000000000050009`	Port is already assigned to a different VID, cannot be assigned by 802.1X	Port is already assigned to a different VID, cannot be assigned by 802.1X.	-
`0x1000000000050010`	Port unlocked 802.1X Port Mode	Port unlocked 802.1X Port Mode.	-
`0x1000000000050011`	Port locked 802.1X Port Mode	Port locked 802.1X Port Mode.	-
`0x1000000000060001`	IPsec Authentication failed	IPsec Authentication failed.	-

Table 2: Access Control Events

Request Error Events

The following Event IDs are available for the Request Error category:

Event ID	Event Name	Description	Note
`0x2000000000010000`	Unauthorised CLI Command	Attempt to execute a CLI command without proper authorisation.	-
`0x2000000000020000`	SNMP OID Request Success	SNMP OID request completed successfully.	-
`0x2000000000020001`	SNMP OID Request Failure	SNMP OID request failed (e.g., due to permissions or bad OID).	-
`0x2000000000030000`	Unauthorised SSH User	SSH login attempt by unauthorised user.	-
`0x2000000000030001`	Unauthorised Web User	Web login attempt by unauthorised user.	-

Event ID	Event Name	Description	Note
`0x2000000000030002`	Unauthorised Console User	Console login attempt by unauthorised user.	-

Table 3: Request Error Events

Control System Events

The following Event IDs are available for the Control System Event category:

Event ID	Event Name	Description	Note
`0x3000000000010001`	Service Start	A system service has started.	-
`0x3000000000010002`	Service Stop	A system service has stopped. Some services will also be stopped and started as part of their restart handling.	-
`0x3000000000010003`	Service Restart	A system service has restarted. Typically this can occur during system reconfiguration.	-
`0x3000000000010004`	Service Died	A system service has unexpectedly terminated. The system will attempt to restart it.	-
`0x3000000000010005`	Service Crash	A system service has crashed. It has died too many times within a short period and attempts to restart it will cease, it is now considered crashed.	-
`0x3000000000010006`	Service Crash Restart	A system service crashed and was automatically restarted.	-
`0x3000000000020001`	Link Up	A network link became active.	-
`0x3000000000020002`	Link Down	A network link became inactive.	-
`0x3000000000030001`	System Boot	The system has booted.	-
`0x3000000000030002`	System Startup	The system startup process has completed.	-
`0x3000000000030003`	System Operational	The system is now operational.	-
`0x3000000000030004`	System Shutdown	The system is shutting down.	-
`0x3000000000030005`	System Upgrade	A system upgrade has occurred.	-
`0x3000000000030006`	System Upgrade Failed	System Upgrade Failed.	-
`0x3000000000030007`	System Upgrade Validation Passed	System Upgrade Validation Passed.	-
`0x3000000000030008`	System Upgrade Validation Failed	System Upgrade Validation Failed.	-
`0x3000000000040000`	ECSC Communication Established	ECSC communication channel established.	-
`0x3000000000040001`	TTDP Stack Ready	TTDP protocol stack is ready.	-
`0x3000000000050002`	DHCP Client Lease Deconfig	DHCP client lease deconfigured.	-

Event ID	Event Name	Description	Note
0x3000000000050003	DHCP Client Lease Renew	DHCP client lease renewed.	-
0x3000000000050004	DHCP Client Lease Bound	DHCP client lease bound to an address.	-
0x3000000000050005	DHCP Client Received NAK	DHCP client received a negative acknowledgment (NAK).	-
0x3000000000050006	DHCP Client Lease Failed	DHCP client lease acquisition failed.	-
0x3000000000050100	DHCP Server Discover	DHCP server received a discover message.	-
0x3000000000050101	DHCP Server Offer	DHCP server sent an offer message.	-
0x3000000000050102	DHCP Server Request	DHCP server received a request message.	-
0x3000000000050103	DHCP Server Acknowledge	DHCP server sent an acknowledge message.	-
0x3000000000050104	DHCP Server Release	DHCP server received a release message.	-
0x3000000000050105	DHCP Server NAK	DHCP server sent a negative acknowledgment (NAK).	-
0x3000000000050106	DHCP Server Decline	DHCP server received a decline message.	-
0x3000000000050107	DHCP Server Inform	DHCP server received an inform message.	-
0x3000000000050108	DHCP Server No Address Available	DHCP server has no address available to assign.	-
0x3000000000060001	Configuration Changed	A configuration change was made.	-
0x3000000000070001	Enter Maintenance Mode	System entered maintenance mode.	-
0x3000000000070002	Exit Maintenance Mode	System exited maintenance mode.	-
0x3000000000080000	MRP Ring OK	MRP ring is operational.	-
0x3000000000080001	MRP Ring Broken	MRP ring is broken.	-
0x3000000000090000	NTP Clock Synchronised	NTP clock synchronised successfully.	-
0x3000000000090001	NTP Update Failed	NTP clock update failed.	-
0x30000000000a0000	FRNT Ring OK	FRNT ring is operational.	-
0x30000000000a0001	FRNT Ring Broken	FRNT ring is broken.	-
0x30000000000b0000	Duplicate IP Address	A duplicate IP address was detected.	-
0x30000000000b0001	Duplicate MAC Address	A duplicate MAC address was detected.	-

Event ID	Event Name	Description	Note
0x30000000000b0002	ARP New Entry	A new ARP entry was added.	-
0x30000000000b0003	ARP Entry Changed	An ARP entry was changed.	-
0x30000000000b0004	ARP Entry Flip Flop	ARP entry is rapidly changing (flip-flop detected).	-
0x30000000000c0000	RICO Port State Change	RICO port state changed.	-
0x30000000000d0000	ATU Full Violation	Address Translation Unit (ATU) is full.	-
0x30000000000e0000	Unknown IPv4 Route Install	An unknown IPv4 route was installed.	-
0x30000000000e0001	Unknown IPv4 Route Delete	An unknown IPv4 route was deleted.	-
0x30000000000e0002	Kernel IPv4 Route Install	A kernel IPv4 route was installed.	-
0x30000000000e0003	Kernel IPv4 Route Delete	A kernel IPv4 route was deleted.	-
0x30000000000e0004	Connected IPv4 Route Install	A connected IPv4 route was installed.	-
0x30000000000e0005	Connected IPv4 Route Delete	A connected IPv4 route was deleted.	-
0x30000000000e0006	Static IPv4 Route Install	A static IPv4 route was installed.	-
0x30000000000e0007	Static IPv4 Route Delete	A static IPv4 route was deleted.	-
0x30000000000e0008	RIP IPv4 Route Install	A RIP IPv4 route was installed.	-
0x30000000000e0009	RIP IPv4 Route Delete	A RIP IPv4 route was deleted.	-
0x30000000000e000a	OSPF IPv4 Route Install	An OSPF IPv4 route was installed.	-
0x30000000000e000b	OSPF IPv4 Route Delete	An OSPF IPv4 route was deleted.	-
0x30000000000e000c	Kernel IPv6 Route Install	A kernel IPv6 route was installed.	-
0x30000000000e000d	Kernel IPv6 Route Delete	A kernel IPv6 route was deleted.	-
0x30000000000e000e	Connected IPv6 Route Install	A connected IPv6 route was installed.	-
0x30000000000e000f	Connected IPv6 Route Delete	A connected IPv6 route was deleted.	-
0x30000000000e0010	Static IPv6 Route Install	A static IPv6 route was installed.	-
0x30000000000e0011	Static IPv6 Route Delete	A static IPv6 route was deleted.	-
0x30000000000e0012	RIP IPv6 Route Install	A RIP IPv6 route was installed.	-
0x30000000000e0013	RIP IPv6 Route Delete	A RIP IPv6 route was deleted.	-
0x30000000000e0014	OSPF IPv6 Route Install	An OSPF IPv6 route was installed.	-
0x30000000000e0015	OSPF IPv6 Route Delete	A OSPF IPv6 route was deleted.	-
0x30000000000e0016	Unknown IPv6 Route Install	An unknown IPv6 route was installed.	-

Event ID	Event Name	Description	Note
0x30000000000e0017	Unknown IPv6 Route Delete	An unknown IPv6 route was deleted.	-
0x30000000000e0018	Multicast IPv4 Route Install	A multicast IPv4 route was installed.	-
0x30000000000e0019	Multicast IPv4 Route Delete	A multicast IPv4 route was deleted.	-
0x30000000000f0000	Link Alarm Actived	A link alarm was activated.	-
0x30000000000f0001	Link Alarm Deactived	A link alarm was deactivated.	-
0x30000000000f0002	Temp Alarm Actived	A temperature alarm was activated.	-
0x30000000000f0003	Temp Alarm Deactived	A temperature alarm was deactivated.	-
0x30000000000f0004	Power Alarm Actived	A power alarm was activated.	-
0x30000000000f0005	Power Alarm Deactived	A power alarm was deactivated.	-
0x30000000000f0006	Digin Alarm Actived	A digital input alarm was activated.	-
0x30000000000f0007	Digin Alarm Deactived	A digital input alarm was deactivated.	-
0x30000000000f0008	Ping Alarm Actived	A ping alarm was activated.	-
0x30000000000f0009	Ping Alarm Deactived	A ping alarm was deactivated.	-
0x30000000000f000a	FRNT Alarm Actived	A FRNT alarm was activated.	-
0x30000000000f000b	FRNT Alarm Deactived	A FRNT alarm was deactivated.	-
0x30000000000f000c	Ring Alarm Actived	A ring alarm was activated.	-
0x30000000000f000d	Ring Alarm Deactived	A ring alarm was deactivated.	-
0x30000000000f000e	Profinet Alarm Actived	A Profinet alarm was activated.	-
0x30000000000f000f	Profinet Alarm Deactived	A Profinet alarm was deactivated.	-
0x30000000000f0010	POE Alarm Actived	A PoE alarm was activated.	-
0x30000000000f0011	POE Alarm Deactived	A PoE alarm was deactivated.	-
0x30000000000f0012	RICO Alarm Actived	A RICO alarm was activated.	-
0x30000000000f0013	RICO Alarm Deactived	A RICO alarm was deactivated.	-
0x30000000000f0014	Media Threshold Alarm Actived	A media threshold alarm was activated.	-
0x30000000000f0015	Media Threshold Alarm Deactived	A media threshold alarm was deactivated.	-
0x30000000000f0016	Media Plug Alarm Actived	A media plug alarm was activated.	-
0x30000000000f0017	Media Plug Alarm Deactived	A media plug alarm was deactivated.	-
0x3000000000100000	Media Plugged In	A media device, e.g. a USB device or SD card, was plugged	-

Event ID	Event Name	Description	Note
		in. This does not indicate that the device has been mounted or is ready for use.	
`0x3000000000100001`	Media Removed	A media device was removed.	-
`0x3000000000100002`	Media Mounted	A media device was mounted. The device is now available and ready for use.	-
`0x3000000000100003`	Media Unmounted	A media device was unmounted.	-
`0x3000000000100004`	Media Mount Failed	Media Mount Failed.	-
`0x3000000000110000`	Firewall Allow Rule Hit	A firewall allow rule was hit.	-
`0x3000000000110001`	Firewall Deny Rule Hit	A firewall deny rule was hit.	-
`0x3000000000110002`	Firewall DSCP Rule Hit	A firewall DSCP rule was hit.	-
`0x3000000000110003`	Firewall Port Forwarding Rule Hit	A firewall port forwarding rule was hit.	-
`0x3000000000110004`	Firewall NAT Rule Hit	A firewall NAT rule was hit.	-
`0x3000000000120000`	Certificate has been revoked	A certificate was revoked.	-
`0x3000000000120001`	CRL distribution point not accessible	The Certificate Revocation List (CRL) distribution point could not be accessed.	-
`0x3000000000120002`	CRL expired	The Certificate Revocation List (CRL) has expired.	-
`0x3000000000120003`	CRL signature verification failed	CRL signature verification failed.	-
`0x3000000000120004`	CRL too large for download	CRL was too large to download.	-
`0x3000000000120005`	SSH host key management	SSH host key management event occurred.	-
`0x3000000000120006`	SSH host key management failure	SSH host key management failure event occurred.	-
`0x3000000000120007`	Weak/deprecated certificate used	Weak/deprecated certificate used.	-
`0x3000000000130000`	Custom Telegram Publish	Custom Telegram Publish.	-
`0x3000000000130001`	Custom Telegram Subscribe	Custom Telegram Subscribe.	-
`0x3000000000130002`	Custom Telegram System Clock Changed	Custom telegram indicating system clock change.	-
`0x3000000000130003`	Custom Telegram Inhibit Flag Changed	Custom telegram indicating inhibit flag change.	-
`0x3000000000130004`	Custom Telegram Bypass-relay Changed	Custom telegram indicating bypass-relay change.	-
`0x3000000000140000`	Package Installed	A software package has been installed.	-
`0x3000000000140001`	Package Removed	A software package has been removed.	-

Event ID	Event Name	Description	Note
`0x3000000000150000`	TPM Error	A Trusted Platform Module (TPM) error occurred.	-
`0x3000000000160000`	System time manually set	System time manually set.	-
`0x3000000000160001`	System time manually set failure	Failure occurred while manually setting system time.	-
`0x3000000000160002`	RTC time manually set	Real-Time Clock (RTC) time was manually set.	-
`0x3000000000160003`	RTC time manually set failure	Failure occurred while manually setting RTC time.	-
`0x3000000000160004`	RTC chip incorrectly configured	RTC chip incorrectly configured.	-
`0x3000000000160005`	Failed setting configuration for RTC chip	Failed setting configuration for RTC chip.	-
`0x3000000000170000`	IPsec Closing Child SA	IPsec is closing a Child Security Association (SA).	-
`0x3000000000170001`	IPsec Established Child SA	IPsec has established a Child Security Association (SA).	-
`0x3000000000170002`	IPsec Reauth IKE SA	IPsec Reauth Internet Key Exchange (IKE) Security Association (SA).	-
`0x3000000000170003`	IPsec Closing IKE SA	IPsec Closing Internet Key Exchange (IKE) Security Association (SA).	-
`0x3000000000180000`	App Start	A container application was started.	-
`0x3000000000180001`	App Stop	A container application was stopped.	-
`0x3000000000180002`	App Restart	A container application was restarted.	-
`0x3000000000180003`	App Attach	A user attached to a running application container console.	-
`0x3000000000180004`	App Reload	An application was reloaded after configuration or image label changes.	-
`0x3000000000180005`	App Update Label On Image	An image label was updated or assigned to an application image.	-
`0x3000000000180006`	App Load All	All configured applications were loaded and initialized.	-
`0x3000000000180007`	App Signature Verification	An application image signature was verified.	-
`0x3000000000180008`	App Delete	An application instance or application image was deleted.	-

Table 4: Control System Events

Note that this simply lists what can be logged, what is logged is dependent on the configuration of the system. For instance, no audit events will be generated related to DHCP server if no DHCP server is configured.

Backup Restore Events

The following Event IDs are available for the Backup Restore Event category:

Event ID	Event Name	Description	Note
`0x4000000000010000`	Update Running Configuration	The system running configuration was updated after successful activation.	-
`0x4000000000020000`	Read Config File	A configuration file was read.	-
`0x4000000000020001`	Read Config File Error	Failed to read configuration file.	-
`0x4000000000020002`	Read Config File Error - JSON	JSON parsing error while reading the configuration file.	-
`0x4000000000020003`	Read Config File Error - No Exist	Configuration file not found.	-
`0x4000000000030000`	Validate Config File	Configuration file validation was initiated.	-
`0x4000000000040000`	Write Config File	Configuration was written to file.	-
`0x4000000000040001`	Write Config File Error	Failed to open configuration file for writing.	-
`0x4000000000040002`	Write Config File Error - JSON	JSON formatting or encoding error while writing the configuration file.	-
`0x4000000000050000`	Copied Config File - CLI	Configuration file copied via CLI command.	-
`0x4000000000050001`	Copy Config File Error - CLI	Error occurred while copying configuration file via CLI.	-
`0x4000000000060000`	Verify Config File Encryption	Configuration file encryption verification.	-
`0x4000000000070000`	Support File Generated	A support file for troubleshooting was created.	-
`0x4000000000080000`	JSON Schema validation successful	Configuration file passed JSON schema validation.	-
`0x4000000000080001`	JSON Schema validation fail	Configuration file failed JSON schema validation.	-
`0x4000000000080002`	JSON Schema validation fail with error	Configuration file failed JSON schema validation with specific error details.	-
`0x4000000000080003`	JSON Schema validation fail, force apply	Configuration was force applied despite JSON schema validation failure.	-

Event ID	Event Name	Description	Note
`0x4000000000080004`	JSON Schema validation fail, force copy	Configuration file was force copied despite JSON schema validation failure.	-
`0x4000000000090000`	Web Enable Config Force Apply	Force apply option temporarily enabled for current web session.	-
`0x4000000000090001`	Web Disable Config Force Apply	Force apply option disabled for current web session.	-
`0x4000000000090002`	Web Config Force Apply	Configuration change was force applied via web interface, validation checks were ignored.	-

Table 5: Backup Restore Events

Configuration Change Events

The following Event IDs are available for the Configuration Change category:

Event ID	Event Name	Description	Note
`0x5000000000020000`	LLDP	Configuration change related to LLDP (Link Layer Discovery Protocol).	-
`0x5000000000020001`	LLDP Port	Configuration change for an LLDP-enabled port.	-
`0x5000000000030000`	DHCP Server	Configuration change to the DHCP server settings.	-
`0x5000000000030001`	DHCP Server Subnet	Configuration change to a DHCP server subnet.	-
`0x5000000000030002`	DHCP Server Host	Configuration change to a DHCP server host entry.	-
`0x5000000000030003`	DHCP Server Route	Configuration change to a DHCP server route.	-
`0x5000000000030004`	DHCP Server Host Match	Configuration change to DHCP server host matching rules.	-
`0x5000000000030005`	DHCP Server Static Leases	Configuration change to DHCP server static lease assignments.	-
`0x5000000000040000`	SSH	Configuration change related to SSH settings.	-
`0x5000000000040001`	SSH Shell access	Configuration change for SSH shell access permissions.	-
`0x5000000000050000`	Telnet	Configuration change related to Telnet settings.	-
`0x5000000000060000`	Port	Configuration change to a network port.	-
`0x5000000000060001`	Ethernet Port	Configuration change to an Ethernet port.	-

Event ID	Event Name	Description	Note
`0x5000000000070000`	Ownership	Configuration change to system or resource ownership.	-
`0x5000000000080000`	System	General system configuration change.	-
`0x5000000000090000`	TFTP	Configuration change related to TFTP settings.	-
`0x50000000000a0000`	DNS	Configuration change related to DNS settings.	-
`0x50000000000a0001`	DNS Forward Rule	Configuration change to DNS forwarding rules.	-
`0x50000000000a0002`	DNS Server	Configuration change to a DNS server entry.	-
`0x50000000000a0003`	DNS Host	Configuration change to a DNS host entry.	-
`0x50000000000a0004`	DNS Search Path	Configuration change to DNS search path settings.	-
`0x50000000000b0000`	NTP	Configuration change related to NTP (Network Time Protocol) settings.	-
`0x50000000000b0001`	NTP Client	Configuration change to NTP client settings.	-
`0x50000000000c0000`	WEB	Configuration change related to web interface settings.	-
`0x50000000000c0001`	HTTP	Configuration change to HTTP settings.	-
`0x50000000000c0002`	HTTPS	Configuration change to HTTPS settings.	-
`0x50000000000d0000`	RSTP	Configuration change related to RSTP (Rapid Spanning Tree Protocol).	-
`0x50000000000d0001`	RSTP Port	Configuration change to an RSTP-enabled port.	-
`0x50000000000e0000`	FRNT	Configuration change related to FRNT (Fast Recovery Network Topology).	-
`0x50000000000e0001`	FRNT Port	Configuration change to a FRNT port.	-
`0x50000000000f0000`	ICMP	Configuration change related to ICMP (Internet Control Message Protocol) settings.	-
`0x5000000000100000`	Management	Configuration change to management interface or settings.	-

Event ID	Event Name	Description	Note
`0x5000000000110000`	Password	Configuration change to a password or password policy.	-
`0x5000000000120000`	VRRP	Configuration change related to VRRP (Virtual Router Redundancy Protocol).	-
`0x5000000000120001`	VRRP Trigger	Configuration change to VRRP trigger settings.	-
`0x5000000000120002`	VRRP Instance	Configuration change to a VRRP instance.	-
`0x5000000000120003`	VRRP Group	Configuration change to a VRRP group.	-
`0x5000000000130000`	RIP Interface	Configuration change to a RIP (Routing Information Protocol) interface.	-
`0x5000000000130001`	RIP Interface MD5	Configuration change to RIP interface MD5 authentication.	-
`0x5000000000130002`	RIP Interface Secret	Configuration change to RIP interface secret settings.	-
`0x5000000000130003`	RIP Interface Auth	Configuration change to RIP interface authentication settings.	-
`0x5000000000140000`	RIP	Configuration change related to RIP protocol.	-
`0x5000000000140001`	RIP Network	Configuration change to a RIP network entry.	-
`0x5000000000150000`	OSPF Interface	Configuration change to an OSPF (Open Shortest Path First) interface.	-
`0x5000000000150001`	OSPF Interface MD5	Configuration change to OSPF interface MD5 authentication.	-
`0x5000000000150002`	OSPF Interface Secret	Configuration change to OSPF interface secret settings.	-
`0x5000000000150003`	OSPF Interface Auth	Configuration change to OSPF interface authentication settings.	-
`0x5000000000160000`	OSPF	Configuration change related to OSPF protocol.	-
`0x5000000000160001`	OSPF Network	Configuration change to an OSPF network entry.	-
`0x5000000000160002`	OSPF Area	Configuration change to an OSPF area.	-
`0x5000000000160003`	OSPF Timers	Configuration change to OSPF timer settings.	-
`0x5000000000160004`	OSPF Redistribute	Configuration change to OSPF route redistribution settings.	-

Event ID	Event Name	Description	Note
`0x5000000000160005`	OSPF Distribute Default	Configuration change to OSPF default route distribution.	-
`0x5000000000170000`	PIM Interface	Configuration change to a PIM (Protocol Independent Multicast) interface.	-
`0x5000000000180000`	PIM	Configuration change related to PIM protocol.	-
`0x5000000000180001`	PIM Rendezvous Point	Configuration change to a PIM rendezvous point.	-
`0x5000000000180002`	PIM Policy	Configuration change to a PIM policy.	-
`0x5000000000180003`	PIM SSM Prefix	Configuration change to a PIM SSM (Source-Specific Multicast) prefix.	-
`0x5000000000190000`	Interface	Configuration change to a network interface.	-
`0x5000000000190001`	Interface IPv4	Configuration change to an IPv4 interface.	-
`0x5000000000190002`	Interface IPv4 Address	Configuration change to an IPv4 address on an interface.	-
`0x5000000000190003`	Interface IPv6	Configuration change to an IPv6 interface.	-
`0x50000000001a0000`	IP	Configuration change related to IP settings.	-
`0x50000000001a0001`	IP Route	Configuration change to an IP route.	-
`0x50000000001a0002`	IP Multicast Route	Configuration change to an IP multicast route.	-
`0x50000000001a0003`	IP NAT	Configuration change to IP NAT (Network Address Translation) settings.	-
`0x50000000001a0004`	Policy Route Match Ip	Configuration change to policy route IP match settings.	-
`0x50000000001a0005`	IP Policy Route Match	Configuration change to IP policy route match rules.	-
`0x50000000001a0006`	IP Policy Route	Configuration change to an IP policy route.	-
`0x50000000001b0000`	Firewall	Configuration change related to firewall settings.	-
`0x50000000001b0001`	Firewall Network	Configuration change to a firewall network object.	-
`0x50000000001b0002`	Firewall Rule	Configuration change to a firewall rule.	-

Event ID	Event Name	Description	Note
`0x50000000001b0003`	Firewall Counter	Configuration change to a firewall counter.	-
`0x50000000001b0004`	Firewall Log	Configuration change to firewall logging settings.	-
`0x50000000001c0000`	PoE	Configuration change related to Power over Ethernet (PoE) settings.	-
`0x50000000001c0001`	PoE Port	Configuration change to a PoE-enabled port.	-
`0x50000000001d0000`	AAA	Configuration change related to AAA (Authentication, Authorization, Accounting) settings.	-
`0x50000000001d0001`	AAA User SSH Key	Configuration change to a user's SSH key in AAA settings.	-
`0x50000000001d0002`	AAA User	Configuration change to a user in AAA settings.	-
`0x50000000001d0003`	AAA Local User	Configuration change to a local user in AAA settings.	-
`0x50000000001d0004`	AAA Local Database	Configuration change to the AAA local user database.	-
`0x50000000001d0005`	AAA Remote Server	Configuration change to a remote AAA server.	-
`0x50000000001d0006`	AAA Server Group	Configuration change to a AAA server group.	-
`0x50000000001d0007`	AAA Server	Configuration change to a AAA server entry.	-
`0x50000000001d0008`	AAA Method	Configuration change to a AAA authentication method.	-
`0x50000000001d0009`	AAA MAC Pattern	Configuration change to a AAA MAC address pattern.	-
`0x50000000001d000a`	AAA Authentication	Configuration change to AAA authentication settings.	-
`0x50000000001d000b`	AAA 802.1x Auth Groups	Configuration change to AAA 802.1x authentication groups.	-
`0x50000000001d000c`	AAA MAC Auth Groups	Configuration change to AAA MAC authentication groups.	-
`0x50000000001d000d`	AAA Login Auth Groups	Configuration change to AAA login authentication groups.	-
`0x50000000001d000e`	AAA Password Policy	Configuration change to AAA password policy.	-
`0x50000000001d000f`	AAA Lockout Policy	Configuration change to AAA lockout policy.	-
`0x50000000001d0010`	AAA Authentication Chain	AAA Authentication Chain.	-

Event ID	Event Name	Description	Note
`0x50000000001e0000`	SSL	Configuration change related to SSL (Secure Sockets Layer) settings.	-
`0x50000000001e0001`	SSL Network	Configuration change to an SSL network object.	-
`0x50000000001e0002`	SSL Pool Range	Configuration change to SSL pool range settings.	-
`0x50000000001e0003`	SSL Internal Route	Configuration change to an SSL tunnel.	-
`0x50000000001e0004`	SSL Client Config	Configuration change to SSL client configuration.	-
`0x50000000001f0000`	Generic Routing Encap	Configuration change to GRE (Generic Routing Encapsulation) settings.	-
`0x5000000000200000`	Audit Logging	Configuration change to audit logging settings.	-
`0x5000000000210000`	IPSec	Configuration change related to IPSec (IP Security) settings.	-
`0x5000000000210001`	IPSec Remote CA	Configuration change to IPSec remote certificate authority settings.	-
`0x5000000000210002`	IPSec Protocol Port	Configuration change to IPSec protocol port settings.	-
`0x5000000000220000`	Tunnel	Configuration change to a network tunnel.	-
`0x5000000000220001`	SSL Tunnel	Configuration change to an SSL tunnel.	-
`0x5000000000220002`	GRE Tunnel	Configuration change to a GRE tunnel.	-
`0x5000000000220003`	IPSec Tunnel	Configuration change to an IPSec tunnel.	-
`0x5000000000230000`	IPv6	Configuration change related to IPv6 settings.	-
`0x5000000000230001`	IPv6 Route	Configuration change to an IPv6 route.	-
`0x5000000000240000`	PTP	Configuration change related to PTP (Precision Time Protocol) settings.	-
`0x5000000000240001`	PTP Clock	Configuration change to a PTP clock.	-
`0x5000000000240002`	PTP Parameters	Configuration change to PTP parameters.	-

Event ID	Event Name	Description	Note
`0x5000000000250000`	TTDP	Configuration change related to TTDP (Time-Triggered Data Protocol) settings.	-
`0x5000000000250001`	TTDP ECN	Configuration change to TTDP ECN (Explicit Congestion Notification) settings.	-
`0x5000000000250002`	TTDP Multicast Route	Configuration change to TTDP multicast route settings.	-
`0x5000000000250003`	TTDP Port	Configuration change to a TTDP port.	-
`0x5000000000250004`	TTDP Internet	Configuration change to TTDP internet settings.	-
`0x5000000000250005`	TTDP Custom TRDP	Configuration change to TTDP CUTE (Custom User Traffic Engineering) settings.	-
`0x5000000000250006`	TTDP Custom TRDP Alias	TTDP Custom TRDP Alias.	-
`0x5000000000260000`	Watchdog	Configuration change to watchdog settings.	-
`0x5000000000260001`	Watchdog Monitor	Configuration change to watchdog monitor settings.	-
`0x5000000000270000`	VLAN	Configuration change related to VLAN (Virtual LAN) settings.	-
`0x5000000000270001`	VLAN Commons	Configuration change to common VLAN settings.	-
`0x5000000000270002`	VLAN dbnum	Configuration change to VLAN database number.	-
`0x5000000000280000`	LAG	Configuration change related to LAG (Link Aggregation Group) settings.	-
`0x5000000000280001`	LAG LACP	Configuration change to LAG LACP (Link Aggregation Control Protocol) settings.	-
`0x5000000000280002`	LAG TTPD	Configuration change to LAG TTPD settings.	-
`0x5000000000280003`	LAG Port	Configuration change to a LAG port.	-
`0x5000000000290000`	Action	Configuration change to an action or action list.	-
`0x5000000000290001`	Action Interface	Configuration change to an action interface list.	-
`0x50000000002a0000`	Trigger	Configuration change to a trigger or trigger list.	-
`0x50000000002b0000`	Alarm	Configuration change to an alarm or alarm settings.	-

Event ID	Event Name	Description	Note
`0x50000000002b0001`	Alarm Trigger	Configuration change to an alarm trigger list.	-
`0x50000000002b0002`	Alarm Action	Configuration change to an alarm action list.	-
`0x50000000002c0000`	Ring	Configuration change related to ring topology settings.	-
`0x50000000002c0001`	Ring Port	Configuration change to a ring port.	-
`0x50000000002c0002`	Ring MRP	Configuration change to ring MRP (Media Redundancy Protocol) settings.	-
`0x50000000002d0000`	Console	Configuration change to console access or settings.	-
`0x50000000002e0000`	RiCo	Configuration change related to RiCo (Ring Coupling) settings.	-
`0x50000000002e0001`	RiCo Coupling Port	Configuration change to a RiCo coupling port.	-
`0x50000000002e0002`	RiCo Subring	Configuration change to a RiCo subring.	-
`0x50000000002f0000`	DDNS	Configuration change related to DDNS (Dynamic DNS) settings.	-
`0x50000000002f0001`	DDNS Provider	Configuration change to a DDNS provider.	-
`0x50000000002f0002`	DDNS Custom Provider	Configuration change to a custom DDNS provider.	-
`0x50000000002f0003`	DDNS Builtin Provider	Configuration change to a built-in DDNS provider.	-
`0x5000000000300000`	Application Container	Configuration change to an application container.	-
`0x5000000000300001`	Application Container Share	Configuration change to an application container share.	-
`0x5000000000300002`	App Environment Variable	Configuration change to an application environment variable.	-
`0x5000000000300003`	App Capability	App Capability.	-
`0x5000000000310000`	SNMP	Configuration change related to SNMP (Simple Network Management Protocol) settings.	-
`0x5000000000310001`	SNMP Address	Configuration change to an SNMP IP address.	-
`0x5000000000310002`	SNMP User	Configuration change to an SNMP user.	-
`0x5000000000310003`	SNMP Trap Host	Configuration change to an SNMP trap host.	-

Event ID	Event Name	Description	Note
0x5000000000310004	SNMP Engine ID	Configuration change to an SNMP engine ID.	-
0x5000000000310005	SNMP Ifindex Persistence	Configuration change to SNMP ifIndex persistence settings.	-
0x5000000000310006	SNMP Command	Configuration change to an SNMP command.	-
0x5000000000320000	Monitor	Configuration change to monitoring settings.	-
0x5000000000320004	Monitor Destination	Configuration change to a monitoring destination.	-
0x5000000000320006	Monitor Source	Configuration change to a monitoring source.	-
0x5000000000330000	Serial Port	Configuration change to a serial port.	-
0x5000000000340000	Serial Port App	Configuration change to a serial port application.	-
0x5000000000350000	GPS	Configuration change related to GPS settings.	-
0x5000000000360000	HSR/PRP	Configuration change related to HSR (High-availability Seamless Redundancy) or PRP (Parallel Redundancy Protocol) settings.	-
0x5000000000360001	HSR/PRP Statistics	Configuration change to HSR/PRP statistics settings.	-
0x5000000000360002	HSR/PRP Port	Configuration change to an HSR/PRP port.	-
0x5000000000360003	HSR/PRP Pairing	Configuration change to HSR/PRP port pairing.	-
0x5000000000370000	Multicast DNS	Configuration change related to mDNS (Multicast DNS) settings.	-
0x5000000000380000	VRF	Configuration change related to VRF (Virtual Routing and Forwarding) settings.	-
0x5000000000390001	MPTCP	Configuration change related to MPTCP (Multipath TCP) settings.	-
0x50000000003a0000	Profinet	Configuration change related to Profinet protocol settings.	-
0x50000000003b0000	DHCP Relay	Configuration change related to DHCP relay settings.	-
0x50000000003b0001	DHCP Relay Interface	Configuration change to a DHCP relay interface.	-
0x50000000003b0002	DHCP Relay Server	Configuration change to a DHCP relay server.	-

Event ID	Event Name	Description	Note
`0x50000000003b0003`	DHCP Relay Option 82	Configuration change to DHCP relay Option 82 settings.	-
`0x50000000003b0004`	DHCP Relay Port	Configuration change to a DHCP relay port.	-
`0x50000000003c0000`	CLI	Configuration change related to CLI (Command Line Interface) settings.	-
`0x50000000003d0000`	SSDP	Configuration change related to SSDP (Simple Service Discovery Protocol) settings.	-
`0x50000000003e0000`	Metrics	Configuration change to system metrics or monitoring settings.	-
`0x50000000003f0000`	Port Access DOT1X	Configuration change related to IEEE 802.1X authentication settings.	-
`0x50000000003f0001`	Port Access MAC auth	Configuration change to MAC-based authentication settings.	-
`0x50000000003f0002`	Port Access	Configuration change to port access control settings.	-
`0x50000000003f0003`	Port Access VLAN Assignment	Port Access VLAN Assignment.	-
`0x5000000000400000`	FDB	Configuration change to the forwarding database (FDB).	-
`0x5000000000400001`	FDB MAC	Configuration change to a MAC address in the forwarding database.	-
`0x5000000000400002`	FDB Group	Configuration change to a group in the forwarding database.	-
`0x5000000000410000`	RNRP	Configuration change related to RNRP (Redundant Network Ring Protocol) settings.	-
`0x5000000000410001`	RNRP Explicit	Configuration change to explicit RNRP settings.	-
`0x5000000000420000`	Vendor	Configuration change to vendor-specific settings.	-
`0x5000000000430000`	Policy	Configuration change to a policy or policy list.	-
`0x5000000000430001`	Policy Network	Configuration change to a policy network object.	-
`0x5000000000430002`	Policy Rule	Configuration change to a policy rule.	-
`0x5000000000430003`	Policy Port	Configuration change to a policy port.	-
`0x5000000000440000`	AT Command	Configuration change to an AT command.	-

Event ID	Event Name	Description	Note
`0x5000000000440001`	AT Map	Configuration change to an AT command map.	-
`0x5000000000440002`	AT Message	Configuration change to an AT command message.	-
`0x5000000000440003`	AT User Message	Configuration change to an AT user message.	-
`0x5000000000450000`	Logging	Configuration change to logging settings.	-
`0x5000000000450001`	Logging Source	Configuration change to a logging source.	-
`0x5000000000450002`	Logging Destination	Configuration change to a logging destination.	-
`0x5000000000450003`	Logging Filter	Configuration change to a logging filter.	-
`0x5000000000450004`	Logging Sink	Configuration change to a logging sink.	-
`0x5000000000460000`	TRDP	Configuration change related to TRDP (Train Real-time Data Protocol) settings.	-
`0x5000000000470000`	Router	Configuration change to router settings.	-
`0x5000000000480000`	RiCh	Configuration change related to RiCh (Ring Channel) settings.	-
`0x5000000000490000`	Factory Reset	Configuration change to factory reset settings or operation.	-
`0x5000000000500000`	Encrypted Secrets	Configuration change to encrypted secrets or secret management.	-
`0x5000000000510000`	Provisioning	Configuration change related to system provisioning.	-
`0x5000000000520000`	PKI	Configuration change related to PKI (Public Key Infrastructure) settings.	-
`0x5000000000520001`	PKI Server Status	Configuration change to PKI server status.	-
`0x5000000000520002`	PKI Enroll Server	Configuration change to PKI enrollment server settings.	-
`0x5000000000520003`	PKI Revocation	Configuration change to PKI certificate revocation settings.	-
`0x5000000000530000`	Bootloader Configuration	Bootloader Configuration.	-
`0x5000000000540000`	WireGuard	WireGuard.	-

Table 6: Configuration Change Events

These Event IDs are generated to account for any configuration change made to any configurable setting in the system. All events under the cluster `0x5000xxxxxxxxxxxx` provide a comprehensive record of configuration changes to any configurable setting in the system.

Referring to the list of auditable Event IDs, various configuration change events can be generated. Instead of having a unique Event ID for every single configuration setting, they are categorised based on the type of setting changed. For example, there are unique Event IDs for changes made to system, interfaces, ports, VLANs, DHCP-Server, etc.

As an example of a configuration change event, consider that we change a few settings under the `system` configuration, like this:

```
example:/#> configure
example:/config/#> system
example:/config/system/#> hostname MySwitch
example:/config/system/#> location My Location
example:/config/system/#> contact My Contact
example:/config/system/#> leave
MySwitch:/#>
```

This should new generate three different audit log entries, with the Event ID representing configuration changes for `system`, which would be `0x5000000000080000`. Therefore, the generated audit log entries would look something like this:

```
MySwitch:/#> audit
MySwitch:/audit/#> show
 Audit log ring buffer, entries 1-3 of 142
SEQ  TIME        LEVEL/FACILITY  USER/ROLE       EVENTID              MESSAGE
209  2025-03-10  Notice          admin           Configuration C ...  system: "contact": from 'my
     12:18:44    Security        administrator   System               previous' → 'My Contact'
                                                 0x5000000000080000
208  2025-03-10  Notice          admin           Configuration C ...  system: "location": from 'other
     12:18:44    Security        administrator   System               location' → 'My Location'
                                                 0x5000000000080000
207  2025-03-10  Notice          admin           Configuration C ...  system: "hostname": from
     12:18:44    Security        administrator   System               'example' → 'MySwitch'
                                                 0x5000000000080000
Page 1 of 48 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all

MySwitch:/#>
```

As we can see they all share the same Event ID, but the message indicates the specific values that were changed. In this case we can see that we changed **“contact”**, **“location”**, and **“hostname”** settings under the `system` configuration.

Configuration Transaction Events

The following Event IDs are available for the Configuration Transaction category:

Event ID	Event Name	Description	Note
`0x5100000000010000`	Start	Configuration transaction started.	-
`0x5100000000010001`	Start - Error	Error occurred while starting configuration transaction.	-
`0x5100000000020000`	End	Configuration transaction ended.	-
`0x5100000000020001`	End - Validation Fail	Configuration transaction ended with validation failure.	-
`0x5100000000020002`	End - Unchanged	Configuration transaction ended with no changes applied.	-

Event ID	Event Name	Description	Note
`0x5100000000020003`	Activation	Configuration transaction activated changes.	-
`0x5100000000020004`	End - Error	Error occurred while ending configuration transaction.	-
`0x5100000000030000`	Abort	Configuration transaction aborted.	-
`0x5100000000040000`	Lock	Configuration transaction lock acquired.	-
`0x5100000000050001`	Unlock	Configuration transaction lock released.	-
`0x5100000000060002`	Unlock - Error	Error occurred while releasing configuration transaction lock.	-

Table 7: Configuration Transaction Events

Audit Log Events

The following Event IDs are available for the Audit Log Event category:

Event ID	Event Name	Description	Note
`0x6000000000010000`	Display	The audit log was displayed or viewed via CLI command.	-
`0x6000000000010001`	Display Category	Audit log entries were displayed filtered by a specific category.	-
`0x6000000000020000`	Clear Ring Buffer	The internal audit log ring buffer was cleared, removing all stored audit events.	-
`0x6000000000030000`	Reload wauditd	The audit daemon (wauditd) was reloaded to apply configuration changes.	-
`0x6000000000040000`	Export	Audit log entries were exported to an external file or destination.	-
`0x6000000000050000`	Search	A search operation was performed on the audit log using pattern matching or regex.	-
`0x6000000000060000`	Time Search	A time-based search was performed on the audit log to find entries within a specific time range.	-
`0x6000000000070000`	Display from Web	The audit log was displayed or accessed via the web interface.	-
`0x6000000000080000`	Remote Shell Access Enabled	Remote shell access (e.g., SSH) was enabled on the system.	-
`0x6000000000080001`	Remote Shell Access changed from outside CLI	Remote shell access setting was modified from outside the CLI (e.g., via SSH or web interface).	-

Table 8: Audit Log Events

File System Events

The following Event IDs are available for the File System Event category:

Event ID	Event Name	Description	Note
`0x7000000000010000`	Copy	A file or directory was copied in the file system.	-
`0x7000000000010001`	Copy error	An error occurred during a file or directory copy operation.	-
`0x7000000000020000`	View	File view operation performed (e.g., file was opened or read).	-
`0x7000000000020001`	View error	An error occurred while viewing or accessing a file or directory.	-
`0x7000000000020002`	View diff	A difference between files was viewed.	-
`0x7000000000030000`	Erase	A file or directory was erased or deleted from the file system.	-
`0x7000000000030001`	Erase error	An error occurred during a file or directory erase operation.	-
`0x7000000000040000`	List directory	A directory listing operation was performed.	-
`0x7000000000040001`	List directory error	An error occurred during a directory listing operation.	-
`0x7000000000050001`	CRL downloaded	A certificate revocation list (CRL) was downloaded to the device.	-
`0x7000000000050002`	Config file created	A PKI configuration file was created on the device.	-
`0x7000000000050003`	Certificate enrolled	A certificate was successfully enrolled using SCEP.	-
`0x7000000000050004`	Certificate renewed	A certificate was successfully renewed using SCEP.	-
`0x7000000000050005`	PKI artifact import operation	A PKI artifact was imported into the device.	-
`0x7000000000050006`	PKI artifact generate operation	A PKI artifact was generated on the device.	-
`0x7000000000050007`	PKI artifact remove operation	A PKI artifact was removed from the device.	-
`0x7000000000050008`	Certificate enrollment failed.	A certificate enrollment operation failed.	-
`0x7000000000050009`	Certificate renewal failed.	A certificate renewal operation failed.	-
`0x700000000005000a`	PKI artifact import operation Fail	PKI artifact import operation failed.	-
`0x7000000000060000`	File Transfer Import	File transfer import operation performed.	-
`0x7000000000060001`	File Transfer Import Fail	File transfer import operation failed.	-

Event ID	Event Name	Description	Note
`0x7000000000060002`	File Transfer Export	File transfer export operation performed.	-
`0x7000000000060003`	File Transfer Export Fail	File transfer export operation failed.	-
`0x7000000000070000`	Upgrade Image Download	Upgrade image download operation performed.	-
`0x7000000000070001`	Upgrade Image Download Fail	Upgrade image download operation failed.	-
`0x7000000000080000`	Package File Download	Package file download operation performed.	-
`0x7000000000080001`	Package File Download Fail	Package file download operation failed.	-
`0x7000000000090000`	Unlock License File Import	Unlock License file import operation performed.	-
`0x7000000000090001`	Unlock License File Import Fail	Unlock License file import operation failed.	-
`0x70000000000a0000`	App Image Import	App Image Import.	-
`0x70000000000a0001`	App Image Import Fail	App Image Import Fail.	-

Table 9: File System Events

Configuration Integrity Events

The following Event IDs are available for the Configuration Integrity Event category:

Event ID	Event Name	Description	Note
`0x8000000000010000`	Integrity Alert	Configuration file integrity verification failed. For instance, file hash does not match stored hash, or no fingerprint exists for the file.	-
`0x8000000000020000`	Integrity OK	Configuration file integrity verification successful. File hash matches stored hash.	-
`0x8000000000030000`	Integrity Warning	Configuration file integrity warning. For instance, hash not found in integrity database or failed to erase fingerprint.	-
`0x8000000000040000`	Update Integrity	Configuration file hash updated in the integrity database.	-
`0x8000000000050000`	Add to Integrity DB	Configuration file fingerprint added to the integrity database for tracking.	-
`0x8000000000060001`	CSTINFO file upload: validation successful	CSTINFO data file uploaded and passed correctness validation (syntax and semantic rules).	-

Event ID	Event Name	Description	Note
`0x8000000000060002`	CSTINFO file upload: validation failed, syntax error	CSTINFO data file upload failed: syntax validation error.	-
`0x8000000000060003`	CSTINFO file upload: validation failed, semantic error	CSTINFO data file upload failed: semantic correctness rules violation.	-
`0x8000000000060004`	CSTINFO file missing - ECSP services limited	CSTINFO data file not found at startup; ECSP services are limited.	-
`0x8000000000060005`	CSTINFO file syntax error - ECSP services limited	CSTINFO data file failed syntax validation at startup; ECSP services are limited.	-
`0x8000000000060006`	CSTINFO file semantic error - ECSP services limited	CSTINFO data file failed semantic correctness validation at startup; ECSP services are limited.	-
`0x8000000000060007`	CSTINFO file hash not stored - contents may be unreliable	CSTINFO data file hash not found in configuration; file integrity cannot be verified.	-
`0x8000000000060008`	CSTINFO file hash mismatch - contents may be unreliable	CSTINFO data file hash differs from stored value; file may have been modified unintentionally.	-
`0x8000000000060009`	CSTINFO file hash validation successful	CSTINFO data file passed all validation checks (correctness and integrity).	-
`0x8000000000070000`	Custom Telegram Dataset File Ok	Custom TRDP telegram dataset file passed all validation checks (correctness and integrity).	-
`0x8000000000070001`	Custom Telegram Dataset File Missing	Custom TRDP telegram dataset file not found; custom telegrams are disabled.	-
`0x8000000000070002`	Custom Telegram Dataset File Error	Custom TRDP telegram dataset file failed validation (syntax or semantic error); custom telegrams are disabled.	-
`0x8000000000070003`	Custom Telegram Dataset File Hash Mismatch	Custom TRDP telegram dataset file hash differs from stored value; file may have been modified unintentionally.	-
`0x8000000000070004`	Custom Telegram Dataset File Hash Accepted	Custom TRDP telegram dataset file hash accepted by operator after validation.	-

Table 10: Configuration Integrity Events

Boot Process Events

The following Event IDs are available for the Boot Process Event category:

Event ID	Event Name	Description	Note
`0x9000000000010000`	Boot Loader Configuration Integrity	Boot Loader Configuration Integrity verification status.	-
`0x9000000000010010`	ID-mem Integrity	ID-mem Integrity verification status. Note: The ID-mem integrity verification should not fail during operation. If it does, this may indicate that the product definition of the product has been manipulated.	-

Table 11: Boot Process Events

Syslog Message Format for Auditable Events

For each audit event, a syslog message is by default generated and sent to the any configured logging destinations. For the audit events, the syslog message is formatted as follows:

```
type="audit"; eventid=<EVENTID>; username=<USERNAME>; userid=<USERID>; userrole=<ROLE>; seqnum=<NUM>;
eventidtext="<EVENT_TEXT>"; msg="<MESSAGE>";
```

The different fields are constructed so that they should be easily parsable. The fields are as follows:

Field	Description
`type`	The type of the message, in this case `audit`, and it will always be `audit` for audit events.
`eventid`	The unique event ID for the audit event, where `<EVENTID>` is the hexadecimal representation of the event ID.
`username`	The username of the user that generated the audit event. Where `<USERNAME>` is the username of the user.
`userid`	The user ID of the user that generated the audit event. Where `<USERID>` is the user ID of the user, as an integer.
`userrole`	The role of the user that generated the audit event. Where `<ROLE>` is the role of the user.
`seqnum`	The sequence number of the audit event. Where `<NUM>` is the sequence number of the event, as an integer.
`eventidtext`	The text representation of the event ID. Where `<EVENT_TEXT>` is the text representation of the event ID and category.
`msg`	A message that provides additional information about the audit event. Where `<MESSAGE>` is the specific message for the event.

Example of an actual syslog message generated for an audit event, in this case a Link Up event:

```
type="audit"; eventid=0x3000000000020001; username=root; userid=0; userrole=system; seqnum=57;
eventidtext="Control System Event - Link Up"; msg="port/interface ethX8";
```

> **NOTE ON THE SYSLOG MESSAGE FORMAT**
>
> This is simply the message part of the syslog message, the header part is not included here. For information on the full syslog message format, please refer to the Logging documentation.

SENDING AUDITABLE EVENTS WITH SYSLOG

When setting up logging on the device, one aspect is to specify the source of the log messages. When wanting to send log messages that originate from the device a source of type `local` is configured. In turn, this type can select a number of different local sources, one of them being `audit`. Therefore, when this local source type is selected, only log messages that fall within these auditable events will be sent to the configured logging destinations.

For more information on setting up logging sources, please refer to the Logging documentation.

If for some reason the syslog messages should not be generated for the auditable events, this can be configured in the CLI. Again, the auditable events will still be generated (unless they too are disabled), but they will not be sent to syslog, they will be stored in the local audit log ring buffer on the device.

Configuration

Configuration options related Audit can be found in the top-level configuration context in the CLI:

```
example:/#> configure
example:/config/#> audit
example:/config/audit/#>
```

`[no] enable [CATEGORY]`

Description

Enable or disable audit event logging

This setting controls whether audit events are logged or not. It is enabled or disabled on a per event category basis.

Default: Enabled, for all categories of audit events.

Example

Enable audit event logging for all categories:

```
example:/config/audit/#> enable
```

Disable audit event logging for all categories:

```
example:/config/audit/#> no enable
```

Enable audit event logging for a specific category:

```
example:/config/audit/#> enable access-control
```

Disable audit event logging for a specific category:

```
example:/config/audit/#> no enable access-control
```

continues on next page

continued

`[no] enable [CATEGORY]`

Multiple categories can be enabled or disabled at the same time:

```
example:/config/audit/#> enable access-control request-error
```

Options

Option	Description
`no`	Disable audit event logging. If a specific category is provided, only that category will be disabled. If no category is provided, all categories will be disabled.
`CATEGORY`	Based on the list of available audit event categories, this is the category to enable or disable. If no category is provided, all categories will be enabled or disabled. **TAB COMPLETION** When providing the category, tab completion can be used to list all available categories.

`[no] syslog`

Description

Enable or disable sending audit log messages to syslog

This setting controls whether audit events are sent to syslog or not.

> **NOTE**
>
> Auditable event messages are always generated, if `enabled`, regardless of whether they are sent to syslog or not. On the device itself, they are stored separately from the syslog messages. If the auditable events are to be sent to a logging destination, this setting must be enabled.

Default: Enabled

Example

Enable sending audit log messages to syslog:

```
example:/config/audit/#> syslog
```

Disable sending audit log messages to syslog:

```
example:/config/audit/#> no syslog
```

continues on next page

continued

`[no] syslog`

Options

Option	Description
`no`	Disable sending audit log messages to syslog.

Operational Commands

Operational commands related to auditable events can be found in the audit context, located top-level `exec` context in the CLI:

```
example:/#> audit
example:/audit/#>
```

`list [eventid | category]`

Description

List the audit Categories and EventIDs that exist in the system.

This command will list all of the existing audit event categories and event IDs that can be available for the system.

> **NOTE ON THE DISPLAYED EVENT IDS**
>
> This list shows every possible event ID and category that can be generated by the system. When these events can be generated is dependent on the configuration of the system.

Example

List all available audit event categories and event IDs:

```
example:/audit/#> list
...
```

Options

Option	Description
`eventid`	List only all available audit event IDs.
`category`	List only all available audit event categories.

`[show] status`

Description

Display basic status information about the internal audit ring buffer.

This command will display the current status of the internal audit ring buffer, including the number of entries currently stored in the buffer.

continues on next page

continued

`[show] status`

Example

Display the current status of the audit ring buffer:

```
example:/audit/#> status
AUDIT RING-BUFFER STATUS
Ring buffer count     : 72
Ring buffer used size : 5.41 KB
Ring buffer max size  : 1.00 MB
```

`clear`

Description

Clear the internal audit ring buffer.

This command will clear the internal audit ring buffer, removing all locally stored audit events.

> **AUTHORISED USERS ONLY**
>
> This command is only accessible by `administrator` level users.

Example

Clear the internal audit ring buffer:

```
example:/audit/#> clear
Are you sure you want to clear the audit log? (y/N) y
Clearing audit log ...
example:/audit/#>
```

Viewing Auditable Events

Auditable events can be accessed and viewed in the CLI, from the `audit` context, accessed from the top level `exec` context:

```
example:/#> audit
example:/audit/#>
```

By default, when show commands are executed, the audit log will be attempted to be displayed in an interactive mode. This mode will display the audit log in a paginated format. If this is undesired, the entire terminal can be set to `no interactive`. Be aware that this will set no interactive mode for the entire CLI. Simply call `interactive` to re-enable the interactive mode.

> **INTERACTIVE MODE OVER CONSOLE CONNECTION**
>
> Be aware that by default when accessing the audit log though the CLI, the CLI will attempt to display the audit log in an interactive mode. Doing this over a console connection can be sluggish, as the output speed is limited by the baud rate of the connection.

> **INTERACTIVE MODE AND TERMINAL SIZE**
>
> The interactive mode will attempt to display the audit log in a paginated format, based on the terminal size.
>
> If the terminal size is not deemed large enough, the interactive mode will not be used, and the output will be displayed in a single page.

Viewing Entires in the Pager

When displaying the audit log in interactive mode, the output will be displayed in a paginated format. The following commands can be used to navigate the output:

Command	Key	Description
Previous Page	`a`	Go to the previous page.
Next Page	`d`	Go to the next page.
Quit	`q`	Quit the pager. It should also be possible to exit the pager with `CTRL+C`. Simply pressing the enter key again will also exit the pager.
Refresh	`r`	Refresh the current page. This will also resize the terminal window.
Go to Page	`g`	Go to a specific page. You will be prompted to enter the page number.
List All	`l`	Display all entries at once. This will print all the entries and exit the pager.

Commands

`show`

Description

Display the entire internal audit ring buffer.

This command will display the entire internal audit ring buffer, showing all of the stored audit events.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

Example

Display the entire audit ring buffer:

```
example:/audit/#> show
- Audit log ring buffer, entries 1-4 of 149 ----------------------------------------------------
SEQ  TIME        LEVEL/FACILITY  USER/ROLE      EVENTID            MESSAGE
218  2025-03-10  Notice          root           Access Control     Authentication successful for
     12:48:37    Auth            system         SSH login Success  user 'admin' from 198.18.1.99
                                                0x1000000000010020
217  2025-03-10  Notice          root           Access Control     SSH login exit for user
     12:48:34    Auth            system         SSH login Exit     'admin' from 198.18.1.99:33202
                                                0x1000000000010021
216  2025-03-10  Info            admin          Audit Log Event    Entire audit log displayed
     12:48:14    Security        administrator  Display            from the CLI.
```

continues on next page

continued

`show`

```
                                          0x6000000000010000
215  2025-03-10  Info            root          Control System ...   interface vlan1 ip
     12:42:57    Security        system        DHCP Client Lea ...  198.18.1.101 mask 24 broadcast
                                               0x3000000000050003   198.18.1.255 router 198.18.1.99

Page 1 of 38 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
example:/#>
```

`show last <NUMBER>`

Description

Display the last `NUMBER` of entries in the internal audit ring buffer.

This command will display the last `NUMBER` of entries in the internal audit ring buffer.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

Example

Display the last 3 entries in the audit ring buffer:

```
example:/audit/#> show last 3
= Audit log ring buffer, entries 1-3 of 3 ==========================================
SEQ  TIME        LEVEL/FACILITY  USER/ROLE      EVENTID            MESSAGE
218  2025-03-10  Notice          root           Access Control     Authentication successful for
     12:48:37    Auth            system         SSH login Success  user 'admin' from 198.18.1.99
                                                0x1000000000010020

217  2025-03-10  Notice          root           Access Control     SSH login exit for user
     12:48:34    Auth            system         SSH login Exit     'admin' from 198.18.1.99:33202
                                                0x1000000000010021

216  2025-03-10  Info            admin          Audit Log Event    Entire audit log displayed
     12:48:14    Security        administrator  Display            from the CLI.
                                                0x6000000000010000

Page 1 of 1 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
example:/#>
```

Options

Option	Description
`NUMBER`	The number of entries to display, provided as an integer.

`show range <START> <NUM>`

Description

Show a range of entries in the audit ring buffer.

The range is specified by a starting entry number `START` and the number of entries to display `NUM`, from the starting entry.

continues on next page

continued

`show range <START> <NUM>`

The `START` number is ordered from lowest to highest, with 0 being the latest entry in the ring buffer. Entries are displayed backwards from `START`, so if `START` is 0 and `NUM` is 5, you will see the five latest entries.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

Example

Display 3 entries beginning from the 6th entry in the audit ring buffer:

```
example:/audit/#> show range 6 3
 Audit log ring buffer, entries 6-8 of 8
SEQ  TIME        LEVEL/FACILITY  USER/ROLE      EVENTID            MESSAGE
219  2025-03-10  Info            admin          Audit Log Event    Entire audit log displayed
     12:48:46    Security        administrator  Display            from the CLI.
                                                0x6000000000010000

218  2025-03-10  Notice          root           Access Control     Authentication successful for
     12:48:37    Auth            system         SSH login Success  user 'admin' from 198.18.1.99
                                                0x1000000000010020

217  2025-03-10  Notice          root           Access Control     SSH login exit for user
     12:48:34    Auth            system         SSH login Exit     'admin' from 198.18.1.99:33202
                                                0x1000000000010021

Page 1 of 1 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
example:/#>
```

Options

Option	Description
`START`	The starting entry number, provided as an integer. The starting entry is ordered from lowest to highest, with 0 being the latest entry in the ring buffer.
`NUM`	The number of entries to display, provided as an integer, from the starting entry.

`show time [[YYYY-MM[-DD]] [hh:mm[:ss]]] [[YYYY-MM[-DD]] [hh:mm[:ss]]]`

Description

Display the entries in the internal audit ring buffer within the specified time range.

This command will display the entries in the internal audit ring buffer that fall within the specified time range.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

continues on next page

continued

`show time [[YYYY-MM[-DD]] [hh:mm[:ss]]] [[YYYY-MM[-DD]] [hh:mm[:ss]]]`

Example

Display the entries in the audit ring buffer that fall within the time range from `2025-03-10 12:48:30` to `2025-03-10 12:48:50`:

```
example:/audit/#> show time 2025-03-10 12:48:30 2025-03-10 12:48:50
╒ Audit log ring buffer, entries 1-3 of 3 ═══════════════════════════════════
│SEQ  TIME        LEVEL/FACILITY  USER/ROLE      EVENTID            MESSAGE
│219  2025-03-10  Info            admin          Audit Log Event    Entire audit log displayed
│     12:48:46    Security        administrator  Display            from the CLI.
│                                                0x6000000000010000
│
│218  2025-03-10  Notice          root           Access Control     Authentication successful for
│     12:48:37    Auth            system         SSH login Success  user 'admin' from 198.18.1.99
│                                                0x1000000000010020
│
│217  2025-03-10  Notice          root           Access Control     SSH login exit for user
│     12:48:34    Auth            system         SSH login Exit     'admin' from 198.18.1.99:33202
│                                                0x1000000000010021
Page 1 of 1 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
```

Options

Option	Description
`YYYY`	The year, provided as an integer.
`MM`	The month, provided as an integer between 1 and 12.
`DD`	The day, provided as an integer between 1 and 31.
`hh`	The hour, provided as an integer between 0 and 23.
`mm`	The minute, provided as an integer between 0 and 59.
`ss`	The second, provided as an integer between 0 and 59.

`show from [[YYYY-MM[-DD]] [hh:mm[:ss]]]`

Description

Display the entries in the internal audit ring buffer from the specified time.

This command will display the entries in the internal audit ring buffer that have been generated with a timestamp that falls after the specified time.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

Example

Display the entries in the audit ring buffer that have been generated from `2025-03-10 12:48:30`:

```
example:/audit/#> show from 2025-03-10 12:48:30
╒ Audit log ring buffer, entries 9-12 of 12 ═══════════════════════════════
│SEQ  TIME        LEVEL/FACILITY  USER/ROLE      EVENTID            MESSAGE
```

continues on next page

continued

`show from [[YYYY-MM[-DD]] [hh:mm[:ss]]]`

```
220  2025-03-10  Info            admin          Audit Log Event     Entire audit log displayed
     12:50:53    Security        administrator  Display             from the CLI.
                                                0x6000000000010000
219  2025-03-10  Info            admin          Audit Log Event     Entire audit log displayed
     12:48:46    Security        administrator  Display             from the CLI.
                                                0x6000000000010000
218  2025-03-10  Notice          root           Access Control      Authentication successful for
     12:48:37    Auth            system         SSH login Success   user 'admin' from 198.18.1.99
                                                0x1000000000010020
217  2025-03-10  Notice          root           Access Control      SSH login exit for user
     12:48:34    Auth            system         SSH login Exit      'admin' from 198.18.1.99:33202
                                                0x1000000000010021
Page 2 of 2 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
```

Options

Option	Description
`YYYY`	The year, provided as an integer.
`MM`	The month, provided as an integer between 1 and 12.
`DD`	The day, provided as an integer between 1 and 31.
`hh`	The hour, provided as an integer between 0 and 23.
`mm`	The minute, provided as an integer between 0 and 59.
`ss`	The second, provided as an integer between 0 and 59.

`show to [[YYYY-MM[-DD]] [hh:mm[:ss]]]`

Description

Display the entries in the internal audit ring buffer to the specified time.

This command will display the entries in the internal audit ring buffer that have been generated with a timestamp that falls before the specified time.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

Example

Display the entries in the audit ring buffer that have been generated to `2025-03-10 12:48:50`:

```
example:/audit/#> show to 2025-03-10 12:48:50
Audit log ring buffer, entries 1-3 of 148
SEQ  TIME        LEVEL/FACILITY  USER/ROLE      EVENTID             MESSAGE
219  2025-03-10  Info            admin          Audit Log Event     Entire audit log displayed
     12:48:46    Security        administrator  Display             from the CLI.
                                                0x6000000000010000
218  2025-03-10  Notice          root           Access Control      Authentication successful for
     12:48:37    Auth            system         SSH login Success   user 'admin' from 198.18.1.99
```

continues on next page

continued

`show to [[YYYY-MM[-DD]] [hh:mm[:ss]]]`

```
                                        0x1000000000010020
217  2025-03-10  Notice        root        Access Control      SSH login exit for user
     12:48:34    Auth          system      SSH login Exit      'admin' from 198.18.1.99:33202
                                           0x1000000000010021
Page 1 of 50 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
```

Options

Option	Description
`YYYY`	The year, provided as an integer.
`MM`	The month, provided as an integer between 1 and 12.
`DD`	The day, provided as an integer between 1 and 31.
`hh`	The hour, provided as an integer between 0 and 23.
`mm`	The minute, provided as an integer between 0 and 59.
`ss`	The second, provided as an integer between 0 and 59.

`show search [case] "STRING"`

Description

Display the entries in the internal audit ring buffer that contain the specified string.

This command will display the entries in the internal audit ring buffer that contain the specified string. The provided search string supports regular expressions. The search string will search based on the Event ID and the message of the audit event.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

Example

Display the entries in the audit ring buffer that contain the string `SSH login`:

```
example:/audit/#> show search "SSH login"
Audit log ring buffer, entries 1-3 of 3
SEQ  TIME        LEVEL/FACILITY  USER/ROLE  EVENTID            MESSAGE
218  2025-03-10  Notice         root       Access Control     Authentication successful for user
     12:48:37    Auth           system     SSH login Success  'admin' from 198.18.1.99
                                           0x1000000000010020
217  2025-03-10  Notice         root       Access Control     SSH login exit for user 'admin'
     12:48:34    Auth           system     SSH login Exit     from 198.18.1.99:33202
                                           0x1000000000010021
115  2025-03-10  Notice         root       Access Control     Authentication successful for user
     12:13:07    Auth           system     SSH login Success  'admin' from 198.18.1.99
                                           0x1000000000010020
Page 1 of 1 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
```

continues on next page

continued

`show search [case] "STRING"`

Options

Option	Description
`case`	If the `case` argument is provided, the search will be case sensitive.
`"STRING"`	The string to search for, provided in double quotes. This is a free-form string that supports regular expressions.

`show category CATEGORY`

Description

Display the entries in the internal audit ring buffer that belong to the specified category.

This command will display the entries in the internal audit ring buffer that belong to the specified category.

> **INTERACTIVE MODE**
>
> The output will, by default, be displayed in an interactive mode, if the terminal allows for it.

Example

Display the entries in the audit ring buffer that belong to the `access-control` category:

```
example:/audit/#> show category "access-control"
╒ Audit log ring buffer, entries 1-5 of 5 ═══════════════════════════════════
 SEQ  TIME        LEVEL/FACILITY  USER/ROLE  EVENTID            MESSAGE
 218  2025-03-10  Notice          root       Access Control     Authentication successful for user
      12:48:37    Auth            system     SSH login Success  'admin' from 198.18.1.99
                                             0x1000000000010020
 217  2025-03-10  Notice          root       Access Control     SSH login exit for user 'admin'
      12:48:34    Auth            system     SSH login Exit     from 198.18.1.99:33202
                                             0x1000000000010021
 115  2025-03-10  Notice          root       Access Control     Authentication successful for user
      12:13:07    Auth            system     SSH login Success  'admin' from 198.18.1.99
                                             0x1000000000010020
 111  2025-03-10  Notice          admin      Access Control     CLI Domain shell exited
      12:12:51    Auth            admini ...   Exit CLI Shell
                                             0x1000000000030001
 110  2025-03-10  Notice          admin      Access Control     CLI Domain shell accessed
      12:12:50    Auth            admini ...   Enter CLI Shell
                                             0x1000000000030000
Page 1 of 1 | Navigation: 'a' prev, 'd' next, 'q' quit, 'r' refresh, 'g' goto, 'l' display all
```

Options

Option	Description
`CATEGORY`	The category to display, based on the available categories.

continues on next page

continued

`show category CATEGORY`

Option	Description
	TAB COMPLETION The available categories can be tab completed in the CLI.

`show raw`

Description

Display the raw audit log file.

This command will display the raw audit log file, showing all of the stored audit events. This is not displaying from the ring buffer storing the audit events, but from a local log file written with syslog. Therefore, the output in this file is how the audit messages would look also when sent to a remote syslog server.

> **NUMBER OF ENTRIES**
>
> The number of entries that are stored in the raw audit log file is most likely going to be a lot less than what is stored in the internal audit ring buffer.

Example

Display the raw audit log file:

```
example:/audit/#> show raw
Mar 10 12:48:34 MySwitch wauditd[1624]: type="audit";
eventid=0×1000000000010021; username=root; userid=0; userrole=system;
seqnum=217; eventidtext="Access Control - SSH login Exit"; msg="SSH login exit
for user 'admin' from 198.18.1.99:33202";
Mar 10 12:48:37 MySwitch wauditd[1624]: type="audit";
eventid=0×1000000000010020; username=root; userid=0; userrole=system;
seqnum=218; eventidtext="Access Control - SSH login Success";
msg="Authentication successful for user 'admin' from 198.18.1.99";
Mar 10 12:48:46 MySwitch wauditd[1624]: type="audit";
eventid=0×6000000000010000; username=admin; userid=500; userrole=administrator;
seqnum=219; eventidtext="Audit Log Event - Display"; msg="Entire audit log
displayed from the CLI.";
```

2.7.3. Logging Support

Introduction

The system produces a wealth of information in log files. Most of the log messages generated in the system are that of general events from the various system components. For some of the more important events that are logged, see Security Related Events.

The following methods for logging are available:

- **Logging of default log files:** By default all logs are written to an internal RAM disk. Log files written to the internal RAM disk are *not persistent* across reboots.
- **Logging of default log files to USB:** When enabled it will log the same things that are logged to the internal RAM disk, but it will be done to a connected external USB drive.
- **Logging to console port:** It is possible to direct logging messages to the console port. Messages of severity level DEBUG or higher are shown on the console port.
- **Logging to user defined sinks:** The logging system supports creation of customised logging sources, destinations and filters. These separate components are combined together to create a logging sink. This is a very flexible way to configure logging in the system. It allows for logging to remote machines, local files, or files on external media (USB or SD Card). The sources can also be configured to receive message from other remote machines. The optional filters can also be used to more precisely control what messages are logged.

> **SEE ALSO**
>
> Related Examples and Use-Cases
> - Logging to Remote Machines
> - Logging and DHCP Option 7
> - Logging Relay
> - Logging and Receiving From Remote Clients

Overview

The system continuously generates a wealth of information in the form of log messages. These messages serve as a vital resource for monitoring system performance, troubleshooting issues, and understanding system activities.

By default, all log messages are stored on the internal RAM disk. However, the system also provides flexibility in log management. You can configure it to direct these messages to custom log files located on either internal or external media. Alternatively, you can set up the system to forward these messages to remote log servers for centralized logging.

Before we delve into the specifics of configuring the logging system, it's crucial to grasp the fundamentals of syslog messages. Understanding what a syslog message is and how it's structured will provide a solid foundation for the subsequent sections.

The Anatomy of a Syslog Message

Depending on the syslog format that is used, a few different fields can be found in a syslog message, in addition to the actual message itself. Regardless of the format, something referred to as **Facility** and **Severity** are always present in a syslog message:

- **Facility:** The facility indicates the system component that generated the log message. It is used to categorise the message based on the type of system component that generated it. The facility

is represented by a number between 0 and 23, where each number corresponds to a specific system component.

These facilities, and some description of how the systems attempts to conform to them, can be seen in **Table 1** below.

#	Facility	Description
0	kern	Kernel log messages
1	user	User-level messages
2	mail	Unused
3	daemon	System daemons
4	auth	Security/Authorisation messages
5	syslog	Unused
6	lpr	Unused
7	news	Unused
8	uucp	Unused
9	cron	Unused
10	authpriv	Unused
11	ftp	Unused
12	ntp	NTP
13	security	Log audit, for audit trail
14	console	Log alert
16	local0	Alarm sub-system
17	local1	Unused
18	local2	PPP
19	local3	Unused
20	local4	OpenVPN, IPsec
21	local5	Reserved, OEM customer specific
22	local6	Unused
23	local7	Unused

Table 1: All of the standard syslog facilities and their uses in the system.

> **NOTE**
>
> The facilities `security` and `console` are displayed as `LOGAUDIT` and `LOGALERT` in Wireshark, which is the de facto tool to debug all network traffic.

- **Severity:** The severity indicates the level of urgency or criticality of the message. It is used to classify the message based on its importance. The severity is represented by a number between 0 and 7, where each number corresponds to a specific level of criticality.

  These severities, and some description of how the systems attempts to conform to them, can be seen in **Table 2** below.

#	Severity	Description
0	emerg	System level service only
1	alert	System level service only
2	crit	System level service only
3	err	Severe error, daemon/service may restart
4	warning	Significant problem, e.g. no connection to Radius server
5	notice	General log message, e.g. login successful
6	info	Informational, less important
7	debug	Developer/low-level debug messages

Table 2: All of the standard syslog severities and their general usage in the system.

The system allows for selection of what format to use, when configuring custom logging destinations. The different formats will be covered briefly in the following sections.

Log Message Formatting

For the examples below, we will use an example log message generated in the following manner: Sent from the program `proc` with the process ID `123`, and the message `Example Message.`. The message is generated with the following facility and severity: `user` and `notice`.

> **NOTE**
>
> The **Facility** and **Severity** values of a syslog message are always present and combined into a single value known as the *Priority*. This *Priority* is computed by multiplying the *Facility* value by 8 and then adding the *Severity* value. For instance, if the *Facility* value is 1 and the *Severity* value is 5, the *Priority* would be calculated as `8 * 1 + 5 = 13`.
>
> In binary, the *Priority* value is represented as an 8-bit number. For the above example, the *Priority* value of 13 would be represented as `00001101`.
>
> In raw output, such as a Wireshark capture, the *Priority* is displayed within angle brackets, like `<13>`.

> **NOTE**
>
> The names of the fields described below, are somewhat based on how they will be displayed in a Wireshark capture.

The system supports three different syslog formats:

- **BSD:** A legacy BSD format. Given our example message, the raw format would look like this:

  `<13>proc[123]: Example message.`

  The individual fields are interpreted as follows:

  - **Priority:** The value within the angle brackets `<13>` is the combined Facility and Severity value.
  - **Message:** The remainder of the text following the colon `:` is the actual log message.

> **NOTE**
>
> In the BSD format, the program name and process ID are not distinctly separated from the message. They are included as part of the overall message text.

- **RFC 3164:** This is a standardised syslog format defined in RFC 3164[1]. In this format, our example message would appear as follows:

  `<13>May 21 10:36:31 example proc[123]: Message.`

  The individual fields are represented by specific parts of the raw message:

  - **Priority:** The value `<13>` within the angle brackets represents the Priority, which is a combined value of Facility and Severity. This value is calculated by multiplying the Facility value by 8 and then adding the Severity value.
  - **Timestamp:** The `May 21 10:36:31` part of the message represents the Timestamp, indicating the date and time when the message was generated. The timestamp is formatted according to the local time and date of the machine that generated the message.
  - **Hostname:** The `example` part of the message indicates the Hostname or IP address of the machine that generated the message. The hostname is used to identify the source of the message in a networked environment.
  - **Process ID:** The `proc[123]` part of the message is the Tag, which usually consists of the process name `proc` and process ID `123`. The tag is used to identify the specific process that generated the message.
  - **Message ID:** The `Message.` part of the message following the colon is the actual log Message. This message contains the detailed information about the event that triggered the log entry.

  > **NOTE**
  >
  > When viewed in a Wireshark capture, the PID `123` will be displayed as part of the **Message ID** filed, before the actual message.

- **RFC 5424:** This is a more modern syslog format defined in RFC 5424[2]. In this format, our example message would appear as follows:

  `<13>1 2024-05-21T12:18:24+00:00 example proc 123 - [meta sequenceId="4"] Message.`

  The individual fields are represented by specific parts of the raw message:

  - **Priority:** The value `<13>` within the angle brackets represents the Priority, which is a combined value of Facility and Severity. This value is calculated by multiplying the Facility value by 8 and then adding the Severity value.
  - **Version:** The `1` following the priority field represents the version of the syslog protocol, as defined in RFC 5424. This version number, '1', signifies that the message conforms to the first revision of the protocol that introduced changes to the header format.

[1] https://tools.ietf.org/html/rfc3164
[2] https://tools.ietf.org/html/rfc5424

- **Timestamp:** The `2024-05-21T12:18:24+00:00` part of the message represents the *Timestamp*, indicating the date and time when the message was generated. The timestamp is formatted according to the ISO 8601 standard.
- **Hostname:** The `example` part of the message indicates the *Hostname* or IP address of the machine that generated the message. The hostname is used to identify the source of the message in a networked environment.
- **Application Name:** The `proc` part of the message is the *Application Name*, which usually identifies the device or application that originated the message.
- **Process Id:** The `123` part of the message is the *Process ID*, which usually identifies the process that generated the message.
- **Message ID:** The `-` part of the message is the *Message ID*, which is a unique identifier for the type of message. In this case, no *Message ID* is provided.
- **Structured-Data:** The `[meta sequenceId="4"]` part of the message is the Structured-Data, which provides additional information about the message in a structured format.
- **Message:** The `Message.` part of the message following the Structured-Data is the actual log Message.

Default Log Files

As mentioned earlier, the system will always create a number of default log files. These log files are stored on the internal RAM disk, under the `/var/log/` directly. In **Table 3** below, the default log files along with their associated facilities and severities are listed.

Log File	Facility	Severity	Description
auth.log	auth, authpriv	*	Logs related to authentication are stored in **auth.log**.
kern.log	kern	*	Kernel logs are stored in **kern.log**.
alarm	local0	*	Alarm logs are stored in **alarm**.
cli	local1	*	Command Line Interface logs are stored in **cli**.
ppp.log	local2	*	Point-to-Point Protocol logs are stored in **ppp.log**.
ntp.log	local3, ntp	*	Network Time Protocol logs are stored in **ntp.log**.
openvpn.log	local4	*	OpenVPN logs are stored in **openvpn.log**.
syslog	All except auth, authpriv	*	Logs from all facilities except ‘auth’, ‘authpriv’ are stored in **syslog**.
messages	All except auth, authpriv, cron, kern, mail, news	*	Logs from all facilities except ‘auth’, ‘authpriv’, ‘cron’, ‘kern’, ‘mail’, ‘news’ are stored in **messages**.

Table 3: Default log files and their associated facilities and severities.

> **NOTE**
>
> There may be additional log files created by the system by default, that are not listed in the table above. However, the log files listed in the table should always be present in the system, if a message is generated that should be logged to one of these log files, based on the facility and severity of the message.

> **WARNING**
>
> As mentioned earlier, the default log files are stored on the internal RAM disk. This means that any log messages sent to a file on the internal media will be lost upon reboot.

The default log file in the system is the **messages** log file. This log file contains most of the log messages generated by the system. The **auth** log file contains log messages related to authentication, such as login attempts and authentication failures. The **kern** log file contains kernel log messages, providing information about the internal operations of the kernel. The **syslog** log file is also good to view if we want to see all log messages, including the kernel messages, except those related to authentication.

In order to list all the currently available log files located on the internal RAM disk, including the default log files, use the `dir` command after entering the file/log context.

```
example:/#> file log
example:/file/log#> dir
/log/                   MOD. DATE/LINK    SIZE  MD5
alarm                   2024-05-20 14:18   276
appendix.log            2024-05-20 14:18   118
auth.log                2024-05-20 14:18   754
bootsettings.log        2024-05-20 14:18    20
btmp                    2024-05-20 14:18     0
kern.log                2024-05-20 14:19  2772
lastlog                 2024-05-20 14:18     0
messages                2024-05-20 14:49  7717
mount.log               2024-05-20 14:18  1170
ntp.log                 2024-05-20 14:18   134
syslog                  2024-05-20 14:49 10489
wtmp                    2024-05-20 14:18  2304

Free: 19.10 MiB | Total: 20.1 MiB
example:/#>
```

> **NOTE**
>
> The individual size for each of the default log files is limited to 100K. Each file will be rotated up to one time, meaning that the system will keep the current log file and the previous log file. As an example, if the **messages** log file reaches 100K, it will be renamed to **messages.0** and a new **messages** log file will be created.
>
> Each default log file will also allow for two compressed log files to be retained. A log file is compressed when we reach the maximum number of rotations for a given log file. When this happens, the oldest rotated log file will be compressed. The compressed log files will be named **messages.1.gz** and **messages.2.gz**. The system will keep the most recent compressed log file and delete the oldest compressed log file when a new compressed log file is created.
>
> As an example, if we say that we have an average message size of 100-200 bytes, we can allow for around 500-1000 messages in each log file. This is a rough estimate, and the actual number of messages that can be stored in a log file will depend on the actual size of the messages.

To view the contents of the system main log file, the `show log` command can be used. As mentioned earlier, the **messages** log file is the main log file in the system, and it will be displayed if nothing else is specified:

```
example:/#> show log
[ ... ]
May 20 14:53:46 example finit[1]: Starting lldpd:, PID: 1058
May 20 14:53:47 example lldpcli[1059]: lldpd should resume operations
──────────────────────────────────────────────────────────────────────────────
messages                                          Mon May 20 14:53:47 2024

example:/#>
```

In order to view a specific log file, the `show logfile FILE` command can be used. For instance, to view the **auth.log** log file:

```
example:/#> show logfile auth.log
[ ... ]
May 20 14:18:15 example login[811]: Authentication successful for user 'admin'
May 20 14:18:15 example login[811]: pam_unix(login:session): session opened
for user admin(uid=0) by LOGIN(uid=0)
May 20 14:18:15 example login[912]: root login on 'console'
──────────────────────────────────────────────────────────────────────────────
auth.log                                          Mon May 20 14:18:15 2024

example:/#>
```

Remember the short-keys to the `show` command when listing long files:

- `R`: show the rest of the file
- `q`: Quit viewer

> **TIP**
>
> Also, the `follow` command is very useful when monitoring log files. It continuously lists new lines as they are appended to a file. Use `Ctrl-C` to stop and return to the command prompt.

```
example:/#> follow log
Press Ctrl-C to abort
May 20 14:53:46 S1 dnsmasq[1047]: read /etc/hosts - 3 names
May 20 14:53:46 S1 finit[1]: Starting lldpd:, PID: 1058
May 20 14:53:47 S1 lldpcli[1059]: lldpd should resume operations
```

Security Related Events

The system produces a lot of different log messages, but there are a few that are considered a bit more important in terms of security. As such they are listen below, in **Table 4**.

Event	Facility	Severity	Example
Login successful	auth	notice	Authentication successful for user 'admin' from 10.0.0.1
Login failed	auth	warning	Authentication failed for user 'admin' from 10.0.0.1
Audit trail	security	notice	User 'admin' changed the configuration
System startup	console	notice	\/ Westermo WeOS v5.7, entering runlevel 3
System shutdown	console	notice	\/ Westermo WeOS v5.7, entering runlevel 6
System update	console	notice	\/ Westermo WeOS v5.7, entering runlevel 9
Enter Maintenance mode	user	notice	Entering maintenance mode for upgrade to WeOS-5.7.x.pkg …
Exit Maintenance mode	user	notice	Exiting maintenance mode for upgrade
Service start	console	notice	Starting mdnsd:1, PID: 1203
Service stop	console	notice	Stopping mdnsd:1, PID: 1203, sending SIGTERM…
Service restarting	console	notice	Restarting mdnsd:1, PID: 1203, sending SIGHUP…
Service Died	console	warning	Service mdnsd:1 died, restarting (1/10)
Service FAIL	console	warning	Service mdnsd:1 keeps crashing, not restarting
Link up	console	notice	Link up on eth1
Link down	console	notice	Link down on eth1
DHCP Unknown host ID	console	warning	DHCPDISCOVER(11:22:33:44:55:66) vlan1 no address available
NTP reply not synced	ntp	info	reply from 10.0.0.1: not synced (alarm), next query 64s
NTP synced	ntp	info	clock is now synced
NTP peer valid	ntp	info	peer 10.0.0.1 now valid
NTP update FAIL	ntp	err	no reply received in time, skipping initial time setting
TTDP stack ready	daemon	notice	Bypass relays in normal position (upon startup; IEC 61375)
ECSC comm. established	daemon	notice	ECSP_CTRL traffic heard from ECSC (upon startup; IEC 61375)

Table 4: Security related events that are logged in the system.

The messages that are generated for these events should mostly conform to the examples given in **Table 4**. However, it is possible that the messages are slightly different. They are indicative of the type of messages that are generated for these events.

Logging Sink Overview

The system provides a highly customizable logging system that allows for the configuration of custom logging sources, destinations, and filters. These components are combined to create what is called a *sink*. A sink is a combination of the mentioned components and allows for the creation of customised logging setups.

In **Figure 1** below, we show an example of the relationship between the different logging components and how they can be combined to create a logging sink. This simple example includes two sources, two destinations, and two filters. There are two sinks that combine the components in the following way:

- **Sink-1:** Utilizes `Source-1`, `Source-2` and `Destination-1`.
- **Sink-2:** Utilizes `Source-2`, `Destination-2`, `Filter-1`, and `Filter-2`.

As we can see, `source-2` is used in both sinks, and `filter-1` and `filter-2` are only used in `sink-2` because they are optional components.

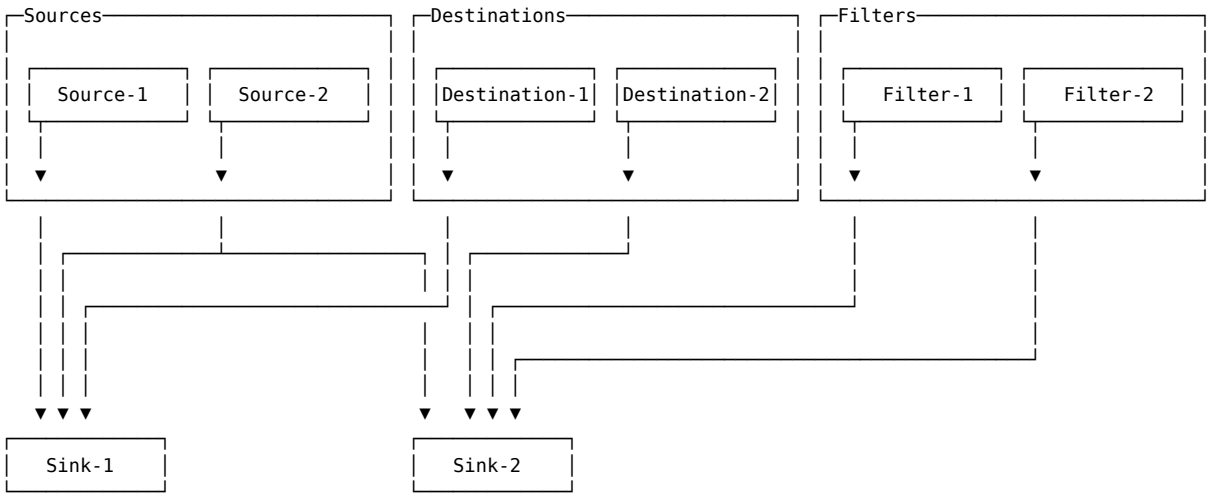


Figure 1: The relationship between the different logging components and how they are combined to create a logging sink. Components can be combined and be reused across multiple sinks.

Configuration of these logging sinks is what allows for specialised logging setups, such as logging to remote machines, local files, or files on external media (USB or SD Card).

> **NOTE**
>
> No log messages will be received from any source or sent to any destination unless a sink is configured to utilise them.

The way in which these components can be defined and combined allows for a very flexible and powerful logging system. The following sections will describe each of these components in more detail.

Supported Transport Protocols

Both logging sources and destinations can use the following transport protocols to send and receive log messages from remote machines:

- **UDP:** UDP (User Datagram Protocol) is a connectionless protocol that allows for fast and lightweight communication. It is commonly used for log message transmission due to its simplicity and low overhead. Log messages sent over UDP are not guaranteed to arrive at the destination, but they are typically sent in a fire-and-forget manner.

  > **WARNING**
  >
  > Messages will be sent in clear text, i.e. the messages will be readable by anyone who can intercept the messages.

- **TCP:** TCP (Transmission Control Protocol) is a reliable and connection-oriented protocol that ensures the delivery of log messages. It provides features such as error checking, flow control, and retransmission of lost packets. Log messages sent over TCP are guaranteed to arrive at the destination, but they may introduce additional latency compared to UDP.

  > **WARNING**
  >
  > Messages will be sent in clear text, i.e. the messages will be readable by anyone who can intercept the messages.

- **TLS:** TLS (Transport Layer Security) is a cryptographic protocol that provides secure communication over a network. It is used to encrypt log messages sent to remote destinations, ensuring the confidentiality and integrity of the data. TLS adds an extra layer of security to log message transmission, protecting against eavesdropping and tampering.

  > **NOTE**
  >
  > TLS will use TCP as the underlying protocol.

  > **TIP**
  >
  > Messages will be encrypted, i.e. the messages will **not** be readable by anyone who can intercept the messages.

  For more information on how certificates are used in the system, see the Certificate Management section.

In conclusion, the choice of transport protocol for log message transmission depends on the specific requirements of your system. UDP offers speed and simplicity, making it suitable for situations where high throughput is more important than guaranteed delivery. TCP, on the other hand, ensures that every log message reaches its destination, at the cost of additional latency. For secure communication, TLS is the go-to choice as it encrypts log messages, protecting them from eavesdropping and tampering. Remember, each protocol has its strengths and trade-offs, and the best choice depends on the balance between speed, reliability, and security in your specific use case.

Sources

A logging source is a component that specifies where the log messages originate from. There are different types of sources available:

- **Local:** Log messages generated by the system itself. Local log messages can be divided into two categories:
  - **Userspace:** Log messages generated by the system itself, including system services, applications, and other non-kernel related sources. These messages provide information about the behaviour and status of various processes running on the system. Userspace log messages are typically used for troubleshooting and monitoring purposes.
  - **Kernel:** Log messages generated by the kernel itself, which is the core component of the operating system. Kernel log messages provide information about internal operations of the kernel, such as hardware events, driver activity, and system errors. These messages are crucial for diagnosing and debugging issues at the system level.

  Both userspace and kernel log messages are important for understanding the overall health and performance of a system. By analysing these log messages, administrators and developers can identify issues, track system behaviour, and make informed decisions for system optimization and troubleshooting.
- **Remote:** Log messages received from remote machines. When a remote source is defined, the system listens for incoming log messages on the specified port. The following transport protocols are supported: UDP, TCP, and TLS.

  Configuring a remote source allows the system to receive log messages from external sources, enabling centralized logging and monitoring of multiple systems.

  However, since our devices are typically not used as log servers, a more common approach would be to use them as log-server relays, forwarding log messages to a dedicated log server. One advantage of this solution is that log messages sent over UDP to the relay can be forwarded to the log server over a more reliable TCP connection or a secure TLS connection.

Destinations

A logging destination is a component that specifies where the log messages are sent to. There are different types of destinations available:

- **File:** Used to send log messages to a local file on the device or to a connected external media device (e.g., USB or SD-Card). When the file destination is specified, a media type and a path relative to that media must be provided.

  > **NOTE**
  >
  > For the **internal** media, the path is relative to the `/log/` folder on the device.

  > **WARNING**
  >
  > Keep in mind that the **internal** media for logging, where the `/log/` folder is located, is stored on the internal RAM disk. This means that any log messages sent to a file on the **internal** media will be lost upon reboot. To retain log messages after a reboot, ensure to log them to a file on an external media device, or alternatively forward them to a remote destination.

  The media definitions are the same as those used in other parts of the system. For more information on how to configure media definitions, see the **Media Settings** part of the Boot Configuration.

When logging to a file, you can specify a log rotation policy by setting the maximum size of each log file, the number of log files to retain, and the number of compressed log files to retain.

- **Remote:** Used to send log messages to a remote machine. The following transport protocols are supported: UDP, TCP, and TLS.

  Configuring a remote destination allows the system to send log messages to an external log server, enabling centralized logging and monitoring of multiple systems. Remote destinations are commonly used to forward log messages to a dedicated log server for storage, analysis, and monitoring.

- **Console:** Used to display log messages directly on the attached console Log messages sent to the console are displayed in real-time, allowing administrators to monitor system activity and diagnose issues directly from the console.

> **WARNING**
>
> Keep in mind that when sending log messages to the console, it may be more difficult to operate the device using the CLI, as the log messages will be interspersed with the command output. It will **not** overwrite any written commands, but it will be displayed in between the command output, making it harder to read.

Each individual destination has a buffer that stores log messages until they are sent to the destination. For instance, if the device cannot reach the remote destination, the log messages will be stored in the buffer until the connection is reestablished.

This buffering mechanism applies not only to logging to remote destinations but also to logging to files on the device. However, it is particularly relevant when logging to local files stored on a connected external media device, rather than when logging to files on the internal RAM disk. This is because the IO operation to the external media device might be slower than the rate at which log messages are generated, especially in situations where a larger burst of log messages occurs. In such cases, the buffer stores the log messages until they can be written to the external media device.

> **NOTE**
>
> The buffer size is limited. If it becomes full, any new log messages will be dropped until there is space in the buffer again.

> **NOTE**
>
> To prevent the log buffer from using too much memory, the maximum number of log messages that can be buffered for each destination is divided among all the destinations in use. This means that if you have multiple destinations, the buffer size for each destination will be smaller than if you only have one destination.

Filters

A logging filter is an optional component that can be used to either include or exclude log messages based on specific criteria. Filters are used to define what type of messages are sent to the logging destination. There are a number of different filter types that can be utilised:

- **Facility:** Filter messages based on the facility of the log message. The facility indicates the type of system component that generated the log message, mentioned in an earlier section.
- **Severity:** Filter messages based on the severity of the log message. The severity indicates the level of criticality of the log message, mentioned in an earlier section.
- **Hostname:** Filter messages based on the hostname of the device that generated the log message. This is done by matching on the HOST field in the syslog message.

  > **TIP**
  >
  > Supports regular expressions for more advanced matching.

- **Message:** Filter messages based on the content of the log message.

  > **TIP**
  >
  > Supports regular expressions for more advanced matching.

- **Program:** Filter messages based on the program name that generated the log message. This is done by matching on the PROGRAM field in the syslog message.

  > **TIP**
  >
  > Supports regular expressions for more advanced matching.

- **Network:** Filter messages based on a provided IP subnet. If the sender of the log message is within the specified subnet, it will be a match.
- **Rate-limit:** Filter messages if they exceed a certain provided rate of log messages per second. This is useful for preventing log flooding attacks.

  It is also possible to perform the rate limit per a specific type of content that is part of the log message:
  - **Host:** The rate limit is performed per host that is sending log messages.
  - **Source IP:** The rate limit is performed per source IP address that is sending log messages.
  - **Facility:** The rate limit is performed per facility that is sending log messages.
  - **Severity:** The rate limit is performed per severity level that is sending log messages.

> **TIP**
>
> It is possible to invert a filter by using the setting it to `invert`. Meaning that the filter will allow all messages that do not match the filter criteria. As an example, if we have a filter that specify severity `debug` and we set it to `inverted`, it will allow all messages that are not of severity `debug`.

Each individual filter can combine multiple of these filter types for the same filter definition. However, if we want to combine multiple filters that, for instance, use the same filter type, we need to create multiple filter definitions and combine them using the `and` or `or` operators, when we assign them to a sink.

Sinks

When all the other components have been defined, a sink object can be created. Configuration of a sink is very straightforward, it is simply specifying the source(s) and destination(s) to use, and optionally the filter(s) to use.

However, when specifying multiple filters, their relation to each other must be taken into account. The filters can be combined using the `and` or `or` operators. Each subsequent filter will be combined with the previous filter using the specified operator. As an example if we have four filters, `filter-1`, `filter-2`, `filter-3`, and `filter-4`, and we combine them in the following way:

```
example:/config/logging/sink-1/#> filter filter-1 and filter-2 or filter-3 and filter-4
```

This means that the sink will allow messages that match `filter-1` and `filter-2` or messages that match `filter-3` and `filter-4`.

When the sink has been finalized and its configuration activated, it will start to send log messages from the source(s) to the destination(s) based on the filter(s) specified.

Configuration

The logging in the system can be configured from the top-level configuration context in the CLI:

```
example:/#> configure
example:/config/#> logging
example:/config/logging/#>
```

We will separate these sections into the following sub-sections:

- General Logging Configuration
- Source Configuration
- Destination Configuration
- Filter Configuration
- Sink Configuration

General Logging Configuration

The general logging settings are configured directly under the `logging` context:

```
example:/config/logging/#>
```

`[no] console`

Description

Use to enable or disable logging directly on the console.

Default: Disabled

Example

```
example:/config/logging/#> console
example:/config/logging/#> no console
```

continues on next page

continued

`[no] console`

Options

Option	Description
`no`	Disable logging to the console.

`[no] external-logging [SIZE [COUNT]]`

Description

Enable or disable logging of default log files external media, e.g. USB/SD-Card.

When enabled the same log files that are logged to the local RAM disk will also be logged to the specified external media.

A folder called /log/ will be created on the external media if it does not exist, this is where the log files will be created.

All log files are automatically compressed from generation .1

> **NOTE**
>
> Sinks can be configured to log to a file on external media, regardless if this setting is enabled or not. This setting only handles logging of the systems default log files to the specified external media.

Default: Disabled

Example

Enable external logging and Keep 10 log files, each 10MiB (uncompressed):

```
example:/config/logging/#> external-logging 10M 10
```

Options

Option	Description
`no`	Disable logging of default log files to usb.
`SIZE`	Max size of each log file, in uncompressed form. Takes an optional k/M size modifier.
`COUNT`	Number of log files to keep, in total.

`[no] external-logging-media [MEDIA]`

Description

Specify the target external media (USB/SD-Card) for the logging.

The selectable medias are based on the configured medias in the boot configuration context. Any media name can be configured, but a warning will be displayed if it is not also configured in the boot configuration.

continues on next page

continued

`[no] external-logging-media [MEDIA]`

> **TIP**
>
> Refer to this page for help on how to correctly set up access to external media devices.

Default: external

Example

```
example:/config/logging/#> external-logging-media my-other-media
```

Options

Option	Description
`no`	Reset the option to default, the default **`external`** media.
`MEDIA`	Dynamically created using the **`media`** command in the boot configuration context. Use the **`do show media`** command in the current context to list all existing media definitions. All usable media definitions will be listed under *Active*. In addition, two default media definitions exist if no changes have been made to medias in the boot context:W • **`internal:`** The representation of the built-in flash. This cannot be changed or removed. • **`external:`** The default media definition for any connected external media device (e.g. USB, SD-Card). It is configured to always match the first partition on said external media device. This definition is not static or locked, it can be freely adjusted by the user, even removed.

`[no] secure-mode`

Description

Enable or disable secure mode. This setting determines whether the unit reject or accept any logging messages from remote machines. By default secure mode is enabled, meaning the unit operates in a secure way and do not accept any messages from remote machines.

Default: Enabled

Example

```
example:/config/logging/#> secure-mode
example:/config/logging/#> no secure-mode
```

Options

Option	Description
`no`	Disable secure mode.

`identification [ hostname | fqdn ]`

Description

Choose how the device will identify itself in syslog messages.

There are two options:

- `hostname` will use the system hostname, while ignoring any configured or received domain name.
- `fqdn` will use the fully qualified domain name, which is a combination of the hostname and the domain name. The domain name is first read from the "domain" command in the "system" configuration context; if unset, it will instead be read from any DHCP leases where the domain name is provided via option 15. If still unavailable, this option will fall back to just using the hostname.

The default is to use the hostname only.

Source Configuration

Each source configuration has its sub-context under the `logging` context:

```
example:/config/logging/#> source example
example:/config/logging/source-example/#>
```

`[no] name NAME`

Description

Change the name of the source. The name is a free form string that identifies the source. The name is used to select the source when configuring a sink.

> **NOTE**
>
> If the name is changed it will also be changed in all sinks that uses this source, so it is not necessary to update all sinks that uses this source.

Example

```
example:/config/logging/source-example/#> name my-source
```

Options

Option	Description
`no`	Does nothing for this command, the source must have a name.
`NAME`	The free form name of the source. It can be up to 15 characters long. It allows for the following characters: `a-z`, `A-Z`, `0-9`, `-`, and `_`.

`[no] type local|remote`

Description

Specify the type of the source.

continues on next page

continued

`[no] type local|remote`

A source can be either local, i.e the device itself, or some other remote device capable of sending syslog messages.

> **NOTE**
>
> If the source is remote it is important to also ensure that the 'secure-mode' option, under the global logging configuration context, is set to disable. While 'secure-mode' is enabled the device will not allow any remote log messages to be received by the device (This option is by default enabled).

Example

```
example:/config/logging/source-example/#> type local
example:/config/logging/source-example/#> type remote
```

Options

Option	Description
`no`	Does nothing for this command, the source must have a type.
`local`	The source is the device itself. **NOTE** When this type is selected, additional settings will be available to configure the source. These settings are specific to the local source type. See the Source Local Specific.
`remote`	The source is a remote device. **NOTE** When this type is selected, additional settings will be available to configure the source. These settings are specific to the remote source type. See the Source Remote Specific.

Source Local Specific

When the source type is set to `local`, additional settings are available to configure the source. These settings are specific to the local source type.

```
example:/config/logging/source-example/#> type local
```

`[no] userspace`

Description

Enable or disable local userspace log messages.

continues on next page

continued

`[no] userspace`

When enabled, the source will collect log messages from userspace processes on the local machine. This includes messages from system daemons, user applications, and other non-kernel processes. When disabled, these messages will not be collected.

Default: Enabled

Example

```
example:/config/logging/source-example/#> userspace
example:/config/logging/source-example/#> no userspace
```

Options

Option	Description
`no`	Disable logging of userspace log messages.

`[no] kernel`

Description

Enable or disable local kernel log messages.

When enabled, the source will collect log messages from the local machine's kernel. This includes messages related to the operation of the kernel, hardware drivers, and system services. When disabled, these messages will not be collected.

Default: Enabled

Example

```
example:/config/logging/source-example/#> kernel
example:/config/logging/source-example/#> no kernel
```

Options

Option	Description
`no`	Disable logging of kernel log messages.

`[no] audit`

Description

Enable or disable local audit log messages.

When enabled, the source will collect log messages from the local machine's audit system. This includes messages related to the operation of the audit system, security events, and other audit-related messages. When disabled, these messages will not be collected.

continues on next page

continued

`[no] audit`

> **NOTE**
>
> Some of these audit log messages may provide similar information as the regular local userspace and kernel log messages, but they will have a specific format and be associated with unique Event IDs.

Default: Disabled

Example

```
example:/config/logging/source-example/#> audit
example:/config/logging/source-example/#> no audit
```

Options

Option	Description
`no`	Disable logging of audit log messages.

Source Remote Specific

When the source type is set to `remote`, additional settings are available to configure the source. These settings are specific to the remote source type.

```
example:/config/logging/source-example/#> type remote
```

`[no] format rfc3164|rfc5424`

Description

Set the syslog formatting to expect from remote clients.

Default: rfc5424

Example

```
example:/config/logging/source-example/#> format rfc3164
example:/config/logging/source-example/#> format rfc5424
```

Options

Option	Description
`no`	Does nothing for this command, the source must have a format.
`rfc3164`	Use the RFC 3164[1] syslog format. This the traditional syslog format that includes a timestamp, hostname, process name, process ID, and the message. Example of a message in RFC 3164 format: `<PRI>Nov 18 09:36:42 hostname proc[123] message`

continues on next page

[1] https://tools.ietf.org/html/rfc3164

continued

`[no] format rfc3164|rfc5424`

Option	Description
`rfc5424`	Use the RFC 5424[2] syslog format. This is a newer syslog format that includes structured data and additional fields. Example of a message in RFC 5424 format: `<PRI>1 2019-11-18T09:36:42.000321+01:00 hostname proc 123 - - message`

`[no] transport udp|tcp|tls`

Description

Specify the transport protocol to be used for the source.

This option allows for specification of the transport protocol to be used for the source. Any client that wants to send messages to the device must use the same transport protocol.

Default: TLS

Example

```
example:/config/logging/source-example/#> transport udp
example:/config/logging/source-example/#> transport tcp
example:/config/logging/source-example/#> transport tls
```

Options

Option	Description
`no`	Does nothing for this command, the source must have a transport.
`udp`	Use the User Datagram Protocol. This is a connectionless protocol that does not guarantee delivery of packets.
`tcp`	Use the Transmission Control Protocol. This is a connection oriented protocol that guarantees delivery of packets.
`tls`	Use the Transport Layer Security protocol. This is a secure protocol that provides encryption and authentication. **NOTE** When this type is selected, additional settings will be available to configure the source. These settings are specific to the TLS transport protocol. See the Source Remote TLS Specific.

[2] https://tools.ietf.org/html/rfc5424

`[no] ip-protocol ipv4|ipv6`

Description

Specify the IP protocol to be used for the source.

This option allows for specification of the IP protocol to be used for the source. Any client that wants to send messages to the device must use the same IP protocol.

Default: IPv4

Example

```
example:/config/logging/source-example/#> ip-protocol ipv4
example:/config/logging/source-example/#> ip-protocol ipv6
```

Options

Option	Description
`no`	Does nothing for this command, the source must have an IP protocol.
`ipv4`	Expect messages to be sent using the IPv4 protocol.
`ipv6`	Expect messages to be sent using the IPv6 protocol.

`[no] listen [IFNAME]`

Description

Specify a specific interface, with static address, to listen on.

This option allows for providing a specific IP address to listen on. By default every available interface will be listened on, i.e. we bind to the address 0.0.0.0.

If a specific interface should be listened on, provide the interface name as the argument to this option.

> **NOTE**
>
> The interface must have a static IP address configured, otherwise the device will not be able to bind to the address. If the interface has more than one address, only the first address will be used.

> **TIP**
>
> If you want to receive only messages from a specific remote host, based on its IP address, configure a filter for to handle that instead.

Default: Listen on all available interfaces.

Example

```
example:/config/logging/source-example/#> listen vlan1
example:/config/logging/source-example/#> no listen
```

continues on next page

continued

`[no] listen [IFNAME]`

Options

Option	Description
`no`	Listen on all available interfaces.
`IFNAME`	The name of the interface to listen on. Only one specific interface can be provided.

`[no] port [PORT]`

Description

Specify another port to use, other than the default.

This configuration option allows for specification of a non-default source port for the remotely received messages.

Default: Depends on the transport protocol. UDP: 514, TCP: 601, TLS: 6514,

Example

```
example:/config/logging/source-example/#> port 5514
```

Options

Option	Description
`no`	Reset the port to the default value.
`PORT`	The port number to use for the source. The port number must be in the range 1-65535.

`[no] keep-hostname`

Description

Enable or disable hostname rewriting.

If this setting is enabled, the hostname specified in the HOST field of the incoming log messages for the source will be preserved.

For instance, this can be useful if the original sender's hostname should be preserved when acting as a syslog relay.

If disabled, the HOST field of the message will be rewritten to indicate that it was sent from this device, if the log source is used when sending to another remote destination.

> **NOTE**
>
> If the received log message contains no hostname, a hostname will be added to the message. For a message that has been received from a remote source, the hostname will be the address of the sender.

Default: Disabled

continues on next page

continued

`[no] keep-hostname`

Example

```
example:/config/logging/source-example/#> keep-hostname
example:/config/logging/source-example/#> no keep-hostname
```

Options

Option	Description
`no`	Disable hostname rewriting.

`[no] use-dns [local-only]`

Description

Enable or disable DNS usage for hostname resolving.

Handling of syslog messages will block on DNS queries. Therefore, if this setting is enabled, a DoS scenario could potentially occur. To prevent this, or reduce the risk, ensure that relevant firewall rules are configured. Furthermore, it's important to ensure that any host that may send messages and can reach this source is resolvable.

If the `local-only` option is specified, hostname resolution attempts will be performed locally only for statically configured hosts. These can be configured under the 'dns' context using the 'host' command.

> **NOTE**
>
> This setting has no effect if the keep-hostname setting is enabled, and an incoming message contains a hostname in its HOST field.

Default: Enabled

Example

```
example:/config/logging/source-example/#> use-dns
example:/config/logging/source-example/#> use-dns local-only
example:/config/logging/source-example/#> no use-dns
```

Options

Option	Description
`no`	Disable DNS usage for hostname resolving.
`local-only`	Only use locally configured DNS entries for hostname resolving. See the DNS Configuration section for more information on how to configure DNS entries.

`[no] use-local-timestamp`

Description

Enable or disable usage of local timestamp for received messages.

If this setting is enabled, the timestamp of the incoming log messages for the remote source will be replaced with the device's local timestamp, i.e. the time of reception and handling. If disabled, the original timestamp included in the message will be preserved, i.e. the time the message was originally generated based on the sender's time.

Default: Disabled

Example

```
example:/config/logging/source-example/#> use-local-timestamp
example:/config/logging/source-example/#> no use-local-timestamp
```

Options

Option	Description
`no`	Disable usage of local timestamp for received messages.

Source Remote TLS Specific

When the source type is set to `remote` and the transport protocol is set to `tls`, additional settings are available to configure the source. These settings are specific to the remote source type and the TLS transport protocol.

```
example:/config/logging/source-example/#> type remote
example:/config/logging/source-example/#> transport tls
```

`[no] peer-verify [untrusted|trusted]`

Description

Select the verification method to be used during TLS handshake.

The peer-verify option can be used to specify the level of verification that should be performed during the TLS handshake.

The value is a combination of two properties:

- First we specify whether the peer is required to provide a certificate or if it is optional:
  - **Required:** When `peer-verify` is specified.
  - **Optional:** When `no peer-verify` is specified.
- Second we specify whether the certificate provided needs to be valid (Trusted) or not (Untrusted):
  - **Trusted:** When the `trusted` option is specified.
  - **Untrusted:** When `untrusted` option is specified.

Based on the combination of these two properties, the following combinations are possible:

- **Optional Untrusted:** The peer is not required to provide a certificate, but if it does, it does not need to be verified.

continues on next page

continued

`[no] peer-verify [untrusted|trusted]`

- **Optional Trusted:** The peer is not required to provide a certificate, but if it does, it needs to be verified.
- **Required Untrusted:** The peer is required to provide a certificate, but it does not need to be verified.
- **Required Trusted:** The peer is required to provide a certificate, and it needs to be verified.

For all the scenarios where `Trusted` is specified, a CA certificate needs to be provided as well, in order to verify the certificate provided by the peer.

> **NOTE**
>
> If only `peer-verify` is specified, it defaults to using `trusted`.
>
> If only `no peer-verify` is specified, it defaults to using `untrusted`.
>
> These two commands are just shorthand commands for the most strict and least strict verification methods.

Default: Required Trusted, i.e. `peer-verify trusted`.

Example

```
example:/config/logging/source-example/#> peer-verify untrusted
example:/config/logging/source-example/#> peer-verify trusted
example:/config/logging/source-example/#> peer-verify
example:/config/logging/source-example/#> no peer-verify untrusted
example:/config/logging/source-example/#> no peer-verify trusted
example:/config/logging/source-example/#> no peer-verify
```

Options

Option	Description
`no`	Make the peer verification optional during the TLS handshake, i.e. the peer is not required to provide a certificate.
`untrusted`	If a certificate is provided, it does not need to be verified.
`trusted`	If a certificate is provided, it needs to be verified.

`[no] ignore-hostname-mismatch`

Description

Enable to ignore certificate hostname mismatch during validation.

If this option is enabled the subject name of the certificate will be ignored during the validation of the certificate in the TLS handshake, i.e. any mismatch between the hostname and the subject name will be ignored.

Default: No

continues on next page

continued

`[no] ignore-hostname-mismatch`

Example

```
example:/config/logging/source-example/#> ignore-hostname-mismatch
example:/config/logging/source-example/#> no ignore-hostname-mismatch
```

Options

Option	Description
`no`	Disable ignoring of certificate hostname mismatch during validation.

`[no] certificate [NAME]`

Description

Specify the Certificate to use.

The selected certificate will be used for authenticating the source against any client that connects to it.

> **NOTE**
>
> The certificate must be imported into the system before it can be used. For more information on how to import certificates, see the Certificate Management section.

Default: None

Example

```
example:/config/logging/source-example/#> certificate my-cert
```

Options

Option	Description
`no`	Remove the certificate from the source.
`NAME`	The name of the certificate to use.

`[no] ca-certificate [NAME]`

Description

Specify the CA Certificate to use.

The selected CA certificate will be used for verifying any certificate that is provided by clients that connect to the source.

continues on next page

continued

`[no] ca-certificate [NAME]`

> **NOTE**
>
> The CA certificate must be imported into the system before it can be used. For more information on how to import certificates, see the Certificate Management section.

> **NOTE**
>
> If the `peer-verify` option is set to `trusted`, a CA certificate must be provided.

Default: None

Example

```
example:/config/logging/source-example/#> ca-certificate my-ca-cert
```

Options

Option	Description
`no`	Remove the CA certificate from the source.
`NAME`	The name of the CA certificate to use.

`[no] revocation`

Description

Enable or disable certificate revocation checking during the TLS handshake. **Default:** Disabled.

If enabled, the system will reject connections if the cached Certificate Revocation List (CRL) is outdated, fails signature verification, or if the certificate is explicitly listed as revoked in the CRL. **Note:** If a CRL is missing, the connection is still accepted.

CRLs must be configured in the `config/pki` context.

Options

Option	Description
`no`	Disable certificate revocation checking.

For details on configuring static CRL distribution points, see the Revocation Configuration Guide.

Destination Configuration

Each destination configuration has its sub-context under the `logging` context:

```
example:/config/logging/#> destination example
example:/config/logging/destination-example/#>
```

`[no] name NAME`

Description

Change the name of the destination. The name is a free form string that identifies the destination. The name is used to select the destination when configuring a sink.

> **NOTE**
>
> If the name is changed it will also be changed in all sinks that uses this destination, so it is not necessary to update all sinks that uses this destination.

Example

```
example:/config/logging/destination-example/#> name my-destination
```

Options

Option	Description
`no`	Does nothing for this command, the destination must have a name.
`NAME`	The free form name of the destination. It can be up to 15 characters long. It allows for the following characters: `a-z`, `A-Z`, `0-9`, `-`, and `_`.

`[no] type file|remote|console`

Description

Specify the type of the destination.

A destination can be either a file, a remote server, or the console.

Example

```
example:/config/logging/destination-example/#> type file
example:/config/logging/destination-example/#> type remote
example:/config/logging/destination-example/#> type console
```

Options

Option	Description
`no`	Does nothing for this command, the destination must have a type.
`file`	The destination is a file on the device or external media. **NOTE** When this type is selected, additional settings will be available to configure the destination. These settings are specific to the file destination type. See the Destination File Specific.
`remote`	The destination is a remote server.

continues on next page

continued

`[no] type file|remote|console`

Option	Description
	NOTE When this type is selected, additional settings will be available to configure the destination. These settings are specific to the remote destination type. See the Destination Remote Specific.
`console`	The destination is the console. **NOTE** When this type is selected, no additional settings are available to configure the destination. **WARNING** Keep in mind that when sending log messages to the console, it may be more difficult to operate the device using the CLI, as the log messages will be interspersed with the command output. It will **not** overwrite any written commands, but it will be displayed in between the command output, making it harder to read.

Destination File Specific

When the destination type is set to `file`, additional settings are available to configure the destination. These settings are specific to the file destination type.

```
example:/config/logging/destination-example/#> type file
```

`[no] media [internal|MEDIA]`

Description

Specifies the target media where the log file will be stored.

The file name/path, set with the `file-name` command, will be relative to the root of the specified media.

> **TIP**
> Refer to this page for help on how to correctly set up access to external media devices.

> **TIP**
> Tab completion is available for the `MEDIA` argument, to list all existing media definitions.

continues on next page

continued

`[no] media [internal|MEDIA]`

Default: None

Example

```
example:/config/logging/destination-example/#> media internal
example:/config/logging/destination-example/#> media external
example:/config/logging/destination-example/#> media my-other-media
```

Options

Option	Description
`no`	Removes the configured media from the destination. However, if another media is not specified, the destination configuration will be invalid.
`internal`	Sets the destination to the internal memory of the device. This is a special media definition that will always exist. When used in the context of a logging media, it refers to the **/log/** directory on the internal RAM disk. Therefore, specified `file-name` will be relative to this directory. **WARNING** As mentioned above, the internal log directory is located on the internal RAM disk. This means that the log files will be lost upon reboot or power loss, because the RAM disk is not persistent.
`MEDIA`	Specifies the name of the external media to use. Any log file, set with the `file-name` command, will be relative to the root of the specified media, e.g. **/media/external/**. The media must be defined in the boot configuration context, see Boot Configuration for more information. The reason for this is to give the user full control over which media devices should be made available on the system in general. **NOTE** By default, the media definition **external** is available for use. This definition is configured to always match the first partition on any connected external media device (e.g. USB, SD-Card). This definition is not static or locked, it can be freely adjusted by the user, even removed. **TIP** Use the `do show media` command in the current context to list all existing media definitions. All usable media definitions will be listed under *Active*.

`[no] file-name [PATH]`

Description

Specify the name or the path of the log file to be used.

The provided `PATH` will be relative to the root of the specified media, set with the `media` command.

> **WARNING**
>
> If a path is specified, ensure that all directories in the path exist. Failure to do so will result in the log file not being created.

Default: None

Example

```
example:/config/logging/destination-example/#> file-name my-log-file.log
example:/config/logging/destination-example/#> file-name /my-folder/my-log-file.log
```

Options

Option	Description
`no`	Removes the configured file name from the destination. However, if another file name is not specified, the destination configuration will be invalid.
`PATH`	The name or path of the log file to use. The file name/path can be up to 127 characters long. The file name part of the `PATH` can include any character, except for the null character `\0` and the slash `/`. The slash is used as a directory separator. In addition, the file name should not include special characters such as `?`, `*`, `|`, `<`, `>`, `&`, `;`, `(`, `)`, or newline. The path can include multiple directories, which are separated by slashes `/`. Each directory in the path must already exist; otherwise, the log file will not be created. The path is relative to the root of the specified media, set with the `media` command.

`[no] rotate [0-100]`

Description

Specify the maximum number of log files to keep.

When the file is rotated, a new log file will be created with the same name, but with a sequence number appended to it. The sequence number will start at 0 and increase for each new log file created. The oldest log file will have the highest sequence number. As an example, if we allow for 5 rotated log files, the files will be named as follows:

continues on next page

continued

`[no] rotate [0-100]`

```
my-log-file.log           -- The current log file
my-log-file.log.0         -- The fifth rotated log file
my-log-file.log.1         -- The fourth rotated log file
my-log-file.log.2         -- The third rotated log file
my-log-file.log.3         -- The second rotated log file
my-log-file.log.4         -- The first rotated log file
```

When the maximum number of log files is reached, the oldest log file will be deleted to make room for the new log file. If `compress` is enabled, the oldest compressed log file will be deleted instead and the oldest log file will be compressed.

Each file is rotated when it reaches a certain size, specified with the `size` command.

> **WARNING**
>
> If log file rotation is disabled, the log file will grow indefinitely. This can lead to the log file consuming all available space on the media. Therefore, if it is disabled, ensure that the log file does not grow too large.

Default: 1 file.

Example

```
example:/config/logging/destination-example/#> rotate 5
example:/config/logging/destination-example/#> no rotate
```

Options

Option	Description
`no`	Use to disable log file rotation.
`0-100`	The maximum number of log files to keep. The value must be in the range 0-100. Setting the rotate value to 0 will disable log file rotation. This is the same as using the `no rotate` command.

`[no] size 1-1000[k|K|M|G|KB|MB|GB]`

Description

Specify the size that triggers log file rotation when reached.

When the log file reaches the specified size, it will be rotated. The number of rotations is specified with the `rotate` command.

Default: 100KB

Example

```
example:/config/logging/destination-example/#> size 100KB
example:/config/logging/destination-example/#> size 10MB
example:/config/logging/destination-example/#> size 1GB
```

continues on next page

continued

`[no] size 1-1000[k|K|M|G|KB|MB|GB]`

Options

Option	Description
`no`	This option has no effect for this command. If `rotate` is enabled, the size must be set. If `rotate` is disabled, the specified size will not be used.
`1-1000[k|K|M|G|KB|MB|GB]`	Specifies the size in bytes, which must be in the range of 1-1000. The size is specified with a size modifier, such as `KB`, `MB`, or `GB`, to indicate kilobytes, megabytes, or gigabytes, which results in the following sizes: • `KB` or `k`: Kilobytes (1024 bytes). • `MB` or `M`: Megabytes (1,048,576 bytes). • `GB` or `G`: Gigabytes (1,073,741,824 bytes).

`[no] compress [0-100]`

Description

Specify the maximum number of compressed log files to keep.

When the number of rotated log files reaches the maximum number of rotations, as specified with the `rotate` command, the oldest log file will be compressed, if this option is enabled. This command specifies the maximum number of such compressed log files to keep. When the maximum number of compressed log files is reached, the oldest compressed log file will be deleted to make room for the new compressed log file.

The compressed log file will have the same name as the original log file, with the sequence number appended to it, starting from the sequence number of the last rotated log file. In addition, the compressed log file will have the `.gz` extension. The highest sequence number will be the oldest.

As an example, if we allow for 3 rotated files and 3 compressed files, the files will be named as follows:

```
my-log-file.log           -- The current log file
my-log-file.log.0         -- The third rotated log file
my-log-file.log.1         -- The second rotated log file
my-log-file.log.2         -- The first rotated log file
my-log-file.log.3.gz      -- The third compressed log file
my-log-file.log.4.gz      -- The second compressed log file
my-log-file.log.5.gz      -- The first compressed log file
```

Default: 2 compressed files.

Example

```
example:/config/logging/destination-example/#> compress 3
example:/config/logging/destination-example/#> no compress
```

continues on next page

continued

`[no] compress [0-100]`

Options

Option	Description
`no`	Disables log file compression.
`0-100`	The maximum number of compressed log files to keep. The value must be in the range 0-100. If the value is set to 0, log file compression will be disabled, which is the same as using the `no compress` command.

Destination Remote Specific

When the destination type is set to `remote`, additional settings are available to configure the destination. These settings are specific to the remote destination type.

```
example:/config/logging/destination-example/#> type remote
```

`[no] format bsd|rfc3164|rfc5424`

Description

Set the syslog formatting to use when sending messages to the remote server.

Default: rfc5424

Example

```
example:/config/logging/destination-example/#> format bsd
example:/config/logging/destination-example/#> format rfc3164
example:/config/logging/destination-example/#> format rfc5424
```

Options

Option	Description
`no`	Does nothing for this command, the destination must have a format.
`bsd`	Use the legacy BSD format. This simple format includes a the process name, process ID, and the message. Example of a message in BSD format: `proc[123]: message.`
`rfc3164`	Use the RFC 3164[1] syslog format. This the traditional syslog format that includes a timestamp, hostname, process name, process ID, and the message. Example of a message in RFC 3164 format: `<PRI>Nov 18 09:36:42 hostname proc[123] message`
`rfc5424`	Use the RFC 5424[2] syslog format. This is the newest syslog format that includes structured data and additional fields.

continues on next page

[1] https://tools.ietf.org/html/rfc3164
[2] https://tools.ietf.org/html/rfc5424

continued

`[no] format bsd|rfc3164|rfc5424`

Option	Description
	Example of a message in RFC 5424 format: `<PRI>1 2019-11-18T09:36:42.000321+01:00 hostname proc 123 - - message`

`[no] transport [udp|tcp|tls]`

Description

Specify the transport protocol to be used for the destination.

This option allows for specification of the transport protocol to be used for the destination. Any server that wants to receive messages from the device must be configured to use the same transport protocol.

Default: TLS

Example

```
example:/config/logging/destination-example/#> transport udp
example:/config/logging/destination-example/#> transport tcp
example:/config/logging/destination-example/#> transport tls
```

Options

Option	Description
`no`	Does nothing for this command, the destination must have a transport.
`udp`	Use the User Datagram Protocol. This is a connectionless protocol that does not guarantee delivery of packets.
`tcp`	Use the Transmission Control Protocol. This is a connection oriented protocol that guarantees delivery of packets.
`tls`	Use the Transport Layer Security protocol. This is a secure protocol that provides encryption and authentication. **NOTE** When this type is selected, additional settings will be available to configure the destination. These settings are specific to the TLS transport protocol. See the Destination Remote TLS Specific.

`[no] ip-protocol ipv4|ipv6`

Description

Specify the IP protocol to be used for the destination.

This option allows for specification of the IP protocol to be used for the destination. Any server that wants to receive messages from the device must use the same IP protocol.

continues on next page

continued

`[no] ip-protocol ipv4|ipv6`

Default: IPv4

Example

```
example:/config/logging/destination-example/#> ip-protocol ipv4
example:/config/logging/destination-example/#> ip-protocol ipv6
```

Options

Option	Description
`no`	Does nothing for this command, the destination must have an IP protocol.
`ipv4`	Transmit messages using the IPv4 protocol.
`ipv6`	Transmit messages using the IPv6 protocol.

`[no] port [PORT]`

Description

Specify a port to use, different from the default one.

This configuration option allows you to specify a non-default destination port for messages sent remotely.

Default: Depends on the transport protocol. UDP: 514, TCP: 601, TLS: 6514,

Example

```
example:/config/logging/destination-example/#> port 5514
```

Options

Option	Description
`no`	This option resets the port to its default value based on the transport protocol.
`PORT`	Specifies the port number to use for the destination. The port number must be within the range of 1-65535.

`[no] host [ [address IPADDR] | [name HOSTNAME|FQDN] | [dhcp IFNAME] ]`

Description

Defines the location of the host for the remote destination.

The host location can be specified in one of the following ways:

- By providing an IPv4 or IPv6 address using the `address` keyword followed by the `IPADDR`. The expected address type is dependent on the IP protocol set for the destination, see the `ip-protocol` command.
- By providing a hostname or a Fully Qualified Domain Name (FQDN) using the `name` keyword followed by the `HOSTNAME` or `FQDN`.

continues on next page

continued

`[no] host [ [address IPADDR] | [name HOSTNAME|FQDN] | [dhcp IFNAME] ]`

- By obtaining the destination address over DHCP using the `dhcp` keyword followed by the `IFNAME`. If a lease received on the specified interface contains option 7, it will be used as the destination address.

Default: None

Example

```
example:/config/logging/destination-example/#> host address 192.168.1.1
example:/config/logging/destination-example/#> host address 2001:db8::1
example:/config/logging/destination-example/#> host name example.com
example:/config/logging/destination-example/#> host name mydevice
example:/config/logging/destination-example/#> host dhcp vlan1
```

Options

Option	Description
`no`	Removes any configured host from the destination. However, if another host is not specified, the destination configuration will be invalid.
`address IPADDR`	Specifies the IPv4 or IPv6 address of the host.
`name HOSTNAME|FQDN`	Specifies the hostname or FQDN of the host.
`dhcp IFNAME`	Specifies the interface name to obtain the destination address over DHCP. If a lease received on the specified interface contains option 7, it will be used as the destination address. **NOTE** It's crucial to ensure that the specified interface has DHCP enabled and can successfully receive a lease from a DHCP server. For detailed instructions on how to configure DHCP on an interface, please refer to the Interface Configuration section. **NOTE** Each destination instance can apply up to two IP addresses, provided by option 7 in the DHCP lease.

`[no] throttle [VALUE]`

Description

Specify a throttle value to rate limit the number of messages sent per second

By setting a throttle value, you can limit the number of messages sent per second to the configured destination. This is particularly useful when the destination has a maximum message handling capacity.

continues on next page

continued

`[no] throttle [VALUE]`

> **NOTE**
>
> It's important to understand how messages are managed when the transmission rate is exceeded.
>
> If the rate of incoming messages surpasses the configured limit, the system doesn't instantly discard the excess messages. Instead, these messages are buffered and transmitted as soon as the rate limit allows.
>
> However, bear in mind that the buffer has a limited capacity. If the buffer reaches its maximum capacity due to a continuous influx of messages exceeding the rate limit, any incoming messages will be discarded until there is available space in the buffer.

> **TIP**
>
> If you prefer to drop messages when the rate limit is exceeded, you can use a `filter` of type `rate-limit`. This filter will instantly discard any messages that surpass the specified rate limit.
>
> For more information on how to configure a rate-limit filter, see the Filter Configuration section.

Default: `no throttle`, i.e. no rate limiting.

Example

```
example:/config/logging/destination-example/#> throttle 100
```

Options

Option	Description
`no`	This option disables throttling, allowing messages to be sent at the maximum possible rate.
`VALUE`	This option allows you to specify the maximum number of messages that can be sent per second to the destination. The value must be a non-negative integer.

Destination Remote TLS Specific

When the destination type is set to `remote` and the transport protocol is set to `tls`, additional settings are available to configure the destination. These settings are specific to the remote destination type and the TLS transport protocol.

```
example:/config/logging/destination-example/#> type remote
example:/config/logging/destination-example/#> transport tls
```

`[no] peer-verify [untrusted|trusted]`

Description

Select the verification method to be used during TLS handshake.

The peer-verify option can be used to specify the level of verification that should be performed during the TLS handshake.

The value is a combination of two properties:

- First we specify whether the peer is required to provide a certificate or if it is optional:
  - **Required:** When `peer-verify` is specified.
  - **Optional:** When `no peer-verify` is specified.
- Second we specify whether the certificate provided needs to be valid (Trusted) or not (Untrusted):
  - **Trusted:** When the `trusted` option is specified.
  - **Untrusted:** When `untrusted` option is specified.

Based on the combination of these two properties, the following combinations are possible:

- **Optional Untrusted:** The peer is not required to provide a certificate, but if it does, it does not need to be verified.
- **Optional Trusted:** The peer is not required to provide a certificate, but if it does, it needs to be verified.
- **Required Untrusted:** The peer is required to provide a certificate, but it does not need to be verified.
- **Required Trusted:** The peer is required to provide a certificate, and it needs to be verified.

For all the scenarios where `Trusted` is specified, a CA certificate needs to be provided as well, in order to verify the certificate provided by the peer.

> **NOTE**
>
> If only `peer-verify` is specified, it defaults to using `trusted`.
>
> If only `no peer-verify` is specified, it defaults to using `untrusted`.
>
> These two commands are just shorthand commands for the most strict and least strict verification methods.

Default: Required Trusted, i.e. `peer-verify trusted`.

Example

```
example:/config/logging/destination-example/#> peer-verify untrusted
example:/config/logging/destination-example/#> peer-verify trusted
example:/config/logging/destination-example/#> peer-verify
example:/config/logging/destination-example/#> no peer-verify untrusted
example:/config/logging/destination-example/#> no peer-verify trusted
example:/config/logging/destination-example/#> no peer-verify
```

continues on next page

continued

`[no] peer-verify [untrusted|trusted]`

Options

Option	Description
`no`	Make the peer verification optional during the TLS handshake, i.e. the peer is not required to provide a certificate.
`untrusted`	If a certificate is provided, it does not need to be verified.
`trusted`	If a certificate is provided, it needs to be verified.

`[no] ignore-hostname-mismatch`

Description

Enable to ignore certificate hostname mismatch during validation.

If this option is enabled the subject name of the certificate will be ignored during the validation of the certificate in the TLS handshake, i.e. any mismatch between the hostname and the subject name will be ignored.

Default: No

Example

```
example:/config/logging/destination-example/#> ignore-hostname-mismatch
example:/config/logging/destination-example/#> no ignore-hostname-mismatch
```

Options

Option	Description
`no`	Disable ignoring of certificate hostname mismatch during validation.

`[no] certificate [NAME]`

Description

Specify the Certificate to use.

The selected certificate will be used for authenticating the destination against the remote server that it connects to.

> **NOTE**
>
> The certificate must be imported into the system before it can be used. For more information on how to import certificates, see the Certificate Management section.

Default: None

Example

```
example:/config/logging/destination-example/#> certificate my-cert
```

continues on next page

continued

`[no] certificate [NAME]`

Options

Option	Description
`no`	Remove the certificate from the destination.
`NAME`	The name of the certificate to use.

`[no] ca-certificate [NAME]`

Description

Specify the CA Certificate to use.

The selected CA certificate will be used for verifying the certificate that is provided by the remote server that the destination connects to.

> **NOTE**
>
> The CA certificate must be imported into the system before it can be used. For more information on how to import certificates, see the Certificate Management section.

> **NOTE**
>
> If the `peer-verify` option is set to `trusted`, a CA certificate must be provided.

Default: None

Example

```
example:/config/logging/destination-example/#> ca-certificate my-ca-cert
```

Options

Option	Description
`no`	Remove the CA certificate from the destination.
`NAME`	The name of the CA certificate to use.

`[no] revocation`

Description

Enable or disable certificate revocation checking during the TLS handshake. **Default:** Disabled.

If enabled, the system will reject connections if the cached Certificate Revocation List (CRL) is outdated, fails signature verification, or if the certificate is explicitly listed as revoked in the CRL. **Note:** If a CRL is missing, the connection is still accepted.

CRLs must be configured in the `config/pki` context.

continues on next page

continued

`[no] revocation`

Options

Option	Description
`no`	Disable certificate revocation checking.

For details on configuring static CRL distribution points, see the Revocation Configuration Guide.

Filter Configuration

Each filter configuration has its sub-context under the `logging` context:

```
example:/config/logging/#> filter example
example:/config/logging/filter-example/#>
```

`[no] name NAME`

Description

Change the name of the filter. The name is a free form string that identifies the filter. The name is used to select the filter when configuring a sink.

> **NOTE**
>
> If the name is changed it will also be changed in all sinks that uses this filter, so it is not necessary to update all sinks that uses this filter.

Example

```
example:/config/logging/filter-example/#> name my-filter
```

Options

Option	Description
`no`	Does nothing for this command, the filter must have a name.
`NAME`	The free form name of the filter. It can be up to 15 characters long. It allows for the following characters: `a-z`, `A-Z`, `0-9`, `-`, and `_`.

`[no] invert`

Description

Enable if the entire filter logic should be inverted.

Normally the configured filter will match messages that conform directly to the specified filter, e.g. if we filter on facility auth any message not matching that facility will be filtered out. However, if we would enable invert, any message matching the filter would be filtered out instead.

continues on next page

continued

`[no] invert`

For instance, by inverting filters we can more easily filter out messages when we may want to include more messages than we exclude.

Default: Disabled, i.e. `no invert`.

Example

```
example:/config/logging/filter-example/#> invert
```

Options

Option	Description
`no`	Disables the invert option.

`[no] facility FACILITY[ FACILITY ...|,FACILITY ...]`

Description

Filter messages based on their facility.

By specifying one or more facilities, only messages that match the specified facilities will be allowed through the filter.

Default: None

Example

```
example:/config/logging/filter-example/#> facility auth
example:/config/logging/filter-example/#> facility auth authpriv
example:/config/logging/filter-example/#> facility auth,authpriv
```

> **TIP**
>
> Both comma and space can be used as separators between facilities.

Individual facilities can be added or removed to the active list, i.e. each call to the command does not replace the existing list of facilities.

```
example:/config/logging/filter-example/#> facility auth
example:/config/logging/filter-example/#> facility authpriv
example:/config/logging/filter-example/#> facility security
example:/config/logging/filter-example/#> show facility
auth,authpriv,security
example:/config/logging/filter-example/#> no facility auth
example:/config/logging/filter-example/#> show facility
authpriv,security
```

In order to remove all of the configured facilities, use the `no facility`, with no specified facilities.

continues on next page

continued

`[no] facility FACILITY[ FACILITY ...|,FACILITY ...]`

```
example:/config/logging/filter-example/#> no facility
```

Options

Option	Description
`no`	Removes all configured facilities from the filter.
`FACILITY[,FACILITY ...]`	The facility or facilities to filter on. Refer to the Facility section for a list of available facilities. **TIP** Tab completion is available for the `FACILITY` argument, to list all existing facilities.

`[no] level LEVEL[ LEVEL ...|,LEVEL, ...|..LEVEL|..]`

Description

Filter messages based on the severity level of the message.

By specifying one or more severity levels, only messages that match any of the specified severity levels will be allowed through the filter.

Default: None

Example

```
example:/config/logging/filter-example/#> level warning
example:/config/logging/filter-example/#> level warning error
example:/config/logging/filter-example/#> level warning,error
```

> **TIP**
>
> Both comma and space can be used as separators between severities.

Also levels can be specified as ranges, e.g. `info..warning` will match all severity levels from `info` to `warning`.

```
example:/config/logging/filter-example/#> level info..warning
```

Further, all levels from a certain level and up can be specified, e.g. `notice..` will match all severity levels from `notice` and up.

```
example:/config/logging/filter-example/#> level notice..
```

Individual severity levels can be added or removed to the active list, i.e. each call to the command does not replace the existing list of severity levels.

continues on next page

continued

`[no] level LEVEL[ LEVEL ...|,LEVEL, ...|..LEVEL|..]`

```
example:/config/logging/filter-example/#> level warning
example:/config/logging/filter-example/#> level error
example:/config/logging/filter-example/#> level crit
example:/config/logging/filter-example/#> show level
warning,error,crit
example:/config/logging/filter-example/#> no level warning
example:/config/logging/filter-example/#> show level
error,crit
```

In order to remove all of the configured severity levels, use the `no level`, with no specified severities.

```
example:/config/logging/filter-example/#> no level
```

Options

Option	Description
`no`	Removes all configured severity levels from the filter.
`LEVEL`	The level of the severity or severities to filter on. Refer to the Severity section for a list of available severities. **TIP** Tab completion is available for the `LEVEL` argument, to list all existing severities.

`[no] hostname "NAME"`

Description

Filter messages based on the hostname field of the message.

It is important to note that this filter will only be performed on the actual hostname field of the the specific log message. This means that this filter will not match on the IP address of a given host, even if the IP address and the host field refers to the same actual host.

> **NOTE**
>
> Make sure to use double quotes around the hostname to match, e.g. `"my-example-host"`.

> **TIP**
>
> Supports the use of regular expressions.

continues on next page

continued

`[no] hostname "NAME"`

> **TIP**
>
> For filtering based on the IP address use the ‘network’ filter instead.

Default: None

Example

Match on a specific hostname:

```
example:/config/logging/filter-example/#> hostname "my-example-host"
```

Use regular expressions to match on multiple hostnames, e.g. all hosts starting with `my-example-host` followed by a digit:

```
example:/config/logging/filter-example/#> hostname "my-example-host[0-9]"
```

Options

Option	Description
`no`	Removes the hostname filter from the filter.
`"NAME"`	The name of the hostname to filter on, or a regular expression to match on.

`[no] message "MESSAGE"`

Description

Filter messages based on the text content of the message.

This filter will match on the actual text content of the message, and will only allow messages through that matches the specified text.

> **NOTE**
>
> This filtering will only be performed on the text content of the message and not on the message headers.

> **NOTE**
>
> Make sure to use double quotes around the message to match, e.g. `"my example message"`.

> **TIP**
>
> Supports the use of regular expressions.

Default: None

continues on next page

continued

`[no] message "MESSAGE"`

Example

Match on a specific message:

```
example:/config/logging/filter-example/#> message "my example message"
```

Use regular expressions to match on specific patterns in the message content. For example, to filter all messages containing the word “error” (in any case):

```
example:/config/logging/filter-example/#> message "^.\*[Ee]rror.\*$"
```

In this example, the regular expression “^.*[Ee]rror.*$” is used. Here’s what each part of the expression means:

- `^` : This symbol represents the start of a line.
- `.*` : This matches any character (except for a newline) 0 or more times.
- `[Ee]rror` : This matches the word “error” in any case (i.e., “Error” or “error”).
- `.*` : This again matches any character (except for a newline) 0 or more times.
- `$` : This symbol represents the end of a line.

So, this regular expression will match any line that contains the word “error” in any case, regardless of what comes before or after “error” in the line.

Options

Option	Description
`no`	Removes the message filter from the filter.
`"MESSAGE"`	The message content to filter on, or a regular expression to match on.

`[no] program "NAME"`

Description

Filter messages based on the name of the program that generated it.

This will filter the messages against the program field for the log messages. It will match on the application name that generated the message, not the process ID or any other field.

> **NOTE**
>
> Make sure to use double quotes around the message to match, e.g. `"my example message"`.

> **TIP**
>
> Supports the use of regular expressions.

Default: None

continues on next page

continued

`[no] program "NAME"`

Example

Use the `program` command to match on a specific program name:

```
example:/config/logging/filter-example/#> program "my-example-program"
```

In this example, the `program` command is used to filter log messages from a specific program named “my-example-program”.

You can also use regular expressions to match on specific patterns in the program name:

```
example:/config/logging/filter-example/#> program "^udhcpc\s*:vlan\d$"
```

In this second example, the regular expression “^udhcpc\s*:vlan\d$” is used. Here’s what each part of the expression means:

- `^` : This symbol represents the start of a line.
- `udhcpc` : This matches the exact string “udhcpc”.
- `\s*` : This matches any whitespace character 0 or more times.
- `:vlan` : This matches the exact string “:vlan”.
- `\d` : This matches any digit.
- `$` : This symbol represents the end of a line.

So, this regular expression will match any line that starts with “udhcpc”, followed by any number of whitespace characters, followed by “:vlan”, and ends with a digit.

Options

Option	Description
`no`	Removes the program filter from the filter.
`"NAME"`	The name of the program to filter on, or a regular expression to match on.

`[no] network ADDR/LEN`

Description

Filter messages based on the provided IP subnet.

If the IP address of a sender belongs to the subnet provided the message will be considered a match. Hence, it is possible to match on specific addresses (using a /32 length) or a range of addresses.

> **NOTE**
>
> The filter will match against the address of the last-hop sender of the message, i.e. if the message have been relayed it will be the IP address of the relay and not the original sender of the message.

Default: None

continues on next page

continued

`[no] network ADDR/LEN`

Example

Filter log messages originating from any IP address within the 192.168.1.0/24 network:

```
example:/config/logging/filter-example/#> network 192.168.1.0/24
```

Filter log messages originating from a specific IP address, 192.168.2.2. The /32 subnet mask indicates a single IP address.

```
example:/config/logging/filter-example/#> network 192.168.2.2/32
```

Options

Option	Description
`no`	Removes the network filter from the filter.
`ADDR/LEN`	The IP address and subnet length to filter on. The subnet length specifies the number of bits in the subnet mask. The subnet length must be in the range of 1-32.

`[no] rate-limit [per host|source-ip|facility|level] RATE`

Description

Filter out messages if they exceed the provided message rate.

When this configuration option is used messages will be filtered out, i.e. dropped, if the messages per second exceed the set rate value.

It is also possible to optionally perform the rate limit per a specific type of content that is part of the message. The following options can be selected from, but only one at a time per filter:

- `host`: The host specified in the message.
- `source-ip`: The source IP address of the sender, if the message has been relayed this is the address of the relay.
- `facility`: The facility of the message.
- `level`: The severity level of the message.

If the `rate-limit` is configured for a specific type it means that the imposed rate is applied for each individual type when a message is handled. As an example, if we have configured a `rate-limit` of 100 per host, and we have two hosts transmitting messages, a total of 200 messages will be allowed, 100 for each of the transmitting hosts.

If no specific type is provided the rate will be applied for every message, regardless of their content.

> **NOTE**
>
> The rate provided will always be in number of messages per second.

Default: No rate limiting, i.e. `no rate-limit`.

continues on next page

continued

`[no] rate-limit [per host|source-ip|facility|level] RATE`

Example

Filter out messages if they exceed 100 messages per second:

```
example:/config/logging/filter-example/#> rate-limit 100
```

Filter out messages if they exceed 100 messages per second per host:

```
example:/config/logging/filter-example/#> rate-limit per host 100
```

Options

Option	Description
`no`	Disables the rate limiting.
`per host`	Perform the rate limiting per host.
`source-ip`	Perform the rate limiting per source IP address.
`facility`	Perform the rate limiting per facility.
`level`	Perform the rate limiting per severity level.
`RATE`	The maximum number of messages that can be sent per second. The value must be a non-negative integer.

Sink Configuration

Each sink configuration has its sub-context under the `logging` context:

```
example:/config/logging/#> sink example
example:/config/logging/sink-example/#>
```

`[no] name NAME`

Description

Change the name of the sink. The name is a free form string that identifies the sink.

Example

```
example:/config/logging/sink-example/#> name my-sink
```

Options

Option	Description
`no`	Does nothing for this command, the sink must have a name.
`NAME`	The free form name of the sink. It can be up to 15 characters long. It allows for the following characters: `a-z`, `A-Z`, `0-9`, `-`, and `_`.

`[no] source SOURCE[ SOURCE ...|,SOURCE ...]`

Description

Define the source(s) from which the log messages should be sent.

By specifying one or more sources, you direct the log messages from the designated sources to the sink.

> **NOTE**
>
> For the sink to be valid, at least one source must be specified.

Default: None

Example

```
example:/config/logging/sink-example/#> source example-1
```

To specify multiple sources simultaneously:

```
example:/config/logging/sink-example/#> source example-1 example-2
example:/config/logging/sink-example/#> source example-1,example-2
```

Options

Option	Description
`no`	This command removes all configured sources from the sink.
`SOURCE`	This argument represents the name of the source to be associated with the sink. For more information on configuring sources, refer to the Source Configuration section. **TIP** Tab completion is available for the `SOURCE` argument, providing a list of all existing sources.

`[no] destination DESTINATION[ DESTINATION ...|,DESTINATION ...]`

Description

Define the destination(s) to which the log messages should be sent.

By specifying one or more destinations, you direct the log messages to the designated destinations.

> **NOTE**
>
> For the sink to be valid, at least one destination must be specified.

Default: None

continues on next page

continued

`[no] destination DESTINATION[ DESTINATION ...|,DESTINATION ...]`

Example

```
example:/config/logging/sink-example/#> destination example-1
```

To specify multiple destinations simultaneously:

```
example:/config/logging/sink-example/#> destination example-1 example-2
example:/config/logging/sink-example/#> destination example-1,example-2
```

Options

Option	Description
`no`	This command removes all configured destinations from the sink.
`DESTINATION`	This argument represents the name of the destination to be associated with the sink. For more information on configuring destinations, refer to the Destination Configuration section. **TIP** Tab completion is available for the `DESTINATION` argument, providing a list of all existing destinations.

`[no] filter [and|or] FILTER [ [and|or] FILTER ...]`

Description

Define optional filter(s) to include or exclude specific log messages.

By default, filters are combined using the AND operator. However, you can also use the OR operator to combine filters. The operator should be specified before the filter.

The order of the filters and the combination operator are crucial. The AND operator takes precedence over the OR operator.

> **NOTE**
>
> When using multiple filters, the AND operator evaluates first, followed by the OR operator. This order of operations can affect the outcome of the filter combination.

Default: None

Example

Specify a single filter:

```
example:/config/logging/sink-example/#> filter filter-1
```

Specify multiple filters using the `and` operator:

continues on next page

continued

`[no] filter [and|or] FILTER [ [and|or] FILTER ...]`

```
example:/config/logging/sink-example/#> filter filter-1 and filter-2
```

If you list multiple filters without specifying an operator, the `and` operator is assumed. Therefore, the following command is equivalent to the previous command:

```
example:/config/logging/sink-example/#> filter filter-1 filter-2
```

Specify multiple filters using the `or` operator:

```
example:/config/logging/sink-example/#> filter filter-1 or filter-2
```

Example: Advanced Filter Combination

Combine filters using both `and` and `or` operators. Remember, the `and` operator takes precedence over the `or` operator:

```
example:/config/logging/sink-example/#> filter filter-1 and filter-2 or filter-3 and filter-4
```

This command works as follows:

- **filter-1** `and` **filter-2**: This part of the command will match log messages that meet both the conditions of **filter-1** and **filter-2**.
- `or`: This operator combines the results of the two surrounding expressions.
- **filter-3** `and` **filter-4**: This part of the command will match log messages that meet both the conditions of **filter-3** and **filter-4**.

So, the entire command will match log messages that either meet both the conditions of **filter-1** and **filter-2**, or meet both the conditions of **filter-3** and **filter-4**.

Remember, the `and` operator takes precedence over the `or` operator. So, the `and` operations are evaluated first, followed by the `or` operation.

Options

Option	Description
`no`	This command removes all configured filters from the sink.
`and`	This operator is used to combine multiple filters. When the `and` operator is used, a log message must meet all the specified filter conditions to be matched. **NOTE** This is the default operator when combining multiple filters, when no operator is specified.
`or`	This operator is also used to combine multiple filters. However, when the `or` operator is used, a log message can meet any one of the specified filter conditions to be matched.

continues on next page

continued

`[no] filter [and|or] FILTER [ [and|or] FILTER ...]`

Option	Description
`FILTER`	This argument represents the name of the filter to be associated with the sink. For more information on configuring filters, refer to the Filter Configuration section. **TIP** Tab completion is available for the `FILTER` argument, providing a list of all existing filters.

Status

In the **exec** context, it is possible to view the current status of the logging setup. We can use the `show logging` command to obtain a summary of the current setup:

```
example:/#> show logging
 Source Summary
SOURCE           PROCESSED
local-userspace  329
sink-remote      879
local-kernel     1

 Destination Summary
DESTINATION  PROCESSED
auth         3
ppp          0
cli          0
sink-file    879
syslog       327
kern         1
openvpn      0
alarm        1
messages     326
ntp          2
sink-relay   879

 Filters
FILTER    MATCHED  NOT_MATCHED
ntp       2        328
auth      3        327
ppp       0        330
cli       0        330
syslog    327      3
kern      1        329
openvpn   0        330
alarm     1        329
messages  326      4

example:/#>
```

> **NOTE**
>
> The logging status information will display information on all the logging configured on the system, this includes any of the default logging as well.
>
> Anything that comes from a custom logging sink configuration should be prefixed with `sink-` when they are referred to in the status output.

This is also a context that can be entered, with additional commands to view the status of specific aspects of the logging setup:

```
example:/#> logging
example:/config/logging/#>
```

`show source [connection|facility|program|severity|summary|verbose]`

Description

Show source related status information.

A number of options are available to show different levels of detail for the source status. If no option is provided, a summary of the source status will be shown.

Example

```
example:/config/logging/#> show source
╒ Sources ═══════════════════════════════════════════════════
TYPE        NAME             DESCRIPTION  PROCESSED
remote      sink-remote      192.168.1.1  892
unix-dgram  local-userspace  /dev/log     318
file        local-kernel     /dev/kmsg    1

client1:/logging/#>
```

Options

Option	Description

`connection`	Show status information for remote connections over TCP.
	
	Example
	
	`example:/config/logging/#> show source connection`
	`[ ... ]`

`facility`	Show a status summary on the number of messages that have been generated per facility.
	
	Example
	
	`example:/config/logging/#> show source facility`
	`[ ... ]`

`program`	Show a status summary on the number of messages that have been generated by each application/program.

continues on next page

continued

`show source [connection|facility|program|severity|summary|verbose]`

Option	Description

	Example
	`example:/config/logging/#> show source program`
	`[ ... ]`

`severity`	Show a status summary on the number of messages per severity level.
	Example
	`example:/config/logging/#> show source severity`
	`[ ... ]`

`summary`	Show a summary of the source status.
	Example
	`example:/config/logging/#> show source summary`
	`[ ... ]`

`verbose`	Show more detailed status information for the source.
	WARNING
	This command includes some verbose information and may occupy additional width in the terminal, more so than usual.
	Example
	`example:/config/logging/#> show source verbose`
	`[ ... ]`

`show destination [file|remote|summary|verbose]`

Description

Show destination related status information.

A number of options are available to show different levels of detail for the destination status. If no option is provided, a summary of the destination status will be shown.

Example

```
example:/config/logging/#> show destination
╒ Destinations ═══════════════════════════════════════════════════════════════════════════════
│TYPE    NAME        DESCRIPTION              PROCESSED  DROPPED  WRITTEN  QUEUED  MEMORY_USAGE
│remote  sink-relay  tls,server1:6514         735        0        735      0       0
│remote  sink-relay  tls,Syslog-Server:6514   285        0        0        0       0
│file    alarm       /var/log/alarm           1          0        1        0       0
│file    sink-file   /log//from-hosts         888        0        888      0       0
│file    syslog      /var/log/syslog          330        0        330      0       0
│file    ntp         /var/log/ntp.log         2          0        2        0       0
│file    messages    /var/log/messages        329        0        329      0       0
```

continues on next page

continued

`show destination [file|remote|summary|verbose]`

```
file    auth        /var/log/auth.log       3          0        3        0       0
file    kern        /var/log/kern.log       1          0        1        0       0
```

Options

Option	Description

`file`	Show status information for file destinations.
	
	Example
	
	`example:/config/logging/#> show destination file`
	`[ ... ]`

`remote`	Show status information for remote destinations.
	
	Example
	
	`example:/config/logging/#> show destination remote`
	`[ ... ]`

`summary`	Show a summary of the destination status.
	
	Example
	
	`example:/config/logging/#> show destination summary`
	`[ ... ]`

`verbose`	Show more detailed status information for the destination.
	
	WARNING
	This command includes some verbose information and may occupy additional width in the terminal, more so than usual.
	
	Example
	
	`example:/config/logging/#> show destination verbose`
	`[ ... ]`

`show filter`

Description

Show filter related status information.

Example

```
example:/config/logging/#> show filter
 Filters
FILTER    MATCHED  NOT_MATCHED
```

continues on next page

continued

`show filter`

```
ntp       2        331
auth      3        330
ppp       0        333
cli       0        333
syslog    330      3
kern      1        332
openvpn   0        333
alarm     1        332
messages  329      4

client1:/logging/#>
```

`show all`

Description

Show all logging status information.

> **WARNING**
>
> This command includes some verbose information and may occupy additional width in the terminal, more so than usual.

Example

```
example:/config/logging/#> show all
[ ... ]
```

`show`

Description

Show a summary of the logging status.

Example

```
example:/config/logging/#> show
[ ... ]
```

Example

This section is intended to provide a simple example of how to configure a logging sink, and the associated components. For additional in-depth configuration examples, refer to the separate How-To guides.

SEE ALSO

Related Examples and Use-Cases
- Logging to Remote Machines
- Logging and DHCP Option 7
- Logging Relay
- Logging and Receiving From Remote Clients

Logging to Custom File on Internal RAM Disk

In this example we want to create a simple setup where we log all messages of severity `notice` or higher to a file on the internal RAM disk. This file will be rotated when it reaches a size of 1MB, and a maximum of 5 rotated files will be kept.

We start by configuring the source:

```
example:/#> configure
example:/config/#> logging
example:/config/logging/#> source local
example:/config/logging/source-local/#> userspace
example:/config/logging/source-local/#> kernel
example:/config/logging/source-local/#> end
example:/config/logging/#>
```

NOTE

A source named `local` exists by default from the factory configuration. But for this example we explicitly configure it to show how it is done.

Next we configure the destination:

```
example:/config/logging/#> destination file-1
example:/config/logging/destination-file-1/#> type file
example:/config/logging/destination-file-1/#> file-name my-file-1
example:/config/logging/destination-file-1/#> media internal
example:/config/logging/destination-file-1/#> size 1M
example:/config/logging/destination-file-1/#> rotate 5
example:/config/logging/destination-file-1/#> end
example:/config/logging/#>
```

Now we configure the filter:

```
example:/config/logging/#> filter filter-1
example:/config/logging/filter-filter-1/#> level notice..
example:/config/logging/filter-filter-1/#> end
example:/config/logging/#>
```

Now that we have all the components we can configure the sink:

```
example:/config/logging/#> sink example
example:/config/logging/sink-example/#> source local
example:/config/logging/sink-example/#> destination file-1
example:/config/logging/sink-example/#> filter filter-1
```

```
example:/config/logging/sink-example/#> end
example:/config/logging/#>
```

The configuration overview should now look like this:

```
example:/config/logging/#> show
Console syslog   : Disabled
External logging : Disabled
Secure mode      : Enabled

 Sources
 NAME
 Type: local      USERSPACE  KERNEL  AUDIT
 local            Yes        Yes     No

 Destinations
 NAME
 Type: file       MEDIA     PATH
 file-1           internal  my-file-1

 Filters
 NAME             INVERT        TYPE  FILTER
 filter-1         No           Level  notice,warning,err,crit,alert,emerg

 Sinks
 NAME             SOURCES       DESTINATIONS   OP FILTERS
 example          local         file-1            filter-1

example:/config/logging/#>
```

Now we can leave the configuration context and apply the changes:

```
example:/config/logging/#> leave
example:/#>
```

At this point the logging from the local device to the file `my-file-1` should be active. It should be possible to view the log messages in the file now:

```
example:/#> show logfile my-file-1
[ ... ]
May 20 12:53:05 example clish[906]: User 'admin' entering domain shell.
May 20 12:53:16 example clish[906]: User 'admin' leaving domain shell.
──────────────────────────────────────────────────────────────────────────
my-file-1                                      Mon May 20 12:53:16 2024
```

Troubleshooting

If you are experiencing issues with your logging configuration, this section provides some general troubleshooting steps to help you identify and resolve the problem.

The status commands, covered in the Status section, can be used to provide an overview of the current logging setup. This can be useful to identify if there are any basic issues with the configuration.

Advanced Troubleshooting

In this section, we provide some advanced troubleshooting steps to help you identify and resolve issues with your logging configuration. For these steps, we will access the system shell to perform additional checks and diagnostics.

> **NOTE**
>
> To access the system shell, you must connect to the device via the console using an administrative account with the appropriate permissions. While it is possible to access the shell via SSH, this feature must first be enabled through a console session.

```
example:/#> shell
Domain shell started.
root@example:~ #
```

The following are some advanced troubleshooting steps you can perform.

Verify the Syslog Configuration

Check that the syslog configuration can be parsed correctly by the application:

```
root@example:~ # syslog-ng -s -f /etc/syslog-ng-writer.conf
root@example:~ # syslog-ng -s -f /etc/syslog-ng-reader.conf
```

If there is no additional output, the configuration is correct.

> **WARNING**
>
> The system should not produce any invalid configurations for the syslog application. If you suspect that the configuration that has been generated is invalid, please contact technical support.

Check the Syslog Application

Check that the syslog applications are correctly running:

```
root@example:~ # initctl status syslog-ng:reader
     Status : running
   Identity : syslog-ng:reader
Description : System log reader daemon
     Origin : /etc/finit.d/syslog-ng.conf
Condition(s): <+pid/syslog-ng:writer>
    Command : /sbin/syslog-ng -c /var/run/syslog-ng/reader.ctl
              -R /var/run/syslog-ng/reader.persist
              -p /var/run/syslog-ng-reader.pid
              -F --cfgfile /etc/syslog-ng-reader.conf
   PID file : /run/syslog-ng-reader.pid
        PID : 614
       User : root
```

```
      Group : admin
     Uptime : 1 min 47 sec
   Restarts : 0 (0/10)
  Runlevels : [S-123456789]
     Memory : --.--
     CGroup : /system/syslog-ng cpu 0 [100, ] mem [, ]
              ├─ 610 syslog-ng -c /var/run/syslog-ng/writer.ctl
                                -R /var/run/syslog-ng/writer.persist
                                -p /var/run/syslog-ng-writer.pid -
              └─ 614 syslog-ng -c /var/run/syslog-ng/reader.ctl
                                -R /var/run/syslog-ng/reader.persist
                                -p /var/run/syslog-ng-reader.pid -
root@zero-12-34-50:~ #
```

```
root@zero-12-34-50:~ # initctl status syslog-ng:writer
     Status : running
   Identity : syslog-ng:writer
Description : System log writer daemon
     Origin : /etc/finit.d/syslog-ng.conf
    Command : /sbin/syslog-ng -c /var/run/syslog-ng/writer.ctl
                              -R /var/run/syslog-ng/writer.persist
                              -p /var/run/syslog-ng-writer.pid
                              -F --cfgfile /etc/syslog-ng-writer.conf
   PID file : /run/syslog-ng-writer.pid
        PID : 610
       User : root
      Group : admin
     Uptime : 5 min 49 sec
   Restarts : 0 (0/10)
  Runlevels : [S-123456789]
     Memory : --.--
     CGroup : /system/syslog-ng cpu 0 [100, ] mem [, ]
              ├─ 610 syslog-ng -c /var/run/syslog-ng/writer.ctl
                                -R /var/run/syslog-ng/writer.persist
                                -p /var/run/syslog-ng-writer.pid -
              └─ 614 syslog-ng -c /var/run/syslog-ng/reader.ctl
                                -R /var/run/syslog-ng/reader.persist
                                -p /var/run/syslog-ng-reader.pid -

root@zero-12-34-50:~ #
```

If the applications are running correctly, the output should look similar to the above.

> **WARNING**
>
> If any of the syslog applications are not running, or any of them reports its status as **crashed**, **stopped**, or similar, a system bug may have been encountered. Please contact technical support.

Generate a Test Log Message

Generate a log message using the `logger` command to test the logging functionality. For example:

```
root@example:~ # logger -t test "This is a test log message"
```

It is also possible to generate messages with a specific facility and severity:

```
root@example:~ # logger -t test -p local0.info "This is a test log message"
```

Then check the relevant log file to see if the message was logged correctly.

2.7.4. Front Panel LEDs

The LED functionality is described in the User Guide[1] of your WeOS5 product. Here the information on LED functionality of *all* WeOS5 products is summarised. Note that your product may not have all LED types listed here.

LED	Status	Description
DC1	OFF	Product has no power
	GREEN	Power OK on DC1
	RED	Power failure on DC1
DC2		See DC1 above (replace DC1 with DC2 where applicable)
DC/AC1		See DC1 above (replace DC1 with DC/AC1 where applicable)
DC	OFF	Product has no power
	GREEN	Power OK on DC1 and DC2
	RED	Power failure on DC1 or DC2
ON	OFF	Unit has no power
	GREEN	All OK, no alarm condition
	RED	Alarm condition, or until product has started up.
TX/FX	OFF	No link
	GREEN	Link established
	GREEN FLASH	Data traffic indication
	YELLOW	Port alarm, or port is set in blocking state by link redundancy protocol.
	YELLOW FLASH	(All TX/FX flashing yellow) Location indicator (“Here I am”). Activated by specific Web/CLI setting.
RSTP	OFF	RSTP disabled
	GREEN	RSTP enabled
	BLINK	Unit elected as RSTP/STP root switch
FRNT	OFF	FRNT disabled
	GREEN	FRNT OK
	RED	FRNT error
	BLINK	Unit configured as FRNT focal point
RING	OFF	All ring protocols are disabled
	GREEN	All ring protocol are OK
	GREEN BLINK	All ring protocol are OK. A ring protocol acts as master/focal-point/root.
	RED	A ring protocol has error

[1] https://www.westermo.com

LED	Status	Description
	RED BLINK	A ring protocol has error. A ring protocol acts as master/focal-point/root.
PoE	OFF	No PoE output
	GREEN	PoE output active
	GREEN FLASH	Error or negotiating power output
L1/L2	OFF	Relays in bypass mode
	GREEN	Relays connected to unit (no bypass)
SIGN	GREEN	Signature checks are enabled and device has successfully validated an image.
	RED	Signature checks are disabled or device has failed to validate an image.
USR1		Unused. Configurable in future release.
USR2		Unused. Configurable in future release.

The ‘ON’ LED constitutes a configurable *summary alarm*. The port LEDs (‘TX/FX’) indicate alarm for link status error if *link alarm* has been configured for the port. See Alarm Configuration for more information on summary alarm and link alarm.

When ‘location indicator’ is activated, all port LEDs flash yellow.

The ‘SIGN’ LED will illuminate RED if the integrity check fails for the first bootable image, even if subsequent images are successfully validated.

2.8. Network Services

2.8.1. DHCP Relay Agent

Introduction

A DHCP relay agent acts as a proxy for all DHCP traffic between DHCP clients on a LAN and a DHCP Server, usually located on a remote network. The most common use-cases for a DHCP relay agents are:

- **Centralised management:** Deploying and managing a DHCP server on every LAN in your network is cumbersome. By use of relay agents, a central DHCP server can be used, and the management effort is substantially reduced. Furthermore, if the relay agent is located in a router or switch on the local LAN, there is no additional equipment cost.
- **IP address per port:** In certain setups IP addresses should be assigned based on the switch port a DHCP client connects to. A DHCP relay agent on that switch can can include port information (DHCP Option 82) when forwarding the DHCP message to the server.

There are three dedicated HowTos that can be used as quick-start guides:
- DHCP IP per-port
- DHCP Relay/Server in a Ring
- DHCP Client/Relay/Server

Overview

One of the main reasons for using DHCP relay agents is to simplify DHCP management in larger infrastructures. Instead of deploying and managing a DHCP server on every LAN, a DHCP relay agent can forward DHCP messages between local DHCP clients and a central DHCP server.

```
DHCP pools:
A: 192.168.0.100-150/24, gw 192.168.0.1
B: 192.168.1.100-150/24, gw 192.168.1.1
C: 192.168.2.100-150/24, gw 192.168.2.1
D: 192.168.3.100-150/24, gw 192.168.3.1

.--------.      .--.-.
| DHCP   |     ( (    )__  Internet/Intranet
| Server +----(_,  \ ) ,_)------------------.
|        |      '-'--`--'                   |
'--------'        |   `---------.           |              .-------.
192.168.100.1     |             |           |          |   | DHCP  |
               .---+---.     .---+---.  .----+----.    +---+ Relay |
               | DHCP  |     | DHCP  |  |         |.1  | .2| Agent |
               | Relay |     | Relay |  |  Router +----+   | (RA3) |
               | Agent |     | Agent |  |         |    |   '-------'
               | (RA1) |     | (RA2) |  '---------'    |
               '---+---'     '-+---+-'                 |   .-----.
    192.168.0.0   |.1       .1|   |.1                 +---+ PC6 |
    ---+-------+--'      --+--'   '--+---             |   '-----'
       |       |           |         |                |
    .--+--. .--+--.     .--+--.   .--+--.             |   .-----.
    | PC1 | | PC2 |     | PC3 |   | PC4 |             +---+ PC6 |
    '-----' '-----'     '-----'   '-----'             |   '-----'
```

- **(V)LAN interfaces:** The DHCP relay agents (here RA1-RA3) serve DHCP clients (here PC1-PC6) on the local LANs. A DHCP relay can serve a single LAN (RA1 and RA3) or multiple LANs (RA2).

  A relay agent serving multiple LANs (RA2) can be configured to forward the DHCP requests from different LANs to different sets of DHCP servers.

- **DHCP Servers:** The relay agent must also know where to forward the DHCP requests from the local PCs, i.e., the relay agent must be configured with IP address of the DHCP server (here

192.168.100.1). For redundancy purposes, the DHCP relay agent allows specifying up to two DHCP servers, to which the relay agent forwards incoming DHCP requests.

DHCP servers listen to UDP port 67 by default. The relay agent can forward packets to a different port on the server, this is neccessary in case you wish to hand out IP addresses per port (DHCP Option 82) on a local DHCP server.

- **Address pools:** The DHCP server will in turn be configured with appropriate address pools (here denoted A-D), from which it can hand out addresses to the local PCs. When a DHCP relay agent receives a DHCP request from a PC, it will add its local IP address into the *giaddr* field of the DHCP message when forwarding it to the server (e.g., RA1 will set *giaddr* to 192.168.0.1) when forwarding requests from PC1 to the DHCP server). Based on the *giaddr*, the DHCP server can distinguish which pool to hand out address from (here "A").
- **Running relay agents on routers or switches:** Relay agents can be run as dedicated servers (RA3), but are typically located inside the local routers (RA1 and RA2). By running the relay agents inside the routers, deployment and management costs are reduced, since no additional equipment is needed.

  Although, not shown in the figure above, it is also possible to run relay agents on (layer-2) switches. This is useful when you wish to assign IP addresses based on the physical port the PC connects to. (See section about DHCP Option 82 below).

DHCP Option 82

In some topologies, you may wish to assign IP addresses based on the switch port a DHCP client connects to. By running a DHCP Relay Agent in the local switch/router, it can include port information when forwarding the DHCP messages (DHCP Option 82). If the packet already contain a Option 82, it will be forwarded unmodified, if not a Option82 will be added.

Circuit ID is set from the portname, or can be overrided per port where you can set it manual (as string or hex).

Remote ID tells how the remote id field of option 82 will be set. It can have several modes:
- *MAC* uses the base MAC address of the unit. (Default)
- *IP* sets it to the giaddr.
- *System name* uses the hostname of the system.
- *Manual* specify Remote ID either as string or hex values.

Server Identity Override

By enabling the force DHCP server override setting, the DHCP relay agent can work-around older DHCP servers that do not support RFC5107, which is a hint/extension to Option 82, by overriding Option 54 in the server response to the client with the relay agent's IP address.

It is recommended to leave this setting disabled and instead either use the local DHCP server, which does not have this issue, or upgrade the main DHCP server to be RFC compliant.

Configuration

The DHCP relay is a top-level configuration context in the CLI:

```
example:/#> configure
example:/config/#> dhcp-relay
example:/config/dhcp-relay/#>
```

`[no] enable`

Description

Enable, or disable DHCP relay agent. This option can be used to temporarily disable the relay agent, the configuration remains in `startup-config.cfg`.

To completely deactivate the relay agent, use `no dhcp-relay` from the top-level configuration context in the CLI.

Options

Option	Description
`no`	Disable relay agent

`[no] server [IPADDR[:PORT]|IPADDR[:PORT],IPADDR[:PORT]]`

Description

Set address and port of DHCP server(s).

For redundancy purposes, the DHCP Relay Agent enables you to specify up to two DHCP servers, to which the Relay Agent forwards all incoming DHCP requests. The listening port for the DHCP server (default 67), can also be set per server.

The optional port setting changes the Internet IP protocol port the relay agent uses to forward requests to the DHCP server(s). This is sometimes desirable to protect DHCP server(s) from receiving client requests directly. I.e., when all requests must go through a relay agent which does not support DHCP snooping or has snooping disabled.

Options

Option	Description
`no`	Remove all servers
`no IPADDR`	Remove server matching `IPADDR`
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1
`PORT`	Port number in range 1-65535.

`[no] udp-listen-port [PORT]`

Description

Set the udp port the relay agent is listening on.

This setting allows changing the relay source port that the relay agent listen on for responses from the upstream DHCP server.

Generally a DHCP server will respond on port 67 when it has received a DHCP request forwarded from a DHCP relay agent. In situations when this is not the case, this setting allows for reconfiguration of the relay agent so it can receive responses from the DHCP server on the wanted udp port.

Default: 67.

continues on next page

continued

`[no] udp-listen-port [PORT]`

Options

Option	Description
`no`	Reset to the default port 67.
`PORT`	Port number in range 1-65535.

`[no] force-server-identity`

Description

Force Server Identity (Option 54), default disabled.

This is a compatibility option for some DHCP servers that do not support RFC5107, sub-option 11 to Option 82.

When enabled the relay agent always sets Option 54 to point back to itself so that the client uses the relay as a full DHCP proxy.

Should only be needed with older DHCP servers.

Options

Option	Description
`no`	Disable Option 54, this is the default

`[no] iface [IFACE]`

Description

Set interface to service. DHCP messages from all ports in this VLAN are forwarded to the configured server(s).

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`no`	Clear the setting
`IFACE`	Enable DHCP Relay to service a given interface, e.g. `vlan1`.

`[no] option82`

Description

Enable DHCP option 82.

Option 82, also called relay agent information, adds a standard field to all incoming DHCP requests before they are forwarded to the server. This field contains the circuit-id and the

continues on next page

continued

`[no] option82`

remote-id. By default, the relay agent will add the access port name as circuit-id and the MAC address of the `iface` as remote-id, respectively.

Options

Option	Description
`no`	Disable Option 82, default

Activating this setting enters the next level of the configuration where the following settings are possible:

`[no] remoteid-type <mac | ip | system-name | manual <string|hex> <VALUE>>`

Description

Set Option 82 Remote ID type.

Set the Remote ID type, used only when Option 82 is enabled, to either use the MAC, IP, system name of the switch or a manual value (string or hex). Default is to use the MAC address.

Options

Option	Description
`no`	Reset `remoteid-type` to its default value, MAC
`mac`	Use the MAC address of `iface` as the Option 82 remote-id
`system-name`	Use the hostname as the Option 82 remote-id
`manual`	Manually specify Option 82 remote-id, using hex or string

Iface Settings

When providing an iface for the DHCP relay to serve, a new sub-context will be opened:

```
example:/config/dhcp-relay/#> iface vlan1
example:/config/dhcp-relay/iface-vlan1/#>
```

`[no] server [IPADDR[:PORT]|IPADDR[:PORT],IPADDR[:PORT]]`

Description

Same as the global dhcp-relay setting.

If this option is not configured it will *inherit* the global DHCP relay server configuration, if set it will override it.

Options

Option	Description
`no`	Remove any iface specific server that has been configured and simply inherit the global server configuration.

continues on next page

continued

`[no] server [IPADDR[:PORT]|IPADDR[:PORT],IPADDR[:PORT]]`

Option	Description
`no IPADDR`	Remove server matching `IPADDR`
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1
`PORT`	Port number in range 1-65535.

`[no] giaddr [IPADDR]`

Description

Set the GIADDR to use in relayed DHCP requests.

The GIADDR (Gateway IP Address) is inserted by the relay agent to identify the subnet on which the DHCP client resides. This field contains an IP address of the relay agent's interface that is on the same network as the client. This allows the DHCP server to see that the request came from a relay agent and to determine which IP address pool to use for the client. The DHCP server will send its reply (unicast) to the DHCP server port (UDP 67) at this GIADDR.

This option can be either **auto** (set by `no giaddr`) or a specific IP address. If a specific IP is set, it will be used as GIADDR regardless of the actual address on the interface. If **auto** the GIADDR will be obtained from the interface itself, in two different ways depending on the configuration of the iface:

- If the iface is configured with `inet static` the GIADDR will be set to the statically configured IP address on the interface.
- If the iface is configured with `inet dhcp` the GIADDR will be set to the IP address that is assigned to the interface by the DHCP client. Before an IP address is assigned to the iface by the DHCP client the relay agent will have no GIADDR to use. In this case any DHCP requests received on the iface will be discarded until the iface is assigned an IP address. At this point the GIADDR will be updated and the relay agent will start to relay.

  If the DHCP lease on the iface is released or expires, the relay will have no GIADDR to use and it will stop relaying any DHCP requests received on the iface.

> **IMPLICIT PRIORITY OF STATIC IP OVER DHCP ADDRESS FOR AUTO**
>
> When giaddr is set to **auto**, and the selected interface is configured with both `inet static` and `inet dhcp`, the static IP address has a higher priority and will be used as GIADDR. This means that a DHCP obtained address for the selected iface will only be used as GIADDR if no static IP address is configured on the iface.

Default: Auto (i.e. `no giaddr`)

Options

Option	Description
`no`	Automatically select address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1

`[no] port [PORT | PORT..PORT | PORT,PORT | all]`

Description

Port specific settings. E.g. exclude from relaying and any Option 82-specific information (circuit-id).

> **NOTE**
>
> Enters a sub-configuration context.

Options

Option	Description
`no`	Disable DHCP relay on the provided `PORT` or disable it on all.
`PORT`	Name of a port, example: eth1.

Port Settings

The port specific settings are located in a sub-context of the iface settings for the DHCP relay:

```
example:/config/dhcp-relay/#> iface vlan1
example:/config/dhcp-relay/iface-vlan1/#> port eth1
example:/config/dhcp-relay/iface-vlan1/port-eth1/#>
```

`[no] enable`

Description

Control relaying for port(s).

This setting controls if relaying should be enabled or disabled for the given port(s). Useful to disable relaying on intra-switch ports.

Default: enabled

Options

Option	Description
`no`	Disable relaying on the port.

`[no] circuitid-type [portname | manual]`

Description

Circuit-ID type for port(s).

Default: portname

Options

Option	Description
`no`	Reset to default, `portname`.

continues on next page

continued

`[no] circuitid-type [portname | manual]`

Option	Description
`portname`	Use the name of the port, e.g. “eth1” for port eth1.
`manual`	Use custom hexadecimal value or string: • `manual hex VALUE` • `manual string VALUE` **NOTE** Hexadecimal values can entered as: 0xXXXXXX, XXXXXX or XX:XX:XX.

2.8.2. DHCP Server

Introduction

Dynamic Host Control Protocol (DHCP) is a mechanism for clients in a network to receive a unique IP address, within its local network area, dynamically upon requesting it. Utilisation of this protocol can make configuration of network topologies a less daunting task, since it is not necessary to manually configure a static IP address for each individual device.

DHCP is capable of providing additional information in conjunction with IP addresses. For a device to properly function in a network, additional parameters might be necessary. Some of the more common parameters that DHCP can provide to its clients include default gateway, domain name, name servers, time servers.

If configuring a network containing multiple LANs, that should use DHCP, it may be inconvenient to configure several DHCP servers to handle each LAN separately. In that case a DHCP Relay can be utilised, see DHCP Relay. This allows for configuration of a single central DHCP Server that can serve multiple LANs. Subsequently, DHCP Relays can be configured to relay DHCP messages to the central DHCP Server from each LAN.

For example DHCP use-cases, see the DHCP HowTos:

- DHCP Server
- DHCP Client Relay Server
- DHCP IP Per Port
- DHCP Relay Server Ring.

Overview

DHCP is a protocol that uses a client/server model, by default operating on port 67 and 68. DHCP is based on the BOOTP protocol as specified in RFC 951[1]. Any client can ask for a IP address from any potential DHCP server on the network. The following image shows the basic steps that occur between a host and a DHCP server when an address is negotiated.

```
 Host-1                                        DHCP server
.-------. DHCP Discover (Broadcast)             .-------.
|       | ------------------------------------> |       |
|       |                  DHCP Offer (Unicast) |       |
|       | <------------------------------------ |       |
|       | DHCP Request (Broadcast)              |       |
|       | ------------------------------------> |       |
|       |                    DHCP ACK (Unicast) |       |
|       | <------------------------------------ |       |
'---+---'                                       '---+---'
    |                                               |
-----+-----------------------------------------------+-----
```

Figure 1: Communication between host client and server.

- **DHCP Discover**: Is sent from the host as a broadcast, so it may reach any potential DHCP server present on the network. The message indicates that the host is interested in acquiring an address configuration from a DHCP server.
- **DHCP Offer**: Is sent as unicast to the host where the Discover originated. Present in the offer is the DHCP configuration that the server wants to provide for the host. If multiple DHCP servers are present on the network they may all send an individual Offer.

[1] https://tools.ietf.org/html/rfc951

- **DHCP Request**: Is sent as broadcast to the server from the host to indicate that it wants to utilise the configuration that was offered by that particular server. The client can only send a request for one offer, in case it received multiple offers from different servers.
- **DHCP ACK**: Is sent from the server to the host as unicast. It servers as a final confirmation that the settings offered have been allocated for the client.

Each DHCP server have a pool of addresses that it oversees and allocates from when potential clients asks for a network configuration. This pool can vary in size and range depending on the configuration. Further, DHCP servers can also be instructed to hand out a specific configuration based on certain identifiers that can be extracted from the clients, e.g. client id, MAC, option 82.

The addresses that are handed out from the server can be specified to last for a given amount of time. One reason that this can be useful is that addresses can be reused if a client no longer needs it. Each time a lease expires, the concerned client needs to renew the address if it wants to continue to use it. In this case the client can request from the server the same address that it recently resided over, in most cases it will receive the same address. Nevertheless, it is possible to configure DHCP servers to hand out configurations with an infinite lease attached to it.

Additional configuration that can be passed by DHCP is known as DHCP options. A multitude of different DHCP options exist, several of which can be configured. To read more about DHCP options refer to RFC 2132[1] (DHCP Options and BOOTP Vendor Extensions). The following options can be configured:

- *DHCP Option 1:* Netmask.
- *DHCP Option 3:* Router IP address.
- *DHCP Option 6:* DNS Server.
- *DHCP Option 7:* Log Server.
- *DHCP Option 12:* Hostname.
- *DHCP Option 15:* Domain Search path.
- *DHCP Option 42:* NTP Server.
- *DHCP Option 51:* Lease time.
- *DHCP Option 66:* TFTP server name.
- *DHCP Option 67:* TFTP file.
- *DHCP Option 121/249:* Classless Static Route

More detailed information is available in RFC 2131[2].

Configuration

Configuration of DHCP server settings can be done at three levels of scope:

- **Global level**: Settings in the global scope can be seen as default values. They are valid for communication with all DHCP clients, unless overridden by a corresponding setting at subnet or host level.
- **Subnet level**: Settings at the subnet level apply to a specific IP subnet. They override corresponding settings done at global level, but may them- selves be overridden by settings at host level. Some settings only apply to the subnet level, e.g., specifying the IP range for the address pool.
- **Host level**: Settings at host level applies to individual DHCP clients. They override corresponding settings at global or subnet level. Some settings only apply to the host level, e.g., specifying the IP

[1] https://tools.ietf.org/html/rfc2132
[2] https://tools.ietf.org/html/rfc2131

address to hand out to a specific host. Assigning IP settings per host is referred to as ”static lease”, as opposed ”dynamic” assignment from a pool.

> **NOTE**
>
> An exception is classless routes (option 121/249); routes defined at subnet level complements routes defined at global level, and routes defined at host level complements routes defined at subnet and global levels.

Global Settings

DHCP server is a top-level configuration context in the CLI:

```
example:/#> configure
example:/config/#> dhcp-server
example:/config/dhcp-server/#>
```

Syntax

`[no] enable`

Description

Enable/disable the DHCP server. Useful to disable a fully setup DHCP server before deployment, the configuration will remain dormant while disabled.

Options

Option	Description
`no`	Disable dhcp-server

`[no] gateway [auto]`

Description

This setting controls the default value of the gateway setting per individual host/subnet configuration.

The value declared in this setting is overridden if it is configured in the host/subnet context.

Options

Option	Description
`no`	Disable gateway
`auto`	Default to this device IP

`[no] name-server auto|IPADDR[,IPADDR]`

Description

This setting controls the default value of the name-server setting per individual host/subnet configuration.

continues on next page

continued

`[no] name-server auto|IPADDR[,IPADDR]`

The value declared in this setting is overridden if it is configured in the host/subnet context.

Options

Option	Description
`no`	Disable name-server
`auto`	Default to this device IP
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1

`[no] log-server IPADDR[,IPADDR]`

Description

Specify optional log servers (DHCP option 7) to include in the DHCP offer. Up to two log servers can be specified.

Options

Option	Description
`no`	Disable the log server option or remove a specific server address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] route <NETWORK/LEN GATEWAY>`

Description

Specify optional classless route(s) assigned to any DHCP client served by this DHCP server. Routes defined here will be sent as DHCP options 121 and 249.

Options

Option	Description
`no`	Remove a specific route or all routes.
`NETWORK`	Network IP address in standard quad-dotted notation, e.g. 192.168.1.0.
`LEN`	Prefix length in CIDR notation (/24).
`GATEWAY`	Gateway IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] ping-check`

Description

Enable or disable ICMP ping check. By default, the DHCP server will check that an address is not in use before offering it to a client. In some rare cases it may be useful to disable this.

Parameters

continues on next page

continued

`[no] ping-check`

Options

Option	Description
`no`	Disable ping-check.

`[no] server-port PORT`

Description

Configure the DHCP server listening (UDP) port within the range 1..65535. The default configuration for a DHCP server listening port is 67.

Options

Option	Description
`no`	Reset to the default value (port 67).
`PORT`	Port number in range 1-65535.

`[no] client-port PORT`

Description

Set DHCP Server source (UDP) port in range 1..65535. By default, the server sends DHCP messages with source UDP port 68.

Options

Option	Description
`no`	Reset to the default value (port 68).
`PORT`	Port number in range 1-65535.

`[no] bootfile FILENAME`

Description

Configure the boot filename (as stored on the TFTP server).

The bootfile name is typically passed within the file field of a BOOTP/DHCP message, but is instead sent as DHCP option 67 if option overloading applies or if the client has requested DHCP option 67.

The value declared in this setting is overridden if it is configured in the host context.

Options

Option	Description
`no`	Disable bootfile.
`FILENAME`	String definition of the file name.

`[no] tftp-server [IPADDR|SERVERNAME]`

Description

Set the domain name or IP address of the boot server.

The TFTP server address to be handed out on DHCP requests from static host leases. Each static lease can also have its own unique server and bootfile setting.

> **NOTE**
>
> The value declared in this setting is overridden if it is configured in the host context.

Options

Option	Description
`no`	Remove any configured tftp-server.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1. Set the IP address of the server in both the option 66 field and next server address siaddr.
`SERVERNAME`	String definition of the tftp server FQDN. Set the domain name of the server. This is typically passed within the sname field of a BOOTP/DHCP message, but is instead sent as DHCP option 66 if option overloading applies or if the client has requested DHCP option 66.

`[no] host [INDEX]`

Description

Create and manage host specific DHCP server settings, i.e., static lease settings. This is typically used to configure a static lease based on MAC, Client-ID or port ID (i.e., DHCP Option 82).

Up to 1024 can be configured. Each entry is given an index in the range 1..1024.

Options

Option	Description
`no`	Remove host.
`INDEX`	Integer value in the range 1..1024.

`[no] subnet IPADDR/LEN | IPADDR[ NETMASK]`

Description

Create and manage DHCP server IP address pools.

When creating a subnet an IP address is provided. In addition, the subnet netmask can be specified as a prefix length or a netmask, with **/24** and **255.255.255.0** respectively being the default values.

continues on next page

continued

`[no] subnet IPADDR/LEN | IPADDR[ NETMASK]`

Up to 64 subnets for a single device can be configured.

Options

Option	Description
`no`	Remove a specific subnet or all of them.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.
`LEN`	Size of the net in CIDR notation (/24).
`NETMASK`	Size of the net in quad-dotted format, e.g. 255.255.255.0.

Host (Static Lease) Settings

Host is a sub-context in the DHCP server configuration:

```
example:/#> configure
example:/config/#> dhcp-server
example:/config/dhcp-server/#> host 1
example:/config/dhcp-server/host-1/#>
```

`[no] lease-time infinite|TIME`

Description

Specify the lease time in seconds, to be used for addresses handed out to DHCP clients. If this value is not configured the lease time will be defaulted to the one configured for the associated subnet.

Options

Option	Description
`no`	Disable host specific lease time, default to use the subnet lease time.
`infinite`	Specify the lease to be handed out with an infinite lease time.
`TIME`	An integer time value in the range: 120-5256000.

`[no] address deny|IPADDR`

Description

Specify the IP address that will be associated with this specific host configuration.

In addition, it is also possible to prohibit the specific host to not be served by this DHCP server.

Options

Option	Description
`no`	In this case it is equivalent do the deny parameter.
`deny`	Configure the host entry to deny the client to receive an address from this DHCP server.

continued

`[no] address deny|IPADDR`

Option	Description
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

continues on next page

`[no] bootfile FILENAME`

Description

Configure the boot filename (as stored on the TFTP server).

The bootfile name is typically passed within the file field of a BOOTP/DHCP message, but is instead sent as DHCP option 67 if option overloading applies or if the client has requested DHCP option 67.

If this value is configured it will override any globally configured value.

Options

Option	Description
`no`	Disable bootfile on host level.
`FILENAME`	String definition of the file name.

`[no] tftp-server [IPADDR|SERVERNAME]`

Description

Set the domain name or IP address of the boot server.

The TFTP server address to be handed out on DHCP requests. Configuring this setting overrides the configuration made on the top-level for dhcp-server.

> **NOTE**
>
> If this value is configured it will override any globally configured value.

Options

Option	Description
`no`	Remove any configured tftp-server.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1. Set the IP address of the server in both the option 66 field and next server address siaddr.
`SERVERNAME`	String definition of the tftp server FQDN. Set the domain name of the server. This is typically passed within the sname field of a BOOTP/DHCP message, but is instead sent as DHCP option 66 if option overloading applies or if the client has requested DHCP option 66.

`[no] hostname HOSTNAME`

Description

Specify the hostname to be included in the DHCP offer, DHCP option 12.

Options

Option	Description
`no`	Remove any configured hostname and disable this option.
`HOSTNAME`	String representation of the host name to be passed in the offer, e.g. “doorcamera”.

continues on next page

`[no] preempt`

Description

Enable or disable lease preemption. If enabled, it allows for the lease to be handed out to a new client if they both share the same client-id (option 61). Before the lease is handed out, a ping-check will be performed, to verify if the address is in use.

Options

Option	Description
`no`	Disable lease preemption for the current host entry.

`[no] gateway IPADDR`

Description

Define a gateway IP address to be included in the DHCP offer, DHCP option 3. It is possible to define one IP address to be used in the DHCP offer.

If no gateway IP address have been specified, the subnet or globally configured gateway IP address will be used in its place.

Options

Option	Description
`no`	Remove any configured gateway IP address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] name-server IPADDR[,IPADDR]`

Description

Specify system domain name servers (DNS) to be used in DHCP offers to clients, DHCP option 6. It is possible to configure two different IP addresses to be used in the offer.

If no name servers have been specified, the subnet or globally configured name servers will be used in its place.

continues on next page

continued

`[no] name-server IPADDR[,IPADDR]`

Options

Option	Description
`no`	Remove all specified name server DHCP options.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] ntp-server IPADDR[,IPADDR]`

Description

Specify optional NTP servers (DHCP option 42) to include in the DHCP offer. Up to two NTP servers can be specified.

Options

Option	Description
`no`	Disable the NTP server option or remove a specific server address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] log-server IPADDR[,IPADDR]`

Description

Specify optional log servers (DHCP option 7) to include in the DHCP offer. Up to two log servers can be specified.

Options

Option	Description
`no`	Disable the log server option or remove a specific server address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] route <NETWORK/LEN GATEWAY>`

Description

Specify optional classless route(s) assigned to this specific host. Routes defined here will be sent as DHCP options 121 and 249.

Options

Option	Description
`no`	Remove a specific route or all routes.
`NETWORK`	Network IP address in standard quad-dotted notation, e.g. 192.168.1.0.
`LEN`	Prefix length in CIDR notation (/24).
`GATEWAY`	Gateway IP address in standard quad-dotted notation, e.g. 192.168.1.1.

© 2026 Westermo Network Technologies AB

`[no] domain ADDR`

Description

Specify a domain name to be used in the DHCP offer. A single domain name option can be specified.

Options

Option	Description
`no`	Disable the domain option.
`ADDR`	Name of the domain (some.domain.tld).

`[no] match mac|clientid|option82`

Description

Specify what the configured host entry will match against. This setting is mandatory for static host entries, since it must have something to match against.

A match type can be specified as mac, clientid or option82:

MAC:

[no] match mac MACADDR

Client ID:

[no] match clientid hex|string CLIENTID

Option-82:

[no] match option82 remote-id hex|string REMOTEID | circuit-id hex|string CIRCUITID

Multiple match rules for a single host entry are supported, but `clientid` and `mac` **cannot** be used at the same time.

Options

Option	Description
`no`	Remove all or a specific match entry.
`MACADDR`	MAC address specified in colon notation `00:07:7c:c0:ff:ee`. Wildcard matching is allowed, e.g. `00:07:7c:*` to match the OUI `00:07:7c`.
`hex`	Indicates that the subsequent value is provided in hexadecimal form. The following syntax forms are valid: `0x1a2b`, `1a:2b`, `1a2b`.
`string`	Indicates that the subsequent value is provided in string format.

Subnet Settings

Subnet is a sub-context in the DHCP server configuration:

```
example:/#> configure
example:/config/#> dhcp-server
example:/config/dhcp-server/#> subnet 192.168.1.0/24
```

```
Created new DHCP server for subnet 192.168.1.0.
example:/config/dhcp-server/subnet-192.168.1.0/#>
```

`[no] gateway IPADDR`

Description

Define a gateway IP address to be included in the DHCP offer, DHCP option 3. It is possible to define one IP address to be used in the DHCP offer.

If no gateway IP address have been specified the globally configured gateway IP address will be used in its place.

Options

Option	Description
`no`	Remove any configured gateway IP address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] name-server IPADDR[,IPADDR]`

Description

Specify system domain name servers (DNS) to be used in DHCP offers to clients, DHCP option 6. It is possible to configure two different IP addresses to be used in the offer.

If no name servers have been specified the globally configured name servers will be used in its place.

Options

Option	Description
`no`	Remove all specified name server DHCP options.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] ntp-server IPADDR[,IPADDR]`

Description

Specify optional NTP servers (DHCP option 42) to include in the DHCP offer. Up to two NTP servers can be specified.

Options

Option	Description
`no`	Disable the NTP server option or remove a specific server address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] log-server IPADDR[,IPADDR]`

Description

Specify optional log servers (DHCP option 7) to include in the DHCP offer. Up to two log servers can be specified.

Options

Option	Description
`no`	Disable the log server option or remove a specific server address.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] route <NETWORK/LEN GATEWAY>`

Description

Specify optional classless route(s) assigned to DHCP clients in this subnet. Routes defined here will be sent as DHCP options 121 and 249.

Options

Option	Description
`no`	Remove a specific route or all routes.
`NETWORK`	Network IP address in standard quad-dotted notation, e.g. 192.168.1.0.
`LEN`	Prefix length in CIDR notation (/24).
`GATEWAY`	Gateway IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] domain ADDR`

Description

Specify a domain name to be used in the DHCP offer. A single domain name option can be specified.

Options

Option	Description
`no`	Disable the domain option.
`ADDR`	Name of the domain (some.domain.tld).

`[no] lease-time infinite|TIME`

Description

Specify the lease time in seconds, to be used for addresses handed out to DHCP clients.

Options

Option	Description
`no`	Disable host specific lease time, default to use the subnet lease time.

continues on next page

continued

`[no] lease-time infinite|TIME`

Option	Description
`infinite`	Specify the lease to be handed out with an infinite lease time.
`TIME`	An integer time value in the range: 120-5256000.

`[no] netmask NETMASK`

Description

Specify the netmask for the subnet to serve.

Options

Option	Description
`no`	Reset the netmask to its default value.
`NETMASK`	Size of the net in quad-dotted format, e.g. 255.255.255.0.

`[no] pool IPADDR_START NUM|IPADDR_END`

Description

Specify the IP address pool from which the DHCP server will handle out leases from.

The range can be defined between two IP addresses: `pool 192.168.1.50 192.168.1.99`. In addition, the range can be specified with an IP address and a number of addresses based on the address: `pool 192.168.1.50 50`.

Options

Option	Description
`no`	Disable to pool configuration, only use static host entries.
`IPADDR_START`	IP address in standard quad-dotted notation, e.g. 192.168.1.1, defines the first address available in the pool.
`NUM`	Integer value for how many dynamic addresses to be available in the pool starting from IPADDR_START specified.
`IPADDR_END`	IP address in standard quad-dotted notation, e.g. 192.168.1.1, defines the last address available in the pool.

© 2026 Westermo Network Technologies AB

2.8.3. Domain Name Service – DNS

Introduction

Most users find it is easier to refer to Internet hosts using domain names than using IP addresses. For example, http://www.example.com[1] is easier to remember than http://93.184.216.119[2].

To facilitate the use of the Domain Name System (DNS), configuration of up to two DNS server entries are supported. It is also possible to set a default domain search path. These settings can also be acquired by DHCP clients.

Using domain names on a switch is also convenient, e.g., when setting up ping triggers, VPN peers or when troubleshooting with tools such as ping or traceroute. It is also convenient to communicate with the switch using domain names. When the switch acquires its IP address dynamically (via DHCP or PPP) from an ISP, maintaining the DNS server entry is cumbersome. To that end the system includes support for dynamic DNS (DDNS). With DDNS enabled, the switch updates its DNS server entry automatically when acquiring a new IP address.

Examples of supported DDNS providers are:
- DYN: http://www.dyndns.org[3]
- freedns: http://freedns.afraid.org[4]
- no-ip: http://www.no-ip.com[5]

See the CLI or Web online help for a more up-to-date list.

> **NOTE**
>
> The DNS server in WeOS will reply with RCODE 5 (Refused) when a request is received for a domain that is not configured in the system. On many clients, including WeOS, this will result in the client not continuing to query the same server with other domains from the client's domain search path, which can lead to failed name resolution if domains in the search path are not configured in the server system.
>
> In practice, this means that if a WeOS name server is used, care must be taken when configuring the domain search path on clients using that server, to ensure that every domain in the search path is configured in the server system. This can be achieved by configuring the static domain search path correctly, considering the lists of requested options for DHCP leases, and managing the static host entries that the server can resolve.

Proxy DNS

The system can act as a DNS proxy server. When enabled, the device responds to DNS queries for known hosts, which are either:
- statically added by using the `host` command, or
- hosts for which this unit acts as DHCP server, see also the `show dhcp-clients` command

As DNS proxy, the device also acts as a caching DNS forwarder; DNS queries of unknown hosts are forwarded to the unit's own DNS server and the answer is cached for fast response of subsequent requests for the same host. Domain specific overrides may be set as domain forwarding rules.

[1] http://www.example.com
[2] http://93.184.216.119
[3] http://www.dyndns.org
[4] http://freedns.afraid.org
[5] http://www.no-ip.com

Conditional Domain Forwarding

The unit can also be configured to perform conditional forwarding based on the domain name in the DNS query. By specifying a domain name and the IP(v4) address of the DNS server to which queries should be forwarded, queries for that domain and all its sub-domains will be forwarded to the specified DNS server.

Up to 1024 rules may be specified. Multiple rules may have the same domain name to provide DNS server redundancy. If a sub-domain rule is specified, the sub-domain rule will have precedence over the domain rule.

For example, consider the following rules:

- domain-forward example.net 10.10.10.10
- domain-forward sub1.example.net 20.20.20.20

With these two rules, the unit will forward queries for www.sub1.example.net[1] to address 20.20.20.20, but queries for www.example.net[2] to address 10.10.10.10.

Configuration

The DNS is a top-level configuration context in the CLI:

```
example:/#> configure
example:/config/#> dns
example:/config/dns/#>
```

`[no] name-server [IPADDR[,IPADDR]]`

Description

Set domain name server(s).

`[no] search [some.domain.tld]`

Description

Set search path list for host name lookups.

`[no] host [HOSTNAME IPADDR]`

Description

Manage entries in the local host table.

`[no] server`

Description

Enable or disable DNS server

In addition to the static commands above, other parts of the system configuration impact the behaviour of DNS name resolution. These include:

- System hostname and domain name, see System Settings

[1] http://www.sub1.example.net
[2] http://www.example.net

- DHCP leases (specifically DHCP options 6, 12 and 15), as described in the Interface settings documentation.

Server configuration

Server is a sub-context in the DNS configuration:

```
example:/#> configure
example:/config/#> dns
example:/config/dns/#> server
```

`[no] enable`

Description

Enable or disable DNS server, the configuration will remain dormant while disabled.

Options

Option	Description
`no`	Disable the DNS server.

`[no] listen IFACE [IFACE ...]`

Description

Interfaces on which to allow server queries.

Options

Option	Description
`no`	Remove all listening ifaces.

`[no] domain-forward <FQDN> <IPADDR>`

Description

Forward queries for a domain to another name server.

Options

Option	Description
`FQDN`	A fully qualified domain name.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] reject-AAAA`

Description

Reject DNS AAAA resolution requests.

continues on next page

continued

`[no] reject-AAAA`

Options

Option	Description
`no`	Disables the immediate rejection of AAAA request.

2.8.4. Network Time Protocol – NTP

Introduction

Time is a tricky concept. For most switching and routing operations having the correct system time is not required. However, for OpenVPN, IPsec, logging, and a few other functions time is of crucial importance.

System time is saved to, and restored from, the Real-Time Clock (RTC). At system bootstrap time is restored from RTC, which may fail if the system has experience power loss for an extended period of time. When the system shuts dow, or reboots, the RTC is updated with the current time.

Time and date can be set manually from the CLI admin-exec context using the `date` command:

```
example:/#> date
Wed Jun 19 12:52:09 UTC 2019
example:/#> help date
Syntax:
    date [[YYYY-MM-DD] hh:mm[:ss]]
Usage:
    System date & time
```

Another option is to enable the system NTP client and set up at least one valid time server. NTP adjusts for clock drift but may sometimes have trouble setting the initial time, if current time differs too much, in which case the `date` command is useful to help push things along.

Timezone

The time zone is the offset from Universal Coordinated Time (UTC), which is stored in the RTC. If time on your system seems off this is usually what you want to change, unless your site keeps all servers, routers, and switches on UTC.

The time zone setting is available in the system configuration context of the CLI:

```
example:/#> configure
example:/config/#> system
example:/config/system/#>
```

`[no] timezone [TIMEZONE [no-dst]]`

Description

Set system time zone.

Try `show timezone keyword` to list matching timezones for your region. E.g.,

An optional ‘no-dst’ can be appended after timezone to specify that the timezone Dayligh Saving Time isn’t used. Note that this also affects historical timezone information.

When setting the timezone using the ‘Etc/GMT+/-offset’ notation, please take into account that the unit is consistent with POSIX which has positive signs west of Greenwich, while many people expect positive signs east of Greenwich. E.g., ‘Etc/GMT+4’ corresponds to 4 hours behind UT (i.e. west of Greenwich) even though many people would expect it to mean 4 hours ahead of UT (i.e. east of Greenwich).

continues on next page

continued

`[no] timezone [TIMEZONE [no-dst]]`

Options

Option	Description
`[no] no-dst`	Disable Daylight Saving time for the timezone
`no`	Reset system time zone to its default: UTC

`list-timezones`

Description

List all available time zone codes

NTP Client

When configured as an NTP client, the device synchronises with all configured NTP peers. The peers are continuously evaluated for best offset, delay and jitter.

> **NOTE**
>
> Before the client is synchronised with its peers, it first asks for a preliminary date/time, which might differ some seconds, or even minutes, from the actual current time. This time is set as the initial system time.

All subsequent responses from the peer are treated as adjustments of the clock frequency in order to edge closer to the time provided from peer.

NTP cannot change the system time when there is a bug time difference, instead it sets time only once, initially, and then simply adjusts that time. If contact with peers cannot be established, preliminary time will not be set, and system is left with either the time restored from the RTC, or that which is set according to the UNIX epoch.

When multiple peers are set up, preference can be set by using the `weight` peer option. This option has the highest priority, regardless of offset, delay and jitter that other servers migh have. Meaning, a server with poor offset, delay and jitter can still be chosen when given a higher weight. By default, all servers have a weight of 1, which means NTP chooses the best peer.

The time obtained from an NTP server is always in UTC, so the time zone must be configured in order to get correct system time.

NTP Server

NTP server can act as a client (1) to an upstream server, with which it synchronises, or be a stand-alone server (2), distributing its own system time, regardless of how accurate it is.

In both cases NTP server will distribute the time to its downstream peers, but the accuracy may differ significantly depending on the origin of time.

Regardless of the configured time zone on NTP server, distributed time is in UTC, per RFC, therefore clients must configure timezone in order to get correct system time.

DHCP Clients

The DHCP server can be set up to distribute NTP server(s). This is can be very useful when managing a complete subnet, in which end-devices act as both DHCP and NTP clients. The setting can be global, per subnet, or even per static lease.

It is possible to have both manually configured NTP peers, and those received from a DHCP server. Time servers which are received from DHCP server have a default weight value of 1, which is not possible to change.

Configuration

NTP is a top-level configuration context in the CLI:

```
example:/#> configure
example:/config/#> ntp
example:/config/ntp/#>
```

`[no] enable`

Description

Enable or disable NTP client/server. When disabled the settings are still saved to `startup-config.cfg`. To completely deactivate the service, use `no ntp` from the main configuration context.

Options

Option	Description
`no`	Disable NTP client/server configuration.

`[no] peer [FQDN|IPADDR]`

Description

Remote NTP peer to synchronise with.

You either can use a private or a public NTP peer, as 'pool.ntp.org' for example. If unset, the NTP server will try to set NTP peer from its DHCP client. This setting opens up the peer specific settings:

Options

Option	Description
`[no] enable`	Enable or disable this peer
`[no] preferred`	Prefer this peer over other sources.
`[no] poll-interval [VALUE]`	Interval, in seconds, between updates (30-720). Default: 512. **Note:** Due to NTP protocol limitations, the poll interval is always set to the nearest *lower* power of two, e.g. 32, 64, 128, 256, 512 (The one exception is if the poll interval is set to 30 or 31, resulting in rounding *up* to 32). For example, if you set 600s, the actual interval will be 512s.

`[no] step-adjust`

Description

Configure step-adjust function.

If the time difference between NTP client and server is greater than a given 'threshold' value, the client will step its clock, instead of slowly adjusting it (slew). This can be repeated for up to 'limit' number of measurements, unless 'limit 0' is set, which would allow to step the clock infinitely.

This is not recommended for regular use, but can be necessary in situations where a network is isolated for long periods of time.

By default, step-adjust is performed by NTP client during the initial contact with NTP server, if the time difference between them is larger than 10 seconds.

Options

Option	Description
`[no] threshold`	Set the step-adjust threshold, defaults to 10.
`[no] limit`	Set the step-adjust limit, defaults to 1.
`no`	Disable step-adjust completely

`[no] server`

Description

Enable or disable the NTP server. When disabled only the client mode is active. This setting opens up the server specific settings:

`[no] listen [IFACE ...]`

Description

Interfaces on which to allow server queries (connection from a downstream client).

Options

Option	Description
`no`	Listen on all interfaces (allow queries from everywhere). This is the default behaviour.

Options

Option	Description
`no`	Disable server functionality

Status

Status of the NTP client and server can be viewed from the admin-exec context using the `show ntp` command. The alias `show sntp` also works.

`[show] ntp [verbose]`

Description

Show NTP client and server status.

When NTP is running, displays the process ID, client synchronisation sources, and server configuration.

Options

Option	Description
`verbose`	Show extended information about the synchronisation sources, including an explanation of the column headers and status indicators.

Examples:

- `show ntp` Show NTP client and server status
- `show ntp verbose` Show extended NTP status with field descriptions

```
example:/#> show ntp
NTP Client/Server running as PID: 587


CLIENT
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* ntp1.example.com              1   9   377   504   -562us[ -642us] +/-   11ms
^- ntp2.example.com              2  10   377   185   +890us[ +810us] +/-   45ms

SERVER
Enabled on: All interfaces
```

The CLIENT section shows the NTP peers where each line represents one configured peer. The columns are:

Column	Description
M	Mode of the source: `^` = server, `=` = peer
S	State: `*` = current best, `+` = combined, `-` = not combined, `x` = may be in error, `~` = too variable, `?` = unusable
Name/IP	Name or IP address of the NTP peer
Stratum	Stratum level of the source
Poll	Polling interval (as a power of 2, in seconds)
Reach	Reachability register (octal)
LastRx	Time since last sample was received (in seconds)
Last sample	Offset between local clock and source at last measurement

The SERVER section shows whether the NTP server functionality is enabled and, if so, on which interfaces it is listening for client queries.

```
example:/#> show ntp verbose
NTP Client/Server running as PID: 587
```

```
CLIENT

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / •- Source state '*' = current best, '+' = combined, '-' = not combined,
| /             'x' = may be in error, '~' = too variable, '?' = unusable.
||                                                 •- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.           |  xxxx = adjusted offset,
||      Log2(Polling interval) --.      |          |  yyyy = measured offset,
||                                \     |          |  zzzz = estimated error.
||                                 |    |           \
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* ntp1.example.com              1   9   377   504   -562us[ -642us] +/-   11ms
^- ntp2.example.com              2  10   377   185   +890us[ +810us] +/-   45ms

SERVER
Enabled on: All interfaces
```

The `verbose` option adds a legend above the sources table, explaining the meaning of each indicator and column.

Restart and Resync

`ntp restart`

Description

Restart the NTP daemon and trigger a resynchronisation with all configured NTP servers. This is useful when NTP peers have been changed or when the system clock needs to be forcibly re-synchronised.

If NTP is not configured, the command prints an informational message and takes no action.

`ntp resync`

Description

Alias for `ntp restart`.

```
example:/#> ntp restart
example:/#>
```

2.8.5. PTP / IEEE 1588

Introduction

Precision Time Protocol (PTP) is a protocol used to synchronise clocks in a network, which is exactly what NTP (Network Time Protocol) does, but the major difference is that PTP performs synchronisation on a sub microsecond level, offering far better accuracy compared to NTP. PTP may also be referred to in a context of IEEE 1588-2008 standard, which defines version 2 of the protocol. PTP is being used in variety of use-cases, which resulted in plethora of configuration options available to users, and, as a consequence, gave rise to several additional standards, each focusing on some specific area.

The purpose of this document is to cover the basics of PTP, without going too much into details of specific use-cases.

For example use-cases, see:
- HowTo PTP

Overview

Since PTP is but a method of synchronising clocks through network, there is a need of time-source. Usually, time-source is a device having fine clock qualities, and in a network with a few potential time-sources, there is a need to select one source with the most accurate clock which will assume the role of Grand Master (GM) clock. Selection process is made possible thanks to Best Master Clock Algorithm (BMCA), which, having analysed clock attributes from all connected time-sources, decides on which clock is the most suitable for the role of GM.

Before moving forward, it is appropriate to mention at least a few common clock types and some basic terminology.

Grandmaster (GM) - device on the network acting as a primary time source for all other devices, itself not synchronising to anyone but instead being connected to a fine time source, such as GPS.

Master - device on the network providing it's time, while at the same time synchronising with another Master or Grandmaster.

Slave - device on the network wishing to synchronise with chosen time source, be it Master or GM.

The following is a non-exhaustive list of PTP clock types.

- Ordinary Clock (OC): station on the network that may act as either GM or Slave. Since GM, as we already know it, distributes its time throughout network, Slave, on the contrary, is a station that wishes to synchronise to GM.
- Transparent Clock (TC): switching device on the network, itself taking no part in BMCA and having no interest in getting synchronised with GM, but capable of forwarding PTP messages with as little impact on synchronisation as possible. Transparency is achieved by compensating for the so called Residence Time (packet delay within TC) and Uplink Delay (packet transmission time between two directly connected PTP stations). Simply put, even though there may be many TCs (hops) between GM and Slave, neither GM nor Slave should notice any significant impact on synchronisation.
- Boundary Clock (BC): switching device on the network, taking part in BMCA and acting as both Slave and Master clock. Being Slave to it's upstream neighbour, it is able to synchronise to it, while at the same time being Master to it's downstream neighbour allows that naighbor to get synchronised with itself.

> **NOTE**
>
> This system is only intended to be used as transparent or boundary clock in production. Ordinary clock mode is for testing purposes only.

BMCA and GM election process

Before synchronisation takes place, suitable time source must be elected. Election process is performed by BMCA algorithm running on each node in PTP network, with the exception of TC. The so called **ANNOUNCE** messages, containing detailed information about clock qualities and issued by each node in PTP network (except TCs), are compared by BMCA. Finally, each node running BMCA issues its own resolution as to which clock has the best clock qualities, and if such resolution is identical in each node, GM clock is elected.

Synchronisation

GM clock conveys its time via timestamp placed into **SYNC** message (in the case of 1-step capable device) or into **FOLLOWUP** message (in the case of 2-step capable device), which is then forwarded to Slave clocks.

The next step is finding out path delay (in the case of end-to-end configuration) or peer delay (in the case of peer-to-peer configuration). Peer delay is calculated between two neighboring stations while path delay is calculated between two ends, i.e. GM and Slave, and the former has the advantage of being more precise. See Table 2. This delay is needed in order to compensate for packet transmission time from GM to Slave.

Finally, when Slave clock receives **SYNC** message and calculates peer/path delay, it may adjust its local clock to that of GM. Synchronisation is a process that goes on *ad infinitum*, since clocks have tendency to drift over time.

It should be noted that PTP ports have different states associated with them, which might change over time. For example, ports fasing GM (these do receive **SYNC** messages on ingress) remain in **MASTER** state, while ports facing Slaves (these never receive **SYNC** messages on ingress) remain in **SLAVE** state. Please refer to Table 4 for a non-exhaustive list of port states.

Synchronisation mechanism is quite complex and depends on numerous details, but the intention of this document is to explain general concept only, leaving those details behind. For an in-depth insight into how PTP works on a protocol level, readers should consider studying IEEE 1588 standard.

Message	Timestamped	Description
ANNOUNCE	No	Used to announce the presence and attributes of the local clock to the network for BMCA selection.
SYNC	Yes	Used by a master clock to convey time.
DELAY REQUEST	Yes	Sent by a slave clock to calculate link delay between a master and slave. Used in the end-to-end delay mechanism.
DELAY RESPONSE	No	Sent by a master clock to a slave clock as a response to DELAY REQUEST for the end-to-end delay mechanism.
PEER DELAY REQUEST	Yes	Used to calculate the delay of a link to incoming slave port. Used in the peer-to-peer delay mechanism.

Message	Timestamped	Description
PEER DELAY RESPONSE	Yes	Response to a request by a slave port. Used in the peer-to-peer delay mechanism.

Table 1: Some common PTP messages

Delay Mechanism	Description
End-to-End	Path delay is calculated from grandmaster to slave clock.
Peer-to-Peer	Path delay is calculated between each pair of clocks in the path between grandmaster and slave clocks.

Table 2: Delay Mechanisms

Timestamping Method	Description
software	Timestamps are generated in software and uses 2-step operation (for testing only).
twostep	Timestamps are generated in hardware and uses 2-step operation.
onestep	Timestamps are generated in hardware and uses 1-step operation.

Table 3: Timestamping Methods

Port States	Description
INITIALIZING	PTP is initialising hardware and communication on this port.
FAULTY	This port is currently experiencing a FAULT and will not transmit PTP messages.
LISTENING	This port is waiting to receive an announce packet from a master.
MASTER	This port is in a master state and is transmitting/forwarding synchronisation information.
UNCALIBRATED	This port has selected a master but is not currently synchronising. **Note that this state is expected on transparent clocks which do not synchronise their clock.**
SLAVE	This port is synchronising to its selected master port.

Table 4: PTP port states. This is not a complete listing or description, please refer to IEEE 1588 standard for a full description.

2-step to 1-step conversion

A 1-step TC automatically converts 2-step **SYNC**/**FOLLOW_UP** to 1-step **SYNC** (timestamping modes `onestep`). The TwoStepFlag bit identifies a **SYNC** message as 2-step, at which point it waits for the **FOLLOW_UP** to get the originTimestamp, places that into the **SYNC**, and sends it as a 1-step **SYNC**.

Simple Use-Case examples

PTP may be employed in networks of different topologies, such as tree or ring. Figure 1 exemplifies network implementing tree topology, where TC acts as a port expander, allowing more devices to be connected to PTP network.

Figure 2 depicts ring topology formed by TCs. Besides allowing more devices to be connected to PTP network, this topology additionally provides failover and is therefore to some extent forgiving with regard to devices that may occasionally fail in one way or another. Note that in order to prevent loops one TC in PTP network will block one of its ring ports.

```
                                                  .-------.
                                                  |       |
                                     +----------+s OC   |
                                     |          |       |
                                     |          '---+---'
                        .-------.       .---+---.      .-------.
                        |       |       |   m   |      |       |
                        |  GM  m+-------+s TC  m+------+s OC   |
                        |       |       |   m   |      |       |
                        '-------'       '---+---'      '---+---'
                                     |          .-------.
                                     |          |       |
                                     +----------+s OC   |
                                                |       |
                                                '-------'
```

Figure 1: Transparent clock as a port expander. The letters ‘m’ and ‘s’ are denoted to represent MASTER and SLAVE ports respectively.

```
                                  .-------.
                                  |       |
                       +----------+s TC  m+----------+
                       |          |       |          |
                       |          '---+---'          |
.-------.       .---+---.                      .---+---.      .-------.
|       |       |   m   |                      |   b   |      |       |
|  GM  m+-------+s TC   |                      |  TC  m+ -----+  OC   |
|       |       |   m   |                      |   s   |      |       |
'-------'       '---+---'                      '---+---'      '-------'
                       |          .-------.          |
                       |          |       |          |
                       +----------+s TC  m|----------+
                                  |       |
                                  '-------'
```

Figure 1: PTP Ring configuration. Let ‘b’ denote a port which is currently blocking PTP traffic. The letters ‘m’ and ‘s’ are denoted to represent MASTER and SLAVE ports respectively.

Global PTP Configuration

```
example:/#> configure
example:/config/#> ptp
example:/config/ptp/#>
```

`[no] enable`

Description

Activate or deactivate PTP

`[no] minor_version [0|1]`

Description

PTP Minor Version

Choose between IEEE 1588-2008 (2.0) and IEEE 1588-2019 (2.1) frame formats (default 2.0), by controlling minor version.

These two frame formats are compatible, however, some older equipment might discard 2.1 frame format due to HW limitation.

`[no|show] clock [TYPE]`

Description

Type of PTP Clock

There are several different types of PTP clocks, implementing different functions in a PTP network. An ordinary clock may act as a time source for other clocks, or synchronise against another ordinary clock.

A transparent clock is used to transparently switch PTP messages, updating timestamps with a correction to adjust for its residence time.

A boundary clock is used to synchronise the switch's own clock and transmit new PTP Sync messages downstream, creating a boundary where changes upstream are isolated from the downstream.

> **NOTE**
>
> The support for an ordinary clock on this system is for testing purposes only.

`[no] description [name STRING] [location STRING]`

Description

User defined description

`[no|show] profile [TYPE]`

Description

PTP Profile

PTP has many options and configurable attributes. A profile is a collection of default values or choices of these options and attributes.

Each profile is specified so that a system can be configured without further user configuration of the profile specific settings.

Specifically, default refers to the default configuration for this system if nothing else is chosen. Power-v1 and power-v2 refer to the 2011 and 2017 editions of the PTP power profile IEEE C37.238 respectively.

`[no] show`

Description

Show a summary of PTP settings

Clock Configuration

Below is a list of all clock settings available in WeOS. Note that some options are specific to a certain type of clock and will be unavailable in other clock types.

`[no] client-only`

Description

Force the local clock to be a client clock

A client clock will not be part of the Best Master Clock Algorithm (BMCA).

`[no] priority1 [NUM]`

Description

Priority1 attribute of the local clock

This attribute is the most important attribute for chosing a master clock in the Best Master Clock Algorithm (BMCA).

`[no] priority2 [NUM]`

Description

Priority2 attribute of the local clock

This attribute is used to determine a grand-master given that they have equal priority1, clock quality and offset variance.

`[no] ports [PORT | PORTA PORTB | PORTA..PORTB]`

Description

Configure Clock ports

Port configuration depends on chosen clock type. Transparent clock requires at least 2 ports to be configured, while Ordinary clock requires only 1.

`[no] show`

Description

Show a summary of PTP clock settings

Profile Configuration

Since PTP has vast range of options many of which are incompatible with each other, a need for some kind of standard arose and took shape of the so called profile. Profile is a set of predefined options, compatible with each other, providing out-of-the-box configuration. Devices configured with the same profile are said to work without the need of any additional configuration.

WeOS in combination with some specific hardware (to be checked in release notes), supports the following two IEEE standards for Power System Applications: IEEE C37.238-2011[1] IEEE C37.238-2017[2]

[1] https://standards.ieee.org/ieee/C37.238/4609/
[2] https://standards.ieee.org/ieee/C37.238/6650/

For simplicity, IEEE C37.238-2011 standard is referred to as Power-v1 in WeOS, while IEEE C37.238-2017 as Power-v2.

`[no] delay endtoend|peertopeer`

Description

Delay Mechanism

Delay mechanism is the method by which PTP calculates the delay on the path between synchronising clocks. This can be done by calculating the delay on each hop or across the full path. These modes are called end-to-end and peer-to-peer respectively.

Peer-to-peer mode is more resistant to changes in the network, while end-to-end mode may allow for non PTP devices in the network.

`[no] domain [NUM]`

Description

Domain number of the local clock

Domain number specifies which clock domain this clock belongs too. Local clock will synchronise to a grand-master belonging to the same domain.

Available range: [0 - 255] Reserved range: [128 - 255] Standard range: [0 - 127]

Domain number 0 is the default domain in profile C37.238-2011 (power-v1).

Domain number 254 is only allowed to be used in profile IEEE C37.238-2011 (power-v2), as an exception, and is that profile's default domain.

`[no] egress-vlan id [NUM] prio [NUM]`

Description

VLAN (8021Q) settings for PTP packets.

VLAN settings for PTP packets on egress. PTP packets will be transmitted with the configured VLAN priority and id.

`[no] timestamping software|twostep|onestep`

Description

Time Stamping method

Timestamps can be generated either in hardware or in software. Software timestamps are generated before being passed to hardware for transmission; increasing jitter and path-delay. Hardware timestamping is done close to the wire, providing much better accuracy.

There are two hardware timestamping methods:

- two-step - first step is taking the timestamp when sending SYNC packet, and second step is sending this timestamp in a FOLLOWUP packet.

continues on next page

continued

`[no] timestamping software|twostep|onestep`

- one-step - taking timestamp and sending it in a SYNC message is done in one step. And if hardware supports it, peer-to-peer messages are also sent in one step.

Software timestamping is always two-step.

Options

Option	Description
`software`	Timestamps are generated in software and uses 2-step operation (for testing only).
`twostep`	Timestamps are generated in hardware and uses 2-step operation.
`onestep`	Timestamps are generated in hardware and uses 1-step operation.

`[no] logAnnounceInterval [INTERVAL]`

Description

Frequency of 'Announce' messages sent by an OC/BC

0=1s interval, 1=2s, 2=4s, 3=8s, −1=0.5s, −2=0.25. As defined in IEEE 1588.

`[no] step_threshold [NUM]`

Description

Offset at which an OC/BC steps the clock

A value of 0 means it will never step beyond the initial synchronisation. The value 0.00002 (20 microseconds) is used for the initial synchronisation

Default: 0

Range: 0-1 (floating point)

`[no] show`

Description

Show a summary of PTP profile settings

Status Overview

Status of local PTP clock and PTP network may be obtaine by commands available inside ptp context of adminexec.

```
example:/#> ptp
example:/ptp/#>
```

`[no] network [clock-ID ID] [boundary-hop NUM]`

Description

Show PTP network summary

Query PTP network for management responses. By default, local boundary domain is queried and summarised. If an ID is specified, a more detailed summary of only that clock's response is displayed.

`[no] query [boundary-hop NUM] DATASET1 DATASET2..`

Description

Query network for PTP management response.

The Precision Time Protocol (PTP), with reference to IEEE 1588 has multitude of parameters which may be of interest for configuration and debugging purposes.

By default, this command sends management packets to the local clock, which responds with data sets as specified by the 1588 standard. If boundary hop is specified, query the network.

`[no] show`

Description

Show summary of PTP clock and ports.

`show` command displays summary of the local clock and and overview of directly connected neighbours.

```
tc:/#> show ptp
Clock
ID                  : 00077c.fffe.785325
Type                : TC P2P
Domain              : 0
Mean Path delay     : 0.0
Master Offset       : -7075504342646.0
Priority1           : 254
Priority2           : 128
Product             : Westermo;RedFox-5728-E-F16G-T12G-HVHV;00:07:7c:78:53:20
Distance            : 1
Description         : tcbridge;room2

Ports
ID                      INTERFACE         STATE           PEER DELAY
00077c.fffe.785325-1    eth5              MASTER          96
00077c.fffe.785325-2    eth11             UNCALIBRATED    90
00077c.fffe.785325-3    eth25             MASTER          8
```

`show network` command queries PTP network for PTP management responses and aggregates a summary. Each clock will be identifiable via its ID. This ID may also be used to provide a summary of a remote clock. Here's how this command may look like:
`show network clock-id 0011b4.fffe.2ed8c7`. If the clock is behind one or more boundary clocks, the option `boundary-hops` may also be specified.

```
tc:/#> show ptp network

Clocks
ID                      TYPE     DOMAIN   MASTER OFFSET    DESCRIPTION
00077c.fffe.785325      TC P2P   0        -775503935118.0  tcbridge;room2
00077c.fffe.7bf4c5      OC       0        0.0              grandmaster;room1
00077c.fffe.78530b      OC       0        1.0              slave1;room3
00077c.fffe.784fda      OC       0        1.0              slave2;room3
```

```
tc:/#> show ptp network  clock-id 00077c.fffe.7bf4c5
Clock
ID                  : 00077c.fffe.7bf4c5
Type                : OC
Domain              : 0
Mean Path delay     : 0.0
Master Offset       : 0.0
Priority1           : 128
Priority2           : 128
Product             : Westermo;RedFox-5728-E-F16G-T12G-HVHV;00:07:7c:7b:f4:c0
Distance            : 0
Description         : grandmaster;room1

Ports
ID                      STATE          PEER DELAY
00077c.fffe.7bf4c5-1    MASTER         95
```

Lastly, the `query` command embodies utility for showing PTP management responses in a more raw manner. It is useful if one knows exactly what one is looking for, but may require more comprehensive knowledge of PTP to be correctly understood.

```
tc:/#> show ptp query
TRANSPARENT_CLOCK_DEFAULT
00077c.fffe.785325-0
  clockIdentity                           00077c.fffe.785325
  numberPorts                             3
  delayMechanism                          2
  primaryDomain                           0
TRANSPARENT_CLOCK_PORT
00077c.fffe.785325-1
  portIdentity                            00077c.fffe.785325-1
  faultyFlag                              0
  logMinPdelayReqInterval                 0
  peerMeanPathDelay                       92
00077c.fffe.785325-2
  portIdentity                            00077c.fffe.785325-2
  faultyFlag                              0
  logMinPdelayReqInterval                 0
  peerMeanPathDelay                       90
00077c.fffe.785325-3
  portIdentity                            00077c.fffe.785325-3
  faultyFlag                              0
  logMinPdelayReqInterval                 0
  peerMeanPathDelay                       8
```

2.9. PKI

2.9.1. Certificate and Key Management and Usage

This document presents the WeOS certificates repository its usage and how to manage it with examples. Certificates and keys can be managed from both CLI and Web interfaces. When using CLI use the `/pki/#>` and `/pki/cert/#>` for management.

> **SECURITY**
>
> Improper timekeeping or time-related attacks can lead to various security vulnerabilities, such as expired certificates being considered valid or incorrect timestamps on logs and transactions. Attackers may exploit these vulnerabilities to bypass security mechanisms, disrupt services, or manipulate data integrity. Ensuring accurate and secure time synchronisation is critical to maintaining the reliability and security of the PKI system.

What is the Certificate Repository?

The WeOS certificate repository holds cryptographic certificates and keys used by WeOS services utilising asymmetric and symmetric cryptography, all-together called PKI Artifacts. An example of such can be found in the WeOS Web server, which by default uses a unique, self-signed certificate (and associated private key) for its HTTPS service.

It is possible to import your own certificates and associated private keys to be used for Web/HTTPS or other services, e.g. tunnels. Certificate or OpenVPN keys may also be generated directly by WeOS. Lastly certificates could enrolled from external servers.

WeOS includes a set of default trusted CAs, used by certain WeOS client services such as DDNS, when contacting certificate based Internet services.

PKI Artifacts

The following groups of PKI artifacts can be stored in the system:

- **Local certificates**: These are end-entity certificates that prove the validity of the system's keys. The system inherently trusts these certificates. Associated private keys should be present in the system's secure storage. These certificates can be imported, generated, or enrolled. End-entity certificates must have their respective chain certificates existing in the Trusted Local CA Store (see below).
- **Trusted Global Root CA Store**: This is a built-in Mozilla Root Store of remote CA certificates that comes with the system. The store is updated with each release of the system. The content of this store cannot be changed in between releases, but some applications within the system can choose to distrust the store. More about it here.
- **Trusted Local CA Store**: This is a store of remote CA certificates that is editable by the user. The content of this store can be changed between releases. A selection of certificates from the Global Root Store can be copied into this one to limit the trust scope.
- **OpenVPN Pre-Shared Keys (keys)**: These are static pre-shared keys shared between the client and server to secure communication without requiring a full public key infrastructure (PKI).

Each of these groups has its own namespace. Thus, it is possible to have a local end-entity certificate, a CA certificate, and an OpenVPN key all holding the same label. What a label is, is explained in the next section.

Labels

A label is used to identify a PKI artifact. Labels are used when configuring applications that use PKI credentials. In a given namespace, a label uniquely represents a credential or a bundle of credentials in the case of the Local CA store.

Certificate Chaining

Certificate chaining is a process in Public Key Infrastructure (PKI) where a trust relationship is established from an end-entity certificate (like a user's or server's certificate) up to a trusted root certificate authority (CA). Each certificate in the chain is signed by the one above it, forming a "chain of trust."

```
  +----------+                    +---------+
  | Root1-CA | root1-ca.pem          | Root2-CA|
  |          |                    |         |
  +--+-------+                    +--+------+
     |                               |
     | Root1 signs Int1              | Root2 signs Int2
     |                               |
  +--v-------+                    +--v------+
  | Int1-CA  |                    | Int2-CA |
  |          |                    |         |
  +--+-------+                    +-------+-+
     |                                     |
     | Int1 signs alice.pem                | Int2 signs bob.pem
     |                                     |
+----v---+  alice.pem + chain1          +-v------+
|        +----------------------------->        |
| Alice  |          bob.pem + chain2    | Bob    |
|        <-----------------------------+        |
+--------+                              +--------+
```

Figure 1: Certificate chaining.

In the Figure 1, Root1-CA and Root2-CA are trusted root authorities (the trust anchors). Int1-CA and Int2-CA are intermediate CAs, each signed by their respective root. Alice and Bob have certificates (*alice.pem* and *bob.pem*) signed by the intermediates.

Alice imported *root2-ca.pem*, Root2-CA's self-signed certificate. This way she gave trust in all certificates whose chain if trusts ends in *root2-ca.pem*. Bob did the same with *root1-ca.pem*.

When Alice presents her certificate *alice.pem*, she also provides the chain1 (*int1-ca.pem* + *root1-ca.pem*), allowing others to verify her identity by following the signatures up to Root1-CA. The same applies to Bob with Root2-CA. This chaining ensures that trust in the root CA extends to the end-entity certificates.

If Alice presents only the end-entity certificate, *alice.pem*, she assumes that Bob is in possession of Alice's chain1 (*int1-ca.pem*, *root1-ca.pem*), which is not guaranteed and is not best practice.

When configuring a service in WeOS, e.g., a tunnel, the user configures only the end-entity certificate (*alice.pem*), and the system will attempt building the chain using existing certificates from the Local CA Store. If building fails the tunnel is not created. Thus, all necessary certs must be present in the system. These can be enrolled from an external server or imported manually.

> **NOTE**
>
> It is important that local end-entity certificates have their respective chain certificates existing in the system.

Certificate bundles

Certificates can be stored as bundles under a single label in WeOS. When importing PEM or PKCS#12 files that contain concatenated certificates, WeOS keeps the certificates grouped in one bundle. The bundle can be inspected from the Web UI (it shows each element of the bundle) or with the CLI `pki cert show full` output where multiple entries share the same label. Bundles are marked with an asterisk in the CLI.

The unit's default Web certificate

The default certificate is always present in WeOS, and when it is deleted, a new one will be generated automatically at reboot.

```
example:/#> pki cert show full
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
Pub      52ff4f77 Jan 19 2038 lynx-12-34-50.local       web-default
Key      N/A      N/A         web-default               web-default
```

It is possible to import (see below) a custom Web server certificate to the repository, and configure the Web server to use that certificate.

Built-in Trusted Global Root-CAs

Included in the WeOS image is a set of trusted root CAs, based on Mozilla's list of trusted root CAs. This list of root CAs is provided for convenience, and is used by WeOS services using HTTPS to securely access Internet services. An example is the WeOS DDNS client, which utilises HTTPS to contact the DDNS provider.

The DDNS client trusts DDNS providers with a certificate issued by any of the built-in trusted root CAs by default. It is possible to configure the DDNS client to only trust certificates issued by CAs manually imported to the certificate repository.

The example below shows how to display the list of trusted CAs.

```
example:/#> pki cert show all
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
CA-auto  2b349938 Dec 31 2030 AffirmTrust C~ercial      AffirmTrust_Commercial
CA-auto  93bc0acc Dec 31 2030 AffirmTrust N~orking      AffirmTrust_Networking
CA-auto  b727005e Dec 31 2040 AffirmTrust Premium       AffirmTrust_Premium
CA-auto  ce5e74ef Jan 17 2038 Amazon Root CA 1          Amazon_Root_CA_1
CA-auto  6d41d539 May 26 2040 Amazon Root CA 2          Amazon_Root_CA_2
 ...
```

Enrollment of Certificates

Automated issuance of digital certificates is popular because it saves time and enables devices to request, obtain, and renew certificates without manual intervention, which is particularly useful in enterprise environments where a large number of devices require certificates for secure communication and authentication purposes.

The system supports Simple Certificate Enrolment Protocol (SCEP) according to RFC 8894. Enrollment is divided into two distinct phases: a *manual* phase and an *automated* phase. During the manual phase, users configure certificates and enrollment servers. In the automated phase, a background task periodically communicates with the enrollment server to request certificates. For a more in depth description of the SCEP protocol configuration see Section SCEP.

Enrollment process relies on external servers that must be accessible from the device. To achieve that other parts of the User Guide, like configuration details of the Firewall and Ethernet Ports might be relevant.

CLI Syntax

PKI exec context

Management of certificates and keys can be done in PKI exec context:

```
example:/#> pki
example:/pki/#>
```

`generate [expire <DAYS>] [modulus <BITS>] [ovpn] label <NAME>`

Description

Generate a self-signed X.509 RSA type certificate or an OpenVPN static key (PSK).

Options

Option	Description
`expire <DAYS>`	Optional certificate validity period in days. Default 365. Applicable only for self-signed certificates.
`modulus <BITS>`	Optional key size in bits. Default 1024. Applicable only for self-signed certificates.
`[ovpn]`	Flag causing generation of a static OpenVPN key instead of a x509 certificate. When selected `expire` and `modulus` are skipped.
`label NAME`	A unique label of the certificate or key to be generated.

Examples:

- Generate a self-signed certificate:

```
generate expire 365 modulus 2048 label mycert
```

- Generate an OpenVPN static key (PSK):

```
generate ovpn label mykey
```

```
import <pkcs|pem|der|ovpn> type <key|pub|ca> [password <SECRET>] [label <NAME>]
[media <LABEL>] [proto <ftp|tftp|scp|sftp|http|https>] [user <USERNAME>]
<address[:port]> <path>
```

Description

Import a certificate or key from a source (FTP/TFTP/SCP/SFTP/HTTP/HTTPS/USB). The software supports importing PKCS#12, PEM and DER certificates. The latter two require the user to provide the type: private, public or CA. This to ensure the certificate is put to proper use. You may also import OpenVPN (ovpn) static keys.

NOTE: TPM based devices support RSA keys up to 4096 bits.

continues on next page

continued

```
import <pkcs|pem|der|ovpn> type <key|pub|ca> [password <SECRET>] [label <NAME>]
[media <LABEL>] [proto <ftp|tftp|scp|sftp|http|https>] [user <USERNAME>]
<address[:port]> <path>
```

SECURITY

When importing a private key make sure the process is not intercepted during the import process, this can lead to security breaches. The private key is stored securely when device is off and decrypted on boot. Never import an untrusted key or certificate without validation, this can lead to a security breach.

Options

Option	Description
`<pkcs|pem|der|ovpn>`	Select format of PKI artifacts to be imported. • pkcs - import PKCS12 bundle. • pem - import PEM certificate, CA certificate or private key. Additional PEM/DER options: 'type public' (import as certificate), 'type ca' (import as CA certificate), and 'type private' (import as private key). • der - import DER certificate, CA certificate or private key. Same options as for 'pem'. • ovpn - import OpenVPN static key.
`type <key|pub|ca>`	Select type of PKI to be imported.
`password <SECRET>`	Password for imported certificate/key (Optional when encrypted).
`label <NAME>`	A unique label of the certificate or key to be generated.
`media <LABEL>`	Label of the media (USB) to import from. Alternative to specifying a file transfer protocol and address.
`proto <ftp|tftp|scp|sftp|http|https>`	Protocol to use. ftp, tftp, scp, sftp, http or https.
`user <USERNAME>`	Username if used by protocol.
`address[:port>`	Address (and optional port) of the server to import from.
`path`	Path on the remote side.

`remove [force] [LABEL [LABEL ...]] [HASH [HASH ...]]`

Description

Remove all or only selected PKI artifacts. Without any input arguments the command removes ALL artifacts. If a label is specified and there are many different types of PKI artifacts having the same label (e.g. key and certificate) you will be prompted before deleting each of them.

force

Description
Force removal of without any questions

LABEL

Description
Label(s) of the PKI artifact to be removed.

HASH

Description
Subject hash of a certificate to be removed.

`trigger`

Description
Trigger the pki daemon to re-read the config and send enrollment/renewal requests to the server if needed.

`show [all | full] [HASH] [LABEL]`

Description
Show the complete state of PKI in the system.
- Status of PKI daemon.
- Status of enrollment server.
- List of revocation distribution points.
- List all PKI artifacts in the system including (certificates, keys, openvpn-keys)

Options

Option	Description
`all`	Display all certificates from the Built-in Trusted Global Root-CA Store.
`full`	Display all certificates from the Trusted Local CA Store.
`HASH`	Details of selected PKI artifacts
`LABEL`	Details of selected PKI artifacts

PKI-cert exec context

There are also certificate specific commands in the cert context:

```
example:/pki/#> cert
example:/pki/cert/#>
```

`enroll <LABEL>`

Description

Enroll a certificate from an enrollment server. This topic is described in details here.

Options

Option	Description
`LABEL`	Unique label of a certificate to enroll. See here for details.

`show [all | full | HASH | LABEL]`

Description

Dump certificate information: attributes & meta data.

Options

Option	Description
`all`	Display all certificates from the preinstalled Trusted Global Root-CA Store.
`full`	Display all certificates from the Trusted Local CA Store.
`HASH`	Details of selected PKI artifacts
`LABEL`	Details of selected PKI artifacts

`remove [force] [LABEL [LABEL ...]] [HASH [HASH ...]]`

Description

Deleting certificate and keys.

Options

Option	Description
`force`	Force removal of without any questions
`LABEL`	Label(s) of the PKI artifact to be removed.
`HASH`	Subject hash of a certificate to be removed.

Examples

Import a PKCS#12 or a PEM certificate

```
example:/#> pki import pkcs password "secret string" ftp://1.2.3.4/bundle.p12
example:/#> pki import pem type public usb://remote.crt
```

Import an OpenVPN static key (PSK or TLS-AUTH)

```
example:/#> pki import ovpn ftp://1.2.3.4/ovpn.key
```

Generate an OpenVPN static key (PSK or TLS-AUTH)

```
example:/#> pki generate ovpn label NAME
```

Show all certificates or display a given label/hash

```
example:/#> pki cert show
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
Pub      52ff4f77 Jan 19 2038 lynx-12-34-50.local       web-default
```

```
example:/#> pki cert show full
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
CA-local fa27d441 Sep  5 2034 OpenXPKI Root~240902 SCEP enroll2+ca
Pub      52ff4f77 Jan 19 2038 lynx-12-34-50.local       web-default
Key      N/A      N/A         web-default               web-default
```

```
example:/#> pki cert show all
Press Ctrl-C or Q(uit) to quit viewer, Space for next page, <CR> for next line.
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
CA-auto  a94d09e5 Dec 31 2030 ACCVRAIZ1                ACCVRAIZ1
CA-auto  cd8c0d63 Jan  1 2030 N/A                      AC_RAIZ_FNMT-RCM
CA-auto  b81b93f0 Dec 20 2043 AC RAIZ FNMT—~EGUROS     AC_RAIZ_FNMT-RCM_SERVIDORES_SEGUROS
CA-auto  b433981b Aug 30 2039 ANF Secure Se~oot CA     ANF_Secure_Server_Root_CA
CA-auto  930ac5d2 Sep 22 2030 Actalis Authe~oot CA     Actalis_Authentication_Root_CA
CA-auto  2b349938 Dec 31 2030 AffirmTrust C~ercial     AffirmTrust_Commercial
CA-auto  93bc0acc Dec 31 2030 AffirmTrust N~orking     AffirmTrust_Networking
CA-auto  b727005e Dec 31 2040 AffirmTrust Premium      AffirmTrust_Premium
CA-auto  9c8dfbd4 Dec 31 2040 AffirmTrust P~um ECC     AffirmTrust_Premium_ECC
CA-auto  ce5e74ef Jan 17 2038 Amazon Root CA 1         Amazon_Root_CA_1
CA-auto  6d41d539 May 26 2040 Amazon Root CA 2         Amazon_Root_CA_2
CA-auto  8cb5ee0f May 26 2040 Amazon Root CA 3         Amazon_Root_CA_3
--More-- (17% of 10619 bytes)
```

Remove certificate and key by label/hash

Removing of certificates and keys is done from the pki exec context:

```
example:/#> pki cert show full
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
Pub      52ff4f77 Jan 19 2038 lynx-12-34-50.local       web-default
Key      N/A      N/A         web-default               web-default
example:/#> pki remove web-default
Remove certificate 'web-default', are you sure (y/N)? y
Removing certificate 'web-default'
Remove private key 'web-default', are you sure (y/N)? y
Removing private key 'web-default'
```

Use `force` parameter to avoid questions:

```
example:/#> pki cert remove force [ ... ]
```

To remove all PKI artifacts

```
example:/#> pki remove
Remove ALL PKI artifacts, are you REALLY sure (y/N)?
```

2.9.2. Simple Certificate Enrollment Protocol

Abstract

This document provides an overview of the Simple Certificate Enrollment Protocol (SCEP) as implemented in WeOS. It explains the protocol's purpose, lists supported features and limitations. Enrollment process and configuration parameters are explained.

A complete example workflow from server configuration to certificate enrollment and renewal is provided in another document: Enrollment of certificate.

Glossary

CA: Certificate Authority. An entity that issues digital certificates.

RA: Registration Authority. An entity that acts as a mediator between the user and the CA, handling requests for digital certificates.

SCEP: Simple Certificate Enrollment Protocol. A protocol for automating certificate enrollment in a PKI environment.

PKI: Public Key Infrastructure. A framework for managing digital certificates and public-key encryption.

CSR: Certificate Signing Request. A request sent from an applicant to a CA to apply for a digital certificate.

CER: Certificate Enrollment Request. The configuration representing a request to enroll a certificate.

Root-CA: The top-level Certificate Authority in a PKI hierarchy.

Intermediate-CA: A subordinate CA that is certified by the Root-CA and can issue certificates to end entities.

NDES: Network Device Enrollment Service. A Microsoft service that implements SCEP.

OpenXPKI: An open-source PKI software that supports SCEP.

SAN: Subject Alternative Name. An extension in X.509 certificates that allows additional identities to be bound to the subject of the certificate.

DN: Distinguished Name. A unique identifier for the subject or issuer of a certificate.

EKU: Extended Key Usage. An extension in X.509 certificates specifying permitted uses of the key.

Introduction

Simple Certificate Enrollment Protocol, or **SCEP**, described in RFC 8894[1], is a protocol designed to simplify and automate the process of certificate issuance and management within a Public Key Infrastructure (PKI) environment.

SCEP provides authentication for IoT devices in industrial applications. It facilitates communication between client devices and a Certificate Authority (RA/CA) to automate the issuance of digital certificates. SCEP is popular because it saves time and enables devices to request and renew certificates automatically.

Features

The following features are supported in the WeOS SCEP implementation:

[1] https://datatracker.ietf.org/doc/html/rfc8894

- **Automated Certificate Enrollment**: The device acts as a SCEP client and receives certificates automatically, reducing administrative overhead by streamlining certificate enrollment.
- **Automated Renewal of Certificate**: Starts renewal of issued certificates after a predefined threshold is reached.
- **Certificate Rollover**: Supports Intermediate-CA certificate rollover.
- **Integration with Certificate Authorities (CAs)**: Works with various CA infrastructures. The implementation of the SCEP protocol on this device has been tested with OpenXPKI and Windows NDES. The device supports renewal of certificates using both *PKCSReq* and *RenewalReq* messages.

At the end of this documentation, there is a section with links to example setups to follow when setting up the SCEP functionality.

> **NOTE**
>
> The implementation of SCEP protocol on this device is tested against OpenXPKI and Windows NDES.

> **SECURITY**
>
> It is possible to use a generic challenge password on the SCEP server; however, it is not recommended since it poses a security risk and opens up for unauthorized users to gain access.

Limitations

The system has the following limitations:

- The rollover of Root-CA certificates is not supported. Adding a new root certificate will replace the existing one. However, the rollover of Intermediate-CA certificates is supported and described here.

Enrollment process

The enrollment process consists of two phases. The first is a manual setup phase where the user configures one or more SCEP servers and certificate enrollment requests. The second phase is automatic, during which certificates are automatically enrolled and renewed.

Manual setup phase

Enrollment of a certificate requires a valid configuration of the certificate and enrollment. The setup phase consists of two steps:

- Step 1: Configuration of enrollment servers is done in the Config context:

```
example:/#> config
example:/config/#> pki
example:/config/pki/#> enroll 1
example:/config/pki/enroll-server-1/#>
```

All server parameters are explained here. After applying the configuration, the device tries to download the Root-CA and all Intermediate-CA certificates and stores them in the local CA-store. The server label can be referenced from other parts of the system, including CSR configuration.

- Step 2: Creation of enrollment requests from within the Exec context:

```
example:/#> pki
example:/pki/#> cert
example:/pki/cert/#> enroll cert1
example:/pki/cert/enroll-cert1/#>
```

A Certificate Enrollment Request (CER) includes parameters for the certificate signing request and settings related to enrollment itself. A detailed list of parameters is provided here.

Labels of pending certificates can be referenced when configuring other services (e.g., tunnels), but these services might not be fully functional until the certificate is enrolled.

Automatic enrollment/renewal phase

After a valid request and enrollment server are configured, the device automatically creates a key-pair and periodically requests a certificate from the server at a predefined interval. The device will try to enroll a certificate if it is:

- not already downloaded,
- not rejected,
- and has a valid enrollment server configured.

Typically, the CA administrator has to approve the request before a certificate can be downloaded. Upon successful enrollment, services utilising certificates (e.g., tunnels) will be reloaded or restarted to transition to the renewed certificate.

When the certificate reaches its renewal threshold, the renewal process starts. The `renewal-threshold` setting on a certificate is a percentage value that determines how much of the enrolled certificate's validity period must have elapsed before attempting to renew the certificate. The default value is 80% of the validity period. Renewal of a certificate must be completed before it expires; otherwise, it must be removed and a new certificate enrolled.

Rollover

The **CA rollover** is the process of transitioning from one Certificate Authority (CA) certificate to another. During the rollover period, both the old and new CA certificates are valid and in use. Once the transition is complete, the old CA certificate is retired, and the new CA certificate takes over entirely.

The system supports rollover of Intermediate-CA certificates. When a new Intermediate-CA is retrieved from the server, it is saved under a unique name, allowing the old certificate to coexist. Every subsequent enrollment or renewal will use the new intermediate for signing. However, both intermediates (old and new) can be used by applications, such as tunnels, when constructing certificate chains.

Rollover of Root-CA certificates is not supported. Any new Root-CA provided by the SCEP server via the *getCACert* command replaces the existing one.

Configuration

The PKI configuration consists of setting up an enrollment server to be used for one or more certificate enrollments.

Configuring an Enrollment Server

This section describes configuration of an enrollment server.

```
example:/#> configure
example:/config/#> pki
example:/config/pki/#> enroll 1
example:/config/pki/enroll-server-1#> show
Server #                        : 1
Label                           : enroll1
Description                     : HQ-server
Status                          : Enabled
URL                             : http://myurl.com:8080/scep/generic
RA-DN-Hash                      : Auto
fingerprint                     : 0542DFC5F97A4812E18FBA0CAE092E82
fingerprint-alg                 : md5sum
Protocol                        : SCEP
Cipher                          : AES128-CBC
Renew-request                   : PKCSReq
```

`[no] label <LABEL>`

Description

The label of this enrollment server. The Root-CA certificate downloaded from that server will be given this label. The label can be referenced to from other parts of the system including during certificate enrollment configuration.

Options

Option	Description
`LABEL`	This is the name of the enrollment server.
`no`	Resets label to enroll#.

`[no] description [STRING]`

Description

The certificate refers to this label, and it will be shown in lists.

Options

Option	Description
`STRING`	This is a short description of the enrollment server.
`no`	Resets Description to empty string.

`[no] enable`

Description

Enables an enrollment server. Status will show result.

Options

Option	Description
`no`	Disables an enrollment server.

`[no] url <URL>`

Description

URL to the enrollment server. Consists of scheme, sub domain, top-level domain, second-level domain, sub directory, port.

Options

Option	Description
`URL`	Url of the enrollment server e.g. `http://example.com/certsrv/mscep/mscep.dll`
`no`	Resets URL to empty string.

`[no] fingerprint <HASH>`

Description

The fingerprint of the Root-CA certificate in DER format used for integrity and authenticity check done before enrollment.

Options

Option	Description
`HASH`	The fingerprint is received from admin of enrollment server.
`no`	Resets fingerprint to empty string.

`[no] fingerprint-alg <md5|sha1|sha256>`

Description

The algorithm used to calculate the fingerprint of the Root CA certificate. SHA-256 is default as stipulated by RFC 8894.

Options

Option	Description
`<md5|sha1|sha256>`	The hashing algorithm used to calculate the fingerprint, md5sum, sha1sum, or sha256sum.
`no`	Resets fingerprint-alg to default algorithm, sha256.

`[no] protocol <scep>`

Description

Protocol to use to talk to the enrollment server.

Options

Option	Description
`[scep]`	Only SCEP is supported.
`no`	Resets Protocol to default, SCEP.

`[no] cipher <aes128-cbc|3des-cbc>`

Description

Type of symmetric encryption when data is sent to the server. AES128-CBC is default as stipulated by RFC 8894.

Options

Option	Description
`<aes128-cbc|3des-cbc>`	Encrypt of data sent to/from the server. Choose one supported by the server.
`no`	Resets Cipher to default AES128-CBC.

`[no] renew-request <pkcs-req | renewal-req>`

Description

By default, a PKCSReq message is sent when renewing a certificate. The RFC 8894 standard introduced a dedicated RenewalReq message type. However, some SCEP servers (e.g., OpenXPKI) do not support it and perform renewal using PKCSReq instead of RenewalReq. Windows NDES handles it correctly.

Options

Option	Description
`<pkcs-req | renewal-req>`	SCEP protocol command used when renewing a certificate. Choose one supported by the server.
`no`	Resets Renew-request to default PKCSReq.

`[no] ra-dn <SUBJECT-HASH>`

Description

Subject-hash of an RA certificate used for signature verification and encryption during communication with the SCEP server. The certificate must have required Key Usage: DigitalSignature, KeyEncipherment and the subject (i.e. hash) of the selected certificate must not change during rollovers events.

The default is “Auto”, which first tries to use dedicated RA certificate(s) (with CA:False flag) if received from server. Otherwise, if dedicated RA certificates are not available, it searches from the bottom of the PKI chain for a CA-certificate with required KeyUsage flags.

Options

Option	Description
`SUBJECT-HASH`	The subject-hash of the RA certificate to use.
`no`	Resets RA-DN to default (Auto).

Configuring an Certificate Enrollment Request

An enrollment request is configured in exec context.

```
example:/#> pki
example:/pki/#> cert
example:/pki/cert/#> enroll mycert
example:/pki/cert/enroll-mycert/#> show
Cert label                      : mycert
Server                          : enroll1
Password                        : SecretChallange
DN                              : CN=mycert.co.uk, C=UK
Renewal-threshold               : 80 %
SAN                             : 192.168.100.101
Keysize                         : 2048
Profile                         : tls-server
```

`[no] server <LABEL>`

Description

Label of an already configured server

Options

Option	Description
`CHALLENGE`	This is the name of the enrollment server defined earlier
`no`	Resets to empty string

`[no] password [CHALLENGE]`

Description

Challenge password obtained from SCEP server or an administrator. Ensure that it is unique.

Options

Option	Description
`CHALLENGE`	Challenge password string
`no`	Resets to empty string

`[no] renewal-threshold <THRESHOLD>`

Description

The setting determines when the system should begin attempting to renew a certificate, based on how much of the certificate's validity period has already passed.

Options

Option	Description
`THRESHOLD`	The value is specified as a percentage between 0 and 99. For example, if the threshold is set to 80, the system will start the renewal process after 80% of the certificate's lifetime has elapsed. Setting the value to 0 disables automatic renewal.

`[no] dn STRING, [STRING], [...]`

Description

Distinguished Name, subject, of this certificate.

Options

Option	Description
`STRING`	Distinguished name, subject, of this certificate. Comma separated list of subject components e.g. CN=Bob Smith, C=SE, etc. Note that returned certificate might contain different set of components
`no`	Resets to empty string

`[no] san [STRING], [STRING], [...]`

Description

Subject Alternate Name (SAN) of this certificate. Comma separated list of SAN components.

Options

Option	Description
`STRING`	SAN components e.g. “dut1.test.com”
`no`	Resets to empty string

`[no]  key-size <1024|2048|4096>`

Description

Key length / Modulus size

Options

Option	Description
`no`	Resets to default

`[no]  profile [STRING]`

Description

Certificate profile name or template name to be included in the certificate request. Supported e.g. by the openxpki SCEP server profiles (pc-client, tls-server, etc.) that are translated into corresponding Extended Key Usage (EKU) flags in the generated X.509 certificate OID 1.3.6.1.4.1.311.20.2.

Options

Option	Description
`STRING`	Profile name
`no`	Resets to empty string

Examples

The process of enrolling a certificate

- Enrollment of certificate

2.9.3. Certificate Revocation

Introduction

Secure communication relies heavily on digital certificates. These certificates validate the identity of devices and ensure encrypted data exchange. However, there are situations where a certificate must no longer be trusted, such as when a private key is compromised or a device is decommissioned. This is where certificate revocation comes into play. In Westermo Operating System (**WeOS**), revocation is implemented in the form of Certificate Revocation List (Certificate Revocation Lists (**CRL**)) checking.

A Certificate Revocation List is a digitally signed list, issued by a Certificate Authority (Certificate Authority (**CA**)), that enumerates certificates that have been revoked before their scheduled expiration date. CRLs are a critical component of the Public Key Infrastructure (Public Key Infrastructure (**PKI**)) used in **WeOS** to maintain trust and security across the network.

Features

- Periodically updated: CRLs are updated at regular intervals to reflect the current status of certificates.
- Locally cached: After every restart, **WeOS** routers cache CRLs in volatile memory to reduce lookup latency and maintain availability during network disruptions.
- Used in certificate validation: During Transport Layer Security (**TLS**) handshakes or other secure operations, **WeOS** checks the **CRL** to ensure that the peer's certificate is still valid.

Limitations

- OpenVPN and logging can be configured to use static **CRL** distribution points (CRL Distribution Point (**CRLDP**)).
- IPsec uses only CRLDPs defined in the incoming certificates.

Overview

Consider the following scenario. Alice wants to communicate with Bob. Bob sends his certificate and a chain leading to Root-**CA**(i.e *bob.pem*, *ca2.pem*, *root.pem*) to Alice for verification.

```
                                    +-------------+
                     publishes      |             |
          root.crl <----------------+  Root-CA    |
                                    |  (root.pem) |
                                    +------|------+
                                           |
                                           | Root signs CA2
                                           |
                                           |
                                    +------v------+
                      publishes     |   CA2       |
            ca2.crl <---------------+  (ca2.pem)  |
                                    |             |
                                    +------|------+
                                           |
                                           | CA2 signs Bob's cert
                  (TLS handshake)          |
                                           |
+-------------+      alice.pem     +------v------+
|             |-------------------->             |
|    Alice    |                    |     Bob     |
|             |  bob.pem + chain   |             |
|            <---------------------             |
+-------------+                    +-------------+
```

Figure 1: CRL in context of TLS handshake.

Alice trusts the Root-**CA** because she added *root.pem* into her own Local-**CA** Store. However when Bob presents his certificate to Alice during a **TLS** handshake, Alice needs to check if Bob's certificate is still valid or if it has been revoked (invalidated before its expiration date).

CRLs are published periodically. The Root-**CA** publishes a root.crl (a list of revoked certificates it issued). CA2 (which is signed by Root-CA) publishes a ca2.crl (a list of revoked certificates it issued, including Bob's if revoked). Both of these certificates are accessible from their respective **CRL** distribution points: http://crl.root.com/root.crl[1] and http://crl.ca2.com/ca2.crl[2].

Alice downloads or has access to the latest ca2.crl. She verifies the signature of ca2.crl and then looks for Bob's certificate serial number in ca2.crl. If Bob's certificate is not in the list, it is valid. If it is in the list, it is revoked and Alice should not trust Bob.

Depending on the service, **WeOS** allows for *static CRLDPs* that are predefined in the configuration, or *dynamic CRLDPs* listed in incoming certificates.

Static vs dynamic CRLDP

- Static **CRLDP**
  - Admin-configured URLs.
  - Available when using OpenVPN and logging services.
- Dynamic **CRLDP**
  - URLs embedded in the peer certificate's **CRL** Distribution Point extension.
  - Currently only available for IPsec.
  - Downloaded when starting the tunnel, cached for further use.

Service	Static **CRLDP**	Dynamic **CRLDP** (from certificate)	Notes
OpenVPN tunnel	Supported	-	Configurable per tunnel.
Logging (Syslog)	Supported	-	Applies to both source and destination.
IPsec	-	Supported	Uses CRLDPs provided in incoming certificates only.
DDNS	-	-	Revocation not supported.

Static CRL Distribution Point

WeOS allows for predefining a list of static CRLDPs, i.e. URLs from where CRLs can be downloaded. Alice configuration should look like this:

```
alice:/#> configure
alice:/config/#> pki
alice:/config/pki/#> revocation 1
alice:/config/pki/revocation-1/#> show
Server #                      : 1
Label                         : revocation1
Description                   : CRL signed by Bob's cert issuer
URL                           : http://crl.ca2.com/ca2.crl
Protocol                      : CRL
Renewal threshold             : 80 %
alice:/#> end
```

[1] http://crl.root.com/root.crl
[2] http://crl.ca2.com/ca2.crl

```
alice:/#> revocation 2
alice:/config/pki/revocation-2/#> show
Server #                        : 2
Label                           : revocation2
Description                     : CRL signed Root-CA
URL                             : http://crl.root.com/root.crl
Protocol                        : CRL
Renewal threshold               : 80 %
```

After configuration, the system downloads CRLs and keep them up-to-date. Other services in the system (e.g. OpenVPN tunnels) that rely on certificates can use these CRLs for revocation checking during **TLS** handshaking.

> **CONFIGURATION**
>
> When using static CRLDPs it is important that the issuer of downloaded CRLs matches the issuer of the end-entity certificate or any other certificate in the chain provided by the peer during the **TLS** handshake.

In our example scenario, Alice will be able to verify the revocation status of bob.pem only if she adds http://crl.ca2.com/ca2.crl[1] to her configuration. If the **CRL** issued by CA2, the issuer of Bob's certificate, is not present in Alice's configuration she will not be able to download it and verify Bob's revocation status. The same is true when validating any other certificate in the chain provided by Bob.

> **NOTE**
>
> Root CAs are self-signed and thus inherently trusted and cannot be revoked. To distrust a self-signed certificate, it must be removed from the system.

> **SECURITY**
>
> The **CRL** distribution point could be targeted by attackers to disrupt the availability of distributed CRLs, making services relying on a CRL not work as expected.

Dynamic CRL Distribution Point

Dynamic **CRL** checking is currently available only for IPsec tunnels and does not require any additional configuration. CRLs are downloaded after the peer's certificate is received and the status is verified. The CRL is cached for any further verification until its expiration.

Revocation Policies

Sometimes it might be desirable to accept a peer's certificate even when the revocation status cannot be verified. This flexibility is important in environments where network connectivity to **CRL** distribution points is unreliable, or where continuous service is critical. This enables organisations to choose the appropriate trade-off between security and operational continuity.

[1] http://crl.ca2.com/ca2.crl

Services in **WeOS** may provide different revocation policies such as `disabled`, `relaxed` or `strict`, allowing administrators to balance security and availability based on operational needs.

Service	Default Policy	Supported Policies	Notes
OpenVPN tunnel	disabled	disabled, relaxed, strict	Policy can be set per tunnel.
Logging	disabled	disabled, enabled	Applies to both source and destination logging configurations.
DDNS	N/A	N/A	Not supported
IPsec	relaxed	relaxed, strict	Policy can be set per tunnel, cannot be disabled

Table 1: Revocation Policy Support by Services

Where available, the **strict** policy will typically enforce immediate rejection of certificates if the **CRL** cannot be checked or is unavailable, maximizing security but potentially disrupting service if the **CRL** source is temporarily unreachable. In contrast, the **relaxed** policy may allow continued operation using cached or expired CRLs, prioritizing service availability even if revocation status cannot be confirmed in real time. Details are provided per service.

Configuration

Configuration of static CRLDPs is done in config context.

```
example:/#> configure pki revocation 1
Creating new revocation entry 1
example:/config/pki/revocation-1#> show
Server #                        : 1
Label                           : revocation1
Description                     :
URL                             :
Protocol                        : CRL
Renewal threshold               : 80 %
```

`[no] label [NAME]`

Description

Assigns a label to the revocation configuration, which will be displayed in lists.

Options

Option	Description
`NAME`	The name of the revocation configuration.
`no`	Resets the label to `revocation#`.

`[no] description [STRING]`

Description

Adds a description to the revocation configuration, which will be displayed in lists.

continues on next page

continued

`[no] description [STRING]`

Options

Option	Description
`STRING`	A short description of the revocation configuration.
`no`	Resets the description to an empty string.

`[no] url [STRING]`

Description

Specifies the URL of the **CRL** location. The URL consists of a scheme, subdomain, top-level domain, second-level domain, subdirectory, and port. Supported schemes are File Transfer Protocol (**FTP**), Trivial File Transfer Protocol (**TFTP**), Hypertext Transfer Protocol (**HTTP**), and Hypertext Transfer Protocol Secure (**HTTPS**).

Options

Option	Description
`STRING`	The URL of the **CRL** location.
`no`	Resets the URL to an empty string.

`[no] protocol [crl]`

Description

Specifies the protocol for the revocation type.

Options

Option	Description
`[crl]`	Currently, only **CRL** is supported.
`no`	Resets the protocol to the default, **CRL**.

`[no] renewal-threshold <1-99>`

Description

Configures the threshold for renewing the **CRL**. The value is specified as a percentage of the **CRL**'s lifetime.

Options

Option	Description
`<1-99>`	The threshold for renewing the **CRL**. The default value is 80 percent, meaning the **CRL** will start renewing when 80 percent of its lifetime has passed.
`no`	Resets the renewal threshold to the default value of 80 percent.

© 2026 Westermo Network Technologies AB

Services

- OpenVPN
- Syslog, (both Source and Destination)
- IPsec

2.10. Routing

2.10.1. Static Routing

Introduction

IP routing enables forwarding of data packets between otherwise isolated network segments, allowing end devices to communicate across networks of different type and topology.

All IP based devices, including routers, have a routing table which is consulted for each outbound IP packet. The path, or route, to be taken is determined based on the destination IP address of the packet. The only real difference between an end-device (or switch) and a router is that the forwarding mechanism is enabled on a router, i.e., inbound packets are allowed to be forwarded based on the routing table.

Three criteria must be met for an IP packet to be routed:

- A route must exist, a default route is sufficient
- The TTL of the packet must be > 1
- IP forwarding must be enabled in the router

The routing table can populated manually by setting up static routes, or automatically by using a dynamic routing protocol like OSPF or RIP, or a combination of both. Static IP routing is usually sufficient for small IP networks, or networks with no redundant paths. To manage routing in larger networks, it is preferred to use dynamic IP routing. Here is an example of how the routing table can look:

IPv4

```
example:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 0.0.0.0/0 [1/0] via 192.168.2.1, vlan1, weight 1, 00:01:44
C>* 169.254.0.0/16 is directly connected, vlan1, 00:01:44
C>* 192.168.2.0/24 is directly connected, vlan1, 00:01:44
```

IPv6

```
example:/#> show ipv6 route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPFv6 | R - RIPng     | [Distance/Metric] | * - FIB route

S>* ::/0 [1/0] via 2001:db8::1, vlan1, weight 1, 00:00:36
C>* 2001:db8::/64 is directly connected, vlan1, 00:00:36
C * fe80::/64 is directly connected, vlan1, 00:00:36
C>* fe80::/64 is directly connected, br0, 00:00:36
```

> **NOTE**
>
> A routing table may contain multiple matching routes for a given destination. The route is selected based on *best match*, which means a `/24`, for example, is a better match than a `/16`. If more than one route matches, the *distance* acts as a tiebreaker.

> **NOTE**
>
> If a route is configured with a next-hop address that is not reachable, that route is *inactive* and may not even be shown in the routing table.

Distance

All routes in the routing table have a distance, which is used to find the best path for a packet if more than one route matches a destination. A lower value is better and a route with distance 0 (connected route) will always be used.

Different routing methods have different default distances which are used unless otherwise specified:

Distance	Protocol
0	Connected
1	Static (manual)
16	Static (DHCP)
110	OSPF
120	RIP

Overview of default route distances for different protocols.

Hence, a route from RIP is worth less than an OSPF route and a static route trumps both. This way multiple routes can be installed to provide automatic fail-over to another routing path in case a router goes down.

The table lists static routes learned from a DHCP server as having the distance value 16. This is to be able to locally override in case the routes (usually just the default route) are wrong or otherwise require adjusting.

IP Forwarding

As mentioned previously, a router must have *IP forwarding* enabled to be able to route packets. This is the default for products with this capability.

IPv4

```
example:/#> configure ip
example:/config/ip/#> forwarding
example:/config/ip/#> leave
```

IPv6

```
example:/#> config ipv6
example:/config/ipv6/#> forwarding
example:/config/ipv6/#> leave
```

DEFAULT BEHAVIOR REGARDING IPV6

Currently, both IPv6 in general and IPv6 forwarding are disabled by default. Both of these must be enabled to allow IPv6 traffic to pass through the device.

Static routes

Static routes are configured from the ip/ipv6 context in the CLI:

IPv4

```
example:/#> configure ip
example:/config/ip/#>
```

IPv6

```
example:/#> config ipv6
example:/config/ipv6/#>
```

A static route requires a destination network and a next-hop destination. The destination network is the network that the route is for, and the destination is instructions how to reach that network from the router.

Static Route to a Next-Hop Gateway

To configure static routes from a given network to a specific next-hop gateway address, use the following commands:

IPv4

```
example:/#> configure ip
example:/config/ip/#> route 192.168.10.0/24 192.168.0.1
example:/config/ip/#> leave
```

IPv6

```
example:/config/ipv6/#> route 2001:db8:10::/64 2001:db8::1
example:/config/ipv6/#> leave
```

Static Route to an Interface

The next-hop can also be specified as an interface instead of an IP address. To configure a static route to an interface, use the following commands:

IPv4

```
example:/config/ip/#> route 192.168.10.0/24 ssl1
example:/config/ip/#> leave
```

IPv6

```
example:/config/ipv6/#> route 2001:db8:10::/64 ssl1
example:/config/ipv6/#> leave
```

USAGE

Using an interface as a next-hop can be useful for point-to-point links, where there is only one destination on the other end of the link. As an example, the interface type in this case could be a SSL or GRE interface.

MULTI-ACCESS INTERFACES

It is generally a good idea to **avoid** using interfaces as next-hops for multi-access interfaces.

Static Route with a Distance

If a specific distance is required, it can be added as an optional parameter. The following example sets the distance to 2:

IPv4

```
example:/config/ip/#> route 192.168.10.0/24 192.168.0.1 2
example:/config/ip/#> leave
```

IPv6

```
example:/config/ipv6/#> route 2001:db8:10::/64 2001:db8::1 2
example:/config/ipv6/#> leave
```

UNREACHABLE DISTANCE

A route with a distance set to 255 is considered unreachable and will not be added to the routing table.

Default Route

Sometimes referred to as the *Default Gateway*, the default route is where an IP device sends all packets for which it does not have a more specific route. It acts as a last resort. To configure a default route, use the following commands:

IPv4

```
example:/config/ip/#> route default 192.168.0.1
example:/config/ip/#> leave
```

IPv6

```
example:/config/ipv6/#> route default 2001:db8::1
example:/config/ipv6/#> leave
```

Blackhole Routes

To prevent traffic from being redirected to the default gateway before routes have been established, you can add a blackhole route. This is is a route for a network that points to the `null0` interface, meaning all traffic for that network will be silently discarded.

When creating a blackhole route it is important that it should have an higher distance (use 255 for blackhole routes) than the real route to the network.

In the following example, we configure a route for a network to a specific next-hop gateway. Additionally, we add a blackhole route for the same network. This blackhole route will discard traffic if the interface for the next-hop gateway goes down, preventing it from being sent to the default gateway.

IPv4

```
example:/config/ip/#> route 172.16.0.0/16 192.168.3.1
example:/config/ip/#> route 172.16.0.0/16 null0 255
example:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

s>* 0.0.0.0/0 [16/0] via 192.168.0.1, vlan1
C>* 169.254.0.0/16 is directly connected, vlan1
s   172.16.0.0/16 [255/0] is directly connected, Null0, bh
s   172.16.0.0/16 [1/0] via 192.168.3.1 inactive
C>* 192.168.0.0/24 is directly connected, vlan1
```

IPv6

```
example:/config/ipv6/#> route 2001:db8:10::/64 2001:db8:2::1
example:/config/ipv6/#> route 2001:db8:10::/64 null0 255
example:/config/ipv6/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> show ipv6 route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPFv6 | R - RIPng     | [Distance/Metric] | * - FIB route

S>* ::/0 [1/0] via 2001:db8::1, vlan1, weight 1, 00:01:45
C>* 2001:db8::/64 is directly connected, vlan1, 00:01:45
C>* 2001:db8:2::/64 is directly connected, vlan2, 00:00:03
S>* 2001:db8:10::/64 [1/0] via 2001:db8:2::1, vlan2, weight 1, 00:00:03
S   2001:db8:10::/64 [255/0] unreachable (blackhole), weight 1, 00:01:46
C * fe80::/64 is directly connected, vlan2, 00:00:03
C * fe80::/64 is directly connected, vlan1, 00:01:45
C>* fe80::/64 is directly connected, br0, 00:01:45
```

TIP

Blackhole routes are also referred to as null routes or discard routes.

Route Monitor

NOTE

Route monitoring is currently only supported for IPv4 static routes.

In cases when there are a multiple paths to a network, it is possible to switch between the available paths using ping trigger alarm, policy route and route monitor.

If the system detects a monitored route target becomes unreachable then it updates the administrative distance of the route to max (255). Making effectivity unreachable and forcing all route on any remaining route(s).

If route became reachable again and the peer is reachable, this clears the alarm and the route monitor update administrative distance to the configured value.

SEE ALSO

- HowTo: Equal Cost Multi-Path Routing
- HowTo: Multiple Upstream Failover
- Alarm
- Policy-Based Routing (PBR)

The following example demonstrates how to configure two default routes, with route monitor.

```
example:/#> configure ip
example:/config/ip/#> route default 192.168.1.99 1 track 1>
```

```
example:/config/ip/#> route default 192.168.2.99 1 track 2>
example:/config/ip/#> show route
Network          Netmask          Gateway          Iface          Dist  TrigId
═══════════════════════════════════════════════════════════════════════════════
0.0.0.0          0.0.0.0           192.168.1.99      *               1    1
0.0.0.0          0.0.0.0           192.168.2.99      *               1    2
example:/config/ip/#> leave
```

This is how the show ip route looks like if both peers are reachable

```
example:/#> show ip route-monitor
Network           Gateway           Trigger-id  Reachable  Installed
0.0.0.0/0         192.168.1.99      1           YES        YES
0.0.0.0/0         192.168.1.99      2           YES        YES
```

2.10.2. Equal-Cost Multi-Path Routing (ECMP)

Introduction

Equal-Cost Multi-Path routing (ECMP), is a routing behaviour that allows for multiple routes for the same destination to be active at the same time, if they share the same cost. This means that flows of packets, bound for any particular destination, could be routed using different paths to reach the destination, if such paths are available.

> **SEE ALSO**
>
> Related Documentation
> - Static Routing

> **SEE ALSO**
>
> Related Examples and Use-Cases
> - Equal-Cost Multi-Path Routing (ECMP) HowTo
> - Multiple Upstream Failover with ECMP Routing

Overview

ECMP is active on the system by default, it is not enabled or disabled by any active action by the operator. If two different routes specify the same destination, it is possible that they will be subject to ECMP, if the following is also true:

- The routes in question have the same cost. As the name implies, **Equal-Cost** Multi-Path routing, will only operate on routes that have the same cost. If the cost is different, only the best of the routes will be inserted into the routing table, as usual.
- The routes need to originate from the same routing protocol in order to be eligible for ECMP routing.

```
                        Path 1 ------------->
                                  .---------.
                192.168.1.0/24    |         |
                       .---------+ RouterA +---------.
                     .´       .1 |         |          `.
                   .´           '---------'            `.
                 .´                                      `.
               .´      Path 2 ------------->               `.
.---------+´                     .---------.                `+---------.
|         |                .1 |         |                  |         |
| Router1 +------------------+ RouterB +------------------+ Router2 +-----------------
|         |  192.168.2.0/24   |         |                  |         |  172.16.0.0/24
'---------+.                    '---------'                .+---------'
               `.                                      .´
                 `.                                  .´
                   `.            .---------.         .´
                     `.       .1 |         |       .´
                       `--------+ RouterC +--------´
                192.168.3.0/24    |         |
                                  '---------'
                        Path 3 ------------->
```

Figure 1: ECMP Routing being employed in order to distribute traffic towards the network 172.16.0.0/24 across three different next hop routers, producing three different paths for the traffic to traverse.

This is an example how it may look if we add a few static routes for the same destination, with different next hops, based on the topology in **Figure 1** above:

```
Router1:/#> configure ip
Router1:/config/ip/#> route 172.16.0.0/24 192.168.1.1
Router1:/config/ip/#> route 172.16.0.0/24 192.168.2.1
Router1:/config/ip/#> route 172.16.0.0/24 192.168.3.1
Router1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Router1:/#>
Router1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 172.16.0.0/24 [1/0] via 192.168.1.1, vlan1, weight 1, 00:00:03
  *                     via 192.168.2.1, vlan2, weight 1, 00:00:03
  *                     via 192.168.3.1, vlan3, weight 1, 00:00:03
C>* 192.168.1.0/24 is directly connected, vlan1, 00:02:01
C>* 192.168.2.0/24 is directly connected, vlan2, 00:00:46
C>* 192.168.3.0/24 is directly connected, vlan3, 00:00:43
```

As we can see from the output of the `show ip route` command, all of the three added routes are active at the same time, for the `172.16.0.0/24` destination. This should now allow traffic bound for the `172.16.0.0/24` network to be distributed on the three different paths, on a per traffic flow basis.

Further, if we for some reason want to exclude one potential route as part of the ECMP routing, we can adjust the `distance` of the route. This will lead to an increase in the cost of the specific route and cause it to no longer be of equal cost to the others. In this situation that route would not be active, as long as the routes with better cost are. As an example, we change the configuration for one of the routes in the following manner:

```
Router1:/#> configure ip
Router1:/config/ip/#> route 172.16.0.0/24 192.168.3.1 50
Router1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Router1:/#>
```

Now when we check the active routes, it should look something like this:

```
Router1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S   172.16.0.0/24 [50/0] via 192.168.3.1, vlan3, weight 1, 00:02:56
S>* 172.16.0.0/24 [1/0] via 192.168.1.1, vlan1, weight 1, 00:02:56
  *                     via 192.168.2.1, vlan2, weight 1, 00:02:56
C>* 192.168.1.0/24 is directly connected, vlan1, 00:11:07
C>* 192.168.2.0/24 is directly connected, vlan2, 00:11:07
C>* 192.168.3.0/24 is directly connected, vlan3, 00:11:07
```

Now we can see that the route we adjusted, with next hop `192.168.3.1`, is not selectable and active in the routing table. Therefore, that specific path will no longer be considered when a specific flow of traffic is directed to a potential next hop.

Path Selection

When a specific destination have multiple valid next hops active in the routing table, a hash will be calculated in order to decide which of the next hops should be selected. The calculation of this hash is performed in such a manner that flows of packets should be distributed towards a specific next hop, and not just individual frames. If this would not be taken into account, it could cause issues with out of order packets.

The hash is calculated based on the header information present in a packet bound for routing. Currently WeOS will look on the following fields in a frame to determine the hash:

- Destination IP address
- Source IP address

When the hash calculation is performed on these values, it will guarantee that frames originating from the same source bound for the same destination, will be selecting the same next hop every time, since the hash calculation will be the same. Therefore, flows of packets should be routed over the same path.

> **NOTE**
>
> If the number of available next hops for a specific destination changes, the path a given packet flow currently selects, based on the hash calculation, could change. This also includes traffic that were not routed through a next hop that may no longer be available.

```
                 ROUTER
                .-----------------------------------------------------.
                |    Hash calc        Hash Mapping    Next Hops       |  192.168.1.0/24
.----.          |                                                     |
| H1 +------>+--->=========--.                                        |       .---------.
'----'          |               |                                     |    .1 |         |
                |               +---> Hash Map 1 ---> 192.168.1.1 --->+------>+   R1    |
.----.          |               |                                     |       |         |
| H2 +------>+--->=========--'                                        |       '---------'
'----'          |                                                     |
                |                                                     |
.----.          |                                                     |
| H3 +------>+--->=========--.                                        |       .---------.
'----'          |               |                                     |       |         |
                |               +---> Hash Map 2 ---> 192.168.2.1 --->+------>+   R2    |
.----.          |               |                                     |    .1 |         |
| H4 +------>+--->=========--'                                        |       '---------'
'----'          |                                                     |
                |                                                     |  192.168.2.0/24
                '-----------------------------------------------------'
```

Figure 1: Simplified overview how flows that enter a router with ECMP, could be mapped towards different next hop addresses.

In **Figure 1** above, we can see a representation how this selection is performed. For some given destination we have two different next hops configured on the device `ROUTER`. The four different hosts have a flow of traffic that they are sending towards the destination.

When the frames enter the `ROUTER` device, they will first undergo a hash calculation, based on the previous mentioned values. The resulting hash value is mapped to either of the two different next hop addresses. Hence, in this example the flows from hosts `H1` and `H2` will be mapped towards `R1` on next hop `192.168.1.1`. Similarly, the flows from `H3` and `H4` will be mapped towards `R2` using the next hop `192.168.2.1`.

Weights

ECMP routes also have an additional concept that is referred to as Weighted Equal-Cost Multi-Path routing. Each individual route will be assigned a specific weight value. This value indicates how large share of possible traffic flows should utilise that particular route. Routes with a higher weight value will receive a larger portion of the possible traffic flows.

Currently, WeOS will have all ECMP routes assigned a weight value of 1, meaning that they routes will attempt and distribute the flows evenly. This weight value can be seen for each individual route when using the `show ip route` command.

> **WARNING**
>
> At the moment no support is present for changing the weights that are assigned to any ECMP routes, they will always have a weight of 1. This includes static as well as dynamic routes.

Route monitor

ECMP static routes, can be augmented using a feature called route monitor. Route monitor track a route using ping trigger and policy route, if the route becomes unreachable then the admin distance of the route will be set to max, this makes the route inactive. All traffic will be directed to the remaining routes.

And if the route becomes reachable again then the admin distance is reset to the configured value.

> **SEE ALSO**
>
> - How to configure

2.10.3. Virtual Router Redundancy Protocol

Introduction

The Virtual Router Redundancy Protocol (VRRP), is a standard protocol that is used to provide automatic assignment of available routers to its participating hosts. The purpose of this is to provide *redundancy* between a host and its router. Further, VRRP can also be used for *load balancing* purposes.

The protocol allows multiple physical routers to act as a single router outwards towards its hosts. In the eyes of a host, only one router actually exists. This allows the routers to seamlessly failover to each other and take over routing duties, in case one or more goes down, without the hosts knowing.

All the participating routers together constitute a *virtual router*, and in normal cases exactly one of these will be selected as the currently active router, the *master router*. The non-selected routers are known as *backup routers*. The participating routers themselves are usually called *physical routers*, to differentiate the concept from the virtual router. An abstraction of this can be seen in **Figure 1** below.

```
                    |
                .--.-.
               ( (    )__
              (_,  \ ) ,_)  Internet/Intranet
                '-'--`--'
                 |
-------+-----+-----+------- 10.0.0.0/24
       |           |
  .----|-----------|----.
  | .--+--.     .--+--. |
  | | R1  | VR1 | R2  | |
  | '--+--'     '--+--' |
  '----|-----------|----'
       |           |
--+----+-----+-----+----+-- 192.168.1.0/24
  |          |          |
.-+--.     .-+--.     .-+--.
| H1 |     | H2 |     | H3 |
'----'     '----'     '----'
```

Figure 1: Depicting how VRRP bundles several physical routers into seemingly one router to all connecting hosts. Here R1 and R2 will form the virtual router VR1. The connecting hosts only know about VR1, and do not need to know about the routers that constitute the virtual router.

The scope of this document is to provide some basic information on VRRP, in addition to all VRRP related configuration options that the system provides.

For example use-cases of VRRP and general setups, refer to the following:

- VRRP Basic
- VRRP Synchronisation groups
- VRRP Load balancing

Overview

A virtual router on a LAN is identified by a **virtual router id** (“vrid”). All participating hosts must use the same VRID to be part of the same virtual router - however, several different VRIDs (and thus several different virtual routers) may be in use on the same LAN simultaneously. Any physical

device can also run several VRRP instances concurrently, for example if the device is connected to several LANs, and should participate in virtual routers for several of them.

When taking over routing duties, the master router assumes use of a **virtual IP address**. Because of this, other devices should use this virtual address as their gateway/router for the hop in question. There is also a set of virtual MAC addresses, one per possible VRID, that is reserved for exclusive use by the master router. The virtual IP address resolves to this MAC. It is highly recommended that the chosen virtual IP address be on the same subnet as the physical addresses the participating routers use on the corresponding interface. It is also not recommended to re-use an address that is already used by a physical device, either one of the participating routers or another device on the network.

> **NOTE**
>
> Using a VRRP virtual IP address ("VIP") as the destination for network services is not recommended. Examples of network services include, but are not limited to, NTP-Server and DHCP-Server.
>
> When a VRRP virtual address is used in this way, it can be advantageous to enable "accept mode", as described below. It should however be noted that the multicast DNS discovery service is not intended to be used for the VIP, regardless of the "accept mode" setting.

In basic layer 3 routing scenarios, a failover (i.e. a change of ownership of the vaddr) will be transparent to applications on devices communicating via the virtual router. Exceptions to this include firewall state, which is not synchronised across hosts participating in the virtual router - in other words, TCP streams and other stateful connections will be broken by a failover.

Signalling between the participating hosts occurs over IP multicast group **224.0.0.18** using IP protocol number 112. The master router will periodically advertise that it is still the master. When 3 advertisement intervals have elapsed without a message being heard, the backup routers perform an election, and another router becomes the new master. Normally all the backup routers are silent, but during elections and startup of participating routers, the backup routers will exchange traffic.

Each participating router is configured with a **priority** value, higher values indicates a higher priority. This priority dictates which of the routers will be selected as the master router. Further, the routers also have a **preempt** option, which when enabled will allow a higher-priority backup router force an election. This will result in it taking over as master router, even when another physical router already is master.

In order to handle specific use cases - for instance, when running VRRP over link layer redundancy protocols, or link aggregates - there is an optional **startup delay** option available, which introduces a configurable delay to the startup of all VRRP instances on the device. Note that using this option together with the **preempt** option may lead to unexpected behaviour, and should be avoided if possible. Specifically, when using both options, if a device receives VRRP advertisements during its startup delay period, it may not become master after the delay expires, even if it has the highest priority and **preempt** is enabled.

There are two mutually incompatible versions of VRRP in common use, version 2 and version 3. Version 3 supports additional features such as sub-second precision for the advertisement interval (and thus failover time) and IPv6 support.

For additional details on VRRP and its workings see RFCs **3768** for VRRPv2 and **5798** for VRRPv3.

Using VRRP synchronisation groups, it is possible to synchronise the state of multiple VRRP instances across multiple devices. This can be used to achieves various goals, for instance to ensure that a backup instance on one device only becomes master if the backup instance on another device also becomes master, which can be useful in certain load balancing scenarios. For more information on synchronisation groups, see the VRRP Synchronisation groups document.

The virtual router IP address, commonly called the "VIP", is sometimes used not only as a gateway address for hosts, but also as a host address for network services. In such cases, it can be useful to enable the "accept" mode of VRRP. This mode is described in RFC 5798.

Configuration

All VRRP configuration is carried out from the *router* configuration context in the CLI.

```
example:/#> configure
example:/config/#> router
```

This context has two additional sub-contexts that are related to VRRP:

- **instance**: Used to configure individual VRRP instances.
- **group**: Used to configure VRRP sync groups.

Additionally, this context also contains the following option, which applies to all instances:

`[no] vrrp-startup-delay <TIME>`

Description

Optionally delay startup by TIME seconds.

This is useful in order to wait for slow interfaces or other services. The setting applies to all VRRP instances, and valid values for TIME are between 0 (exclusive) and 300 (inclusive). This value is in whole seconds.

Options

Option	Description
`no`	Remove any delay and starts VRRP instances as soon as possible.

Instances

VRRP instances are configured from the *router* configuration context in the CLI. When creating a new VRRP instance an ID needs to be provided during the initial creation step:

```
example:/config/router/#> vrrp 1
example:/config/router/vrrp-1/#>
```

Syntax

`[no] enable`

Description

Activate or deactivate VRRP instance.

continues on next page

continued

`[no] enable`

Options

Option	Description
`no`	Disable the instance.

`[no] address <IP>`

Description

Virtual router IP address, must be in same subnet as interface.

> **NOTE**
>
> Currently there is only support for one virtual IP address per virtual router.

Options

Option	Description
`no`	Remove the configured IP address.
`IP`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] vrid <VRID>`

Description

Set router ID. It is used to identify and represent each virtual router.

Options

Option	Description
`no`	Remove the configured VRID.
`VRID`	Number in the range of **1-255**.

`[no] iface <IFNAME>`

Description

Set interface to listen on.

Options

Option	Description
`no`	Remove the configured interface .
`IFNAME`	Name of an interface. Current available interfaces on the system can be shown in the following manner:

continues on next page

continued

`[no] iface <IFNAME>`

Option	Description
	example:/config/router/vrrp-1/#> do show iface INTERFACE OPER ADDRESS/ LENGTH SOURCE MAC/PTP ADDRESS lo UP 127.0.0.1/8 static 00:00:00:00:00:00 vlan1 UP 172.31.1.1/24 static 0c:f4:21:44:a6:00

`[no] interval <INTERVAL> [msec]`

Description

Advertisement interval, in seconds or in 10 millisecond intervals.

Default: **1** second.

> NOTE
>
> - If *VRRPv2* is used the value must be a positive integer no larger than 255.
> - If *VRRPv3* is used the value must be between 10 and 40000 milliseconds, inclusive.
> - An interval lower than 100 milliseconds may impact system performance negatively.

Options

Option	Description
`no`	Reset to the default value.
`INTERVAL`	Time in seconds, unless *msec* is specified.
`msec`	If specified the *INTERVAL* value provided will be treated as milliseconds and not seconds. Sub-second time values can only be utilised if *VRRPv3* is used.

`[no] priority <1-255>`

Description

Router priority, 255 only if IP owner.

Default: **100**.

Options

Option	Description
`no`	Reset to the default value.

`[no] version <2 | 3>`

Description

Set VRRP version.

Default: **2**.

continues on next page

continued

`[no] version <2 | 3>`

Options

Option	Description
`no`	Reset to the default value.

`[no] preempt [delay <0-1000>]`

Description

Preempt existing router of lower priority, with optional delay.

Options

Option	Description
`no`	Disable router preemption.
`delay`	Set the optional delay value, between **0-1000**

`[no] track trigger <ID> adjust <DELTA>`

Description

Track event trigger to adjust priority dynamically.

Create a ping trigger and connect to it here to adjust the priority of the VRRP instance depending on the upstream connectivity.

Example:

```
example:/config/router/vrrp-1/#> track trigger 1 adjust -10
```

Options

Option	Description
`no`	Remove the configured tracker.
`ID`	Id number of the trigger to use. Triggers are created under the alarm context, the following is an example how to create a trigger: example:/config/#> alarm example:/config/alarm/#> example:/config/alarm/#> trigger ping example:/config/alarm/trigger-1/#> peer 192.168.0.1 For more information on trigger creation see the configuration page for alarms.
`DELTA`	Integer value to use when dynamically adjusting the VRRP priority. Must be in the range: **−255 to 255**.

`[no] vmac`

Description

Create and/or use a virtual interface for VRRP. Uses the VRRP MAC address.

continues on next page

continued

`[no] vmac`

Default: **Enabled**

Options

Option	Description
`no`	Disable the vmac.

`[no] accept`

Description

Enable or disable Accept_Mode, as per RFC 5798. Enabled by default.

This setting is only standardised for VRRP version 3, but is also available for version 2. When enabled, the instance will accept packets destined to the virtual IP address, regardless of VRRP state.

This can be useful in certain scenarios, for instance if the virtual IP address is used as a destination for network services.

Synchronisation Groups

VRRP synchronisation groups are configured from the *router* configuration context in the CLI. When creating a new sync group a valid ID must be provided during the creation process:

```
example:/config/router/#> group test
example:/config/router/vrrp-grp-test/#>
```

Note: A synchronisation group must contain *exactly two* VRRP instances.

Syntax

`[no] member <VRRP_INSTANCE> [ <VRRP_INSTANCE>...]`

Description

Set the synchronisation group VRRP member instances.

Options

Option	Description
`no`	Remove configured member instance(s).
`VRRP_INSTANCE`	Id of a configured VRRP instance to be used as a group member.

`[no] description`

Description

Add or edit free-text description.

continues on next page

continued

`[no] description`

Options

Option	Description
`no`	Remove the configured description.

Status

The status for all active VRRP instances on a router can be observed from the admin context:

```
example:/#> show vrrp

VRRP Instances
INSTANCE NAME     BASE INTERFACE    VRID ADDRESS         PRIORITY STATE   GROUP
ecn1              vlan1             1    10.0.0.100      100(100) BACKUP  group2
two               vlan1             2    10.0.0.101      100(100) BACKUP  group1
three             vlan1             3    10.0.0.102      100(100) BACKUP  group1
four              vlan1             5    10.0.0.200      100(100) BACKUP  group2
five              vlan1             99   10.0.0.199      100(100) BACKUP  -

VRRP Groups
GROUP NAME        DESCRIPTION       INSTANCE 1        INSTANCE 2        SYNC STATE
group1                              two               three             BACKUP
group2            group2            ecn1              four              BACKUP
```

Note that for the VRRP instances, the “PRIORITY” column shows both the configured priority and the effective priority in parentheses. The effective priority is the one that is actually used in the VRRP protocol, and may differ from the configured priority if there are dynamic adjustments in place, for example due to tracking.

For the VRRP groups, the “SYNC STATE” column shows the current state of the synchronisation group. The state is determined based on the states of the member instances and their synchronisation status and all group members will transition to this state.

If a more detailed view of a specific VRRP instance is desired, the following command can be used. This includes statistics and other details about the instance:

```
example:/#> show vrrp ecn1
VRRP Instance: ecn1
  Internal name: ecn1_vlan1_1
  Interface: vrrp-ecn1
    Base interface: vlan1
    VMAC interface: vrrp-ecn1
  VRID: 1
  State: BACKUP
    Want state: BACKUP
    Last transition: 1432.39 seconds ago
  Priority: 100 (Effective: 100)
  Master priority: 120
  Version: 2
  Advert interval: 1.00 seconds
  Master advert interval: 1.00 seconds
  VIP set: No
    VIP: 10.0.0.100
  Last master address: 10.0.0.2
```

```
  Source address: 10.0.0.1
  Last received packet source address: 10.0.0.2
  Sync: Yes
    Sync group name: group2
    Internal Sync group name: VG_group2
    Sync state: BACKUP
  Preempt: Enabled
    Preempt Delay: 0 seconds
  Accept Mode: Disabled
  Statistics:
    Advertisements received: 1432
    Advertisements sent: 0
    Times became master: 0
    Times released master: 0
    Packet length errors: 0
    Advertisement interval errors: 0
    IP TTL errors: 0
    Invalid type received: 0
    Address list errors: 0
    Invalid auth type: 0
    Auth type mismatches: 0
    Authentication failures: 0
    Priority zero received: 0
    Priority zero sent: 0
```

Troubleshooting

The most commonly encountered issue with VRRP is that the instance stays in FAULT state. This is often caused by the underlying base interface being down, or the configured virtual IP address not being in the same subnet as the physical address of the base interface. In these cases, the instance will transition to FAULT state and remain there until the underlying issue is resolved.

Another common issue is that VRRP instances are grouped in a way prevents the synchronisation group from settling into a stable state. This can be caused by incompatible priorities on the member instances across different devices, or by using the preemption setting.

If an instance unexpectedly remains in the MASTER state, this can be caused by a loss of connectivity to the other participating routers, for example due to a network issue or a misconfiguration of the base interface. This can be verified by the "`Advertisements received`" and "`Last received packet source address`" fields in the detailed instance view, the latter of which should show the address of the master router. If these fields are empty, it indicates that no advertisements have been received from the master router, which can be a sign of a connectivity issue.

A rare but possible configuration is one where the virtual IP address is set to an address that is already used by the base interface. This is not recommended, but if it is done, then **Accept Mode** should always be enabled, to ensure connectivity. See the `[no] accept` configuration option for more information on this setting.

When the firewall is in use, it is important to ensure that the necessary rules are in place to allow VRRP traffic to pass between the participating routers. This can either be done by explicitly allowing traffic to the VRRP multicast group (224.0.0.18) and/or by allowing the VRRP protocol (IP protocol number 112).

2.10.4. Open Shortest Path First Protocol

Introduction

The Open Shortest Path First (OSPF) protocol is a dynamic routing protocol. Utilisation of a dynamic routing protocol can help simplifying configuration, since there is no need to for manual configuration of static routes. Further, it also helps improving the robustness of the network, by automatically being able to adapt the routing based on changes in the network topology.

OSPF is a protocol that falls within the group of interior gateway protocols (IGPs). OSPF is an example of a *link-state* routing protocol. An advantage that *link-state* routing protocols, like OSPF, have over distance vector routing protocols, such as RIP, is the fast convergence after a topology change and increased scalability.

For example use-cases and example setups, refer to:

- OSPF Basic
- OSPF Areas
- OSPF Route Summarization
- OSPF Route Redistribution

Overview

As mentioned OSPF is an example of what is known as a *link-state* protocol. In a link-state protocol each individual router announce information about their identity (router-id), along with their directly connected networks, and other neighbouring routers. All this information is flooded throughout the OSPF domain, this ensures that each router will gain complete knowledge about every router and link in the entire topology. This is necessary in order to be able to compute the best path (the least cost path) to reach every destination.

```
   Net-A                              Net-B
  .--.-.                             .--.-.
 ( (    )__                        ( (    )__
(_,  \ ) ,_)                       (_,  \ ) ,_)
  '-'--`--'                          '-'--`--'
     |                                  |
    |                                  |
     |                                  |
.----+-----.                       .----+-----.
|          |                       |          |
| Router-A +-----------------------+ Router-B |
|          |                       |          |
'----+-----'                       '----+-----'
     |                                  |
     |                                  |
     |                                  |
.----+-----.                       .----+-----.
|          |                       |          |
| Router-C +-----------------------+ Router-D |
|          |                       |          |
'----+-----'                       '----+-----'
     |                                  |
     |                                  |
     |                                  |
  .--.-.                             .--.-.
 ( (    )__                        ( (    )__
(_,  \ ) ,_)                       (_,  \ ) ,_)
  '-'--`--'                          '-'--`--'
   Net-C                              Net-D
```

Figure 1: Simple network topology with interconnected routers and networks.

As an example, *Router-A* in Figure 1 would send out OSPF messages informing other routers about its router-id, its connected networks, i.e., Net-A and the links towards the routers B and C. The other routers would also do the same thing.

OSPF Hierarchy and Areas

Being a link-state protocol OSPF requires routers ti keep a lot of routing information in their database:

- Routers will keep a database with information of every router and in link in the entire OSPF domain.
- OSPF routers can also redistribute and keep routing information learnt from external sources (static routes, routes learnt via other routing protocols, etc.).

Therefore, to reduce the burden of keeping state information about the entire OSPF domain on every single router, the domain can be split into OSPF areas. An example is presented in Figure 2, here the routers have been divided into four different areas. When an OSPF network is split into multiple areas, each router will only have full knowledge of the topology within the area it is part of. Routers will also keep *summary* information about destinations outside their own are, but no additional information on the topologies of other areas.

```
                 . . . . . . . . . . . . . . . . . . . . . . . .
                 .              Area 0.0.0.0                   .
                 .   .---.     (Backbone area)        .---.    .
                 .   | R |                            | R |    .
                 .   '---'            .---.           '---'    .
                 .                    | R |                    .
                 .                    '---'                    .
                 .                                             .
                 .                                             .
            .-------.             .-------.             .-------.
            |       | . . . . . . .|       |. . . . . . . |       |
            |  ABR  |              |  ABR  |              |  ABR  |
. . . . . . |       |   . . . . . .|       |. . . . . .   |       | . . . . . .
.  .---.    '-------'   .  .---.   '-------'   .---.  .   '-------'    .---.  .
.  | R |        .       .  | R |               | R |  .       .        | R |  .
.  '---' .---.  .       .  '---'     .---.     '---'  .       .  .---. '---'  .
.    | R |  .       .            | R |            .       .  | R |        .
.    '---'  .       .            '---'            .       .  '---'        .
.           .       .                             .       .               .
. Area 0.0.0.1  .       .       Area 0.0.0.2          .       . Area 0.0.0.3  .
. . . . . . . . .       . . . . . . . . . . . . . . . .       . . . . . . . . .
```

Figure 2: Sample OSPF hierarchy with a backbone area and three other areas.

Every network that an OSPF router should be aware of must always be associated with an area. If multiple areas are not necessary they will belong to the backbone, area 0.0.0.0.

Areas are organised in a two-level hierarchy. The top of this hierarchy will always be constituted by the backbone, area 0. Since it is limited to two levels all other areas must be connected to the backbone area. Any router that serve networks that are part of different areas are known as an Area Border Router (ABR). Direct connections between areas that do not traverse through the backbone is prohibited.

Virtual Links

Even though all areas must be connected directly to the backbone (area 0), situations could present themselves where it is not physically possible to connect an area directly to the backbone. Because of this OSPF provides a feature referred to as *virtual links*, that can allow an area to establish a connection to the backbone through a non-backbone area.

> **NOTE**
>
> Virtual links are **currently NOT supported.**

```
. . . . . . . . . . . .    . . . . . . . . . . . .     . . . . . . . . . . . .
.  Area 0.0.0.0     .    .    Area 0.0.0.1     .     .    Area 0.0.0.2     .
.             .    .                     .     .                     .
.        .---.        .    .        .---.        .     .        .---.        .
.        | R |        .    .        | R |        .     .        | R |        .
.        '---'        .    .        '---'        .     .        '---'        .
.                    .-------.                  .-------.                    .
.  .---.       .---. |       |   Virtual Link   |       | .---.       .---.  .
.  | R |       | R | |  ABR  +==================+  ABR  | | R |       | R |  .
.  '---'       '---' |       |                  |       | '---'       '---'  .
.                    '-------'                  '-------'                    .
.        .---.        .    .        .---.        .     .        .---.        .
.        | R |        .    .        | R |        .     .        | R |        .
.        '---'      .    .        '---'    .     .        '---'        .
. . . . . . . . . . . .    . . . . . . . . . . . .     . . . . . . . . . . . .
```

Figure 3: An example of a virtual link between area 2 and the backbone, going through area 1.

Area Types

Areas in OSPF can be designated a number of specific area types. The purpose for this is to allow for an even increased amount of control on the routing information that is distributed throughout the OSPF network. The different area types allow to more specifically decide what type of routing information is allowed. Being able to configure the OSPF network to reduce the amount of routing information that is passed to specific areas, can help reduce the size of the link state database and routing table, that is maintained by each individual router. The following area types can be configured:

- *Standard area:* This is the default type of all areas added to the OSPF domain, except for area 0. This area type will accept any link update, summary routes, and external routes.
- *Backbone area (area 0):* The main area of an OSPF network, all other areas must be connected to this backbone area. All other areas that wants to communicate with each other must do so through this area. Otherwise, it has the same functionality as a standard area.
- *Stub area:* This area type do not accept any external network routes outside of the OSPF domain. Hence, the routers inside the area will not store any routing information to external destinations outside of the OSPF network. Instead, the router connected to the backbone (ABR) will advertise a default route,towards itself, into the area so that external network destinations are reachable. Also, no router inside the area is allowed to redistribute routing information learnt from external sources, no *autonomous system border router* (ASBR) is allowed.
- *Totally stubby area:* Behaves in the same manner as a stub area, and in addition this area type does not allow summary routes from other ares inside the OSPF network. Meaning that the routers in the area will not store any routing information to destinations within the OSPF network. Like a regular *stub area* the ABR will advertise a default route towards itself, so that both destinations external to OSPF and external to the area within the OSPF network, can be reached.
- *Not so stubby area (NSSA):* This type of area is very similar to a stub area, with the difference being that it allows ASBR routers inside its area. This makes it possible to advertise external routes through a stubby area.

- *NSSA totally stub area:* Is a combination of a *Totally stubby area* and a *NSSA area*.

> **TIP**
>
> For some example use-cases on different OSPF areas refer to here.

Router Types

Within an OSPF domain, routers can be divided into a number of different types. Their role and position, within the OSPF domain, will decide what kind of type it will become. The following are the different router classifications that are used within an OSPF network:

- *Internal Router (IR):* Are routers where all of their connections belong to the same area.
- *Backbone Router (BR):* Are routers that has at least one interface connected to the backbone area. ABR routers are always backbone routers. Routers that only have interfaces connected to the backbone, are also backbone routers.
- *Area Border Router (ABR):* Are routers which are attached to multiple different areas. These routers must also be connected to the backbone area (area 0). An ABR summarise topology information of its attached areas and distribute it to the backbone area.
- *Autonomous System Boundary Router (ASBR):* Routers that are attached to routers, part of other Autonomous Systems, running a different routing protocol. ASBRs distribute this external routing information into the OSPF domain.

```
. . . . . .   . . . . . .   . . . . . .
.   Area 1  .   .  Area 0 .   .  Area 2 .
.           .   .         .   .         .
. .-------. .-------. .-------. .-------. .
. |       | |       | |       | |       | .
. |  IR   +-+-------+-+  ABR  +-+---+---+-+  ABR  +-+-------+-+  IR   | .
. |       | |       | |  BR   | |   | | |  BR   | |       | |       | .
. '---+---' '-------' '-------' |   | '-------' '---+---' .
.     |             .   .     |     .   .             |     .
. .---+---.         .   . .---+---. .   .         .---+---. .       .--.-.
. |       |         .   . |       | .   .         |       | .      ( (    )__
. |  IR   |         .   . |  BR   | .   .         | ASBR  +-------(_,  \ ) ,_)
. |       |         .   . |       | .   .         |       | .       '-'--`--'
. '-------'         .   . '-------' .   .         '-------' .   External Network
. . . . . .   . . . . . .   . . . . . .
```

Figure 4: A small network of a number of routers in different areas, with the purpose to show different OSPF router types.

In Figure 4 above a simple example example is provided to show how the different router types are assigned.

Configuration

The configuration of OSPF is separated into two different parts. The actual router configuration and OSPF related settings for each individual interface.

Router

OSPF is configured under the `router` section of the global configuration context:

```
example:/#> configure
example:/config/#> router
example:/config/router/#> ospf
example:/config/router/ospf/#>
```

`[no] router-id <ID>`

Description

Set the OSFP router identifier, which must be unique within the OSPF domain.

The router ID is a 32-bit value, given in a dotted decimal form <a.b.c.d>, where a-d are numbers in the range 0-255. Commonly the router ID is set equal to one of the router's IP addresses.

Default: Auto (`no router-id`)

Options

Option	Description
`no`	Use automatically assigned router-id. The router-id will automatically try to be assigned based on the following selection processes: 1. If the lo interface have a statically configured address (other than 127.0.0.1) that address will be used as the router-id. 2. The current highest statically configured ip address for any of the interfaces used by OSPF. 3. If no static address has been configured the first available address, on any of the interfaces used by OSPF, will be selected as the router-id.
`ID`	ID in a dotted decimal form <a.b.c.d>, where a-d are numbers in the range 0-255.

> **NOTE**
>
> The router-id will not be sticky if set to auto. If any configuration changes are made the mentioned selection process will be performed again, so it could result in a new router-id to be selected. In order to be sure what the router-id will be set to in any situation, simply specify it manually.

`[no] distance <1-255>`

Description

Set the administrative distance to be used for all routes the are learnt via OSPF.

For additional information on routing distance, see this.

Default: 110

Options

Option	Description
`no`	Reset the distance to the default value.

`[no] passive-interface`

Description

Enable/Disable passive-interface, that control if OSPF should be passive on all interfaces by default.

Whether a specific interface should be passive or not, is configurable for each interface.

Default: Active (`no passive-interface`)

Options

Option	Description
`no`	Set to *active*, not passive.

`[no] neighbor <IPADDR[,IPADDR,...]>`

Description

Manually define OSPF neighbours.

This may be useful when intermediate switches do not propagate IP multicast, or when using OSPF in NBMA (non-broadcast multiple access) networks.

> **NOTE**
>
> Remember to set the network type of each interface to `non-broadcast` if you want OSPF to send unicast Hello messages on NBMA links.

Default: Disabled

Options

Option	Description
`no`	Remove configured neighbour.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] network <NETWORK> [area <AREA-ID>]`

Description

Specify a network to announce and an OSPF area the network belongs to.

Default: Disabled, no `network` entries have been configured when first activating OSPF. The backbone area (0.0.0.0) is used as default area.

Options

Option	Description
`no`	Remove any configured network.
`NETWORK`	IP address in standard quad-dotted notation including netmask, e.g. 192.168.1.1/24.

continues on next page

continued

`[no] network <NETWORK> [area <AREA-ID>]`

Option	Description
`AREA-ID`	The area ID is a 32-bit number, and is entered in dotted decimal form, or as an integer (0..2 32 − 1).

`[no] area <AREA-ID>`

Description

Manage area specific settings: stub, route summarisation, etc.

> **NOTE**
>
> Enters a sub-configuration context for area specific settings.

Options

Option	Description
`no`	Remove custom area settings.
`AREA-ID`	The area ID is a 32-bit number, and is entered in dotted decimal form, or as an integer (0..2 32 − 1).

`[no] redistribute <connected|static|rip> [metric <0-16777214>] [metric-type <1|2>]`

Description

Redistribute external routing information into the OSPF domain.

Default: Disabled

Options

Option	Description
`no`	Remove redistribution of external routing information.
`connected`	Redistribute routes to directly attached networks, which have not been defined to belong to any OSPF area in the “network” command.
`static`	Redistribute static routes set with the `ip route` command.
`rip`	Redistribute routes set by RIP.
`metric`	The cost of reaching the external AS (autonomous system).
`metric-type 1`	External type 1 routes, E1 in the routing table, include the cost of the ASBR itself in the LSA.
`metric-type 2`	External type 2 routes, E2 in the routing table, do NOT include the cost of the ASBR, this is the default.

© 2026 Westermo Network Technologies AB

`[no] distribute-default [always] [metric <0-16777214>] [metric-type <1|2>]`

Description

Introduce a default route into the OSPF domain, I.e., announce that this router can reach network 0.0.0.0/0.

Default: Disabled

Options

Option	Description
`no`	Disable distribution of a default route.
`always`	Always advertise the default route, regardless if one exist or not. Otherwise, it will only be advertised if it exist.
`metric`	The cost of reaching the external AS (autonomous system).
`metric-type 1`	External type 1 routes, E1 in the routing table, include the cost of the ASBR itself in the LSA.
`metric-type 2`	External type 2 routes, E2 in the routing table, do NOT include the cost of the ASBR, this is the default.

Area Settings

OSPF area specific settings can be configured from a sub-context under the router OSPF configuration:

```
example:/config/router/ospf/#> area 1
example:/config/router/ospf/area-0.0.0.1/#>
```

`[no] default-cost <0-16777215>`

Description

Set the cost of the default route injected into a stub area. This setting only applies to the ABRs of a stub or NSSA area.

Default: 1

Options

Option	Description
`no`	Reset to the default value.

`[no] range <NETWORK> [advertise|not-advertise|substitute <NETWORK>] [cost 0-16777215]`

Description

Configure inter-area route summarisation, route filtering or route substitution.

Default: Disabled

continues on next page

continued

`[no] range <NETWORK> [advertise|not-advertise|substitute <NETWORK>] [cost 0-16777215]`

Options

Option	Description
`no`	Remove configuration.
`NETWORK`	IP address in standard quad-dotted notation including netmask, e.g. 192.168.1.1/24.
`advertise`	Use to aggregate routes (within the area) that match the specified `NETWORK` range, before distributing the routes outside the area. Thus, all routes within the provided range are *summarised* as a single route, when advertised outside the area.
`not-advertise`	Use to prohibit routes (within the area) matching the specified `NETWORK` range, to be distributed outside the area. Thus, all routes within the provided range are *filtered*.
`substitute`	Use to substitute routes (within the area) matching the specified `NETWORK` range, with another specified `NETWORK` before distributing the routes outside the area. Thus, all routes within the provided range are *substituted* with another `NETWORK`.

`[no] stub [no-summary]`

Description

Configure an area as a *stub* area.

To create a stub area, all routers in the area (ABRs as well as internal routers) must declare the area as stub.

Default: Disabled (i.e., areas are *regular* OSPF areas by default)

Options

Option	Description
`no`	Reset to a *regular* OSPF area.
`no-summary`	Configure the area as a *totally stubby area*, this should be done for all ABRs in the area.

`[no] nssa [no-summary]`

Description

Configure an area as a *not so stubby area*.

To create a nssa area, all routers in the area (ABRs as well as internal routers) must declare the area as nssa.

Default: Disabled (i.e., areas are *regular* OSPF areas by default)

continues on next page

continued

`[no] nssa [no-summary]`

Options

Option	Description
`no`	Reset to a *regular* OSPF area.
`no-summary`	Configure the area as a *NSSA totally stub area*, this should be done for all ABRs in the area.

Timers Settings

Exposed OSPF timers can be configured from a sub-context under the router OSPF configuration:

```
example:/config/router/ospf/#> timer
example:/config/router/ospf/timer/#>
```

`[no] lsa-min-arrival <0-600000>`

Description

Set minimum delay in receiving new version of a LSA in milliseconds.

Default: 1000

Options

Option	Description
`no`	Reset to the default value.

`[no] lsa-min-interval <0-5000>`

Description

Set delay between sending LSAs in milliseconds.

Default: 5000

Options

Option	Description
`no`	Reset to the default value.

`[no] spf-delay <0-600000>`

Description

Set delay from first change received untill SPF calculation in milliseconds.

Default: 0

continues on next page

continued

`[no] spf-delay <0-600000>`

Options

Option	Description
`no`	Reset to the default value.

`[no] spf-min-hold-time <0-600000>`

Description

Set initial hold time between consecutive SPF calculations in milliseconds.

Default: 50

Options

Option	Description
`no`	Reset to the default value.

`[no] spf-max-hold-time <0-600000>`

Description

Set maximum hold time in milliseconds.

Default: 5000

Options

Option	Description
`no`	Reset to the default value.

Interface Specific Settings

Interface settings that concern OSPF can be found in the `ospf` sub-context located under the individual `iface` configuration context:

```
example:/#> configure
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> ospf
example:/config/iface-vlan1/ospf/#>
```

`[no] passive [auto]`

Description

Control passive-interface setting for OSPF on this interface.

Default: Auto (Follows router ospf `passive-interface` setting)

continues on next page

continued

`[no] passive [auto]`

Options

Option	Description
`no`	Set to *active*, not passive.
`auto`	Follow router ospf `passive-interface` setting.

`[no] priority <0-255>`

Description

Configure the designated Router (DR) election priority for this interface. A higher value increases the chance to be elected the designated router.

A priority of 0 indicates that the device is not eligible for election.

Default: 1

Options

Option	Description
`no`	Reset to the default value.

`[no] cost <1-65535>`

Description

Configure the interface OSPF cost.

Default: 10

Options

Option	Description
`no`	Reset to the default value.

`[no] hello-interval <1-65535>`

Description

Configure OSPF hello interval (in seconds) for this interface.

> **NOTE**
>
> The hello interval setting must be the same on neighbour routers.

Default: 10 (seconds)

continues on next page

continued

`[no] hello-interval <1-65535>`

Options

Option	Description
`no`	Reset to the default value.

`[no] dead-interval <1-65535>`

Description

Configure OSPF dead interval (in seconds) for this interface.

> **NOTE**
>
> The dead interval setting must be the same on neighbour routers.

Default: 40 (seconds)

Options

Option	Description
`no`	Reset to the default value.

`[no] network [broadcast|non-broadcast|point-to-point]`

Description

Explicitly set the OSPF network type.

This setting is useful if the OSPF network type for the specific interface needs to be explicitly set. The network type can be set to any of the following explicit modes:

- auto
- broadcast
- non-broadcast
- point-to-point

By default no explicit network type will be selected and the selection performed automatically. This selection is performed separately for each individual interface, based on the type of the interface.

Default: auto (`no network`)

Options

Option	Description
`no`	Reset to the default value.
`broadcast`	Used when multiple routers are connected to the same broadcast domain, e.g an ethernet LAN network. This network type is useful when all the routers on the network need to be able to communicate with one another.

continues on next page

continued

`[no] network [broadcast|non-broadcast|point-to-point]`

Option	Description
	In auto mode, this will generally be the default for essentially all interface types on the device. The exception to this being any point-to-point interface or loopback.
`non-broadcast`	If selected it must also be used together with the `neighbor` setting in the OSPF router context, i.e, on NBMA links this should be set to send unicast Hello.
`point-to-point`	This is a network type where the interface is a link between exactly two routers. A packet sent from one router will always have just one recipient. In auto mode, this will be the default for any point-to-point interface type, e.g. GRE tunnel interfaces.

`[no] auth <md5 KEYID | plain> [hash] <SECRET>`

Description

Configure authentication of OSPF messages for this interface.

Two different authentication methods can be selected:

- *MD5:* Uses a `KEYID` of 0-255 and a `SECRET` text string of 8-16 characters (Both the `SECRET` and the `KEYID` must be the same on neighbour routers).
- *Plain:* Uses a `SECRET` text string of 4-8 characters (The `SECRET` must be the same on neighbour routers).

> **NOTE**
>
> Please note, only one MD5 key currently supported.

Default: Disabled

Options

Option	Description
`no`	Disable authentication of OSPF messages on the interface.

`[no] retransmit-interval <1-65535>`

Description

Set time between retransmitting lost link state advertisements in seconds for this interface.

Default: 5 seconds

continues on next page

continued

`[no] retransmit-interval <1-65535>`

Options

Option	Description
`no`	Reset to the default value.

`[no] transmit-delay <1-65535>`

Description

Set link state transmit delay in seconds for this interface.

Default: 1 second

Options

Option	Description
`no`	Reset to the default value.

Status

The current status of a OSPF configured router can be observed in the exec context in the following manner:

```
OSPF Routing Process, Router ID: 20.0.0.2
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millisec(s)
Minimum hold time between consecutive SPFs 50 millisec(s)
Maximum hold time between consecutive SPFs 5000 millisec(s)
Hold time multiplier is currently 1
SPF algorithm last executed 7m31s ago
Last SPF duration 24 usecs
SPF timer is inactive
LSA minimum interval 5000 msecs
LSA minimum arrival 1000 msecs
Write Multiplier set to 20
Refresh timer 10 secs
This router is an ASBR (injecting external routing information)
Number of external LSA 1. Checksum Sum 0×0000eba6
Number of opaque AS LSA 0. Checksum Sum 0×00000000
Number of areas attached to this router: 1
Area ID: 0.0.0.0 (Backbone)
  Number of interfaces in this area: Total: 2, Active: 2
  Number of fully adjacent neighbours in this area: 2
  Area has no authentication
  SPF algorithm executed 7 times
  Number of LSA 8
  Number of router LSA 4. Checksum Sum 0×00016360
  Number of network LSA 4. Checksum Sum 0×00024d06
  Number of summary LSA 0. Checksum Sum 0×00000000
  Number of ASBR summary LSA 0. Checksum Sum 0×00000000
  Number of NSSA LSA 0. Checksum Sum 0×00000000
```

```
    Number of opaque link LSA 0. Checksum Sum 0x00000000
    Number of opaque area LSA 0. Checksum Sum 0x00000000
```

In addition we could also check the OSPF neighbours:

```
Router-A:/#> show ip ospf neighbour

Neighbour ID   Pri State        Dead Time Address    Interface       RXmtL RqstL DBsmL
10.0.14.1       1 Full/Backup    36.044s 10.0.1.2   vlan1:10.0.1.1  0     0     0
10.0.12.1       1 Full/DR        37.909s 10.0.2.1   vlan2:10.0.2.2  0     0     0
```

We can also check the OSPF database in the following way:

```
Router-A:/#> show ip ospf database

       OSPF Router with ID (10.0.11.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.0.11.1       10.0.11.1       1434 0x8000000c 0x3463 3
10.0.12.1       10.0.12.1       1675 0x8000000c 0x8709 3
10.0.13.1       10.0.13.1       1406 0x8000000b 0xf296 3
10.0.14.1       10.0.14.1       1425 0x80000011 0xbdc6 3

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.1.1        10.0.11.1       1444 0x80000004 0x1de8
10.0.2.1        10.0.12.1       1675 0x80000004 0xec19
10.0.3.1        10.0.13.1       1666 0x80000004 0xf011
10.0.4.2        10.0.13.1       1436 0x80000004 0xf10c
```

Routes

The purpose of OSPF is to ensure that we learn routes to reach all of our intended destinations. To check the routes that the are known and what routes are active we can use the `show ip route` command as such:

```
example:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O   10.0.1.0/24 [110/10] is directly connected, vlan1, 19:45:35
C>* 10.0.1.0/24 is directly connected, vlan1
O   10.0.2.0/24 [110/10] is directly connected, vlan2, 19:45:35
C>* 10.0.2.0/24 is directly connected, vlan2
O>* 10.0.3.0/24 [110/20] via 10.0.2.1, vlan2, 19:44:55
O>* 10.0.4.0/24 [110/20] via 10.0.1.2, vlan1, 19:41:00
O   10.0.11.0/24 [110/10] is directly connected, vlan3, 19:45:35
C>* 10.0.11.0/24 is directly connected, vlan3
O>* 10.0.12.0/24 [110/20] via 10.0.2.1, vlan2, 19:44:55
O>* 10.0.13.0/24 [110/30] via 10.0.1.2, vlan1, 19:41:00
  *                       via 10.0.2.1, vlan2, 19:41:00
O>* 10.0.14.0/24 [110/20] via 10.0.1.2, vlan1, 19:41:00
```

Any route denoted with an “O” is a route that has been learnt by OSPF. In this case we can see that some routes that OSPF have learnt are not used. The reason for this is that in this case also have connected routes to the same destination. A connected route has a lower administrative distance and will be selected over the OSPF route.

If we are interested in isolating the routes that OSPF have knowledge of we can use the `show ip ospf route` command:

```
Router-A:/#> show ip ospf route
============ OSPF network routing table ============
N    10.0.1.0/24           [10] area: 0.0.0.0
                           directly attached to vlan1
N    10.0.2.0/24           [10] area: 0.0.0.0
                           directly attached to vlan2
N    10.0.3.0/24           [20] area: 0.0.0.0
                           via 10.0.2.1, vlan2
N    10.0.4.0/24           [20] area: 0.0.0.0
                           via 10.0.1.2, vlan1
N    10.0.11.0/24          [10] area: 0.0.0.0
                           directly attached to vlan3
N    10.0.12.0/24          [20] area: 0.0.0.0
                           via 10.0.2.1, vlan2
N    10.0.13.0/24          [30] area: 0.0.0.0
                           via 10.0.1.2, vlan1
                           via 10.0.2.1, vlan2
N    10.0.14.0/24          [20] area: 0.0.0.0
                           via 10.0.1.2, vlan1

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

2.10.5. Routing Information Protocol

Introduction

The Routing Information Protocol (RIP) is a dynamic routing protocol. RIP is an example of a distance vector routing protocol, and historically it has been one of the most widely used intra-domain unicast routing protocol within the Internet.

RIP is relatively simple to setup, but does not handle topology changes as rapidly as the OSPF dynamic routing protocol. Therefore, OSPF is generally preferred over RIP when it is possible to select dynamic routing protocol.

For information on OSPF, see the OSPF configuration page.

Overview

RIP enabled routers will exchange routing information with their neighbours periodically. This in order to construct their own routing tables, with entries describing how to reach each of the participating routers.

In **Figure 1** below, we have an example network of 5 RIP enabled routers. Each RIP router only needs to know the networks where other RIP enabled routers are located. Afterwards they will automatically construct their routing tables, no need for configuring static routing entries.

```
                 |                         |
             .---+---.                 .---+---.
             |       |                 |       |
             |   R   |                 |   R   |
             |       |                 |       |
             '---+---'                 '---+---'
                 |.1                       |.2
-------------+------------+-------------+-------------
 10.0.1.0/24             |.3
                      .---+---.
                    .1|       |.1
                  +---+  R1   +---+
                  |   |       |   |
                  |   '-------'   |
 10.0.2.0/24      |               |       10.0.3.0/24
-------------+----+---         ---+-----+-------------
             |.2                       |.2
             .---+---.                 .---+---.
             |       |                 |       |
             |   R   |                 |   R   |
             |       |                 |       |
             '---+---'                 '---+---'
                 |                         |
```

Figure 1: Showing a router R1 connected to other routers across three different networks. With RIP running on each of the routers they will automatically learn routes to each other.

In order for a RIP router to find the best path, between the source and destination network, it utilises hop count as a metric. The hop count is simply the amount of routers between the source and destination network. Therefore, when adding entries to the routing table, the path with the lowest hop count will be selected. For more information on routing table entries, see the Static Routing section.

> **NOTE**
>
> RIP has a maximum hop count of 15. This means that any destination that is more than 15 routers away from the source will be unreachable. One reason for this limitation is to prevent routing loops.

There are two different versions of RIP v1 and v2. One big difference is that *RIPv2* routers exchange routing information using IP multicast (IP address 224.0.0.9). While *RIPv1* uses broadcast to exchange routing information.

Authentication

RIPv2 allows authentication of RIP messages. This can be helpful to avoid that false routing information is injected into your network (deliberately or by mistake). This is configured for each `iface` using the `auth` command.

Route Redistribution

It is possible to redistribute routing information learnt externally (OSPF, con- nected routes or static routes) inside the RIP routing domain, using the `redistribute` command.

You can also let a RIP router inject a default route (0.0.0.0/0) into your RIP domain, using the `distribute-default` command.

Passive Interface

In some situations you may wish to include a router's subnets as part of the RIP routing domain without running RIP on the associated network interface. This can be done using the `passive` command under each `iface`.

Configuration

The configuration of RIP is separated into two different parts. The actual router configuration and RIP related settings for each individual interface.

Router

RIP is configured under the `router` section of the global configuration context:

```
example:/#> configure
example:/config/#> router
example:/config/router/#> rip
example:/config/router/rip/#>
```

Syntax

`[no] version <1|2>`

Description

Set the RIP version to use.

Default: version 2.

continues on next page

continued

`[no] version <1|2>`

Options

Option	Description
`no`	Reset to the default setting.

`[no] distance <1-255>`

Description

Set protocol distance.

Default: 120.

Options

Option	Description
`no`	Reset to the default setting.

`[no] timers [update SEC] [invalid SEC] [flush SEC]`

Description

Control RIP protocol timers.

Defaults:

- `update` 30 seconds.
- `invalid` 180 seconds.
- `flush` 120 seconds.

Options

Option	Description
`no`	Reset to the default settings.
`SEC`	Time value in whole seconds.

`[no] passive-interface`

Description

Control if RIP should be passive on all interfaces by default.

Use the network command, or the per interface “passive” command to activate RIP on an interface. The neighbour command can also be useful.

Default: `no passive-interface`, i.e, RIP is disabled on all interfaces.

Options

Option	Description
`no`	Reset to the default setting.

`[no] neighbor <IPADDR[,IPADDR,...]>`

Description

Explicitly define a RIP neighbour.

This can be useful when the neighbour in question cannot handle IP multicast and thus cannot receive RIP routing information. When explicitly defining a RIP neighbour it will receive routing updates over IP unicast.

Options

Option	Description
`no`	Remove configured RIP neighbours.
`IPADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1

`[no] network <IFACE|NETWORK[,NETWORK,...]>`

Description

Define RIP enabled networks.

Options

Option	Description
`no`	Remove all configured RIP networks.
`IFACE`	Enable RIP for a given interface, e.g. `vlan1`.
`NETWORK`	IP address in standard quad-dotted notation including netmask, e.g. 192.168.1.1/24.

`[no] redistribute <connected|static|ospf>`

Description

Redistribute external routing information.

- Use “redistribute connected” to redistribute routes to directly attached networks, which have not been defined to belong to any RIP area in the `network` command.
- Use `redistribute static` to redistribute static routes set with the `ip route` command.
- Use `redistribute ospf` to redistribute OSPF routes.

Redistribution of the default route is handled separately with the `distribute-default` command.

Default: Disabled for all three alternatives.

Options

Option	Description
`no`	Disable redistribution.

`[no] distribute-default`

Description

Inject default route, i.e., announce network 0.0.0.0/0.

Default: Disabled.

Options

Option	Description
`no`	Disable distribution of the default route.

Interface Specific Settings

To configure RIP related settings for an interface is located under the `iface` context:

```
example:/#> configure
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> rip
example:/config/iface-vlan1/rip/#>
```

Syntax

`[no] passive [auto]`

Description

Control passive-interface setting for RIP on this interface.

Default: Auto. Follows `router rip` `passive-interface` setting.

Options

Option	Description
`no`	Disable.
`auto`	Use same setting as system wide RIP setting.

`[no] split-horizon [poisoned-reverse]`

Description

Enable, or disable, split horizon on this interface.

Split horizon is a method that helps preventing routing loops. The principle is that a router will not advertise a route back to the interface it was learned from.

Default: Enabled.

Options

Option	Description
`no`	Disable.
`poisoned-reverse`	Actively advertise routes as unreachable over the interface which they were learned.

`[no] send-version <1,2>`

Description

RIP version to send on this interface.

Default: Auto. Follows `router rip` `version` setting.

Options

Option	Description
`no`	Resets to the default setting.

`[no] receive-version <1,2>`

Description

RIP version to receive on this interface.

Default: Auto. Follows `router rip` `version` setting.

Options

Option	Description
`no`	Resets to the default setting.

`[no] auth <md5 KEYID | plain> [hash] <SECRET>`

Description

Configure authentication of RIP messages on this interface.

Default: None.

Options

Option	Description
`no`	Remove any configured authentication.
`md5`	Use md5 cryptographic authentication. The `SECRET` when using md5 are text strings of 4-32 characters. The `KEYID` is a key identifier (0-255) associated with MD5 keys.
`plain`	Use a clear-text password as authentication. The `SECRET` when using plain text are text strings of 4-16 characters.

Status

The current status of a RIP configured router can be observed in the exec context in the following manner:

```
example:/#> show rip
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 17 seconds
```

```
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:
  Default version control: send version 2, receive version 2
    Interface        Send  Recv   Key-chain
    vlan1            2     2
  Routing for Networks:
    vlan1
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    10.0.2.1                 0         0       120   00:00:10
  Distance: (default is 120)
```

We can also see any potential routes that RIP have added using `show ip route` as such:

```
example:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

R>* 10.0.1.0/24 [120/2] via 10.0.2.1, vlan1, 00:02:08
C>* 10.0.2.0/24 is directly connected, vlan1
```

Any route denoted with an `R` is a route that have been learned by RIP.

2.10.6. Static Multicast Routing

About

This document gives an overview of the concepts around IP multicast routing and how to set up and debug static multicast routes.

Introduction

Multicast is an efficient data distribution mechanism for purposes of reaching more than one receiver. IP multicast applications, such as a camera or sensor, can send one packet only to reach many receivers. Switches and routers in the network infrastructure duplicate the packet as it makes its way to each receiver.

A receiver of multicast is said to be a *subscriber to a group*, a group is an IP address. In IPv4 the entire 224.0.0.0/4 block is reserved for multicast, i.e., 224.0.0.0 – 239.255.255.255. However, not all address are available to the end-user.

In the reserved IPv4 multicast range, a small subnet, 224.0.0.0/24 or 224.0.0.*, is further reserved for control protocols, e.g., IGMP, RIPv2 and OSPF.

Like regular IP addresses, IP multicast groups must be translated to Ethernet (LAN) MAC addresses. However, the range of reserved MAC multicast addresses is too small. For example group 225.1.2.3 and 226.1.2.3 both map to 01:00:5e:01:02:03. See RFC 1112[1] for details. This limitation may be a problem in switched networks where the switch fabric only supports filtering on MAC level.

On a per LAN basis (layer-2) IP multicast is managed by IGMP (routers) and IGMP Snooping (switches). Managing multicast on this level is important due to its inherent broadcast nature. Knowledge of this can be very important when debugging multicast (re)distribution and routing.

Routing of IP multicast can be done either dynamically (e.g., DVMRP, PIM) or statically. WeOS supports both PIM and static multicast routing.

Overview

Compared to unicast routing, multicast routing uses a separate routing table and is handled a little bit differently. To be able to route multicast you need the following:

- Enable IP forwarding
- Enable IP multicast forwarding
- A multicast route, (S,G) or (*,G)
- Multicast traffic, where packets have a TTL > 1

The two enable flags simply control routing and multicast routing, respectively. However, if IP forwarding is disabled toggling the multicast forwarding flag will have no effect.

A multicast route is made up of a *group*, an optional prefix length of the group, an *inbound interface*, an optional *source address* and one or more *outbound interfaces*. There can be at most 256 multicast routes with at most 32 outbound interfaces per route.

Multicast groups may not overlap for the same source interface or a mix of any and a specific source interface. The only exception is a source specific /32 and a sourceless /32 route for legacy purposes.

The source, or *sender address*, is optional but the underlying Linux kernel still require a source address to be able to route the traffic. A dedicated multicast routing daemon manages this by adding rules to the kernel on-demand based on the *source-less* rules specified. For each new multicast stream, from a given group and inbound interface, the routing daemon checks to see if a

[1] https://tools.ietf.org/html/rfc1112

matching mroute rule exists and then adds that source specific rule to the kernel. This may cause some initial delays in activation of such rules.

Notation:
- (S,G) - source-specific
- (*,G) - source-less

Configuration

To set up a source-specific route of group 225.1.2.3 from sender 192.168.2.42 from vlan1 to vlan2 and vlan3:

```
example:/#> configure ip
example:/config/ip/#> forwarding
example:/config/ip/#> multicast-forwarding
example:/config/ip/#> mroute group 225.1.2.3 source 192.168.2.12 in vlan1 out vlan2 vlan3
example:/config/ip/#> leave
```

To set up a source-less route of another group:

```
example:/#> configure ip
example:/config/ip/#> forwarding
example:/config/ip/#> multicast-forwarding
example:/config/ip/#> mroute group 225.3.2.1 in vlan1 out vlan2 vlan3
example:/config/ip/#> leave
```

```
example:/#> configure ip
example:/config/ip/#> forwarding
example:/config/ip/#> multicast-forwarding
example:/config/ip/#> mroute group 225.3.5.0/24 in vlan1 out vlan2 vlan3
example:/config/ip/#> leave
```

To setup up a multicast route that tracks a VRRP instance(only forwards packets if the vrrp instance is in Master state):

```
example:/#> configure
example:/config/#> router
example:/config/router/#> vrrp vrrp1
example:/config/router/vrrp-vrrp1/#> iface vlan1
...
example:/config/router/vrrp-vrrp1/#> end
example:/config/router/#> end
example:/config/#> ip
example:/config/ip/#> forwarding
example:/config/ip/#> multicast-forwarding
example:/config/ip/#> mroute group 225.3.2.2 track vrrp1 in vlan1 out vlan2 vlan3
example:/config/ip/#> leave
```

Such a multicast route may not overlap without another multicast route. This means that they can not have same group and the same input interface as another multicast route.

Status

To verify that the route is correctly set:

```
example:/#> show ip mroute

(*,G) Template Rules
ROUTE (S,G)                    IIF                PACKETS      BYTES  OIFS
(*, 225.3.2.1)                 vlan1                    0          0  vlan2 vlan3
(*, 225.3.5.0/24)              vlan1                    0          0  vlan3 vlan2
(*, 225.3.2.2)                 vlan1                    0          0  vlan3 vlan2

(S,G) Rules
ROUTE (S,G)                    IIF                PACKETS      BYTES  OIFS
(192.168.2.12, 225.1.2.3)      vlan1                    0          0  vlan3 vlan2

Kernel MFC Table
ROUTE (S,G)                    IIF                PACKETS      BYTES  OIFS
(192.168.2.12, 225.1.2.3)      vlan1                    0          0  vlan3 vlan2
```

Worth noting here is that the (*,G) route acts as a template. When the sender, here *192.168.2.42*, is discovered a dedicated (S,G) rule is added automatically. The packet counters for (*,G) routes will therefore always be zero.

2.10.7. Protocol Independent Multicast - Sparse Mode (PIM-SM)

Introduction

Protocol Independent Multicast (PIM) is a family of IP multicast routing protocols. While this family include a number of different variants, this document focus specifically on the **Sparse Mode** (SM) variant.

> **NOTE**
>
> The system currently only supports PIM-SM.

The central aspect of PIM, regardless of what mode is being utilised, is the **Protocol Independent** part and what it entails. The PIM protocol itself contains no method or functionality that allow it to perform any from of topology discovery. Instead, PIM relies on the system's underlying unicast routing table in order to perform Reverse-Path Forwarding (RPF). It is **Independent** because it does not matter what unicast routing protocol is used by the device to populate its unicast routing table. The unicast routes could be provided by, for instance, using OSPF or RIP. The routes could even by provided by the means of static routes.

For an introduction and overview to multicast in general, refer to the **Static Multicast Routing** document provided in the link below.

> **SEE ALSO**
>
> Related Documentation
> - Static Multicast Routing
> - Static Routing
> - OSPF
> - RIP
> - IGMP

> **SEE ALSO**
>
> Related Examples and Use-Cases
> - PIM-SM Basic

Overview

As mentioned PIM is a family of multicast routing protocols that have a few different variants, but this document considers the Sparse Mode only.

As the name hints at PIM **Sparse Mode** is suitable for use in network topologies where any interested receivers, of any given multicast group, will be **sparsely** scattered throughout the network. What this really means is that it is expected that the majority of the subnets in the network have no interest in receiving most of the multicast traffic, at least not at the same time.

PIM-SM generally scales pretty well in larger network topologies, as long as the above mentioned condition holds, that most of the subnets in the network are not interested in all of the relevant multicast groups at the same time.

TIP

In direct opposite, to the scenarios that PIM-SM is suitable for, its sibling PIM Dense Mode (PIM-DM) is instead the method that is preferred if you have a smaller scale network with multicast receivers that are *densely* populated.

However, the operating system currently does not support PIM-DM.

In **Table 1** directly below, a number of terms and concepts that are generally used when talking about PIM-SM are briefly summarised. It is good to quickly glance over these, if the reader is interested and/or new to PIM. However, note that this document is not intended to give a full deep dive description of all the inner workings of PIM-SM. It is intended to provide a high level basic understanding of the general characteristics and concept.

Term	Shorthand	Description
Rendezvous Point	RP	The central point for a given multicast group, or group range, within a PIM domain. Any receiver or sender for a given multicast group will essentially meet at the RP.
First-Hop Router	FHR	The first PIM router that is attached to the source of a specific multicast stream. This device starts the register process for any stream that enters on it.
Last-Hop Router	LHR	The last PIM router that is connected to any receiver that is interested in a specific multicast group.
Designated Router	DR	The router responsible for forwarding PIM Join messages towards the RP on any given multi-access network.
Multicast Route	mroute	An IP multicast route, consisting of a specific group and possible source IP of said group. It also specifies one inbound interface and one, or more, outbound interface(s).
S-G mroute	(S,G) mroute	This is a source-specific multicast route, where **S** is the source IP of the sender and **G** is the IP Multicast group, e.g. `(192.168.1.1,225.1.2.3)`.
Star-G mroute	(*,G) mroute	This is a source-less multicast route, where ***** indicates that the source IP of the sender could be any address, i.e. it is a wildcard, e.g. `(*,225.1.2.3)`.
Shared Tree, RP Tree or (*,G) Tree	RPT	The Shared Tree for a specific multicast group originates from the RP, and it has the receiver DR as its leaf node. Each PIM router part of the RPT has (*,G) mroutes installed for the specific group, with the inbound being the interface

Term	Shorthand	Description
		towards the RP and the outbound towards the receiver DR.
Shortest-Path Tree or (S,G) Tree	SPT	The Shortest-Path Tree originates from the router directly connected to the source of a given multicast group. A SPT can be formed between the source and the RP and also all the way to the LHR. Each router along the path of the SPT has (S,G) mroutes installed for the given group. The SPT represent the best path that a stream can take from its source to reach any receivers. It is possible that said path intersects along the RPT, but not necessary.
Reverse-Path Forwarding	RPF	Reverse-Path Forwarding is performed by consulting the unicast routing table in order to find the expected interface towards a specific source IP address. This is for instance used in order to find the path to the RP or to the source of a specific multicast stream.
Downstream	-	Direction towards the multicast **receiver(s)**.
Upstream	-	Direction towards the multicast **sender(s)**.

Table 1: Brief overview and description of some relevant terms and concepts central to PIM-SM.

Control Messages

PIM-SM utilise a number of different control messages in order to function. In **Table 2** below the most relevant types of messages are briefly summarised:

Message	Destination	Description
Hello	All PIM Routers (224.0.0.13)	Hello messages are transmitted on all the configured PIM interfaces on the device. The messages serve to announce to the other PIM routers on the subnet that the source sender is also a PIM router. The Hello messages are also used to elect which of the routers on the subnet should serve as the DR.
Register	Rendezvous Point (Unicast)	Message sent towards the RP from a FHR in order to inform it that a specific multicast stream is now present. The register frame contain the entire multicast frame encapsulated within it. If the RP has any receivers that have joined for the specific group, the encapsulated frame will be forwarded towards them.

© 2026 Westermo Network Technologies AB

Message	Destination	Description
NULL Register	Rendezvous Point (Unicast)	A special register frame that is sent from the FHR towards the RP, the purpose of this frame is to inform the RP that a specific multicast stream is still available. A NULL Register frame will not encapsulate the multicast frame.
Register-Stop	FHR, source of the Register (Unicast)	Sent by the RP back towards the originating FHR router, when it is no longer necessary to continue sending register frames towards the RP.
Join/Prune (J/P)	All PIM Routers (224.0.0.13)	Sent by a PIM router to indicate if a specific multicast group wants to be received (Joined) downstream or if it should not be received anymore (Pruned). It is normal to refer to simply Join or Prune as individual message types, even though they are in fact a single type of message. A Join/Prune message could be either a (*,G) Join/Prune, for the RPT, or a (S,G) Join/Prune, for the SPT.

Table 2: A list of some relevant PIM control messages.

IGMP messages are also vital for the operation of PIM. The PIM join process will be initiated when a PIM router receives an IGMP Membership Report (IGMP Join). When received, the PIM router will generate a (*,G) Join message and send it along the RPT towards the RP.

When an IGMP Leave is received on a PIM router a (*,G) Prune will be generated instead, and sent on the path towards the RP.

Reverse-Path Forwarding

PIM relies on unicast routing to perform RPF in order to figure out what interface some PIM control messages will be transmitted on and to verify that multicast frames enter on expected interfaces. This can be referred to as a RPF Interface.

In other words, an underlying unicast routing setup is required for PIM to be able to function. As mentioned earlier, the independent part of Protocol *Independent* Multicast is that any unicast routing protocol can be used, even static routes. As long as the routing table is populated correctly so the RPF can be performed.

> **TIP**
>
> The operating system support both the OSPF and RIP unicast routing protocols.

> **WARNING**
>
> A default route cannot be used to perform a RPF lookup. Instead provide a specific static route if this is necessary.

RPF is used in order to handle PIM Join/Prune messages to build Shared Trees (RPT) and Shortest-Path Trees (SPT):

- The RPT is constructed by means of (*,G) Join/Prune messages. The router will find the outbound interface based on the source address for the RP. The source of the RP will always be known, since each PIM router needs to have it configured. When the outbound interface is determined, the Join will be transmitted on that interface.
- The SPT is constructed by means of (S,G) Join/Prune messages. For a (S,G) Joins/Prune the location of the source is known. So in this case an RPF lookup is performed for said source in order to determine the outbound interface where the Join will be transmitted.

RPF is also performed in order to ensure that multicast traffic from a given source enter the device on the expected interface, in order to prevent any multicast routing loops. If a multicast frame enter on an unexpected interface the frame is discarded. The check is slightly different depending on the tree state of the router:

- **SPT:** If the router has a (S,G) mroute installed for the source of the given multicast frame, the RPF check is performed on the source address of said multicast frame.
- **RPT:** If the router only has a (*,G) mroute installed matching the group of the received multicast frame, the RPF check is performed on the IP address of the RP.

Interfaces

Each PIM router in the network must specify all the interfaces that PIM should operate on. All the following interfaces must be known to PIM:

- Interfaces connected to other PIM routers, so that they can form neighbour relationships with each other.
- Interfaces connected to any potential multicast receiver.
- Interfaces connected to any potential multicast sender.
- The interface where the RP is located (The interface with the IP address corresponding to the configured RP IP address), if the device is intended to serve as a RP.

> **NOTE**
>
> Interfaces facing receivers must also have IGMP enabled. This is a separate setting from the `interface` option that exist in the PIM Router configuration. Ensure that it is enabled on relevant interfaces.

Currently the system allows the following interface types to be used as PIM Interfaces:

- VLAN
- Loopback (lo)
- GRE

Neighbour Formation

Each PIM router transmit PIM Hello messages to the link-local multicast address 224.0.0.13 (All PIM Routers), in order to inform other potential PIM routers on the same network that it exist. These

messages are transmitted continually on a set interval that is default 30 seconds, but can be adjusted if desired.

Any PIM router that receives a PIM Hello frame will add the source sender of the message as a PIM neighbour. There is no acknowledge process or anything similar. A device will keep a neighbour until it expires or as long as it continues to receive Hello messages. As shown in **Figure 2** below, all the PIM routers on the same subnet will consider each other as neighbours, not just the directly adjacent router.

```
Neighbors in 10.0.0.0/24                                  Neighbors in 10.1.0.0/24
. . . . . . . . . . . . . . . . . . . . . . . . . . .    . . . . . . . . . . . . . . .
.                                                .    .                            .
. .--------.             .--------.             .--------.             .--------.  .
. |        | 10.0.0.0/24 |        | 10.0.0.0/24 |        | 10.0.1.0/24 |        |  .
. |   R1   +-------------+   R2   +-------------+   R3   +-------------+   R4   |  .
. |        | .1       .2 |        | .2       .3 |        | .3       .4 |        |  .
. '--------'             '--------'             '--------'             '--------'  .
.                                                .    .                            .
. R1 Hello  ------------> -------> ------------>  .    .                            .
. R2 Hello  <------------          ------------>  .    .                            .
. R3 Hello  <------------ <------- <------------  .    .  ------------>            .
. R4 Hello                                       .    .  <------------             .
. . . . . . . . . . . . . . . . . . . . . . . . . . .    . . . . . . . . . . . . . . .
```

Figure 2: A group of PIM routers and their neighbour relationships. The routers R1, R2 and R3 in subnet 10.0.0.0/24 all form neighbours with each other, not just the directly adjacent routers. Router R3 also form a neighbour relationship with router R4 located in another subnet 10.1.0.0/24.

The neighbours for a subnet elect one of them to serve as the Designated Router (DR) and be the router responsible for the communication with the RP. For instance, if any of the routers have a connected device requesting a multicast stream it is the responsibility of the DR to send a PIM Join for that group towards the RP.

The election process of the DR is based on the priority that is included in the Hello message. By default the value is 1, but can be adjusted if desired. If the best priority value is shared by multiple neighbours the DR is elected based on the router with the highest IP address.

Modes of Operation

PIM-SM also have a few different modes of operation:

- **Any Source Multicast (ASM):** Is as the name suggests, a mode where multicast streams originating from any source can be received and handled within the PIM domain. When ASM is used the configuration of Rendezvous Points becomes necessary. In this mode it is also not necessary for the potential receivers to know the source of the senders for interested multicast streams.
- **Source Specific Multicast (SSM):** In this mode it is required that all the receivers know beforehand where the sender of a specific multicast group is located. When SSM is used it is not necessary to configure any RPs within the PIM domain. However, IGMPv3 is mandatory on the multicast clients because it includes support for (S,G) IGMP reports (join).

One way to look at the differences between SSM and ASM, is that with SSM the receiver knowns exactly what it wants to receive and from who. But with ASM the receiver only know what it wants to receive, but not necessarily who is the one that will provide it.

> **NOTE**
>
> As of right now the operating system supports only the ASM mode.

PIM-SM With Any Source Multicast

In PIM-SM ASM the potential receivers does not need to know the sources of any of the multicast that they may want, the receivers can simply join a group and if any traffic for that group is being transmitted, it will be forwarded towards the receiver.

However, as mentioned earlier, in the ASM mode one ore more Rendezvous Points need to be specified. All routers that constitute the PIM domain must be aware of the location of the RP. What this means is that they must be configured in the same manner in this regard.

In **Figure 1** below we have an example representation of a simple PIM-SM ASM setup. With R4 acting as the RP for the PIM domain. All the routers will be individually configured to know that the router R4 is the acting RP.

```
          .-----. (Receiver)            .-----. (Receiver)
          | H1 |                        | H2 |
          '-+--'                        '--+-'
            |                              |
            |                              |
(LHR) .---+----.                     .----+---. (LHR)
 (DR) |        |                     |        | (DR)
      |   R1   +---------------------+   R2   |
      |        |                     |        |
      '---+----'                     '----+---'
          |                              |
          |                              |
          |                              |
          |                              |
          |                              |
(FHR) .---+----.                     .----+---. (RP)
      |        |                     |        |
      |   R3   +---------------------+   R4   |
      |        |                     |        |
      '---+----'                     '--------'
          |
          |
          .-+--. (Sender)
          | S1 |
          '----'
```

Figure 1: A simple example of a PIM-SM setup using the ASM model.

Various other settings can also be adjusted depending on the specifics of a given setup, but may not be necessary. However, the above mentioned must always be configured for operation to ensue.

Receiver Joins

When a receiver wants to join a given group it will start by sending a IGMP Membership Report indicating that it wants traffic from the specific group. This IGMP Report will find its way to the receiver-side DR for the specific subnet, and it will become the LHR for that specific group.

The now LHR will first add a (*,G) mroute entry to its multicast routing table, where the outbound interface will be the interface the IGMP Membership report was received on, and the incoming interface being the RPF interface for the RP.

The next step is that the LHR will generate a PIM Join message that will be transmitted towards the RP for the given group. This message will be forwarded hop by hop until it reaches the RP. Each router along this path will also install a (*,G) mroute into its multicast routing table. Similar to the LHR, the outbound interface will be the receiving interface of the Join, and the inbound will be the RPF interface towards the RP. A simplistic illustration of this process can be seen in **Figure 3** below.

```
                                                   (DR)
(RP)                      (Intermediate)           (LHR)
.--------.                .--------.               .--------.
|        | vlan1    vlan2 |        | vlan1    vlan2 |        | vlan1          .------.
|   R4   +----------------+   R2   +----------------+   R1   +----------------+  H1  |
|        |                |        |                |        |                '------'
'--------'                '--------'                '--------'

          <---------------          <---------------          <---------------
           PIM Join/Prune            PIM Join/Prune            IGMP Report
           Join: (*,225.1.2.3)       Join: (*,225.1.2.3)       Join: 225.1.2.3


                          Add mroute on R2:         Add mroute on R1:
                           (*,225.1.2.3)             (*,225.1.2.3)
                            in vlan2 out vlan1        in vlan2 out vlan1
```

Figure 3: A simplified overview what happens when host H1 start to request a specific multicast group 225.1.2.3. From the initial IGMP Report that reaches the receiver side DR, PIM Join messages will be generated and transmitted towards the known RP. Along the way (*,225.1.2.3) mroutes are installed in routing tables of the routers, this forms the RPT.

In the example provided by **Figure 3** no mroute is installed on the RP. In this case the multicast stream for 225.1.2.3 is not yet active and known by the RP. If the stream was already active and known by the RP it would install a (S,G) mroute for 225.1.2.3, with the inbound interface being set to the one where the stream is registered, and the outbound interface to vlan1 (Towards the receiver).

All of these installed (*,G) routes on the path from the LHR to the RP together constitute the so called RPT (Shared Tree). As can be seen in **Figure 4** the RPT is always rooted at the RP with the DRs, with active receivers, as the leaf nodes. The example setup is similar to the one presented earlier in **Figure1**.

```
          .-----. (Receiver)             .-----. (Receiver)
          | H1 |                         | H2 |
          '-+--'                          '--+-'
            |                               |
            |                               |
(LHR) .---+----.                     .----+---. (LHR)
 (DR) |        |                     |        | (DR)
      |   R1   +---------------------+   R2   |
      |        |                     |        |
      '---+----' <----------------- '----+---'
          |             RPT          ▲    |
          |                          |    |
          |                     RPT  |    |
          |                          |    |
          |                               |
(FHR) .---+----.                     .----+---. (RP)
      |        |                     |        |
      |   R3   +---------------------+   R4   |
      |        |                     |        |
      '---+----'                     '--------'
          |
          |
          .-+--. (Sender)
          | S1 |
          '----'
```

Figure 4: Example how the RPT looks when the host H1 has requested a specific group. The RPT is always rooted at the RP that is responsible for the given group that the receiver has joined.

Multicast Stream Registration

The goal of the multicast stream registration process is to inform the relevant RP that a specific multicast stream is now present and available on the network for any potential receivers.

The process is started when a FHR receives a multicast frame from a sender on any of its configured PIM interfaces. The FHR encapsulates the multicast data in a PIM Register frame, this frame is subsequently sent towards the RP as a unicast frame, as shown in **Figure 5**. As any other PIM router the FHR also knowns the IP address of the relevant RP beforehand.

```
(Source)                (FHR)                    (RP)
                        .--------.               .--------.
.------.                |        |               |        |
|  S1  +----------------+   R3   +----------------+   R4   |
'------'                |        |               |        |
                        '--------'               '--------'
        --------------►          --------------►
        Multicast Stream         Register (UC)
```

Figure 5: Simplified view how the register process is started. The sender S1 starts to transmit a multicast stream. When this stream enters the FHR R3 it will send a PIM Register frame as unicast towards the RP.

When the RP receives a PIM register frame two scenarios are possible. Either a receiver is already requesting the specific multicast group downstream, or it is currently not requested by anyone potential receiver.

Stream Already Requested Downstream

When the RP receives the Register message it will decapsulate the message to obtain the included multicast frame. If the destination group of said frame is already requested downstream the RP will transmit the frame along the RPT towards the intended receiver.

At the same time the RP will add a (S,G) mroute to its routing table. The RP then sends a PIM (S,G) Join towards the source, the FHR, for the multicast stream. Each router along the way to the FHR install a (S,G) mroute in their routing tables. Together this will constitute the SPT, for the specific sender and the group, between the sender and the RP. An example of this is shown in **Figure 6**.

```
(Source)                (FHR)                    (RP)
                        .--------.               .--------.
.------.          vlan2 |        | vlan1    vlan2 |        | vlan1   225.1.2.3 joined
|  S1  +----------------+   R3   +----------------+   R4   +-------- Downstream
'------'                |        |               |        |
192.168.1.99            '--------'               '--------'
                                   ◄-----------------
                                    PIM Join/Prune
                                    Join: (192.168.1.99,225.1.2.3)


                        Add mroute on R3:        Add mroute on R4:
                         (192.168.1.99,225.1.2.3)  (192.168.1.99,225.1.2.3)
                         in vlan2 out vlan1        in vlan2 out vlan1
```

Figure 6: The RP sends a (S,G) Join back towards the source, so that all the routers on that path adds a (S,G) mroute in their routing tables. Doing this creates the SPT between the source and the RP.

When the SPT has been constructed multicast traffic will start to flow and be routed towards the RP. Therefore, the encapsulated multicast frames will no longer be needed. So when the RP receives the multicast stream along the SPT it will send a Regsiter-Stop message towards the FHR, instructing it to stop sending encapsulated register frames.

When the FHR have received a Register Stop frame it will start a Register Stop timer. Before this timer expires the FHR will send a NULL Register towards the RP. If the current state is to be maintained the RP will send another Register-Stop message back. If the timer is allowed to expire, the FHR will again start to encapsulate the multicast traffic in Register messages.

> **WARNING**
>
> Currently a problem exist where the encapsulated frames may not be decapsulated and forwarded correctly from the RP. What this means is that the multicast traffic will not start to flow until the RP has sent the (S,G) Join towards the sender, that will allow it to add a (S,G) route towards the RP for the concerned group. Since the RP perform this action as soon as it receives a register for a group that is requested downstream, the traffic should start to flow almost immediately anyway.

Stream Not Currently Requested

When the RP receives the Register frame but there are currently no interested receivers for the encapsulated multicast frame, the RP will drop the frame, since there is no RPT to send it along. At the same time the RP will directly send a Register-Stop frame towards the source FHR, to instruct it that it should not continue to send any encapsulated Register frames.

When the FHR receives the Register-Stop the multicast stream will be dropped at the FHR. If a receiver requests the specific group at this point the RP will send a PIM Join directly towards the FHR so that the SPT can be constructed and the traffic can start to flow.

During this time the FHR will periodically send NULL Register frames towards the RP, in order to inform it that the stream is still active. If the stream disappears the Register will timeout after the Keep alive Time expires.

Switchover to SPT on LHR

Initially when multicast traffic arrive at the LHR it will do so over the RPT. However, this may not be the optimal path for the multicast stream to take through the network. Therefore a so called SPT switchover is performed. An example of this is presented in **Figure 7** below.

```
                    .-----. (Receiver)            .-----. (Receiver)
                    | H1 |                        | H2 |
                    '-+--'                        '--+-'
                      |      (S,G) Prune             |
                      |      ----------------►       |
                  .---+----.                    .----+---.
                  |        |                    |        |
                  |   R1   +--------------------+   R2   |
                  |        |                    |        |
                  '---+----' ◄---------------- '----+---'
                |     |    ▲        RPT         ▲    |     |
                |     |    |                    |    |     |
(S,G) Join      |     |    | SPT            RPT |    |     | (S,G) Prune
                |     |    |                    |    |     |
                ▼     |                         |    ▼
                  .---+----.                    .----+---.
                  |        |                    |        |
                  |   R3   +--------------------+   R4   |
                  |        |                    |        |
                  '---+----'                    '--------' (RP)
                      |
                      |
                    .-+--. (Sender)
                    | S1 |
                    '----'
```

Figure 7: The SPT switchover process initiated from the LHR R1 towards the FHR R3. R1 will send a (S,G) Join towards R3 to create the SPT. It will then also send a (S,G) Prune to stop the transmission of the multicast stream along the RPT

Before the switchover can be completed the SPT needs to be constructed. In order to do this the LHR sends a PIM (S,G) Join message towards the source. The outbound interface for this (S,G) Join is obtained by a Reverse-Path Forwarding lookup. This process is initiated the instant the LHR receives any multicast traffic along the RPT.

The SPT between the source and the LHR is constructed in the same manner as the SPT between the Source and the RP, described earlier. Each router along the path will install a (S,G) mroute in their routing tables, where the incoming interface will be the RPF interface towards the Source, and the outbound the interface the Join was received on.

Subsequently, when multicast traffic starts to flow along the newly established SPT a PIM (S,G) Prune message will be sent along the RPT towards the RP. This should stop the multicast stream from also being transmitted along the RPT, so that it now only uses the SPT instead.

> **NOTE**
>
> As long as `spt-switchover` is enabled the SPT will always be created this way, even if it is along the RPT.

At this point the traffic is flowing as efficiently as it can by using the shortest path through the network from the source to the receiver.

Configuration

Configuration of PIM-SM is primarily performed under the router context, containing most of the options. Further, PIM-SM also possess a number of settings related to the individual interface(s) it is set to operate on.

> **WARNING**
>
> Due to an existing problem, it is currently not possible to use a configuration that combines any static multicast routes with PIM. If both are configured at the same time, PIM will take precedence.

Enabling Multicast Forwarding

In order for any multicast routing to be useable on the system, including PIM-SM, `multicast-forwarding` must be enabled. This setting is disabled by default and is enabled in the following manner:

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> multicast-forwarding
example:/config/ip/#>
```

> **NOTE**
>
> If the setting flooding of unknown multicast is turned off, forwarding of multicast may not work properly. The recommended setting for flooding of unknown multicast is ‘auto’, for more info see flooding unkown multicast.

> **NOTE**
>
> It is still possible to configure PIM-SM, but if `multicast-forwarding` is not enabled a valid functional PIM-SM configuration will NOT be generated and applied.

Router Configuration

Configuration of PIM-SM is performed within the `pim` sub-context located under the `router` section in the global configuration context.

```
example:/#> configure
example:/config/#> router
example:/config/router/#> pim
example:/config/router/pim/#>
```

> **NOTE**
>
> Currently, since only **Sparse Mode (SM)** is the available mode, no other mode can be selected for PIM. It will therefore always default to be SM.

`[no] enable`

Description

Enable or disable pim.

Default: Enabled

Example

```
example:/config/router/pim/#> enable
example:/config/router/pim/#> no enable
```

Options

Option	Description
`no`	Use to disable PIM.

`[no] rendezvous-point ADDR [GROUP/LEN]`

Description

Configure static Rendezvous Point(s) (RP).

A necessity for PIM to be able to operate is that at least one Rendezvous Point (RP) exist, and that its location is specified here. The RP is necessary in order for the PIM routers to know where to send any Join messages and Register messages towards.

Currently the only supported manner in which RPs can be configured is using the static approach.

continues on next page

continued

`[no] rendezvous-point ADDR [GROUP/LEN]`

In order to configure a static RP the IP address where the RP is located must be provided. Further, it is possible to specify a network prefix for specific multicast groups to match on for this RP.

If no network prefix is provided for multicast group matching, any will match, i.e. it would be the same as 224.0.0.0/4.

It is possible to provide up to 128 different RP specifications.

> **NOTE**
>
> When static RPs are utilised it is important that the configuration is the same for all of the PIM routers that constitute the PIM domain.

Default: None

Alias: `rp`

Example

```
example:/config/router/pim/#> rendezvous-point 10.0.1.1 225.1.2.0/24
example:/config/router/pim/#> rendezvous-point 10.0.2.1
```

Options

Option	Description

`no`	Remove all configured rendezvous points:
	`example:/config/router/pim/#> no rendezvous-point`
	Alternatively, remove a specifically configured rendezvous point, providing either the full value or the numeric ID of the entry.
	`example:/config/router/pim/#> no rendezvous-point 10.0.1.1 225.1.2.0/24`
	`example:/config/router/pim/#> no rendezvous-point 1`

`ADDR`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.
`GROUP/LEN`	IP multicast group in standard quad-dotted notation including netmask in CIDR notation, e.g. 225.1.2.0/24.

`[no] interface [IFACE [IFACE...]]`

Description

Specify the interface(s) on which to enable PIM to operate on.

PIM must be enabled to operate on any interface that faces any multicast sources (senders) or destinations (receivers). The interface on which the Rendezvous Point (RP) is configured must also be specified.

continues on next page

continued

`[no] interface [IFACE [IFACE...]]`

Any intermediate PIM router, that does not directly face a multicast sender or a receiver, must also specify the interfaces where its PIM neighbours will be located.

This setting will NOT enable receiving of IGMP reports, or querying on the provided interfaces. This configuration is handled individually for each specific VLAN, under each individual VLAN context.

More PIM related interface configuration options can be found under the configuration context for each individual interface, under the PIM sub-context.

> **NOTE**
>
> The selection of at lest one PIM interface is mandatory in order to activate the service.

Default: None

Alias: `iface`

Example

```
example:/config/router/pim/#> interface vlan1
example:/config/router/pim/#> interface vlan1 vlan2
```

Options

Option	Description
`no`	Remove all or a specifically configured interface.
`IFACE`	Name of the interface, e.g. vlan1.

`[no] passive-interface`

Description

Control if PIM should be passive on all interfaces by default.

This setting simply controls what the default, auto mode, will mean for the individual pim interface(s). Each interface can have this value explicitly set under their own PIM configuration sub-context.

If passive-interface is enabled here it means that no PIM control messages should be handled by the interface(s) by default, i.e. no sending or receiving of any PIM control messages.

Default: `no passive-interface`, i.e. handling of PIM control messages is the default behaviour on all configured interfaces.

Example

```
example:/config/router/pim/#> passive-interface
example:/config/router/pim/#> no passive-interface
```

continues on next page

continued

`[no] passive-interface`

Options

Option	Description
`no`	Disable passive interface by default for any configured interface that has not explicitly adjusted it.

`[no] keepalive-timer [1-65535]`

Description

Set the timeout value for a (S,G) multicast stream.

This is the timeout value, in seconds, for how long the knowledge of a detected (S,G) multicast stream will be kept alive, even in the absence of any (S,G) join messages.

When a PIM router receives a specific (S,G) frame this value is the period of time it will be kept alive, even if no more frames are received. As long as the (S,G) stream is active and frames are received the keep alive timeout will be restarted with this value.

> **NOTE**
>
> If a value below 31 seconds is selected, be aware that some hardware platforms cannot see data flowing in better than 30 seconds chunks.

Default: 210 seconds

Example

```
example:/config/router/pim/#> keepalive-timer 100
```

Options

Option	Description
`no`	Reset the value of the setting back to its default.

`[no] rp-keepalive-timer [1-65535]`

Description

Set the RP specific timeout value for a (S,G) multicast stream.

Similar to the `keepalive-timer`, but specific to the RP. Instead of being started when a (S,G) frame is received, it is started when the RP sends a Register-Stop towards the DR that sent a register for a specific (S,G) stream.

It is important to note that the keepalive timer on the RP must be at least the value of the calculated Register-Stop time. If it is not the RP may age out the group before the next NULL-Register arrives.

continues on next page

continued

`[no] rp-keepalive-timer [1-65535]`

The keepalive time for the group is set to the maximum value between the `keepalive-timer` and `rp-keepalive-timer`, when the RP sends a Register-Stop towards the DR that registered the group.

> **NOTE**
>
> If a value below 31 seconds is selected, be aware that some hardware platforms cannot see data flowing in better than 30 seconds chunks.

Default: 185 seconds

Example

```
example:/config/router/pim/#> rp-keepalive-timer 100
```

Options

Option	Description
`no`	Reset the value of the setting back to its default.

`[no] register-suppression-time [11-65535]`

Description

Set the time value for Register Suppression.

This value is used to calculate the time during which a First Hop Router (FHR) will stop transmitting Register frames towards the RP, after it has received a Register-Stop frame from the RP.

The resulting Register-Stop timer will be set to a random value within the following interval:

- `((0.5*register-suppression-time),(1.5*register-suppression-time))-5`

In order to ensure that this timing value is not calculated to be less than zero, we limit the minimum value that can be set to 11 seconds.

Initially the Register frame from the FHR will contain the multicast payload encapsulated in the frame. But after it has received a Register-Stop from the RP, the subsequent Register frame will be a so called Null-Register. This is the same as a Register, but it will not contain any encapsulated multicast payload.

Default: 60 seconds

Example

```
example:/config/router/pim/#> register-suppression-time 40
```

continues on next page

continued

`[no] register-suppression-time [11-65535]`

Options

Option	Description
`no`	Reset the value of the setting back to its default.

`[no] join-prune-interval [1-65535]`

Description

Set the transmission interval of periodic Join/Prune frames.

A PIM router will periodically send Join/Prune messages toward its upstream PIM neighbours, in order to keep each other updated in regard to which groups are requested (Joined) or not (Pruned) by receivers downstream.

> **NOTE**
>
> The hold time for Join/Prune messages is not possible to configure individually. It will always be based on this value and it will be set to 3.5x this value, i.e. by default it will be 210 seconds.

Default: 60 seconds

Example

```
example:/config/router/pim/#> join-prune-interval 40
```

Options

Option	Description
`no`	Reset the value of the setting back to its default.

`[no] spt-switchover`

Description

Enable or disable spt-switchover.

This setting can be relevant, if no Shortest-Path Tree (SPT) switchover is wanted on the last hop router. If disabled only the shared tree (RPT) will be used instead. In other words, no (S,G) routes will be installed on any of the routers on the RPT towards the Rendezvous Point (RP).

When SPT switchover is enabled the last hop router will try to find the optimal, i.e. shortest, path towards the multicast source. If a better path exist this will be used instead of the shared tree (RPT).

The SPT switchover will be initiated on the Designated Router (DR) on the receiver side, as soon as the first multicast frame is received. The DR will at this point send a (S,G) join message towards the source of the multicast frame. Any router along the way will install a (S,G) route in their routing tables, together these routers make up the SPT.

continues on next page

continued

`[no] spt-switchover`

During the switchover process, the DR will start to receive frames from both the RPT and SPT. In order to fully switch to the SPT a prune will be sent towards the RP, informing it that it will not need the specific (S,G) traffic from the RPT anymore.

> **NOTE**
>
> The SPT switchover can only be disabled on a LHR. The RP will still perform the SPT switchover towards the upstream source regardless of this setting.

Default: Enabled

Example

```
example:/config/router/pim/#> spt-switchover
example:/config/router/pim/#> no spt-switchover
```

Options

Option	Description
`no`	Disable SPT switchover.

`[no] register-policy [permit|deny ADDR/LEN [,ADDR/LEN ...]]`

Description

Specify if a specific source IP is permitted or denied.

When a PIM router receives a register frame it will perform a lookup of the source address for the frame, against this list of permitted or denied source addresses.

If the source address of the register frame matches any permit entry in the list it will be allowed, and normal operation will ensue. On the other hand, if the source address of a register frame matches a deny entry, the register will be denied. Further, a Register-Stop frame will be sent towards DR, from which the register originated from.

> **NOTE**
>
> If there are no entries defined in the list no lookup can or will be preformed, hence any source will be accepted.

> **NOTE**
>
> If no prefix len is provided for the address it will be treated as a /32 address.

Default: None

continues on next page

continued

`[no] register-policy [permit|deny ADDR/LEN [,ADDR/LEN ...]]`

Example

```
example:/config/router/pim/#> register-policy permit 192.168.1.0/24
example:/config/router/pim/#> register-policy deny 192.168.1.10/32, 192.168.1.20/32
```

Options

Option	Description
`no`	Remove all or a specific entry.
`ADDR/LEN`	IP address in standard quad-dotted notation including netmask in CIDR notation, e.g. 192.168.1.1/24.

`[no] ssm-policy [permit|deny GROUP/LEN [,GROUP/LEN ...]]`

Description

Specify a range of group addresses where PIM will permit or deny SM over.

One or more Group ranges can be specified, indicating to PIM if SM should be denied or permitted for any groups that fall within the range.

> **NOTE**
>
> If no prefix len is provided for the group it will be treated as a /32 address.

Default: None

Example

```
example:/config/router/pim/#> ssm-policy deny 224.1.2.3/30
example:/config/router/pim/#> ssm-policy deny 224.1.2.3/30, 224.3.2.1/32
```

Options

Option	Description
`no`	Remove all or a specific entry.
`GROUP/LEN`	IP multicast group in standard quad-dotted notation including netmask in CIDR notation, e.g. 225.1.2.0/24.

Interface Configuration

Interface settings that concern PIM-SM are found in the `pim` sub-context located under the configuration context for each individual `iface`:

```
example:/#> configure
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> pim
example:/config/iface-vlan1/pim/#>
```

> **NOTE**
>
> The interface must still be added as a PIM interface with the `interface` command in the PIM routing context.

`[no] priority <1-65535>`

Description

Designated Router (DR) election priority for this interface.

This setting can be useful to specify what device is the preferred DR for a given LAN segment.

The higher the number the higher the priority, i.e. 10 is a higher priority than 1.

Default: 1

Example

```
example:/config/iface-vlan1/pim/#> priority 10
```

Options

Option	Description
`no`	Reset the value of the setting back to its default.

`[no] hello-interval <1-65535>`

Description

Set the pim hello interval for the iface.

This interval dictates how often PIM hello messages will be transmitted for the specific interface. These hello messages are what PIM uses in order to form PIM neighbours.

Default: 30 Seconds

Example

```
example:/config/iface-vlan1/pim/#> hello-interval 24
```

Options

Option	Description
`no`	Reset the value of the setting back to its default.

`[no] hold-interval <1-65535>`

Description

Set the PIM hold interval for the iface.

continues on next page

continued

`[no] hold-interval <1-65535>`

The hold interval specifies the time a neighbour learned from a hello message will be kept, if not more hello messages are received. Each time a new hello message is received this timer is restarted.

It is important that this value is not lower than the hello interval. If it is the neighbour will be aged out before a new hello message will arrive. The default ratio between the hello and hold interval is that the hold time is 3.5x times the hello interval.

Default: 105 Seconds

Example

```
example:/config/iface-vlan1/pim/#> hold-interval 84
```

Options

Option	Description
`no`	Reset the value of the setting back to its default.

`[no] passive [auto]`

Description

Control passive-interface setting for PIM on this interface.

If this option is enabled the interface will act passively in regard to PIM control messages, i.e. no control messages will be sent or received on the interface.

Default: `auto`

Example

```
example:/config/iface-vlan1/pim/#> passive
example:/config/iface-vlan1/pim/#> no passive
example:/config/iface-vlan1/pim/#> passive auto
```

Options

Option	Description
`no`	Disable acting as a passive interface.
`auto`	Follow the global setting `passive-interface` located in the PIM router context.

Status

A number of status commands are available that can be used to query the status of PIM. They are all accessed in the `ip` sub-context under the global exec context:

```
example:/#> ip
example:/ip/#>
```

All the PIM status commands are always prefixed with `show pim`. To see all available commands press tab after the prefixed line:

```
example:/ip/#> show pim
assert                  join                    rpf
assert-internal         jp-agg                  secondary
assert-winner-metric    local-membership        state
bsm-database            mroute                  statistics
bsr                     neighbour                upstream
bsrp-info               nexthop                 upstream-join-desired
channel                 rendezvous-point        upstream-rpf
interface               rp-info
```

Examples

Examples of a few different status commands.

Interface

Check the status of all the configured PIM interfaces:

```
example:/ip/#> show pim interface

PIM Interfaces
INTERFACE  STATE  ADDRESS       PIM-NBRS  PIM-DR    FHR  IF-CHANNELS
pimreg     up     0.0.0.0       0         local     0    0
vlan1      up     10.0.1.1      1         10.0.1.2  0    0
vlan2      up     10.0.4.1      1         10.0.4.3  0    0
vlan7      up     192.168.10.1  0         local     0    1
```

Column	Description
INTERFACE	The name of the interface.
STATE	The oper state of the interface.
ADDRESS	The current configured IP address of the interface.
PIM-NBRS	The number of PIM neighbours located on the interface.
PIM-DR	Who the current DR for the subnet the interface is part of is, where *local* signifies that the device itself is the DR for the subnet.

Neighbour

List the neighbours that have been established by the router:

```
example:/ip/#> show pim neighbour

PIM Neighbours
INTERFACE  NEIGHBOR  UPTIME    HOLDTIME  DR-PRIO
vlan1      10.0.1.2  01:10:16  00:01:28  1
vlan2      10.0.4.3  01:19:30  00:01:44  1
```

Column	Description
INTERFACE	The interface that the neighbour was located on, i.e. the interface that received the Hello message from the neighbour.
NEIGHBOR	The IP address where the neighbour is located.

Column	Description
UPTIME	The amount of time since the neighbour was initially detected.
HOLDTIME	The current hold time for the neighbour. If this expires the neighbour will be removed. The timer is reset every time a new Hello message is received from it.
DR-PRIO	The DR Priority reported by the neighbour in its hello message.

Rendezvous Point

Show the status of the configured rendezvous point(s):

```
example:/ip/#> show pim rendezvous-point

PIM Rendezvous Point Info
RP-ADDRESS   GROUP/PREFIX-LIST   OIF    I-AM-RP   SOURCE   GROUP-TYPE
10.0.2.4     224.0.0.0/4         vlan1  no        Static   ASM
```

Column	Description
RP-ADDRESS	The source IP address for the RPs location.
GROUP/PREFIX-LIST	The group or range of groups that the specific RP serves.
OIF	The outgoing interface to reach the RP.
I-AM-RP	Indicator if the router itself is the RP.
SOURCE	How the RP has been assigned. Static indicates that it was statically configured on the device.

Join/Prune

Show the current status of Join/prune for any multicast groups that have some intended receiver that has joined:

```
example:/ip/#> show pim join

PIM Join
INTERFACE   ADDRESS    SOURCE          GROUP       STATE   UPTIME    EXPIRE   PRUNE
vlan1       10.0.4.3   192.168.30.99   225.2.2.2   JOIN    00:31:57  03:13    --:--
```

Column	Description
INTERFACE	Interface that the J/P was received on.
ADDRESS	Address of the interface.
SOURCE	The source in the J/P message. If it is a (*/G) J/P message the source will be shown as *****. If it is a (S/G) J/P message the source S will be displayed.
GROUP	The group that has been joined.
STATE	The state of the entry, for instance if it is Joined or Pruned.
UPTIME	How long since the initial J/P message for the entry was received.
EXPIRE	How long until the entry expires, if no J/P messages including it are received.

Upstream

Show the status of current upstreams:

```
example:/ip/#> show pim upstream

PIM Upstream
IIF     SOURCE          GROUP       STATE  UPTIME    JOINTIMER  RSTIMER   KATIMER   REFCNT
vlan1   *               225.2.2.2   J      00:03:55  00:00:58   --:--:--  --:--:--  1
vlan2   192.168.30.99   225.2.2.2   J      00:03:55  00:00:58   --:--:--  00:03:23  2
```

Column	Description
IIF	The incoming interface for traffic from the specific upstream source.
SOURCE	Source of the upstream.
GROUP	The multicast group for the upstream source.
STATE	The current state in regards to the upstream.
UPTIME	The total uptime of the upstream.
JOINTIMER	The timer indicating when a new J/P message will be transmitted, it will be sent when it reaches 0.
RSTIMER	The Register suppress timer. indicates when the FHR will send a new register frame to inform the RP that the stream is still active it will be sent when it reaches 0.
KATIMER	The Keep Alive timer. Indicating how long the upstream will be kept alive, even if it disappears.
REFCNT	The reference counter for the upstream.

List installed mroutes

Show the currently installed mroutes on the router:

```
example:/ip/#> show pim mroute
IP Multicast Routing Table
Flags: S - Sparse, C - Connected, P - Pruned
       R - SGRpt Pruned, F - Register flag, T - SPT-bit set
SOURCE          GROUP       FLAGS  PROTO  INPUT  OUTPUT  TTL  UPTIME
*               225.2.2.2   SC     IGMP   vlan1  pimreg  1    00:23:16
                                   IGMP          vlan7   1
192.168.30.99   225.2.2.2   ST     STAR   vlan2  vlan7   1    00:23:15
```

Column	Description
SOURCE	Displays a source address if the route entry is a (S,G) mroute. Will display a ***** if the route is a (*,G) mroute.
GROUP	The multicast group to route for the mroute.
FLAGS	Flags describing additional information for the state of the mroute.
INPUT	The incoming interface for the multicast stream.
OUTPUT	The outgoing interfaces for the route, any matching multicast to be routed will be sent on these interfaces.
UPTIME	The total time that the specific route have been installed in the multicast routing table.

Reverse-Path Forwarding

Show the current status how the RPF would be selected for a (S,G) entry:

```
example:/ip/#> show pim rpf
RPF Cache Refresh Delay:    50 msecs
RPF Cache Refresh Timer:    0 msecs
RPF Cache Refresh Requests: 6
RPF Cache Refresh Events:   4
RPF Cache Refresh Last:     00:37:39
Nexthop Lookups:            10
Nexthop Lookups Avoided:    0

PIM RPF
SOURCE          GROUP      RPFIFACE  RPFADDRESS  RIBNEXTHOP  METRIC  PREF
*               225.2.2.2  vlan1     10.0.1.2    10.0.1.2    20      110
192.168.30.99   225.2.2.2  vlan2     10.0.4.3    10.0.4.3    20      110
```

Column	Description
SOURCE	The specific source address to use when performing the RPF lookup in the unicast routing table. If the source is a ***** the address of the RP will be used.
GROUP	The relevant multicast group.
RPFIFACE	The determined interface to use in order to reach the source.
RPFADDRESS	The IP address of the RPF.
RIBNEXTHOP	The nexthop address determined to use in order to reach the source.

Troubleshooting

Multicast routing can be tricky sometimes and when it comes to PIM-SM there are a number of things that could cause problems getting the setup to work as intended. The following is a list, in no particular order, of common things that could be worth checking if any problems are occurring:

- Has the underlying unicast routing been configured correctly on all of the routers constituting the PIM domain? If it is not the Reverse-Path Forwarding (RPF) can potentially face some problems.
  - If using OSPF or RIP are the networks for the senders and receivers included? The redistribute connected command could be used for this.
  - Check that the RPF does not try to rely on default routes, since this will not work. In this case use a more specific static route instead.
  - Is the unicast routing consistent throughout the network? If it is not, multicast traffic may be dropped. If multicast traffic enter on the wrong interface, according to the interface determined by the RPF, it will dropped in order to avoid potential routing loops.
- Have all the required interfaces been specified as PIM interfaces?
- Is multicast routing enabled on the device? By default this is disabled.
- Is IGMP enabled on the interfaces facing any receivers? This is not the same as configuring an interface as a PIM interface.
- Is the Rendezvous Point configuration consistent across all of the PIM routers within the PIM domain, i.e. do they specify the same RPs?
- Is the multicast traffic that is being transmitted done so with the correct TTL? This is a common issue when trying to route any multicast traffic.

2.10.8. Hardware offloading for Routing

Introduction

On certain WeOS products routing is offloaded to hardware, which means that network packets do not reach IP software stack, and as the result gives enhanced performance.

Each hardware running WeOS has different capabilities for Offloading ranging between Non-existent to high capabilities.

Viper-TBN

As an example, consider a train setup with consist ports X1-X4 on vlan1, and backbone ports X5-X8 on vlan2. In such case IP forwarding of unicast, multicast and 1-1 NAT between interfaces vlan1 and vlan2 would be offloaded.

Further Details

- *IPv4:* Only forwarding of IPv4 traffic can be offloaded, while e.g. ARP packets are handled by the CPU.
- *Only VLAN network interfaces and physical ports:*
  - Only traffic coming in on physical Ethernet ports can be offloaded. Traffic coming in on virtual ports or interfaces such as L2 SSL VPN interfaces or GRE tunnels can not be offloaded.
  - Only traffic going out through a network interface *vlan1*, *vlan2*, etc. and then through a physical Ethernet port associated with that VLAN can be offloaded. The exception to this is traffic routed to a blackhole interface; such traffic is dropped in hardware too.
  - For IP multicast traffic, the incoming Ethernet port must be associated with a VLAN interface. For unicast traffic the incoming port could either be associated with a VLAN or be an *isolated port interface* (see VLAN configuration for further details on isolated port interfaces).
- *IP TTL:* Only IP packets with TTL of 2 or higher will be offloaded.
- *Non-fragmented packets:* Only non-fragmented IP packets will be offloaded.

Redfox and Lynx-5000

As of this release these product has limited Offloading Capabilities and therefore, is only recommended to be enabled in production systems after rigorous system testing.

Limitations

- Routed IPv6 traffic is handled by the CPU
- IP multicast traffic will be routed by the CPU
- Firewall forwarding chain will not impact any routed Unicast traffic
- NAT will not be performed on any routed traffic
- Only VLAN interfaces can be used, usage of Port-interfaces will not perform any traffic forwarding
- Policy-Based Routing will not function
- SSL-tunnel will function to some extent via the CPU but is strongly discouraged from being used in this release

To Enable/Disable Offloading

The default setting for offloading is *disabled*. Hardware offloading is enabled using the following command:

`[no] offload`

Description

Enable or Disable CPU offloading

Options

Option	Description
`no`	Disable offloading

```
example:/#> configure ip
example:/config/ip/#> offload
example:/config/ip/#> leave
```

2.10.9. Policy-Based Routing (PBR)

Introduction

Policy-Based Routing (PBR) allows for more specific routing decisions based on specified policies, defined by the administrator. Utilizing PBR can allow the user to route traffic throughout the network in a more precise manner, when the situation deem it necessary or advantageous.

> **SEE ALSO**
>
> Related Documentation
> - Static Routing

> **SEE ALSO**
>
> Related Examples and Use-Cases
> - Policy-Based Routing
> - Multiple Upstream Failover

Overview

When it comes to regular routing, whether it being statically configured routes or dynamically obtained, the routing decision is based on the destination address of the specific frame that is being routed. When performing policy-based routing, a routing decision can be based on additional criteria.

The system allows for configuration of multiple different PBR instances. Each of these instances consist of a next hop gateway address, along with a number of match options that are combined to create the policy that specify the conditions any given frame needs to fullfil in order to select the specified route.

Policy Matching Options

Each PBR instance need to specify how the specific route can be selected. This is achieved by specifying a number of match rules that constitute the policy used for mapping routing bound frames towards the associated next hop address. When configuring a policy-based route, at least one match rule always needs to be provided.

In **Table 1** below, the options that are currently available in the system for the policy definitions are presented. These values may be present in any frame that is bound for routing on the device. If a given frame matches a defined policy rule, an attempt will be made to route the frame towards the policy's associated next hop address.

Match Type	Description
saddr	Match frames on the source IP address field in the IP header.
daddr	Match frames on the destination IP address field in the IP header.
sport	Match frames on the source port.
dport	Match frames on the destination port.

Table 1: The currently available match types that are useable for defining the match rule that will constitute the policy for any given policy-based route instance.

As an example, it is possible to perform a specific routing decision based on the originating source address. This would allow for different routing decisions for different hosts, even though the destination might still be the same.

> **NOTE**
>
> If multiple match types are used in the configuration of the same policy-based route instance, a single frame will need to correctly match all of the values provided, i.e. it is **NOT** sufficient to partially match a policy.

A unique match type can only be used once per PBR instance. If a specific rule needs to be applied more than one time for the same intended next hop gateway, a new PBR instance needs to be added to handle that case.

Incoming Interface

In addition to specifying the match rules for a given PBR instance, an in-interface also needs to be selected. Hence, the match rules will only ever be checked for frames that ingress on the provided interface. Therefore, it is important that the administrator correctly selects the indented ingressing interfaces when configuring a PBR instance.

For each individual PBR instance it is possible to select one or multiple different interfaces. In addition to this, it is also possible to **not** specify a specific in-interface. In this case, all available and applicable interfaces will be selected for the PBR instance.

> **NOTE**
>
> A specific interface can be set as the in-interface for multiple different PBR instances.

Locally Originated Traffic

It is possible to have a PBR instance operate only on locally originated traffic. This can be done by specifying the `lo` interface as the in-interface. This can allow the user to direct traffic from the device itself towards a specific route when the matching criteria is fulfilled.

Routing Selection and Interaction With Regular Routes

In order to have a better understanding how policy routes function in the system and how the selection works, it is good to mention how it works for regular routes. Normally each route that is added, whether statically or dynamically, will be inserted into the main default routing table. The selection of the table is based on a set of selection rules. By default there is an implicit rule that states *from all lookup main*, indicating that any frame will always utilise the main default routing table.

When a policy-based route is added, it will add another rule, that will be considered before the implicit default rule. The conditions of this rule will be based on the match options that were configured for the PBR instance. Hence, if a frame matches the rules added from a PBR definition, the frame will be instructed to use another routing table. It is in this specific table that the next hop route for the specific PBR instance have been inserted. If multiple different PBR instances specify the same next hop, they will refer to the same specific routing table.

```
ip rules                            routing tables
.----------------------.
|                      |            .-------------------------.
| PBR Instance 1 rules +----------> | PBR Instance 1 Next hop |
|                      |            '-------------------------'
|                      |
|                      |            .-------------------------.
| PBR Instance 2 rules +----------> | PBR Instance 2 Next hop |
|                      |            '-------------------------'
|                      |
|                      |            .-------------------------.
| from all lookup main |----------> | Static routes           |
|                      |            | OSPF routes             |
|                      |            | RIP routes              |
|                      |            | etc...                  |
'----------------------'            '-------------------------'
```

Figure 1: Simplified image on the relation between ip rules and table selection.

A policy-based route rule will always be considered before the implicit rule specifying the main default routing table. Therefore, any policy-based route will always be prioritised over any route located in the main routing table, if the routing bound frame in question matches the conditions for the specific policy-based route.

Policy-Based Route Preference

The ordering of the rules for each individual policy-based route definition can be adjusted using the preference value. Adjusting this value will affect the ordering of the rules that are added to the ip rules list. Utilisation of this option can allow the administrator to adjust the lookup order of the policies. Regardless of the the value of this setting, the policy-based routes will always come before the default ip rule.

If no specific preference value is provided, the preference value is considered automatic. Any policy-route definition with a preference specified will always be considered before those set to automatic.

For any policy-based route instance that have the same preference (Including if they are set to automatic), the ordering will be based on the instance id.

Configuration

Policy-based routes are configured from the IP configuration context:

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#>
```

When creating a policy route a sub-context will be entered:

```
example:/config/ip/#> policy-route
example:/config/ip/policy-route-1/#>
```

If an existing policy route needs to be adjusted, the specific ID for the given route needs to be provided. This can also be done if you want a new route to be created with a specific ID:

```
example:/config/ip/#> policy-route 5
example:/config/ip/policy-route-5/#>
```

TIP

The priority of the policy-based routes are not just adjusted using the ID of the policy route itself, it can be adjusted with the `preference` command for each specific route. Hence, the ordering of IDs is not only what is used to dictate the priority, unless all routes `preference` are set to `auto`.

NOTE

It is also important to ensure the IP Forwarding is enabled, by default this should be the case.

`[no] enable`

Description

Enable or disable the policy-based route instance.

Options

Option	Description
`no`	Disable the policy route instance.

`[no] next-hop [GATEWAY]`

Description

Specify the next hop IP address for frames matching the policy.

Just like a regular static route, a policy route specifies a specific next hop address. This next hop will only be usable if a frame matches the policy defined by the `match` command, in addition to ingressing on the interface specified by the `in-iface` command.

Example

```
example:/config/ip/policy-route-1/#> next-hop 192.168.1.99
```

Options

Option	Description
`no`	Remove the currently configured next-hop.
`GATEWAY`	IP address in standard quad-dotted notation, e.g. 192.168.1.1.

`[no] in-iface [IFACE [,IFACE ...]]`

Description

Specify the incoming interface(s) to consider for the route.

The policy-based route will only be considered for frames that ingress on the provided interface(s). If a frame enters on another interface, even though it may match the configured `match` rule(s), it will not be able to resolve the specific policy-based route.

continues on next page

continued

`[no] in-iface [IFACE [,IFACE ...]]`

A specific interface can be used in multiple different PBR instance definitions.

Default: `no in-iface` (Any)

Example

Some examples on how one, multiple or no specific (Any) `in-iface` can be specified.

```
example:/config/ip/policy-route-1/#> in-iface vlan1
example:/config/ip/policy-route-1/#> in-iface vlan1,vlan2
example:/config/ip/policy-route-1/#> no in-iface
```

Options

Option	Description
`no`	Set no specific incoming interface, meaning that any applicable interface will be used for the policy route.
`IFACE`	Name of the interface, e.g. vlan1.

`[no] preference [1-64]`

Description

Set the preference value of the policy-based route instance.

Each specific policy-based route match rule will be considered in order. For any rule that potentially can match several frames, this setting can be used to tune the ordering of how the rules are considered.

The lower the number the higher the preference for the specific entry, i.e. 1 is the highest preference and will be considered first.

If `no preference` is specified, the route will have a preference considered auto. The route will still have an internal order among all the other policy-based routes that are set to auto.

> **NOTE**
>
> A policy-based route with any preference value set will always be considered before any route that has its preference set to auto.

Default: `no preference` (Auto)

Options

Option	Description
`no`	Set no specific preference, meaning that the preference value will be auto.

`[no] match [saddr ADDR/LEN | daddr ADDR/LEN | sport PORT | dport PORT]`

Description

Set the match rule(s) specifying the policy how this route is selected.

Each policy route needs at lest one match rule to be specified, otherwise there is no policy for frames bound for routing to match against.

A unique match rule type can only be used one time per PBR instance. If a specific type of rule needs to be applied more than one time for the same target next hop, a new rule needs to be added to handle that situation.

Example

Some examples on how the various match options can be configured.

```
example:/config/ip/policy-route-1/#> match saddr 192.168.30.1/32
example:/config/ip/policy-route-1/#> match daddr 192.168.2.0/24
example:/config/ip/policy-route-1/#> match sport 67
example:/config/ip/policy-route-1/#> match dport 68
example:/config/ip/policy-route-1/#> show match

Match Rules
SADDR            DADDR            SPORT  DPORT
192.168.30.1/32  192.168.2.0/24   67     68
```

Options

Option	Description
`no`	Remove the configured match rules.

match

Description

A policy route definition can use the following conditions as match specifiers:

Options

Option	Description
`saddr`	Match frames on the source IP address field in the IP header.
`daddr`	Match frames on the destination IP address field in the IP header.
`sport`	Match frames on the source port.
`dport`	Match frames on the destination port.

Options

Option	Description
`ADDR/LEN`	IP address in standard quad-dotted notation including netmask in CIDR notation, e.g. 192.168.1.1/24.
`PORT`	Port number in range 1-65535.

Status Overview

In order to see the active status of configured policy routes the following command can be used:

```
example:/#> show ip policy-route
PRIO  SADDR           DADDR           SPORT  DPORT  IN-IFACE  NEXT-HOP
300   198.18.10.0/24  198.18.20.0/24  500    501    vlan1     10.0.0.1
```

> **NOTE**
>
> A policy-based route will not be shown using the regular `show ip route` command in the CLI. In order to check the status of any policy-based route, the `show ip policy-route` needs to be utilised.

Some additional advanced commands exist for troubleshooting any potential problems:

```
example:/#> show ip policy-route interface
show pbr interface
  vlan1(18) with pbr-policy vlan1-map

example:/#> show ip policy-route map
  pbr-map vlan1-map valid: yes
    Seq: 1 rule: 300
        Installed: yes Reason: Valid
        SRC Match: 198.18.10.0/24
        DST Match: 198.18.20.0/24
        Nexthop-Group: group-0
          Installed: yes Tableid: 10000

example:/#> show ip policy-route nexthop
show pbr nexthop
Nexthop-Group: group-0 Table: 10000 Valid: 1 Installed: 1
  Valid: 1 nexthop 10.0.0.1
```

2.11. Tunnels and VPN

2.11.1. GRE Tunnelling Guide

Introduction

This document provides an introduction to Generic Routing Encapsulation (GRE) tunnels and how to configure them.

GRE is an encapsulation method for tunnelling data packets over a routing protocol. It is specified in RFC 2784[1]. It can encapsulate arbitrary data packets over an arbitrary delivery protocol, but the most common use is to create an IP over IP tunnel. Which is also what is currently supported.

GRE works by adding a special (GRE) header in front of the encapsulated packet. It is a stateless protocol and it does not provide any security features at all; it lacks encryption and authentication, and it does not detect lost packets, replay attacks or other spoof attacks.

Secure tunnelling can be achieved by setting up GRE over an IPsec VPN.

A GRE tunnel is defined by its endpoints, the beginning and the end of the tunnel. An endpoint is defined by a routable address, defined by the delivery protocol, typically a public IP address. The following section describes a common example.

IP-in-IP Tunnel

GRE can be used as a generic IP-in-IP tunnel. E.g., if GW1 and GW2 are NAT gateways, a GRE tunnel can be used to tunnel traffic between local subnets, here 192.168.1.0/24 and 192.168.2.0/24. The tunnel endpoints 10.0.72.29 and 10.0.38.33 should be routable IP addresses over the shared intranet, or Internet.

```
                                  .--.-.
                                 ( (    )__
                                (_,  \ ) ,_)  Internet/Intranet
                                  '-'--`--'
              10.0.72.0/24      |    |        10.0.38.0/24
             .------------------'    '--------------------.
             |                                            |
             |.29                                         |.33
         .---+---.                                    .---+---.
         |       |            GRE Tunnel              |       |
         |  GW1  |===================================|  GW2  |
         |       |                                    |       |
         '---+---'                                    '---+---'
             |.1                                          |.1
             |     192.168.1.0/24      192.168.2.0/24     |
   ---+------+---+---------------      ---+----------+----+----
      |          |                        |          |
      |.11       |.5                      |.11       |.5
    .-+--.     .-+--.                   .-+--.     .-+--.
    | H1 |     | H2 |                   | H3 |     | H4 |
    '----'     '----'                   '----'     '----'
```

Figure 1: IP-in-IP tunnel with GRE.

The Protocol

GRE is a generic protocol intended to encapsulate any protocol. Currently only IP (v4 or v6) in IP (v4 and v6) is supported. The below figure illustrates the structure of a packet when IP is tunnelled through GRE over IP.

[1] https://tools.ietf.org/html/rfc2784

By default, GRE traffic is sent on a per-packet basis by consulting the routing table, like any other IP packet. It is, however, also possible to configure the GRE tunnel to only use a specific network interface.

```
.-----------------------------------------------------------------------------------------.
| GRE Delivery protocol - Outer IP Header | GRE Header | GRE Payload - Inner IP header and body |
|-----------------------------------------|------------|----------------------------------------|
| Dest IP | Src IP | TTL |      ...       |    ...     | Dest IP | Src IP | TTL |      ...      |
'-----------------------------------------------------------------------------------------'
```

Figure 2: Packet header and payload for IP in GRE over IP.

The IP addresses in the outer IP header are the addresses of the *endpoints* while the addresses in the inner IP header is the addresses of the sending and receiving hosts.

The TTL of the outer IP header (see Figure 1) is by default set equal to the TTL of the encapsulated IP packet. It is possible to configure a specific TTL for the outer header for each GRE tunnel.

The checksum field in the GRE header is an optional field in the protocol and is unset in packets transmitted by WeOS. However, if a checksum is included in received packets it is validated.

Configuration

GRE tunnels are configured in two steps:

1. First the tunnel is created and defined with its endpoints, which defines the route of the GRE packets
2. Then a new (GRE) network interface is created

The second step is to configure the GRE interface to enable routing, through the GRE tunnel, of the encapsulated IP packets.

Optional tunnel parameters will use default values if not configured. Tunnel endpoints must always be specified.

GRE Tunnel

```
example:/#> configure
example:/config/#> tunnel
example:/config/tunnel/#> gre N
example:/config/tunnel/gre-N#> local ADDR_LOC
example:/config/tunnel/gre-N#> remote ADDR_REM
```

`N`

Description

unique instance number for each GRE tunnel (0-255)

`ADDR_LOC`

Description

IP address for local endpoint

`ADDR_REM`

Description
IP address for remote endpoint

Optional: Configure the GRE tunnel allow traffic only via a specific network interface.

```
example:/config/tunnel/gre-N#> outbound IFACE
```

`N`

Description
unique instance number for each GRE tunnel (0-255)

`IFACE`

Description
network interface to be used for GRE tunnel traffic

Optional: Configure the TTL of the outer IP header

```
example:/config/tunnel/gre-N#> ttl TTL_OUT
```

`N`

Description
unique instance number for each GRE tunnel (0-255)

`TTL_OUT`

Description
TTL for the outer header of the GRE tunnel

Optional: Set GRE tunnel description.

```
example:/config/tunnel/gre-N#> description TEXT
```

`N`

Description
unique instance number for each GRE tunnel (0-255)

`TEXT`

Description
Free form description of this GRE tunnel

GRE Interface

Any GRE can have an IP address assigned to it. This can be done under the specified interface settings, see example.

```
example:/config/tunnel/gre-1#> end
example:/config/tunnel/#> end
example:/config/#> iface greN
example:/config/iface-greN/#>
```

`N`

Description
the corresponding instance number of the previously created GRE tunnel

We can use any supported method to assigning an IP address. In this example we will assign a static address.

```
example:/config/iface-greN/#> inet static ADDR[/LEN]
```

`N`

Description
the corresponding instance number of the previously created GRE tunnel

`ADDR`

Description
the IP address for the GRE interface

`LEN`

Description
the length of the netmask

2.11.2. OpenVPN Guide

Introduction

A VPN tunnel is a mechanism for sending private data over a public network. WeOS includes the open source OpenVPN suite to provide VPN tunnel services. More in-depth information about OpenVPN can be found at the website https://openvpn.net[1].

For example use-cases see the OpenVPN HowTos:
- OpenVPN LAN Bridging HowTo
- OpenVPN Host to LAN HowTo
- OpenVPN Internal Routes HowTo

The WeOS administration interface in many places refers to SSL when working with OpenVPN tunnels, such as the **show tunnel ssl** command. This is for legacy reasons. SSL (Secure Socket Layer) as a protocol is no longer in active use since it has been replaced by TLS (Transport Layer Security).

Client-Server Architecture

WeOS implements OpenVPN in a client-server, where the clients connect to the server. When used with *certificate authentication*, multiple clients can connect to a single server. In *pre-shared key authentication* mode, a single client per server is allowed.

It is possible to run multiple OpenVPN instances on a single WeOS device as long the instances use different TCP or UDP ports. If so, the device can act either as server or as client on a per-instance basis.

Peer Authentication Methods

WeOS supports two types of peer authentication with OpenVPN:
- Certificates
- PSK (Pre-Shared Key)

The keys or certificates must be imported to the WeOS certificate repository before they can be used in setting up a VPN tunnel.

It is possible for the server to require VPN clients to authenticate with username and password in addition to certificate/PSK. These user credentials can be verified against a remote authentication server (RADIUS) or in a local user database at the server (see WeOS AAA).

Configuration

This section provides more information about the various configuration options available for VPN tunnels. OpenVPN provides a plethora of options and WeOS supports a useful subset of these.

Most WeOS options translate to an OpenVPN option as documented in the option descriptions. When searching for an option in the OpenVPN documentation or manual page, prepend "–" to the option name to reduce the number of search hits. For example, search for "–remote" rather than "remote".

OpenVPN configuration context is entered via the tunnel configuration context:

```
example:/config/#> tunnel
example:/config/tunnel/#> ssl 1
example:/config/tunnel/ssl-1/#>
```

[1] https://openvpn.net

Common Options (Server & Client Mode, Certificate & PSK Authentication)

`[no] ssl ID`

Description

Create new OpenVPN instance, or enter the configuration context of an existing instance. `no ssl ID` removes configuration for an instance. Up to MAX_OPENVPN_INSTANCES instances can be configured.

`[no] enable`

Description

Activate, or deactivate, this OpenVPN instance. If deactivated, the rest of the OpenVPN configuration is kept.

`[no] description STRING`

Description

This is a short text description of the tunnel. The default is set to *ssl* followed by the tunnel ID (e.g. *ssl0*). The description is displayed when listing tunnels using CLI or Web. It is limited to 15 characters and may not contain spaces.

This is a WeOS-specific option (not passed to OpenVPN).

`[no] server`

Description

Set OpenVPN instance in “server” or “client” mode. A new tunnel endpoint is by default in server mode. A tunnel endpoint in client mode can be brought to server mode by issuing the **server** command in the tunnel configure context. Client mode is invoked by issuing the **no server** command.

`[no] method <cert|secret>`

Description

Select the authentication method for tunnel establishment: Certificate (i.e. TLS mode) or PSK (Pre-shared Key), where default is `method secret`.

The option must have the same value for both client and server.

In PSK mode, the corresponding OpenVPN option is named *secret*. In Certificate mode, *tls-server* or *tls-client*.

`[no] type <tap|tun>`

Description

Determines the protocol layer which is transmitted across the tunnel.

continues on next page

continued

`[no] type <tap|tun>`

- **tap** makes the tunnel endpoint a *tap* device which acts as a virtual port, receiving and transmitting link layer packets, including link layer multicast and broadcast packets.
- **tun** makes the tunnel endpoint a *tun* device which acts as an interface, receiving and transmitting IP packets.

The option must have the same value for both client and server.

The corresponding OpenVPN option is named *dev-type*.

`[no] protocol <tcp|udp|mptcp>`

Description
Protocol to encapsulate the tunnel traffic, TCP or UDP (default).

Both server and client must use the same value for this option.

The corresponding OpenVPN option is named *proto*.

`[no] port PORT`

Description
This option defines the TCP or UDP port on which the OpenVPN server will listen. This option needs to be configured if multiple servers should run on the same WeOS unit, or if the default port (1194) cannot be used.

Matching values have to be configured for client and server.

The corresponding OpenVPN option is named *lport*.

`[no] auth <sha1|sha256|md5>`

Description
Select the hash function to be used for data authentication. Defaults to `auth sha256`.

The hash is calculated over the encrypted package and combined with a key, which is the pre-shared key (in PSK mode) or auto-generated (in TLS mode), to produce a digital signature. The signature is then appended to the packet. The receiver performs the same calculation and compares the calculated signature to the received one. If they do not match, the package is dropped.

The corresponding OpenVPN option is named *auth*.

`[no] keepalive interval <SEC> restart <SEC>`

Description
When this option is enabled, OpenVPN will periodically ping the remote peer across the tunnel control channel at the specified *interval* (only if there is no other traffic being sent across the

continues on next page

continued

`[no] keepalive interval <SEC> restart <SEC>`

tunnel). This ping may serve to keep ports open in a stateful firewall as well as to detect a broken connection.

The second value represents the *restart* timeout - If no successful ping reply is received within this time, the tunnel will restart, causing any domain name lookups to be carried out again. This is useful when either or both peers are using dynamic IP addresses.

When configured on a server, the option values will also be pushed to the client(s).

The default configuration is for *keepalive* to be enabled, with a ping period of 10 seconds and a restart timeout of 60 seconds.

The corresponding OpenVPN option is named *keepalive*.

`[no] compression [adaptive]`

Description

If compression is enabled (`compression`), OpenVPN will try to compress all data to be sent across the tunnel. If the data is already compressed, this will be a waste of CPU cycles and compression should be disabled. If *adaptive* compression is set (`compression adaptive`), OpenVPN will periodically sample the compression ratio and disable compression for a period of time when the ratio is low.

Unless a client is configured with compression disabled (`no compression`), the server will push its compression settings (`compression` or `compression adaptive`) to the client.

The related OpenVPN options are *comp-lzo* and *comp-noadapt*.

Options Common to Server Mode (Certificate & PSK Authentication)

`[no] local-address <ADDRESS>`

Description

This server-only option allows specifying a single address that the OpenVPN server will listen for incoming connections on. If this option is not enabled, the server will accept connections on any address.

The corresponding OpenVPN option is named *local*.

`[no] bind-dev IFNAME`

Description

This server-only option allows specifying a single interfaces where the OpenVPN server will listen for incoming connections. If this option is not enabled, the server will listen on all available interfaces.

The corresponding OpenVPN option is named *bind-dev*.

`[no] aaa-method <remote-server ID | local ID>`

Description

This option defines how user authentication works in conjunction with OpenVPN. If enabled, it will cause the server to request clients to send access credentials (e.g. username and password), and will then ask WeOS to validate these before allowing the connection to succeed.

The configuration accepts either a local-db id or a server-id (RADIUS), which should be defined in advance in the AAA subsystem.

The option is server-only.

Options Common to Client Mode

`[no] peer <ADDRESS|DOMAIN>`

Description

This option identifies the remote peer by domain name or IP address. A client must set this option in order to inititate tunnel establishment to the server.

> **NOTE**
>
> Dynamic DNS (DDNS) comes in handy on the server side if the server acquires its IP address dynamically.

The corresponding OpenVPN option is named *remote*.

`[no] identity USERNAME password PASSWORD`

Description

This option is used to provide a user name and password to be used as an additional level of authentication.

This option is client-only and must be provided if the server is configured to require such authentication (See *AAA-Method*).

The corresponding OpenVPN option is named *auth-user-pass*

Options Only for PSK Authentication

Options common to Client and Server Mode when using PSK Authentication.

`[no] secret LABEL`

Description

When using PSK mode, this option is used to specify the key to use for data channel encryption/decryption as well as for authentication. (Both keys will be extracted from the same key file.)

continues on next page

continued

`[no] secret LABEL`

This option requires a static key which must have been previously imported into the WeOS device using the certificate import function. The same key must be present on both client and server.

`[no] crypto <bf-cbc|des-ede3-cbc|aes-128-cbc|aes-192-cbc|aes-256-cbc>`

Description

Select the cipher algorithm to use for the data channel. Default is `crypto aes-128-cbc`.

This option can also be used to disable encryption (`no crypto`), in which case data will be transmitted in clear text.

This option must have the same value for client and server.

The corresponding OpenVPN option is named *cipher*

(For control channel encryption, see the *TLS-Auth* option.)

Options Only for MPTCP protocol

`[no] Mptcp-monitor IFNAME1 IFNAME2 IFNAME3`

Description

This option is used to provide the ifname(s) to create and monitor multi path tcp connection between the client and server.

This option requires a unique path i.e policy-route for each ifname to reach the server.

Options Only for Certificate Authentication

This section presents settings only available when using Certificate Authentication. *First options applicable both for Client and Server modes are shown.* The subsections below lists further settings specific to:

- Client Mode for Certificate Authentication
- Server Mode for Certificate Authentication
- Server Settings per Connecting Client

`[no] certificate LABEL`

Description

In Certificate mode, this option specifies the certificate used by this unit to authenticate itself.

The certificate is specified using the labels assigned when they were imported to the WeOS certificate storage.

Related OpenVPN options are: *cert*, *key*.

`[no] ca-certificate LABEL`

Description

In Certificate mode, this options specifies the CA-certificate used to verify the certificate presented by the remote side.

The certificates are specified using the labels assigned when they were imported to the WeOS certificate storage.

The corresponding OpenVPN options is named *ca*.

`[no] revocation <disabled|relaxed|strict>`

Description

This option is used to specify the revocation checking policy. The default is `disabled`. The other options are `relaxed` and `strict`.

Options

Option	Description
`disabled`	Revocation checking is disabled. This is the default.
`relaxed`	Revocation checking is enabled, but the certificate is rejected only if it is explicitly revoked in a correctly signed and present CRL list. If the CRL has expired (i.e., the NextUpdate time has passed), it is still considered a valid revocation source.
`strict`	Revocation checking is enabled. If the CRL is not present, is outdated, or fails the signature check, the connection will be rejected.
`no`	Disables revocation checking.

If availability of the OpenVPN tunnel is paramount, the `relaxed` option is recommended. This will allow the tunnel to remain up even if the CRL could not be downloaded in time.

See the Revocation Configuration Guide for information on how to configure CRLs.

`[no] tls-auth label <LABEL> [direction <0|1>]`

Description

This option is used to add authentication for OpenVPN TLS control channel packets when operating in Certificate mode. It does not work in PSK mode.

The option works to harden the tunnel security and provides resilience to DoS (Denial of Service) attacks where a fake client tries to make the OpenVPN server waste system resources on processing of fake connection attempts.

This option requires a static key which must have been previously imported into the WeOS device using the certificate import function. The same key must be present on both client and server.

An optional direction parameter can be specified. If so, different keys are used for server-to-client vs client-to-server traffic. Both keys will be extracted from the same key file.

continues on next page

continued

`[no] tls-auth label <LABEL> [direction <0|1>]`

Both client and server must use the same value for this option, with the exception for the optional direction which should have opposite values for server vs clients (e.g. 0 for server and 1 for clients).

The corresponding OpenVPN option is named *tls-auth*.

`[no] renegotiation-timeout SEC`

Description

In certificate mode, this option determines the period at which the data channel encryption key is renegotiated. The default value is 3600 seconds. If client and server use different values for this option, the lower value will be used.

The corresponding OpenVPN option is named *reneg-sec*.

`[no] data-ciphers <CIPHER[,CIPHER ...]>`

Description

List of ciphers to include in cipher negotiation between clients and servers. Defaults to the same default as the included OpenVPN version.

See “help data-ciphers” for a list of available ciphers.

`[no] data-ciphers-fallback <CIPHER>`

Description

Fallback cipher to use when connecting to peers without support for negotiation.

See “help data-ciphers” for a list of available ciphers.

`[no] tls-version-min <1.0|1.1|1.2|1.3>`

Description

Minimum TLS version accepted from peers.

Client Specific Settings for Certificate Authentication

This section presents settings only available in *client* mode for Certificate Authentication.

`[no] pull`

Description

This option only applies to clients running in certificate mode. Setting **no pull** will inhibit all pushed configurations (such as network routes) from the server.

Pull is enabled by default.

The corresponding OpenVPN option is named *pull*.

Server Specific Settings for Certificate Authentication

This section presents settings only available in *server* mode for Certificate Authentication. See also the following section presenting Server Settings per Connecting Client.

`[no] max-clients NUM`

Description

This option limits the number of clients allowed to be simultaneously connected to a Server. When this number is reached, the server will reject an incoming request. Defaults to *25*. `no max-client` will remove the limit. **NOTE**: After automatic renewal of client certificate the server will see the client as if it is a new client.

This option is only available on the server.

The corresponding OpenVPN option is named *max-clients*

`[no] pool start IPADDR <num NUM | end IPADDR> [netmask NETMASK]`

Description

In certificate mode, an OpenVPN server may act like a DHCP server and assign IP addresses to clients. This option configures the IP range and size of the pool.

This is a server-only option.

The corresponding OpenVPN option is named *ifconfig-pool*.

`[no] push-network NETWORK/LEN`

Description

This option allows definition of a set of networks which should be reachable by a client. The server will push these networks to the client, which will create static routes to direct traffic across the tunnel to reach the networks.

This is a server-only option. It only works in Certificate mode.

The corresponding OpenVPN option is named *push route*.

`[no] client-to-client`

Description

In certificate mode, it is possible for the OpenVPN server to act as a router and forward traffic from one client to another instead of passing all incoming traffic to the tunnel endpoint. This option enables the client-to-client routing behaviour.

This option is server-only and is not available in PSK mode.

The corresponding OpenVPN option is named *client-to-client*.

`[no] duplicate-cn`

Description

In certificate mode, each client is associated with a *common name* encoded in its certificate. This option allows multiple clients to use the same common name.

If this option is not enabled, the default behaviour is for the server to disconnect any previous client using the same common name as the one connecting.

This option is available on the server only.

The corresponding OpenVPN option is named *duplicate-cn*.

Server Settings per Connecting Client

In server mode when using certificate authentication, it is possible to assign/push a specific IP address to the connecting client. The *Common Name* in the client's certificate is used to identify the client.

```
example:/config/#> tunnel
example:/config/tunnel/#> ssl 1
example:/config/tunnel/ssl-1/#> server
example:/config/tunnel/ssl-1/#> method cert
example:/config/tunnel/ssl-1/#> client-configuration 1
example:/config/tunnel/ssl-1/client-config-1/#>
```

`[no] client-configuration ID`

Description

In certificate mode, each client is associated with a *common name* encoded in its certificate. This sub-context allows the server to assign a fixed IP addresses to each client based on the client's common name. ID is a number to identify the sub-context. Up to MAX_CLIENT_CONFIGURATIONS configurations can be configured.

This option is valid on the server only.

Related OpenVPN option is *client-config-dir*, see also `common-name` setting below.

`[no] common-name COMMON-NAME`

Description

The X.509 certificate *common name* (CN) used to match this CN binding.

The corresponding OpenVPN option is named *client-config-dir*.

`[no] address ADDRESS/LEN`

Description

The IPv4 address to push to the client with this *common name*.

The corresponding OpenVPN option is named *ifconfig-push*.

`[no] iroute ADDRESS/LEN`

Description

Configure an internal route to a specific client.

This command can be used when a specific client may have another network behind it. What the iroute command will do is to inform the server which client is responsible for handling the specifically provided network.

It is worth noting that this is a route internal to openvpn, it has nothing to do with the device's general routing table.

Because of the above reason it is still necessary to add a route for the specific network to the device's routing table. The reason we need this is because we need a route for the packets to reach openvpn. When the frames reach openvpn iroute will be used to direct the frames to the correct client.

As an example, we add an iroute for the network 10.0.0.0/24, then we also require a route for the network to reach the ssl interface. This can be achieved by adding a static route, under the IP context, in the following manner (Assuming the tunnel is named ssl1):

route 10.0.0.0/24 ssl1

If we do not add the route like in the above example, the frame will not reach the openvpn, and the iroute will serve no purpose.

This option is valid for tun type tunnels.

MAX_IROUTES iroute entries can be configured per client-configuration.

The corresponding OpenVPN option is named *client-config-dir*.

Summary of Options

In the table below, *C* and *S* refer to *Client* and *Server* modes.

WeOS Option	OpenVPN option(s)	PSK	Cert
enable	N/A	C,S	C,S
description	N/A	C,S	C,S
[no] server	N/A	C,S	C,S
method	secret, tls-server, tls-client	C,S	C,S
type	dev-type	C,S	C,S
protocol	proto	C,S	C,S
port	lport, remote	C,S	C,S
crypto	cipher	C,S	C,S
auth	auth	C,S	C,S
keepalive	keepalive	C,S	C,S
compression	comp-lzo, comp-noadapt	C,S	C,S
local-address	local	S	S
bind-dev	bind-dev	S	S
aaa-method	N/A	S	S

WeOS Option	OpenVPN option(s)	PSK	Cert
peer	remote	C	C
identity	auth-user-pass	C	C
Mptcp-monitor	N/A	C	C
secret	secret	C,S	
certificate	cert, key		C,S
ca-certificate	ca		C,S
revocation	crl-verify		C,S
tls-auth	tls-auth		C,S
renegotiation-timeout	reneg-sec		C,S
pull	pull		C
max-clients	max-client		S
pool	ifconfig-pool		S
push-network	push route		S
client-to-client	client-to-client		S
duplicate-cn	duplicate-cn		S
client-configuration	client-config-dir		S
common-name	client-config-dir		S
address	ifconfig-push		S
iroute	client-config-dir		S

Feature Parameters

- MAX_OPENVPN_INSTANCES: 8
- MAX_CLIENT_CONFIGURATIONS: 100
- MAX_IROUTES: 8

2.11.3. IPsec Tunnelling Guide

Introduction

WeOS provides virtual private network (VPN) support via IPsec VPNs. A switch can act as a VPN gateway in site-to-site, site-to-host, or host-to-host scenarios. Configured as a VPN gateway, it can be used to securely connect branch office networks with a central office network. Data traffic is protected by encrypted tunnels when sent over the Internet.

Overview

```
                          .--.-.
                         ( (    )__
                        (_,  \ ) ,_)  Internet/Public Network
                          '-'--`--'
                           |    |
        .------------------'    '--------------------.
        |                                            |
        |Public IP:                                  |Public IP:
        |192.168.0.1                                 |192.168.0.2
    .---+----.                                   .---+----.
    |        |           IPsec Tunnel            |        |
    | Alice  |===================================|  Bob   |
    |        | Responder               Initiator |        |
    '---+----'                                   '---+----'
        |                                            |
     .--.-.                                       .--.-.
    ( (    )__                                   ( (    )__
   (_,  \ ) ,_)                                 (_,  \ ) ,_)
     '-'--`--'                                    '-'--`--'
   Central Office network                     Branch office network
```

Figure 1: VPN tunnel connecting two branch office networks.

IPsec concepts

- **Peers**: The two VPN gateways (Alice and Bob) are referred to as IPsec peers. The peers constitute the end points, or gateways, of the secure tunnel. One of the peers will take the role of tunnel initiator and the other takes the responder role.

- **Initiator and responder**: The VPN initiator is the peer that is responsible for establishing the tunnel by contacting the other peer - the responder. In Figure 1 we assume that Alice is the responder and Bob is the initiator. A switch configured as a VPN gateway can act both as a responder (default) and as an initiator.

- **Local and remote subnet**: Each peer defines policies that describe which traffic is allowed to pass through the established tunnel. Each peer defines a local and a remote subnet, and all traffic between these subnets is sent securely through the tunnel. The policies work both ways, i.e. only traffic matching an inbound policy is allowed after decryption.

- **IDs**: IKE identities of both peers are used during the authentication round.

- **Dead Peer Detection (DPD)**: The connectivity through an established IPsec tunnel may be broken unexpectedly, e.g., one of the peers go down or is disconnected, or if some kind of routing, NAT or firewall problem occurs on the path between them. Dead Peer Detection (DPD) can be used to discover and manage such situations. In DPD, the peers exchange keep-alive messages to monitor if the remote peer is still reachable. If a peer determines connectivity to be broken, appropriate actions should be taken. Active DPD checking is only enforced if no IKE or ESP packet has been received for the configured DPD delay.

- **Pre-shared-key authentication**: A pre-shared key (PSK) is a shared secret that both peers know before establishing a secure connection. It is an easy-to-deploy option but requires strong secrets,

i.e. chosen randomly with sufficient entropy (>256 bits). Care should be taken to protect the secret from leaking so that nobody can impersonate a gateway.

- **NAT-Traversal (NAT-T)**: Network Address Translation (NAT), also called IP masquerading, can interfere with protocols like IPsec, which relies on end-to-end connectivity and unaltered packet headers. NAT-Traversal (NAT-T) is a build-in solution that enables IPsec traffic to pass through NAT devices by encapsulating IPsec packets within UDP packets (a.k.a. ESP-in-UDP). This allows the packets to be routed correctly even when NAT is present, ensuring secure VPN connections can be established across NAT boundaries. **NOTE**: NAT-T is applied when peers are separated by a NAT gateway. If a single device acts as both the NAT and IPsec gateway at the same time, then IPsec traffic will be exempted from NAT.

IPsec Protocol

IPsec is divided into two phases. You can think of the handshake taking place in phase 1 and secure data transmission happening in phase 2.

More specifically:

- **Phase 1** establishes an Internet Key Exchange Security Association (IKE_SA) where negotiation is performed in a secure manner for parameters such as encryption and authentication using the IKEv2 protocol.
- **Phase 2** uses the established IKE_SA(s) to securely negotiate IPsec SA(s), often called CHILD_SA(s) and enable secure data transmission using the ESP protocol.

IKE and ESP settings are by default set to Auto. If you have specific requirements you can instead manually specify the exact algorithms to use for each stage in the communication. If a classical cipher (like *aes128*) is selected, then an authentication algorithm and Diffie-Hellman group also need to be specified, while a modern AEAD algorithm automatically handles authentication during the cipher stage.

> **NOTE**
>
> It is important to select strong cipher suites to ensure the security of the tunnel is not compromised.

Options for IKE (phase 1)

- **Classical** Cipher: Specify authentication algorithm, e.g., *aes128-sha256-x25519*.
- **AEAD** Cipher: Requires a PRF parameter instead of authentication, e.g., *aes128gcm16-prfsha256-x25519*.

Options for ESP (phase 2)

- **Classical** Cipher: Same as in phase 1.
- **AEAD** Cipher: Becomes something like *aes128gcm16-x25519*. No PRF parameter.

> **NOTE**
>
> *aes128gcm16* is the same algorithm as *aes128gcm128* (128 bits is 16 bytes). WeOS currently only supports to specify the GCM/CCM length in bytes while you may find both syntaxes in examples online.

Retransmission

Retransmission refers to the process of resending certain packets during the establishment of an IPsec connection or after DPD timeout. The process consists of 5 attempts with progressively longer relative timeout between attempts. If the peer does not respond within the retransmission attempts, totaling at 165s, the connection fails and a configured *dpd-action* is taken. This process is used during *Connecting* and *Retransmission* states.

Start policy

When configuring an IPsec tunnel, the user can select the role of each peer: responder or initiator. Responders act like servers, waiting for incoming connections, while initiators act like clients and start the connection.

To improve connectivity and availability, a *trap* start policy is configured for both roles. The trap policy triggers tunnel establishment as soon as matching traffic (i.e., traffic destined for the remote subnet) is detected.

In most cases this is not noticeable, since the initiator typically starts the tunnel immediately after a successful configuration. However, if tunnel establishment initially fails (for example due to authentication errors, missing certificates, or the peer not being ready), both peers (responder and initiator) will attempt to establish the tunnel again when traffic is detected – provided that the initial problem has been resolved.

The start policy is not configurable parameter and currently cannot changed in config.

Supported features

- **IKEv2**: The Internet Key Exchange Version 2 (IKEv2) auxiliary protocol responsible for mutual authentication.
- **IKEv1**: Version 1 is considered deprecated, as per RFC 9395. However, it is supported for backwards compatibility with WeOS-4 devices. Only Main mode is available.
- **PSK** (pre-shared keys): Initiator and responder use a shared secret (password) for authentication.
- **Certificate-based authentication**: X.509 certificates can be used for tunnel authentication. A local certificate (with associated private key) is presented to the peer, and a CA certificate is used to validate the peer's certificate chain.
- **Certificate revocation using CRL**: Certificate revocation is checked against CRLs, and the behaviour when CRL information is missing or unavailable is controlled by a configurable revocation policy (see the `revocation` command below).
- **Static IPs**: The local address needs to be specified as an IP address that exists on an interface. Either use a static IP address for the outgoing interface (preferred) or wait until an IP address has been assigned to the outgoing interface before entering the IP address in the IPsec configuration.
- **CLI** and **WebGUI** configuration.

Supported scenarios

For examples of use-cases, refer to the following document:

- Site-2-Site
- Host-2-Host
- IKEv1
- Certificate authentication

Current limitations

Configuration

Configure IPsec in the tunnel overview context:

```
example:/#> config
example:/config/#> tunnel
```

Tunnels overview context

Here you can create or delete an IPsec tunnel instance, or show summary of your tunnels.

`[no] ipsec ID`

Description

Create new instance, or enter the configuration context of an existing instance.

Options

Option	Description
`no`	remove an instance entirely.

```
example:/config/tunnel/#> ipsec 1
example:/config/tunnel/ipsec-1/#>
```

> **NOTE**
>
> Up to 25 instances is supported. However, the exact number of connections can be further limited for performance reasons, as it depends on the platform, the traffic load of the established tunnels as well as the configuration of your unit.

Instance Configuration Syntax

To configure and show details of a single IPsec tunnel instance

```
example:/config/tunnel/#> ipsec 1
example:/config/tunnel/ipsec-1/#> help
```

`[no] enable`

Description

Enable/disable tunnel. Useful to disable a fully configured tunnel before deployment, the configuration will remain dormant while disabled.

`[no] version 1 | 2`

Description

Select IKE protocol version.

continues on next page

continued

`[no] version 1 | 2`

> **NOTE**
>
> IKEv1 is considered deprecated according to RFC 9395.

Options

Option	Description
`no`	Set default IKEv2 version of the protocol.

`[no] dpd-action <clear|trap|restart>`

Description

Dead Peer Detection (DPD) action to perform for this tunnel when communication is lost. Since trap start policy is set for both roles setting trap here is redundant. Only *clear* is available for a responder while an initiator can take any action.

Options

Option	Description
`clear`	Close the tunnel and does not take further action (default for Responder).
`trap`	Install a trap policy, which will catch matching traffic and tries to re-negotiate the tunnel on-demand.
`restart`	Immediately try to re-negotiate the tunnel under a fresh IKE handshake (default for initiator).
`no`	Disable dead peer detection. Without DPD, the connection may remain active even if the peer becomes unresponsive.

`[no] dpd-delay <SEC>`

Description

Period time for DPD packets. Active DPD checking is only enforced if no IKE or ESP packet has been received for the configured DPD delay.

`[no] dpd-timeout <SEC>`

Description

(IKEv1 only) Timeout before declaring the peer dead after no response to DPD probes and retransmissions. This option is ignored for IKEv2 and is only available when the tunnel is configured to use IKEv1.

`[no] initiator`

Description

Role, initiator (client) or responder (server). The initiator is the one that sends the initial request for the IKE handshake.

Options

Option	Description
`no`	Reset role to responder.

`[no] ike cipher <option> auth|prf <option> dh <option>`

Description

Phase-1 handshake negotiation. `no ike` activates default settings.

> NOTE

AEAD ciphers needs a PRF algorithm while the other ones needs an auth algorithm instead. The AEAD ciphers are those with CCM or GCM in the name.

Options

Option	Description
`cipher`	Encryption algorithm
`auth`	Authentication mechanism
`prf`	Pseudo-random function
`dh`	Diffie-Hellman group
`no`	This value selects the default proposals, a set of algorithms which are safe and is usually a good choice for interoperability.

`[no] esp cipher <option> [auth <option>] dh <option>`

Description

Phase-2 handshake negotiation proposal.

> NOTE

AEAD ciphers do not use an auth algorithm. The AEAD ciphers are those with CCM or GCM in the name.

continues on next page

continued

`[no] esp cipher <option> [auth <option>] dh <option>`

Options

Option	Description
`cipher`	Encryption algorithm
`prf`	Pseudo-random function
`dh`	Diffie-Hellman group
`no`	This value selects the default proposals, a set of algorithms which are safe and is usually a good choice for interoperability.

`[no] method <psk|cert>`

Description

Select authentication method for the tunnel.

Options

Option	Description
`psk`	Use a pre-shared key for authentication (default).
`cert`	Use X.509 certificate-based authentication.

`[no] secret <SECRET>`

Description

Mandatory pre-shared key when PSK method is used.

Options

Option	Description
`SECRET`	MIN:8 MAX:63 Valid characters are ASCII characters 33-126, except '#' which is the CLI comment character. Secrets starting with the "0s" prefix are considered base64 encoded, whereas "0x" designates a hex-encoded value.

`[no] certificate <LABEL>`

Description

Label of a local certificate to use when `method cert` is selected. The certificate must have an associated private key present in the system.

Options

Option	Description
`LABEL`	Use the specified certificate label.

`[no] ca <LABEL>`

Description

Label of a CA certificate from the Trusted Local CA Store used to validate the certificate chain coming from the peer. Currently it can be a single certificate, i.e. not a bundle.

Options

Option	Description
`LABEL`	Use the specified CA certificate label.

`[no] revocation <relaxed|strict>`

Description

Set the certificate revocation checking policy using CRL for this IPsec tunnel. The CRL revocation checking does not require any additional configuration and relies on CRL Distribution Points defined in the incoming certificate chain. It cannot be disabled and performs according to selected policy during the authentication phase of the tunnel negotiation and at every next re-authentication attempt (see the `reauth-time` parameter). The timeout is 10s.

Options

Option	Description
`relaxed (default)`	The tunnel fails authentication only if a certificate is explicitly revoked, i.e. it is known to be bad. If no revocation information is available (for example, CRL cannot be fetched), the certificate is treated as valid.
`strict`	The tunnel fails authentication if no valid revocation information is available. In other words, if the certificate is not known to be unrevoked (for example, missing or expired CRL), authentication fails.

`[no] peer <ADDRESS>`

Description

When acting as an initiator, the peer refers to the remote server IP to connect to. As a responder, it can be used to allow a specific IP or ANY address, which is the default setting when set to `no peer`.

`[no] local-address <ADDRESS>`

Description

This is the local address of the outbound interface. It must be set for the tunnel to work.

```
[no] local-id <inet <ADDRESS> | name <DOMAIN|USER> | email <USER@DOMAIN> | key <ID> |
dn <STRING>>
```

Description

Set the identifier (type and value) for the VPN gateway. The local-id is used by the VPN gateway during the IKE handshake. Typically the ”name” type with a simple ID text string (e.g., alice) can be used to identify the VPN gateway.

When using certificate authentication, the IKE identity must be contained in the certificate, either as the subject DN or as a subjectAltName. The identity defaults to the certificate’s subject DN if set to Auto.

> **NOTE**
>
> When using PSK authentication, each tunnel must have a unique (local-id, remote-id) pair; reusing a pair can cause the wrong pre-shared key to be selected.

Options

Option	Description
`no`	Sets local-id to Auto, which defaults to different values depending on the selected authentication method. When using the pre-shared-key method, the local-address is selected. When using the certificate method, the identity defaults to the certificate’s subject DN if not specified.

Supported ID types:

Type	Description	Example
inet	IPv4 address	192.168.1.10
name	Fully qualified domain name	server.example.com
email	E-mail address	`user@example.com`
key	Opaque key identifier	my_vpn_key
dn	ASN.1 DER-encoded hex string	0x3049310b…

> **NOTE**
>
> The `dn` type expects an ASN.1 DER-encoded Distinguished Name as a hex string (e.g. `0x3049310b...`), not a human-readable string like `CN=Alice, O=Example`. Use a tool such as `openssl` to obtain the correct encoding from a certificate. To use a human-readable representation of a Distinguished Name, write it directly without a type prefix (e.g. `CN=Alice, O=Example`) and the type will be autodetected as a Distinguished Name.

`autodetect <VALUE>` is the default when a value is given without an explicit type keyword. WeOS deduces the type from the string content. `show` displays the deduced type in parentheses, e.g. `autodetect (name) gateway.example.com`. Use an explicit type when the value could be ambiguous.

continues on next page

continued

```
[no] local-id <inet <ADDRESS> | name <DOMAIN|USER> | email <USER@DOMAIN> | key <ID> |
dn <STRING>>
```

Setting `no local-id` / `no remote-id` selects **`auto`**, which picks a sensible default: the local address for PSK or the certificate Subject DN for certificate authentication. On the remote side, a responder with `auto` accepts any peer identity while an initiator defaults to the peer's IP address.

```
[no] remote-id <inet <ADDRESS> | name <DOMAIN|USER> | email <USER@DOMAIN> | key <ID>
| dn <STRING>>
```

Description

Remote peer ID. IKE identity to expect for authentication round.

Types are the same for local-id.

When using certificate-based authentication, the IKE identity must be contained in the peer's certificate, either as the subject DN or as a subjectAltName. It is possible to use wildcards to match remote identities. Connections with exact matches are preferred.

Options

Option	Description
`no`	Results in the "Auto" setting, which has different meanings depending on the role. When acting as a responder it means accepting any remote peer; as an initiator it is set to the peer's IP address.

```
[no] local-subnet <ADDRESS/LEN | ADDRESS[ NETMASK]>
```

Description

Subnet to expose to the other side of the VPN tunnel. Only traffic from this IP range is allowed to enter the tunnel through this gateway, and traffic arriving through the tunnel is only accepted when destined to an address in this range.

Options

Option	Description
`no`	Only traffic to/from the IP address of the local-address interface will be allowed through the tunnel, e.g., "192.168.0.1/32".

```
[no] remote-subnet <ADDRESS/LEN | ADDRESS[ NETMASK]>
```

Description

Remote subnet allowed to access your local site. `no remote-subnet` gets replaced by peer's address.

`[no] description <STRING>`

Description

Short free form description of this IPSec tunnel.

`[no] sa-rekey-time <SECONDS[s] | MINUTESm | HOURSh | DAYSd>`

Description

Time after which the IPsec CHILD_SA (phase 2) is rekeyed. This controls how often the data-plane keys are refreshed for this tunnel. CHILD_SA rekeying refreshes key material, optionally using a Diffie-Hellman exchange if a group is specified in the proposal. To avoid rekey collisions initiated by both ends simultaneously, a random value is subtracted to form the effective sa-rekey-time lifetime.

`[no] reauth-time <SECONDS[s] | MINUTESm | HOURSh | DAYSd>`

Description

Time after which IKE re-authentication (phase 1) is performed. Re-authentication revalidates credentials and may cause a short interruption when enabled. It is disabled by default. Use this feature to align revocation checking to the refresh interval of revocation sources (for example, CRL refresh time). This is necessary to limit the possibility of undetected use of revoked certificates for tunnel authentication.

An example alignment is `0.5 x (CRL-refresh-time)`. You can configure the re-authentication time to align it with CRL refresh time or, alternatively, OCSP response caching time.

Cipher Suites

A cipher suite is a combination of algorithms used in different stages of the secure communication. The *cipher* is used for encryption, *authentication* is used to verify message integrity (that it is received unchanged), *prf* is used to derive subkeys and the *Diffie–Hellman* group is used during key exchange.

Classical ciphers

Classical ciphers are specified together with an authentication algorithm and a Diffie-Hellman group. The cipher algorithms supported are:

Algorithm	Description
aes128	AES, 128 bit key in CBC mode.
aes128ctr	AES, 128 bit key in CTR mode.
aes256	AES, 256 bit key in CBC mode.
aes256ctr	AES, 256 bit key in CTR mode.

AEAD ciphers

Authenticated Encryption (with Associated Data) algorithms automatically provide authentication while encrypting, so no separate authentication algorithm should be specified together with these ciphers. CCM ciphers use a Counter mechanism with CBC-MAC and GCM ciphers use a mechanism called Galois/Counter mode.

Algorithm	Description
aees128ccm12	128 bit AES-CCM with 12 byte ICV.
aes128ccm16	128 bit AES-CCM with 16 byte ICV.
aees128gcm12	128 bit AES-GCM with 12 byte ICV.
aes128gcm16	128 bit AES-GCM with 16 byte ICV.

Authentication

These should be specified if a classical cipher is selected. All supported algorithms use the HMAC construct with an inner hash primitive. The inner primitive can be one of:

Algorithm	Description
sha1	Deprecated, but needed for WeOS-4 compatibility.
sha256	SHA2-256
sha384	SHA2-384
sha512	SHA2-512

Pseudo Random Function

The PRF algorithm is used for key derivation. It should only be specified for IKE (phase 1) and only if an AEAD cipher has been selected for IKE. In that case, choose one of:

- **prfsha256**
- **prfsha384**
- **prfsha512**

Diffie-Hellman group

The Diffie-Hellman group is used for key exchange. Choose one of:

Group Name	Bit Length	Description
modp2048	2048	modp, group 14, RFC 3526
modp3072	3072	modp, group 15
modp4096	4096	modp, group 16
ecp256	256	elliptic curve, group 19, RFC 5903
ecp384	384	elliptic curve, group 20
ecp521	521	elliptic curve, group 21
x25519	256	elliptic curve 25519, group 31, RFC 8031

Firewall consideration

When configuring a firewall, it is important to ensure that rules do not unintentionally block or disrupt IPsec traffic. IPsec relies on specific protocols and ports (such as UDP 500, UDP 4500, and ESP) to establish and maintain secure tunnels. If the firewall is too restrictive, it can prevent tunnel negotiation, key exchange, or data transfer, leading to failed or unstable VPN connections. Therefore, firewall rules should be carefully reviewed and tested to allow necessary IPsec-related traffic while still maintaining overall network security.

© 2026 Westermo Network Technologies AB

> **NOTE**
>
> IPsec traffic is exempted from NAT if a gateway acts as an IPsec and NAT gateway at the same time.

Status and debugging

The tunnel exec context allow for showing status of an already configured IPsec tunnels or execute control commands to the IPsec daemon. When debugging a tunnel, examining the log is quite useful.

Show status

```
example:/#> tunnel
example:/tunnel/#> show ipsec
ID  DESCRIPTION      STATUS              TIME (since last rekey)
1   ipsec1           Up(Installed)       0 Days 2 Hours 11 Mins 28 Secs
```

States

A tunnel can have one of the following execution statuses:

- **Down(Unknown)**: A tunnel is configured but the daemon does not do anything about it. This usually indicates a misconfiguration. During this state the command **show ipsec** will not provide Security-Association (SA) details.

- **Down(Connecting)**: The initiator is trying to establish a new tunnel. Remaining in this state can indicate problems reaching the peer.

- **Down(IKE Established)**: The phase 1 handshake was successful but no CHILD_SA was created. When persistent, this might indicate configuration problems, for example when subnets on both sides do not match. Examine *syslog* to understand the issue.

- **Up(Installed)**: Everything works; the tunnel is operational and sending encrypted data.

> **NOTE**
>
> This is the only “Up” state.

- **Down(Retransmission)**: A Dead Peer Detection (DPD) request went unanswered. See the retransmission section for details.

To show details of an ongoing connection and the currently loaded configuration of a specific IPsec tunnel instance:

```
example:/tunnel/#> show ipsec 1
Current state for tunnel ipsec1 is UP(Installed).

Security-Association(SA) details:
ipsec1: #6, ESTABLISHED, IKEv2, 20759114c78bf131_i* 4fb9e973f3067b48_r
  local  'alice' @ 192.168.0.1[500]
  remote 'bob' @ 192.168.0.2[500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/ECP_256
  established 5s ago, rekeying in 14304s
  tunnel1: #5, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA2_256_128
```

```
    installed 5s ago, rekeying in 3267s, expires in 3955s
    in  c2032981,      0 bytes,     0 packets
    out cc173955,      0 bytes,     0 packets
    local  192.168.0.1/32
    remote 192.168.0.2/32

Loaded configuration for the connection:
ipsec1: IKEv2, no reauthentication, rekeying every 14400s, dpd delay 30s
  local:  192.168.0.1
  remote: 192.168.0.2
  local pre-shared key authentication:
    id: alice
  remote pre-shared key authentication:
    id: bob
  tunnel1: TUNNEL, rekeying every 3600s, dpd action is start
    local:  dynamic
    remote: dynamic
```

Control commands

`[no] ipsec <restart|monitor>`

Description

Restart or monitor the IPsec daemon.

Options

Option	Description
`restart`	force a restart of the IPsec daemon, this will restart all tunnels.
`monitor`	debug the IPsec packets passing through the kernel, this will show both ESP and IKE packets.

2.11.4. WireGuard Tunnelling Guide

Introduction

WireGuard is a modern, fast, and secure VPN protocol designed to be simpler and more performant than traditional VPN solutions like IPsec and OpenVPN. WeOS includes WireGuard support to provide efficient and secure VPN tunnel services.

WireGuard offers several advantages:

- **Simple configuration**: Minimal settings with sensible defaults
- **High performance**: Low overhead and fast connection speeds
- **Modern cryptography**: Uses state-of-the-art cryptographic algorithms
- **NAT traversal**: Built-in support for operating behind NAT devices

For more information about WireGuard, visit https://www.wireguard.com[1].

For example use-cases and example setups, refer to:

- WireGuard Site-To-Site

Overview

WireGuard operates using a simple peer-to-peer model. Each device (peer) has a pair of cryptographic keys (private and public), and traffic is routed based on public keys rather than IP addresses. This cryptokey routing approach provides both routing and access control.

```
                        .--.-.
                       ( (    )__
                      (_,  \ ) ,_)  Internet/Public Network
                        '-'--`--'
                         |    |
     .------------------'    '--------------------.
     |                                             |
     |Public IP:                                   |Public IP:
     |203.0.113.1:51820                           |198.51.100.1:51820
 .---+----.                                     .---+----.
 |        |        WireGuard Tunnel             |        |
 | Peer A |=====================================| Peer B |
 |  wg1   |                                     |  wg1   |
 '---+----'                                     '---+----'
     |                                             |
     |10.0.1.0/24                                  |10.0.2.0/24
  .--.-.                                         .--.-.
 ( (    )__                                     ( (    )__
(_,  \ ) ,_)                                    (_,  \ ) ,_)
  '-'--`--'                                      '-'--`--'
Site A Network                                 Site B Network
```

Figure 1: WireGuard tunnel connecting two site networks.

WireGuard Concepts

- **Interface**: A WireGuard interface (e.g., wg1, wg2) represents a VPN tunnel endpoint. Each interface has its own private/public key pair and listen port.
- **Peers**: Each WireGuard interface can have multiple peers. A peer represents a remote endpoint that can communicate through the tunnel.
- **Public Key**: The primary identifier for peers. Each peer is identified by its public key, which is used for authentication and encryption.
- **Allowed IPs**: A list of IP addresses and subnets that define which traffic is routed to/from a specific peer. This provides both routing and access control (cryptokey routing).

[1] https://www.wireguard.com

- **Endpoint**: The remote IP address and port of a peer.
- **Persistent Keepalive**: An optional interval for sending keepalive packets to maintain NAT traversal and connection state.
- **Preshared Key**: An optional additional symmetric key for post-quantum resistance and defence-in-depth security.

WireGuard Cryptography

WireGuard uses a fixed set of modern cryptographic algorithms:

- **Curve25519** for Elliptic Curve Diffie-Hellman (ECDH) key exchange
- **ChaCha20** for symmetric encryption
- **Poly1305** for message authentication
- **BLAKE2s** for cryptographic hashing
- **HKDF** for key derivation

The protocol uses the Noise protocol framework[1] for handshakes, providing forward secrecy, identity hiding, and resistance to key compromise.

Configuration

This section provides information about configuring WireGuard tunnels in WeOS.

CLI Configuration

WireGuard configuration context is entered via the tunnel configuration context:

```
example:/config/#> tunnel
example:/config/tunnel/#> wireguard 1
example:/config/tunnel/wireguard-1/#>
```

Common Interface Options

`[no] enable`

Description

Activate or deactivate this WireGuard interface. If deactivated, the configuration is preserved but the tunnel is not active.

`[no] description STRING`

Description

A short text description of the tunnel. The description is displayed when listing tunnels using CLI or Web. It is limited to 64 characters.

`[no] private-key KEY`

Description

The private key for this interface. Keys are base64-encoded strings generated using Curve25519. A private key is generated by default. Keys can also be generated externally using the `wg` command-line tool and copied into WeOS.

continues on next page

[1] https://noiseprotocol.org/

continued

`[no] private-key KEY`

Important: Keep private keys secure and never share them.

`public-key`

Description

Display the public key corresponding to the configured private key. This read-only value should be shared with peers that need to connect to this interface.

`[no] listen-port PORT`

Description

The UDP port on which this interface listens for incoming connections. Default is 51820. If running multiple WireGuard interfaces, each must use a unique port.

Peer Configuration

Each WireGuard interface can have one or more peers configured:

```
example:/config/tunnel/wireguard-1/#> peer 1
example:/config/tunnel/wireguard-1/peer-1/#>
```

`[no] peer INSTANCE`

Description

Add a peer identified by its public key, or enter the configuration context of an existing peer. `no peer PUBLIC-KEY` removes the peer configuration.

`[no] description STRING`

Description

A short text description of the peer. The description is displayed when listing peers using CLI or Web. It is limited to 64 characters.

`[no] enable`

Description

Activate or deactivate this WireGuard peer. If deactivated, the configuration is preserved but the peer connection is not active.

`[no] public-key KEY`

Description

The public key of the peer. This identifies and authenticates the peer.

`[no] preshared-key KEY`

Description
An optional symmetric key shared with this peer for additional security and post-quantum resistance. Same key must be used on both sides.

`[no] endpoint IP`

Description
The remote endpoint IP address, hostname or FQDN for this peer.

`[no] listen-port PORT`

Description
The UDP port on which the peer listens for incoming connections.

`[no] persistent-keepalive SECONDS`

Description
Send keepalive packets every N seconds to maintain NAT traversal. Typical values are 25 seconds. Set to 0 to disable (default). This is useful for peers behind NAT or with stateful firewalls.

`[no] allowed-ip IP/MASK`

Description
Add an IP address or subnet that is allowed to communicate with this peer. This command can be used multiple times to add multiple allowed IP ranges.

Allowed IPs serve two purposes:
- **Outbound**: Traffic destined to these IPs is routed to this peer
- **Inbound**: Only packets with source IPs in this list are accepted from this peer

Key Management

WireGuard in WeOS uses autogenerated Curve25519 key pairs. Keys can also be generated externally using the `wg` command-line tool and imported into WeOS.

Generate private/public key pair

```
$> wg genkey | tee privatekey | wg pubkey > publickey
```

Generate preshared key

```
$> wg genpsk > presharedkey
```

Important: Keep private keys secure and never share them.

Monitoring and Statistics

CLI Show Commands

View WireGuard tunnel status and statistics:

`show tunnel wireguard`

Description

Display summary information for all configured WireGuard interfaces, including interface name, public key, and listen port. Also shows peer information including endpoints, transfer statistics, and handshake status.

`show tunnel wireguard IFACE`

Description

Display detailed information for a specific WireGuard interface, including:

- Interface configuration
- Public key
- Listen port
- Peer details (public key, endpoint, allowed IPs)
- Latest handshake time
- Data transfer (bytes sent/received)
- Persistent keepalive setting

Security Considerations

Key Management

- **Protect private keys**: Private keys should never be shared or transmitted insecurely
- **Regular key rotation**: Consider rotating keys periodically for enhanced security
- **Preshared keys**: Use preshared keys for additional post-quantum resistance

Network Security

- **Firewall rules**: Configure firewall rules to restrict access to WireGuard ports
- **Allowed IPs**: Use restrictive allowed-ip configurations to limit peer access
- **Network segmentation**: Consider using separate WireGuard interfaces for different security zones
- **Monitoring**: Regularly review tunnel statistics and handshake times

Best Practices

- Use strong, randomly generated keys
- Limit allowed-ip ranges to only necessary networks
- Enable persistent keepalive only when needed (NAT traversal)
- Regularly review and audit peer configurations
- Use descriptive names for tunnels and peers for easier management

Troubleshooting

Common Issues

No handshake occurring:

- Verify that the public keys are correctly configured on both peers

- Check that the endpoint address and port are correct and reachable
- Ensure UDP port (default 51820) is not blocked by firewalls
- Verify that private keys are correctly configured

Handshake succeeds but no traffic flows:
- Check the `allowed-ip` configuration on both peers
- Verify that routing is correctly configured for the tunnel network
- Ensure that IP forwarding is enabled if routing between networks
- Check firewall rules on both peers

Connection drops frequently:
- Enable `persistent-keepalive` (typically 25 seconds) to maintain NAT mappings
- Check for network stability issues
- Verify that no aggressive firewall timeout settings are interfering

Cannot reach specific IPs through tunnel:
- Verify that the destination IPs are in the peer's `allowed-ip` list
- Check routing tables to ensure traffic is routed to the WireGuard interface
- Confirm that the remote peer has appropriate routing configured

Debug Commands

`show tunnel wireguard [INTERFACE]`

Description

Display current tunnel status, peer information, and statistics

`show log`

Description

Check system logs for WireGuard-related errors or warnings

`show interface [INTERFACE]`

Description

Verify that the WireGuard interface is up and has the correct IP configuration

`ping iface [INTERFACE] [DESTINATION]`

Description

Test connectivity through a specific WireGuard interface

References

- WireGuard Website[1]
- WireGuard Whitepaper[2]

[1] https://www.wireguard.com
[2] https://www.wireguard.com/papers/wireguard.pdf

2.12. Firewall and NAT

2.12.1. Firewall Guide

Introduction

When connecting a network to the Internet or any untrusted network, a router with Firewall functionality can be used to protect against undesired access to your local network and devices. This is done by applying rules to different parts/hooks of the path through which packets flow. These hooks are **prerouting**, **input**, **forward**, and **postrouting** and can be seen in *Figure 2*. *Figure 1* shows an example of a basic network using a Firewall to protect its local devices.

```
                                 .--.-.
                                ( (    )__
                               (_,  \ ) ,_)  Internet/Intranet
                                 '-'--`--'
                                     |  WAN (vlan4)                        ^
                                .---+----.                                 |
                                |        |                               Public
                         +-----+ GW/FW  +-----+                          ------
                         |  .99|        |.99  |                          Private
LAN (vlan1)              |     '----+---'     |      LAN (vlan2)           |
192.168.1.0/24           |          |.99      |      192.168.2.0/24        V
            ----+----+           |         +----+----
                |.1              |              |.1
              .-+--.             |            .-+--.
              | H2 |             |            | H3 |
              '----'             |            '----'
                                 |
                                 |
                               ----+---- WEB (vlan3)
                                 |.1   192.168.3.0/24
                               .-+--.
                               | H1 |
                               '----'
```

Figure 1: Showing an example network setup using a Firewall. In the case of this example, the Firewall can be set up to allow all private subnets (vlan1, vlan2, vlan3) to communicate with each other and with the Gateway itself, while blocking external traffic from e.g. the Internet.

> **TIP**
>
> A real configuration of the example in **Figure 1** can be seen on the Firewall HowTo page.

Overview

The WeOS Firewall supports the following types of functionality:

- Input Filtering
- Forward Filtering
- Prerouting Filtering
- NAT (Network Address Translation)
- Port Forwarding
- ALG Helpers

> **NOTE**
>
> For products of software level *Extended*, all of the above mentioned functions are available.
> For products of software level *Standard*, only input filtering is available.

- **Prerouting Filtering:** The *prerouting* filtering support is used to classify and modify traffic before the routing decision (see *Figure 2*). In WeOS, *prerouting* filtering currently supports only *DSCP mangling*, allowing packets to be marked with a DSCP value for *QoS* handling further along the forwarding path.
- **Input Filtering:** The *input* filtering support is primarily used to control accessibility to services on the device itself. Input rules are checked after the routing decision (see *Figure 2*) and apply to packets that are directed to the local device and processes running on the device itself. An *input* filtering rule is configured to either **drop** or **accept** a packet.
- **Forward Filtering:** The *forward* filtering support is primarily used to control what traffic is allowed to be routed to connected devices. Forwarding rules are checked after the routing decision (see *Figure 2*) and apply to packets that are not directed to the local device. A *forward* filtering rule is configured to either **drop** or **accept** a packet.
- **NAT (Network Address Translation):** *NAPT (Network Address Port Translation)* is the most common *NAT* form. It allows a common (public) IP address to be shared by a set of hosts on a private network. *NAT* rules are checked during the *postrouting* as seen in *Figure 2*. This form of *NAT* is sometimes referred to as *IP Masquerading* or *PAT (Port Address Translation)*.

> **NOTE**
>
> *IPsec* traffic is exempted from being subjected to *NAT* unless it is packaged in NAT-traversing UDP packets.

- **Port Forwarding:** *Port forwarding* allows a service (such as a Web server) located on a private network to be made accessible from another public network. This is done by mapping a public IP address and public port number to a private IP address and a corresponding private port number. Traffic designated for the public IP and the specified public port will be redirected to the private IP and its specified port. *Port forwarding* rules are checked during *prerouting* as shown in *Figure 2*.

```
                    prerouting ---Port Forwarding---
                        |
                        v
                .-----------------.
                | Routing decision|
                '---+---------+---'
                    |         |
                    v         v
                  input    forward
                    |         |
                <--'          |
local device                  |
              ---.            |
                    |         |
                    v         |
     .----------------.       |
     |Routing decision|       |
     '----------------'       |
                    |         |
                    '---. .---'
                        |
                        v
                   postrouting ---NAT---
```

Figure 2: Representation of the different hooks where Firewall rules apply.

- **ALG Helpers:** The WeOS firewall utilises connection tracking; a filter rule allowing traffic to pass through the firewall in one direction will implicitly allow traffic of established connections (and traffic of related connections) to also pass in the reverse direction. Connection tracking can be

configured to handle more complex protocols by enabling *ALG helpers* a.k.a. *Application Level Gateway helpers*. Some network protocols are more complex and therefore more difficult than others to be handled by the connection tracking function in a firewall or NAT device. An example is FTP, which utilises one port (often 21) as a control channel to exchange commands and another random port as a data channel for actual file transfers. To enable a PC to download files through a firewall from an FTP server on the Internet, the firewall must inspect the FTP control connection to learn which connections to let through. To make the firewall handle such protocols correctly, protocol-specific *ALG helpers* can be enabled. *ALG helpers* have some impact on the unit's routing performance and are therefore disabled by default.

Order of Firewall Rules

Configured firewall rules have an order. For *input*, *forward*, *NAT*, and *port-forward* rules, packets are evaluated against the rules in order and the first matching rule will decide how to treat the packet. For *prerouting* DSCP rules, all matching rules are evaluated in order and the last matching rule determines the final DSCP value applied to the packet.

Configuration

The *Firewall* configuration is a sub-context to the *IP* configuration context in the *CLI*:

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> firewall
example:/config/ip/firewall#>
```

By default the firewall is disabled. The command **enable** can be used to enable the firewall.

> **WARNING**
>
> It is vitally important to configure the firewall correctly before enabling it. Failing to do so can get you permanently locked out. For example, if you are managing the firewall through *SSH*, *HTTP* or *HTTPS*, you will not be able to reach the device if you enable the firewall without first explicitly allowing the management traffic in question.

To prevent lockout, you have to specify an input rule for your management interface. The following example enables *HTTP* on *vlan1*:

```
example:/config/ip/firewall#> input accept in vlan1 dport http proto tcp
```

In *Figure 1*, this rule would enable devices on the LAN to access the web interface of the Gateway over *HTTP*. More examples can be seen in the Firewall HowTo page.

> **NOTE**
>
> Offloaded traffic is not subject to firewall processing, and the number of tracked connections is limited to *4096*.

Syntax

```
[no] prerouting dscp VALUE [pos POSITION] [in IFACE] [saddr ADDR[/LEN]] [sport PORT|
RANGE] [daddr ADDR[/LEN]] [dport PORT|RANGE] [proto PROTO] [bind-counter COUNTER]
[counter] [comment COMMENT] [log]
```

Description

Manage prerouting rules.

Create or delete a prerouting rule. Prerouting rules apply to all incoming and forwarded packets.

Options

Option	Description
`no`	Delete specified rule.
`dscp VALUE`	A DSCP value (0 - 63) to be assigned to packets matching configured criteria.
`pos POSITION`	The position in the list of rules you want to apply your command to.
`in IFACE`	Inbound Interface: The interface where the packet comes in.
`saddr`	Source IP Address/Subnet: The source IP address of the packet.
`sport`	Source Port: The source port of the packet.
`daddr`	Destination IP Address/Subnet: The destination address of the packet.
`dport`	Destination Port: The destination port of the packet.
`ADDR[/LEN]`	IP address specified as either a single IP address, or a whole IP subnet.
`PORT|RANGE`	Port specified as either a single port, or a range of ports.
`proto PROTO`	Protocol: The protocol type of the IP payload. Can be specified as a well-known protocol name (e.g. `tcp`, `udp`, `icmp`, `gre`, `esp`, `ah`, `ospf`, `vrrp`, `sctp`) or as an IANA protocol number (1–255). Use a protocol name or number when filtering traffic that does not use TCP or UDP ports, such as GRE tunnels (47) or OSPF (89). For a full list of IANA protocol numbers see https://www.iana.org/assignments/protocol-numbers/[1]. Common protocol numbers: Number — Name 1 — icmp 6 — tcp 17 — udp 47 — gre 50 — esp 51 — ah

continues on next page

[1] https://www.iana.org/assignments/protocol-numbers/

continued

```
[no] prerouting dscp VALUE [pos POSITION] [in IFACE] [saddr ADDR[/LEN]] [sport PORT|
RANGE] [daddr ADDR[/LEN]] [dport PORT|RANGE] [proto PROTO] [bind-counter COUNTER]
[counter] [comment COMMENT] [log]
```

Option	Description
	<table><tr><th>Number</th><th>Name</th></tr><tr><td>89</td><td>ospf</td></tr><tr><td>112</td><td>vrrp/carp</td></tr><tr><td>132</td><td>sctp</td></tr></table>
`bind-counter COUNTER`	Assigns the rule to COUNTER.
`counter`	Add a counter that only shows in the raw output. This counter is reset any time the firewall configuration is changed.
`comment COMMENT`	Add a comment to the rule.
`log`	Log traffic matching this rule.

```
[no] input accept|drop [pos POSITION] [in IFACE] [saddr ADDR[/LEN]] [sport PORT|
RANGE] [daddr ADDR[/LEN]] [dport PORT|RANGE] [proto PROTO] [bind-counter COUNTER]
[counter] [comment COMMENT] [log]
```

Description

Manage input rules.

Create or delete an input rule. Input rules are checked after the routing decision and apply to packets that are directed to the local device and processes running on the device itself.

Options

Option	Description
`no`	Delete specified rule.
`accept|drop`	Accept or drop traffic matching the specified rule.
`pos POSITION`	The position in the list of rules you want to apply your command to.
`in IFACE`	Inbound Interface: The interface where the packet comes in.
`saddr`	Source IP Address/Subnet: The source IP address of the packet.
`sport`	Source Port: The source port of the packet.
`daddr`	Destination IP Address/Subnet: The destination address of the packet.
`dport`	Destination Port: The destination port of the packet.
`ADDR[/LEN]`	IP address specified as either a single IP address, or a whole IP subnet.
`PORT|RANGE`	Port specified as either a single port, or a range of ports.
`proto PROTO`	Protocol: The protocol type of the IP payload. Can be specified as a well-known protocol name (e.g. `tcp`, `udp`, `icmp`, `gre`, `esp`, `ah`, `ospf`, `vrrp`, `sctp`) or as an IANA protocol number (1–255). Use a protocol name or number when filtering traffic that does not use

continues on next page

continued

```
[no] input accept|drop [pos POSITION] [in IFACE] [saddr ADDR[/LEN]] [sport PORT|
RANGE] [daddr ADDR[/LEN]] [dport PORT|RANGE] [proto PROTO] [bind-counter COUNTER]
[counter] [comment COMMENT] [log]
```

Option	Description
	TCP or UDP ports, such as GRE tunnels (47) or OSPF (89). For a full list of IANA protocol numbers see https://www.iana.org/assignments/protocol-numbers/[1]. Common protocol numbers: Number: Name 1: icmp 6: tcp 17: udp 47: gre 50: esp 51: ah 89: ospf 112: vrrp/carp 132: sctp
`bind-counter COUNTER`	Assigns the rule to COUNTER.
`counter`	Add a counter that only shows in the raw output. This counter is reset any time the firewall configuration is changed.
`comment COMMENT`	Add a comment to the rule.
`log`	Log traffic matching this rule.

```
[no] forward accept|drop [pos POSITION] [in IFACE] [out IFACE] [saddr ADDR[/LEN]]
[sport PORT|RANGE] [daddr ADDR[/LEN]] [dport PORT|RANGE] [proto PROTO] [bind-counter
COUNTER] [counter] [comment COMMENT] [log]
```

Description

Manage forward rules.

Create or delete a forward rule. Forward rules are checked after the routing decision and apply to packets that are not directed to the local device.

Options

Option	Description
`no`	Delete specified rule.
`accept|drop`	Accept or drop traffic matching the specified rule.
`pos POSITION`	The position in the list of rules you want to apply your command to.
`in IFACE`	Inbound Interface: The interface where the packet comes in.

continues on next page

[1] https://www.iana.org/assignments/protocol-numbers/

continued

```
[no] forward accept|drop [pos POSITION] [in IFACE] [out IFACE] [saddr ADDR[/LEN]]
[sport PORT|RANGE] [daddr ADDR[/LEN]] [dport PORT|RANGE] [proto PROTO] [bind-counter
COUNTER] [counter] [comment COMMENT] [log]
```

Option	Description
`out IFACE`	Outbound Interface: The interface where the packet is sent out.
`saddr`	Source IP Address/Subnet: The source IP address of the packet.
`sport`	Source Port: The source port of the packet.
`daddr`	Destination IP Address/Subnet: The destination address of the packet.
`dport`	Destination Port: The destination port of the packet.
`ADDR[/LEN]`	IP address specified as either a single IP address, or a whole IP subnet.
`PORT|RANGE`	Port specified as either a single port, or a range of ports.
`proto PROTO`	Protocol: The protocol type of the IP payload. Can be specified as a well-known protocol name (e.g. `tcp`, `udp`, `icmp`, `gre`, `esp`, `ah`, `ospf`, `vrrp`, `sctp`) or as an IANA protocol number (1–255). Use a protocol name or number when filtering traffic that does not use TCP or UDP ports, such as GRE tunnels (47) or OSPF (89). For a full list of IANA protocol numbers see https://www.iana.org/assignments/protocol-numbers/[1]. Common protocol numbers: Number / Name 1 / icmp 6 / tcp 17 / udp 47 / gre 50 / esp 51 / ah 89 / ospf 112 / vrrp/carp 132 / sctp
`bind-counter COUNTER`	Assigns the rule to COUNTER.
`counter`	Add a counter that only shows in the raw output. This counter is reset any time the firewall configuration is changed.
`comment COMMENT`	Add a comment to the rule.
`log`	Log traffic matching this rule.

[1] https://www.iana.org/assignments/protocol-numbers/

`[no] nat [pos POSITION] out IFACE [saddr ADDR[/LEN]] [comment COMMENT] [log]`

Description

Manage Network Address Translation rules.

Create or delete a NAT rule of NAPT type. NAPT is also known as IP masquerading, NAT overload and many-to-one NAT.

NAT can be used to hide private subnets behind a single public IP.

Options

Option	Description
`no`	Delete specified rule.
`pos POSITION`	The position in the list of rules you want to apply your command to.
`IFACE`	The outbound interface used for NAPT. Outgoing packets handled by this rule will appear to originate from the IP number configured (the primary address) or acquired (DHCP) for this interface.
`ADDR[/LEN]`	Specify that packets must originate from a specific subnet for this rule to apply.
`comment COMMENT`	Add a comment to the rule.
`log`	Log traffic matching this rule.

`[no] port-forward [pos POSITION] [in IFACE] [daddr ADDR[/LEN]] dport PORT|RANGE [to-dport PORT] to-daddr ADDR proto PROTO [comment COMMENT] [log]`

Description

Manage port forwarding rules.

Create or delete a Port Forwarding rule. Port Forwarding can be used to expose an internal device to the Internet.

Options

Option	Description
`no`	Delete specified rule.
`pos POSITION`	The position in the list of rules you want to apply your command to.
`in IFACE`	Inbound Interface: The interface where the packet comes in.
`daddr`	Destination IP Address/Subnet: The destination IP address of the packet.
`dport`	Destination Port: The destination port of the packet.
`to-daddr`	Forward Destination IP Address/Subnet: The destination address where the packets are to be forwarded.
`to-dport`	Forward Destination Port: The destination port where the packets are to be forwarded.
`ADDR[/LEN]`	IP address specified as either a single IP address, or a whole IP subnet.
`PORT|RANGE`	Port specified as either a single port, or a range of ports.
`proto PROTO`	Protocol: The protocol type of the IP payload.

continues on next page

continued

`[no] port-forward [pos POSITION] [in IFACE] [daddr ADDR[/LEN]] dport PORT|RANGE [to-dport PORT] to-daddr ADDR proto PROTO [comment COMMENT] [log]`

Option	Description
`comment COMMENT`	Add a comment to the rule.
`log`	Log traffic matching this rule.

`policy forward|input drop|accept`

Description

Set default filter policy.

Options

Option	Description
`forward|input`	Specify which hook to apply the policy to.
`drop|accept`	Specify which policy to apply.

`move input|forward|nat|port-forward id ID pos NEW-ID`

Description

Reorder rules.

Change the order of existing rules.

Options

Option	Description
`input|forward|nat|port-forward`	Specify from which rule set to move.
`ID`	ID of the rule to move.
`NEW-ID`	New ID of the rule to move.

`counter NAME [reset]`

Description

Show or modify counters.

Options

Option	Description
`NAME`	Create a new *named counter* called NAME.
`reset`	Reset the counter NAME.

`[no] log [limit none | ENTRIES/second|minute|hour|day ]`

Description

Handle logging for the firewall.

This command is used to globally enable or disable firewall logging and set a limitation on the maximum rate of entries that will be stored. Note that you need to add the “log” flag on individual rules as well.

Options

Option	Description
`no`	Globally disable logging.
`none`	No rate limitation for logging. Be warned that no limit may open for denial-of-service attacks!
`ENTRIES`	Number of entries to be logged in the specified time unit. Should be in the range 1-10000.
`second|minute|hour|day`	Time unit for the rate limit. The lowest possible rate is 2/day.

`[no] log input|forward|nat|port-forward NUM`

Description

Toggle logging flag on an existing firewall rule.

Options

Option	Description
`no`	Disable logging for the specified rule.
`input|forward|nat|port-forward`	The type of the specified rule.
`NUM`	The specified rule number.

`[no] alg ftp|tftp [PORT]`

Description

Application Level Gateway (NAT) helpers.

Options

Option	Description
`ftp|tftp`	Specify what helper should be enabled.
`PORT`	Specify port for control channel. If not specified, default values will be 21 for FTP and 69 for TFTP.

`show`

Description

Show all rules.

Firewall Configuration Overview

In order to see the current rules for the Firewall, the following command can be used:

```
example:/config/ip/firewall/#> show
Firewall is enabled.
Logging is Enabled, limited to 99 entries/hour

Prerouting Rules
Pos  Rule
001 prerouting dscp 53 in vlan1 saddr 10.0.2.0/24
002 prerouting dscp 46 in


Input Rules
Pos  Rule
001 input accept saddr 10.0.2.0/24
002 input accept in vlan1
N/A drop


Forward Rules
Pos  Rule
001 forward accept in vlan1 out vlan2
002 forward accept in vlan2 daddr 192.168.3.1/32 dport 80 proto tcp
N/A drop


NAT Rules (NAPT/masquerading)
Pos  Rule
N/A ipsec out accept
    "IPsec-NAT-exempt-policy"
001 nat out vlan2


Port-Forward Rules
Pos  Rule
001 port-forward in vlan2 dport 80 to-daddr 192.168.3.1 proto tcp


Application Level Gateway
Alg Helper
FTP  ENABLED  PORT 2222
TFTP ENABLED  PORT 69
```

Advanced: The command `show ip firewall raw` can be used in *admin-exec* mode to show the complete Linux *nftables* (*NFT*) ruleset, which contains the rules added in the WeOS firewall.

Counters

There are two types of counters available: temporary *inline counters* and *named counters*.

Temporary inline counters

Temporary *inline counters* are mainly intended for temporary debugging; see the section on *named counters* for more information about runtime-persistent counters. The *inline counters* are only visible in the raw firewall output and are reset every time the firewall configuration changes.

```
example:/config/ip/firewall/#> input accept dport 22 proto tcp counter
example:/#> show ip firewall raw
```

```
...
tcp dport ssh counter packets 10 bytes 443 accept comment "cli-input-009"
...
```

Named counters

A *named counter* can be created and bound to one or more firewall rules. A *named counter* can be bound to *input* or *forward* rules. A *named counter* is persistent during runtime.

Example of a counter that counts all inbound web interface traffic:

```
example:/config/ip/firewall/#> counter web
example:/config/ip/firewall/#> input accept dport 80 proto tcp bind-counter web
example:/config/ip/firewall/#> input accept dport 443 proto tcp bind-counter web

example:/#> show ip firewall counter web
Users
001 input accept dport 80 proto tcp bind-counter web
002 input accept dport 443 proto tcp bind-counter web

Value
packets 26 bytes 1560
```

Status of Firewall Rules

The `show ip firewall` command in *admin-exec* can be used to show the status of the applied firewall rules, and the output looks identical to the output in the configuration context.

The following keywords can be used in combination with the `show ip firewall` command:

- *counters*: show detailed information about the counters.
- *conntrack*: show information about the active network connections.

```
example:/#> show ip firewall conntrack
[DESTROY] udp      17 src=169.254.159.156 dst=239.255.255.250 sport=1900 dport=1900
[UNREPLIED] src=239.255.255.250 dst=169.254.159.156 sport=1900 dport=1900
[DESTROY] udp      17 src=198.18.1.1 dst=239.255.255.250 sport=1900 dport=1900
[UNREPLIED] src=239.255.255.250 dst=198.18.1.1 sport=1900 dport=1900
 [UPDATE] tcp      6 432000 ESTABLISHED src=198.18.1.99 dst=198.18.1.1 sport=36934
dport=22 src=198.18.1.1 dst=198.18.1.99 sport=22 dport=36934 [ASSURED] mark=0 use=1
 [UPDATE] unknown  2 510 src=198.18.1.1 dst=224.0.0.1 [UNREPLIED] src=224.0.0.1
dst=198.18.1.1 mark=0 use=1
```

2.12.2. Stateless NAT Guide

Introduction

This document gives an overview of Stateful NAT vs. Stateless NAT before diving into a few common use-cases, and concludes with an overview of the steps required to configure.

> **NOTE**
>
> Stateless NAT is also commonly referred to as '1-1 NAT' (one-to-one NAT) and DNAT (destination NAT)

For an example use-case, see the Stateless NAT HowTo.

Overview

All hosts on a *public* network have a unique IP address used to route IP packets to their intended destination. Scarcity of IPv4 addresses forced the ever expanding Internet to adopt *private* IP addresses, see RFC 1918[1], for local networks. When connecting such a local network to the public Internet the private addresses must be translated to one or more public addresses; referred to Network Address Translation (NAT). The unit conducting the translation is typically referred to as a NAT gateway.

There are two common techniques:

- Stateful NAT, NAPT, or IP masquerading: share one public IP address
- Stateless NAT, 1-to-1 NAT, or DNAT: map range of public addresses

Stateful NAT

Stateful NAT is for sharing one public IP address, it is also commonly referred to as Network Address Port Translation, or NAPT, and also as IP masquerading.

```
                    .-----.
                    | WWW |    203.0.113.42
                    '--+--'
                       |
                  .--.-.
                 ( (    )__
                (_,  \ ) ,_)  Internet/Intranet
                  '-'--`--'
                    |
  ^                 |.33       10.0.0.0/24
  |            .---+--.
Public         |      | vlan2
------         |  GW  |
Private        |      | vlan1
  |            '---+--'
  V                 |.1
                    |          192.168.1.0/24
     ---+------+---+----------+---------
        |          |          |
        |.11       |.5        |.139
      .-+--.     .-+--.     .-+--.
      | H1 |     | H2 |     | H3 |
      '----'     '----'     '----'
```

Figure 1: All private hosts masqueraded as 10.0.0.33 behind GW.

The NAT gateway has only one public IP address which is shared with all hosts on the private network. E.g., when `H1` sends a packet to `WWW`, the gateway replaces the source IP from 192.168.1.11 to 10.0.0.33 before forwarding the packet to the public side.

[1] https://tools.ietf.org/html/rfc1918

However, since all hosts on the private network map to the same public address, the gateway must also modify the (UDP/TCP) port numbers for each connection and keep a record of all connections to be able to track the return traffic. Stateful NAT is not covered further here.

Stateless NAT

With stateless NAT, there is exactly one public address for each private address. Commonly used for 1-to-1 NAT where a range of public addresses map to a range of internal/private addresses.

In the below image the range 10.0.2.0/24 can be used to address each of the devices `H1` to `H3` from the other side of the gateway. For each address the gateway translates the 10.0.2.0/24 range to 192.168.1.0/24.

```
                  .------.
                  | WWW |     203.0.113.42
                  '--+--'
                     |
                .--.-.
              ( (    )__
             (_,  \ ) ,_)  Internet/Intranet
               '-'--`--'
                  |
  ^               |.33       10.0.0.0/24
  |          .---+--.
Public       |      | vlan2
------       |  GW  |
Private      |      | vlan1
  |          '---+--'
  V               |.1        (10.0.2.0/24)
                  |          192.168.1.0/24
     ---+------+---+----------+---------
        |          |          |
        |.11       |.5        |.139
      .-+--.     .-+--.     .-+--.
      | H1 |     | H2 |     | H3 |
      '----'     '----'     '----'
```

Figure 2: All private hosts have a corresponding public IP.

With stateless NAT, each host on the private side is known by a unique address on the public side. For example, H1 with private address 192.168.1.11 is known by address 10.0.2.11 on the public side, H2 with private address 192.168.1.5 is known by address 10.0.2.5 on the public side, etc. The address translation at the NAT gateway is a simple operation, just swapping the network part. Thus, the NAT gateway does not need to keep any state.

Configuration

Stateless NAT configuration is a sub-context to the IP configuration context in the CLI:

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> stateless-nat
example:/config/ip/stateless-nat#>
```

Syntax

```
dnat in IN-IFACE dst ADDR[/LEN] to-dst ADDR [passive]
```

The `dnat` setting performs stateless network address translation (NAT) on the IP destination address. It is worth mentioning that Addresses are translated in both directions.

`in IN-IFACE`

Description
inbound interface (typically the public interface)

`dst ADDR[/LEN]`

Description
address or range to translate *from* with an optional prefix len. If no `/LEN` is given, a single address is translated, same as `/32`

`to-dst ADDR`

Description
address or range to translate *to*. Notice the lack of `/LEN` here

`[passive]`

Description
Optionally deactivate the DNAT rule without removing it from the configuration

> **NOTE**
> In interactive mode the CLI will ask if you wish to add a *loopback route* to the `dst` address or range. This is useful when also enabling *Proxy ARP* for this address range. For more information, see Multiple private networks and use of proxy ARP.

Apply configuration changes and return to *Admin Exec* context using:

```
example:/config/ip/stateless-nat#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

2.13. Train

2.13.1. TTDP (IEC 61375-2-5)

Glossary

- **CN** - Consist Network
- **ECN** - Ethernet Consist Network
- **ECSC** - ETB Control Service Client[1]
- **ECSP** - ETB Control Service Provider[2]
- **ETB** - Ethernet Train Backbone
- **ETBN** - Ethernet Train Backbone Node
- **TCMS** - Train Control and Management System
- **UUID** - Universally Unique IDentifier[3]
- **TRDP** - Train Real-Time Data Protocol

Introduction

Train Topology Discovery Protocol (TTDP) is part of the IEC 61375 series of protocols.

In short, TTDP dynamically discovers the **backbone routers (ETBNs)** and **consist networks (ECNs)** in the train, and the way that they are connected. Furthermore, TTDP assigns IP addresses to the ETBNs and ECNs, and establishes routing and NAT entries in ETBNs based on these IP assignments. TTDP is designed to accomplish this as a distributed protocol running on the ETBNs.

Overview

TTDP is managing train backbone inauguration (ETB inauguration) in train networks adopting the IEC 61375 train topology model. TTDP is a distributed protocol running on the train router nodes, here referred to as ETB nodes or simply ETBNs.

The TTDP protocol provides a wide range of services described in the sections below.

Link Aggregation Control

TTDP acts as a link aggregation control protocol on the backbone network. Each ETBN sends a message, called TTDP HELLO, on each port on each aggregated link, i.e., aggregates `dir1` and `dir2`. This process gives both information of the MAC address of the neighbouring ETBN and control over which ports are available in the aggregate. The aggregates are setup as redundant communication links. All communication is done over one link until that link is not available any more, which is discovered by time outs on TTDP HELLO messages.

Discovery of Train Topology

With TTDP, ETBNs will discover other ETBNs present on the backbone and exchange information. This TTDP signalling in turn allows each ETBN to gather information about:

- **Consists:** The identity of present consists (Consist Universally Unique Identifier, or simply `CstUUID`) and their order/position in the train backbone.
- **ETBNs:** The identity (ETBN MAC address) and order of the ETBNs on the backbone, and also the consist where each of the ETBNs reside. The ETBNs even learn the position of “missing” ETBNs (known, but currently down); this is true as long as at least one ETBN in each consist is up and running.

[1] See communication profile
[2] See communication profile
[3] See RFC 4122

- **ECNs:** Every ETBN advertises information about the consist networks (ECNs) present in its consist. Thereby, each ETBN is able to gather information on all ETBNs and ECNs, and how they are interconnected.

Agreement on the network topology layout is accomplish in the following steps, each ETBN sends out a TTDP TOPOLOGY frame on a high priority VLAN, i.e., ID 492. The frame contains information of the IDs of the neighbouring ETBNs and the ETBN's view of how all ETBNs in the backbone are connected. The frame also contains the ECN connections for the consist that the ETBN reside within. A CRC called `etbTopoCnt` is calculated, by each ETBN, over its correct view of the network. Once all nodes have the same view, i.e., the same `etbTopoCnt` value, the network can be inaugurated if inauguration is not inhibited.

IP Address Assignment

TTDP is capable of assigning IP addresses dynamically based on the discovered topology. Addresses are allocated from a range defined in IEC 61375-2-5, and concern:

- **ETBN addresses on the ETB subnet:** An IP address is allocated to each ETBN on the ETB subnet.
- **Train-wide IP subnet allocation for ECN subnets:** ECNs typically make use of local (private) addresses in range 10.0.0.0/18. TTDP allocates a train-wide/18-subnet to each of the discovered ECNs.
- **Virtual addresses on ETB subnet:** In topologies where ECNs are connected to two or more ETBNs for redundancy, TTDP allocates a virtual IP address to be shared between the ETBNs on the ETB subnet. A virtual IP address is assigned for each ECN with redundant ETBNs.

It should be noted that the multicast DNS discovery service is not intended to be used on the ETB subnet, and WeOS will not send mDNS broadcasts from the ETB interface.

External Interfaces

The canonical interface used for control, status, and diagnostics between the ETBNs and devices on the ECNs consists of the TRDP telegrams defined in IEC61375-2-3 Annex E.

```
     .----------------------------.
    /           CstUUID            \ cst1
    '------------------------------'
       .-------.    .-------.
     --+       +----+       +--
<-dir1| ETBN1 |    | ETBN2 |dir2->
     --+       +----+       +--
       '-+--+--'    '--+--+-'
         |  |          |  |
       --+----------------+-- CN1
            |          |
         ---+----------+-------- CN2
```

Figure 1: A consist with two consist networks managed by two redundant ETB nodes.

Usage Recommendations

Configuration Boundaries

When configuring the ECNs connected to the ETBNs it is recommended that the ECN with ID 1 is connected to ETBN with **`local-id`** 1. Once all IDs for ECNs connected to the first ETBN has been assigned, the ECNs connected to the next ETBN can be assigned. The IDs must be in sequence without any gaps. The configuration of the consist topology is done though the **`node`** command. It is important that the **`node`** configuration is identical on all nodes within a consist. The information

is shared in the TTDP TOPOLOGY frame and is used by other ETBNs to discover the topology and calculate the `etbTopoCnt` value.

The configuration of ECN ID to VLAN ID is done through **`ecn`**. In the consist shown in Figure 1, it is possible to connect different VLAN IDs to the same CN if the ports in the VLAN are untagged. The **`ecn`** setting is local information that is not shared in the TTDP TOPOLOGY frame.

The ports configured to be included in the aggregates `dir1` and `dir2` must be bypass ports. This requirement makes it possible to handle black car scenarios, i.e., that some ETBN do not have power and should instead acts as cables.

Inauguration Inhibition

Inauguration must be inhibited on all ETBNs/ECSPs by means of ECSC_Control TRDP messages (or SNMP) once the train composition is in inaugurated state.

Even when TOPOLOGY frames received from other ETBNs indicate inauguration inhibition an ETBN can join once the correct `etbTopoCnt` has been calculated. A so called `NodeRecovery` can be made if either the ETBN is an intermediate node, which was previously a known node, that disappeared and is back again. Or, the ETBN is an intermediate node and wakes up late in a consist containing at least an already inaugurated ETBN.

With the command **`recovery-mode`** it is possible to change this behaviour to require a recovering ETBN to wait for ECSP_CTRL data from a specific multicast IP. This will make the node wait to join the ETB until inauguration inhibition has been lifted.

Nodes not having Inauguration Inhibition locally set can re-inaugurate as a new train composition, and become lost from the remainder of the train, in cases of ETB instability, malfunction or overload. This will lead to loss of connectivity between these nodes and the remainder of the train.

This gets indicated with a lengthening indication in the train with Inhibit still active and can be resolved by all nodes by removing Inhibition to allow the lost nodes back in again. Having Inauguration Inhibition set locally on every node prevents this from occurring.

Remote Inhibition

When using the “remote inhibition” functionality as mentioned in IEC61375-2-5:2014 section 8.7.6, there are different options available for how the value should be reported to the ECSC and other applications. The **`inhibit-report-mode`** command, described below, controls the semantics of the **`etbInhibit`** field in the ECSP_STATUS telegram, which can be used for this purpose.

ETB Sink/Source Data Handling

All ETB data should be verified based on the current `etbTopoCnt` value to make sure no data from previous inauguration is sent or received by end devices.

When an inauguration occurs, all end devices shall disable ETB traffic until the ETBN again is in state inaugurated and a new `etbTopoCnt` value is available and stable. This is recommended even for traffic sent with a `etbTopoCnt` value of 0, intended for intra-consist communication.

Multicast Streams

When setting up and sending/receiving multicast streams that is not TRDP related, it is highly recommended to not use TRDP ports 17224 and 17225, since the TRDP stack will attempt to parse data coming in through these ports, leading to reduced performance.

Additionally, it is highly recommended to not use the TRDP related multicast group 239.193.0.1, unless the traffic is meant for all ECSPs.

Also, please ensure that the train backbone can handle the loads of all multicast streams being sent/received from/to the consist networks.

Configuration

TTDP can be activated from the top-level configuration context in the CLI.

```
example:/#> configure
example:/config/#> ttdp
Activating TTDP with default settings.
example:/config/ttdp/#>
```

> **NOTE**
>
> TTDP will be enabled by default when entering the configuration context.

For configuration examples, see

- Single consist net with one ETB node
- Single consist net with two redundant ETB nodes
- Two consist nets with two redundant ETB nodes
- Port settings for ETB backbone ports
- ECN-over-ETB with ECN subnetting

Syntax

`[no] enable`

Description

Enable TTDP.

> **NOTE**
>
> Simply entering the TTDP config context will enable TTDP when leaving the TTDP context or config context.

Options

Option	Description
`no`	Disable TTDP. The current configuration is saved, for when you re-enable the service again. **NOTE** The TTDP configuration can be completely removed from the **config context** using **no ttdp**.

`[no] operational-mode <full|etbn|off>`

Description

Select operational mode, either *full*, *etbn*, or *off* mode.

In `full` mode, all ECSP and ETBN services are available. This includes the ETBN functionality as specified in IEC61375-2-5, used for ETB inauguration among other things, and the ECSP functionality as specified in IEC61375-2-3, which includes CSTINFO message exchange, TCN-DNS, TRDP services, and so on.

In `etbn` mode, all functionality related to IEC61375-2-3, specifically the ECSP functionality, is turned off for this node. The node will not read the static consist information[1], take part in the exchange of CSTINFO messages, and so on. This applies regardless of any other settings related to the ECSP functionality described herein.

In `off` mode, all ECSP and ETBN services are disabled. No other configuration options are available, with the exception of those that relate to the *custom TRDP telegrams* functionality.

> **NOTE**
>
> This setting is independent of the *custom TRDP telegrams* functionality, which is configured separately. That functionality remains available, regardless of which operational mode is selected. See the documentation section about custom TRDP telegrams for more information.

> **NOTE**
>
> Selecting `etbn` mode disables the TRDP interfaces specified in IEC61375-2-3 Annex E.

`[no] uuid <UUID-STRING>`

Description

Set the consist UUID. Each ETBN belonging to the same consist must use the exact same UUID, and no two consists may have the same UUIDs. Thus, the UUID is what defines individual consists.

Options

Option	Description
`UUID-STRING`	One of the following: • A literal UUID string, in the RFC 4122 format. *This is the preferred method of setting the consist UUID.* `etbn1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111`

continues on next page

[1] See communication profile

continued

`[no] uuid <UUID-STRING>`

Option	Description

	• The string “auto-string”, followed by a *seed string* used to generate the UUID automatically. In this case, the *seed string* is used to generate the UUID using the algorithm in RFC 4122 chapter 4.3 (“Name-Based UUID”). Devices configured to use the same *seed string* will generate the same consist UUID.
	`etbn1:/config/ttdp/#> uuid auto-string "uic948002343044"`
	• The string “auto-mac”. In this case, the UUID is generated, using the same algorithm as above, based on the base MAC address of the device. Different devices will generate different UUIDs.
	`etbn1:/config/ttdp/#> uuid auto-mac`

`no`	Reset UUID to all zeros.

`[no] node ETBN_NB ecn ECN_ID[,ECN_ID]`

Description

Add ECN(s) to an ETBN to describe the consist topology.

Options

Option	Description
`ETBN_NB`	The static relative position of an ETBN in the consist. Not to be confused with ETBN id which is assigned to the ETBN during inauguration and depends on the train composition. Valid range: 1-32.
`ECN_ID`	The consist net id. Valid range: 1-4. **NOTE** Limited to max 4 consist networks per consist.
`no`	Remove all ECNs from ETBN_NB. Syntax: no node ETBN_NB.

`[no] ecn ECN_ID IFACE <PREFIX_LEN>`

Description

Map ECN with ECN_ID to interface IFACE.

Options

Option	Description
`ECN_ID`	ECN id. Valid range: 1-4.

continues on next page

continued

`[no] ecn ECN_ID IFACE <PREFIX_LEN>`

Option	Description
	NOTE ECN_ID must fulfil the following criterias: 1. Each CN must have a unique number 2. There must be one and only one CN with the number 1 3. There can be no gaps in the sequence of assigned numbers **NOTE** Limited to max 4 consist networks per consist.
`IFACE`	The interface used for the ECN_ID.
`PREFIX_LEN`	The prefix length of this ECN, if different to the address used by the ECN interface. When the ECN is locally split, this optional parameter must be specified (most often as **18**) to specify the prefix length of the entire ECN, for TTDP R-NAT purposes.
`no`	Remove the ECN_ID's interface mapping. **NOTE** This does not remove the Interface itself, just the mapping to the ECN.

`[no] dir1 PORT[,PORT]`

Description

Set ETBN backbone port(s) for direction 1.

Options

Option	Description
`PORT`	The port(s) to be added to aggregate in direction 1. **NOTE** Limited to max 2 ports per direction.
`no`	Remove all ports from dir1's aggregate.

`[no] dir2 PORT[,PORT]`

Description

Set ETBN backbone port(s) for direction 2.

continues on next page

continued

`[no] dir2 PORT[,PORT]`

Options

Option	Description
`PORT`	The port(s) to be added to aggregate in direction 2. **NOTE** Limited to max 2 ports per direction.
`no`	Remove all ports from dir2's aggregate.

`[no] local-id ETBN_NB`

Description

Set this ETBN's static position within the consist.

Options

Option	Description
`ETBN_NB`	The static relative position of an ETBN in the consist. Not to be confused with ETBN id which is assigned to the ETBN during inauguration and depends on the train composition. Valid range: 1-32. **NOTE** ETBN_NB must fulfil the following criterias: • Each node must have a unique number • There must be one node with the number 1 • There can be no gaps in the sequence of assigned numbers
`no`	Set to 0.

`[no] backbone-id <ID>`

Description

Set backbone id to `ID`.

Options

Option	Description
`ID`	Backbone id. Default: 0 (TCMS). Valid values: • `0` or `tcms` — Operational backbone (TCMS) • `1`, `multimedia` or `mm` — Multimedia backbone

continues on next page

continued

`[no] backbone-id <ID>`

Option	Description
	NOTE Backbone id 2 and 3 are not supported.
`no`	Reset to defalut 0 (TCMS)

`[no] backbone-iface <IFACE>`

Description

Set VLAN used for TTDP signalling on the ETB. Default: vlan2.

Options

Option	Description
`IFACE`	Interface to be used on ETB. NOTE Ensure that IGMP snooping is **disabled** on IFACE.
`no`	Reset to default (vlan2).

`[no] ecsp-addr <IP-ADDRESS>`

Description

Local ECSP address. Should be set to a local address of the `ecsp-iface` interface, on which the CN-facing ECSP service runs.

Options

Option	Description
`IP-ADDRESS`	Local ECSP address. Default 10.0.0.1.
`no`	Reset to default (10.0.0.1).

This setting is only available if the operational mode is set to `full`.

`[no] ecsc-addr <IP-ADDRESS>`

Description

ECSC address. Telegrams to the local ECSC will be sent to this address.

continues on next page

continued

`[no] ecsc-addr <IP-ADDRESS>`

Options

Option	Description
`IP`	ECSC address. Default 10.0.0.99.
`no`	Reset to default (10.0.0.99).

This setting is only available if the operational mode is set to `full`.

`[no] ecsp-leader-addr <IP-ADDRESS>`

Description

To enable sync of TTDB set this to the virtual IP address for the CN that governs ECSP Master/Backup mode.

> **NOTE**
>
> Only set this when using redundant ETBN's in a consist. That is, when VRRP instance(s) for TTDP redundancy have been configured in this consist.

Options

Option	Description
`IP-ADDRESS`	ECSP leader address. Default 0 (disabled)
`no`	Disable sync of TTDB between Master/Backup ECSP.

This setting is only available if the operational mode is set to `full`.

`[no] ecsp-iface <IFACE>`

Description

Local ECSP interface. The CN-facing ECSP service will run on this interface.

Options

Option	Description
`IFACE`	Local ECSP interface. Default: vlan1.
`no`	Reset to default (vlan1).

`[no] mroute group <MULTICAST_GROUP> out <NET_ID[,NET_ID]>`

Description

Configure custom IEC 61375 multicast routes.

The specified MULTICAST_GROUP will be routed to NET_ID from every other connected ECN as well as from the local ETB.

continues on next page

continued

`[no] mroute group <MULTICAST_GROUP> out <NET_ID[,NET_ID]>`

Options

Option	Description
`MULTICAST_GROUP`	Must be a valid multicast group (RFC2365) that is not within the reserved IEC61375-2-5 group range 239.192.0.0/24.
`NET_ID`	NET_ID can be a numeric ECN_ID or the special value “etb”, which maps to the ETB that the local ETBN is connected to. In this case the multicast group will be routed from all local CNs to the ETB.
`no`	Removes all configured multicast routes.

> **NOTE**
>
> This setting needs to be configured on all the potential source and receiver ETBNs.

`[no] recovery-mode <default|wait IP-ADDRESS>`

Description

Specifiy how to handle recovery of a lost node.

Options

Option	Description
`wait <IP-ADDRESS>`	The IP-Address as listening address in wait/deferred mode
`no`	Reset to default recovery mode.

> **NOTE**
>
> In `wait` mode, the node will wait until it receives ECSP_CTRL data on the multicast address `IP-ADDRESS`, and positively reads that inhibition is not set from that data, before activating the bypass relays (i.e. joining the ETB) and starting to transmit TOPOLOGY frames. Thus, an ECSC **must** be up and running, signalling ECSP_CTRL inhibit off (0), before the node will function properly.

`[no] cn-bitmask-endianness <little|big>`

Description

Select CN bit mask endianness little (default) or big.

Options

Option	Description
`no`	Reset to default CN bit mask endianness (little).

continues on next page

continued

`[no] cn-bitmask-endianness <little|big>`

> **NOTE**
>
> This setting is only useful for interoperability.

`[no] dns-lookup-mode <local|global>`

Description

Set the (TCN-)DNS lookup preference mode to “local scope first” or “global scope always”.

Options

Option	Description
`no`	Reset to default lookup order, i.e. Static Consist Info
`local`	Sets the lookup order to try the local scope before the global scope
`global`	Sets the lookup to always use the global scope

This setting is only available if the operational mode is set to `full`.

`[no] etbn-ctrl-bypass`

Description

Enable (or disable) control of ETB bypass relays via the ETBN Control Interface (see IEC61375-2-3 Annex E)
Disabled: the **byPassCtrl** field of the ETBN_CTRL telegram is ignored
Enabled: the **byPassCtrl** field of the ETBN_CTRL telegram is used used for controlling the ETB bypass relays
The semantics of the **byPassCtrl** field is specified in IEC61375-2-3 Annex E.

Options

Option	Description
`no`	Disable control of ETB bypass relays via ETBN Control interface

This setting is only available if the operational mode is set to `full`.

`[no] inhibit-report-mode <mode>`

Description

Choose how the **etbInhibit** field of the ECSP_STATUS telegram should be populated.

The semantics of this field are specified in IEC61375-2-3 Annex E, though not very clearly. As an additional limitation of the service interface specified in Annex E, there is no way for the ECSC or any other ECN device to read the global “remote inhibition” flag. To resolve this, this configuration option makes it possible to alter the behaviour of the etbInhibit field in the periodic ECSP_STATUS telegram to allow it to be used for that purpose.

continues on next page

continued

`[no] inhibit-report-mode <mode>`

The following options are available:

Options

Option	Description
`0`	Default behaviour. As specified in IEC61375-2-3 Annex E • 0: n/a • 1: inhibit not requested on ETB • 2: inhibit set on local ETBN • 3: inhibit set on remote ETBN • 4: inhibit set on local and remote ETBN “Remote ETBN” only considers ETBNs in the local train composition. This is the default.
`1`	As mode 0, but “remote ETBN” now means “non-local ETBN in the local, or the remote, train composition”.
`2`	Bitfield mode. The etbInhibit field is now a bitmask containing the following bits: • 1: local ETBN • 2: remote ETBN in local train composition • 4: remote ETBN in remote train composition
`3`	As mode 0, but “ETBN” now means “train composition”.

This setting is only available if the operational mode is set to `full`.

`[no] ecsp-inhibit-sync`

Description

Enable or Disable synchronisation of inauguration inhibition between redundant ETBNs (ECSPs).

When *Enabled*, and this node is a backup ECSP (i.e. backup VRRP router) for the ECN with local *CN ID* 1, it will attempt to copy the **local** inauguration inhibition value from the master ECSP in the local consist. The master node will have had its local value set by the ECSC or some other external agent. This copying is performed by passively monitoring the TTDP TOPOLOGY frames from the local consist.

After the value is copied, the backup ECSP will behave as if the value had been set locally on itself, until a ECSP masterhood change takes place. In case there are several backup ECSPs, all backups that have this setting enabled will copy the value from the master ECSP as described above.

The purpose of this function is to allow for seamless master router/ECSP failover without losing the global inauguration inhibition state, in case it is only set in one consist. The function performs best if used on relatively simple consist ECN configurations, and will in certain more complex cases not work, if the local master ECSP cannot be unambiguously identified.

When *Disabled*, no special action is taken when this node is a backup router/ECSP. The inauguration inhibition value will not be copied from the master ECSP.

continues on next page

continued

`[no] ecsp-inhibit-sync`

This setting is only available if the operational mode is set to `full`.

`[no] ecsp-ttdb-sync`

Description

Select the behaviour of the ECSP-SYNC function. This function lets backup ECSPs sync their internal −2-3 state with that of the master ECSP in the local consist. The following options are available:

Options

Option	Description
`0`	Disable this function. Note that the backups will still sync from the master, in certain cases when this is required. Backups will not take part in the −2-3 inauguration process on the ETB and will transmit CSTINFOCTRL messages and fetch fetch CSTINFO for other consists as normal, over the ETB. This is the default.
`1`	Sync non-local consists only. The backups will only request CSTINFO for other consists from the local master ECSP. Backups will not transmit CSTINFOCTRL.
`2`	Sync all consists. Backups will request the entire TTDB from the local master.
`no`	Disable synchronisation of the TTDB between ECSPs (0).

This setting is only available if the operational mode is set to `full`.

`[no] tcp`

Description

Enable/disable TCP support for certain TRDP telegrams.

This setting affects the following telegrams:

ComID	Name
102/103	Train directory info
104/105	Static consist information
106/107	Train network directory
108/109	Operational train directory
110/111	Read complete TTDB
122/123	Confirm/correction
140/141	TCN DNS

This setting is only available if the operational mode is set to `full`.

`[no] cst-mgrp-dnat`

Description

Enable or disable dynamic NAT for multicast groups.

`[no] ecn-over-etb VID[,VID...]`

Description

Enable or disable automatic forwarding of ECNs (and/or other VLANs) over the ETB.

This setting is used to automatically forward selected VLANs over the ETB, and is useful for consists without dedicated ECN cabling. Note that such cabling may not be used in conjunction with this feature, unless measures are taken to avoid switching loops.

The traffic is sent over the ETB cabling using VLAN tags, and not forwarded outside of the local consist.

The VLANs are specified as a comma-separated list of VLAN IDs. Up to four VIDs can be specified.

Options

Option	Description
`VID`	List of VLAN IDs to automatically forward over the ETB while respecting consist boundaries. Up to four VIDs can be specified.
`no`	Disable this function.

`[no] cstinfo-validate`

Description

Validate and verify the CSTINFO data file (`/cfg/cstinfo.json`).

Attempt to validate a preexisting CSTINFO data file against the internal JSON schema and the additional semantic rules. If successful, calculate a SHA256 hash over the contents of the file, and optionally store this hash on the device for future verification upon (re)start of the IP-TCN stack.

The results of the verification will be stored in the WeOS audit log. If syntactically and semantically correct, the CSTINFO data file will be used by the IP-TCN software regardless of the result of the hash verification.

Options

Option	Description
`no`	Remove any stored SHA256 hash of the CSTINFO data file.

`[no] custom-trdp NAME`

Description

Enter or create a *custom TRDP telegram* context.

continues on next page

continued

`[no] custom-trdp NAME`

This command is used to configure the *custom TRDP telegram* function. See the custom TRDP telegrams configuration section for more information.

`[no] trdp-validate`

Description

Validate the dataset used for the *custom TRDP telegram* function.

This command is used to configure the *custom TRDP telegram* function. See the custom TRDP telegrams configuration section for more information.

`show`

Description

Show a summary of ttdp settings. This includes the relevant user-configurable parameters in this context, as well as the custom TRDP telegram configuration.

2.13.2. Status

To see the current status of TTDP, run the `show ttdp` command in `Admin context`, like so:

```
example:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 1
etbn-ip      : 10.128.0.1
local-id     : 1
own-mac      : 0c:40:79:45:55:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0×7af91a80
EtbTopoCnt   : 0×ec518201
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan1     10.0.0.1/21     1

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:40:79:45:55:00      Direct
      2      0c:40:79:71:03:00      Direct



CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111  1       Direct     10.128.64.0/18
22222222-2222-2222-2222-222222222222  2       Direct     10.128.128.0/18
```

2.13.3. Custom multicast routes

See a simple multicast example on how to configure custom multicast routes.

2.13.4. IEC 61375-2-3 (Communication Profile)

Introduction

The communication profile (IEC 61375-2-3), which is part of IEC 61375, ensures interoperability between consists of trains with respect to the exchange of information.

For this it defines all items which are needed, the so-called communication profile, of the Train Communication Network, as defined in TTDP (IEC 61375-2-5).

Overview

This part of the IEC 61375 revolves around three major parts:

1. Communication profile configuration
2. TCN-DNS
3. ECSP / ECSC

Communication profile configuration

The communication profile configuration file specifies functions and vehicles, and all their properties (such as id, name, etc.), in the local consist. This is also known as the “CSTINFO data”. This configuration file need to be consistent across all ETBNs that are part of the same consist (redundant nodes).

The data is structured hierarchically. The top scope of CSTINFO represents the local consist. As such, the root object contains information about the consist, as a list of consist-level functions, as well as the vehicles that the consist contains. These vehicles then also contain functions.

A ‘function’ in this context means an individual device, an interface on a device, or a multicast group - anything that can be resolved to one IP address using TCN-DNS. Functions can belong directly to a consist (which is often the case for multicast groups) or to a vehicle, which in turn belongs to a consist.

A ‘vehicle’ refers to an actual train car, a physical entity, which constitutes part of a consist. A consist can include one or more vehicles.

See the sections on CSTINFO and the TTDB data model in IEC 61375-2-3 for a more detailed discussion of these terms.

As mentioned above, the CSTINFO data for the local consist is provided to the system by means of a file on the ETBN device. This file is in JSON format, must be placed in a specific location, described below, and conform to the supplied JSON schema.

Consist and vehicle properties

The TTDB data model allows for opaque consist and vehicle “properties” data that is not parsed or interpreted by the implementation. This data is provided in the JSON file as Base64-encoded strings, using the alphabet in RFC 4648. To make the file more user-friendly, the encoded data is formatted as an array of strings (so that line breaks may be inserted between the array elements). The implementation will concatenate the string array, in order, and decode the Base64 data when the file is processed. The decoded data will not be parsed or interpreted in any way except for being included in the CSTINFO telegrams that the device transmits as per IEC 61375-2-3.

Since several vehicles in the file may use identical properties data, the file is structured in a way where vehicle properties are referenced indirectly - each vehicle object has an optional `propSlot` attribute, which maps to a `slot` attribute of a properties object in the `vehPropList` array. Consist properties are defined directly in the consist object. See the comm profile configuration schema for additional details.

TCN-DNS (Train Communication Network Domain Name System)

TCN-DNS specifies one of two interfaces (where the other interface is DNS according to RFC1035) that an application can use to lookup names, much like a DNS lookup, but with a few differences:

1. Queries whose URIs resolve to any local device will resolve relative to the querier, not the resolver.
2. Some queries will resolve to two IP addresses which represents an IP range, which can include more than two functions.

> **NOTE**
>
> Point 2 only applies to TCN-DNS via TRDP.

ECSP / ECSC (ETB Control Service Provider / ETB Control Service Client)

The ECSP, running on each ETBN, handles all communication and keeps a database of all vehicles, functions and operational states and statuses.

The ECSC, running on a selected device on the train, communicates with the ECSP in order to set and get various data sets.

TCN ECHO Server Interface

The TCN ECHO Server function can be used by end devices, running a compatible TCN ECHO Client function, to perform reachability and latency measurements to the Server (running on a WeOS device configured as an ETBN). When performing latency measurements, no guarantees are given regarding response time, other than that the TCN ECHO Server will attempt to respond as fast as possible.

A TCN ECHO Client needs to send an TCN ECHO request telegram via TRDP. When the TCN ECHO Server receives such a telegram, it will, as quickly as possible, send an TCN ECHO response telegram back to the TCN ECHO Client.

See Annex A - TCN ECHO request/response telegram below for a description of the fields and content of the TCN ECHO request and response telegrams.

Configuration

To setup the communication profile, perform the following steps:

1. Create a communication profile configuration file (see configuration options below).
2. Ensure that the file conforms to the provided JSON schema.
3. (Optional) Generate a SHA256 hash of the contents of the file using any suitable external tool.
4. Copy that configuration file to **/cfg/cstinfo.json** on the device. Do this for every ETBN in the consist, including redundant ETBNs.
5. (Optional) use the `cstinfo-validate` command in the `config/ttdp` context of the WeOS CLI to validate the file, verify its integrity by comparing the checksum from step 3. to a hash computed by WeOS, and store the hash for later verification.

> **NOTE**
>
> Ensure all devices each have a json file with correct content.

> **NOTE**
>
> Ensure the filenames are correct. That is, the communication profile configuration file ends up with the name **cstinfo.json** in **/cfg/** on the device.

> **NOTE**
>
> Validation and verification of the CSTINFO file using a SHA256 hash is optional. Upon every start of the IP-TCN software, the provided file will be checked for correctness and a new SHA256 hash will be calculated. Regardless of whether this hash matches any stored hash, the results of this comparison will be stored in the WeOS audit log. As long as it is syntactically correct, the file will be used for IEC 61375-2-3 services, regardless of whether the hash verification succeeds or not.

For an example configuration of a communication profile, see Single consist net with two redundant ETB nodes HowTo

Configuration Options

The structure of the configuration file must conform to comm profile configuration schema.

Additionally, the CSTINFO data contained in the file must follow the data model described in IEC 61375-2-3. This implies the following (though this list is non-exhaustive). If these rules are not followed, the file will be rejected as “semantically incorrect”, even though it may still be syntactically valid (i.e. conform to the JSON schema).

- Functions must be related to (by means of the `etbId` attribute) either an ETB that has been defined in the `etbInfoList`, or use the special value “255”.
- Functions must have a `cnId` attribute that corresponds to a CN that has been defined (by means of `cnCnt`, since CN numbers start at 1 and are consecutive) for the ETB that it is related to, or use the special value “0”.
- Vehicles that use properties (via the `propSlot` attribute) must reference a properties object that is defined in the `vehPropList` array. The `propSlot` value must match a `slot` value.
- The `cstVehNo` numbers of defined vehicles must start at 1 and be consecutive without gaps.
- There must at most be 3072 functions defined in the consist, in total (at consist level as well as in all the vehicles).
- When the consist and vehicle properties have been decoded from the Base64 format, the total size of the resulting CSTINFO structure must be at most 65536 bytes on the wire.

> **NOTE**
>
> IEC 61375-2-3 limits the number of functions in the CSTINFO structure to 1024. This limit is increased to 3072 in the WeOS implementation.

See the sections on CSTINFO and the TTDB data model in IEC 61375-2-3 for a more detailed discussion of these rules and their meaning.

Annex A

TCN ECHO request response telegram

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               cmd               |             reserved          |0
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            challenge                          |1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |2
|                                                               |3
|                                                               |4
|                             payload                           |5
|                                                               |6
|                                                               |7
|                                                               |8
|                                                               |9
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: TCN ECHO request/response telegram layout

> **NOTE**
>
> Both the request and response telegrams use the dataset shown in figure 1 above.

The TCN ECHO request and response telegrams have ComID 170 and must be handled according to the TRDP specification (see).

The TCN ECHO request and response exchange is done by means of TRDP PD (process data) telegrams, as defined in Annex A in the IEC 61375-2-3 standard, using ComID 170 and the dataset shown in Figure 1 above. A description of each field in the dataset follows:

cmd

Type: Unsigned Int 16
Byte offset: 0

TCN ECHO command:
1 = request (sent by TCN ECHO Client)
2 = reply (sent by TCN ECHO Server)

reserved

Type: Unsigned Int 16
Byte offset: 2

Reserved. Recommended value 0.
TCN ECHO Server will copy this as-is into the reply telegrams' reserved field.

challenge

Type: Unsigned Int 32
Byte offset: 4

The TCN ECHO Server will perform an XOR operation with 0x11257731 over the received challenge and copy the calculated result into the reply telegrams' challenge field.

payload

Type: Array of Unsigned Int 8, length 32
Byte offset: 8

The TCN ECHO Server will copy the received payload as-is into the reply telegrams' payload field.

2.13.5. TRDP custom telegrams

Introduction

The *Custom Telegrams* function is a feature that allows for the creation and management of user-defined telegrams using the TRDP communications protocol. This functionality is designed to enhance the flexibility and adaptability of train communication systems, enabling the integration of custom messages and data structures with existing TRDP-based applications.

To understand this section, the reader should have a basic understanding of IP-TCN and the TRDP protocol as defined in IEC61375-2-3.

Overview

WeOS includes an implementation of the TRDP protocol, as part of IP-TCN. This implementation includes certain interfaces, most notably the ones described in IEC61375-2-3 chapter 5 (CSTINFO exchange and related functionality) and Annex E (the service interface) from the same standard.

Additionally, WeOS provides infrastructure for *custom telegrams*. There is a predefined set of *data items*, each with its own semantics and meaning, available for inclusion in *datasets*. These *datasets* can then be used as the payloads for individual *telegrams*.

Currently, only TRDP Process Data (TRDP type 'Pd', not 'Pr' nor 'Pp') telegrams are supported, both incoming (TRDP *subscribers*) and outgoing (TRDP *publishers*). The addressing, periodicity/timeout, and ComID of these telegrams is user configurable, along with the payload (which consists of one of the previously mentioned *datasets*).

Configuration

Configuration of custom telegrams in WeOS involves two steps. First, *datasets* are constructed from *data items*, and the *dataset definition file* is uploaded to the device. Then, *telegrams* are configured to use these *datasets* as their payloads.

The *data items* themselves are listed and described in detail in an external document. Consider reviewing this document carefully to understand the available options and their implications.

Selecting data items

WeOS provides a set of predefined *data items* that can be used in custom telegrams. These data items are defined in an external document. The selection of the appropriate data items depends on the specific requirements of the application.

Data items have defined sizes, which must be taken into account when constructing datasets. They also have a predefined *direction*, which limits the context in which they can be used (input, output, or both).

The following sections describe how to use the data items to create datasets and configure telegrams.

Defining datasets

First, *datasets* are constructed by choosing the appropriate *data items*, and ordering them by specifying their offsets in the dataset. Note that each *data item* has an intrinsic size, in bits, but the *dataset* is constructed as a series of bytes. Certain *data items*, smaller than one byte, can be packed into the available space within a byte, but all other items must start at the beginning of a byte.

> **NOTE**
>
> Gaps are allowed in the dataset. While the *custom telegrams* configuration does not enforce data alignment (other than what was described above), it is recommended to ensure that natural alignment is preserved, and to only place data items at offsets which are multiples of their sizes.

The following constraints apply when defining datasets:

- Each dataset must have a unique identifier.
- Up to 20 datasets may be defined in the *dataset definition file*.
- Each dataset may contain up to 1000 *data items*. This in effect also limits the number of *data items* in a telegram to 1000.
- The size of a dataset is limited to 1432 bytes. The size must be a multiple of 4 bytes.
- When adding data items that refer to device resources (e.g. ports or VLANs), the following limits apply:
  - Data from up to 32 device ports can be included across all defined datasets.
  - Data from up to 8 VLANs can be included across all defined datasets.
  - Data from up to 4 FRNTv2 rings can be included across all defined datasets. FRNTv2 is the only ring protocol currently supported.

Creating a dataset definition file

The *datasets* are defined in the form of a JSON file. This file must conform to the provided JSON schema file, and must be uploaded to the WeOS device before it can be used.

The file must be named `custom-trdp-dataset.json` and placed in the `/cfg` directory on the device. The file can be edited and uploaded to the device using any available method, as long as it is placed in the correct location. See the uploading section below for more information.

The *dataset definition file* consists of a JSON objects, each representing a single variable, which in turn refers to a *data item*:

```
{
    "type" : "...",
    "arg" : "...",
    "comment" : "...",
    "byte" : 0,
    "bit" : 0
}
```

The `"type"` field must correspond to the name of one of the predefined *data item types*. The list of available items is fixed, and available in the WeOS distribution along with descriptions of their behaviour. The `"comment"` field can be used to provide additional context or information about the data item - this is ignored by the WeOS runtime.

The position of each data item within the dataset is determined by the `"byte"` and `"bit"` attributes. These attributes specify the exact location of the item within the dataset, allowing for precise control over the layout of the data. The `"bit"` attribute is only used for packing items smaller than one byte.

Finally, the `"arg"` field contains the argument(s) provided for the data item. Whether an item needs arguments, and what they represent, depends on the specific item. This information is also available in the data item master list, included with the WeOS distribution. Commonly, this field is used to further specify which specific resource the data item refers to. For instance, a data item that

represents a statistical value (e.g. a packet counter) for a port might require the port name (e.g. “eth0”) as its argument.

Note that duplicates of the same data item may be used in the same dataset, often with different arguments. This can, for instance, be used to extract multiple statistics for different ports.

Each item also has a predefined direction (“in”, “out”, or “both”) that determines how it can be used. For most items, the direction is determined by the item’s semantics - it is, for instance, not reasonable to *set* the value of a packet counter on the device - while certain items may be used in both directions (e.g. an item that represents a CRC32 of the dataset it is contained in).

Once *data items* have been chosen and ordered, they are bundled together into *dataset* objects. A dataset object contains data items in the `"dataItems"` array, and has the following structure:

```
{
    "id" : "...",
    "dir" : "...",
    "size" : 0,
    "aliases" : ["alias1", "alias2"],
    "dataItems" : [
        {
            "type" : "...",
            "arg" : "...",
            "comment" : "...",
            "byte" : 0,
            "bit" : 0
        },
        // ... more data items ...
    ]
}
```

Note that a dataset object with an empty `"dataItems"` array is still considered valid. This can, for instance, be used to quickly construct “empty” datasets containing only zeros (by using the `"size"` attribute) for use in “empty” telegrams.

The `"id"` field is a unique identifier for the dataset, and must be specified. If several datasets are defined, they must each have a unique `"id"`. The identifier provided here will later be used to select a dataset for use in each configured *telegram*.

Note that the dataset object has a specified `"size"`, which must be equal to or larger than the sum of the sizes of its *data items*. This is to allow a gap (empty space) at the trailing end of the dataset, in which case the dataset size will be larger than the sum of the `"byte"` value, and the size, of the final item.

The `"dir"` field specifies the direction of the dataset, which can be either `"in"` or `"out"` (or even `"both"`, though the practical use for this is limited). This indicates whether the dataset is used for input or output purposes, and the value must be consistent with the direction of all individual data items contained within it. The following rules must be respected:

- If a dataset is marked as `"in"`, no contained data item may be of the type `"out"`.
- If a dataset is marked as `"out"`, no contained data item may be of the type `"in"`.
- If a dataset is marked as `"both"`, all contained data items must also be of the type `"both"`.

The `"aliases"` field specifies the alias keys, these can be used as placeholders in the arg field below to replace a string value. Instead of using the name of a vlan, example vlan1 one can use a key vlanA, and instead of a port ethX5 one might use the alias portA.

These aliases can then be configured in the cli to assign the value vlan1 to key vlanA and ethX5 to vlanA. This enable the usage of a generic dataset file, and unique configuration will be done in the cli for each device.

Datasets are then put in an array called `"datasets"`, which serves as the top-level object of the JSON file:

```
{
    "datasets" : [
        {
            "id" : "dataset1",
            "dir" : "in",
            "size" : 10,
            "aliases" : ["portA"],
            "dataItems" : [
                {
                    "type" : "...",
                    "arg" : "...",
                    "comment" : "...",
                    "byte" : 0,
                    "bit" : 0
                },
                // ... more data items in dataset1 ...
            ]
        },
        // ... more dataset objects ...
    ]
}
```

Packing data items into bytes

Data items smaller than 8 bits in size are *packed* into the available space within a byte. This is done by using the `"bit"` attribute of the item when constructing the dataset. Gaps are allowed in the byte (similarly to how they are allowed between bytes).

Items with sizes less than one byte are not allowed to span byte boundaries. For instance, an item with size 4 bits must be placed at a bit offset of 0, 1, 2, or 3 in a byte.

Items which have a size of 8 or more bits must be placed starting at byte boundaries - in this case, the `"bit"` attribute is not used (it is implicitly 0).

Other than the above, sub-byte items are added to the dataset as usual.

Consider the following configuration excerpt:

```
// ...
{
    "type" : "portAggregated",
    "arg" : "ethX1",
    "byte" : 24,
    "bit" : 0
},
{
    "type" : "portVlansAllTagged-C",
    "arg" : "ethX1",
    "byte" : 24,
    "bit" : 1
},
{
```

```
    "type" : "portMulticastAware",
    "arg" : "ethX1",
    "byte" : 24,
    "bit" : 2
},
{
    "type" : "portAutoNeg",
    "arg" : "ethX1",
    "byte" : 24,
    "bit" : 3
},
{
    "type" : "portMdixStatus-C",
    "arg" : "0",
    "byte" : 24,
    "bit" : 4
},
{
    "type" : "portLinkState",
    "arg" : "ethX1",
    "byte" : 24,
    "bit" : 6
},
{
    "type" : "portEnabled",
    "arg" : "ethX1",
    "byte" : 24,
    "bit" : 7
},
// ...
```

With the configuration above, byte 24 will look as follows, starting from the least significant bit:

- `portAggregated` (bit 0)
- `portVlanAllTagged-C` (bit 1)
- `portMulticastAware` (bit 2)
- `portAutoNeg` (bit 3)
- `portMdixStatus-C` (bit 4-5)
- `portLinkState` (bit 6)
- `portEnabled` (bit 7)

A note about data marshalling

When a dataset has been defined, it can be used as the payload of TRDP telegrams configured separately. Using the TRDP protocol, there are a few different ways that this can be done. TRDP includes a *marshalling* mechanism that allows the application to provide the TRDP layer with both the structure (in essence, type information), along with data, and then let the TRDP implementation take care of the details of formatting and transmitting said data, abstracting away such responsibilities from the application. This is termed *marshalling* and is intended to allow for TRDP to transmit data between different systems and implementations without each individual application needing to concern itself with the data representation details.

This functionality is not used in the current implementation. Instead, the implementation does the following: once the *custom telegrams datasets* have been prepared as described as above, and after the definition file has been uploaded to and validated by the device as discussed below, the implementation will use the `"size"` attribute of each dataset to create a static buffer of fixed length

which will later be used as the payload for the corresponding telegram, bypassing any TRDP marshalling completely. As far as the TRDP layer is concerned, the telegram payload consists of one array of UINT8 values, the length of which is equal to the specified dataset size. This means that the data sent through the TRDP protocol is, for all intents and purposes, a raw byte stream. Any specific requirements such as data alignment, padding, etc. must therefore be included into the dataset definition from the beginning.

When the telegram is sent on the wire, the first byte of the TRDP PD-PDU payload (see IEC61375-2-3 Figure A.11) will contain the first byte of the *custom telegram* dataset.

> **NOTE**
>
> Since TRDP marshalling is not used, it is up to the application to ensure that the *custom telegram* dataset is correctly interpreted on the other end of the TRDP transmission. The WeOS device will transmit and receive all *custom telegram* payloads as TRDP arrays of UINT8s.

Uploading the dataset definition file to the device

Once the *dataset definition file* is created, it needs to be uploaded to the device. Currently, there are two methods for performing this task:

- Using the WeOS Web interface, by navigating to "`Configure / Routing / TCN`" and using the "Upload" button.
- By manually copying the file to `/cfg/custom-trdp-dataset.json`, on the filesystem (e.g. using the external USB storage) and then using the WeOS user interface to validate it (see the `trdp-validate` command below).

In either case, the file must be located at `/cfg/custom-trdp-dataset.json` on the WeOS device, and must be *validated* before it can be used by the system. Validation entails two steps:

1. Syntax validation: The WeOS system checks the file for any structural or syntax errors or inconsistencies. This is done by validating the file against the provided JSON schema.
2. Semantic validation: The WeOS system verifies that the dataset conforms to certain logical rules (e.g. that no two data items in any dataset overlap).

This step is performed automatically by WeOS. Using the Web interface to upload the file is recommended - if the external USB storage method is used, the file will be validated automatically at a later point. In either case, only validated files can be used for configuring telegrams.

Configuring telegrams

Once a dataset definition file has been successfully uploaded and validated, the next step is to configure the telegrams that will use this dataset. Telegram configuration is performed in the WeOS CLI. At least one telegram must be configured in order for the custom telegrams functionality to be enabled; however, datasets can be defined, and the dataset definition file can be validated, independently of the datasets being used in telegrams.

Configuring a telegram involves selecting exactly one from the previously defined datasets, and then supplying additional data as needed by the TRDP layer. Each telegram has a `direction` setting, which determines whether the telegram is incoming (received by the WeOS device) or outgoing (transmitted by the WeOS device). The available direction(s) depend on the selected dataset.

Once a direction has been set, the telegram can be further configured. Depending on the direction, either the destination IP address (for outgoing telegrams) or the listening IP address (for incoming telegrams) must be specified. The former determines the address that the telegram is sent to

(which would, in nearly all cases, be a remote address) while the latter determines the address on which the device will be listening for the incoming telegrams. This must be a local address that is already owned by the device.

The following constraints apply when configuring telegrams:
- Each telegram must have a unique identifier.
- Up to 20 custom telegrams may be defined.
- The lowest periodicity for a telegram is 100 ms.
- The ComID of a telegram must be larger than 1000.

CLI configuration

The following CLI commands are available in the `ttdp` context, directly under the `config` context of the WeOS CLI. Note that this is the same context that is used for configuring IP-TCN. More information about the commands available in this context is available in the corresponding section of the documentation.

`show`

Description

List the current IP-TCN configuration, along with a list of the configured custom telegrams.

Using `show custom-trdp` will show only the configured custom telegrams.

`[no] custom-trdp NAME`

Description

Enter a *custom telegram* configuration sub-context, or create a new one.

Options

Option	Description
`NAME`	The identifier of an existing, or new, custom telegram object **NOTE** `NAME` must fulfil the following criteria: • Each name must be unique (it identifies the telegram) • Only upper- and lowercase letters and numbers are allowed • The name must not be empty, and the maximum allowed length is 31
`no`	Delete all configuration for the telegram `NAME`.

`[no] trdp-validate`

Description

Validate and verify the dataset definition data file (`/cfg/custom-trdp-dataset.json`).

This command is used to manually request validation of the dataset file.

Attempt to validate a preexisting dataset definition data file against the internal JSON schema and the additional semantic rules. If successful, calculate a SHA256 hash over the contents of

continues on next page

continued

`[no] trdp-validate`

the file, and optionally store this hash on the device for future verification by the *custom telegrams* implementation.

Once the hash has been calculated, the user will be asked whether to store it.

The *custom telegrams* feature will only be available if this validation is successful.

If the hash is updated for a new file, the configured custom-trdp instances will be removed to ensure no faulty configuration persist.

Options

Option	Description
`no`	Remove any stored SHA256 hash of the dataset definition data file. Remove all configured custom-trdp instances.

Telegram context CLI commands

Once inside the sub-context that corresponds to a specific telegram instance, the following commands become available:

`show`

Description

Show the current configuration of the custom telegram object.

`[no] enable`

Description

Enable or disable the custom telegram.

This command can be used to enable or disable the current telegram, while retaining any configuration. Disabled telegrams are neither sent nor received.

> **NOTE**
>
> Disabling all telegrams effectively disables the *custom telegrams* feature.

`[no] dataset DSID`

Description

Link this telegram to a previously defined dataset.

Options

Option	Description
`DSID`	The identifier (`"id"`) of an existing dataset object.

continues on next page

continued

`[no] dataset DSID`

Option	Description
`no`	Remove the link to the dataset. This prevents further configuration of the telegram until a new dataset is linked.

`dir <in|out>`

Description

Set the direction of the telegram.

Options

Option	Description
`in`	Set the direction to incoming. This makes the `listen-addr` setting mandatory and forbids the `dst-addr` setting.
`out`	Set the direction to outgoing. This makes the `dst-addr` setting mandatory and forbids the `listen-addr` setting.

`[no] comid NUM`

Description

Set the TRDP ComID for the telegram.

Options

Option	Description
`NUM`	The TRDP ComID to assign to the telegram. This is an unsigned 32-bit integer in decimal format and must be larger than `1000`.
`no`	Remove the TRDP ComID assignment from the telegram. This is equivalent to setting it to `0`, which is an illegal value.

`[no] dst-addr ADDR`

Description

Set the destination address for the telegram (the address the telegram will be sent to).

This setting is only relevant for outgoing telegrams.

Options

Option	Description
`ADDR`	The destination address to assign to the telegram. This must be a valid IP address in the format `X.X.X.X`, where `X` is a decimal number between `0` and `255`.
`no`	Remove the destination address assignment from the telegram.

`[no] listen-addr ADDR`

Description

Set the listen address for the telegram (the local address on which to listen for the telegram).

This setting is only relevant for incoming telegrams.

Options

Option	Description
`ADDR`	The listen address to assign to the telegram. This must be a valid IP address in the format `X.X.X.X`, where `X` is a decimal number between `0` and `255`.
`no`	Remove the listen address assignment from the telegram.

`[no] period TIME`

Description

Set the periodicity for the telegram.

This setting is only relevant for outgoing telegrams.

Options

Option	Description
`TIME`	The periodicity in milliseconds. This determines the nominal value of the telegram's transmission interval. Must be a value between 100 and 3600000.
`no`	Remove the periodicity assignment from the telegram.

> **NOTE**
>
> The real-world wall clock periodicity of the telegram may vary depending on network conditions and other factors.

`[no] timeout TIME`

Description

Set the timeout for the telegram. See below for details.

This setting is only relevant for incoming telegrams.

Options

Option	Description
`TIME`	The timeout in milliseconds. This determines the maximum time to wait for a response before considering the transmission failed. Must be a value between 100 and 3600000.

continues on next page

continued

`[no] timeout TIME`

Option	Description
`no`	Remove the timeout assignment from the telegram. This is equivalent to setting the timeout to infinite (which disables it).

> **NOTE**
>
> It is recommended to set the timeout 3 times longer than the expected periodicity of the incoming telegram.

`[no] transmit <always|active-ecsp>`

Description

Set the transmission option of the telegram.

Options

Option	Description
`always`	This setting makes the device to always send the custom telegram no matter if the device is an ECSP or not.
`active-ecsp`	This setting makes the device to only transmit the custom telegram when it is an active ECSP.
`no`	Sets the transmit option to always.

> **NOTE**
>
> The transmit setting is only available for outgoing custom telegrams when operational mode is set to `full`.

Timeout behaviour of incoming telegrams

As shown above, every incoming telegram (with `dir` set to `in`) can have a timeout value configured. Currently, WeOS only uses this value to determine when (or whether) to write a message, indicating that the telegram has timed out, to the system log. If the timeout is disabled no such message will be written. Thus, in other words, not setting the timeout effectively means that the timeout is infinite.

Note that this functionality is not the same as the different *timeout behaviour* settings provided by TRDP and certain TRDP implementations. This is because the *custom telegrams* functionality in WeOS is always event driven and edge-triggered, meaning that incoming telegrams are consumed and processed if and when they are received. In other words, if a telegram is no longer being received, no action will be performed based on the data it contained.

Using custom telegrams

> **NOTE**
>
> Configuring a large number of telegrams, or using low transmission periods, may impact system performance. It is important to monitor the system closely in such cases.

Once datasets and telegrams have been configured, and the configuration has been committed, transmission and reception of the telegrams is managed automatically. No further user action is required.

Certain incoming telegrams may contain *data items* that will change the WeOS state, and this state can then be monitored or queried by users using the normal WeOS mechanisms. This includes, for instance, the `date` command in the top level context of the WeOS CLI, which can be used to display the current date and time - a value which can in turn be set using the **syncDateTime-C** data item.

Outgoing telegrams can not be monitored on the WeOS device directly, with the exception of indirect means such as port packet transmission statistics. To monitor the outgoing data in greater detail, either software tools on the receiving device, or network monitoring equipment, must be used.

Note however that the TRDP layer conforms to the statistics layer described in IEC61375-2-3 Annex D. The interfaces specified therein can be used to extract statistics about the TRDP publishers and subscribers that have been set up on the device, which includes those set up by the *custom telegrams* function.

Some information, including a list of the configured telegrams, can be shown using the `show custom-trdp` command in the **Admin context** of the WeOS CLI.

2.13.6. “Custom telegram” data items list

> **NOTE**
>
> While all data items are available in all operational modes, some data items are only *active* (meaning that the relevant input data is processed, and the output data is populated) in certain modes.

Direction incoming

Input data-items active in all operational modes

lifesign8

- Size: 8 bits
- Format: unsigned integer
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: Lifesign = 0
- Conditional check:
  Life-sign changed (differs from the previously received telegram)

lifesign16

- Size: 16 bits
- Format: unsigned integer in big endian format
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: Lifesign = 0
- Conditional check:
  Life-sign changed (differs from the previously received telegram)

lifesign32

- Size: 32 bits
- Format: unsigned integer in big endian format
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: Lifesign = 0
- Conditional check:
  Life-sign changed (differs from the previously received telegram)

crc32-fcs

- Size: 32 bits
- Format: unsigned integer in little endian format (IEC 61375-2-3 appendix A.3)
- Arguments: start:integer, length:integer
- Validation: Number of arguments == 2; 1st argument is an integer; 2nd argument is an integer
- Initialisation:
  Start = 1st argument
  Length = 2nd argument
- Conditional check:
  Checksum match the TRDP FCS CRC (CRC32 according to IEEE802.3) calculated over the configured section of the incoming dataset (Start, Length)

syncDateTime-C

- Size: 48 bits
- Format: Array of 6 unsigned 8-bits integers
- Coding:
  Byte 0: Year- 2000 [0-255] (0 = 2000)
  Byte 1: Month [1-12] (1 = January)
  Byte 2: Day of month [1-31]
  Byte 3: Hour [0-23]
  Byte 4: Minute [0-59]
  Byte 5: Second [0-59]
- Arguments: None
- Validation: Number of arguments == 0
- Action:
  If the input date/time is valid (within the ranges specified in the coding) and deviates more than 3 seconds to the system time then set the Real-Time Clock to the input date/time.

bypassCtrl

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = No action (keep old state)
  1 = Disable bypass relays (Connect ETB switch to ETB lines)
  2 = Enable bypass relays (Decouple ETB switch from ETB lines)
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: OldCmd = 0
- Action:
  NewCmd = value of data-field
  If NewCmd != OldCmd and NewCmd is 1 or 2 then request control of bypass relays according to NewCmd
  If NewCmd is 0, 1 or 2 then OldCmd = NewCmd

Input data-items active in 'full' operational mode

These data-items are available in all operational modes, but input data is only processed in 'full' operational mode (in all other modes the input data is ignored and no actions are performed).

etbInhibitCtrl

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = No action (keep old state)
  1 = No inhibit request
  2 = Inhibit request
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: OldCmd = 0
- Action:
  NewCmd = value of data-field
  If NewCmd != OldCmd and NewCmd is 1 or 2 then request control of bypass relays according to

NewCmd
If NewCmd is 0, 1 or 2 then OldCmd = NewCmd

Direction outgoing

Output data-items active in all operational modes

lifesign8

- Size: 8 bits
- Format: unsigned integer
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: Lifesign = 0
- Prepare:
  Increase lifesign by 1
  Data = Lifesign

lifesign16

- Size: 16 bits
- Format: unsigned integer in big endian format
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: Lifesign = 0
- Prepare:
  Increase lifesign by 1
  Data = Lifesign

lifesign32

- Size: 32 bits
- Format: unsigned integer in big endian format
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation: Lifesign = 0
- Prepare:
  Increase lifesign by 1
  Data = Lifesign

value1

- Size: 1 bit
- Format: unsigned integer
- Arguments: value:integer
- Validation: Number of arguments == 1; 1st argument is an integer [0, 1]
- Initialisation: Value = 1st argument
- Prepare: Data = Value

value2

- Size: 2 bits
- Format: unsigned integer
- Arguments: value:integer
- Validation: Number of arguments == 1; 1st argument is an integer [0, 3]
- Initialisation: Value = 1st argument

- Prepare: Data = Value

value4

- Size: 4 bits
- Format: unsigned integer
- Arguments: value:integer
- Validation: Number of arguments == 1; 1st argument is an integer [0, 15]
- Initialisation: Value = 1st argument
- Prepare: Data = Value

value8

- Size: 8 bits
- Format: unsigned integer
- Arguments: value:integer
- Validation: Number of arguments == 1; 1st argument is an integer [0, 255]
- Initialisation: Value = 1st argument
- Prepare: Data = Value

value16

- Size: 16 bits
- Format: unsigned integer in big endian format
- Arguments: value:integer
- Validation: Number of arguments == 1; 1st argument is an integer [0, 65535]
- Initialisation: Value = 1st argument
- Prepare: Data = Value

value32

- Size: 32 bits
- Format: unsigned integer in big endian format
- Arguments: value:integer
- Validation: Number of arguments == 1; 1st argument is an integer [0, 4294967295]
- Initialisation: Value = 1st argument
- Prepare: Data = Value

crc32-fcs

- Size: 32 bits
- Format: unsigned integer in little endian format (IEC 61375-2-3 appendix A.3)
- Arguments: start:integer, length:integer
- Validation: Number of arguments == 2; 1st argument is an integer; 2nd argument is an integer
- Initialisation:
  Start = 1st argument
  Length = 2nd argument
- Prepare:
  Data = TRDP FCS CRC (CRC32 according to IEEE802.3) calculated over the configured section of the outgoing dataset (Start, Length)

firmwareVersion

- Size: 32 bits
- Format: Array of 4 unsigned 8-bits integers

- Coding:
  Version is encoded as 4 unsigned integers [0, 255]: A.B.C.D
  Byte 0: A
  Byte 1: B
  Byte 2: C
  Byte 3: D
- Arguments: None
- Validation: Number of arguments == 0
- Initialisation:
  Version = Value of DISTRIB_RELEASE in /etc/lsb-release
- Prepare: Data = Version

cpuLoad

- Size: 8 bits
- Format: unsigned integer
- Coding: 1 = 1%
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = System load

devTemp-C

- Size: 8 bits
- Format: signed integer
- Coding:
  0: Temperature is unknown/invalid 0..127: Temperature 0 to 127 degree Celsius 128..255: Temperature −1 to −128 degree Celsius
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = Device temperature

oprTime

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 second
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = System uptime

oprTime-C

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 0.1 hour
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = System uptime

dateTime

- Size: 48 bits
- Format: Array of 6 unsigned 8-bits integers

- Coding:
  Byte 0: Year- 1900 [0-255] (0 = 1900)
  Byte 1: Month [0-11] (0 = January)
  Byte 2: Day of month [1-31]
  Byte 3: Hour [0-23]
  Byte 4: Minute [0-59]
  Byte 5: Second [0-59]
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = Real-Time Clock time

dateTime-C

- Size: 48 bits
- Format: Array of 6 unsigned 8-bits integers
- Coding:
  Byte 0: Year- 2000 [0-255] (0 = 2000)
  Byte 1: Month [1-12] (1 = January)
  Byte 2: Day of month [1-31]
  Byte 3: Hour [0-23]
  Byte 4: Minute [0-59]
  Byte 5: Second [0-59]
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = Real-Time Clock time

portEnabled

- Size: 1 bit
- Format: unsigned integer
- Coding:
  0 = Disabled
  1 = Enabled
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Enable state of configured port

portLinkState

- Size: 1 bit
- Format: unsigned integer
- Coding:
  0 = Down
  1 = Up
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Link state of configured port

portAutoNeg

- Size: 1 bit
- Format: unsigned integer

- Coding:
  0 = Disabled
  1 = Enabled
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Auto negotiation state of configured port

portMdixSetting

- Size: 2 bits
- Format: unsigned integer
- Coding:
  0 = Invalid, N/A
  1 = Normal (mdi)
  2 = Crossover (mdi-x)
  3 = Auto
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = MDI-X setting of configured port

portMdixStatus

- Size: 2 bits
- Format: unsigned integer
- Coding:
  0 = Invalid, N/A
  1 = Normal (mdi)
  2 = Crossover (mdi-x)
  3 = Auto
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = MDI-X status of configured port

portMdixStatus-C

- Size: 2 bits
- Format: unsigned integer
- Coding:
  0 = Crossover (mdi-x)
  1 = Normal (mdi)
  2 = Auto
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = MDI-X status of configured port

portAggregated

- Size: 1 bit
- Format: unsigned integer

- Coding:
  0 = Port not assigned to aggregate
  1 = Port assigned to aggregate
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Enable state of configured port

portRxError

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of error packets inbound of configured port

portTxError

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of error packets outbound of configured port

ringStatus-C

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Ring disabled
  1 = Ring master node
  2 = Ring slave node
  3 = Ring broken
  4 = Other states
- Arguments: ring identity:integer
- Validation: Number of arguments == 1; 1st argument is an integer
- Initialisation: FRNTv2 ring identity = 1st argument
- Prepare: Data = Status of configured FRNTv2 ring

ringPortStatus-C

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Unknown
  1 = Disabled
  2 = Blocking
  3 = Listening
  4 = Learning
  5 = Forwarding

- Arguments: ring identity:integer, ring port index:integer
- Validation: Number of arguments == 2; 1st argument is an integer; 2nd argument is an integer [1, 2]
- Initialisation:
  FRNTv2 ring identity = 1st argument
  Ring port index = 2nd argument
- Prepare: Data = Status of configured ring port

ringPortNumber-C

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Ring port unknown (not in list of port names)
  1..N = Ring port equals the Nth port in the list of port names
- Arguments: ring identity:integer, ring port index:integer, [port name: string, ... , port name:string]
- Validation: Number of arguments >= 2; 1st argument is an integer; 2nd argument is an integer [1, 2]; 3rd to Nth argument are strings
- Initialisation:
  FRNTv2 ring identity = 1st argument
  Ring port index = 2nd argument
  Port name 1 = 3rd argument
  Port name 2 = 4th argument
  Port name 3 = 5th argument

:

- Prepare: Data = Number of port (1..N) that match the name of the configured ring port (or 0 if there is no match)

igmpEnabled

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Disabled
  1 = Enabled
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = IGMP snooping is allowed and enabled on at least one vlan

numberOfPorts

- Size: 8 bits
- Format: unsigned integer
- Arguments: None
- Validation: Number of arguments == 0
- Prepare: Data = Number of configured ports

portMulticastAware

- Size: 1 bit
- Format: unsigned integer

- Coding:
  0 = No VLAN assigned to the port has multicast snooping enabled
  1 = At least one VLAN assigned to the port has multicast snooping enabled
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Prepare: Data = Port is assigned to at least one VLAN with multicast snooping enabled

portFlooding

- Size: 1 bit
- Format: unsigned integer
- Coding:
  0 = Disabled (unknown multicasts are blocked)
  1 = Enabled (unknown multicasts from other ports in the same vlan is flooded on this port)
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Prepare: Data = Flooding of unknown multicast

portVlanAllTagged-C

- Size: 1 bit
- Format: unsigned integer
- Coding:
  0 = This port has an untagged vlan (or it is not in any vlan)
  1 = All vlans on port is tagged
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Prepare: Data = Port is tagged on all vlans it is assigned to

portVlanId-C

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding:
  If the port has an untagged vlan the identity of that vlan is given as vlan number 1 and the identities of its tagged vlans are given (in ascending identity order) as vlan number 2 and upward
  If the port has no untagged vlan the identities of its tagged vlans are given (in ascending identity order) as vlan number 1 and upward
- Arguments: port name:string, vlan number:string
- Initialisation:
  Port name = 1st argument
  Vlan number = 2nd argument
- Validation: Number of arguments == 2; 1st argument is a string; 2nd argument is an integer
- Prepare: Data = Identity of the requested vlan (see coding)

portModule-C

- Size: 8 bits
- Format: unsigned integer in big endian format
- Coding:
  0 = Invalid
  1 = 1000M, full duplex
  2 = 1000M, half duplex
  3 = 100M, full duplex

4 = 100M, half duplex
5 = 10M, full duplex
6 = 10M, half duplex
7 = Adaptive
8 = Disabled

- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Speed and Duplex mode of configured port

portRx

- Size: 64 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of packets received on configured port

portTx

- Size: 64 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of packets transmitted from configured port

portRxFiltered

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of filtered packets inbound on configured port

portTxFiltered

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of filtered packets outbound on configured port

portRxDropped

- Size: 32 bits
- Format: unsigned integer in big endian format

- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of inbound packets dropped

portRxCollisions

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of inbound packets collisions

portTxCollisions

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of outbound packets collisions

portSpeedMode

- Size: 4 bits
- Format: unsigned integer
- Coding:
  0 = Invalid or N/A
  1 = 10Mbit/s
  2 = 100Mbit/s
  3 = 1000Mbit/s
  15 = Unknown
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Speed of configured port

portSpeedMode-C

- Size: 4 bits
- Format: unsigned integer
- Coding:
  0 = 10Mbit/s (Invalid, Unknown or N/A)
  1 = 100Mbit/s
  2 = 1000Mbit/s
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Speed of configured port

portDuplexMode

- Size: 4 bits
- Format: unsigned integer
- Coding:
  0 = Half duplex (invalid or N/A)
  1 = Full duplex
  15 = Unknown
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Duplex mode of configured port

portDuplexMode-C

- Size: 4 bits
- Format: unsigned integer
- Coding:
  0 = Half duplex (invalid, unknown or N/A)
  1 = Full duplex
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Duplex mode of configured port

portRxLimit

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 256kbit/s
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Rate limit / Ingress bandwidth limit on configured port

portRxLimitBcast

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 256kbit/s
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Rate limit / Ingress bandwidth limit for broadcast on configured port

portTxLimit

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 256kbit/s
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Traffic shaping / Outbound bandwidth limit on configured port

portDefaultVid

- Size: 16 bits
- Format: unsigned integer in big endian format
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Default vlan identity of configured port

portPriority

- Size: 8 bits
- Format: unsigned integer
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Priority of configured port

portRxMcast

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of multicast packets received by configured port

portTxMcast

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of multicast packets transmitted by configured port

portRxBcast

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of broadcast packets received by configured port

portTxBcast

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument

- Prepare: Data = Number of broadcast packets transmitted by configured port

portRxUcast

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of unicast packets received by configured port

portTxUcast

- Size: 32 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1 packet
- Arguments: port name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Port name = 1st argument
- Prepare: Data = Number of unicast packets transmitted by configured port

vlanIdentity

- Size: 16 bits
- Format: unsigned integer in big endian format
- Arguments: vlan name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Vlan name = 1st argument
- Prepare: Data = Identity of configured vlan

vlanAgingTime

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1s
- Arguments: vlan name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Vlan name = 1st argument
- Prepare: Data = Aging time of configured vlan

vlanMcastSnooping

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Disabled for vlan, Disabled globally
  1 = Disabled for vlan, Enabled globally
  2 = Enabled for vlan, Disabled globally
  3 = Enabled for vlan, Enabled globally
- Arguments: vlan name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Vlan name = 1st argument
- Prepare: Data = Status of multicast snooping of configured vlan

vlanQueryEnabled

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Disabled
  1 = Enabled
- Arguments: vlan name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Vlan name = 1st argument
- Prepare: Data = Status of multicast querier of configured vlan

vlanQueryInterval

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1s
- Arguments: vlan name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Vlan name = 1st argument
- Prepare: Data = Multicast querier interval of configured vlan

vlanRouterTimeout

- Size: 16 bits
- Format: unsigned integer in big endian format
- Coding: 1 = 1s
- Arguments: vlan name:string
- Validation: Number of arguments == 1; 1st argument is a string
- Initialisation: Vlan name = 1st argument
- Prepare: Data = Router timeout

vlanPortTagged

- Size: 1 bit
- Format: unsigned integer
- Coding:
  0 = Port not tagged on vlan
  1 = Port tagged on vlan
- Arguments: vlan name:string, port name:string
- Validation: Number of arguments == 2; 1st argument is a string; ; 2nd argument is a string
- Initialisation:
  Vlan name = 1st argument
  Port name = 2nd argument
- Prepare: Data = Port tagged on vlan

vlanPortUntagged

- Size: 1 bit
- Format: unsigned integer
- Coding:
  0 = Port not untagged on vlan
  1 = Port untagged on vlan
- Arguments: vlan name:string, port name:string
- Validation: Number of arguments == 2; 1st argument is a string; ; 2nd argument is a string

- Initialisation:
  Vlan name = 1st argument
  Port name = 2nd argument
- Prepare: Data = Port untagged on vlan

vlanPortMember

- Size: 1 bit
- Format: unsigned integer
- Coding:
  0 = Port neither tagged nor untagged on vlan
  1 = Port tagged or untagged on vlan
- Arguments: vlan name:string, port name:string
- Validation: Number of arguments == 2; 1st argument is a string; 2nd argument is a string
- Initialisation:
  Vlan name = 1st argument
  Port name = 2nd argument
- Prepare: Data = Port tagged or untagged on vlan

bypassState

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Bypass disabled (ETB switch connected to ETB lines)
  1 = Bypass enabled (ETB switch decoupled from ETB lines)
- Arguments: None
- Prepare: Data = State of bypass function

Output data-items active in 'etbn' and 'full' operational mode

These data-items are available in all operational modes but output data is only prepared in 'etbn' and 'full' operational mode (in all other operational modes the output data is set to zero).

etbnState

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = ETBN not operational
  1 = ETBN in operation
- Arguments: None
- Prepare: Data = ETBN operational state

etbnInaugState

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Init
  1 = Not inaugurated
  2 = Inaugurated
  3 = Ready for inauguration
- Arguments: None
- Prepare: Data = ETBN inauguration state

etbnPosition

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Unknown
  1 = Single node
  2 = Middle node
  3 = End node in TCN direction 1
  4 = End node in TCN direction 2
- Arguments: None
- Prepare: Data = Position of the ETBN

etbnRole

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Undefined
  1 = Master
  2 = Backup
  3 = Not redundant
- Arguments: None
- Prepare: Data = ETBN node role

etbLineState

- Size: 8 bits
- Format: unsigned integer
- Coding:
  Bit 0: Line A, ETBN direction 1 (0 = Not trusted, 1 = Trusted)
  Bit 1: Line B, ETBN direction 1
  Bit 2: Line C, ETBN direction 1
  Bit 3: Line D, ETBN direction 1
  Bit 4: Line A, ETBN direction 2
  Bit 5: Line B, ETBN direction 2
  Bit 6: Line C, ETBN direction 2
  Bit 7: Line D, ETBN direction 2
- Arguments: None
- Prepare: Data = Indication of ETB line status

etbTopoCnt

- Size: 32 bits
- Format: unsigned integer in big endian format
- Arguments: None
- Prepare: Data = ETB topography counter

etbInhibit

- Size: 8 bits
- Format: unsigned integer
- Coding:
  1 = Inhibit not requested on ETB
  2 = Inhibit set on local ETBN

3 = Inhibit set on remote ETBN
4 = Inhibit set on local and remote ETBN
- Arguments: None
- Prepare: Data = Inauguration inhibit indication

Output data-items active in 'full' operational mode

These data-items are available in all operational modes but output data is only prepared in 'full' operational mode (in all other operational modes the output data is set to zero).

cstCnt

- Size: 8 bits
- Format: unsigned integer
- Coding: Number of consists in train [1, 63]
- Arguments: None
- Prepare: Data = Number of consists in train

cstOrient

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Unknown
  1 = Direct (same as train reference direction)
  2 = Inverse to train reference direction
- Arguments: None
- Prepare: Data = Consist orientation

etbLength

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Not lengthening
  1 = Lengthening detected
- Arguments: None
- Prepare: Data = Indication of train lengthening in case train inauguration is inhibited

etbShort

- Size: 8 bits
- Format: unsigned integer
- Coding:
  0 = Not shorting
  1 = Shorting detected
- Arguments: None
- Prepare: Data = Indication of train shortening in case train inauguration is inhibited

opTrnTopoCnt

- Size: 32 bits
- Format: unsigned integer in big endian format
- Arguments: None
- Prepare: Data = Operational train topography counter

ownTrnCstNo

- Size: 8 bits
- Format: unsigned integer
- Coding: Position of own consist in train [1, 63]
- Arguments: None
- Prepare: Data = Position of own consist in train

2.14. Policy Filtering

2.14.1. Policy Traffic Filtering

Policy traffic filtering introduces the ability to filter traffic across multiple layers of the frame or packet, and offload the filtering to the hardware.

> **NOTE**
>
> As of this release, policy traffic filtering is only supported for the Redfox 7000/5000 and Lynx 5000-series.

Introduction

Usage of policy traffic filtering allows for the possibility to offload filtering of packets directly in hardware.

When using the firewall subsystem, it requires packets to be forwarded by the CPU of the switch to perform the filtering of network traffic.

However, these operations will come at a cost in routing performance. Since packets are filtered in hardware, filters need to be configured per direction, ingress or egress, and per port.

This entails that policy traffic filtering is not capable of monitoring and tracking active network connections (stateful) and is therefore stateless. If returning traffic should be accepted, filters need to be configured in both ingress and egress directions. An additional benefit with policy traffic filtering is the ability of matching across multiple layers of the packet. *E.g.* matching on a packets MAC address, something not possible with the firewall subsystem.

Configuration

Policies are configured and enabled from the policy configuration context in the CLI.

```
example:/#> configure
example:/config/#> policy
example:/config/policy/#>
```

To delete all policy related configurations execute the following command.

```
example:/#> configure
example:/config/#> no policy
```

`[no] enable`

Description

Enable or disable policies.

> **NOTE**
>
> Configured policies will not be deleted upon *no enable*, rather they are saved but not applied.

continues on next page

continued

`[no] enable`

Options

Option	Description
`no`	Disable the policies.

`show`

Description

Show the current policy settings

Ingress and Egress Policy Specific Settings

The following context creates and manages the groups of filters for ingress or egress policies. The name or id is used to distinguish between different groups of ingress or egress policies. The following options can both be found when configuring an ingress policy or egress policy.

```
example:/config/policy/#> <ingress|egress>-policy <NAME|id>
```

To delete all ingress- or egress-policy configuration for a specific id or name execute the following command.

```
example:/config/policy/#> no <ingress|egress>-policy <NAME|id>
```

`[no] description <STRING>`

Description

Free form description of the current policy group.

A valid string contains ASCII characters in the range of [32-126]. Maximum of 128 characters allowed.

Options

Option	Description
`no`	Remove the saved description.

`[no] name <STRING>`

Description

Set name of the current policy group.

A valid string contains alphabetic and numeric characters. Maximum of 16 characters allowed.

Options

Option	Description
`no`	Remove the saved name.

`[no] filter <id>`

Description

Enter the filter configuration context.

Create a filter with an id that is used as a ruleset for ingress or egress traffic control. Valid filter id range [1-50].

Options

Option	Description
`no`	Remove saved filter of the given id.

`show`

Description

Show the current settings of the ingress or egress policy.

Filter Specific Settings

The following context creates and manages the individual filters for ingress or egress policies, depending on the current context. The filter options are identical in both the ingress-policy context and the egress-policy context. Valid filter id range [1-50].

```
example:/config/policy/ingress-policy-1/#> filter <id>
```

To delete the filter of a given id, execute the following command.

```
example:/config/policy/ingress-policy-1/#> no filter <id>
```

`[no] match [MATCH]`

Description

Set match rule(s) specifying the policy when the filter action should be applied. It is possible to match on several match conditions within the same filter. However, if two match statements are entered using the same match condition. The older match statement will be overwritten.

Currently the following are supported options when using the match statement.

Options

Option	Description
`no`	Remove a specific match condition.
`smac`	Match frames on the source MAC address field in the frame header. A wildcard (*) can be used to match on specific blocks of the MAC address.
`dmac`	Match frames on the destination MAC address field in the frame header. A wildcard (*) can be used to match on specific blocks of the MAC address.
`saddr`	Match frames on the source IP address field in the IP header. A CIDR (/) can be used to match on a range of IP addresses.

continues on next page

continued

`[no] match [MATCH]`

Option	Description
`daddr`	Match frames on the destination IP address field in the IP header. A CIDR (/) can be used to match a on range of IP addresses.
`ethtype`	Match incoming traffic by its ethertype. Do note that using a custom ethertype field which is not 0x0800 (IPv4) removes the possibility of matching information found in the IP header.
`proto`	Match incoming traffic by its IP protocol. This can be a specified the protocol number or protocol name defined by IANA.
`sport`	Match incoming traffic by its source IP port. To match on the source IP port, the IP protocol has to be set to TCP or UDP.
`dport`	Match incoming traffic by its destination IP port. To match on the destination IP port, the IP protocol has to be set to TCP or UDP.
`dscp`	Match incoming traffic by its DSCP value. This can be a decimal value or DSCP name defined by IANA.

`pos [POSITION]`

Description

Position of the current filter in relation to other filter of the same ingress or egress policy. A lower position entails a higher priority. Filters are checked in the order of their respective positions, in ascending order. The position can not be the same as another filter. Valid positions include [1-1000].

`action <accept|drop|trap>`

Description

Specify the action to be performed when a filter match is found. A newly created filter will default to drop if nothing else is specified.

Options

Option	Description
`accept`	Let the matching packet pass through the filter.
`drop`	Let the matching packet be dropped and not pass the filter.
`trap`	Let the matching packet be trapped and sent to the CPU. Only recommended when certain packets need to be sent through the CPU. Only supported in ingress direction.

Port Specific Settings

The following context is used to configure and manage traffic filter policies for a specific port. In the port context, ingress and egress policies are applied to the given port.

```
example:/config/policy/#> port <PORT>
```

`ingress-chain`

Description

Enter the ingress chain context.

Configure and manage ingress policies for the selected port.

`egress-chain`

Description

Enter the egress chain context.

Configure and manage egress policies for the selected port.

Port Ingress and Egress Chain Specific Settings

The following context manages the ingress or egress chains of the specific port. Previously created ingress or egress policies are applied in the following context.

> **NOTE**
>
> Ingress polices can not be applied on a egress chain, and *vice versa*. If an identical filter should be placed in both ingress and egress direction. Two identical filters need to be created, one ingress filter and one egress filter.

To manage and configure ingress chains enter:

```
example:/config/policy/port-eth1/#> ingress-chain
```

To manage and configure egress chains enter:

```
example:/config/policy/port-eth1/#> egress-chain
```

`[no] ingress <NAME|id> [pos POSITION] [vid VID]`

Description

Apply a previously created ingress policy to current port.

> **NOTE**
>
> This command is only available in the ingress-chain context.

Options

Option	Description
`NAME|id`	The name or id of the ingress policy.
`pos`	The position of the ingress policy for the current port. A lower position entails a higher priority. Policies are checked and evaluated in ascending order.

continues on next page

continued

`[no] ingress <NAME|id> [pos POSITION] [vid VID]`

Option	Description
`vid`	The optional vid argument allows control of which VLANs the selected policy should act on. If no vid is specified the policy will for all VLANs.

`[no] egress <NAME|id> [pos POSITION] [vid VID]`

Description

Apply a previously created egress policy to current port.

> NOTE
>
> This command is only available in the egress-chain context.

Options

Option	Description
`NAME|id`	The name or id of the egress policy.
`pos`	The position of the egress policy for the current port. A lower position entails a higher priority. Policies are checked and evaluated in ascending order.
`vid`	The optional vid argument allows control of which VLANs the selected policy should act on. If no vid is specified the policy will for all VLANs.

`default-action <accept|drop>`

Description

Selection of action to be applied if there is no policies matching for a packet. The default-action is per default set to drop.

Options

Option	Description
`accept`	If no policies match the incoming or outgoing packet, the packet will be passed.
`drop`	If no policies match the incoming or outgoing packet, the packet will be dropped.

Statistics

Policies which have been activated on a port will generate statistics which can be viewed. The statistics displays the number of packets which have matched upon each configured rule, per port.

```
example:/#> show policy
╒ Ingress-policies counters ═══════════════════════════════════════════════
│PORT      ACTION        MATCH   VALUE                PACKETS
│eth1      drop          proto:  tcp                  24
│                        dport:  22
│          Default-Drop                               58
```

```
└──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┘

┌ Egress-policies counters ═══════════════════════════════════════════════════════════════════════════════════════════
│PORT      ACTION        MATCH   VALUE                 PACKETS
│eth1      accept        saddr:  192.168.10.0/24       78
│          Default-Drop                                34
└──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
```

2.15. Configuration Integrity

2.15.1. Boot Configuration Integrity

Boot Configuration Integrity is an automatic integrity verification of boot configuration (CLI under '/boot' context menu or in the web interface under 'Maintenance->Boot Settings'). This feature is enabled on platforms using the barebox bootloader.

Overview

The main idea behind this feature is to have both bootloader and WeOS check validity of the bootloader environment variables. And in the case of unauthorized modifications warn about it. The bootloader will print a warning on the serial console, and WeOS will log the integrity violation on every boot in the audit log and trigger an alarm. The Power On LED will be set by WeOS to red implicitly due to an alarm being set. Changes to boot configuration are prohibited when a violation is detected.

Note: There are situations when upgrading and downgrading that can result in a Boot Configuration Integrity violation. If that occurs, please use steps below to clear the violation alarm.

Clearing integrity violation

To clear the integrity violation use `boot-integrity restore` command under '/boot/loader' configuration context. Or in the web interface under 'Maintenance->Boot Settings'. Restoring will log to audit log and the alarm will be cleared on reboot. Make sure settings are checked to be correct before doing a restore.

2.15.2. Configuration-integrity

Configuration-integrity is an automatic integrity verification of configuration files.

Overview

The main idea behind this feature is to protect configuration files from malicious modification in order to ensure the integrity of the system.

Configuration-integrity uses secure storage in order to store hashes of configuration files. This is needed in order to verify integrity of configuration files before application or modification.

The following configuration files are protected by the feature:

Static configuration files:
- factory-config.cfg
- safe-config.cfg
- no-config.cfg

Running configuration file:
- running-config.cfg

Startup configuration file:
- startup-config.cfg

Static configuration files are handled in a simplified way, since these, even if modified, reset to their original state during boot. Their hashes will be added to secure storage once, during first boot, and the only occasion an update might be needed is when installing a branding package. These static files have implicit trust.

Branding package might have custom factory-config, safe-config and no-config files, and these would replace their standard counterparts. Since the default policy is to trust the branding package, no integrity verification is performed, only an update.

running-config's integrity is updated upon configuration change through CLI or Web, or upon copying other config files into it.

CLI and Web changes simply update the integrity of the running-config, while copy command will first verify the source file that is being copied into the running-config. If integrity verification of the source succeeds, the copy is performed and the integrity information will be updated.

startup-config's integrity information is updated either upon copy command, or when applying settings in the Web. Upon 'copy', the source file's integrity will always be verified and an appropriate audit log entry will appear in the audit log. When applying settings through the Web, the integrity of the startup-config will be updated to reflect recent changes.

Backup/restore

WeOS can generate unlimited number of backups, but only 5 latest fingerprints will be stored in the secure storage, while older ones will be overwritten.

Restoring of any backup is possible but an alert will be issued in the audit log, unless the configuration that is being restored is one of the five latest backups generated by this device.

Default policy

The default policy is to always allow copying files and restoring backups even if their integrity isn't intact, but an appropriate audit log entry will be issued to signal a potential threat.

2.16. Application Hosting

2.16.1. Application Hosting - Introduction

What is Application Hosting?

Application Hosting is a feature that allows users to deploy and run containerised applications directly within WeOS. It provides an isolated environment for applications while leveraging system resources such as networking, storage, and computing power. This approach ensures flexibility, security, and scalability for running third-party or custom applications alongside core system services.

What is an Application Image?

An application image is a packaged representation of an application, including:

- The executable binaries.
- Required libraries and dependencies.
- Configuration files and metadata.

Images follow the OCI (Open Container Initiative) specification, ensuring compatibility with standard container runtimes. They serve as the source for creating application containers.

What is an Application?

An application in this context refers to the running instance of an application image inside a container. It operates in an isolated environment with its own:

- Network namespace (dedicated interfaces and virtual Ethernet pairs). Alternatively it can share network namespace with the host operating system.
- Filesystem paths (including access to external media for persistent storage).
- Resource limits (CPU, memory, etc.).

Applications can be managed through CLI or Web interface, allowing configuration, lifecycle control, and monitoring.

Underlying Technology

The Application Hosting feature is built on the OCI container model, which defines:

- Standard image format for portability.
- Runtime specification for consistent container execution.
- Namespace and cgroup isolation for security and resource control.

Runtime Security Controls

To help protect the host system and other applications, Application Hosting utilise kernel-enforced security controls seccomp and AppArmor.

Seccomp

Application Hosting makes use of seccomp filtering to limit which Linux system calls an application can use. This reduces the risk of misuse if an application is compromised by blocking the execution of a set of system calls. Application hosting uses a white-list approach where seccomp profiles define a set of system calls that are explicitly allowed, and system calls that are not in the list will be denied.

AppArmor

AppArmor enforces access rules for files, devices, and other system resources, so each application only uses what it is explicitly allowed to access. The privileges of an application will be restricted by an AppArmor profile. These profiles regulate allowed and denied actions for:

- Various types of networking.
- Capabilities.
- Path permissions (i.e. read, write, executable, etc.).
- Signals.
- Shell commands.

Together with seccomp, these controls strengthen isolation, reduce attack surface, and provide an additional layer of security for containerised workloads running on WeOS.

What Can You Do with Applications? (Use Cases)

- **Network Services:** Host custom network services (e.g., DHCP, DNS, monitoring agents) within isolated containers.
- **Firewall:** Give exclusive access to a port facing an unsafe network to filter ingress and egress traffic.
- **Data Processing:** Run analytics or processing tools without interfering with the base OS. This could be as simple as measuring traffic flow or it could be running advanced machine learning algorithms for Deep Packet Inspection (DPI), Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS).
- **Third-Party Integrations:** Deploy vendor-provided applications or tools in a controlled environment.

© 2026 Westermo Network Technologies AB

2.16.2. Application Hosting - Environment

Applications in WeOS run in an isolated environment whilst leveraging system resources from the device, such as file storage, networking, and computing power.

Filesystem

```
     Application Instance
     ^ ^ ^ ^ ^ ^  ^ ^ ^ ^
     | | | | | |  | | | |        Filesystem Operations (open/read/write)
     v v v v v v  v v v v
.---------.
|  Share  |                      Host Operating System Path (ro/rw)
:---------'---------.
|      Overlay      |            Non-Persistent Storage (rw)
:-------------------'---------.
|      Application Image      |  SquashFS File (ro)
'-----------------------------'
```

Since applications run in an isolated environment it will only see the files from its own file system. Files not backed by persistent storage are writable for applications, but wont persist over reboots since they are only backed by non-persistent RAM. In order to provide persistent storage to applications, even though the image is read-only, directories on external media can be shared from WeOS to an application. A bind mount is used to import the path to the application, as such the path can be mounted under a different path inside the application. E.g. a directory on a USB stick, `/media/external/instance1/`, can be mounted as `/etc/` in the application.

Network

Applications may be connected to different layers of the WeOS networking stack. In a system with no applications running, a typical configuration may look something like this:

```
     .-------.  .-------.
     | vlan1 |  | vlan2 |       L3
     '---+---'  '---+---'
         '-.      .-'
       .---+------+---.
       |      br0     |         L2
       '-+---+--+---+-'
   .-----' .-'  '-. '-----.
.--+-.  .--+-.  .-+--.  .-+--.
|eth1|  |eth2|  |eth3|  |ssl1|  Port
'----'  '----'  '----'  '----'
```

At the bottom we see the physical and virtual ports, loosely this is any single port that can transport Ethernet, e.g. a 1000-baseT port or an OpenVPN tunnel in L2 mode. These are then typically connected to the bridge (br0) to enable L2 forwarding (switching). On top of the bridge sit the VLAN interfaces. This is usually where we enter the IP stack and thus it is between these interfaces that L3 forwarding (routing) takes place. Applications can be configured to integrate with the operating system on all three layers.

L3 Interface

```
WeOS                                      # Application Instance
     .-------.  .-------.    .-------.    #   .------.
     | vlan1 |  | vlan2 |    | myapp |    #   | host |
     '---+---'  '---+---'    '---+---'    #   '--+---'
         '-.      .-'            |        #      |
       .---+------+---.          |        #      |
```

```
       |      br0     |          |       #      |
       '-+---+--+---+-'          |       #      |
   .-----' .-'  '-. '-----.      |       #      |
.--+-.  .--+-.  .-+--.  .-+--.   |       #      |
|eth1|  |eth2|  |eth3|  |ssl1|   |       #      |
'----'  '----'  '----'  '----'   |       #      |
                                 '--------------'
```

In this configuration, the application is connected to the host by creating a virtual Ethernet pair (vEth pair) where one end (`myapp`) is attached to WeOS's network namespace and the other one (`host`) is connected to the application's ditto.

Depending on the WeOS configuration, access can be limited via the firewall, traffic can be masqueraded to appear to originate from the WeOS itself (NAPT), the internal subnet can be exported using a dynamic routing protocol like OSPF.

L2 Interface

```
WeOS                                     # Application Instance
     .-------.  .-------.                 #
     | vlan1 |  | vlan2 |                 #
     '---+---'  '---+---'                 #
         '-.      .-'                     #
       .---+------+---.                   #   .------.
       |      br0     +-------------.      #   | host |
       '-+---+--+---+-'             |      #   '--+---'
   .-----' .-'  '-. '-----.         |      #      |
.--+-.  .--+-.  .-+--.  .-+--.  .---+---.  #      |
|eth1|  |eth2|  |eth3|  |ssl1|  | myapp |  #      |
'----'  '----'  '----'  '----'  '---+---'  #      |
                                   '-------------'
```

Just like with an L3 Interface, communication between the host operating system and the guest application is through a vEth pair. The difference being that instead of using `myapp` as an L3 interface, it is attached to the bridge.

WeOS still has control over which networks the application can reach through the VLAN configuration. As the interface now works as a port, it can be a member of one or more VLANs.

In this setup, the application can be attached to the same broadcast domain as devices connected to the physical ports (`eth1..eth4`). I.e. it can resolve neighbors using ARP, send IP broadcasts etc. From WeOS's perspective it is perceived as an additional port.

Exclusive Port Access

```
WeOS                       # Application Instance
     .-------.  .-------.  #
     | vlan1 |  | vlan2 |  #
     '---+---'  '---+---'  #
         '-.      .-'      #
       .---+------+---.    #
       |      br0     |    #
       '-+---+--+-----'    #
   .-----' .-'  '-.        #
.--+-.  .--+-.  .-+--.     #  .----.
|eth1|  |eth2|  |ssl1|     #  |eth3|
'----'  '----'  '----'     #  '----'
```

In this mode, an existing physical port is transferred to the application's namespace. WeOS will not be able to use the port for any services. The application is free to send and receive any packets, setup netfilter and tc rules etc., just as with any standard NIC.

This mode is most often used in combination with either an L3 Interface or an L2 Interface. Imagine a firewall and routing application which is used to connect to, and filter traffic from, an ISP. It uses a physical port on one side to connect to the ISP (`eth4`). Once traffic has been filtered, routed, and possibly address translated, it is sent to WeOS via an L2 Interface (`fw0`/`host`):

```
WeOS                                # Firewall Application
     .-------.  .-------.           # .-------.  .-------.
     | vlan1 |  | vlan2 |           # | vlan1 |  | vlan2 |
     '---+---'  '---+---'           # '---+---'  '---+---'
         '-.      .-'               #     '---.  .---'
       .---+------+---.             #       .-+--+-.
       |      br0     +-------.     #       | host |
       '-+---+--+-----'       |     #       '--+---'
   .-----' .-'  '-.           |     #          |
.--+-.  .--+-.  .-+--.    .---+---. #          |            .----.
|eth1|  |eth2|  |ssl1|    |  fw0  | #          |            |eth3|
'----'  '----'  '----'    '---+---' #          |            '----'
                              '-----------------'
```

The application has full control over which traffic flows that are allowed to ingress the system, but WeOS can still be used to provide local services such as DHCP.

2.16.3. Application Hosting - Getting started

External Media

External media is a core component of the Application Hosting feature, serving as the primary storage location for application images and related data. This design ensures flexibility, scalability, and isolation between the operating system and hosted applications.

Image Storage

All application images are stored on external media. This approach decouples application content from the system root filesystem, reducing risk and simplifying updates.

Application Data

Applications can save their runtime files, logs, and persistent data on external media. This guarantees that application state and user data remain intact across system upgrades or application restarts. To enable an application to save persistent data, a media share has to be configured for the application.

Enabling Application Hosting

By default, Application Hosting is disabled in the operating system. This entails that no feature functionality is exposed until the license package is installed. To enable this feature, a dedicated license package must be installed, followed by a reboot. To obtain a installation package, contact Westermo Network Technologies AB.

Enabling Application Hosting via CLI

Enabling Application Hosting via the CLI is done through the package context in WeOS.

```
example:/#> package
example:/package/#> install proto tftp 198.19.10.1 app-hosting-license-signed.wmopkg
```

It is possible to install the license file from the `package` context using the following file transfer protocols: *FTP*, *TFTP*, *SCP*, *SFTP*, *HTTP*, or *HTTPS*.

Refer to the Packages Support page for more information regarding installation of packages.

Enabling Application Hosting via WebGUI

Enabling Application Hosting via the WebGUI is done by navigating to the `Maintenance` tab and then selecting the `Package` page. Selecting `Install Package` will allow you the select the license file on your computer and upload it to the device.

Import of Application Images

To configure and run an application in WeOS, an application image needs to be imported to a configured directory on external media. This can be done by placing the image directly in the external media directory before inserting it to the device, or by importing it using built-in methods in WeOS. If you want to import an image in WeOS via CLI it is possible to navigate to `file/app-hosting` context and use the supported file transfer commands. See App-Hosting HowTo - Import Images for an example of how this can be done.

Alternatively, this can done via the WebGUI in WeOS. Navigate to the `Maintenance` tab and enter the `App Hosting` page. Once entered press the `Install Image` button and follow the instructions which will allow you to upload an image from your computer.

Labeling Application Images

In order for an application to be able to use an imported image, the image needs to be labelled. Images are labelled in the CLI through the `app-hosting` admin exec context, more information can be found through Application Hosting - Management. To label an image in the WebGUI, you can navigate to `App Hosting` in the `Maintenance` tab.

2.16.4. Application Hosting - Configuration

Configuration of application hosting is done from the `app-hosting` configuration context in the CLI.

```
example:/#> configure
example:/config/#> app-hosting
example:/config/app-hosting/#>
```

Hardware Resource Allocation

With the application hosting feature it is possible to restrict how much of the hosts' CPU and RAM the applications are able to use. All applications will share the allocated resource and can not be set independently per application.

`[no] cpu [LIMIT]`

Description

Percentage of CPU power to be shared amongst all applications.

Options

Option	Description
`LIMIT`	The percentage of CPU power to be shared. Valid range: [10-50]
`no`	Restore to default value of 25%.

`[no] ram [LIMIT]`

Description

Percentage of RAM to be shared amongst all applications.

Options

Option	Description
`LIMIT`	The percentage of RAM to be shared. Valid range: [10-50]
`no`	Restore to default value of 25%.

Storage

The storage context includes two commands for specifying which external media on your device to search for application images and what path they are located at.

```
example:/config/app-hosting/#> storage
example:/config/app-hosting/storage/#>
```

`[no] media <MEDIA>`

Description

Storage media for application image files.

continues on next page

continued

`[no] media <MEDIA>`

Specify the name of the media as listed in the media boot context. To list all available media from current context use `do show media`. Only media listed under external are available for application hosting.

Options

Option	Description
`MEDIA`	Specify the external media for storage of application images.
`no`	Restore to default media, `external`.

`[no] path [PATH]`

Description

Storage path to the application images.

Specify the path to the application images. The path is relative to the external media selected with the `media` command.

Options

Option	Description
`PATH`	Specify the storage path for application images relative to the media.
`no`	Restore to default media, `app/images/`.

`show`

Description

Show the current storage settings

App Creation and Configuration

To create an application, or configure an existing application, use the `app` command. This will create a new app and open a new configuration context to configure the newly created application, if no app with the specified name exists. If the app exists, you will enter the configuration context of an existing app.

`[no|show] app <NAME>`

Description

Configuration of applications.

Options

Option	Description
`NAME`	Name of existing application. If the name does not exist a new application will be created.
`no`	Remove existing application.

continues on next page

continued

`[no|show] app <NAME>`

Option	Description
`show`	Show configuration of existing application.

```
example:/config/app-hosting/#> app example
example:/config/app-hosting/app-example/#>
```

Upon entering the application configuration context it is possible to edit and configure the application.

`[no] enable`

Description

Enable the application.

Options

Option	Description
`no`	Disable the application.

`[no] autostart`

Description

Enable autostart of the application. With autostart enabled the application will automatically start upon boot and reconfiguration of the system.

Options

Option	Description
`no`	Disable autostart of the application.

`[no] description [DESCRIPTION]`

Description

Set a description for the application.

Options

Option	Description
`DESCRIPTION`	Description of the application, limited to a maximum of 255 characters.
`no`	Remove the description for the application.

`[no] capability [CAPABILITY]`

Description

Set capabilities for the application. For a full list of capabilities refer to the Linux man pages.

> **NOTE**
>
> With AppArmor strict configured, you will not be able to set all capabilities for the application.

Options

Option	Description
`CAPABILITY`	Set capabilities for an application. By default no capabilities are set and a good practice is to only set the capabilities needed by the application since granting unnecessary capabilities can lead to security risks. **CAP_SYS_NICE**: Real-time scheduling, CPU affinity, I/O scheduling, etc. **CAP_NET_RAW**: Needed for raw sockets. **CAP_NET_ADMIN**: Firewall, tunnels, sockets, protocols, TC, routing, etc. **CAP_NET_BIND_SERVICE**: Allow binding to privileged ports. **CAP_FOWNER**: Allows `chmod`, `chown`, etc. **CAP_MKNOD**: Allows creation of device files.
`no`	Remove configured capabilities for an application.

`[no] cmd [PATH]`

Description

Initialisation command for the application.

Options

Option	Description
`PATH`	Path to initialisation command for the application. Example: `cmd /bin/sh` - Useful to debug the bootstrap process.
`no`	Restores to default command: `/sbin/init`

`[no] shell [SHELL]`

Description

Shell to use when attaching to the application.

continues on next page

continued

`[no] shell [SHELL]`

Options

Option	Description
`SHELL`	Shell to use when attaching to the application. Example: `shell bash` - used in Ubuntu-based images. `shell sh` - used in Alpine-based images.
`no`	Restores to default shell: `sh`

`[no] netns <user-ns|host-os>`

Description

The network namespace to assign an application to.

Options

Option	Description
`user-ns`	The application will get its own, unique, network namespace.
`host-os`	The application will share network namespace with the host OS. This removes the possibility to reserve ports to the application.
`no`	Restores to default setting: user-ns

`[no] apparmor <strict|relaxed> [complain-mode]`

Description

Select AppArmor profile

> **NOTE**
>
> With the strict profile, you will not be able to configure port and veth shares with `none` IP address option for the application. You will also not be able to set all capabilities for the application.

Options

Option	Description
`strict`	Offers high security and low flexibility.
`relaxed`	Offers low security and high flexibility, only configurable for administrators.
`complain-mode`	Allows applications to perform privileged operations that would normally be denied by the defined profile. Instead, these operations are allowed with an `apparmor="ALLOWED"` message to the syslog through the kernel audit subsystem. This is useful for debugging during application development, not recommended for production use. If the “complain

continues on next page

continued

`[no] apparmor <strict|relaxed> [complain-mode]`

Option	Description
	mode" parameter is omitted, the profile runs in "enforce mode": violations are both logged and blocked.

The complete profiles are available at Application Hosting Appendix.

Options

Option	Description
`no`	Restores to default setting: strict

`[no] seccomp <strict|relaxed>`

Description

Select seccomp profile

> **NOTE**
>
> With the strict profile, you will not be able to configure port and veth shares with `none` IP address option for the application.

Options

Option	Description
`strict`	Offers high security and low flexibility.
`relaxed`	Offers low security and high flexibility, only configurable for administrators.

The complete profiles are available at Application Hosting Appendix.

Options

Option	Description
`no`	Restores to default setting: strict

`[no] share <RES-TYPE> <RES-NAME> [<as> <NAME>] [OPTS]`

Description

Manage shared resources to the application: ports, vEth-pairs, and external media directories. This command can be used to assign physical ports, manage virtual ethernet pairs, and share files/folders between the host OS and the app.

continues on next page

continued

`[no] share <RES-TYPE> <RES-NAME> [<as> <NAME>] [OPTS]`

> **NOTE**
> - Paths won't automatically be removed if the share is later removed.
> - Because of restrictions in strict Seccomp and AppArmor profiles, port and veth shares with `none` IP address option are limited to applications with relaxed profiles.

Options

Option	Description
`RES-TYPE`	Type of resource to share: port, veth, or media
`RES-NAME`	Name of the resource to be shared, *e.g.* `eth3`, `lan0`, or `external app/data`
`NAME`	Name of the resource within the application.
`OPTS`	Share specific options, `read-only` and `create` for media shares and IP configuration for veth and port shares
`no`	Remove configured shares.

Examples:

Virtual Ethernet port for VLANs or iface. The application side of the vEth pair will be set to use DHCP for IP address assignment.

```
example:/config/app-hosting/app-example/#> share veth fw0 as lan0 dhcp
```

Move physical port eth12 to an application. The static IP address of `1.2.3.4/24` will be assigned to the port within the application.

```
example:/config/app-hosting/app-example/#> share port eth12 static 1.2.3.4/24
```

Move physical port dsl3 to an application as eth1. No IP address will be assigned the port within the application.

```
example:/config/app-hosting/app-example/#> share port dsl3 as eth1 none
```

Share host directory `app/data` on media `external` as `/mnt` in the application.

```
example:/config/app-hosting/app-example/#> share media external app/data as /mnt
```

`[no] environment [NAME] [VALUE]`

Description

Set environment variables for the application environment.

continues on next page

continued

`[no] environment [NAME] [VALUE]`

Options

Option	Description
`NAME`	The name of the variable within the application.
`VALUE`	The value of the variable within the application.
`no`	Removes configured environment variables.

Examples:

Set the value `foo` for variable `FOO`

```
example:/config/app-hosting/app-example/#> environment FOO foo
```

Set the value `foo bar` for variable `FOOBAR`

```
example:/config/app-hosting/app-example/#> environment FOOBAR "foo bar"
```

Empties the value of variable `BAR`

```
example:/config/app-hosting/app-example/#> environment `BAR`
```

`[no] image <LABEL>`

Description

Specify the image an application should use. An application will not start if there is no image associated with it.

Options

Option	Description
`LABEL`	The label of an image. Images are identified by their labels, to allow seamless switching between them, *e.g.* when upgrading an image. In practice this means that whenever an image is upgraded, application does not have to be reconfigured to point to this new image, as it uses image label, which already points to the new image. Multiple applications may use the same image, by specifying the same label. Run `do show app image` to list available images and their labels.
`no`	Removes the configured image for an application.

`show`

Description

Show the configuration for the current application.

2.16.5. Application Hosting - Management

Management and status of images and created applications can be found under `app-hosting` context under admin exec mode.

```
example:/#> app-hosting
example:/app-hosting/#>
```

`[no] label <IMAGE> <LABEL>`

Description

Set/change the label of an image. Labels are used to identify images when configuring apps.

IMPORTANT: the effect of changing a label is only applied after issuing a `reload [LABEL]` command.

If the label already exists on another image, it will be moved. After moving the label to a new image, the `reload [LABEL]` command should be used to update and restart any running apps.

Options

Option	Description
`IMAGE`	The image to set a new label for. Needs to be an existing image on the system.
`LABEL`	The name of the label. Maximum 15 characters.
`no`	Removes the configured label for an image.

`reload [LABEL]`

Description

Update applications whose images match the label, and restart them. If no label is specified, all applications are updated and restarted. The effect of changing a label is only applied after a reload or reconfiguration. Useful if you have updated an existing image that you want to deploy to all applications using the old image.

Options

Option	Description
`LABEL`	The label of an image you want to reload. All applications configured with this label are restarted.

`attach <APP>`

Description

Attach to running application. Run `exit` or press `ctrl-d` to detach.

Options

Option	Description
`APP`	Name of the application you want to attach to.

`start <APP>`

Description

Start a stopped application.

Options

Option	Description
`APP`	Name of the application you want to start.

`stop <APP>`

Description

Stop a running application.

Options

Option	Description
`APP`	Name of the application you want to stop.

`restart <APP>`

Description

Restart a running application.

Options

Option	Description
`APP`	Name of the application you want to restart.

`show log <APP>`

Description

Show generated logs for an application.

Options

Option	Description
`APP`	Name of the application you want to view logs for.

`follow log <APP>`

Description

Follow generated logs for an application.

continues on next page

continued

`follow log <APP>`

Options

Option	Description
`APP`	Name of the application you want to follow logs for.

`show [app [APP]]`

Description

Show a summary of application hosting.

Options

Option	Description
`app [APP]`	Show summary of all enabled applications or the details of one application.

WeOS upgrade

It is strongly recommended to stop all applications before launching upgrade of WeOS. This is to allow smooth upgrade of the system and avoid out-of-memory issues, since applications altogether might consume up to 50% of system RAM.

HowTo Guide

For detailed instructions on how to enable the feature, import images, and run applications, refer to:

- HowTo Application Hosting

2.16.6. Application Hosting - Common Problems

Program inside application is getting "permission denied"

Applications are very restricted by default. Start by setting AppArmor and Seccomp profiles to `relaxed`. If that does not help, consider what capabilities may be required. Some common ones are:

- CAP_NET_BIND_SERVICE: allows binding to a TCP/UDP port < 1024.
- CAP_NET_RAW: allows opening raw sockets.
- CAP_NET_ADMIN: allows managing network interfaces.
- CAP_SETUID/CAP_SETGID: allows setting UID/GID.
- CAP_CHOWN: allows changing file ownership.
- CAP_DAC_OVERRIDE: allows bypassing file permission checks.

CAP_SYS_ADMIN is not recommended to use as it grants many permissions, but for debugging it can be useful. It is also possible to set all capabilities at once.

```
example:/config/app-hosting/app-testapp/#> capability privileged
```

Note that the strict AppArmor profile disallows many capabilities.

Program inside application is getting "Function not implemented"

Applications are restricted to a set of allowed syscalls defined in the relaxed and strict profiles. The default action for seccomp filtering is to set an errno when a process tries to do the syscall. The chosen errno is `ENOSYS`, which translates to "Function not implemented". This is an uncommon error to get and seccomp filtering may be the cause. Verify that this is the case by looking either in the audit log or in syslog. Seccomp denials are logged with information such as PID responsible, executable name as well as which syscall (number) was denied.

No network traffic on shared port/veth

A shared port should receive traffic as long as the cable is plugged in properly.

A veth port must either be added to a VLAN on the host for L2 forwarding to happen, or assigned an IP and routing rules for L3 forwarding.

```
example:/config/vlan-1#> untagged veth1
```

No IP address on shared port/veth

An IP address can be acquired through different means. WeOS provides two options as shown below.

```
example:/config/app-hosting/app-testapp/#> share veth veth1 as eth0 static 10.1.2.3/24
example:/config/app-hosting/app-testapp/#> share veth veth2 as eth1 dhcp
```

Static IP configures the application's port `eth0` with the given IP. DHCP starts a DHCP client in the network namespace on `eth1`. For DHCP to work properly there must be a DHCP server somewhere on the network, and host `veth2` should probably be part of the desired VLAN.

Static and DHCP also works for `share port`.

When AppArmor or Seccomp is set to `strict`, one of the methods above must be used. If both are `relaxed`, it is possible to assign `CAP_NET_ADMIN`, and possibly `CAP_NET_RAW`, and have the application itself set up an IP address or manage its own DHCP client.

Unable to assign an image

Application hosting uses labels on the images to assign and seamlessly swap images without reconfiguration. See Application Hosting HowTo.

Application Status: Unknown

The application failed during creation/start. Likely due to misconfiguration. A common cause is that the application startup command does not exist in the application. System logs can usually provide more details on what went wrong.

```
example:/#> show log
```

Application Status: Stopped

The application exited with exit code zero, indicating everything went as expected and gracefully shut down. Check app logs to verify. If an application exits with a non-zero exit code it is assumed to be a crash and the application is restarted.

```
example:/app-hosting/#> show log testapp
```

Application Status: Bad Signature

This indicates the application failed the signature check on startup. When importing an image in WeOS it will also warn you of this.

Products without Secure Boot can bypass this with the command below. This bypasses signature enforcement for both WeOS and application images. **Do not use this in a production environment.**

```
example:/boot/loader/#> allow-untrusted
```

Virtual products lack a boot loader. Instead, a setting specific for application hosting exists. This is not available on hardware.

```
example:/config/app-hosting/#> allow-untrusted
```

3. Hardening Guides

3.1. Introduction

3.1.1. Overview

The purpose of this hardening guide is to supply guidance on available capabilities, and how to harden WeOS. The focus of this guide is entirely on hardening WeOS. For product and environmental considerations, see **Product Security Guide** and applicable product user guides.

Structure

The hardening guide is divided into four main sections:

- **Security Principles** - *Introduces general concepts related to Hardening*
- **Security Architecture** - *Gives an overview of the Security Architecture of WeOS*
- **Hardening Practices** - *Implementation guidance of security capabilities in WeOS*
- **Checklists** - *Checklists with a selection of security capabilities*

3.1.2. Security Principles

Working with Hardening

Hardening a network is not a one-time simple tick-the-box exercise. It takes time, effort and continuous consideration what should be secured and why.

Dynamic Network Environment

Network environments are inherently dynamic, which make it difficult to perform a “one-and-done” hardening. Examples that contribute to a dynamic network environment include:

- Evolving threat landscape – *new creative ways to break things on purpose*
- Changes being performed to network design or environment – *no network is static*
- Feature releases containing new security capabilities applicable for hardening – *new cool features*

Feature releases including new security capabilities are developed to respond the evolving threat landscape. Therefore, it is important to evaluate currently implemented and new security capabilities on a regular basis.

Proactive Hardening & Attack Surface Reduction

Proactive hardening refers to securing components before any incident occurs. It involves configuring components to minimise potential weaknesses—such as disabling unused services, enforcing strong access controls, and removing default credentials. These measures are implemented in advance to prevent exploitation and ensure resilience against both internal and external threats.

Attack surface reduction is a key outcome of hardening. The attack surface includes all the points where an unauthorized user could attempt to gain access or extract data. By limiting exposed interfaces, restricting access, and eliminating unnecessary functions, the risk of threats such as unauthorized access, misconfigurations, or exploitation of outdated software is significantly reduced.

Use of Secure Protocols

Secure communication protocols are fundamental to protecting data as it travels across networks. These protocols are designed to ensure that information remains confidential, unaltered, and accessible only to authorised parties. They achieve this through mechanisms such as encryption, authentication, and integrity checks.

Confidentiality ensures that data is not readable by unauthorized individuals. This is typically achieved through encryption, which transforms readable data into a format that can only be interpreted by those with the correct decryption key. Integrity guarantees that the data has not been tampered with during transmission, often using cryptographic hashes or checksums. Authentication verifies the identity of the sender and/or receiver, ensuring that communication is taking place between trusted parties.

Protocols such as HTTPS, TLS, SSH, and IPsec are widely used to secure web traffic, remote access, and network communications. These protocols are essential when transmitting sensitive information, especially over public or untrusted networks like the Internet.

However, not all protocols offer these protections by default. Legacy or unencrypted protocols such as HTTP, FTP, and Telnet transmit data in plain text, making them vulnerable to a range of security threats. These include eavesdropping, where attackers intercept and read data; man-in-the-middle attacks, where data is altered or redirected; and spoofing, where attackers impersonate legitimate users or systems.

© 2026 Westermo Network Technologies AB

Risk should be evaluated where secure protocols are not available or cannot be used. Risk reduction in legacy systems can include for example stronger access controls, logging & monitoring and stricter segmentation. One common approach is to use VPNs or Tunnels, which create secure communication channels between networks or devices, however it is not a solution for all use-cases.

3.1.3. Security Architecture

Overview

The WeOS Security Architecture is designed to enhance the security and reliability, especially in mission-critical environments where security is essential. It defends against unauthorized access, data leakage, and system misuse through a layered security model. Features, such as login settings and access controls, are visible and user-configurable. Others, like secure boot and encrypted storage for secrets, operate in the background. Together, user-configurable settings and built-in protections serve a unified purpose; to protect mission critical assets.

```
+-----------------------------------------------+
|         Unauthorized Access Protection        |
|   Provides authentication and authorization   |
+-----------------------------------------------+
|            Availability Protection            |
|    Provides system reliability and uptime     |
+-----------------------------------------------+
|               Data Protection                 |
| Provides safeguards for sensitive information |
+-----------------------------------------------+
|                 Audit Trail                   |
|   Provides traceability and accountability    |
+-----------------------------------------------+
|                  Protects                     |
|            Mission Critical Assets            |
+-----------------------------------------------+
```

Figure 1: Security Layers in WeOS

Unauthorized Access Protection

Unauthorized access protection is enforced through multiple layers of security. Local and remote authentication verifies that only authorised users can access system status and make changes. Role-Based Access Control (RBAC) restricts user actions based on assigned roles, minimizing the risk of privilege misuse. Local console access protection limits sensitive commands to users with physical access to the device, enhancing security by protecting from remote misuse. Pre-login banners provide configurable messages to raise awareness and present legal notices before login. Additionally, Port Network Access Control using 802.1X blocks unauthorized devices from connecting to the network.

Availability Protection

Availability protection helps maintain system stability and resilience against faults and resource exhaustion. The Watchdog feature automatically recovers the system from hardware or software malfunctions. The Audit Ring Buffer maintains uninterrupted audit logging by overwriting the oldest entries. The Firewall protects system availability by blocking malicious or unwanted traffic. Additionally, remote session limits help prevent resource exhaustion by restricting the number of concurrent remote management sessions.

Data Protection

The system incorporates multiple layers of integrity and authenticity checks to provide secure startup and data protection. Secure Boot validates software and firmware during startup, with support for hardware- or software-based implementations depending on the capabilities of the executing hardware platform. Boot Integrity verifies the bootloader to detect tampering, while Configuration Integrity and Schema Validation confirms that configuration files and inputs are legitimate before being applied. Sensitive data such as passwords are protected through Encrypted Secrets and Secure Storage, which prevent secrets being stored in plaintext. Additionally, Private Key Protection prevents keys from being stored unencrypted on persistent media, while disabling

execution from external media helps prevent malware by blocking unauthorized code from running when external devices are connected.

Audit Trail

The audit logging enhances traceability and compliance by securely logging system events, activities, and configuration changes. These logs provide a record of actions taken within the system, supporting accountability and forensic analysis when needed.

3.2. Hardening Practices

3.2.1. General Information

Hardening Overview

The hardening practices are provided in two sections;

- **Fundamental Hardening** reflects hardening practices that should be considered as mandatory, regardless of implemented design or environment. Fundamental Hardening does not require any external dependencies.
- **Enhanced Hardening** reflects hardening practices that provide enhanced security. Generally specific to implemented design, environment or system requirements. Enhanced hardening may rely on external dependencies, which introduces additional requirements for systems outside the device itself.

Security Maintenance

Configuration Context

Throughout the hardening guide, how to get to a given context has intentionally been left out. Carefully note each context which a given setting is applicable to. As example, `config` / `system` context must be entered before typing `encrypt-secrets password`.

```
example:/config/system/#> encrypt-secrets password
```

Working with Configuration

WeOS has multiple configurations, stored in persistent storage and an active copy running in memory. Permanent changes to the configuration should be saved or committed to prevent configuration inconsistencies if the device is rebooted or loses power. Configuration changes can be saved on a device with the following command:

```
Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Use an encrypted protocol for backup & restore configurations via remote interfaces, such as HTTPS. Backup/archived configurations, and the backup repository, must be protected from unauthorized access.

Monitor Security Advisories

Stay informed about vulnerabilities and recommended mitigations by subscribing to Westermo security advisories:

- Security Advisories ▷ Westermo[1]

Report Vulnerabilities

If you discover a security vulnerability, report it to Westermo PSIRT:

- Report a Vulnerability ▷ Westermo[2]

[1] https://www.westermo.com/support/product-security/security-advisories
[2] https://www.westermo.com/support/report-vulnerability

Keep Firmware Updated

Firmware updates and upgrades are released regularly to address security vulnerabilities, and add new security capabilities. It is therefore important to continuously monitor for new releases and apply patches when appropriate, based on operational requirements and risk assessments. When applying a firmware update or upgrade, it is recommended to use an encrypted transfer protocol, such as HTTPS, especially when performing updates via remote interfaces. This ensures the integrity and confidentiality of the update process and helps prevent unauthorized access or tampering during transmission, however this is of higher importance for non-secure-boot-devices.

Removal of Secrets

Clear Secrets

Factory Reset via Web Interface, CLI or Console will clear all secrets described in Factory Reset. Example for performing factory reset can be performed with the following command:

```
example:/#> factory-reset
```

Reference: Factory Reset

Purge Secrets

Performing Factory Reset via Bootloader: This operation can only be executed through the Console port. It will purge the configuration partitions, including those where sensitive data such as secrets are stored, including secrets mentioned in Factory Reset. This approach is recommended when the device is taken out of its installed environment.

1. To enter the bootloader menu, press `Ctrl + C` upon boot.‘
2. `Enter 4` and `Press Enter` when the following boot menu is presented:

   ```
   Barebox Boot Menu
         1: Primary Partition
         2: Secondary Partition
         3: Network (BOOTP)
         4: Factory Reset
         5: System Recovery
   ```

3. `Enter y` and `Press Enter` when the following option is presented:

   ```
   Performing a factory reset will erase all configuration data,
   but will keep the currently installed firmware.

   Are you sure that want to erase all configuration data? y/[n]
   ```

   > **IMPORTANT**
   >
   > The prompt is case-sensitive. If the correct character `y` is not entered exactly, the purge operation will not be executed.

4. The device will now purge the partitions and reboot

   ```
   Erasing Barebox configuration ..................................... [ OK ]
   Erasing System configuration ...................................... [ OK ]
   ```

```
Device was successfully factory reset.

The device will now reboot.
```

3.2.2. Fundamental Hardening

Local administrator accounts and passwords

Initial User Account

Factory-config contains publicly known username and password for the admin account. Publicly known passwords are prime targets for malicious automated scanners. By using default credentials, anyone with access can gain full privileges to the device.

> **TIP**
>
> Consider using strong password with high entropy, consider setting up SSH passwordless authentication by leveraging SSH-keys.

It is possible to remove the initial administrator account, WeOS requires that at least one administrator exists on the device, therefore a new user with the role administrator must first be created. Note that an administrator cannot remove itself. It is recommended to create a new administrator user and remove the default admin account, this will make it harder to guess the username. As minimum, update the default admin account with a new password.

As minimum, replace the default admin password.

```
example:/config/aaa/user-admin#> password alg sha512 secret NewPassword

example:/config/aaa/user-admin#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

As recommended, create a new user with role administrator and remove the default admin account.

```
example:/config/aaa/#> user MyNewAdminAccount
example:/config/aaa/user-MyNewAdminAccount#> password alg sha512 secret NewPassword

example:/config/aaa/user-MyNewAdminAccount#> show
```

Logout as admin and login with MyNewAdminAccount to remove the admin account since a user cannot remove itself

```
example:/config/aaa/#> no user admin

example:/config/aaa/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: AAA

Encrypt Secrets

It is important to protect sensitive information in the configuration backup files. This can be achieved with the Encrypt secrets functionality. The device support different encryption sources, which should be selected depending on requirements and policies. In this example a password will

be used in order to be able to decrypt all sensitive information in the configuration file, this method enables portability if the same password is used across multiple devices. Backup of the secret and configuration file should be stored separately.

```
example:/#> encrypt-password C0nF!Gp4sSw@RD382
example:/config/system/#> encrypt-secrets password
example:/config/system#> leave

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Encrypted Secrets

Remote administration and network services

Set an acceptable session timeout

Setting a timeout for idle connections ensures that sessions automatically close after a defined period of inactivity. Without a timeout, or if it's set too long, idle connections may persist indefinitely, which can potentially lead to session hijacking, allowing unauthorized users to access the device. Timeout value depends on the environment (window-of-opportunity) and desired user-experience. It is recommended to set the session timeout for administrative connections to five minutes or less.

> **NOTE**
>
> Factory-config does not contain a default session timeout for the console connection. Failing to set session timeout on the console may result in unauthorized users being able to access the device through an old user session.

Note timeout unit for each setting.

```
example:/config/management/web/#> session-timeout 5
example:/config/management/ssh/#> idle-timeout 300
example:/config/management/cli/#> timeout 300

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Web Interface | Secure Shell (SSH) | Command Line Interface (CLI)

Limit authentication attempts

Lockout users with successive failed login

The lockout policy is a security feature that prevents unauthorized access to the system by locking out users who enter incorrect login credentials too many times. It is recommended to restrict failed remote administration attempts to a maximum of three or less. To mitigate locking out all users (resulting in Denial-Of-Service scenario), it is recommended to exclude the console port to always be able to login via console, however it depends on the installed environment.

```
example:/config/aaa/lockout-policy/#> enable
example:/config/aaa/lockout-policy/#> exclude console
```

```
example:/config/aaa/lockout-policy/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Fail Delays

It is recommended to introduce a delay of at least one second between login attempts to significantly slow down brute force attempts.

```
example:/config/management/ssh/#> fail-delay 2

example:/config/management/ssh/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: AAA | Secure Shell (SSH)

Limit Remote Management Sessions

Limit remote management sessions is available to mitigate exhausting system resources. The max-remote-sessions value depends on the environment and desired user-experience.

> **NOTE**
>
> If a limit on remote management sessions is set, it is strongly recommended to set inactivity timeouts for all enabled remote login methods, to avoid locking out users.

```
example:/config/management/#> max-remote-sessions 3

example:/config/management/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Limit Remote Sessions

Enforce Usage of Strong Passwords

A password policy should be enforced to ensure users use more complex passwords, making it harder to guess or crack. Various parameters can be configured to increase the password entropy, however the policy should prioritise long passphrases (minimum 15 characters) over strict complexity rules. The longer and more complex password, the higher the entropy. Note that the password policy applies to user passwords. Unique and complex password/secret should also be used for routing authentication, VPN tunnels, DDNS, Simple Network Management Protocol (SNMP) auth, Radius/TACACS+ shared secret, and anywhere else passwords/secrets are stored in the configuration.

```
example:/config/aaa/password-policy/#> enable
example:/config/aaa/password-policy/#> min-length 15
example:/config/aaa/password-policy/#> reject-username

Depending on password policy, configure:
```

```
example:/config/aaa/password-policy/#> uppercases 4
example:/config/aaa/password-policy/#> lowercases 4
example:/config/aaa/password-policy/#> digits 5
example:/config/aaa/password-policy/#> specials 2

example:/config/aaa/password-policy/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Password Compliance Policy

Replace Self-Signed Certificate for Web

Using a default self-signed certificate leaves devices vulnerable to spoofing, MITM interception, and trust warnings that users may ignore. Replacing it with a certificate issued by a proper PKI enables chain-of-trust validation, centralized revocation, lifecycle management, and auditability. This improves security posture, reduces operational risk, and aligns with compliance expectations for encrypted management interfaces. For more information about retrieving certificates using SCEP and utilising CRL, see Public Key Infrastructure.

Prepare a PKCS12 bundle containing the end-entity certificate, matching key and necessary CA certificates. The bundle is protected by password.

```
  openssl pkcs12 -export -out newcert.p12 -inkey newcert.pem -in dut1.pem -passout
$ pass:westermo -certfile int-ca.pem -certfile root-ca.pem
```

Verify that device's web interface is not trusted when the `web-default` certificate is used:

```
$ curl --cacert root-ca.pem https://192.168.122.70/
curl: (60) SSL certificate problem: self-signed certificate
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
```

Import PKI artifacts using secure copy (SCP) from user alice, verify import and apply to the web:

```
example:/pki/#> import pkcs password westermo label newcert user alice proto scp 192.168.122.1 /certs/newcert.p12

example:/pki/#> show cert full

example:/config/management/web/https/#> certificate newcert

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Verify that device's web interface is now trusted:

```
$ curl --cacert root-ca.pem https://192.168.122.70/ | head
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2844    0  2844    0     0   110k      0 --:--:-- --:--:-- --:--:--  111k
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
```

```
<title>newcert - </title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Reference: PKI - Certificates

Notification & Consent Banners

Login banners on network devices serve a critical role in both cybersecurity and legal compliance. Login banners are not only informational, they establish clear boundaries and expectations for anyone attempting to access the system. Informing that any unauthorized attempts may result in legal action can be a barrier to cross for some unauthorized users. However it also provides other important parts that could be required depending on organizational or regulatory requirements, such as informing about disclaimers and consent to logging and monitoring. For further read and examples, visit Guidance on Consent Banners at CISA.gov[1]

```
example:/config/management/web/#> login-banner "This is a message."
example:/config/management/ssh/#> login-banner “This is a message."
example:/config/management/console/#> login-banner “This is a message."

example:/config/management/web/#> show
example:/config/management/ssh/#> show
example:/config/management/console/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Web Interface | Secure Shell (SSH) | Console

Local Audit-Logging

It is strongly recommended to use time-synch server for universal time across the system and remote syslog sinks for offloading auditable events to a separate device for data handling, see Enhanced Hardening. If a remote system can not be used, consider setting time manually and leverage persistent logging to external media, this approach should be considered as a last resort (not recommended) since local logging and lack of universal time-synch will make incident response handling difficult and time consuming.

Apply rules to packet flows

When connecting a network to the Internet or any non-trusted network a router with Firewall functionality can be used to protect against undesired access and traffic to your local network and devices. Undesired traffic can include for example ICMP flooding and Ping-of-death types of attacks. A firewall should be considered where traffic is moving in, out or between different Security Zones (See Product Security Guide for further read). By doing so, strict control of the network traffic can be enforced. The firewall rules that is applicable is dependent on the logical environment, which can be defined with which traffic that is required to flow in, out or between different Security Zones. The following rules can be applied for enabling management traffic (SSH, HTTPS) on a specific VLAN, in this case VLAN 500.

[1] https://www.cisa.gov/resources-tools/resources/guidance-consent-banners

> **WARNING**
>
> If you don't allow any management access (ssh, https) on any interface you will not be able to reach the device after enabling the Firewall. Be careful and don't lock yourself out!

```
example:/config/ip/firewall/#> policy input drop
example:/config/ip/firewall/#> policy forward drop
example:/config/ip/firewall/#> input accept in vlan500 dport https proto tcp
example:/config/ip/firewall/#> input accept in vlan500 dport ssh proto tcp
example:/config/ip/firewall#> enable

example:/config/ip/firewall/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Firewall

Limit Broadcast, Multicast and Unknown Unicast Traffic

The switch can be configured to limit the rate of a port's incoming traffic, this is called ingress rate limiting. By default a port accepts packets at a rate up to the link speed, with ingress rate limiting activated the switch starts dropping packets when data arrives above the given rate threshold. This feature can be useful as a complement to layer-2 priority handling when congestion within the network is to be avoided. To minimise the risk of high levels of traffic with a broadcast behaviour entering the network, ingress rate limiting can be used to limit the amount of Broadcast, Multicast and Unknown Unicast that are allowed in on a port.

```
example:/config/#> port eth2
example:/config/port-eth2/#> ingress-rate-limit 1M match bc,mc,u-uni

example:/#> show port eth2

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: EthernetPorts

Recovery from unresponsive system

The watchdog is a functionality that can help the system recover from various possible malfunctions that could cause the system to enter a state where it is not able to operate correctly. Anything from a hardware fault or program errors may cause the watchdog to trigger. By default, Watchdog is enabled. The system allows for disabling the watchdog. However, this is not something that is recommended. The current status of the watchdog can be observed with:

```
example:/#> show watchdog
```

Reference: Watchdog

Disable Unused Protocols and Services

Upon initial startup, WeOS is loaded with a factory-default configuration. This setup is intentionally designed to prioritise ease of use, enabling users to quickly access and begin working with the device. While a large set of security policies is enabled by default, the configuration prioritises user

accessibility and ease of setup, intentionally limiting strict security controls to streamline the initial experience.

> **WARNING**
>
> Each factory-enabled service and protocol should be carefully conceded if it should be enabled or disabled as part of the hardening.

Default Management Services

All management services are enabled per interface, i.e., VLAN1, VLAN2, VLAN500 etc. Changing a management service on one interface, does not impact other interfaces. All implementations of management services must be configured individually per interface. The table below is a summary of management services with related default values.

Service	Port	Default Value	Description
HTTP	80	Enabled	Disable HTTP. HTTP does not encrypt data between the client and server, making it vulnerable to interception. If Web access is required, use HTTPS.
HTTPS	443	Enabled	Only leave HTTPS enabled if you use the web interface on the device, disable HTTPS on all interfaces that does not require access . HTTPS uses its own certificate for validation and default certificates should be replaced for additional security
SSH	22	Enabled	Only leave SSH enabled if you use the remote console services on the device. If not, disable SSH on all interfaces that does not require remote console access
Telnet	23	Disabled	Keep disabled, only enable if there is a very specific requirement for Telnet. Use SSH as the preferred method.
SNMP (poll)	161	Enabled *poll community set to “public”*.	Disable SNMPv2. SNMPv2 is vulnerable to interception since it transmits authentication community strings and other network data in plaintext. The “public” poll community is publicly known information, anyone with basic knowlege can get operational information about the device. Instead use SNMPv3 which provides authentication and encryption
SNMP (traps)	162	Disabled	Leave blank. Use SNMPv3 as the preferred option if SNMP is required. Disable SNMPv2 attributes via console.

Reference: HTTP | HTTPS | Secure Shell (SSH) | Telnet | SNMP

Default Protocols & Services

It is possible to disable default enabled protocols and services, however it should be carefully considered if it should be enabled or disabled, since it depends on the environment and requirements of the network.

To disable SSDP and MDNS:

```
example:/config/discover/#> no mdns
example:/config/discover/#> no ssdp

example:/config/discover/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

LLDP can be useful for troubleshooting purposes, however it also provides network information which can be disclosed to users. Where it is possible, disable LLDP. If it is not possible, only enable LLDP on interfaces connected to trusted devices. Disable it on edge ports or guest networks to prevent exposure. In the following example, port eth9 is a edge port.

```
example:/config/lldp/#> no port eth9
example:/config/lldp/#> show port

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: LLDP

Default Settings

The default settings described have effect on how the device behaves. Carefully consider each setting and it´s impact related to the installed environment.

To disable IGMP snooping per VLAN:

```
example:/#> config vlan 1
example:/config/vlan-1/#> no multicast-snooping

example:/config/vlan-1/#> show multicast-snooping

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: IGMP Snooping

IP Forwarding is enabled by default on devices with routing capability. Enabling IP Forwarding enables routing between VLANs without explicit firewall rules or access control lists, allowing lateral movement and increases the risk of compromise spreading across security zones. If routing is not required on the device, it should be disabled. However if routing is required, firewall rules should be applied.

```
example:/config/ip/#> no forwarding
```

Reference: IP Forwarding

Disable Unused Physical Ports

Leaving unused ports enabled on a device could allow an adversary to connect a rogue device to the network and perform information gathering or compromise attempts. All unused ports should be disabled. Unused ethernet ports should always be disabled. For more complex and dynamic environments, such as a road-warrior scenario, consider PortNetwork AccessControl. The feature PortProvisioning can, in some cases, simplify the disable unused ports process.

```
example:/config/port-eth1/#> no enable

example:/#> show port

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

It is possible to disable the auxiliary ports, this will disable USB or SD card. By default, audit logs are not persistently stored and require the need for remote logging sinks. By remote logging sinks, this include multiple syslog servers or external media, such as USB or SD card. Depending on design and requirements, the auxiliary port may be used for persistently storing audit-records. If the auxiliary port is unused, it should be disabled.

```
example:/boot/#> no aux-port

example:/boot/#> show aux-port

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

It is possible to disable the console port, this will block all access to the device via the port except for factory-reset purposes. Disabling the console port is not recommended and is considered as a essential function. The port should always be available for mitigating security issues and solving functional issues, since remotely managing the device might be unavailable at that time.

```
example:/config/management/console/#> no login

example:/config/management/console/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Ethernet Ports | Aux | Console

Avoid Untagged Native VLAN on Trunk Ports

In WeOS, any VLAN where a port is configured as untagged can be considered the native VLAN for that port. Therefore, when configuring the VLANs on the device, consider how you combine the untagged and tagged ports, so that a trunk port is not configured as untagged for a VLAN that is not intended, especially because all ports are by default set as untagged in VLAN 1.

Generally it is recommended to avoid having untagged (native) VLANs on any trunk port(s):

- Ensure that any incorrect untagged (native) VLANs are not configured on trunk ports.
  - If an untagged VLAN is required on a trunk port, ensure that it does not facilitate access on unintended access ports.
- If a native VLAN is required on a trunk port, consider which VLAN is used, to avoid providing possible access on unintended access ports.

FRAMES TAGGED WITH VID 0 COUNTS AS UNTAGGED

Important to note, frames that are tagged with VID 0 (generally used for priority tagging), are treated as untagged when they ingress on a port.

Example of Untagged VLAN on Trunk Ports

As an example where a potential misconfiguration can lead to unintended access, consider the following scenario:

```
                    Access port                 Trunk                   Access port
  .--.-.                        .--------.                    .--------.                      .--.-.
 ( (    )__                 U:1 |        |     U:1 T:100      |        | U:100               ( (    )__
(_,  \ ) ,_)--------------------+   S1   +--------------------+   S2   +--------------------(_,  \ ) ,_)
  '-'--`--'                     |        |                    |        |                       '-'--`--'
    LAN-1                       '--------'                    '--------'                         LAN-2

Send frame: [0,100] ----------->          [100] ------------->          [] ------------->
Send frame: [1,100] ----------->          [100] ------------->          [] ------------->
```

In the above scenario, an access port on **S1** is configured as untagged in VLAN

1. Similarly, an access port on **S2** is configured as untagged in VLAN 100. Lastly, there is a trunk configured between **S1** and **S2** that is tagged with VLAN 100, however, the trunk ports on both **S1** and **S2** are still left configured as untagged in VLAN 1 by mistake (This could typically happen if configured from the device's factory default configuration).

TAGS ARE STRIPPED WHEN EGRESSING ON AN UNTAGGED PORT

Remember that when a VLAN tagged frame has ingressed on a port tagged in the same VLAN, the tag will be stripped when egressing on a port untagged in the same VLAN.

Because of this, if frames are sent from **LAN-1** with nested VLAN tags, where the outer is tagged with VLAN 1 (or VID 0, counts as an untagged frame) and the inner tagged with VLAN 100, it can seemingly "hop" from **LAN-1** to **LAN-2**, which was not the intention. The reason this happens is because the outer VLAN tag will be stripped from the frame, and it will egress in the untagged (Native) VLAN 1, that is also configured on the trunk port by mistake. What was the inner VLAN tag, VLAN 100, will now be the outer VLAN tag when we egress **S1** towards **S2**. When the frame arrives on **S2**, tagged in VLAN 100, it will now be allowed to egress on the access port untagged in VLAN 100 towards **LAN-2**.

Reference: VLAN

3.2.3. Enhanced Hardening

AAA (Authentication, Authorisation, Accounting)

The three individual terms refer to who is allowed to access what device (Authentication), if access is granted what are the limitations of their available actions (authorisation), and to recount what actions they performed (accounting).

Remote Authentication

For the remote user authentication, different types are supported, see Remote Server Settings - type. In this case, a Radius server will be used. The login context tells the switch to use server 1 as primary authentication method, however it will fallback to built-in accounts for users that are not present on the server. Fallback is also present if the radius server is not available. Specify `encryption-type` to encrypt traffic. If not specified, traffic will be un-encrypted which leaves it susceptible to attacks like packet sniffing, which can compromise credentials or allow for forged requests.

```
example:/config/aaa/#> remote-server 1
Creating new remote server 1
example:/config/aaa/remote-server-1/#> password RADiuSseCret
example:/config/aaa/remote-server-1/#> address 192.168.1.1
example:/config/aaa/remote-server-1/#> type radius
example:/config/aaa/remote-server-1/#> encryption-type peap-mschapv2
example:/config/aaa/remote-server-1/#> ca-cert servercertlabel
example:/config/aaa/remote-server-1/#> auth-port 1812

example:/config/aaa/remote-server-1/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

System requirements: Remote authentication server.

Reference: AAA | AAA - HowTo Centralized Authentication

Apply principle of least privilege

Least privilege is a security concept that authorises access to a person or entity at the lowest privilege level necessary to perform authorised tasks. Many tasks do not require privileged level access, such as viewing status of network interfaces or reviewing routing tables.

To implement least privilege, administrators should assign a suitable role to authorised users. An administrator's role is critical to securing the device and network from potential security issues. Users should be carefully evaluated before being assigned the role `administrator` on the device.

Role	Privileges	Examples of permission
Administrator	Very high	Full access to audit logs and configuration, including security features
Engineer	High	Read access to audit logs, write access to most configuration, excluding security features
Operator	Medium	Read access to audit logs and configuration
Auditor	Low	Read access to audit logs and device status
Viewer	Very low	Access to device status
Guest	Very low	(**Deprecated**, replaced by Viewer) Access to device status

```
example:/config/aaa/#> user alice
example:/config/aaa/user-alice/#> role viewer

example:/config/aaa/user-alice/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: AAA - Authorisation-Levels | AAA - Restricted Users

Force password expiration

When a new user is added, it is recommended to change the password at first login. This can be enforced when the initial password is set with the `expire` flag. Consider enforcing password change at first login or prior to login on the remote authentication server.

```
example:/config/aaa/#> user alice

example:/config/aaa/user-alice/#> password expire alg sha512 secret TemporaryPassword

example:/config/aaa/user-alice/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: AAA

Accounting

Accounting keeps records of relevant resources accessed or actions performed - who did what, when, and how. The detailed information creates a chain of events, which is the sequence of these logged activities over time. From a security perspective, this information is key for responding to a security incident. Auditable Events help to provide this information (see Auditable Events for detailed information and support).

By default audit logging is enabled and can be displayed with:

```
example:/audit/#> show
```

Reference: Auditable Events

Remote logging and monitoring

Time management

Network Time Protocol (NTP) is used to synchronise device clocks globally, ensuring that log message timestamps are accurate. Accurate timestamps are essential for investigating a security incident and correlating logs across multiple devices or timezones. Apart of being important for logging purposes, another example of where accurate time is required is for validating certificates. To maintain reliable timekeeping, each device should connect to at least two trusted time sources to ensure accuracy and availability of the provided information. Internal time servers should be used as the primary source for all devices, which should synchronise with trusted external sources. This will reduce the number of external requests and ensure that local timestamps are being synchronised in the event an external time server is unreachable.

```
example:/config/ntp/#> peer 10.5.8.1
example:/config/ntp/#> peer 10.5.8.2

example:/config/ntp/#> show

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

System requirements: Time server.

Reference: Network Time Protocol - NTP

Centralized remote logging

Once auditable events has been generated, it must be processed and stored for further analysis. Remote sinks should be used to transfer (audit) logs for centralized archiving and analysis. It is vital to transfer auditable events as soon as possible, and at least two different external sinks to ensure availability. Centralized storage provides a new set of possibilities, where there are more storage and processing power available. If actively monitored and evaluated, it provides the opportunity to detect unusual activities, which can be responded to before leading to a serious security incident. For setting up remote sinks for Auditable Events, see Logging to Remote Machines HowTo and Transmitting Auditable Events to Remote Machines HowTo. As destination transport protocol, use TLS. If TLS is not used, data between the client and server will not be encrypted, making it vulnerable to interception.

```
example:/config/logging/destination-example/#> transport tls
```

System requirements: Logging server

Reference: Auditable Events | Auditable Events - HowTo Transmitting to Remote Machines

Monitoring via SNMP

SNMP is used for network management and network monitoring. It enables proactive monitoring of network device health and performance, helping to detect anomalies, prevent security breaches, and ensure the availability of critical systems.

SNMPv2 read-write community strings are sent in clear text, making it vulnerable to interception. If SNMPv2 write is configured, the intercepted information can be used by an adversary to gain control of the device. It is recommended to remove all SNMP read-write community strings and upgrade to SNMPv3 with encryption and authentication. If SNMPv2 read-write community strings is required for remote administration and cannot be removed, implement unique and strong community strings to make it harder to guess from intercepted information.

Disable SNMPv2.

```
example:/config/management/snmp/#> no rocommunity
example:/config/management/snmp/#> no rwcommunity
```

Create a read only SNMPv3 user, verify and save to startup configuration.

```
example:/config/management/snmp/#> rouser USER auth sha512 AUTHPASSWORD crypto aes256 CRYPTOPASSWORD 1

example:/config/management/snmp/#> show
```

```
Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

System requirements: SNMP manager

Reference: SNMP

Port Network Access Control

802.1X

To gain internal network access, an adversary must either find a way through the external perimeter or by gaining access from inside of the network. Port Network Access Control (PNAC) helps block unauthorized devices from connecting physically to the network.

Protection of unauthorized devices connecting to the internal network can be mitigated with multiple protections, which helps to create a strong defence-in-depth. This could include:

- Where possible, disable unused ports, it should be the first choice
- Implement physical access protections (see Product Security Guide for further information)
- For connected devices and dynamic scenarios such as a road-warrior scenario, 802.1X can be used where it is applicable
- Monitor auditable events for link up and link down

Not all devices are 802.1X capable, to mitigate this, the device supports both 802.1X and MAC Authentication. For legacy devices which are not 802.1X capable, MAC Authentication can be used. However 802.1X is preferred over MAC Authentication, since MAC Authentication can be bypassed with MAC spoofing. Note that a MAC address is commonly printed on most devices.

Note that WeOS requires an external RADIUS server to authenticate the 802.1X supplicant. See Remote Authentication

```
example:/config/port-access/#> port eth1..eth4
example:/config/port-access/#> dot1x
example:/config/port-access/dot1x/#> authentication-server 1

example:/#> show port-access

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

System requirements: Remote authentication server

Reference: Port Access Control

Port monitoring

Port monitoring is typically used for connecting an NIDS, diagnosing a problem, or using a network analyzer to monitor the network.

An adversary connected to the destination port will be able to collect network traffic sent through all the source ports. Therefore, environmental risks related to a permanent solution for example NIDS should be considered. Port monitoring should only be enabled for those ports where it is necessary, and all instances should be disabled once they are no longer needed.

```
example:/config/monitor-1/#> source port eth1
example:/config/monitor-1/#> destination eth3
```

```
example:/#> show monitor

Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Reference: Port Monitor

VPN and Tunnels

Not all network protocols provide built-in features for integrity, authentication, and confidentiality. While this may be acceptable within a local conduit, it becomes critical when data is transmitted between conduits.

To ensure secure communication across different sites or when accessing remote areas via the internet, it is essential to use VPNs or secure tunnels. These technologies offer network-to-network protection and should always be strongly considered when:

- Communicating between geographically separated sites
- Transmitting data over public networks such as the internet
- Accessing hard-to-reach remote locations

By using VPNs and tunnels, organisations can maintain the confidentiality and integrity of their data across distributed environments.

For OpenVPN consider the following options: `tls-version-min` should use TLS 1.3 or as lowest TLS 1.2. Versions lower than TLS 1.2 are considered weak and should be avoided. `data-ciphers-fallback` should be disabled since it includes known weak ciphers.

For further read and configuration examples, see OpenVPN and IPsec

System requirements: Supported Server/Client.

Reference: OpenVPN | IPsec

Public Key Infrastructure

Simple Certificate Enrollment Protocol (SCEP) and Certificate Revocation Lists (CRL) are designed for simplifying and automating the process of certificate issuance and management within a Public Key Infrastructure (PKI) environment. If SCEP is not applicable for the use-case, certificates can be uploaded to the device manually.

Revocation Policy support varies between services that is being integrated to work with SCEP (see configuration guide for exact support). Revocation Policies has been implemented to give the option to balance security and availability based on operational needs. `relaxed` is a good option for balancing both availability and security, for example it allows a VPN tunnel to remain up even if the CRL could not be downloaded in time, which could be useful in hard-to-reach-remote-areas. However the associated risk with the option should carefully be considered and selected for each use-case. Selectable Revocation Policies:

- `disabled` - Revocation check is disabled (default)
- `relaxed` - Revocation check is enabled, reject if certificate is revoked
- `strict` - Revocation check is enabled, reject if CRL is not present, revoked, outdated or fail signature check

Certificates have defined start and end dates (“valid from” and “expires on”). If the time of the device is incorrect, it might see a valid certificate as expired or not yet valid, which may lead to

disruption in the related service. It is therefore important to have a accurate time, see Time management.

For enrollment, it is recommended that the `challengePassword` is a unique one-time authenticator to limit the ability of an attacker who can capture the authenticator from the client or CA and reuse it to request further certificates. For further read see: RFC8894[1].

System requirements: Certificate Authority (RA/CA) server.

Reference: SCEP | Revocation

[1] https://www.rfc-editor.org/rfc/rfc8894.html#name-challengepassword-shared-se

3.3. Checklists

3.3.1. IEC/ISA 62443-4-2 checklist

Introduction

This checklist is a collection of steps to harden devices related to IEC/ISA 62443-4-2 which is connected to the IEC/ISA 62443-3-3 requirements. This specific checklist is a guidance for utilising appropriate security capabilities in WeOS related to IEC/ISA 62443-4-2. It does not provide exact guidance since each environment are unique, has different requirements and therefore capabilities. This list could be used as input for the IEC/ISA 62443-3-3 or 62443-2-1 Factory Acceptance Test (FAT) checklist. Administrators are advised to evaluate each option for its potential risk before implementing the option.

As defined in IEC TS 62443-1-1 there are a total of seven foundational requirements (FRs). Each FR have defined requirements named component requirements (CR). Devices running WeOS are classified as a network device (NDR) which include additional requirements on top of all CRs.

Security Level (SL) are defined from 1 to 4 and describes the level of means, resources, skills and motivation that is reflected in the component capability.

- SL 1 – casual.
- SL 2 – simple means with low resources, generic skills and low motivation.
- SL 3 – sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- SL 4 – sophisticated means with extended resources, IACS specific skills and high motivation.

Checklist

Security Level Target (SL-T) of this checklist is up to and including SL 2. Level of achieved SL (SL-A) depends on the implemented security capabilities, selected product and installed environment. (For further read and information, see Product Security Guide and Product Technical Specifications).

Related Foundational Requirement (RFR in table) refers to:

- FR 1 - Identification & authentication control (IAC)
- FR 2 - Use control (UC)
- FR 3 - System integrity (SI)
- FR 4 - Data confidentiality (DC)
- FR 5 - Restricted data flow (RDF)
- FR 6 - Timely response to events (TRE)
- FR 7 - Resource availability (RA)

RFR	Description	Security Capability	Implemented? YES, NO, N/A
FR 1 - IAC	For unique human user identification and authentication, remote authentication and/or local users has been configured	AAA	
FR 1 - IAC	Default authenticators has been replaced	initial-user-account	

RFR	Description	Security Capability	Implemented? YES, NO, N/A
FR 1 - IAC	New user accounts has enforced password expiration	force-password-expiration	
FR 1 - IAC	Configuration has been implemented to protect authenticators from unauthorized disclosure	encrypt-secrets	
FR 1 - IAC	Configuration has been implemented to enforce password strength	enforce-usage-of-strong-passwords	
FR 1 - IAC	Self Signed certificates has been replaced. Certificate based services has a valid certificate.	public-key-infrastructure replace-self-signed-certificate-for-web	
FR 1 - IAC	Revocation check is enabled	public-key-infrastructure	
FR 1 - IAC	Configuration is implemented to enforce a limit on the number of consecutive invalid access attempts.	lockout-users-with-successive-failed-login	
FR 1 - IAC	Fail-delay mechanisms are implemented to slow down repeated invalid attempts.	fail-delays	
FR 1 - IAC	Pre-login system use notification banners has been implemented	notification-concent-banners	
FR 2 - UC	The configured user role matches the responsibilities in accordance with principle of least privilege, and has been considered for each user	apply-principle-of-least-privilege	
FR 2 - UC	Configuration has been implemented to terminate sessions after a time period of inactivity	set-an-acceptable-session-timeout	
FR 2 - UC	Selected audit logging categories are enabled	accounting	
FR 2 - UC	External media and/or centralized remote logging is configured for archivation of auditable events	centralized-remote-logging	
FR 2 - UC	Synchronised system wide time source has been configured	time-management	
FR 3 - SI FR 4 - DC	Secure variants of protocols, which include communication integrity and authentication has	monitoring-via-snmp disable-unused-protocols-and-services remote-authentication	

RFR	Description	Security Capability	Implemented? YES, NO, N/A
	been implemented where it is applicable	centralized-remote-logging vpn-and-tunnels	
FR 5 - DC	Network segmentation has been considered and implemented which is reflected in VLAN, firewall and VPN configuration	apply-rules-to-packet-flows vpn-and-tunnels	
FR 5 - DC FR 7 - RA	Firewall is enabled and firewall rules has been configured to block unwanted traffic	apply-rules-to-packet-flows	
FR 6 - DC	Auditors have been configured with role to support read-only of audit records	apply-principle-of-least-privilege centralized-remote-logging	
FR 6 - DC FR 7 - RA	Configuration has been implemented to enable the device to continuously be monitored	monitoring-via-snmp port-monitoring	
FR 7 - RA	Watchdog is enabled	recovery-from-unresponsive-system	
FR 7 - RA	Configuration has been implemented to limit the remote management sessions	limit-remote-management-sessions	
FR 7 - RA	Broadcast, Multicast and Unknown Unicast is limited where feasible	limit-broadcast-multicast-and-unknown-unicast-traffic	
FR 7 - RA	Unused services, ports and users has been disabled	disable-unused-protocols-and-services disable-unused-physical-ports	

Integration considerations

Restriction of communication

WeOS firewall has capabilities to block IP and defined services individually. For ease-of-use and more granular restriction, consider DNS-blocking, email blocking and other application blocking communications to be handled on a system level.

Protection from unauthorized device access

Consider implementing Port Network Access Control to add protection against unauthorized devices accessing the internal network.

4. HowTos

4.1. AAA and User Management

4.1.1. AAA - Login with Builtin User Accounts HowTo

About

This document aims to show how to use the built-in user accounts.

For an overview of all available configuration settings for AAA refer to the configuration guide:

- AAA Configuration Guide

Introduction

Configure

This section will cover how to handle built in user accounts.

Builtin users

The table below lists the built-in interactive user accounts present in a factory preset WeOS. These accounts only apply to the login service, and some of them are limited to the console port. It is also possible to create new built-in accounts to replace the existing ones.

Service	Purpose
admin	Administrative account for configuring and management of the device. This is accessible from console, SSH and Web (HTTP/HTTPS) by default.
factory	Factory reset function. Only accessible from console.

Admin

The default setting for the built-in admin account is as follows:

- Login: admin
- Password: admin

The password of the built-in admin user can (and should!!) be changed. as shown in the following example:

```
example:/#> configure
example:/config/#> aaa
example:/config/aaa/#> user admin
example:/config/aaa/user-admin#> password alg sha512 secret NewPassword
example:/config/aaa/#> leave
```

Factory

The built-in factory account can be used to conduct a factory reset via the console port (only).

- Login: factory
- Password: reset

```
example login: factory
Password: reset (typed, but not echoed back to screen)

Factory Reset -- Restore factory default settings.
=> Are you sure (y/N)? y
Erasing barebox configuration
```

```
Backing up theme catalog.
Erasing all certificate and config files.
Restoring branding.
All done, rebooting ...
```

A new account

It is possible to create multiple built-in accounts, up to 16 in one system.

Let us create a new built-in account for user Alice. By default Alice is granted administrator privileges but we change it to guest and set a password hashed using sha512 algorithm.

```
example:/config/aaa/#> user alice
example:/config/aaa/user-alice/#>
example:/config/aaa/user-alice/#> role guest
example:/config/aaa/user-alice/#>
password alg sha512 secret CorrectHorseBatteryStaple
example:/config/aaa/user-alice/#> show
Status              : Enabled
Name                : alice
Role                : guest
Hash                :
$6$ThcjPrACA$.ptkqbX5nZ72lcLk4MdjnGZSHR1p1LFI3BO1U.ykhPj4JUjOOStySXrYp7BO2CNwVfsw4Qr/
avb2DI7Qd.x6k1
SSH-keys            :
```

Now, let us add two ssh-keys to alice's account:

```
example:/config/aaa/user-alice/#> ssh-user-key import
Paste SSH user key(s) and hit enter twice:
ssh-rsa AAAAB3NzaC ... G6KCGOnv alice@host1
ssh-ecdsa-sha2-nistp256 AAAAE2VjZHNh ... UVpefXTOeObU3pGMLrk= alice@host2
```

As a result there are two built-in accounts in the system: *admin* provided in the factory configuration who has *administrator* role and the newly created *alice* who is a *guest*.

```
example:/config/aaa/user-alice/#> end
example:/config/aaa/#> show
User                Role           SSH-key   ENA
admin               administrator  NO        YES
alice               guest          YES       YES
```

This settings will be applied upon calling `leave` command.

Removing admin account

It is possible to disable or even remove the default built-in *admin* account. To do so create a new account with administrator role and log-in using it to disable the default *admin* account. **Note:** a user cannot disable or remove themselves.

In example below let us create a new administrator *bob* and disable the default *admin*.

When logged-in as *admin*:

```
example:/config/aaa/#> user bob
example:/config/aaa/user-bob/#>
```

```
example:/config/aaa/user-bob/#>
example:/config/aaa/user-bob/#> password secret NewPassword
example:/config/aaa/user-bob/#> end
example:/config/aaa/#> show
User                Role           SSH-key  ENA
admin               administrator  NO       YES
bob                 administrator  NO       YES
example:/config/aaa/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> cp run start
example:/#> exit
```

When logged-in as *bob*:

```
example:#> configure aaa
example:/config/aaa/#> user admin
example:/config/aaa/user-admin/#>no enable
Disabling user admin
example:/config/aaa/user-admin/#> end
example:/config/aaa/#> show
User                Role           SSH-key  ENA
admin               administrator  NO        NO
bob                 administrator  NO       YES
zero-12-34-50:/config/aaa/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
```

This way the only remaining built-in account with administrator role is *bob*.

4.1.2. AAA - Login with Local Database HowTo

About

This document aims to show how to set up an AAA method in the system with Local Database, and how it is connected to the login service.

For an overview of all available configuration settings for AAA refer to the configuration guide:

- AAA Configuration Guide

Introduction

Configure

This section will cover how to configure a Local Database AAA method and how to connect it to the login service.

Local Database

Local user databases are useful for storing authentication credentials with no need for any external infrastructure. The lists consist of username and pass- word pairs.

When a local database is created, a numeric ID is associated with it. Additionally, a description string may be associated with the instance to make it easier to remember its purpose, e.g., ”maintainers” for a local database of system maintainers.

The following example shows how to create a local-db method:

```
example:/#> configure
example:/config/#> aaa
example:/config/aaa/#> local-db 1
Creating new local db 1
example:/config/aaa/local-db-1/#> description maintainers
example:/config/aaa/local-db-1/#> username alice password AliceSecreT
example:/config/aaa/local-db-1/#> username bob hash $1$HTwvtaTw$l5X0AkyuFD1zqHK.QWr/x0
example:/config/aaa/local-db-1/#> end
example:/config/aaa#/>
```

Notes on the configuration:

- The default type for local databases is ‘hash’. It only accepts passwords that are stored as hashes.
- It is possible, as illustrated by user ‘bob’ in the example above, to enter hashes directly.
- It is possible to store passwords in clear-text. In the example above, the command `local-db 1 plain` should be entered instead to create a local-db of type ‘plain’. In that case, using the keyword `hash` is not allowed and ‘bob’ must enter his clear-text password, `username bob password BobSeCret`.

4.1.3. AAA - Login with Centralized Authentication HowTo

About

This document aims to show how to set up an AAA method in the system with centralized authentication, and how it is connected to the login service.

For an overview of all available configuration settings for AAA refer to the configuration guide:

- AAA Configuration Guide

Introduction

Centralized authentication can be used to simplify account administration, this holds especially true when networks grow in size. Instead of manually adding each user and its credentials to each device, the device itself can be instructed to ask a remote server if the provided user information is valid and should provide access.

Support exist in the system for three different types of remote authentication protocols, **RADIUS** (Remote Authentication Dial-IN User Service), **TACACS+** (Terminal Access Controller Access-Control System Plus), and **LDAP** (Lightweight Directory Access Protocol).

RADIUS and TACACS+ are client/server protocols that host a database of usernames and associated credentials on the server. The client forwards provided credentials to the server to verify if they are valid.

LDAP is a directory-access protocol used to look up and authenticate users against a central directory service, such as OpenLDAP or Active Directory®. The device binds to the LDAP server using a service account (bind DN) and searches for the user entry to authenticate.

Currently the device can only act as a client, no support exist to serve as a server for any of the protocols.

Configure

This section will cover how to configure a remote server AAA method and how to connect it to the login service.

Remote Server

When a central authentication server is defined in the system, a numeric ID is provided that will be associated with it. This ID will be used when mapping this specific method to a service with AAA support, e.g. the login service. The following examples shows how to create `remote-server` method using unencrypted RADIUS and encrypted RADIUS (peap-mschapv2):

```
example:/#> configure
example:/config/#> aaa
example:/config/aaa/#> remote-server 1
Creating new remote server 1
example:/config/aaa/remote-server-1/#> password RADiuSseCret
example:/config/aaa/remote-server-1/#> address 192.168.1.1
example:/config/aaa/remote-server-1/#> type radius
example:/config/aaa/remote-server-1/#> message-authenticator false
example:/config/aaa/remote-server-1/#> auth-port 1812
example:/config/aaa/remote-server-1/#> end
example:/config/aaa/#>
```

```
example:/#> configure
example:/config/#> aaa
```

```
example:/config/aaa/#> remote-server 1
Creating new remote server 1
example:/config/aaa/remote-server-1/#> password RADiuSseCret
example:/config/aaa/remote-server-1/#> address 192.168.1.1
example:/config/aaa/remote-server-1/#> type radius
example:/config/aaa/remote-server-1/#> encryption-type peap-mschapv2
example:/config/aaa/remote-server-1/#> ca-cert servercertlabel
example:/config/aaa/remote-server-1/#> auth-port 1812
example:/config/aaa/remote-server-1/#> end
example:/config/aaa/#>
```

Notes on the configurations:

- The provided `address` is the location of the RADIUS server, with its associated `password` required for access to the server.
- The `type` of remote server is specified as a RADIUS server. The `auth-port` is also configured to the RADIUS default port 1812.
- The available `encryption-type` of remote server depends on the protocol. Classical and unencrypted authentication can be specified with encryption-type `none`.
- Depending on protocol, server and client node authentication can be done with certificates/pfx:es. These packages are imported with the normal certificate function and referenced trough their label.
- Validation of the RADIUS protocol attribute “message authenticator” can be turned off for non-encrypted RADIUS. It is mandatory for RADIUS that uses EAP. Default is on. You can disable validation when working with less secure/older RADIUS servers.

In case the desired method is to use TACACS+ instead, simply change the `type` and the `auth-port`:

```
example:/config/aaa/#> remote-server 1
example:/config/aaa/remote-server-1/#> type tacacs
example:/config/aaa/remote-server-1/#> auth-port 49
example:/config/aaa/remote-server-1/#> end
example:/config/aaa/#>
```

LDAP

To use LDAP, set the `type` to `ldap`. Only one LDAP server entry can be configured, with a primary address and an optional secondary address for failover. Unlike RADIUS and TACACS+, LDAP uses a bind DN and password (service account) to search the directory, rather than forwarding user credentials directly to the server.

> **WARNING**
>
> Unencrypted LDAP transmits all data — including user passwords — in plain text over the network. Use `ldaps` or `ldap-starttls` encryption in production environments.

The following example shows how to configure a basic unencrypted LDAP server against an Active Directory® domain:

```
example:/#> configure
example:/config/#> aaa
```

```
example:/config/aaa/#> remote-server 0
Creating new remote server 0
example:/config/aaa/remote-server-0/#> type ldap
example:/config/aaa/remote-server-0/#> description adserver
example:/config/aaa/remote-server-0/#> no connection-security
example:/config/aaa/remote-server-0/#> primary-address 192.168.1.10
example:/config/aaa/remote-server-0/#> secondary-address 192.168.1.11
example:/config/aaa/remote-server-0/#> no auth-port
example:/config/aaa/remote-server-0/#> bind-dn cn=ldap-reader,ou=service-accounts,dc=corp,dc=com
example:/config/aaa/remote-server-0/#> bind-password S3rviceAccP@ss
example:/config/aaa/remote-server-0/#> base-dn dc=corp,dc=com
example:/config/aaa/remote-server-0/#> user-id-mapping sAMAccountName
example:/config/aaa/remote-server-0/#> no role-base-dn
example:/config/aaa/remote-server-0/#> search-filter (objectClass=user)
example:/config/aaa/remote-server-0/#> role-map administrator NetworkAdmins
example:/config/aaa/remote-server-0/#> role-map operator NetworkOps
example:/config/aaa/remote-server-0/#> end
example:/config/aaa/#>
```

If we want to use a FQDN we could configure something like:

```
example:/config/dns/#> show
ADDRESS          NAME              FQDN
192.168.1.10                       myserver.com
```

To use LDAP over SSL (LDAPS) or with STARTTLS, set the `connection-security` accordingly and supply a CA certificate label. Optionally, a client certificate can be provided for mutual TLS authentication:

```
example:/config/aaa/#> remote-server 0
example:/config/aaa/remote-server-0/#> type ldap
example:/config/aaa/remote-server-0/#> description adserver
example:/config/aaa/remote-server-0/#> connection-security ldaps
example:/config/aaa/remote-server-0/#> ciphers-tls1.2 TLS_CHACHA20_POLY1305_SHA256
example:/config/aaa/remote-server-0/#> ciphers-tls1.3 TLS_AES_128_GCM_SHA256
example:/config/aaa/remote-server-0/#> elliptic-curve-only Off
example:/config/aaa/remote-server-0/#> ca-cert my-ca-cert
example:/config/aaa/remote-server-0/#> no client-cert my-client-cert
example:/config/aaa/remote-server-0/#> no revocation
example:/config/aaa/remote-server-0/#> primary-address ldap.corp.com
example:/config/aaa/remote-server-0/#> auth-port 10636
example:/config/aaa/remote-server-0/#> bind-dn cn=ldap-reader,ou=service-accounts,dc=corp,dc=com
example:/config/aaa/remote-server-0/#> bind-password S3rviceAccP@ss
example:/config/aaa/remote-server-0/#> base-dn dc=corp,dc=com
example:/config/aaa/remote-server-0/#> user-id-mapping sAMAccountName
example:/config/aaa/remote-server-0/#> search-filter (objectClass=user)
example:/config/aaa/remote-server-0/#> end
example:/config/aaa/#>
example:/config/aaa/#> credential-cache On
example:/config/aaa/#> credential-cache-expiry 7d
```

When using a certificate it is important that the `primary-address` is written in the certificate's `SAN` field. The certificate might have one or more IP addresses or a FQDN's in the `SAN` field and one of them should be used as a primary address for the LDAP server.

For OpenLDAP, use `uid` as the `uid-mapping` attribute and an appropriate `base-dn` and `search-filter`:

```
example:/config/aaa/#> remote-server 0
example:/config/aaa/remote-server-0/#> type ldap
example:/config/aaa/remote-server-0/#> primary-address 10.0.0.5
```

```
example:/config/aaa/remote-server-0/#> bind-dn cn=reader,dc=example,dc=com
example:/config/aaa/remote-server-0/#> bind-password ReaderPass
example:/config/aaa/remote-server-0/#> base-dn ou=users,dc=example,dc=com
example:/config/aaa/remote-server-0/#> role-base-dn ou=groups,dc=example,dc=com
example:/config/aaa/remote-server-0/#> user-id-mapping uid
example:/config/aaa/remote-server-0/#> search-filter (objectClass=inetOrgPerson)
example:/config/aaa/remote-server-0/#> end
example:/config/aaa/#>
```

Notes on the LDAP configuration:

- `ciphers-tls1.2` List of TLS 1.2 cipher suites to include in cipher negotiation between clients and servers.
- `ciphers-tls1.3` List of TLS 1.3 cipher suites to include in cipher negotiation between clients and servers.
- `elliptic-curve-only` Enable or disable use of elliptic curve for key exchange.
- `primary-address` (required) and `secondary-address` (optional fallback) replace the `address` command used for RADIUS and TACACS+. When using a CA certificate, the primary address must be in the certificate. To test making a client certificate we can created a certificate using the LDAP server certificate:
  1. Create client key and a certificate request (no password and return on all questions)
     `openssl req -newkey rsa:2048 -days 365000 -nodes -keyout client-key.pem -out client-req.pem`
  2. Create client certificate
     `openssl x509 -req -in client-req.pem -days 365000 -CA ldap.crt -CAkey ldap.key -set_serial 01 -out client-cert.pem`
  3. Check we got the right ISSUER with:
     `openssl x509 -in client-cert.pem -noout -text`
  4. Check CA can validate client with:
     `openssl verify -CAfile ldap.crt client-cert.pem`
  5. Import the client and CA certificates into WeOS5.
- `bind-dn` and `bind-password` are the credentials of the service account used to search the directory. The service account should have read-only access to the user entries.
- `base-dn` defines the starting point in the directory tree for user lookups. e.g. start searching from this point.
  - `role-base-dn` (optional) defines the DN suffix that a role-map's remote LDAP role is combined with when matching group membership. If `role-base-dn` is set to ***ou=groups,dc=example,dc=com*** and a role-map remote role is ***westermo_admin***, the user must be a member of the LDAP group with DN ***CN=westermo_admin,ou=groups,dc=example,dc=com***. Without `role-base-dn`, matching is done by CN only, regardless of its position in the LDAP tree.
  - `user-id-mapping` specifies which LDAP attribute holds the username. Use `uid` for OpenLDAP and `sAMAccountName` for Active Directory® (default).
- `role-base-dn` (optional) defines the DN that a role-map remote role must have. If role-base-dn is specified as ***ou=groups,dc=example,dc=com*** and the role-map remote role is specified as ***westermo_admin*** the users must be a memberOf ***westermo_admin/groups/example.com*** in the LDAP server.

- `uid-mapping` specifies which LDAP attribute holds the username. Use `uid` for OpenLDAP and `sAMAccountName` for Active Directory® (default).
- `search-filter` narrows the search to matching entries. Parentheses are added automatically if omitted.
- `role-map` maps a local device role (e.g. `administrator`, `operator`) to an LDAP group DN. Multiple role maps can be configured. Users not matched by any role map will be denied access.
- For `ldaps`, the default port is 636. For `ldap-starttls` and plain `ldap`, the default port is 389.
- When using `ldaps` or `ldap-starttls`, a CA certificate (`ca-cert`) must be imported into the PKI store and referenced by its label. Optionally, a client certificate (`client-cert`) can be configured for mutual TLS. The CA certificate is typically the LDAP servers public certificate.
- Only one LDAP server entry can be configured, with a primary and an optional secondary address for failover. LDAP servers cannot be added to server groups. The remote-server should be added to an authentication chain e.g.:

```
example:/config/aaa/#> auth-chain
example:/config/aaa/auth-chain/#> show
Order          : server-0
Local fallback : always
On-reject      : next
```

Troubleshoot LDAP with `ldap-debug`

The `ldap-debug` command tests the first LDAP server in the AAA authentication chain. It uses the server's configured address, port, encryption settings, bind DN, password, base DN, search filter, UID-mapping attribute, and certificates.

From the `aaa` exec context, run the command with an optional LDAP username:

```
example:/#> aaa
example:/aaa/#> ldap-debug [USERNAME]
```

The command checks TCP connectivity, the LDAP connection and RootDSE, the configured bind credentials, and the configured base DN search. When a username is supplied, it also searches for the user and displays its `memberOf` groups to help verify the WeOS role mapping. For StartTLS and LDAPS, the command performs additional TLS certificate checks when a CA certificate is configured.

Examples:

```
example:/aaa/#> ldap-debug
example:/aaa/#> ldap-debug alice
```

Troubleshoot LDAP with `query-ldap-server`

The `query-ldap-server` command makes it possible to query the configured LDAP server to test connectivity. Bind credentials and uid attributes are taken from the configured LDAP server.

```
example:/aaa/#> query-ldap-server
example:/aaa/#>
Syntax:
  query-ldap-server [user <USERNAME>] [filter <FILTER>] [base <BASE-DN>] [scope <base|one|sub>] [ATTR ...]
```

```
Example:
  query-ldap-server user jdoe
  query-ldap-server user jdoe memberOf
  query-ldap-server user jdoe filter "(memberOf=cn=admins,dc=corp,dc=com)"
  query-ldap-server filter "(cn=John*)" cn memberOf
```

This tool is used to send queries when we know we have a valid connection to the LDAP server.

Server Group

If multiple different remote servers are utilised for redundancy purposes (both servers have identical user databases and are of the same type) they can be added to a server group.

For `tacacs` and `radius` groups the behavior is:

- If the device gets a response from one server (accept or reject), it will not try the other server(s) in the server group.
- If the request sent to one server times out, it tries to send the request to the next server. This goes on until a response is received or until all servers in the groups have been tried.
- A current limitation for radius servers. If multiple radius servers in a radius group contain different settings for message-authenticator, only the setting from the first server in the server group will be used.

For `encrypted-radius` groups the behavior is:

- If the request gets accepted by one server, it will not try the other server(s) in the server group.
- If the request times out or is rejected, it tries to send the request to the next server. This goes on until a accept is received or until all servers in the groups have been tried.

When a server group is defined in the system, a numeric ID is provided that will be associated with it. This ID will be used when mapping this specific method to a service with AAA support, e.g. the login service. Creation of a `server-group` is carried out as follows:

```
example:/#> configure
example:/config/#> aaa
example:/config/aaa/#> remote-server 1
Creating new remote server 1
example:/config/aaa/remote-server-1/#> password RADiuSseCret
example:/config/aaa/remote-server-1/#> address 192.168.1.1
example:/config/aaa/remote-server-1/#> type radius
example:/config/aaa/remote-server-1/#> auth-port 1812
example:/config/aaa/remote-server-1/#> end
example:/config/aaa/#> remote-server 2
Creating new remote server 2
example:/config/aaa/remote-server-2/#> password RADiuSseCret2
example:/config/aaa/remote-server-2/#> address 10.0.1.1
example:/config/aaa/remote-server-2/#> type radius
example:/config/aaa/remote-server-2/#> auth-port 1812
example:/config/aaa/remote-server-2/#> end
example:/config/aaa/#> server-group 1
Creating new server group 1
example:/config/aaa/server-group-1/#> server 1,2
example:/config/aaa/server-group-1/#> type tacacs
example:/config/aaa/server-group-1/#> end
```

Attach to login service

In order to configure the login service to authenticate users logging in against a `remote-server`, the ID of the created method needs to be supplied in the following manner:

```
example:/config/aaa/#> auth-chain
example:/config/aaa/auth-chain/#> order server-1
example:/config/aaa/auth-chain/#> end
example:/config/aaa/#>
```

Any login attempt to the device will try to authenticate against the configured server. If the request sent to the server results in reject or timeout, the device falls back to using the built-in accounts (e.g. the admin account).

4.2. Ethernet/LAN Ports

4.2.1. Ethernet Port Speed/Duplex Howto

```
 Host-A                                       Host-B
.-------.                                    .-------.
|       |                                    |       |
|       |                                    |       |
|   1   |                                    |   1   |
'---+---'                                    '---+---'
    |                                            |
    +--------------------------------------------+
```

Typical scenarios

Default settings aim for plug and play

The default settings for speed-duplex aims for plug and play. The examples below describe the behaviour when connecting two WeOS units, but likely they also apply when connecting a WeOS unit with some other unit.

Fixed copper ports

Auto-negotiation will result in 1000-full when connecting two Gbit copper ports.

```
Host-A:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Up     1000-T  1000M-Full  Forwarding      vlan U:1
eth2             Down   1000-T  ----------  No-Link         vlan U:1
 ...
Host-A:/#>
```

Auto-negotiation will result in 100-full when connecting two FE ports or a Gbit with an FE port.

```
Host-A:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Up     1000-T   100M-Full  Forwarding      vlan U:1
eth2             Down   1000-T  ----------  No-Link         vlan U:1
 ...
Host-A:/#>
```

Fibre SFPs

Auto-negotiation will result in 1000-full when connecting two Gbit Fibre SFPs.

```
Host-A:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Up    1000-LX  1000M-Full  Forwarding      vlan U:1
eth2             Down   No-SFP  ----------  No-Link         vlan U:1
 ...
Host-A:/#>
```

When connecting two FE Fibre SFPs, the link will operate in 100-full (without auto-negotiation).

```
Host-A:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Up     100-FX   100M-Full  Forwarding      vlan U:1
eth2             Down   No-SFP  ----------  No-Link         vlan U:1
```

```
 ...
Host-A:/#>
```

If a Gbit and a FE Fibre SFP are connected, the link will not come up.

> **NOTE**
>
> There are many different types of Gbit and Fibre SFP technologies (multi-mode, single mode, BiDi, etc.), thus it is not only the speed which need to match.

Configuring auto-negotiation advertisement

Auto-configuration enables the peers to agree on the best common speed/duplex mode. Assume the peer is using auto-negotiation, and you wish to limit the speed to 10 Mbit/s. Then you could set **speed-duplex 10-full** or **speed-duplex 10-half, 10-full**. The link will come up in *10-full* if the peer supports that mode.

```
Host-A:/#> config
Host-A:/config/#> port eth1
Host-A:/config/port-eth1/#> auto-negotiate
Host-A:/config/port-eth1/#> speed-duplex 10-full,10-half
Host-A:/config/port-eth1/#> leave
Host-A:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Up     1000-T    10M-Full  Forwarding      vlan U:1
eth2             Down   1000-T  ----------  No-Link         vlan U:1
 ...
Host-A:/#>
```

When both peers use auto-negotiation, but have no common mode, the link will not come up.

Trouble-shooting

Only one peer using auto-negotiation

If one side has auto-negotiation enabled while the peer has not, the link is likely to come up with the right speed, but wrong duplex mode.

Consider the case when A-side uses *auto-negotiation*, and B is set to *100-full* and auto-negotiation disabled: The result will be that A sets its port to 100-half (it can determine the speed, but not the duplex mode, see §28.2.3.1 of[1].). B sets link to 100-full. When reading the RMON counters, A is likely to report lots of *late collision* errors, while B is likely so report *rx checksum* errors.

Mismatching fixed configurations

If A and B both disable auto-configuration, they must set the same speed-duplex mode. If the speeds do not match, the link will not come up. If the speed matches, but not the duplex mode (e.g., 100-full and 100-half), the link will come up but the same kind of error as described in the previous section.

[1] IEEE 802.3-2012 Clause 28: “Physical Layer link signalling for Auto-Negotiation on twisted pair”

4.2.2. Port Monitor Basic HowTo

About

This document provides a basic port monitor use-case, and how it is configured using the CLI.

For an overview of all available configuration settings for Monitor refer to the configuration guide:

- Monitor Configuration Guide

Introduction

In some situations, it may be necessary to capture or sniff traffic that enters on the switch. Perhaps it is needed for performing some troubleshooting, or some sort of security monitoring/investigation.

Considering a switch only forwards packages to the port where the destination is located at, in order to sniff traffic on a switch, port monitoring needs to be utilised. This is in stark contrast to say a hub, where all traffic is copied to all ports on the device.

```
.---------.          .---------.          .---------.
|         |          |         |          |         |
|   H1    +----------+   S1    +----------+   H2    |
|         |     eth1 |         | eth2     |         |
'---------'          '----+----'          '---------'
                          | eth3
              |
                |
                     .----+----.
                     |         |
                     | Sniffer |
                     |         |
                     '---------'
```

Figure 1: Example network where the switch S1 is configured to monitor/mirror traffic from H1 and H2 to the sniffer device.

In the example above, we have a switch S1. We are interested in all traffic that ingresses and egresses on eth1 towards or from H1. Further, we are also interested in all traffic that egresses on eth2 towards H2. Therefore, we will set up a port monitor on the switch in order to duplicate the traffic we want to the connected sniffer device located on eth3.

Configuration

Configure the switch to monitor ports eth1 and eth2 and duplicate the packages to eth3.

```
S1:/#> configure
S1:/config/#> monitor 1
S1:/config/monitor-1/#> source port eth1
S1:/config/monitor-1/#> source port eth2 egress
S1:/config/monitor-1/#> destination eth3
S1:/config/monitor-1/#> leave
S1:/#>
```

With this configuration, any egressing or ingressing on eth1 and egressing traffic on eth2, will be duplicated to eth3. Doing this enables our sniffing device to observe the traffic of interest.

Show monitor configuration

To verify that the monitoring instance have been configured correctly we can check its status in the admin context in the following manner.

```
Server:/#> show monitor
Instance: 1
Source(s)      : eth1 inbound | outbound
               : eth2 outbound
Destination(s) : eth3
```

4.2.3. Storm Control HowTo

About

This document provides basic storm control use-cases, and how they are configured using the CLI.

For an overview of all available configuration settings for storm control refer to the port configuration guide:

- Port Configuration Guide

Storm control utilises port error states when a storm is detected which will operatively disable the port, whilst still keeping the port administratively up. For more information regarding port error states, refer to the error state section in the port configuration guide:

- Port Error States

Introduction

A broadcast storm can occur when a device in the network sends excessive broadcast, multicast, or unknown-unicast traffic, or when a loop forms in the network topology. This can quickly congest the link bandwidth and consume switch resources, effectively bringing down the entire network segment.

Storm control works by combining an ingress rate limit with automatic port error-disable. When the ingress rate on a port exceeds the configured threshold for the selected traffic type, the port is placed into an error state, stopping the offending traffic from propagating through the network. The port can either recover automatically after a configurable timeout or require manual intervention to bring it back up.

```
                          .---------.
                          |         |
                          | Server  |
                          |         |
                          '----+----'
                               | eth6
.-------.             .-----+-----.             .-------.
|       |        eth4 |           | eth5        |       |
|  H1   +-------------+    S1     +-------------+  H2   |
|       |             |           |             |       |
'-------'             '-----+-----'             '-------'
                               | eth3
                               |
                            .--.-.
                           ( (    )___
                          (_,  \ ) ,_)
                            '-'--`--'
                        Untrusted Network
```

Figure 1: Switch S1 with an untrusted network segment on eth3 where a broadcast storm could originate.

In the example above, switch S1 connects trusted hosts on eth4 and eth5, a server on eth6, and an untrusted network segment on eth3. A misbehaving device on the untrusted segment could generate a broadcast storm that would flood all other ports. To protect the network, we configure storm control on eth3 so that the port is automatically disabled if excessive broadcast traffic is detected.

Example 1: Storm control with automatic recovery

In this example, we configure an ingress rate limit of 10 Mbit/s for broadcast traffic on eth3 and enable storm control with a 30-second timeout. If the broadcast rate exceeds 10 Mbit/s, the port is placed into an error state and automatically recovers after 30 seconds.

```
S1:/#> configure
S1:/config/#> port eth3
S1:/config/port-eth3/#> rate-limit
S1:/config/port-eth3/rate-limit/#> ingress bps 10M match bc
S1:/config/port-eth3/rate-limit/#> storm-control timeout 30
S1:/config/port-eth3/rate-limit/#> leave
S1:/#>
```

With this configuration, when a broadcast storm on eth3 exceeds 10 Mbit/s the port is placed into an error state. After 30 seconds the error state is cleared and the port returns to normal operation. If the storm is still present, the port will be error-disabled again.

Example 2: Storm control with manual recovery

In some environments it may be preferable to keep the offending port disabled until an operator has investigated the issue. This is done by enabling storm control without a timeout.

Here we protect against both broadcast and unknown-unicast storms on eth3, with an ingress rate limit of 5 Mbit/s.

```
S1:/#> configure
S1:/config/#> port eth3
S1:/config/port-eth3/#> rate-limit
S1:/config/port-eth3/rate-limit/#> ingress bps 5M match bc,u-uni
S1:/config/port-eth3/rate-limit/#> storm-control
S1:/config/port-eth3/rate-limit/#> leave
S1:/#>
```

When the threshold is exceeded the port enters an error state and stays there until the operator manually clears it.

```
S1:/#> port clear-error eth3
S1:/#>
```

Alternatively, the port can be brought back up with:

```
S1:/#> port up eth3
S1:/#>
```

Example 3: Protecting multiple ports

Storm control can be configured on multiple ports at once. In this example, all customer-facing ports (eth1 through eth3) are given an ingress rate limit of 1 Mbit/s for broadcast and multicast traffic, with a 60-second recovery timeout.

```
S1:/#> configure
S1:/config/#> port eth1..eth3
S1:/config/port-eth1..eth3/#> rate-limit
S1:/config/port-eth1..eth3/rate-limit/#> ingress bps 1M match bc,mc
S1:/config/port-eth1..eth3/rate-limit/#> storm-control timeout 60
```

```
S1:/config/port-eth1..eth3/rate-limit/#> leave
S1:/#>
```

Verifying if storm control has triggered, and current configuration

To determine if the cause for the port being down is due to storm control disabling the port, first use the `show port` command in the admin exec context. Any link reporting `Err-Disabled` indicates that a WeOS service has disabled the port.

```
S1:/#> show port
PORT              LINK     TYPE        SPEED  STATE           OWNER/VLAN
eth1              Down      N/A  ————————  No-Link          vlan U:1
eth2              Down      N/A  ————————  No-Link          vlan U:1
eth3              Down    1000-T  ————————  Err-Disabled     vlan U:1
eth4              Up      1000-T  1000M-Full  Forwarding      vlan U:1
eth5              Up      1000-T  1000M-Full  Forwarding      vlan U:1
eth6              Up      1000-T  1000M-Full  Forwarding      vlan U:1
eth7              Down    1000-T  ————————  No-Link          vlan U:1
eth8              Down    1000-T  ————————  No-Link          vlan U:1
eth9              Down    1000-T  ————————  No-Link          vlan U:1
eth10             Down    1000-T  ————————  No-Link          vlan U:1
```

To determine the exact cause for the port being rendered `Err-Disabled`, it is possible to show a detailed view by using `show port PORT`. This view also display the current configuration for the port.

```
S1:/#> show port eth3
eth3
Link            : Down
Oper. status    : Disabled
Admin status    : Enabled
Description     : eth3
Type            : 1000-T
State           : Err-Disabled
Alarm           : N/A
VLAN            : U:1
MAC             : 00:07:7c:53:df:87
Autoneg         : Enabled
Speed           : ——————————
Duplex          :
MDI/MDIX        :
MDI/MDIX status:
Polarity status:
Gbit-cu-master : auto
Fastlink        : Disabled
Flow Control    : Disabled
Prio mode       : tag
Prio            : 0
Ingress limit   : 10000 kbps match bc
Egress limit    : None
Storm Control   : Enabled (timeout 60s)
Shielded mode   : Unshielded
PHY power mode : Low power (CLASS B)
Bridge group    : Disabled

Error State: Err-Disabled (effective)
```

```
STATE            SERVICE          REASON
Err-Disabled     Storm Control    Storm detected
```

> **NOTE**
>
> Enabling rate-limit in your network may affect the performance of protocols and redundancy services dependent on the flow of management traffic, such as FRNT, MRP, and RSTP. If management traffic connected to these services start getting rate-limited their performance can not be guaranteed.

Disabling storm control

To remove storm control and the ingress rate limit from a port:

```
S1:/#> configure
S1:/config/#> port eth3
S1:/config/port-eth3/#> rate-limit
S1:/config/port-eth3/rate-limit/#> no storm-control
S1:/config/port-eth3/rate-limit/#> no ingress
S1:/config/port-eth3/rate-limit/#> leave
S1:/#>
```

4.2.4. Health Monitor HowTo

About

This document provides basic Health Monitor use-cases, and how the feature is configured.

For an overview of all available configuration settings for Health Monitor, refer to the port configuration guide:

- Port Configuration Guide

Health Monitor utilises port error states when the health of a link is considered to have deteriorated. When the criteria configured with Health Monitor are met, it sets the port to `Err-Disabled`. For more information regarding port error states, refer to the error state section in the port configuration guide:

- Port Error States

Introduction

When individual components degrade over time, small and localized faults may escalate into widespread network disruptions. These kind of faults may not only have a large impact on the whole network, but they can also be very difficult to isolate, and time-consuming to troubleshoot. Health Monitor provides a centralized capability to continuously monitor the health of physical network connections.

Health Monitor offer the ability to track link state changes (`UP/DOWN`) of a port. If the number of state changes exceed a given threshold during a given time-window, it places the port in `Err-Disabled`. If a timeout is configured, it will automatically clear the error. If no timeout is given, the error has to be manually cleared.

One example use case for this feature is to configure it on a port that is part of a ring. A faulty SFP or a faulty cable between two devices could potentially disrupt the whole ring. A blocked port may start forwarding if the link between two devices goes down in order to recover. If the link then goes up again, the ring protocol may start forwarding on the link that was down and put the other link in blocking mode. A fault could trigger this kind of reconvergence over and over again, and cause the network to never reach a stable state.

Examples

In this section we show different example use-cases of how one could configure Health Monitor.

Health Monitor with manual recovery

In this example, we configure Health Monitor `state-changes` with a threshold of 10 and 60 seconds as time-window.

```
example:/#> configure
example:/config/#> port eth2
example:/config/port-eth2/#> health-monitor
example:/config/port-eth2/health-monitor/#> state-changes threshold 10 window 60
example:/config/port-eth2/health-monitor/#> leave
example:/#>
```

If Health Monitor observes 10 or more state-changes on port eth2 within 60 seconds, the port will be placed in `Err-Disabled`. Since we gave no timeout parameter, the error will have to be manually cleared by either `port clear-error eth2` or by administratively forcing the port up with `port up eth2`.

Health Monitor with automatic recovery

Here we configure Health Monitor `state-changes` with a threshold of 5 and 120 seconds as time-window with a 20 minute timeout.

```
example:/#> configure
example:/config/#> port eth2
example:/config/port-eth2/#> health-monitor
example:/config/port-eth2/health-monitor/#> state-changes threshold 5 window 120 timeout 20
example:/config/port-eth2/health-monitor/#> leave
example:/#>
```

If there are 5 or more state-changes on port eth2 within 120 seconds, the port will be placed in `Err-Disabled`. As we set a 20 minute timeout, the error will be cleared automatically after 20 minutes. It is still possible to manually clear the error state if needed.

Protecting multiple ports

It is possible to configure Health Monitor for multiple ports at once. Here we configure Health Monitor `state-changes` for the port range `eth0..eth7`.

```
example:/#> configure
example:/config/#> port eth0..eth7
example:/config/port-eth0..eth7/#> health-monitor
example:/config/port-eth0..eth7/health-monitor/#> state-changes threshold 5 window 120 timeout 20
example:/config/port-eth0..eth7/health-monitor/#> leave
example:/#>
```

Verifying if Health Monitor has triggered an error state

To the determine if Health Monitor have placed the port in `Err-Disabled` or not, we first use `show port` command in the admin exec context. Any link reporting `Err-Disabled` indicates that a service has disabled the port.

```
example:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Down      N/A  ----------  No-Link         vlan U:1
eth2             Down      N/A  ----------  No-Link         vlan U:1
eth3             Down    1000-T  ----------  Err-Disabled    vlan U:1
eth4             Up      1000-T  1000M-Full  Forwarding      vlan U:1
 ...
example:/#>
```

To determine the exact cause for the port being rendered `Err-Disabled`, we can show a detailed view by using `show port PORT`. This view also display the current configuration for the port.

```
example:/#> show port eth3
eth3
Link           : Down
Oper. status   : Disabled
Admin status   : Enabled
Description    : eth3
Type           : 1000-T
State          : Err-Disabled
 ...
State Changes  : threshold 5 window 120 timeout 20
Shielded mode  : Unshielded
PHY power mode : Low power (CLASS B)
```

```
Bridge group   : Disabled

Error State: Err-Disabled (effective)
STATE             SERVICE          REASON
Err-Disabled      Health Monitor   Excessive link state changes detected
```

Here we can see that Health Monitor was the service that place the port in `Err-Disabled`, and that the reason was “Excessive link state changes detected”.

Disabling Health Monitor

To disable Health Monitor `state-changes` you can either do `no health-monitor` in the port config context or `no state-changes` in the health-monitor context:

```
example:/#> configure
example:/config/#> port eth2
example:/config/port-eth2/#> no health-monitor
example:/config/port-eth2/#> leave
example:/#>
```

```
example:/#> configure
example:/config/#> port eth2
example:/config/port-eth2/#> health-monitor
example:/config/port-eth2/health-monitor/#> no state-changes
example:/config/port-eth2/health-monitor/#> leave
example:/#>
```

4.3. Bridging/Switching

4.3.1. VLAN HowTo

Introduction

This document outlines a couple of common VLAN use-cases and details how to set up VLANs. VLANs enables isolating groups of users and network devices in a shared LAN infrastructure.

The first use-case segments the LAN ports on a single switch into multiple VLANs.

```
.---------------------.
|                     |
|    Switch/Router    |
|                     |
'--+-+-+-------+-+-+--'
   | | |       | | |
   | | |       | | |
   VLAN 1      VLAN 2
```

Figure 1: VLANs on single switch.

Th second use-case exemplifies how VLANs can span multiple switches by use of VLAN trunk lines.

```
.---------------------.            .---------------------.
|                     |            |                     |
|    Switch/Router    |            |    Switch/Router    |
|                     |            |                     |
'--+-+-+----+-+-+---+-'            '-+---+-+-+----+-+-+--'
   | | |    | | |   |   VLAN Trunk   |   | | |    | | |
   | | |    | | |   '----------------'   | | |    | | |
   VLAN 1   VLAN 2                        VLAN 1   VLAN 2
```

Figure 2: VLANs spanning multiple switches.

> **NOTE**
>
> The guide below assumes that your device has a *factory default* configuration and that you can access the CLI via its console port.

Use-case 1: Single Switch

```
.---------------------.
|                     |
|    Switch/Router    |
|                     |
| VLAN 1      VLAN 2  |
'--+-+-+-------+-+-+--'
 | | |       | | |
 | | |       | | |
```

Figure 3: VLAN and port view.

We wish to segment the Ethernet ports on our switch into two VLANs; VLAN 1 and 2. In Figure 3, each VLAN is assigned three Ethernet ports.

```
    .--------------------.
    |                    |
    |     Switch/Router  |
    |                    |
    |  vlan1       vlan2 |
    '----+-----------+----'
      .1 |           | .1
       |           |
192.168.1.0/24 |           |
---+-----+-----+--+--      |
 |     |        |        | 192.168.2.0/24
         --+--+-----+-----+----
           |        |     |
```

Figure 4: Network interface view.

Adding a VLAN also creates a management interface for each VLAN; *vlan1* and *vlan2*. Here we have assigned IP addresses *192.168.1.1/24* and *192.168.2.1/24*, respectively. This means we can manage the device remotely using SSH, HTTP/HTTPS, SNMP, etc., via hosts on VLAN 1 and 2.

On switches with IP forwarding[1], we can now also enable *routing* of traffic between VLAN 1 and 2.

Logging in to your switch

Log in using the *admin* and default password (*admin*).

> **NOTE**
>
> Remember to change the admin password before deploying to production!

```
example-41-5a-80 login: admin
Password: <i>Password not echoed</i>
.--.--.--.____.____.______.____.___._______.____.
|  |  |  |  -__|__  __|_     _|  -__|   _|        |  _  |
|_______|____|____| |___| |____|__| |__|__|__|____|
                                        www.westermo.com

\\/ Westermo WeOS v5.3.x master/5.2.0-rc1-964-gbdd6c89 -- Jul 24 11:58 CEST 2018
Type: 'help' for help with commands, 'exit' to logout or leave a context.

example-41-5a-80:/#>
```

Inspect the current VLAN and network interface status using the *show vlan* and *show iface* CLI commands. In factory default configuration, all Ethernet ports are assigned to VLAN 1 (untagged). The example below shows the CLI status assuming all Ethernet ports are disconnected; *vlan1* interface will have *operational status* DOWN.

```
example-41-5a-80:/#> show vlan
VID  NAME             UNTAGGED/TAGGED
   1 vlan1            U:ALL
                      T:

example-41-5a-80:/#> show iface
INTERFACE NAME    OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             DOWN  NONE                            00:07:7c:41:5a:81
example-41-5a-80:/#>
```

[1] Assuming the switch has software level *Extended*.

Create a Second VLAN

To configure a second VLAN, here VLAN 2, enter the *Global Configuration* context and list VLAN configuration overview.

```
example-41-5a-80:/#> configure
example-41-5a-80:/config/#> show vlan
VID  NAME              UNTAGGED/TAGGED
   1 vlan1             U:ALL
                       T:

example-41-5a-80:/config/#>
```

Here all ports are assigned to VLAN 1 (untagged).

We will now create a second VLAN (VLAN 2), and assign it a set of ports (*eth15* to *eth28*). We assign all ports *untagged*, assuming they will all be *access ports* where hosts in VLAN 2 can connect.

```
example-41-5a-80:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
example-41-5a-80:/config/vlan-2/#> untagged eth15..eth28
Moving untagged port eth15 from vid 1 to vid 2.
Moving untagged port eth16 from vid 1 to vid 2.
Moving untagged port eth17 from vid 1 to vid 2.
Moving untagged port eth18 from vid 1 to vid 2.
Moving untagged port eth19 from vid 1 to vid 2.
Moving untagged port eth20 from vid 1 to vid 2.
Moving untagged port eth21 from vid 1 to vid 2.
Moving untagged port eth22 from vid 1 to vid 2.
Moving untagged port eth23 from vid 1 to vid 2.
Moving untagged port eth24 from vid 1 to vid 2.
Moving untagged port eth25 from vid 1 to vid 2.
Moving untagged port eth26 from vid 1 to vid 2.
Moving untagged port eth27 from vid 1 to vid 2.
Moving untagged port eth28 from vid 1 to vid 2.
example-41-5a-80:/config/vlan-2/#> show
 VLAN ID          : 2
 Name             : vlan2
 Channel          : 0
 Priority         : Disabled
 Untagged         : U:eth15..eth28
 Tagged           : T:
 Forbid           : F:
 IGMP             : Disabled
example-41-5a-80:/config/vlan-2/#> end
example-41-5a-80:/config/#> show vlan

VID  NAME              UNTAGGED/TAGGED
   1 vlan1             U:eth1..eth14
                       T:

   2 vlan2             U:eth15..eth28
                       T:

example-41-5a-80:/config/#> leave Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
example-41-5a-80:/#> copy running-config startup-config
```

That was it! To view the VLAN operational status, we can run the *show iface* command again. As you can see, both VLAN 1 and 2 have the intended ports assigned, and both have operational status DOWN, implying that all ports on the respective VLANs have link down.

```
example-41-5a-80:/#> show iface
INTERFACE NAME    OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             DOWN  NONE                            00:07:7c:41:5a:81
vlan2             DOWN  NONE                            00:07:7c:41:5a:81
example-41-5a-80:/#> show vlan
VID  NAME             UNTAGGED/TAGGED
   1 vlan1            U:eth1..eth14
                      T:

   2 vlan2            U:eth15..eth28
                      T:

example-41-5a-80:/#>
```

Configure IP Settings

We will now configure the intended IP addresses for network interface *vlan1* (192.168.1.1/24) and *vlan2* (192.168.2.1/24). First we inspect the existing interface status and configuration overview.

```
example-41-5a-80:/#> show iface
INTERFACE NAME    OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             DOWN  NONE                            00:07:7c:41:5a:81
vlan2             DOWN  DISABLED                        00:07:7c:41:5a:8f
example-41-5a-80:/#> configure
example-41-5a-80:/config/#> show iface
INTERFACE NAME    ADM.  ADDRESS/LEN         DISTANCE  MTU    MAC ADDRESS
lo                UP    127.0.0.1/8               16  AUTO   AUTO
vlan1             UP    dhcp                       1  AUTO   AUTO
            link-local
vlan2             UP    DISABLED                  16  AUTO   AUTO

example-41-5a-80:/config/#>
```

As you can see, there exist three network interfaces; *lo*, *vlan1* and *vlan2*. *lo* is the loopback interface and is not discussed further here.

First we configure interface *vlan1*, which is the network interface associated with VLAN 1. In the factory default configuration, *vlan1* is configured to (1) acquire an IP address via DHCP, and (2) also to setup a link-local address (range 169.255.0.0/16). These are good defaults, but in this example we remove both these methods and give *vlan1* a static IP address of 192.168.1.1/24.

```
example-41-5a-80:/config/#> iface vlan1
example-41-5a-80:/config/iface-vlan1/#> show
Name          : vlan1
Admin Mode    : Up
MAC Address   : AUTO
IP Addresses  :
  Inet dhcp
    arping    : Enabled
    Client ID : Disabled
  Inet link-local
```

```
MTU           : AUTO (1500)
ICMP Redirect : Sending
Proxy ARP     : Disabled
ARP Notify    : Enabled
ARP Accept    : Enabled
Distance      : 1
example-41-5a-80:/config/iface-vlan1/#> inet static 192.168.1.1/24
example-41-5a-80:/config/iface-vlan1/inet-static-192.168.1.1/#> end
example-41-5a-80:/config/iface-vlan1/#> show
Name          : vlan1
Admin Mode    : Up
MAC Address   : AUTO
IP Addresses  :
  Inet dhcp
    arping    : Enabled
    Client ID : Disabled
  Inet link-local
  Inet static
    192.168.1.1/24
MTU           : AUTO (1500)
ICMP Redirect : Sending
Proxy ARP     : Disabled
ARP Notify    : Enabled
ARP Accept    : Enabled
Distance      : 1
example-41-5a-80:/config/iface-vlan1/#> no inet dhcp
example-41-5a-80:/config/iface-vlan1/#> no inet link-local
example-41-5a-80:/config/iface-vlan1/#> show
Name          : vlan1
Admin Mode    : Up
MAC Address   : AUTO
IP Addresses  :
  Inet static
    192.168.1.1/24
MTU           : AUTO (1500)
ICMP Redirect : Sending
Proxy ARP     : Disabled
ARP Notify    : Enabled
ARP Accept    : Enabled
Distance      : 1
example-41-5a-80:/config/iface-vlan1/#> end
example-41-5a-80:/config/#> show iface
INTERFACE NAME    ADM.  ADDRESS/LEN         DISTANCE  MTU    MAC ADDRESS
lo                UP    127.0.0.1/8               16  AUTO   AUTO
vlan1             UP    192.168.1.1/24             1  AUTO   AUTO
vlan2             UP    DISABLED                  16  AUTO   AUTO

example-41-5a-80:/config/#>
```

Next, we configure interface *vlan2*, which is the newly created network interface associated with VLAN 2. New network interfaces have no IP setting by default. Here we assign it address 192.168.2.1/24.

```
example-41-5a-80:/config/#> iface vlan2
example-41-5a-80:/config/iface-vlan2/#> show
Name          : vlan2
Admin Mode    : Up
MAC Address   : AUTO
```

```
IP Addresses  :
  No addresses configured.
MTU           : AUTO (1500)
ICMP Redirect : Sending
Proxy ARP     : Disabled
ARP Notify    : Enabled
ARP Accept    : Enabled
Distance      : 16
example-41-5a-80:/config/iface-vlan2/#> inet static 192.168.2.1/24
example-41-5a-80:/config/iface-vlan2/inet-static-192.168.2.1/#> end
example-41-5a-80:/config/iface-vlan2/#> show
Name          : vlan2
Admin Mode    : Up
MAC Address   : AUTO
IP Addresses  :
  Inet static
    192.168.2.1/24
MTU           : AUTO (1500)
ICMP Redirect : Sending
Proxy ARP     : Disabled
ARP Notify    : Enabled
ARP Accept    : Enabled
Distance      : 16
example-41-5a-80:/config/iface-vlan2/#> end
example-41-5a-80:/config/#> show iface
INTERFACE NAME    ADM.  ADDRESS/LEN         DISTANCE  MTU    MAC ADDRESS
lo                UP    127.0.0.1/8               16  AUTO   AUTO
vlan1             UP    192.168.1.1/24             1  AUTO   AUTO
vlan2             UP    192.168.2.1/24            16  AUTO   AUTO

example-41-5a-80:/config/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example-41-5a-80:/#> copy running-config startup-config
example-41-5a-80:/#> show iface
INTERFACE NAME    OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             DOWN  192.168.1.1/24      static      00:07:7c:41:5a:81
vlan2             DOWN  192.168.2.1/24      static      00:07:7c:41:5a:8f
example-41-5a-80:/#>
```

Operational Status

In the examples we have show, the operational status of the VLANs and associated network interfaces has been stated as DOWN. The reason is that all ports in these VLANs have had *link down*, as all ports have been disconnected. The criteria for operational status for (VLAN) network interfaces, VLANs and Ethernet ports are as follows:

- Network interfaces: A (VLAN) network interface is operationally UP if it is *Enabled* **and** if its associated VLAN is operationally UP (see below).
- VLAN: A VLAN is operationally UP if at least one of its associated (Ethernet) ports are operationally UP (see below).
- Ethernet Port: An Ethernet port is operationally UP if it is *Enabled* **and** if it has determined physical link UP.

Thus, before any Ethernet cable has been connected on the switch, the operational status of network interfaces and ports will all be DOWN (see below).

```
example-41-5a-80:/#> show iface
INTERFACE NAME    OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             DOWN  192.168.1.1/24      static      00:07:7c:41:5a:81
vlan2             DOWN  192.168.2.1/24      static      00:07:7c:41:5a:8f

example-41-5a-80:/#> show port
PORT    LINK     TYPE       SPEED  STATE           VLAN
eth1    Down   No-SFP  ———————  No-Link         U:1
eth2    Down   No-SFP  ———————  No-Link         U:1
eth3    Down   No-SFP  ———————  No-Link         U:1
eth4    Down   No-SFP  ———————  No-Link         U:1
eth5    Down   No-SFP  ———————  No-Link         U:1
eth6    Down   No-SFP  ———————  No-Link         U:1
eth7    Down   1000-T  ———————  No-Link         U:1
eth8    Down   1000-T  ———————  No-Link         U:1
eth9    Down   1000-T  ———————  No-Link         U:1
eth10   Down   1000-T  ———————  No-Link         U:1
eth11   Down   No-SFP  ———————  No-Link         U:1
eth12   Down   No-SFP  ———————  No-Link         U:1
eth13   Down   No-SFP  ———————  No-Link         U:1
eth14   Down   No-SFP  ———————  No-Link         U:1
eth15   Down   1000-T  ———————  No-Link         U:2
eth16   Down   1000-T  ———————  No-Link         U:2
eth17   Down   1000-T  ———————  No-Link         U:2
eth18   Down   1000-T  ———————  No-Link         U:2
eth19   Down   No-SFP  ———————  No-Link         U:2
eth20   Down   No-SFP  ———————  No-Link         U:2
eth21   Down   No-SFP  ———————  No-Link         U:2
eth22   Down   No-SFP  ———————  No-Link         U:2
eth23   Down   1000-T  ———————  No-Link         U:2
eth24   Down   1000-T  ———————  No-Link         U:2
eth25   Down   1000-T  ———————  No-Link         U:2
eth26   Down   1000-T  ———————  No-Link         U:2
eth27   Down   No-SFP  ———————  No-Link         U:2
eth28   Down   No-SFP  ———————  No-Link         U:2
```

If we now connect hosts to one port on each VLAN (here ports *eth7* and *eth15*), we will get port UP on these ports. This will in turn make the interfaces *vlan1* and *vlan2* come up.

```
example-41-5a-80:/#> show port
PORT    LINK     TYPE       SPEED  STATE           VLAN
eth1    Down   No-SFP  ———————  No-Link         U:1
eth2    Down   No-SFP  ———————  No-Link         U:1
eth3    Down   No-SFP  ———————  No-Link         U:1
eth4    Down   No-SFP  ———————  No-Link         U:1
eth5    Down   No-SFP  ———————  No-Link         U:1
eth6    Down   No-SFP  ———————  No-Link         U:1
eth7    Up     1000-T  1000M-Full  Forwarding   U:1
eth8    Down   1000-T  ———————  No-Link         U:1
eth9    Down   1000-T  ———————  No-Link         U:1
eth10   Down   1000-T  ———————  No-Link         U:1
eth11   Down   No-SFP  ———————  No-Link         U:1
eth12   Down   No-SFP  ———————  No-Link         U:1
eth13   Down   No-SFP  ———————  No-Link         U:1
eth14   Down   No-SFP  ———————  No-Link         U:1
eth15   Up     1000-T  1000M-Full  Forwarding   U:2
eth16   Down   1000-T  ———————  No-Link         U:2
```

```
eth17   Down   1000-T  ──────────  No-Link         U:2
eth18   Down   1000-T  ──────────  No-Link         U:2
eth19   Down   No-SFP  ──────────  No-Link         U:2
eth20   Down   No-SFP  ──────────  No-Link         U:2
eth21   Down   No-SFP  ──────────  No-Link         U:2
eth22   Down   No-SFP  ──────────  No-Link         U:2
eth23   Down   1000-T  ──────────  No-Link         U:2
eth24   Down   1000-T  ──────────  No-Link         U:2
eth25   Down   1000-T  ──────────  No-Link         U:2
eth26   Down   1000-T  ──────────  No-Link         U:2
eth27   Down   No-SFP  ──────────  No-Link         U:2
eth28   Down   No-SFP  ──────────  No-Link         U:2
example-41-5a-80:/#> show iface
INTERFACE NAME    OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             UP    192.168.1.1/24      static      00:07:7c:41:5a:81
vlan2             UP    192.168.2.1/24      static      00:07:7c:41:5a:8f
example-41-5a-80:/#>
```

Use-case 2: Multiple Switches

This case shows how to extend VLANs over multiple switches. This implies that VLANs will share switches and also cables, here referred to as VLAN trunk cables or VLAN trunks.

```
.--------------------.              .--------------------.
|                    |              |                    |
|    Switch/Router   |              |    Switch/Router   |
|       (Alice)      |              |       (Bob)        |
|                    |              |                    |
'--+-+-+----+-+-+---+-'              '-+---+-+-+----+-+-+--'
 | | |    | | |   |    VLAN Trunk    |   | | |    | | |
 | | |    | | |   '------------------'   | | |    | | |
VLAN 1    VLAN 2      VLAN 1 and 2       VLAN 1   VLAN 2
```

Figure 5: Extending VLANs over Multiple Switches

Figure 5 shows a case with two switches (Alice and Bob) and two VLANs (1 and 2), but the scenario can easily be extended to make use of more switches and VLANs. Additional information:

- IP subnets: We assume that IP subnet 192.168.1.0/24 is used on VLAN 1, and 192.168.2.0/24 is used on VLAN 2, just as in Use Case 1 above.
- Switch *Alice* will be assigned address 192.168.1.1/24 and 192.168.2.1/24 on her interfaces *vlan1* and *vlan2*, just like the single switch in Use Case 1. These addresses are only used for remote management of Alice (SSH, HTTP/HTTPS, SNMP, Ping, etc.) and possibly also for IP forwarding if Alice should act as router.
- Similarly, switch *Bob* will be assigned address 192.168.1.2/24 and 192.168.2.2/24 on his interfaces *vlan1* and *vlan2*.

What is new here, as compared to Use Case 1, is that each switch must configure a VLAN trunk port. This port should carry packets both for VLAN 1 and VLAN 2, implying use of VLAN tagging.

In our examples we have assumed that port eth1 on Alice (and Bob) is configured as the VLAN trunk port. Except from that, the mapping of ports to VLANs is the same here as in Use Case 1.

In the following sub-sections we show the VLAN and network interface configuration of Alice and Bob. Only the necessary steps are shown; for more details, see the corresponding steps in Use Case 1.

Configure VLANs

The CLI commands below are used for configuring VLAN 1 and 2 on switch *Alice*. It is assumed that *Alice* has factory default configuration before configuration.

> **NOTE**
>
> Use the same commands to set up VLANs at *Bob*.

```
alice:/#> configure
alice:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
alice:/config/vlan-2/#> untagged eth15..eth28
Moving untagged port eth15 from vid 1 to vid 2.
Moving untagged port eth16 from vid 1 to vid 2.
Moving untagged port eth17 from vid 1 to vid 2.
Moving untagged port eth18 from vid 1 to vid 2.
Moving untagged port eth19 from vid 1 to vid 2.
Moving untagged port eth20 from vid 1 to vid 2.
Moving untagged port eth21 from vid 1 to vid 2.
Moving untagged port eth22 from vid 1 to vid 2.
Moving untagged port eth23 from vid 1 to vid 2.
Moving untagged port eth24 from vid 1 to vid 2.
Moving untagged port eth25 from vid 1 to vid 2.
Moving untagged port eth26 from vid 1 to vid 2.
Moving untagged port eth27 from vid 1 to vid 2.
Moving untagged port eth28 from vid 1 to vid 2.
alice:/config/vlan-2/#> tagged eth1
alice:/config/vlan-2/#> end
alice:/config/#> vlan 1
alice:/config/vlan-1/#> tagged eth1
alice:/config/vlan-1/#> end
alice:/config/#> show vlan
VID  NAME             UNTAGGED/TAGGED
   1 vlan1            U:eth2..eth14
                      T:eth1

   2 vlan2            U:eth15..eth28
                      T:eth1

alice:/config/#> end
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

Configure IP Settings

```
alice:/#> configure
alice:/config/#> iface vlan1
alice:/config/iface-vlan1/#> inet static 192.168.1.1/24
alice:/config/iface-vlan1/inet-static-192.168.1.1/#> end
alice:/config/iface-vlan1/#> no inet dhcp
alice:/config/iface-vlan1/#> no inet link-local
alice:/config/iface-vlan1/#> end
alice:/config/#> iface vlan2
alice:/config/iface-vlan2/#> inet static 192.168.2.1/24
alice:/config/iface-vlan2/inet-static-192.168.2.1/#> end
alice:/config/iface-vlan2/#> end
```

```
alice:/config/#> show iface

INTERFACE NAME    ADM.  ADDRESS/LEN         DISTANCE  MTU    MAC ADDRESS
lo                UP    127.0.0.1/8               16  AUTO   AUTO
vlan1             UP    192.168.1.1/24             1  AUTO   AUTO
vlan2             UP    192.168.2.1/24            16  AUTO   AUTO

alice:/config/#> end
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

Set up *Bob* in a similar manner, only with different management addresses.

```
bob:/#> configure
bob:/config/#> iface vlan1
bob:/config/iface-vlan1/#> inet static 192.168.1.2/24
bob:/config/iface-vlan1/inet-static-192.168.1.2/#> end
bob:/config/iface-vlan1/#> no inet dhcp
bob:/config/iface-vlan1/#> no inet link-local
bob:/config/iface-vlan1/#> end
bob:/config/#> iface vlan2
bob:/config/iface-vlan2/#> inet static 192.168.2.2/24
bob:/config/iface-vlan2/inet-static-192.168.2.2/#> end
bob:/config/iface-vlan2/#> end
bob:/config/#> show iface

INTERFACE NAME    ADM.  ADDRESS/LEN         DISTANCE  MTU    MAC ADDRESS
lo                UP    127.0.0.1/8               16  AUTO   AUTO
vlan1             UP    192.168.1.2/24             1  AUTO   AUTO
vlan2             UP    192.168.2.2/24            16  AUTO   AUTO

bob:/config/#> end
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
bob:/#>
```

Verifying Connectivity

If you now connect port *eth1* on Alice to port *eth1* on Bob, hosts connected to *VLAN 1 ports* at Alice will be able to communicate with hosts connected to *VLAN 1* ports on Bob (and the same goes for hosts on VLAN 2 ports).

As a basic test you can check that the VLAN trunk ports (eth1) on Alice and Bob come up, and that Alice can *ping* Bob over both VLANs.

> **NOTE**
>
> In this case, port *eth1* is an SFP port at both Alice and Bob, where we have used Gbit Fibre SFPs at both ends. The SFPs must be inserted before booting the switches to be detected.

```
alice:/#> show port
PORT    LINK     TYPE       SPEED  STATE           VLAN
eth1    Up     1000-SX  1000M-Full  Forwarding      T:1,2
eth2    Down    No-SFP  ──────────  No-Link         U:1
eth3    Down    No-SFP  ──────────  No-Link         U:1
eth4    Down    No-SFP  ──────────  No-Link         U:1
```

```
eth5    Down   No-SFP  ──────────  No-Link         U:1
eth6    Down   No-SFP  ──────────  No-Link         U:1
eth7    Down   1000-T  ──────────  No-Link         U:1
eth8    Down   1000-T  ──────────  No-Link         U:1
eth9    Down   1000-T  ──────────  No-Link         U:1
eth10   Down   1000-T  ──────────  No-Link         U:1
eth11   Down   No-SFP  ──────────  No-Link         U:1
eth12   Down   No-SFP  ──────────  No-Link         U:1
eth13   Down   No-SFP  ──────────  No-Link         U:1
eth14   Down   No-SFP  ──────────  No-Link         U:1
eth15   Down   1000-T  ──────────  No-Link         U:2
eth16   Down   1000-T  ──────────  No-Link         U:2
eth17   Down   1000-T  ──────────  No-Link         U:2
eth18   Down   1000-T  ──────────  No-Link         U:2
eth19   Down   No-SFP  ──────────  No-Link         U:2
eth20   Down   No-SFP  ──────────  No-Link         U:2
eth21   Down   No-SFP  ──────────  No-Link         U:2
eth22   Down   No-SFP  ──────────  No-Link         U:2
eth23   Down   1000-T  ──────────  No-Link         U:2
eth24   Down   1000-T  ──────────  No-Link         U:2
eth25   Down   1000-T  ──────────  No-Link         U:2
eth26   Down   1000-T  ──────────  No-Link         U:2
eth27   Down   No-SFP  ──────────  No-Link         U:2
eth28   Down   No-SFP  ──────────  No-Link         U:2
alice:/#> ping count 3 192.168.1.2
Press Ctrl-C to abort PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: seq=0 ttl=64 time=0.715 ms
64 bytes from 192.168.1.2: seq=1 ttl=64 time=0.481 ms
64 bytes from 192.168.1.2: seq=2 ttl=64 time=0.466 ms

--- 192.168.1.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.466/0.554/0.715 ms
alice:/#> ping count 3 192.168.2.2
Press Ctrl-C to abort PING 192.168.2.2 (192.168.2.2): 56 data bytes
64 bytes from 192.168.2.2: seq=0 ttl=64 time=1.142 ms
64 bytes from 192.168.2.2: seq=1 ttl=64 time=0.480 ms
64 bytes from 192.168.2.2: seq=2 ttl=64 time=0.460 ms

--- 192.168.2.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.460/0.694/1.142 ms
alice:/#>
```

4.3.2. Layer-2 QoS HowTo

Introduction

This document describes how to configure switches to prioritise traffic in a switched LAN infrastructure. We refer to this as *layer-2 QoS priority*, or simply *layer-2 priority*. Other sources may refer to this as *class of service (CoS)*.

Consider Figure 1 below. The cloud represents a set of LAN switches and cables, used to interconnect a set of hosts (end devices), here denoted as H1-H9. The cloud constitutes the *LAN infrastructure*, surrounded by hosts connecting at its edge.

```
      H1   H2
       |   |
       .--.-.     H3
      ( (    )__ /
H5--(_,  \ ) ,_)--H4
       '-'--`--'
      / |   |  \
    H6  H7  H8 H9
```

Figure 1: Switched LAN Infrastructure (cloud represents LAN switches).

Then compare this with Figure 2 below, showing a sample physical representation of the same network. The LAN infrastructure consists of four switches (S1-S4), connected in a ring to achieve redundancy. Switches use a layer-2 redundancy protocol such as FRNT or RSTP to ensure proper connectivity within the ring.

```
     .-----------------.      .-----------------.
     |        S1       |      |        S2       |
.----+1               2+------+1               2+----.
|    |  3   4   5   6  |      |  3   4   5   6  |    |
|    '--+---+---+---+--'      '--+---+---+---+--'    |
|       |   |   |   |            |   |   |   |       |
|      H1  H2                   H3  H4               |
|  (VoIP) (SNMP             (VoIP)  (SNMP            | VLAN 1 &
|          Mgr)                      Agent)          | VLAN 10
|                                                    | on Trunk
|                                                    | Links
|    .-----------------.      .-----------------.    |
|    |        S3       |      |        S4       |    |
'----+1               2+------+1               2+----'
     |  3   4   5   6  |      |  3   4   5   6  |
     '--+---+---+---+--'      '--+---+---+---+--'
        |   |   |   |            |   |   |   |
       H5  H6  H7                H8  H9
   (VoIP) (FTP)              (SNMP (PC)
                             Agent)
```

Figure 2: Switched LAN Infrastructure Example (Ring Topology).

The ports where hosts connect (ports 3-6 on each switch) are referred to as *access ports*, while ports interconnecting switches are referred to as *VLAN trunk ports* or *trunk ports*.

Now assume you have the following situation:

- H1, H3 and H5 are located on a specific VLAN (VLAN 10) used for IP telephony traffic. Other hosts are located on VLAN 1.
- H2 is an SNMP Manager station monitoring SNMP Agents H4 and H8
- H6 is a File Server (bulk data)
- H7 is a device capable of sending *priority tagged* frames (IEEE 802.1p, VID 0).
- H9 is a maintenance PC

To assign appropriate priority, you have the following tasks:

- How can you configure your switches to give priority to the IP telephony traffic, aiming to achieve good voice quality? (H1, H3 and H5)
- What priority should be given to the SNMP traffic? Perhaps SNMP traps from the agents should be given relatively high priority, while regular SNMP Get/Set and Response traffic can have lower priority? (H2, H4 and H8)
- Bulk data downloads from the file server is assumed not to be time critical, thus should be given low priority. The same goes for all traffic from the maintenance PC. (H6 and H9).
- The device H7 is sending both low priority data and high priority data, signalling the priority with a priority VLAN tag. How should the switch be configured to take advantage of this?

In addition to the traffic from the mentioned hosts, there is also high priority traffic for the layer-2 redundancy protocol (FRNT, etc.) on the trunk links. This signalling traffic is sent with highest priority to ensure the network is working properly.

General Approach

```
        .-----------------.
        |     Switch      |     <-- Trunk ports (1&2)
    ----+1               2+---      VLAN tag holds priority
        |  3   4   5   6  |
        '--+---+---+---+--'     <-- Access ports (3-6)
           |   |   |   |            Determine priority here
          H1      H2
```

The general recommendation for determining and conveying Layer-2 priority is summarised below:

- Determine priority at the *access ports*, when the user traffic is *entering the LAN infrastructure*. In our example, ports 3-6 on the switches are access ports.
- Use VLAN tagging on the *trunk ports*, letting the tag carry the priority (802.1p) determined at the access ports. In our example, ports 1 and 2 are trunk ports.
- When configuring priority, avoid assigning the highest priority to any kind of user traffic, as that should be reserved for network signalling protocols (FRNT, etc.)

Capabilities

A packet's priority is determined on ingress, i.e., when the packet enters the switch. Three *priority modes* are available for configuration per port.

- **Port Priority mode**: If priority mode 'port' is configured on the port, then all ingress packets will be assigned the ingress port's *port priority*.
- **VLAN tag mode**: If priority mode 'VLAN tag' is configured on the port, then all ingress packets arriving with an 802.1Q tag will get priority specified in the priority field of the VLAN tag.

  If no VLAN tag is present in the arriving packet, priority is based on the *port priority* of the ingress port, see above.
- **IP DSCP/ToS mode**: If priority mode 'ip' is configured, then all ingress IPv4 packets will be assigned priority based on the DSCP value (formerly ToS) in the packets IP header.

  If the incoming packet is not an IP packet (e.g., an ARP packet), priority is based on the *port priority* of the ingress port, see above.
- **prefer [tag|ip] mode**: If priority mode 'prefer tag' is configured, then all ingressing VLAN tagged IP packets, and VLAN tagged non-IP packets, will be assigned priority based on the PCP value in the VLAN tag. Untagged IP packets will get priority based on the DSCP field in the IP header.

If priority mode ‘prefer ip’ is configured, then all ingressing VLAN tagged IP packets, and untagged IP packets, will be assigned priority based on the DSCP value in the IP header. For VLAN tagged non-IP packets use PCP priority in VLAN tag.

If none of the above matches, priority is based on the *port priority* of the ingress port, see above.

Considering the general recommendations for layer-2 priority, ‘VLAN tag’ priority should be used on trunk ports, while ‘port priority’ or ‘IP DSCP/ToS’ (or even ‘VLAN tag’) priority can be used on access ports.

By *default*, ports are configured with:
- priority mode ‘VLAN tag’
- port priority ‘0’ (zero)

Configuration

The following sections show configuration for each of the switches in Figure 2, focusing on layer-2 priority settings. IP network and layer-2 redundancy settings are also shown for completeness. In this example FRNT is used for layer-2 redundancy, with S1 as ‘focal point’.

The switches have factory configuration at start (all ports in VLAN 1). Each switch shall have two VLANs, VLAN 1 where most hosts reside and VLAN 10 for the IP telephony hosts. For management purposes, switches are assigned IP addresses each VLAN (subnets 192.168.1.0/24 and 192.168.10.0/24).

Switch 1 (S1)

```
        .-----------------.
        |        S1       |
    ----+1               2+---
        |  3   4   5   6  |
        '--+---+---+---+--'
           |   |   |   |
         H1   H2
       (VoIP) (SNMP Mgr)
```

General Configuration (S1)

In this section we show settings for VLANs, IP settings and Layer-2 redundancy. Although this is just an example of how it could be done, but shown for completeness. For layer-2 priority, the most important part here is that *trunk ports* are configured for VLAN tagging.

The VLAN configuration steps are as follows. Initially all ports are associated *untagged* with VLAN 1.

```
s1:/#> configure
s1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
s1:/config/vlan-10/#> untagged eth3
Moving untagged port eth3 from vid 1 to vid 10.
s1:/config/vlan-10/#> tagged eth1,eth2
s1:/config/vlan-10/#> end
s1:/config/#> vlan 1
s1:/config/vlan-1/#> tagged eth1,eth2
s1:/config/vlan-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s1:/#>
```

The IP address configuration steps are as follows. Initially interface *vlan1* gets addresses via DHCP as well as a link-local address (we remove the DHCP method; the link-local is kept, but could also have been removed).

```
s1:/#> configure
s1:/config/#> iface vlan1
s1:/config/iface-vlan1/#> no inet dhcp
s1:/config/iface-vlan1/#> inet static 192.168.1.1/24
s1:/config/iface-vlan1/inet-static-192.168.1.1/#> end
s1:/config/iface-vlan1/#> end
s1:/config/#> iface vlan10
s1:/config/iface-vlan10/#> inet static 192.168.10.1/24
s1:/config/iface-vlan10/inet-static-192.168.10.1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s1:/#>
```

FRNT operates on the ring ports (eth1 and eth2). S1 is selected as Focal Point (this is the default setting).

```
s1:/#> configure
s1:/config/#> frnt
Activating FRNT0 with default settings, remember to change the ring ports!
Invalid settings: No ring ports defined
s1:/config/frnt-1/#> ports eth1,eth2
s1:/config/frnt-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s1:/#>
```

Layer-2 Priority Configuration (S1)

By default, all ports are configured with priority mode *tag*, i.e., it primarily looks at the priority field carried as part of the VLAN tag.

- We keep the priority mode setting for the trunk ports (*eth1* and *eth2*).
- Port 3 (*eth3*) should be used for IP telephony. We select priority-mode *port*, and then set port priority to *5* to give it reasonably high priority without affecting FRNT signalling.
- Other access ports are also set to priority mode *port*, but the default port priority 0 (zero) is kept.

```
s1:/#> configure
s1:/config/#> port eth3
s1:/config/port-eth3/#> priority-mode port
s1:/config/port-eth3/#> priority 5
s1:/config/port-eth3/#> end
s1:/config/#> port eth4..eth6
s1:/config/port-eth4..eth6/#> priority-mode port
s1:/config/port-eth4..eth6/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s1:/#>
```

Switch 2 (S2)

```
        .-----------------.
        |        S2       |
    ----+1               2+---
        |  3   4   5   6  |
        '--+---+---+---+--'
           |   |   |   |
         H3  H4
      (VoIP) (SNMP Agent)
```

General Configuration (S2)

The configuration is very similar to S1. The major differences are:

- S2 is assigned IP address 192.168.1.2/24 and 192.168.10.2/24 on interface vlan1 and vlan10 respectively
- S2 is an FRNT *Member* switch, not a *Focal-Point*

The VLAN configuration steps are the same as for S1.

```
s2:/#> configure
s2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
s2:/config/vlan-10/#> untagged eth3
Moving untagged port eth3 from vid 1 to vid 10.
s2:/config/vlan-10/#> tagged eth1,eth2
s2:/config/vlan-10/#> end
s2:/config/#> vlan 1
s2:/config/vlan-1/#> tagged eth1,eth2
s2:/config/vlan-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s2:/#>
```

The IP address configuration steps are similar to the steps of S1, only the host part of the IP address differs.

```
s2:/#> configure
s2:/config/#> iface vlan1
s2:/config/iface-vlan1/#> no inet dhcp
s2:/config/iface-vlan1/#> inet static 192.168.1.2/24
s2:/config/iface-vlan1/inet-static-192.168.1.2/#> end
s2:/config/iface-vlan1/#> end
s2:/config/#> iface vlan10
s2:/config/iface-vlan10/#> inet static 192.168.10.2/24
s2:/config/iface-vlan10/inet-static-192.168.10.2/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s2:/#>
```

Switch S2 is configured as FRNT member (no focal-point).

```
s2:/#> configure
s2:/config/#> frnt
Activating FRNT0 with default settings, remember to change the ring ports!
Invalid settings: No ring ports defined
s2:/config/frnt-1/#> ports eth1,eth2
s2:/config/frnt-1/#> no focal-point
s2:/config/frnt-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s2:/#>
```

Layer-2 Priority Configuration (S2)

The layer-2 priority configuration on Switch 2 is very similar to the settings on Switch 1, e.g., port 3 is also used for IP telephony. The major difference here is that we have an SNMP agent device on port 4, assumed to send SNMP traps with a higher DSCP value than other data. We want to give higher priority to the SNMP traps, so we configure port 4 with priority mode *ip*.

```
s2:/#> configure
s2:/config/#> port eth3
s2:/config/port-eth3/#> priority-mode port
```

```
s2:/config/port-eth3/#> priority 5
s2:/config/port-eth3/#> end
s2:/config/#> port eth4
s2:/config/port-eth4/#> priority-mode ip
s2:/config/port-eth4/#> end
s2:/config/#> port eth5..eth6
s2:/config/port-eth5..eth6/#> priority-mode port
s2:/config/port-eth5..eth6/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s2:/#>
```

Switch 3 (S3)

```
        .--------------------.
        |         S3         |
    ----+1                  2+---
        |  3    4    5    6  |
        '--+----+----+----+--'
           |    |    |   |
          H5   H6   H7
      (VoIP) (FTP) (Prio Tag)
```

General Configuration (S3)

The configuration of S3 has much in common with S1 and S2. The only difference, compared to S2, is IP assignment; S3 is assigned IP address 192.168.1.3/24 and 192.168.10.3/24 on interface vlan1 and vlan10 respectively.

The VLAN and FRNT configuration steps are exactly the same as for S2, see above.

The IP address configuration steps are similar to the steps of S1 and S2, only the host part of the IP address differs.

```
s3:/#> configure
s3:/config/#> iface vlan1
s3:/config/iface-vlan1/#> no inet dhcp
s3:/config/iface-vlan1/#> inet static 192.168.1.3/24
s3:/config/iface-vlan1/inet-static-192.168.1.3/#> end
s3:/config/iface-vlan1/#> end
s3:/config/#> iface vlan10
s3:/config/iface-vlan10/#> inet static 192.168.10.3/24
s3:/config/iface-vlan10/inet-static-192.168.10.3/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s3:/#>
```

Layer-2 Priority Configuration (S3)

Layer-2 priority configuration on S3 intends to achieve the following:

- Trunk ports 1 and 2 use priority mode *tag* (as in S1 and S2)
- Give high priority to IP telephony on port 3 (as on S1 and S2)
- The File Server H6 on port 4 should be given low priority, avoiding that bulk data affects more important traffic
- H7 attached to port 5 sends priority tagged packets (VLAN tagged packets with VID 0). If we wish to base priority on the VLAN tag, the port needs to be configured with priority-mode *tag*. This is the default, i.e., no configuration is needed

```
s3:/#> configure
s3:/config/#> port eth3
```

```
s3:/config/port-eth3/#> priority-mode port
s3:/config/port-eth3/#> priority 5
s3:/config/port-eth3/#> end
s3:/config/#> port eth4,eth6
s3:/config/port-eth4,eth6/#> priority-mode port
s3:/config/port-eth4,eth6/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s3:/#> cp running-config startup-config
s3:/#>
```

Switch 4 (S4)

```
        .-----------------.
        |        S4       |
    ----+1               2+---
        |  3   4   5   6  |
        '--+---+---+---+--'
           |   |   |   |
          H8   H9
  (SNMP Agent) (PC)
```

General configuration (S4)

The configuration of S4 has much in common with S1-S3. The differences are:

- VLAN 10 has no access port on switch S4
- S4 is assigned IP address 192.168.1.4/24 and 192.168.10.4/24 on interface vlan1 and vlan10 respectively.

The VLAN configuration on S4 is as follows:

```
s4:/#> configure
s4:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
s4:/config/vlan-10/#> tagged eth1,eth2
s4:/config/vlan-10/#> end
s4:/config/#> vlan 1
s4:/config/vlan-1/#> tagged eth1,eth2
s4:/config/vlan-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s4:/#>
```

The IP address configuration steps are similar to the steps of S1-S3, only the host part of the IP address differs.

```
s4:/#> configure
s4:/config/#> iface vlan1
s4:/config/iface-vlan1/#> no inet dhcp
s4:/config/iface-vlan1/#> inet static 192.168.1.4/24
s4:/config/iface-vlan1/inet-static-192.168.1.4/#> end
s4:/config/iface-vlan1/#> end
s4:/config/#> iface vlan10<
s4:/config/iface-vlan10/#> inet static 192.168.10.4/24
s4:/config/iface-vlan10/inet-static-192.168.10.4/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s4:/#>
```

The FRNT configuration steps are exactly the same as for the other member switches, S2 and S3:

```
s4:/#> configure
s4:/config/#> frnt
Activating FRNT0 with default settings, remember to change the ring ports!
Invalid settings: No ring ports defined
s4:/config/frnt-1/#> ports eth1,eth2
s4:/config/frnt-1/#> no focal-point
s4:/config/frnt-1/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s4:/#>
```

Layer-2 Priority Configuration (S4)

Layer-2 priority configuration on S4 intends to achieve the following:

- Trunk ports 1 and 2 use priority mode *tag* (as in S1 and S2)
- The SNMP Agent (H8) port 3 should have priority based on IP DSCP, as in S2, enabling higher priority for SNMP traps than other traffic from H8
- The Maintenance PC (H9) on port 4 should be given low priority, avoiding that maintenance data affects more important traffic. Ports 4-6 use priority mode ‘port’, with the (default) port priority ‘0’

```
s4:/#> configure
s4:/config/#> port eth3
s4:/config/port-eth3/#> priority-mode ip
s4:/config/port-eth3/#> end
s4:/config/#> port eth4..eth6
s4:/config/port-eth4..eth6/#> priority-mode port
s4:/config/port-eth4..eth6/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
s4:/#>
```

4.3.3. Multicast Distribution HowTo

Introduction

The intention of this document is to give an overview of available mechanisms in WeOS to control multicast distribution. Focus is on distribution on a LAN although multicast routing is also covered, albeit to a smaller extent.

The following topics are covered:

- *What is multicast?*
- *Operating modes of IGMP*
- *Ideal distribution*

For additional information, please see the configuration guides on IGMP Snooping and Transparent Bridging.

Tips

The simplest of tools are often the best when debugging. Familiar tools like `ping` and `tcpdump`, when debugging unicast, are your friends also when debugging multicast.

However, when debugging *routed multicast*, **remember the TTL**. The default TTL value for multicast in most TCP/IP stacks is 1. This is because multicast in its unregulated form is broadcast. More on this below. Use `-t TTL` option to both `ping` and `mcjoin`.

```
sender $ ping -t 64 -i eth0 225.1.2.3
```

or

```
sender $ mcjoin -s -t 64 -i eth0 225.1.2.3
```

On the receiving side we must generate an IGMP report before we can see the multicast stream with tcpdump or wireshark:

```
receiver $ mcjoin -i eth0 225.1.2.3 &
receiver $ tcpdump -i eth0 dst host 225.1.2.3
```

If using the same Linux PC for both sending and receiving traffic, make sure to isolate your network interfaces from each other using network namespaces, *ip netns*. Otherwise the traffic will be short circuited in your host's TCP/IP stack.

A tool like mcjoin[1] is required on the receiver to tell the (Linux) PC to send an IGMP Join for the given group, it can also be used to send pure UDP multicast. Only after receiving the IGMP Join will the switches forward multicast in the direction of your receiver. More on IGMP Join and Membership Reports below.

Also, the switches *switch* most of the traffic, so you will likely not see any of your traffic reach the CPU of each switch. Only routers will get the traffic, but again not all traffic.

[1] https://github.com/troglobit/mcjoin[2]
[2] https://github.com/troglobit/mcjoin

What is Multicast?

Without any active form of regulation, *multicast is broadcast*. So in networks with malfunctioning IGMP/MLD snooping devices, one option can be to disable the snooping entirely. Of course this is discouraged.

WeOS support two forms of regulation: automatic using *IGMP/MLD snooping*, and manual using static MAC FDB or IP MDB filters.

IGMP/MLD

Internet Group Management Protocol, IGMP, is an IPv4 protocol originally devised for multicast subscription on LANs between a multicast router and its end devices. Multicast Listener Discovery Protocol, MLD, is the IPv6 variant and works similarly to its IPv4 counterpart, but with different IP addresses and has another dedicated MAC address range.

The standards governing IGMP, MLD, and IGMP/MLD snooping (for switches), are defined in the following set of RFCs:

- https://tools.ietf.org/html/rfc2236[1]
- https://tools.ietf.org/html/rfc3376[2]
- https://tools.ietf.org/html/rfc4604[3]
- https://tools.ietf.org/html/rfc4541[4]

The IGMP/MLD protocol itself is fairly simple: periodically a *Querier* sends out an IGMP Query: *Who wants multicast?*, to which end devices, and all intermediate switches with IGMP snooping, reply with an *IGMP Membership Report* stating which groups they want (to join or leave).

- The Querier is elected among all IGMP and IGMP Snooping capable devices based on the numerically lowest IP address. (Address 0.0.0.0 is reserved for so called “proxy queries”)
- The Querier is the designated distribution point for all multicast on the LAN. This means all switches on the LAN must forward its *known* and *unknown* multicast to that point

Known multicast is that which has at least one receiver. Switches along the path from the receiver to the elected IGMP querier forward reports towards the querier and opens up forwarding of the given group(s) in the direction of the receiver.

Unknown multicast must also be forwarded to the querier so that when a receiver requests it the querier can immediately forward it.

An end device (receiver) may at any point leave or join a group. I.e., a membership report can be sent at any time in the query interval. The leave process is not immediate unless the switch port has *Fast Leave* enabled.

An end device that fails (or neglects to send leave) to respond to two queries will have its group memberships timed out.

Static Filters

WeOS also supports setting static multicast filters (MAC or IP group), as described in the transparent bridge configuration guide. This can help in some cases, but also has its limitations.

When IGMP snooping is enabled, a static filter is *logically OR:ed* with any dynamic filter from IGMP snooping.

[1] https://tools.ietf.org/html/rfc2236
[2] https://tools.ietf.org/html/rfc3376
[3] https://tools.ietf.org/html/rfc4604
[4] https://tools.ietf.org/html/rfc4541

When IGMP snooping is disabled, multicast is by default flooded like broadcast, or more specifically, controlled by the *multicast-flood-unknown* setting which is enabled by default. The exception is for matching *static filters* where traffic is only forwarded on the listed ports. However, in such a setup *all multicast groups* must be listed in the FDB; otherwise traffic to not listed groups will still be flooded on all ports. This is not obvious at first glance.

> **TIP**
>
> Usually, static FDB entries are best suited for opening up multicast on ports towards end devices that do not speak IGMP. In addition, static entries avoids flooding the given groups on other ports.

```
example:/#> configure
example:/config/#> fdb
example:/config/fdb/#> group 225.1.2.3 vlan 1 eth3,eth5
example:/config/fdb/#> leave
example:/#> copy run start
```

The above example adds 225.1.2.3 statically to ports eth3 and eth5 in VLAN 1.

IGMP/MLD Snooping

> **NOTE**
>
> WeOS supports IGMP snooping, while support for MLD snooping is pending. See the IGMP Snooping Configuration Guide for more information.

The IGMP/MLD snooping implementation in WeOS supports the following two modes:
- Auto mode
- Proxy mode

> **NOTE**
>
> Never mix IGMP modes on a LAN unless you know what you are doing!

Auto Mode

Auto and *Proxy* modes are very similar. In auto mode the switch will take part in the previously mentioned Querier election, in proxy mode it will not. Proxy mode is also the fallback mode on a VLAN where no IP address on the VLAN interface is found.

> **NOTE**
>
> Engineering the IP address plan for optimal placement of the Querier and its backups can be tricky. For more help, see the CLI commands `ip multicast-router-timeout` and `ip multcast-querier-robustness` in IGMP snooping configuration guide.

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> multicast-router-timeout 24
```

```
example:/config/ip/#> leave
example:/#> copy run start
```

Do not set a router timeout less than the IGMP querier interval.

Proxy Mode

In *Proxy* mode the switch does not take part in the Querier election (in any VLAN). By default, all queries sent in proxy mode use source address 0.0.0.0, which is exempt in the election process.

The most characteristic distinction Proxy mode has, compared to auto mode, is that it can do *nothing* unless explicitly asked.

- Proxy mode never forwards a query from a neighbouring proxy mode switch
- Proxy mode only responds with an IGMP Report to queries

This effectively means that multicast is localised as much as possible. Hence, proxy mode in combination with IGMP *Fast Leave* on trunk ports can be a very attractive option for setups with low-bandwidth trunks (DSL).

The *proxy query* mode can be disabled - then the unit will not act proxy querier, not even as fall-back mode in interfaces lacking IP address.

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> no multicast-proxy-query
example:/config/ip/#> leave
example:/#> copy run start
```

In a LAN with *Proxy mode-only* switches and with this feature is disabled, multicast distribution will be severely affected. It is recommended to have at least one device on the LAN send queries, either with an actual source IP address or address 0.0.0.0.

4.3.4. IEEE 802.1X Basic Setup HowTo

Introduction

This document describes a basic setup of port access control using 802.1X. For an overview of all available configuration settings for Port Access Control, refer to the configuration section in the Port Access Control Guide.

Overview

Below is the topology of the devices in this setup.

```
                                                |
                                              .--.-.
                        Authentication      ( (    )__
                           Server          (_,  \ ) ,_)
                          .------.           '-'--`--'
                          |RADIUS|                | GW 10.0.2.1
                          '--+---'                |    VLAN 2
                             | 10.0.1.2           |    10.0.2.0/24
                             |                    | 10.0.2.2
                          .--+------+------+------+--.
            Authenticator |  5      6      7      8  |
           (Switch/Router) |           WeOS            |
              10.0.1.1    |  1      2      3      4  |
                          '--+------+------+------+--'
                             X      X      X      X  <== Controlled Ports
                             |    VLAN 1              (All Blocked by
                             |    10.0.1.0/24          Authenticator)
                          .--+-.
                          | H1 | (Supplicant)
                          '----' IP 10.0.1.43/24
                                 GW 10.0.1.1
```

Figure 1: A basic setup of port access control using 802.1X. Ports eth1-eth4 are controlled. Here ports eth1-eth7 are on VLAN 1 and eth8 is on VLAN 2, but they could be on the same VLAN.

Configuration

First VLANs and IP addresses of the WeOS router is configured, see Figure 1 above. It is assumed that all ports are assigned (untagged) to VLAN 1 from start. For more information, see VLAN configuration.

```
example:/#> config
example:/config/#> vlan 2
example:/config/vlan-2/#> untagged eth8
example:/config/vlan-2/#> end
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> no inet
example:/config/iface-vlan1/#> inet static 10.0.1.1/24
example:/config/iface-vlan1/inet-static-10.0.1.1/#> end
example:/config/iface-vlan1/#> end
example:/config/#> iface vlan2
example:/config/iface-vlan2/#> inet static 10.0.2.2/24
example:/config/iface-vlan2/inet-static-10.0.2.2/#> end
example:/config/iface-vlan2/#> end
example:/config/#> ip
example:/config/ip/#> route default 10.0.2.1
```

```
example:/config/ip/#> leave
example:/#>
```

It is assumed that an external RADIUS server is already setup with appropriate settings for 802.1X. In this example, the RADIUS server is located at IP address 10.0.1.2, and is listening to the default UDP port (1812). In WeOS, a (RADIUS) remote server instance configured. This is done in the AAA context.

```
example:/#> config
example:/config/#> aaa
example:/config/aaa/#> remote-server 1
example:/config/aaa/remote-server-1/#> description MyRADIUS
example:/config/aaa/remote-server-1/#> address 10.0.1.2
example:/config/aaa/remote-server-1/#> password TopS3cret
example:/config/aaa/remote-server-1/#> leave
example:/#>
```

See Remote Server Settings for more information on how to configure a RADIUS remote server instance.

> **NOTE**
>
> For best performance, it is recommended to use IP address (rather than domain name) when defining the RADIUS server.

With an authentication server instance created, enabling 802.1X access control on ports eth1-eth4 can be done as follows.

```
example:/#> config
example:/config/#> port-access
Creating new Port Access instance 1.
example:/config/port-access-1/#> port eth1..eth4
example:/config/port-access-1/#> description "Site A"
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> dot1x
Authentication server not set.
example:/config/port-access-1/dot1x/#> authentication-server 1
example:/config/port-access-1/dot1x/#> leave
example:/#>
```

> **PORT METHOD**
>
> In this example we explicitly set the port method to `mac-based`, which is also the default setting. With this method, each individual client must be authenticated and its MAC address is unlocked on the port. For more information about port methods, see Port Method.

Status

Port access status can be shown by issuing `show port-access` in the admin context.

```
example:/#> show port-access
Number of Port Access Instances: 1
```

```
Table Instance Dividers        : Disabled

 802.1X Authentications
#  MAC-ADDRESS        SUPPLICANT     PORT     SERVER           DATE-ADDED
1  00:80:c8:3c:25:b8  wpa_suppl      eth1     10.0.1.2         Nov 22 14:13:29

 MAC Auth Authentications
#  MAC-ADDRESS        PORT     METHOD                          DATE-ADDED
No authenticated MAC addresses found.

 Ports
PORT     METHOD       STATE                            AUTHENTICATED-MAC
eth1     MAC Based    1 Authentication                 00:80:c8:3c:25:b8
eth2     MAC Based    0 Authentications                -
eth3     MAC Based    0 Authentications                -
eth4     MAC Based    0 Authentications                -

 VLAN Assignment
PORT     ENABLED  STATE        FROM     ASSIGNING-CLIENT   VLAN
eth1     No       Unassigned   -        -                  -
eth2     No       Unassigned   -        -                  -
eth3     No       Unassigned   -        -                  -
eth4     No       Unassigned   -        -                  -

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

For more information about the status output and the different tables, see Port Access Control Status.

4.3.5. MAC authentication using accept list HowTo

Introduction

This document describes a setup of MAC authentication using a local MAC accept list to authorise MAC addresses.

Overview

For an overview of all available configuration settings for Port Access Control, refer to the configuration section in the Port Access Control Guide.

Below is the topology of the devices in this setup.

```
                                                |
                                              .--.-.
                                            ( (    )__
                                          (_,  \ ) ,_)
                                            '-'--`--'
                                                | GW 10.0.2.1
                                                |    VLAN 2
                                                |    10.0.2.0/24
                                                | 10.0.2.2
 Authenticator  .--+------+------+------+--.
(Switch/Router) |  5      6      7      8  |
with Accept-List|           WeOS           |
   10.0.1.1     |  1      2      3      4  |
                '--+------+------+------+--'
                   X      X      X      X  <== Controlled Ports
                   |    VLAN 1              (All Blocked by
                   |    10.0.1.0/24          Authenticator)
                .--+-.
                | H1 | (Host/Supplicant)
                '----' IP 10.0.1.43/24
                       GW 10.0.1.1
```

Figure 1: A basic setup of port access control using MAC Authentication. Ports eth1-eth4 are controlled. Here ports eth1-eth7 are on VLAN 1 and eth8 is on VLAN 2, but they could be on the same VLAN.

Configuration

First VLANs and IP addresses of the WeOS router is configured, see Figure 1 above. It is assumed that all ports are assigned (untagged) to VLAN 1 from start. For more information, see VLAN configuration.

```
example:/#> config
example:/config/#> vlan 2
example:/config/vlan-2/#> untagged eth8
example:/config/vlan-2/#> end
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> no inet
example:/config/iface-vlan1/#> inet static 10.0.1.1/24
example:/config/iface-vlan1/inet-static-10.0.1.1/#> end
example:/config/iface-vlan1/#> end
example:/config/#> iface vlan2
example:/config/iface-vlan2/#> inet static 10.0.2.2/24
example:/config/iface-vlan2/inet-static-10.0.2.2/#> end
example:/config/iface-vlan2/#> end
```

```
example:/config/#> ip
example:/config/ip/#> route default 10.0.2.1
example:/config/ip/#> leave
example:/#>
```

A MAC authentication list configured. This is done in the AAA context.

```
example:/#> config
example:/config/#> aaa
example:/config/aaa/#> mac-auth 1
example:/config/aaa/mac-auth-1/#> mac match 00:80:c8:3c:25:b8 description "My PC"
example:/config/aaa/mac-auth-1/#> leave
example:/config/#> leave
example:/#>
```

See MAC match Settings for more information on how to configure a MAC authentication list.

With a MAC authentication list created, enabling MAC Authentication on ports eth1-eth4 can be done as follows.

```
example:/#> config
example:/config/#> port-access
Creating new Port Access instance 1.
example:/config/port-access-1/#> port eth1..eth4
example:/config/port-access-1/#> description "Site A"
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> mac-authentication
example:/config/port-access-1/mac-auth/#> accept-list 1
example:/config/port-access-1/mac-auth/#> leave
example:/#>
```

> **PORT METHOD**
>
> In this example we explicitly set the port method to `mac-based`, which is also the default setting. With this method, each individual client must be authenticated and its MAC address is unlocked on the port. For more information about port methods, see Port Method.

Status

Port access status can be shown by issuing `show port-access` in the admin context.

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled

╒ 802.1X Authentications ═══════════════════════════════════════════════
# MAC-ADDRESS        SUPPLICANT    PORT    SERVER          DATE-ADDED
No authenticated 802.1X clients found.

╒ MAC Auth Authentications ═══════════════════════════════════════════
# MAC-ADDRESS        PORT    METHOD                          DATE-ADDED
1 00:80:c8:3c:25:b8  eth1    Whitelist                       Nov 22 18:40:21
```

```
Ports
PORT     METHOD       STATE                          AUTHENTICATED-MAC
eth1     MAC Based    1 Authentication               00:80:c8:3c:25:b8
eth2     MAC Based    0 Authentications              -
eth3     MAC Based    0 Authentications              -
eth4     MAC Based    0 Authentications              -

VLAN Assignment
PORT     ENABLED   STATE        FROM     ASSIGNING-CLIENT   VLAN
eth1     No        Unassigned   -        -                  -
eth2     No        Unassigned   -        -                  -
eth3     No        Unassigned   -        -                  -
eth4     No        Unassigned   -        -                  -

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

For more information about the status output and the different tables, see Port Access Control Status.

4.3.6. MAC authentication using RADIUS (MAB) HowTo

Introduction

This document describes a setup of MAC authentication using a RADIUS to authorise MAC addresses (also referred to as MAB). For an overview of all available configuration settings for Port Access Control, refer to the configuration section in the Port Access Control Guide.

Overview

Below is the topology of the devices in this setup.

```
                                                |
                                              .--.-.
                      Authentication        ( (    )__
                         Server            (_,  \ ) ,_)
                        .------.             '-'--`--'
                        |RADIUS|                | GW 10.0.2.1
                        '--+---'                |    VLAN 2
                           | 10.0.1.2           |    10.0.2.0/24
                           |                    | 10.0.2.2
                        .--+------+------+------+--.
          Authenticator |  5      6      7      8  |
         (Switch/Router) |           WeOS           |
            10.0.1.1    |  1      2      3      4  |
                        '--+------+------+------+--'
                           X      X      X      X  <== Controlled Ports
                           |    VLAN 1              (All Blocked by
                           |    10.0.1.0/24          Authenticator)
                        .--+-.
                        | H1 | (Host/Supplicant)
                        '----' IP 10.0.1.43/24
                               GW 10.0.1.1
```

Figure 1: A basic setup of port access control using MAC Authentication. Ports eth1-eth4 are controlled. Here ports eth1-eth7 are on VLAN 1 and eth8 is on VLAN 2, but they could be on the same VLAN.

Configuration

First VLANs and IP addresses of the WeOS router is configured, see Figure 1 above. It is assumed that all ports are assigned (untagged) to VLAN 1 from start. For more information, see VLAN configuration.

```
example:/#> config
example:/config/#> vlan 2
example:/config/vlan-2/#> untagged eth8
example:/config/vlan-2/#> end
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> no inet
example:/config/iface-vlan1/#> inet static 10.0.1.1/24
example:/config/iface-vlan1/inet-static-10.0.1.1/#> end
example:/config/iface-vlan1/#> end
example:/config/#> iface vlan2
example:/config/iface-vlan2/#> inet static 10.0.2.2/24
example:/config/iface-vlan2/inet-static-10.0.2.2/#> end
example:/config/iface-vlan2/#> end
example:/config/#> ip
example:/config/ip/#> route default 10.0.2.1
```

```
example:/config/ip/#> leave
example:/#>
```

It is assumed that an external RADIUS server is already setup with appropriate settings for MAC Authentication (WeOS uses RADIUS attribute *Calling-ID* to convey the MAC address). In this example, the RADIUS server is located at IP address 10.0.1.2, and is listening to the default UDP port (1812). In WeOS, a (RADIUS) remote server instance configured. This is done in the AAA context.

```
example:/config/#> aaa
example:/config/aaa/#> remote-server 1
example:/config/aaa/remote-server-1/#> description MyRADIUS
example:/config/aaa/remote-server-1/#> address 10.0.1.2
example:/config/aaa/remote-server-1/#> password TopS3cret
example:/config/aaa/remote-server-1/#> leave
example:/#>
```

See Remote Server Settings for more information on how to configure a RADIUS remote server instance.

> **NOTE**
>
> For best performance, it is recommended to use IP address (rather than domain name) when defining the RADIUS server.

With an authentication server instance created, enabling MAC Authentication on ports eth1-eth4 can be done as follows.

```
example:/config/#> port-access
Creating new Port Access instance 1.
example:/config/port-access-1/#> port eth1..eth4
example:/config/port-access-1/#> description "Site A"
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> mac-authentication
example:/config/port-access-1/mac-auth/#> authentication-server 1
example:/config/port-access-1/mac-auth/#> leave
example:/#>
```

> **PORT METHOD**
>
> In this example we explicitly set the port method to `mac-based`, which is also the default setting. With this method, each individual client must be authenticated and its MAC address is unlocked on the port. For more information about port methods, see Port Method.

Status

Port access status can be shown by issuing `show port-access` in the admin context.

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled

╒ 802.1X Authentications ═══════════════════════════════════════════════╕
```

```
#   MAC-ADDRESS         SUPPLICANT     PORT     SERVER            DATE-ADDED
No authenticated 802.1X clients found.


MAC Auth Authentications
#   MAC-ADDRESS         PORT     METHOD                          DATE-ADDED
1   00:80:c8:3c:25:b8   eth1     MAB (10.0.1.2:1812)             Nov 22 18:12:55


Ports
PORT     METHOD       STATE                              AUTHENTICATED-MAC
eth1     MAC Based    1 Authentication                   00:80:c8:3c:25:b8
eth2     MAC Based    0 Authentications                  -
eth3     MAC Based    0 Authentications                  -
eth4     MAC Based    0 Authentications                  -


VLAN Assignment
PORT     ENABLED   STATE        FROM     ASSIGNING-CLIENT    VLAN
eth1     No        Unassigned   -        -                   -
eth2     No        Unassigned   -        -                   -
eth3     No        Unassigned   -        -                   -
eth4     No        Unassigned   -        -                   -


- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

For more information about the status output and the different tables, see Port Access Control Status.

4.3.7. Dynamic VLAN Assignment with 802.1X HowTo

Introduction

This document describes how dynamic VLAN assignment can be configured for 802.1X authentication. For an overview of all available configuration settings for Port Access Control, refer to the configuration section in the Port Access Control Guide.

> **SEE ALSO**
>
> Related Documentation
> - Port Access Control Guide
> - 802.1X Basic Setup HowTo
> - Dynamic VLAN Assignment with MAC Authentication Bypass (MAB) HowTo

Overview

This is a simplified example designed to provide a clear, basic demonstration of what needs to be configured on the WeOS device to get VLAN assignment working with port access control using 802.1X. The focus is on understanding the core configuration concepts rather than necessarily providing a complete, production-ready deployment. Real-world implementations may require additional security considerations, redundancy, and more complex network topologies.

For this use case example the following topology, or similar, is considered:

```
                                Authentication
                                    Server          Supplicant-1 -> VLAN 10
                                   .------.         Supplicant-2 -> VLAN 20
                                   |RADIUS|         Supplicant-3 -> VLAN 30
                                   '--+---'         Supplicant-4 -> VLAN 40
                                      | 10.0.1.2
                                      | VLAN 1
                                      |
        |               |           |           |               |
    .--.-.          .--.-.          |       .--.-.          .--.-.
   ( (    )__      ( (    )__       |      ( (    )__      ( (    )__
  (_,  \ ) ,_)    (_,  \ ) ,_)      |     (_,  \ ) ,_)    (_,  \ ) ,_)
    '-'--`--'       '-'--`--'       |       '-'--`--'       '-'--`--'
        |               |           |           |               |
        |   20.0.0.0/24 |           |           | 40.0.0.0/24   |
        |       VLAN 20 |           | VLAN 1    | VLAN 40       |
        |               |           | 10.0.1.1  |               |
        | 10.0.0.0/24 .-+-----------+-----------+-. 30.0.0.0/24 |
        | VLAN 10     | 6          5           4 |     VLAN 30 |
        '-------------+ 7         WeOS         3 +-------------'
                      |    1                2    |
                      '----+----------------+----'
                           X    Start in    X    <== Controlled Ports
                           |     VLAN 1     |     (All Blocked by
                           |                |      Authenticator)
                        .--+-.            .-+--.
     00:11:22:33:44:55 | H1 |            | H2 | 00:aa:bb:cc:dd:ee
                        '----'            '----'
                      Supplicant-1      Supplicant-3
                      Supplicant-2      Supplicant-4
```

Figure 1: A basic setup of port access control using 802.1X. It presents two different hosts H1 and H2, that can have two different supplicants each that should place the ports eth1 and eth2 in different VLANs on the authenticator WeOS device, based on the supplicant used on the host.

> **PORT NAMES**
>
> The port numbers as presented in **Figure 1** will be referred to as eth1-eth7 in the configuration examples below, i.e. 1 -> eth1, 2 -> eth2, etc.

With this example setup the goal is to have the ports, connected to the hosts, on the WeOS device be assigned dynamically to different VLANs based on the supplicant that connects. For example, if Supplicant-1 connects to the port, the port should be assigned to VLAN 10, but if Supplicant-2 connects to the same port, the port should be assigned to VLAN 20. The same applies for Supplicant-3 and Supplicant-4, but with VLANs 30 and 40 respectively.

In this case it is assumed that whatever RADIUS server that is used to process the authentication requests from the WeOS device is configured to return the correct VLAN assignment information for each supplicant. The RADIUS server must therefore be configured to supply the following, in addition to everything else that is required for the authentication to succeed:

```
Tunnel-Type = VLAN,
Tunnel-Medium-Type = 6,
Tunnel-Private-Group-Id = [numeric VLAN tag, e.g. 10, 20, 30, 40]
```

It is this **Tunnel** information that is used by the WeOS device to determine which VLAN to assign the port to when a successful authentication occurs.

> **MULTIPLE TUNNEL GROUPS**
>
> A RADIUS server may include multiple sets of tunnel attributes in a single response, each identified by a Tag field (0–31) per RFC 2868. When multiple valid tunnel groups are present, the group with the **lowest Tag value** is selected for VLAN assignment. A tunnel group is only considered valid if `Tunnel-Type` is VLAN (13), `Tunnel-Medium-Type` is IEEE-802 (6), and `Tunnel-Private-Group-ID` contains a numeric VID in the range 1–4094. Since a port can only be assigned untagged to a single VLAN, only one tunnel group is ever used regardless of how many the server provides. If at least one `Tunnel-Type` attribute is present but no valid tunnel group can be formed, 802.1X treats the response as if no VLAN information was provided (i.e. the “no VID in response” rules apply).

> **ONLY ASSIGNABLE UNTAGGED**
>
> As described in the VLAN Assignment section in the Port Access Control Guide, ports can only be assigned untagged to VLANs when doing dynamic VLAN assignment. Further, this also means that a port can only ever be assigned to one VLAN at a time when doing dynamic VLAN assignment. Additionally, this means that it is the first authenticated client that determines which VLAN the port is assigned.

Configuration

The first thing to do is to configure the base VLAN vlan1 and its associated interface. As mentioned in the topology description, the ports that face the hosts (eth1 and eth2) start in VLAN 1, and this is

where they will be moved back to when the supplicants disconnect. In addition, the port on the WeOS device that faces the RADIUS server (eth5) is also in VLAN 1.

```
example:/#> config
example:/config/#> vlan 1
example:/config/vlan-1/#> untagged eth1,eth2,eth5
example:/config/vlan-1/#> end
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> inet static 10.0.1.1/24
example:/config/iface-vlan1/inet-static-10.0.1.1/#> end
example:/config/iface-vlan1/#> end
example:/config/#>
```

The VLANs that are to be assigned dynamically must also be configured. Both so that there is a port providing access to each VLAN and because any VLAN to be assigned dynamically must exist on the device, these will not be created automatically when the RADIUS server returns the VLAN assignment information, for more information, see the section on assignable VLANs in the Port Access Control Guide.

```
example:/config/#> vlan 10
example:/config/vlan-10/#> untagged eth7
example:/config/vlan-10/#> end
example:/config/#> vlan 20
example:/config/vlan-20/#> untagged eth6
example:/config/vlan-20/#> end
example:/config/#> vlan 30
example:/config/vlan-30/#> untagged eth3
example:/config/vlan-30/#> end
example:/config/#> vlan 40
example:/config/vlan-40/#> untagged eth4
example:/config/vlan-40/#> end
example:/config/#> iface vlan10
example:/config/iface-vlan10/#> inet static 10.0.0.1/24
example:/config/iface-vlan10/inet-static-10.0.0.1/#> end
example:/config/iface-vlan10/#> end
example:/config/#> iface vlan20
example:/config/iface-vlan20/#> inet static 20.0.0.1/24
example:/config/iface-vlan20/inet-static-20.0.0.1/#> end
example:/config/iface-vlan20/#> end
example:/config/#> iface vlan30
example:/config/iface-vlan30/#> inet static 30.0.0.1/24
example:/config/iface-vlan30/inet-static-30.0.0.1/#> end
example:/config/iface-vlan30/#> end
example:/config/#> iface vlan40
example:/config/iface-vlan40/#> inet static 40.0.0.1/24
example:/config/iface-vlan40/inet-static-40.0.0.1/#> end
example:/config/iface-vlan40/#> end
example:/config/#>
```

Next, the RADIUS server definition to be used by the port access instance is configured. The RADIUS server is located at IP address 10.0.1.2 and uses the default RADIUS port (1812) in this example.

```
example:/config/#> aaa
example:/config/aaa/#> remote-server 1
Creating new remote server 1
example:/config/aaa/remote-server-1/#> description MyRADIUS
```

```
example:/config/aaa/remote-server-1/#> address 10.0.1.2
example:/config/aaa/remote-server-1/#> password TopS3cret
example:/config/aaa/remote-server-1/#> end
example:/config/aaa/#> end
```

PERFORMANCE CONSIDERATION

For best performance, it is recommended to use IP address (rather than domain name) when defining the RADIUS server.

At this point the port access instance can be configured. The ports eth1 and eth2 are added as port access ports, and the 802.1X is enabled for the instance.

```
example:/config/#> port-access
Creating new Port Access instance 1.
example:/config/port-access-1/#> port eth1,eth2
example:/config/port-access-1/#> description "VLAN Assignment Example"
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> dot1x
example:/config/port-access-1/dot1x/#> authentication-server 1
example:/config/port-access-1/dot1x/#> end
example:/config/port-access-1/#>
```

PORT METHOD

In this example the port method is explicitly set to `mac-based`, which is also the default setting. In terms of VLAN assignment the behaviour should be the same even if `port-based` is used as the port method. See Port Method for more information about port methods.

Finally the VLAN assignment is also enabled for the port access instance. It is enabled and the VLANs to allow are selected.

```
example:/config/port-access-1/#> vlan-assignment
example:/config/port-access-1/vlan-assignment/#> allowed-vlans vlan10 vlan20 vlan30 vlan40
example:/config/port-access-1/vlan-assignment/#> end
example:/config/port-access-1/#>
```

THE ALLOWED VLANS CAN ALSO BE SPECIFIED BY THEIR VID

The allowed VLANs can also be specified by their numeric VLAN tag, e.g. **10, 20, 30, 40** using the `allowed-vid` command instead of `allowed-vlans`. They manipulate the exact same setting, and can be used interchangeably:

```
example:/config/port-access-1/vlan-assignment/#> allowed-vid 10 20 30 40
```

At this point the port access configuration should look something like:

```
example:/config/port-access-1/#> show
Port Access General Settings
```

```
Status                  : Enabled
Description             : (No description)
Ports                   : eth1..eth2
Port Method             : MAC Based

802.1X Settings
Status                  : Enabled
Authentication server   : 1
Active authentication   : 30 Seconds
Re-authenticate         : Disabled
Quiet period            : 10 Seconds

MAC Authentication Settings
Not configured.

VLAN Assignment Settings
Status                  : Enabled
Allowed VIDs            : 10 (vlan10), 20 (vlan20), 30 (vlan30), 40 (vlan40)
Default VLAN            : Disabled
example:/config/port-access-1/#>
```

At this point the basic configuration is done, and the ports eth1 and eth2 should be able to authenticate and be assigned to the correct VLANs based on the supplicant used and the information returned from the RADIUS server. Therefore, exit the configuration context to apply the configuration.

```
example:/config/port-access-1/#> leave
[*] System setup running ...
[*] System Ready ...
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

Using Separate Port Access Instances to Isolate Ports to Certain VLANs

In the example above a single port access instance was used for both ports eth1 and eth2, and it would be possible for both ports to be assigned to any of the VLANs 10, 20, 30 and 40 based on the supplicant used. If the goal is to isolate the ports so that eth1 can only be assigned to VLAN 10 and VLAN 20, and eth2 can only be assigned to VLAN 30 and VLAN 40, two separate port access instances can be used, one for each port, and the allowed VLANs can be configured separately for each instance.

In such a case the configuration of port access instance 1 for port eth1 would look something like this:

```
example:/config/#> port-access 1
Creating new Port Access instance 1.
example:/config/port-access-1/#> port eth1
example:/config/port-access-1/#> description "Site A - Port 1"
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> dot1x
example:/config/port-access-1/dot1x/#> authentication-server 1
example:/config/port-access-1/dot1x/#> end
example:/config/port-access-1/#> vlan-assignment
example:/config/port-access-1/vlan-assignment/#> allowed-vlans vlan10 vlan20
example:/config/port-access-1/vlan-assignment/#> end
```

```
example:/config/port-access-1/#> end
example:/config/#>
```

And the configuration of port access instance 2 for port eth2 would look something like this:

```
example:/config/#> port-access 2
Creating new Port Access instance 2.
example:/config/port-access-2/#> port eth2
example:/config/port-access-2/#> description "Site A - Port 2"
example:/config/port-access-2/#> port-method mac-based
example:/config/port-access-2/#> dot1x
example:/config/port-access-2/dot1x/#> authentication-server 1
example:/config/port-access-2/dot1x/#> end
example:/config/port-access-2/#> vlan-assignment
example:/config/port-access-2/vlan-assignment/#> allowed-vlans vlan30 vlan40
example:/config/port-access-2/vlan-assignment/#> end
example:/config/port-access-2/#> end
example:/config/#>
```

With these two separate port access instances, port eth1 can now only be assigned to VLAN 10 and VLAN 20, and port eth2 can only be assigned to VLAN 30 and VLAN 40, based on the supplicant used and the information returned from the RADIUS server. In this example only the port config and VLAN assignment settings are different between the two instances, but all the other settings can also be configured differently if needed, e.g. different authentication servers, different port methods, etc.

Status

This section demonstrates how to verify the port access and VLAN assignment status at different stages. The following examples show the output before any authentication takes place, after initial authentication with specific supplicants, and after the ports are reassigned to different VLANs when different supplicants connect.

Before Authentication

Before any supplicant has attempted to authenticate, the status of the port access instance should look something like this:

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled

╒ 802.1X Authentications ═══════════════════════════════════════════════
# MAC-ADDRESS        SUPPLICANT    PORT    SERVER          DATE-ADDED
No authenticated 802.1X clients found.

╒ MAC Auth Authentications ═══════════════════════════════════════════
# MAC-ADDRESS        PORT    METHOD                          DATE-ADDED
No authenticated MAC addresses found.

╒ Ports ══════════════════════════════════════════════════════════════
PORT    METHOD      STATE                           AUTHENTICATED-MAC
eth1    MAC Based   0 Authentications               -
eth2    MAC Based   0 Authentications               -
```

```
└──────────────────────────────────────────────────────────────────────────────┘

┌ VLAN Assignment ──────────────────────────────────────────────────────────────
│ PORT     ENABLED  STATE       FROM     ASSIGNING-CLIENT   VLAN
│ eth1     Yes      Unassigned  -        -                  -
│ eth2     Yes      Unassigned  -        -                  -
└──────────────────────────────────────────────────────────────────────────────

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

In addition the ports eth1 and eth2 are in their starting VLAN (vlan1):

```
example:/#> show vlan
example:/#> show vlan
VID  NAME              TAGGED/UNTAGGED/POLICY NEST
   1 vlan1             T:
                       U:eth1..eth2, eth5
                       N:
  10 vlan10            T:
                       U:eth7
                       N:
  20 vlan20            T:
                       U:eth6
                       N:
  30 vlan30            T:
                       U:eth3
                       N:
  40 vlan40            T:
                       U:eth4
                       N:
example:/#>
example:/#> show port
example:/#> show port
PORT              LINK      TYPE        SPEED   STATE            OWNER/VLAN
eth1              Up         N/A      Unknown   Forwarding       vlan U:1
eth2              Up         N/A      Unknown   Forwarding       vlan U:1
eth3              Up         N/A      Unknown   Forwarding       vlan U:30
eth4              Up         N/A      Unknown   Forwarding       vlan U:40
eth5              Up         N/A      Unknown   Forwarding       vlan U:1
eth6              Up         N/A      Unknown   Forwarding       vlan U:20
eth7              Up         N/A      Unknown   Forwarding       vlan U:10
example:/#>
```

After Authentication

Now let's assume that authentication is done with Supplicant-1 on host H1 and Supplicant-3 on host H2. The ports eth1 and eth2 should then be assigned to VLAN 10 and VLAN 30 respectively, and the status should look something like this:

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled
```

```
802.1X Authentications
#  MAC-ADDRESS         SUPPLICANT     PORT    SERVER          DATE-ADDED
1  00:11:22:33:44:55   Supplicant-1   eth1    10.0.1.2        Mar 11 10:23:15
2  00:aa:bb:cc:dd:ee   Supplicant-3   eth2    10.0.1.2        Mar 11 10:24:08

MAC Auth Authentications
#  MAC-ADDRESS         PORT    METHOD                          DATE-ADDED
No authenticated MAC addresses found.

Ports
PORT    METHOD      STATE                          AUTHENTICATED-MAC
eth1    MAC Based   1 Authentication               00:11:22:33:44:55
eth2    MAC Based   1 Authentication               00:aa:bb:cc:dd:ee

VLAN Assignment
PORT    ENABLED  STATE       FROM     ASSIGNING-CLIENT   VLAN
eth1    Yes      Assigned    Server   00:11:22:33:44:55   VID 10 (vlan10)
eth2    Yes      Assigned    Server   00:aa:bb:cc:dd:ee   VID 30 (vlan30)

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

The VLAN and port status should now also reflect the new VLAN assignment for the ports:

```
example:/#> show vlan
VID  NAME             TAGGED/UNTAGGED/POLICY NEST
   1 vlan1            T:
                      U:eth5
                      N:
  10 vlan10           T:
                      U:eth1, eth7
                      N:
  20 vlan20           T:
                      U:eth6
                      N:
  30 vlan30           T:
                      U:eth2..eth3
                      N:
  40 vlan40           T:
                      U:eth4
                      N:
example:/#>
example:/#> show port
PORT             LINK     TYPE       SPEED   STATE          OWNER/VLAN
eth1             Up        N/A     Unknown  Forwarding     vlan U:10
eth2             Up        N/A     Unknown  Forwarding     vlan U:30
eth3             Up        N/A     Unknown  Forwarding     vlan U:30
eth4             Up        N/A     Unknown  Forwarding     vlan U:40
eth5             Up        N/A     Unknown  Forwarding     vlan U:1
eth6             Up        N/A     Unknown  Forwarding     vlan U:20
```

```
eth7              Up        N/A     Unknown  Forwarding      vlan U:10
example:/#>
```

As you can see, eth1 has been dynamically assigned to VLAN 10 (vlan10) and eth2 has been assigned to VLAN 30 (vlan30) based on the authentication information received from the RADIUS server. The ports are now untagged members of their respective assigned VLANs instead of the initial VLAN 1.

> **REGARDING MULTIPLE SUPPLICANTS ON THE SAME PORT**
>
> At this point the port's VLAN is more or less "locked" to the first authenticated supplicant for the given port. If any other supplicant would connect to the same port, it would have to request the same VLAN assignment to be able to authenticate successfully.
>
> However, in the use case presented in **Figure 1** multiple supplicants will not be present on the same port. But it is worth mentioning since this could be the case if there are multiple different hosts that could connect to the same port, e.g. if another switch is connected to the port and has multiple hosts behind it.

Ports eth1 and eth2 will now remain assigned to VLAN 10 and VLAN 30 respectively until the authenticated clients are deauthenticated, either by disconnecting the supplicant, when the port is brought down, etc. When this happens the ports will be moved back to their starting VLANs, in this use case it is VLAN 1.

Now let's say that Supplicant-1 and Supplicant-3 have been deauthenticated, and then Supplicant-2 and Supplicant-4 connect to the ports eth1 and eth2 respectively. The ports should then be assigned to VLAN 20 and VLAN 40 respectively, and the status should look something like this:

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled

 802.1X Authentications
#  MAC-ADDRESS        SUPPLICANT    PORT    SERVER          DATE-ADDED
1  00:11:22:33:44:55  Supplicant-2  eth1    10.0.1.2        Mar 11 11:35:42
2  00:aa:bb:cc:dd:ee  Supplicant-4  eth2    10.0.1.2        Mar 11 11:36:18

 MAC Auth Authentications
#  MAC-ADDRESS        PORT    METHOD                          DATE-ADDED
No authenticated MAC addresses found.

 Ports
PORT    METHOD      STATE                          AUTHENTICATED-MAC
eth1    MAC Based   1 Authentication               00:11:22:33:44:55
eth2    MAC Based   1 Authentication               00:aa:bb:cc:dd:ee

 VLAN Assignment
PORT    ENABLED  STATE       FROM     ASSIGNING-CLIENT   VLAN
eth1    Yes      Assigned    Server   00:11:22:33:44:55   VID 20 (vlan20)
eth2    Yes      Assigned    Server   00:aa:bb:cc:dd:ee   VID 40 (vlan40)
```

```
└──────────────────────────────────────────────────────────────────────────────┘

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

The VLAN and port status should now reflect the new VLAN assignments:

```
example:/#> show vlan
VID  NAME              TAGGED/UNTAGGED/POLICY NEST
   1 vlan1             T:
                       U:eth5
                       N:
  10 vlan10            T:
                       U:eth7
                       N:
  20 vlan20            T:
                       U:eth1, eth6
                       N:
  30 vlan30            T:
                       U:eth3
                       N:
  40 vlan40            T:
                       U:eth2, eth4
                       N:
example:/#>
example:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
eth1             Up        N/A     Unknown  Forwarding      vlan U:20
eth2             Up        N/A     Unknown  Forwarding      vlan U:40
eth3             Up        N/A     Unknown  Forwarding      vlan U:30
eth4             Up        N/A     Unknown  Forwarding      vlan U:40
eth5             Up        N/A     Unknown  Forwarding      vlan U:1
eth6             Up        N/A     Unknown  Forwarding      vlan U:20
eth7             Up        N/A     Unknown  Forwarding      vlan U:10
example:/#>
```

As shown above, eth1 has been dynamically reassigned to VLAN 20 (vlan20) and eth2 has been reassigned to VLAN 40 (vlan40), demonstrating how the same ports can be dynamically moved between different VLANs based on which supplicant authenticates on the host.

4.3.8. Dynamic VLAN Assignment with MAC Authentication Bypass (MAB) HowTo

Introduction

This document describes how dynamic VLAN assignment can be configured for MAC authentication bypass (MAB). For an overview of all available configuration settings for Port Access Control, refer to the configuration section in the Port Access Control Guide.

> **SEE ALSO**
>
> Related Documentation
> - Port Access Control Guide
> - MAC authentication using RADIUS (MAB) HowTo
> - Dynamic VLAN Assignment with 802.1X HowTo

Overview

This is a simplified example designed to provide a clear, basic demonstration of what needs to be configured on the WeOS device to get VLAN assignment working with port access control using MAC authentication bypass (MAB). The focus is on understanding the core configuration concepts rather than necessarily providing a complete, production-ready deployment. Real-world implementations may require additional security considerations, redundancy, and more complex network topologies.

For this use case example the following topology, or similar, is considered:

```
                                Authentication
                                    Server
                                .-------.         00:11:22:33:44:55 -> VLAN 10
                                |RADIUS|          00:aa:bb:cc:dd:ee -> VLAN 20
                                '--+---'
                                   | 10.0.1.2
                                   | VLAN 1
        |                          |                            |
    .--.-.                         |                        .--.-.
   ( (    )__                      |                       ( (    )__
  (_,  \ ) ,_)                     |                      (_,  \ ) ,_)
    '-'--`--'                      |                        '-'--`--'
        |                          |                            |
        |                          |                            |
        |                          | VLAN 1                     |
        |                          | 10.0.1.1                   |
        | 10.0.10.0/24 .------------+-------------. 10.0.20.0/24 |
        | VLAN 10      |            4             |     VLAN 20  |
        '--------------+ 5         WeOS         3 +--------------'
                       |    1                2    |
                       '----+----------------+----'
                            X    Start in    X    <== Controlled Ports
                            |     VLAN 1     |     (All Blocked by
                            |                |      Authenticator)
                         .--+-.            .-+--.
        00:11:22:33:44:55 | H1 |            | H2 | 00:aa:bb:cc:dd:ee
                         '----'            '----'
```

Figure 1: A basic setup of port access control using MAB with dynamic VLAN assignment. The ports eth1 and eth2 are controlled by port access on the authenticator device WeOS. The RADIUS server

is configured to handle MAB and provide a specific VLAN assignment for each MAC address of the connecting hosts H1 and H2.

> **PORT NAMES**
>
> The port numbers as presented in **Figure 1** will be referred to as eth1-eth5 in the configuration examples below, i.e. 1 -> eth1, 2 -> eth2, etc.

With this example setup the goal is to have the ports be assigned dynamically to different VLANs based on the MAC address of frames originating from the connected hosts. For example, when the WeOS device observes a frame from host H1 with MAC address **00:11:22:33:44:55** on port eth1, it will trigger a MAB authentication request, and upon successful authentication, the port should be assigned to VLAN 10. Similarly, when a frame from host H2 with MAC address **00:aa:bb:cc:dd:ee** is seen on port eth2, the port should be assigned to VLAN 20 after successful authentication.

In this case it is assumed that whatever RADIUS server that is used to process the authentication requests from the WeOS device is configured for MAC authentication bypass (MAB) and to return the correct VLAN assignment information for each MAC address. The RADIUS server must therefore be configured to supply the following, in addition to everything else that is required for the authentication to succeed:

```
Tunnel-Type = VLAN,
Tunnel-Medium-Type = 6,
Tunnel-Private-Group-Id = [numeric VLAN tag, e.g. 10 or 20]
```

It is this **Tunnel** information that is used by the WeOS device to determine which VLAN to assign the port to when a successful authentication occurs.

> **MULTIPLE TUNNEL GROUPS**
>
> A RADIUS server may include multiple sets of tunnel attributes in a single response, each identified by a Tag field (0–31) per RFC 2868. When multiple valid tunnel groups are present, the group with the **lowest Tag value** is selected for VLAN assignment. A tunnel group is only considered valid if `Tunnel-Type` is VLAN (13), `Tunnel-Medium-Type` is IEEE-802 (6), and `Tunnel-Private-Group-ID` contains a numeric VID in the range 1–4094. Since a port can only be assigned untagged to a single VLAN, only one tunnel group is ever used regardless of how many the server provides. If at least one `Tunnel-Type` attribute is present but no valid tunnel group can be formed, MAB will **reject the authentication request**. This differs from 802.1X, which instead treats such a response as if no VLAN information was provided.

> **ONLY ASSIGNABLE UNTAGGED**
>
> As described in the VLAN Assignment section in the Port Access Control Guide, ports can only be assigned untagged to VLANs when doing dynamic VLAN assignment. Further, this also means that a port can only ever be assigned to one VLAN at a time when doing dynamic VLAN assignment. Additionally, this means that it is the first authenticated MAC address that determines which VLAN the port is assigned.

Configuration

The first thing to do is to configure the base VLAN vlan1 and its associated interface. As mentioned in the topology description, the ports that face the hosts (eth1 and eth2) start in VLAN 1, and this is

where they will be moved back to when the hosts disconnect. In addition, the port on the WeOS device that faces the RADIUS server (eth4) is also in VLAN 1.

```
example:/#> config
example:/config/#> vlan 1
example:/config/vlan-1/#> untagged eth1,eth2,eth4
example:/config/vlan-1/#> end
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> inet static 10.0.1.1/24
example:/config/iface-vlan1/inet-static-10.0.1.1/#> end
example:/config/iface-vlan1/#> end
example:/config/#>
```

The VLANs that are to be assigned dynamically must also be configured. Both so that there is a port providing access to each VLAN and because any VLAN to be assigned dynamically must exist on the device, these will not be created automatically when the RADIUS server returns the VLAN assignment information, for more information, see the section on assignable VLANs in the Port Access Control Guide.

```
example:/config/#> vlan 10
example:/config/vlan-10/#> untagged eth5
example:/config/vlan-10/#> end
example:/config/#> vlan 20
example:/config/vlan-20/#> untagged eth3
example:/config/vlan-20/#> end
example:/config/#> iface vlan10
example:/config/iface-vlan10/#> inet static 10.0.10.1/24
example:/config/iface-vlan10/inet-static-10.0.10.1/#> end
example:/config/iface-vlan10/#> end
example:/config/#> iface vlan20
example:/config/iface-vlan20/#> inet static 10.0.20.1/24
example:/config/iface-vlan20/inet-static-10.0.20.1/#> end
example:/config/iface-vlan20/#> end
example:/config/#>
```

Next, the RADIUS server definition to be used by the port access instance is configured. The RADIUS server is located at IP address 10.0.1.2 and uses the default RADIUS port (1812) in this example.

```
example:/config/#> aaa
example:/config/aaa/#> remote-server 1
Creating new remote server 1
example:/config/aaa/remote-server-1/#> description MyRADIUS
example:/config/aaa/remote-server-1/#> address 10.0.1.2
example:/config/aaa/remote-server-1/#> password TopS3cret
example:/config/aaa/remote-server-1/#> end
example:/config/aaa/#> end
```

> **PERFORMANCE CONSIDERATION**
>
> For best performance, it is recommended to use IP address (rather than domain name) when defining the RADIUS server.

At this point the port access instance can be configured. The ports eth1 and eth2 are added as port access ports, and MAC authentication (MAB) is enabled for the instance.

```
example:/config/#> port-access
Creating new Port Access instance 1.
example:/config/port-access-1/#> port eth1,eth2
example:/config/port-access-1/#> description "VLAN Assignment Example"
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> mac-authentication
example:/config/port-access-1/mac-auth/#> authentication-server 1
example:/config/port-access-1/mac-auth/#> end
example:/config/port-access-1/#>
```

PORT METHOD

In this example the port method is explicitly set to `mac-based`, which is also the default setting. See Port Method for more information about port methods.

Finally the VLAN assignment is also enabled for the port access instance. It is enabled and the VLANs to allow are selected.

```
example:/config/port-access-1/#> vlan-assignment
example:/config/port-access-1/vlan-assignment/#> allowed-vlans vlan10 vlan20
example:/config/port-access-1/vlan-assignment/#> end
example:/config/port-access-1/#>
```

THE ALLOWED VLANS CAN ALSO BE SPECIFIED BY THEIR VID

The allowed VLANs can also be specified by their numeric VLAN tag, e.g. **10, 20** using the `allowed-vid` command instead of `allowed-vlans`. They manipulate the exact same setting, and can be used interchangeably:

```
example:/config/port-access-1/vlan-assignment/#> allowed-vid 10 20
example:/config/port-access-1/vlan-assignment/#> end
example:/config/port-access-1/#>
```

At this point the port access configuration should look something like:

```
Port Access General Settings
Status                  : Enabled
Description             : (No description)
Ports                   : eth1..eth2
Port Method             : MAC Based

802.1X Settings
Not configured.

MAC Authentication Settings
Status                  : Enabled

MAB
Authentication server   : 1 (MyRADIUS)
Service Type            : Call Check (10)

Local List
```

```
Accept list              : Not set ...

VLAN Assignment Settings
Status                   : Enabled
Allowed VIDs             : 10 (vlan10), 20 (vlan20)
Default VLAN             : Disabled
```

At this point the basic configuration is done, and the ports eth1 and eth2 should be able to authenticate and be assigned to the correct VLANs based on the MAC address of the connecting host and the information returned from the RADIUS server. Therefore, exit the configuration context to apply the configuration.

```
example:/config/port-access-1/#> leave
[*] System setup running ...
[*] System Ready ...
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

Using Separate Port Access Instances to Isolate Ports to Certain VLANs

In the example above a single port access instance was used for both ports eth1 and eth2, and it would be possible for both ports to be assigned to either VLAN 10 or VLAN 20 based on the MAC address of the connecting host. If the goal is to isolate the ports so that eth1 can only be assigned to VLAN 10, and eth2 can only be assigned to VLAN 20, two separate port access instances can be used, one for each port, and the allowed VLANs can be configured separately for each instance.

In such a case the configuration of port access instance 1 for port eth1 would look something like this:

```
example:/config/#> port-access 1
Creating new Port Access instance 1.
example:/config/port-access-1/#> port eth1
example:/config/port-access-1/#> description "Port 1 - VLAN 10 Only"
example:/config/port-access-1/#> port-method mac-based
example:/config/port-access-1/#> mac-authentication
example:/config/port-access-1/mac-auth/#> authentication-server 1
example:/config/port-access-1/mac-auth/#> end
example:/config/port-access-1/#> vlan-assignment
example:/config/port-access-1/vlan-assignment/#> allowed-vlans vlan10
example:/config/port-access-1/vlan-assignment/#> end
example:/config/port-access-1/#> end
example:/config/#>
```

And the configuration of port access instance 2 for port eth2 would look something like this:

```
example:/config/#> port-access 2
Creating new Port Access instance 2.
example:/config/port-access-2/#> port eth2
example:/config/port-access-2/#> description "Port 2 - VLAN 20 Only"
example:/config/port-access-2/#> port-method mac-based
example:/config/port-access-2/#> mac-authentication
example:/config/port-access-2/mac-auth/#> authentication-server 1
example:/config/port-access-2/mac-auth/#> end
example:/config/port-access-2/#> vlan-assignment
```

```
example:/config/port-access-2/vlan-assignment/#> allowed-vlans vlan20
example:/config/port-access-2/vlan-assignment/#> end
example:/config/port-access-2/#> end
example:/config/#>
```

With these two separate port access instances, port eth1 can now only be assigned to VLAN 10, and port eth2 can only be assigned to VLAN 20, based on the MAC address and the information returned from the RADIUS server. In this example only the port config and VLAN assignment settings are different between the two instances, but all the other settings can also be configured differently if needed, e.g. different authentication servers, different authentication methods, etc.

Status

This section demonstrates how to verify the port access and VLAN assignment status at different stages. The following examples show the output before any authentication takes place, after initial authentication when hosts with specific MAC addresses connect, and after the ports are reassigned when the hosts disconnect and reconnect.

Before Authentication

Before any host has attempted to authenticate, the status of the port access instance should look something like this:

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled

 802.1X Authentications
 #  MAC-ADDRESS        SUPPLICANT    PORT    SERVER          DATE-ADDED
 No authenticated 802.1X clients found.

 MAC Auth Authentications
 #  MAC-ADDRESS        PORT    METHOD                          DATE-ADDED
 No authenticated MAC addresses found.

 Ports
 PORT    METHOD      STATE                           AUTHENTICATED-MAC
 eth1    MAC Based   0 Authentications               -
 eth2    MAC Based   0 Authentications               -

 VLAN Assignment
 PORT    ENABLED  STATE       FROM     ASSIGNING-CLIENT   VLAN
 eth1    Yes      Unassigned  -        -                  -
 eth2    Yes      Unassigned  -        -                  -

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

In addition the ports eth1 and eth2 are in their starting VLAN (vlan1):

```
example:/#> show vlan
VID  NAME              TAGGED/UNTAGGED/POLICY NEST
   1 vlan1              T:
                       U: eth1..eth2, eth4
  10 vlan10             T:
                       U: eth5
  20 vlan20             T:
                       U: eth3
example:/#>
example:/#> show port
PORT              LINK      TYPE        SPEED   STATE           OWNER/VLAN
eth1              Up         N/A      Unknown   Forwarding      vlan U:1
eth2              Up         N/A      Unknown   Forwarding      vlan U:1
eth3              Up         N/A      Unknown   Forwarding      vlan U:20
eth4              Up         N/A      Unknown   Forwarding      vlan U:1
eth5              Up         N/A      Unknown   Forwarding      vlan U:10
example:/#>
```

After Authentication

Now let's assume that the WeOS device observes frames from host H1 with MAC address **00:11:22:33:44:55** on port eth1 and from host H2 with MAC address **00:aa:bb:cc:dd:ee** on port eth2. After successful MAB authentication, the ports eth1 and eth2 should be assigned to VLAN 10 and VLAN 20 respectively, and the status should look something like this:

```
example:/#> show port-access
Number of Port Access Instances: 1
Table Instance Dividers        : Disabled

 802.1X Authentications
 #  MAC-ADDRESS        SUPPLICANT     PORT     SERVER           DATE-ADDED
 No authenticated 802.1X clients found.

 MAC Auth Authentications
 #  MAC-ADDRESS        PORT     METHOD                          DATE-ADDED
 1  00:11:22:33:44:55  eth1     Server (10.0.1.2)          Mar 12 09:15:23
 2  00:aa:bb:cc:dd:ee  eth2     Server (10.0.1.2)          Mar 12 09:15:45

 Ports
 PORT     METHOD       STATE                            AUTHENTICATED-MAC
 eth1     MAC Based    1 Authentication                 00:11:22:33:44:55
 eth2     MAC Based    1 Authentication                 00:aa:bb:cc:dd:ee

 VLAN Assignment
 PORT     ENABLED  STATE        FROM     ASSIGNING-CLIENT   VLAN
 eth1     Yes      Assigned     Server   00:11:22:33:44:55   VID 10 (vlan10)
 eth2     Yes      Assigned     Server   00:aa:bb:cc:dd:ee   VID 20 (vlan20)

- Use 'show port-access audit' to display related events in the audit log.
- Use 'show port-access INSTANCE' to show details for a specific instance.
- Use 'port-access instance-dividers' to enable dividers between instances
  in the tables.
example:/#>
```

The VLAN and port status should now also reflect the new VLAN assignment for the ports:

```
example:/#> show vlan
VID  NAME              TAGGED/UNTAGGED/POLICY NEST
   1 vlan1             T:
                       U: eth4
  10 vlan10            T:
                       U: eth1, eth5
  20 vlan20            T:
                       U: eth2 .. eth3
example:/#>
example:/#> show port
PORT              LINK      TYPE        SPEED   STATE           OWNER/VLAN
eth1              Up         N/A      Unknown  Forwarding      vlan U:10
eth2              Up         N/A      Unknown  Forwarding      vlan U:20
eth3              Up         N/A      Unknown  Forwarding      vlan U:20
eth4              Up         N/A      Unknown  Forwarding      vlan U:1
eth5              Up         N/A      Unknown  Forwarding      vlan U:10
example:/#>
```

As you can see, eth1 has been dynamically assigned to VLAN 10 (vlan10) and eth2 has been assigned to VLAN 20 (vlan20) based on the authentication information received from the RADIUS server for their respective MAC addresses. The ports are now untagged members of their respective assigned VLANs instead of the initial VLAN 1.

> **REGARDING MULTIPLE MAC ADDRESSES ON THE SAME PORT**
>
> At this point the port's VLAN is more or less "locked" to the first authenticated MAC address. If another device with a different MAC address tries to authenticate on the same port, it will be authenticated, but the port will remain in the VLAN assigned by the first authenticated MAC address. This behaviour follows the MAC Based port method. For more information about port methods and their behaviour, see the Port Access Control Guide. This is particularly relevant in scenarios where a single port may have a switch or hub behind it with multiple devices, or where a device may change its MAC address and reconnect.

Ports eth1 and eth2 will now remain assigned to VLAN 10 and VLAN 20 respectively until the authenticated clients are deauthenticated, either by disconnecting, when the port is brought down, etc. When this happens the ports will be moved back to their starting VLANs, in this use case it is VLAN 1.

4.3.9. FRNT HowTo

About

This document details some common use-cases for FRNT and how to configure them.

For an overview of available configuration settings for FRNT refer to the configuration guide:

- FRNT Configuration Guide

Introduction

The Fast Reconfiguration Network Topology Protocol (FRNT) handles fast reconfiguration in switched ring topologies. One of the switches has the role of FRNT focal point while the other switches are referred to as FRNT members. When the switches are connected in a ring, it is the responsibility of the focal point to break the loop by putting one of its ports (port ”M”) in blocking mode. When choosing FRNT ports, fixed speed and full duplex is recommended and also to avoid copper SFPs.

For information about using FRNT together with DHCP functionality in order to create a redundant network setup with dynamic address allocation, see:

- DHCP Relay Server Ring

Single ring

The topology presented in this example consists of four different devices that create a redundant ring setup.

```
.-------.                .-------.
|       |M             N|       |
|  S1   +-||------------+  S2   |
|     fp|1             2|       |
'---+---'                '---+---'
   2|N                     1|M
    |         FRNT          |
   1|M                     2|N
.---+---.                .---+---.
|       |N             M|       |
|  S4   +---------------+  S3   |
|       |2             1|       |
'-------'                '-------'
```

Figure 1: FRNT network operating in ring mode.

Configuration

Set up the focal point:

```
S1:/#> configure
S1:/config/#> ring
S1:/config/ring/#> frnt
S1:/config/ring/frnt/#> ports eth1, eth2
S1:/config/ring/frnt/#> leave
S1:/#>
```

Set up rest of the devices (S2 - S4), which aren’t focal points

```
Sx:/#> configure
Sx:/config/#> ring
Sx:/config/ring/#> frnt
Sx:/config/ring/frnt/#> no focal
Sx:/config/ring/frnt/#> ports eth1, eth2
```

```
Sx:/config/ring/frnt/#> leave
Sx:/#>
```

> **NOTE**
>
> Double ring is not yet supported

Bus

The main use case for configuring an FRNT bus is to achieve a horseshoe topology, by using the FRNT bus together with Ring Coupling. On the bus edges, only use port M.

```
.-------.            .-------.            .-------.            .-------.
|       |M          N|       |M          N|       |M          M|       |
|  S1   +-||---------+  S2   +------------+  S3   +------------+  S4   |
|     fp|1          2|       |1          2|       |1          2|       |
'-------'            '-------'            '-------'            '-------'
```

Figure 2: FRNT network operating in bus mode.

Configuration

Set up the focal point:

```
S1:/#> configure
S1:/config/#> ring
S1:/config/ring/#> frnt
S1:/config/ring/frnt/#> ports eth1
S1:/config/ring/frnt/#> leave
S1:/#>
```

Set up rest of the devices (S2 - S3), which aren't focal points

```
Sx:/#> configure
Sx:/config/#> ring
Sx:/config/ring/#> frnt
Sx:/config/ring/frnt/#> no focal
Sx:/config/ring/frnt/#> ports eth1, eth2
Sx:/config/ring/frnt/#> leave
Sx:/#>
```

Lastly set up the device that is at the end of the bus:

```
S4:/#> configure
S4:/config/#> ring
S4:/config/ring/#> frnt
S4:/config/ring/frnt/#> no focal
S4:/config/ring/frnt/#> ports eth2
S4:/config/ring/frnt/#> leave
S4:/#>
```

4.3.10. MRP HowTo

About

This document presents an example use-case where a basic MRP ring is configured.

- MRP Configuration Guide

Introduction

The Media Redundancy Protocol (MRP) is useful when redundancy is wanted on Layer 2 networks. What this entails is setting up a topology of switches in a ring, where MRP is used to ensure that the devices in the ring are reachable.

This is done by making sure that no broadcast storm will occur while the ring is fully connected, by blocking one of the ring ports on the Manager (MRM). Further, if there is a break in the ring the blocking ring port on the MRM needs to be opened to ensure the reachability of all the devices in the ring.

```
 .-------.                                  .-------.
 |       | eth1                        eth1 |       |
 |  MRM  +----------------------------------+ MRC-1 |
 |       |                                  |       |
 '---+---'                                  '---+---'
eth2 | vlan1                               eth2 | vlan1
     | IP: 10.0.0.1                             | IP: 10.0.0.2
     | Untagged: eth1,eth2                      | Untagged: eth1,eth2
     |                                          |
     |                                          |
     |                                          |
     |                                          |
     | vlan1                                    | vlan1
     | IP: 10.0.0.4                             | IP: 10.0.0.3
eth2 | Untagged: eth1,eth2                 eth2 | Untagged: eth1,eth2
 .---+---.                                  .---+---.
 |       | eth1                        eth1 |       |
 | MRC-3 +----------------------------------+ MRC-2 |
 |       |                                  |       |
 '-------'                                  '-------'
```

Figure 1: A simple ring configured with MRP as redundancy.

All the devices **MRM**, **MRC-1**, **MRC-2** and **MRC-3** have their eth1 and eth2 ports configured as their MRP ring ports.

Configuring the MRP Ring

When configuring a MRP ring only one of the devices can serve as the Media Redundancy Manager (MRM) and the rest of the devices will serve as Media Redundancy Clients (MRC).

> **TIP**
>
> When configuring the ring its good to ensure that there is no closed loop while the ring is being set up. This could be done by bringing down the ports or unplugging the cable so that there is no loop. Otherwise, it is likely that a broadcast storm could occur.

MRM Configuration

```
mrm:/#> configure
mrm:/config/#> ring mrp 1
mrm:/config/ring/mrp-1/#> mode manager
mrm:/config/ring/mrp-1/#> profile 200
mrm:/config/ring/mrp-1/#> ports eth1, eth2
```

```
mrm:/config/ring/mrp-1/#> leave
mrm:/#>
```

MRCs Configuration

```
mrc-x:/#> configure
mrc-x:/config/#> ring mrp 1
mrc-x:/config/ring/mrp-1/#> profile 200
mrc-x:/config/ring/mrp-1/#> ports eth1, eth2
mrc-x:/config/ring/mrp-1/#> leave
mrc-x:/#>
```

Status

When MRP have been configured on all the devices we can check the status of the MRP ring.

```
mrm:/#> show mrp 1
Ring ID              : 1
Enabled              : Yes running as PID 2233
Mode                 : MRM
Status               : Closed (CHK_RC)
Profile              : 200
Port 1               : eth1 BLOCKED (UP)
Port 2               : eth2 FORWARDING (UP)
VLAN ID              : 0
React on link change : Enabled
Transitions          : 1
Last topology change : 0 Days 0 Hours 0 Mins 10 Secs
MRM Src Addr         : 74:69:6e:67:20:6d
```

Test Connections

In order to verify if all the devices in the node are reachable we can ping the addresses of the vlans that the MRP ring ports are assigned to:

```
mrm:/#> ping 10.0.0.2
Press Ctrl-C to abort PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: seq=0 ttl=64 time=1.903 ms
[ ... ]

mrm:/#> ping 10.0.0.3
Press Ctrl-C to abort PING 10.0.0.3 (10.0.0.3): 56 data bytes
64 bytes from 10.0.0.3: seq=0 ttl=64 time=1.843 ms
[ ... ]

mrm:/#> ping 10.0.0.4
Press Ctrl-C to abort PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: seq=0 ttl=64 time=1.701 ms
[ ... ]
```

The same could be repeated after breaking a link or bringing down a port so that the ring must reconverge. All the devices should still be accessible on the same addresses.

4.3.11. MRP Over Layer 2 Tunnel

About

This document presents an example use-case where a MRP ring is configured and part of the ring is connected over a Layer 2 OpenVPN tunnel.

- MRP Configuration Guide

Introduction

This is a setup that could be used where a Layer 2 redundancy in the form of a ring topology is wanted, but where part of the ring topology needs to be connected over another network. A Layer 2 ring topology would not really be possible in this kind of situation, but if it is combined with a Layer 2 OpenVPN tunnel it becomes a possible setup.

```
                    Public Network/Internet
                            .--.-.
                           ( (    )__
      .----------------(_,  \ ) ,_)---------------.
      |                   '-'--`--'                |
      |                                            |
      | vlan2                                      | vlan2
      | IP: 198.18.10.1                            | IP: 198.18.10.2
eth1  | Untagged: eth1                       eth1  | Untagged: eth1
 .-------.                                    .-------.
 |       | ssl0                         ssl0  |       |
 |  MRM  +==================================+ MRC-1 |
 |       |         OpenVPN L2 Tunnel          |       |
 '---+---'                                    '---+---'
eth2  | vlan1                                eth2  | vlan1
      | IP: 10.0.0.1                               | IP: 10.0.0.2
      | Untagged: ssl0,eth2                        | Untagged: ssl0,eth2
      |                                            |
      |                                            |
      |                                            |
      |                                            |
      | vlan1                                      | vlan1
      | IP: 10.0.0.4                               | IP: 10.0.0.3
eth2  | Untagged: eth1,eth2                  eth2  | Untagged: eth1,eth2
 .---+---.                                    .---+---.
 |       | eth1                         eth1  |       |
 | MRC-3 +----------------------------------+ MRC-2 |
 |       |                                    |       |
 '-------'                                    '-------'
```

Figure 1: A MRP ring configured normally, except for the link between the MRM and MRC-1, that instead uses a L2 tunnel to connect as ring ports.

The devices **MRM** and **MRC-1** both have their eth1 ports connected to the WAN (Public network/Internet). The Layer 2 Tunnel will be established over this connection. In this example it is also assumed that an OpenVPN key has already been generated and imported to both the **MRM** and **MRC-1** and labelled *psk1*. Both devices have their eth2 and ssl0 ports configured as their MRP ring ports.

The devices **MRC-2** and **MRC-3** both have their eth1 and eth2 ports configured as their MRP ring ports.

> **NOTE**
>
> This example assumes that the vlan interfaces 1 and 2 have already been configured on the devices in the example.

Configuring the Layer 2 Tunnel

In this setup the **MRM** device will serve as the OpenVPN server node and the **MRC-1** will be configured as a client and peer towards it.

Server Configuration

The **MRM** device can be configured as a OpenVPN server node in the following manner.

```
mrm:/#> configure
mrm:/config/#> tunnel ssl 0
mrm:/config/tunnel/ssl-0/#> server
mrm:/config/tunnel/ssl-0/#> method psk
mrm:/config/tunnel/ssl-0/#> secret psk1
mrm:/config/tunnel/ssl-0/#> type tap
mrm:/config/tunnel/ssl-0/#> leave
mrm:/#>
```

The Layer 2 tunnel will have a virtual port interface *ssl0* created on the device. This is the port interface we will use as one of the ring ports later when configuring the MRP instance.

Remember to add the newly created *ssl0* port to a vlan that is intended to be used for the MRP ring ports. This must be done because the port interface must be part of the bridge, and adding it to a vlan will make it part of the bridge.

```
mrm:/#> configure
mrm:/config/#> vlan 1
mrm:/config/tunnel/vlan-1/#> untagged ssl0
mrm:/config/tunnel/ssl-0/#> leave
mrm:/#>
```

In this case we also need to ensure that we enable flooding to the CPU of unknown packets, because we combine virtual port interfaces with physical. Therefore, this needs to be enabled if traffic should be handled as you would expect. Enable it in the following manner:

```
mrm:/#> configure
mrm:/config/#> system
mrm:/config/system/#> flooding unicast multicast
mrm:/config/system/#> leave
mrm:/#>
```

Client Configuration

The **MRC-1** device can be configured as OpenVPN client that peers against the server in the following way:

```
mrc-1:/#> configure
mrc-1:/config/#> tunnel ssl 0
mrc-1:/config/tunnel/ssl-0/#> no server
mrc-1:/config/tunnel/ssl-0/#> method psk
mrc-1:/config/tunnel/ssl-0/#> secret psk1
mrc-1:/config/tunnel/ssl-0/#> type tap
mrc-1:/config/tunnel/ssl-0/#> peer 198.18.10.1
mrc-1:/config/tunnel/ssl-0/#> leave
mrc-1:/#>
```

The client will now attempt to peer against the server and establish the Layer 2 tunnel connection.

© 2026 Westermo Network Technologies AB

Same as with the server, we add the *ssl0* port interface to the vlan that we want to use for our MRP ring ports.

```
mrc-1:/#> configure
mrc-1:/config/#> vlan 1
mrc-1:/config/tunnel/vlan-1/#> untagged ssl0
mrc-1:/config/tunnel/ssl-0/#> leave
mrc-1:/#>
```

Again as we did on the server, we need to enable flooding to the CPU of unknown packets:

```
mrc-1:/#> configure
mrc-1:/config/#> system
mrc-1:/config/system/#> flooding unicast multicast
mrc-1:/config/system/#> leave
mrc-1:/#>
```

Status

The status of the tunnel can be checked by either checking the port status, since the tunnel endpoint ssl0 is treated as a port:

```
mrm:/#> show port
PORT             LINK     TYPE       SPEED  STATE           OWNER/VLAN
[ ... ]
ssl0             Up        SSL              Forwarding      vlan U:1
```

We can also check the status of the tunnel connection in the following way:

```
mrm:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
0        ssl0            UP       0 Days 2 Hours 30 Mins 44 Secs
```

> **NOTE**
>
> Remember that it could take a few seconds for the tunnel to establish its connection and for the tunnel status to be displayed as *UP*.

Adjust keepalive For Better MRP Reconfiguration

If the tunnel connection goes down it will take a while for it to notice that the connection is down, so the ssl port can be in forwarding when it is really down. This could potentially cause some instability in the MRP ring if the tunnel link goes up and down.

In order to make the tunnel more faster in detecting connection issues the *keepalive* setting can be adjusted:

```
mrm:/#> configure
mrm:/config/#> tunnel ssl 0
mrm:/config/tunnel/ssl-0/#> keepalive interval 1 restart 2
mrm:/config/tunnel/ssl-0/#> leave
mrm:/#>
```

```
mrc-1:/#> configure
mrc-1:/config/#> tunnel ssl 0
mrc-1:/config/tunnel/ssl-0/#> keepalive interval 1 restart 2
mrc-1:/config/tunnel/ssl-0/#> leave
mrc-1:/#>
```

Doing this could help make the MRP connection better if link changes occur in the tunnel part of the ring.

> **NOTE**
>
> This issue could be better solved with a link check protocol like IEEE 802.1ag CFM (Connectivity Fault Management). Unfortunately it is currently not available in the product.

Testing Connection

We can verify the connection by simply attempting to ping the address of the interface that is associated with the *ssl0* port interface. In this example we added the *ssl0* interface to vlan1 that has the ip address of *10.0.0.2* on the **MRC-1**. Therefore, if the tunnel have been established correctly we should be able to ping that address from the **MRM** device:

```
mrm:/#> ping 10.0.0.2
Press Ctrl-C to abort PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: seq=0 ttl=64 time=1.903 ms
64 bytes from 10.0.0.2: seq=1 ttl=64 time=0.925 ms
[ ... ]
```

Configuring the MRP Ring

When configuring a MRP ring only one of the devices can serve as the Media Redundancy Manager (MRM) and the rest of the devices will serve as Media Redundancy Clients (MRC). The MRM is responsible for ensuring that there will be no loop in the network by blocking one of its ring ports. If there is a link break somewhere in the ring, it is also the MRM that will ensure that all nodes in the ring will still be reachable by putting its blocking ring port into forwarding.

> **TIP**
>
> When configuring the ring its good to ensure that there is no closed loop while the ring is being set up. This could be done by bringing down the ports or unplugging the cable so that there is no loop. Otherwise, it is likely that a broadcast storm could occur.

MRM Configuration

```
mrm:/#> configure
mrm:/config/#> ring mrp 1
mrm:/config/ring/mrp-1/#> mode manager
mrm:/config/ring/mrp-1/#> profile 200
mrm:/config/ring/mrp-1/#> ports ssl0, eth2
mrm:/config/ring/mrp-1/#> leave
mrm:/#>
```

MRC-1 Configuration

```
mrc-1:/#> configure
mrc-1:/config/#> ring mrp 1
mrc-1:/config/ring/mrp-1/#> profile 200
mrc-1:/config/ring/mrp-1/#> ports ssl0, eth2
mrc-1:/config/ring/mrp-1/#> leave
mrc-1:/#>
```

MRC-2 Configuration

```
mrc-2:/#> configure
mrc-2:/config/#> ring mrp 1
mrc-2:/config/ring/mrp-1/#> profile 200
mrc-2:/config/ring/mrp-1/#> ports eth1, eth2
mrc-2:/config/ring/mrp-1/#> leave
mrc-2:/#>
```

MRC-3 Configuration

```
mrc-3:/#> configure
mrc-3:/config/#> ring mrp 1
mrc-3:/config/ring/mrp-1/#> profile 200
mrc-3:/config/ring/mrp-1/#> ports eth1, eth2
mrc-3:/config/ring/mrp-1/#> leave
mrc-3:/#>
```

Status

When MRP have been configured on all the devices we can check the status of the MRP ring.

> **TIP**
>
> It is best to check the status on the MRM. It is possible to check the status on the MRCs, but the MRM will be more accurate. Since the MRCs obtain some of its status information from the *MRP Test* frames that are sent from the MRM, while the MRM have direct access to the status information.

The status of the MRP instance can be observed in the following manner:

```
mrm:/#> show mrp 1
Ring ID              : 1
Enabled              : Yes running as PID 2233
Mode                 : MRM
Status               : Closed (CHK_RC)
Profile              : 200
Port 1               : ssl0 BLOCKED (UP)
Port 2               : eth2 FORWARDING (UP)
VLAN ID              : 0
React on link change : Enabled
Transitions          : 1
Last topology change : 0 Days 0 Hours 0 Mins 10 Secs
MRM Src Addr         : 74:69:6e:67:20:6d
```

Test Connections

In order to verify if all the devices in the node are reachable we can ping the addresses of the vlans that the MRP ring ports are assigned to:

```
mrm:/#> ping 10.0.0.2
Press Ctrl-C to abort PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: seq=0 ttl=64 time=1.903 ms
[ ... ]

mrm:/#> ping 10.0.0.3
Press Ctrl-C to abort PING 10.0.0.3 (10.0.0.3): 56 data bytes
64 bytes from 10.0.0.3: seq=0 ttl=64 time=1.843 ms
[ ... ]

mrm:/#> ping 10.0.0.4
Press Ctrl-C to abort PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: seq=0 ttl=64 time=1.701 ms
[ ... ]
```

The same could be repeated after breaking a link or bringing down a port so that the ring must reconverge. All the devices should still be accessible on the same addresses.

4.3.12. IEEE 802.1Q Tunneling

About

This application note describes how to configure IEEE 802.1Q VLAN tunneling (Q-in-Q) in WeOS. The concept can be used over any core network where VLAN trunking is already used, regardless of any underlying layer-2 redundancy protocol; e.g. RSTP, FRNT, MRP.

Figure 1 shows the basic setup that will be used in the document. It depicts two customers, which use identical VLANs on each respective network, that want to connect their two sites over a shared provider network.

Introduction

Throughout this document we will use terms like *customer* and *service provider*. Customers connect to *access ports* on the provider network. The provider network has *service ports*, which are VLAN trunks.

The chosen example may at first glance only apply to traditional Internet Service Providers (ISP). However, the same model can also be applied to industrial networks where isolation and tunneling over a shared network infrastructure is required.

Please note, this document describes Q-in-Q using stacked IEEE 802.1Q VLAN tags, not provider tagging according to IEEE 802.1ad, which may also be referred to as Q-in-Q.

In IEEE 802.1Q tunneling we encapsulate frames by adding an .1Q tag to all ingressing frames. The new tag is added before the first one, to ensure that simpler layer-2 switches use the encapsulating outer tag. When adding a new (outer) VLAN we *push* a tag, when frames are decapsulated we *pop* a tag.

```
.-------------.                    .--------------.
|             |                    |              |
| Customer A1 |                    |  Customer A2 |
| VLANs 1..60 |                    |  VLANs 1..60 |
|             |                    |              |
|    eth1     |                    |     eth1     |
'------+------'                    '-------+------'
       |                                   |
       |      Customer A Network           |
.~- .~-|.~- .~- .~- .~- ..~- .~- .~- .~-~-| .~- .~-
       |      Provider Network             |
       |                                   |
.------+--- ---.                   .-------+------.
|    eth1      |                   |      eth1    |
|              |eth3         eth3 |              |
|   Switch C1  +===================+  Switch C2   |
|              |   VLAN 10, 20     |              |
|    eth2      |                   |     eth2     |
'------+--- ---'                   '-------+------'
       |                                  |
       |                                  |
.~- .~- .~- .~- ..~- .~- .~- .~- .~- .~-.~-|.~- .~- .~-
       |        CustomerB Network         |
       |                                  |
.------+------.                   .-------+------.
|     eth1    |                   |      eth1    |
|             |                   |              |
| Customer B1 |                   |  Customer B2 |
| VLANs 1..90 |                   |  VLANs 1..90 |
|             |                   |              |
'-------------'                   '--------------'
            --------- VLAN Trunk
            ========= Tunnel connection
```

Figure 1: Q-in-Q Tunneling Setup Overview.

A VLAN tagged frame, figure 1, encapsulate the frame from the EtherType/Length field. The C-VID naming used here refer to the leading two bytes 8100 of the VLAN tag, which identifies the frame as an IEEE 802.1Q VLAN EtherType. Here VLAN ID (VID) between 1- 60 for customer A and between 1-90 for customer B is used.

Ingress Policy

This allows the customer to use any VLAN ID on their internal network. All ingressing traffic on these access ports will be encapsulated with a second VLAN tag, VID 10 and 20, respectively. The encapsulating VLAN is the default vlan of the access port.

Egress Policy

This allows the traffic egressing on access port to restore to its customer VID (C-VID), the service vlan ID (VID 10 and 20) will be removed from the frame and the packet will egress with C-VID tag.

Configuration

Switch C1

```
C1:/#> configure
C1:/config/#> vlan 10
C1:/config/vlan-10/#> untagged eth1
C1:/config/vlan-10/#> nest eth1
C1:/config/vlan-10/#> tagged eth3
C1:/config/#> vlan 20
C1:/config/vlan-20/#> untagged eth2
C1:/config/vlan-20/#> nest eth2
C1:/config/vlan-20/#> tagged eth3
C1:/config/#>
```

Switch C2

```
C2:/#> configure
C2:/config/#> vlan 10
C2:/config/vlan-10/#> untagged eth1
C2:/config/vlan-10/#> nest eth1
C2:/config/vlan-10/#> tagged eth3
C2:/config/#> vlan 20
C2:/config/vlan-20/#> untagged eth2
C2:/config/vlan-20/#> nest eth2
C2:/config/vlan-20/#> tagged eth3
C2:/config/#>
```

Customer A1, A2, B1, B2

Customer VID (CVID) can be any value from 1 to 4095

```
example:/#> configure
example:/config/#> vlan CVID
example:/config/vlan-CVID/#> tagged eth1
example:/config/#>
```

Tunnel Isolation

To ensure complete isolation of the Q-in-Q tunnels, we remove the core switch CPU connection from each of the customer VLANs:

Switch C1

```
C1:/#> configure
C1:/config/#> iface vlan10
C1:/config/vlan-10/#> no enable
C1:/config/vlan-10/#> end
C1:/config/#> iface vlan20
C1:/config/vlan-20/#> no enable
C1:/config/vlan-20/#> end
C1:/config/#>
```

Switch C2

```
C2:/#> configure
C2:/config/#> iface vlan10
C2:/config/vlan-10/#> no enable
C2:/config/vlan-10/#> end
C2:/config/#> iface vlan20
C2:/config/vlan-20/#> no enable
C2:/config/vlan-20/#> end
C2:/config/#>
```

Layer-2 Services

> **NOTE**
>
> services like IGMP snooping, LLDP, etc. are currently not supported for such VLAN configurations and should be disabled.

The tunnel isolation step (above) only disables the switch CPU layer-3 connection, we also need to disable WeOS default layer-2 services like

IGMP snooping:

```
example:/#> configure
example:/config/#> vlan 10
example:/config/vlan-10/#> no igmp
example:/config/vlan-10/#> end
example:/config/#> vlan 20
example:/config/vlan-20/#> no igmp
example:/config/vlan-20/#> end
example:/config/#>
```

And LLDP

```
example:/config/#> lldp
example:/config/lldp/#> port eth1,eth2,eth3
example:/config/lldp/port-eth1..3/#> no enable
example:/config/lldp/port-eth1..3/#> leave
example:/#>
```

> **NOTE**
>
> services like IGMP snooping, LLDP, etc. are currently not supported for such VLAN configurations and should be disabled.

4.3.13. HSR and PRP HowTo guide

This document, being a non-exhaustive guide, outlines some practical examples of how to configure and use High-Availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP).

For an overview of what HSR/PRP is and how it may be used, refer to:

- High-Availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP)

Westermo RedBox supports the types `hsr` (HSR-SAN), `prp` (PRP-SAN), and `hsr-prp` (HSR-PRP). Set the RedBox type with the command `type`. The ports are automatically selected, as the RedBox has dedicated ports for A, B, and interlink. The ports are labelled as such on the front plate.

```
redbox:/config/hsr-prp-1/#> show
Enabled                 : YES
Type                    : HSR
Mode                    : H
Supervision frames      : YES
Life check interval     : 2 s
Entry forget time       : 6 (640 ms)
Node reboot interval    : 500 ms
Proxy node forget time  : 64 s
Node forget time        : 64 s
Port A                  : ethA
Port B                  : ethB
Multicast octet         : 00
```

Basic HSR setup

A basic HSR setup only requires creating an instance of the type.

```
redbox:/#> config hsr-prp 1
redbox:/config/hsr-prp-1/#> type hsr
redbox:/config/hsr-prp-1/#> leave
redbox:/#>
```

Figure 1 shows a basic HSR ring setup. The ring can be expanded with more HSR devices, and provides seamless failover on link failure.

```
      +---------+
      |         |
      |   SAN   |
      |         |
      +----+----+
           |
           |
      +----I----+
      |         |
      | redbox  |
+-----A HSR-SAN B------+
|     |         |      |
|     +---------+      |
|                      |
|                      |
|     +---------+      |
|     |         |      |
+-----A redbox  B------+
      | HSR-SAN |
      |         |
      +----I----+
           |
           |
      +----+----+
      |         |
      |   SAN   |
      |         |
      +---------+
```

Figure 1: Two SAN devices connected redundantly via two HSR-SAN RedBoxes.

HSR-PRP coupling setup

Figure 2 shows a setup with three of the possible RedBox modes. LAN A and LAN B may contain other PRP and non-PRP devices that have redundancy through the ring.

```
                  +---------+
                  |         |
                  |   SAN   |
                  |         |
                  +----+----+
                       |
                       |
                  +----I----+
                  |redbox-1 |
                  |         |
            +-----A PRP-SAN B------+
            |     |         |      |
            |     +---------+      |
            |                      |
      +~~~~~~~~~~~~~+       +~~~~~~~~~~~~~+
      |             |       |             |
      |             |       |             |
      |   LAN A     |       |    LAN B    |
      |             |       |             |
      |             |       |             |
      +~~~~~~~~~~~~~+       +~~~~~~~~~~~~~+
            |                      |
            |                      |
            |                      |
      +-----I---+             +----I----+
      |redbox-2 |             |redbox-3 |
      |         |             |         |
      | HSR-PRP A-------------A HSR-PRP |
      |         |             |         |
      +-----B---+             +----B----+
            |                      |
            |     +---------+      |
            |     |redbox-4 |      |
            +-----A         B------+
                  | HSR-SAN |
                  |         |
                  +----I----+
                       |
                       |
                  +----+----+
                  |         |
                  |   SAN   |
                  |         |
                  +---------+
```

Figure 2: A PRP-SAN RedBox connects a SAN redundantly to an HSR ring, with another SAN device attached at the other side of the ring.

Note: The configuration in Figure 2 is not valid if redbox-2 and redbox-3 are PTP Transparent Clocks, as IEC 62439-3:2016 does not specify how a TC should behave in such a situation.

Configure the LAN ID of the HSR-PRP interlink ports with the respective LAN they connect to, and the Net ID (PRP ID) to the same for both because those A and B are associated with each other. The Net ID prevents frames from LAN A from being sent to LAN B, and vice versa.

A completely separate PRP network also connected to the ring must have another Net ID, as those frames may still enter other PRP networks. The standard specifies a total of 6 possible PRP networks in a single topology (values 1-6, HSR-SAN devices like redbox-4 use the value 0, 7 is reserved for future use).

Below is the configuration for each of the devices in Figure 2.

```
redbox-1:/#> config hsr-prp 1
redbox-1:/config/hsr-prp-1/#> type prp
redbox-1:/config/hsr-prp-1/#> leave
redbox-1:/#>
```

```
redbox-2:/#> config hsr-prp 1
redbox-2:/config/hsr-prp-1/#> type hsr-prp
redbox-2:/config/hsr-prp-1/#> lan-id A
redbox-2:/config/hsr-prp-1/#> prp-id 1
redbox-2:/config/hsr-prp-1/#> leave
redbox-2:/#>
```

```
redbox-3:/#> config hsr-prp 1
redbox-3:/config/hsr-prp-1/#> type hsr-prp
redbox-3:/config/hsr-prp-1/#> lan-id B
redbox-3:/config/hsr-prp-1/#> prp-id 1
redbox-3:/config/hsr-prp-1/#> leave
redbox-3:/#>
```

```
redbox-4:/#> config hsr-prp 1
redbox-4:/config/hsr-prp-1/#> type hsr
redbox-4:/config/hsr-prp-1/#> leave
redbox-4:/#>
```

Multiple rings

Each PRP LAN can connect to multiple rings for more redundancy.

```
+~~~~~~~~+      +~~~~~~~~+
| LAN A |       | LAN B |
| Net 1 |       | Net 1 |
+~~~~~~~~+      +~~~~~~~~+
    |    \    /    |
    |     \  /     |
    |      \/      |
    |      /\      |
    |     /  \     |
    |    /    \    |
+~~~~~~~~+      +~~~~~~~~+
|       |       |       |
|Ring 1 |       |Ring 2 |
|       |       |       |
+~~~~~~~~+      +~~~~~~~~+
```

Figure 3: PRP LANs connected to two separate rings.

4.3.14. Ring with two nodes HowTo

This document describes how to configure two nodes in a ring. See Figure 1.

```
        .-------.
        |       |         dir1 = (eth3)
<-dir1-+ RICH1 +-dir2->  dir2 = (eth4)
        |       |
        '-+---+-'
          |   |
        .-+---+-.
        |       |         dir1 = (eth3)
<-dir1-+ RICH2 +-dir2->  dir2 = (eth4)
        |       |
        '-------'
```

Figure 1: A ring with two nodes.

NOTE: In order to get two nodes to manage a single ring, they both need to have the exact same configuration with respect to **iface** and **uuid** in the ring chaining context.

```
RICH1:#> configure
RICH1:/config/#> iface vlan10
RICH1:/config/iface-vlan10/#> no inet
RICH1:/config/iface-vlan10/#> inet static 10.0.0.1/18
RICH1:/config/iface-vlan10/inet-static-10.0.0.1/#> end
RICH1:/config/iface-vlan10/#> end
RICH1:/config/#> ring
RICH1:/config/ring/#> frntv2 1
RICH1:/config/ring/frnt-1/#> port eth1
RICH1:/config/ring/frnt-1/port-eth1/#> end
RICH1:/config/ring/frnt-1/#> port eth2
RICH1:/config/ring/frnt-1/port-eth2/#> end
RICH1:/config/ring/frnt-1/#> end
RICH1:/config/ring/#>
RICH1:/config/ring/#> coupling 1
RICH1:/config/ring/coupling-1/#> coupling-port eth3
RICH1:/config/ring/coupling-1/coupling-port-eth3/#> priority 100
RICH1:/config/ring/coupling-1/coupling-port-eth3/#> end
RICH1:/config/ring/coupling-1/#> end
RICH1:/config/ring/#> coupling 2
RICH1:/config/ring/coupling-2/#> coupling-port eth4
RICH1:/config/ring/coupling-2/coupling-port-eth4/#> priority 100
RICH1:/config/ring/coupling-2/coupling-port-eth4/#> end
RICH1:/config/ring/coupling-2/#> end
RICH1:/config/ring/#> chaining
Activating RiCh with default settings.
RICH1:/config/ring/chaining/#> uuid 11111111-1111-1111-1111-111111111111
RICH1:/config/ring/chaining/#> address 225.1.2.3
RICH1:/config/ring/chaining/#> iface vlan10
RICH1:/config/ring/chaining/#> dir1 1
RICH1:/config/ring/chaining/#> dir2 2
RICH1:/config/ring/chaining/#> mode active
RICH1:/config/ring/chaining/#> end
RICH1:/config/ring/#> end
RICH1:/config/#> end
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
RICH1:/#>
```

Repeat the same steps for RICH2 with a lower priority for the coupling ports to make them work as backups.

To enable SNMP-traps create an action and connect a trigger to it according to the example below.

```
RICH1:#> configure
RICH1:/config/#> alarm
RICH1:/config/alarm/#> action
RICH1:/config/alarm/action-3/#> target snmp
RICH1:/config/alarm/action-3/#> end
RICH1:/config/alarm/#> trigger RiCh
RICH1:/config/alarm/trigger-3/#> action 3
RICH1:/config/alarm/trigger-3/#> end
RICH1:/config/alarm/#> end
RICH1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
RICH1:/#>
```

4.3.15. A ring with six nodes HowTo

This document describes how to configure six nodes in a ring.

See Figure 1 below

```
        .-------.   .-------.   .-------.
        |       |   |       |   |       |
<-dir1-+ RICH1 +---+ FRNT1 +---+ RICO1 +-dir2->
        |       |   |       |   |       |
        '---+---'   '-------'   '---+---'
            |                       |
        .---+---.   .-------.   .---+---.
        |       |   |       |   |       |
<-dir1-+ RICO2 +---+ FRNT2 +---+ RICH2 +-dir2->
        |       |   |       |   |       |
        '-------'   '-------'   '-------'
                  dir1 = (eth3)
                  dir2 = (eth4)
```

Figure 1: A single ring with six nodes.

It is enough to have one node running an instance of RiCh in active mode. However, in this example there are two nodes (RICH1, RICH2) that are configured with RiCh mode active.

- **RICH** - Nodes that have a FRNTv2 instance, 2 RiCo instances and RiCh running as active.
- **RICO** - Nodes that have a FRNTv2 instance, 2 RiCo instances and RiCh running as passive.
- **FRNT** - Nodes that have a FRNTv2 instance running.

NOTE: In order to configure a ring with six nodes, all extremity nodes need to have the exact same configuration with respect to **iface** and **uuid** in the `ring chaining context`. Other nodes in the ring, here FRNT1 and FRNT2, only needs to be configured with the same iface.

```
RICH1:#> configure
RICH1:/config/#> iface vlan10
RICH1:/config/iface-vlan10/#> no inet
RICH1:/config/iface-vlan10/#> inet static 10.0.0.1/18
RICH1:/config/iface-vlan10/inet-static-10.0.0.1/#> end
RICH1:/config/iface-vlan10/#> end
RICH1:/config/#> ring
RICH1:/config/ring/#> frntv2 1
RICH1:/config/ring/frnt-1/#> port eth1
RICH1:/config/ring/frnt-1/port-eth1/#> end
RICH1:/config/ring/frnt-1/#> port eth2
RICH1:/config/ring/frnt-1/port-eth2/#> end
RICH1:/config/ring/frnt-1/#> end
RICH1:/config/ring/#> coupling 1
RICH1:/config/ring/coupling-1/#> coupling-port eth3
RICH1:/config/ring/coupling-1/coupling-port-eth3/#> priority 100
RICH1:/config/ring/coupling-1/coupling-port-eth3/#> end
RICH1:/config/ring/coupling-1/#> end
RICH1:/config/ring/#> coupling 2
RICH1:/config/ring/coupling-2/#> end
RICH1:/config/ring/#> chaining
Activating RiCh with default settings.
RICH1:/config/ring/chaining/#> uuid 11111111-1111-1111-1111-111111111111
RICH1:/config/ring/chaining/#> iface vlan10
RICH1:/config/ring/chaining/#> address 225.1.2.3
RICH1:/config/ring/chaining/#> dir1 1
RICH1:/config/ring/chaining/#> dir2 2
RICH1:/config/ring/chaining/#> mode active
```

```
RICH1:/config/ring/chaining/#> end
RICH1:/config/ring/#> end
RICH1:/config/#> end
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
RICH1:/#>
```

Repeat the same steps for RICH2. RICO1 and RICO2 shall use the same configuration except that RiCh mode is passive. Configure the coupling ports on RICO1 and RICO2 to a lower priority to make them work as backup coupling connections. The nodes FRNT1 and FRNT2 should only configure vlan and frntv2.

4.4. IP Host Services

4.4.1. Dynamic DNS (DDNS) Basic HowTo

About

The purpose of this document is to provide a basic DDNS use-case, showing some CLI examples.

For an overview of all available DDNS configuration settings please refer to configuration guide by following this link: DDNS Configuration Guide

Introduction

DDNS is a service offered by many Internet Service Providers (ISPs) to allow end-users to register a dynamic (DHCP/PPPoE/PPPoA) address with a DNS record.

DDNS service maps domain names to IP addresses and lets users access stations (routers) from internet. Whenever IP address associated with hostname changes, DDNS service takes care of such update.

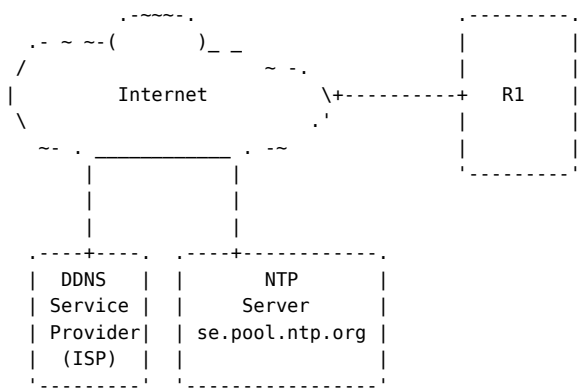


Figure 1: Example network where router R1 is configured to connect to DDNS service provider in order to update its hostname. Synchronisation with NTP peer is made in order to assure correct operation of WeOS' applications.

NTP configuration

In order to insure proper functionality of WeOS, correct time needs to be set. Below are the steps necessary for configuring NTP:

```
R1:/#> configure
R1:/config/#> ntp
R1:/config/ntp/#> peer se.pool.ntp.org
R1:/config/ntp/peer-se.pool.ntp.org/#> leave
R1:/#>
```

After some time (5 minutes approximately), R1 should be synchronised wiht `se.pool.ntp.org` which can be verified by the following adminexec command:

```
R1:/#> show ntp
NTP Client/Server running as PID: 1344
4/4 peers valid, clock synced, stratum 2

peer
   wt tl st  next  poll          offset       delay      jitter
193.182.111.12 from pool se.pool.ntp.org
    1 10  2   27s   33s        -0.803ms     6.181ms     0.790ms
```

```
192.36.143.130 from pool se.pool.ntp.org
 *  1 10  1   20s   32s        -1.131ms     5.989ms     0.140ms
91.209.0.19 from pool se.pool.ntp.org
    1 10  2   20s   30s        -0.043ms    12.596ms     0.178ms
91.209.0.17 from pool se.pool.ntp.org
    1 10  2   25s   31s        -0.319ms    13.189ms     0.117ms
```

The date/time in WeOS should now be in sync with remote peer, and can be verified by running:

```
R1:/#> time
Thu Apr 30 17:34:51 UTC 2020
```

For more details on how to configure NTP please have a look at: NTP Configuration Guide

Interface configuration

Configure the Router to obtain IP address from ISP through DHCP.

```
R1:/#> configure
R1:/config/#> iface eth1
R1:/config/iface-eth1/#> inet dhcp
R1:/config/iface-eth1/inet-dhcp/#> leave
R1:/#>
```

After successfull communication to ISP, IP address should be obtained. This can be verified with the following adminexec command:

```
R1:/#> show iface
INTERFACE          OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
eth1               UP    170.70.122.76/32    dhcp        52:54:00:12:34:57
```

Interface `eth1` should now have IP address assigned to it.

For more details on how to configure Interface please have a look at: Interface Configuration Guide

Builtin provider configuration

Below is an example configuration of 2 builtin providers. For the sake of more extensive example, we will configure 2 instances of the same provider, in this case `Freedns`.

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> ddns
R1:/config/ip/ddns/#> provider-id 1
R1:/config/ip/ddns/provider-1/#> provider freedns
R1:/config/ip/ddns/provider-1/#> hostname myhostname.org
R1:/config/ip/ddns/provider-1/#> username myusername
R1:/config/ip/ddns/provider-1/#> password mypassword
R1:/config/ip/ddns/provider-1/#> end
R1:/config/ip/ddns/#> provider-id 2
R1:/config/ip/ddns/provider-2/#> provider freedns
R1:/config/ip/ddns/provider-2/#> hostname anotherhost.com
R1:/config/ip/ddns/provider-2/#> username root toor
R1:/config/ip/ddns/provider-2/#> leave
R1:/#>
```

Verification of the status for our host entries (myhostname.org & anotherhost.com) can be done via adminexec command shown below:

```
Server:/#> show ip ddns
HOSTNAME                        STATUS
myhostname.org                  170.70.122.76
anotherhost.com                 193.28.25.75
```

On success, configured hostnames should be shown with respective IP address.

Notice that by default `SSL` option is enabled in DDNS configuration, which means that HTTPS protocol is used for communication with providers.

By default `ca-store` option is set to `auto` - meaning that only built-in certificates (those shipped with WeOS) are used.

To check exactly what certificates are used and their expiery date one may run the following command:

```
Server:/#> show cert all
TYPE     HASH     EXPIRES     NAME
CA-auto  a94d09e5 Dec 31 2030 ACCVRAIZ1
CA-auto  cd8c0d63 Jan  1 2030 N/A
CA-auto  930ac5d2 Sep 22 2030 Actalis Authentication Root CA
CA-auto  157753a5 May 30 2020 AddTrust External CA Root
CA-auto  2b349938 Dec 31 2030 AffirmTrust Commercial
CA-auto  93bc0acc Dec 31 2030 AffirmTrust Networking
CA-auto  b727005e Dec 31 2040 AffirmTrust Premium
CA-auto  9c8dfbd4 Dec 31 2040 AffirmTrust Premium ECC
CA-auto  ce5e74ef Jan 17 2038 Amazon Root CA 1
CA-auto  6d41d539 May 26 2040 Amazon Root CA 2
CA-auto  8cb5ee0f May 26 2040 Amazon Root CA 3
CA-auto  de6d66f3 May 26 2040 Amazon Root CA 4
--More-- (17% of 10619 bytes)
```

If user wishes to use his own CA certificates for DDNS, he may achieve this by importing those certificates first with the help of `cert` command, and then setting `ca-store` option inside DDNS context to `local`:

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> ddns
R1:/config/ip/ddns/#> ca-store local
R1:/config/ip/ddns/#> leave
R1:/#>
```

For more details on how to manage certificates please have a look at: Certificate Management Guide

Custom provider configuration

Below is an example DDNS configuration of custom provider with custom checkip server. Notice that `SSL` is explicitely disabled for both checkip and ddns provider in this particular example. This is simply to show that not all providers allow communication over HTTPS protocol, but disabling `ssl` option will handle such case.

Default IP address check interval is 43200 seconds (12 hours), but let's suppose that our provider allows more often checks, such as 12000 seconds.

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> ddns
R1:/config/ip/ddns/#> interval 12000
R1:/config/ip/ddns/#> provider-id 5
R1:/config/ip/ddns/provider-5/#> custom twodns
R1:/config/ip/ddns/provider-5/#> server twodns.de /
R1:/config/ip/ddns/provider-5/#> no ssl
R1:/config/ip/ddns/provider-5/#> checkip-server checkmyip.org /checkip
R1:/config/ip/ddns/provider-5/#> no checkip-ssl
R1:/config/ip/ddns/provider-5/#> hostname myhostname1.org myhostname2.org
R1:/config/ip/ddns/provider-5/#> username myusername
R1:/config/ip/ddns/provider-5/#> password mypassword
R1:/config/ip/ddns/provider-5/#> leave
R1:/#>
```

Verification of status for specified host entries can be done via adminexec command shown below:

```
Server:/#> show ip ddns
HOSTNAME                        STATUS
myhostname1.org                 184.10.10.20
myhostname2.org                 184.10.10.20
```

On success, configured hostnames should be shown with respective IP addresses.

4.4.2. IPv6 Basic HowTo

About

This document aims to provide a basic guide on how to configure IPv6 on the device. The end goal is to have the device being accessible via IPv6 on the local network.

Following this should have the device being accessible via IPv6, using basic management services such as SSH, WEB and SNMP.

> **NOTE**
>
> This document will not cover anything related to routing, this is a basic guide to have the device accessible via IPv6 on the local network.

> **NOTE**
>
> When it comes to routing and IPv6, the device currently only supports basic static unicast routing.

> **SEE ALSO**
>
> Related Documentation
> - Interface Configuration
> - IGMP/MLD Snooping

Introduction

IPv6 is the next generation of the Internet Protocol, which is designed to replace regular IPv4. IPv6 provides a much larger address space than IPv4, which is running out of addresses. However, this is not generally a problem for devices on a local network, as they can use private addresses.

As mentioned, this document will focus on the setup of IPv6 on the device for local network access. Where the main targeted scenario is the possibility to establish management connections to the device using IPv6.

We can consider a scenario, as presented in **Figure 1** below, where we have a few switches with IPv6 enabled, that are connected to a number of hosts. The hosts are also IPv6 enabled and can communicate with the switches using IPv6.

```
Local Network: 2001:db8:1::/64

      .--------.                    .--------.                    .--------.
      |        |                    |        |                    |        |
      |   S1   +--------------------+   S2   +--------------------+   S3   |
      |        |                    |        |                    |        |
      '---+----'                    '---+----'                    '---+----'
          |vlan1: .101                  |vlan1: .102                  |vlan1: .103
          |                             |                             |
          |                             |                             |
--+-------+-------+---         --+-------+-------+---         --+-------+-------+---
  |       |       |             |       |       |             |       |       |
  |.1     |.2     |.3           |.4     |.5     |.6           |.7     |.8     |.9
.-+--.  .-+--.  .-+--.        .-+--.  .-+--.  .-+--.        .-+--.  .-+--.  .-+--.
| H1 |  | H2 |  | H3 |        | H4 |  | H5 |  | H6 |        | H7 |  | H8 |  | H9 |
'----'  '----'  '----'        '----'  '----'  '----'        '----'  '----'  '----'
```

Figure 1: Basic IPv6 network scenario with switches and hosts.

The devices **S1**, **S2** and **S3** are the switches in the scenario that are WeOS devices, and the ones that will be the focus of the later configuration examples. The hosts **H1** to **H9** are just some example devices that are connected to the switches. The network is using the IPv6 address range `2001:db8:1::/64`.

The upcoming section and its subsections will provide some introductory information on IPv6 addresses. If this information is superfluous, feel free to skip ahead to the section covering the actual Configuration of the devices.

The Anatomy of an IPv6 Address

The subsequent subsections will provide some introductory information on IPv6 addresses, both how they differ compared to IPv4 addresses, as well as how they are structured. The goal is to provide basic foundational knowledge how to write and understand an IPv6 address, especially when coming from an IPv4 background.

Basic Differences Between IPv4 and IPv6

The following are just a few high-level differences between IPv4 and IPv6 addresses, in order to provide a basic understanding of the differences between the two.

Feature	IPv4	IPv6
Address Length	32 bits	128 bits
Maximum Number of Addresses	2^32, approximately 4.3 billion	2^128, approximately 3.4 x 10^38, i.e. 340 undecillion. For practical purposes, it is essentially infinite.
Address Groups	4 groups of 3 decimal digits **[0-9]**	8 groups of 4 hexadecimal digits **[0-9A-F]**
Group Separator	Dots **(.)**	Colons **(:)**
Leading Zeros	Can be omitted	Can be omitted
Consecutive Zeros	Can **NOT** be replaced	Can be replaced by double colons **(::)**
Fragmentation	Routers are allowed to fragment packets.	Routers are **NOT** allowed to fragment packets, instead the end devices are solely responsible for any fragmentation.
Checksum	Included in the header	Removed from the header.

Table 1: Basic differences between IPv4 and IPv6 addresses.

IPv6 Address Format

Writing an IPv6 address can be a bit confusing at first, if you are mostly used to working with IPv4 addresses, but it is quite simple once you get the hang of it. It is in some ways similar to how a MAC address is written, by using colons to separate the groups and using hexadecimal digits.

As mentioned, an IPv6 address is much larger than an IPv4 address, at 128 bits compared to 32 bits. This means that an IPv6 address is divided into 8 groups of 4 hexadecimal digits, separated by colons. A basic example of a full IPv6 address would be:

`2001:0db8:0001:0002:7357:0000:0000:0001`

However, since there are a lot of zeros in the above address, it is possible to omit the leading zeros, as such:

`2001:db8:1:2:7357:0:0:1`

Furthermore, it is also possible to replace consecutive groups of zeros with double colons, as such:

`2001:db8:1:2:7357::1`

Using some of these methods can make the address a bit shorter and easier to read.

> **NOTE**
>
> Double colons can only be used once in an address, as it would otherwise be impossible to determine how many groups of zeros were replaced. For instance, the address `2001:db8::1::1` is not a valid address, as it is impossible to determine how many groups of zeros were replaced by the double colons.

An IPv6 address is typically divided into three parts, as shown in **Figure 2** below, where the parts are as follows:

- **Global Routing Prefix**: This is the part of the address that is used to route the packet to the correct network.
- **Subnet ID**: This is the part of the address that is used to identify the local network.
- **Interface ID**: This is the part of the address that is used to identify the a specific network interface on the local subnet.

```
                          128 Bits in Total
<------------------------------------------------------------------------>
<-------N Bits-------> <-M Bits-> <----------128-N-M Bits---------->
┌──────────────────────┬──────────┬──────────────────────────────────┐
│ Global Routing Prefix │ Subnet ID │           Interface ID           │
└──────────────────────┴──────────┴──────────────────────────────────┘

Typically the Global Routing Prefix is 48 bits, the Subnet ID is 16 bits and
the Interface ID is 64 bits. So in this case N=48, M=16 and the Interface ID
is 64 bits.
```

Figure 2: Basic structure of an IPv6 address. The address is divided into three parts, the Global Routing Prefix, the Subnet ID and the Interface ID.

Based on the structure in **Figure 2**, the addresses for the devices S1, S2 and S3 in the scenario presented in **Figure 1** would be:

```
S1: 2001:0db8:0001:0000:0000:0000:0000:0101/64
S2: 2001:0db8:0001:0000:0000:0000:0000:0102/64
S3: 2001:0db8:0001:0000:0000:0000:0000:0103/64
    <------------> <-> <------------------>
           A        B            C

A: Global Routing Prefix - 48 bits
B: Subnet ID             - 16 bits
C: Interface ID          - 64 bits
```

Figure 3: IPv6 addresses for devices S1, S2 and S3 in the scenario presented in Figure 1. This is a breakdown what each part of the address represents.

IPv6 Address Types

In IPv6 there are a number of different address types, the most common ones are:

- **Unicast Address**: An address that identifies a single interface.
- **Multicast Address**: An address that identifies a group of interfaces.

- **Anycast Address**: An address that identifies a group of interfaces, where the packet is sent to the nearest interface in the group.

> **NOTE**
>
> One thing to take note of is that the IPv6 does not have a broadcast address, as IPv4 (255.255.255.255) does. Instead, multicast must be used to reach all devices on the local network.

IPv6 Unicast

The different types of unicast addresses are displayed in the table below:

Address Type	Range	Description	Example
Global Unicast	2000::/3	A routable unicast address, similar to a public IPv4 address.	2001:db8:1::1
Link-Local	fe80::/10	An address used to communicate with other devices on the local network.	fe80::5054:ff:fe12:3450
Loopback	::1/128	An address used to communicate with the local device itself.	::1
Unique Local	fc00::/7	An address that is similar to a private IPv4 address, not intended to be routed on the Internet.	fc00:1::1
Embedded IPv4	::/80	An address that embeds an IPv4 address.	2001:db8:c000:221::

Table 2: Different types of IPv6 Unicast addresses.

The most common type will be the **Global Unicast** address, which is the equivalent of a public IPv4 address. The **Link-Local** address is also important. It will always be assigned to any interface that is configured with any other type of IPv6 address implicitly, even if not explicitly specified. It is required for some IPv6 features to work correctly.

When assigning a Unicast address to an interface a prefix length will also be assigned, if no prefix length is specified, the default is `/64`. This is the most common prefix length used for IPv6 addresses, as it is the smallest prefix length that is supported by all IPv6 devices. The prefix length is used to determine the size of the network that the address is assigned to. Assigning the prefix length is pretty much the same as using the CIDR notation for IPv4.

IPv6 Multicast

Same as for IPv4, IPv6 also has multicast addresses. Multicast addresses are used to send frames to multiple destinations at the same time.

Multicast addresses in IPv6 will always be in the range **ff00::/8**. This gives the total range of multicast addresses in IPv6 as **ff00:0000:0000:0000:0000:0000:0000:0000 - ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff**. This can be seen as equivalent to the IPv4 multicast range **224.0.0.0/4**.

> **TIP**
>
> It is good reminder that if you see an address that starts with **ff**, it is a multicast address.

The IPv6 multicast address has a bit more advanced structure than the IPv4 multicast address. Each multicast address have certain bits that are reserved for special purposes. The initial part of the multicast address is divided into the following parts, as can also be seen in **Figure 4** below:

- The first 8 bits are always set to **ff**.
- The next 4 bits are the so called **Flags** field.
- The last 4 bits are the so called **Scope** field.
- The remaining 112 bits are the **Group ID**.

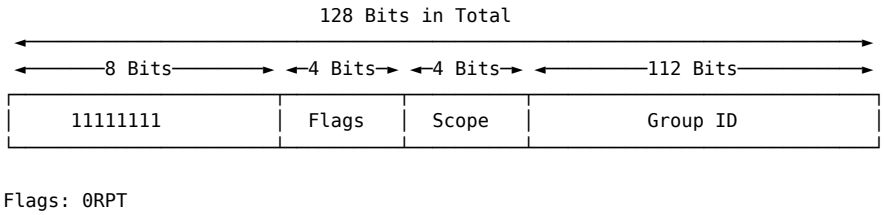


Figure 4: Basic structure of an IPv6 multicast address. The address is divided into three parts, the Flags field, the Scope field and the Group ID, with the initial part always set to ff.

The **Flags** field is divided into the following parts:

Flag	When Set to 1	When Set to 0
R (Rendezvous)	Rendezvous Point is embedded in the address.	Rendezvous Point is not embedded in the address.
P (Prefix)	Prefix Based.	Not Prefix Based.
T (Transient)	Dynamic, an administratively assigned address.	Permanent, a predefined multicast address (Assigned by IANA).

Table 3: The Flags field in an IPv6 multicast address.

The **scope** field is divided into the following parts:

Value	IPv6 Address	IPv4 Counterpart	Scope	Description
0x1	ffx1::/16	-	Interface-Local	The scope is limited to the local interface, usable for loopback.
0x2	ffx2::/16	224.0.0.0/24	Link-Local	The scope is limited to the local link. Frames within this scope may not be routed.
0x4	ffx4::/16	-	Admin-Local	Administratively defined scope.
0x5	ffx5::/16	-	Site-Local	The scope is limited to the local site.
0x8	ffx8::/16	239.192.0.0/14	Organisation-Local	The scope is limited to the local organisation.
0xE	ffxe::/16	224.0.1.0-238.255.255.255	Global	The scope is global. Allowed to be routed on the Internet.

Table 4: The Scope field in an IPv6 multicast address.

As an example, the multicast address `ff02::1` is a non-transient (Predefined), link-local multicast address.

As another example, we can take the ND multicast address for the IPv6 address `2001:db8:1::101` for `S1`, that would be `ff02::1:ff00:101`. Similar to the one above it is a non-transient, link-local multicast address.

Well Known IPv6 Multicast Addresses

There are some well known IPv6 multicast addresses that are reserved for different purposes. In the table below we have a list of some of the well known IPv6 multicast addresses:

Multicast Address	Description
ff02::1	All Nodes
ff02::2	All Routers
ff02::5	OSPFv3 Routers
ff02::6	OSPFv3 DR Routers
ff02::8	IS-IS Routers
ff02::9	RIP Routers
ff02::a	EIGRP Routers
ff02::d	PIM Routers
ff02::12	VRRP Routers
ff02::16	MLDv2 Reports
ff02::1:2	All DHCP Servers and Relay Agents
ff02::1:3	All LLMNR Nodes
ff0x::fb	mDNSv6
ff0x::101	Network Time Protocol (NTP)
ff0x::181	Precision Time Protocol (PTP) version 2 message, except peer delay measurement
ff02::6b	Precision Time Protocol (PTP) version 2 peer delay measurement

Table 5: Some well known IPv6 multicast addresses.

IPv6 Multicast MAC Address Representation

Same as for IPv4 multicast, IPv6 multicast addresses also have a corresponding MAC address. The MAC address is calculated by taking the last 32 bits of the IPv6 multicast address and appending them to the prefix `33:33`. In **figure 5** below, we have an example of how the MAC address is calculated for the IPv6 multicast address `ff02::1`.

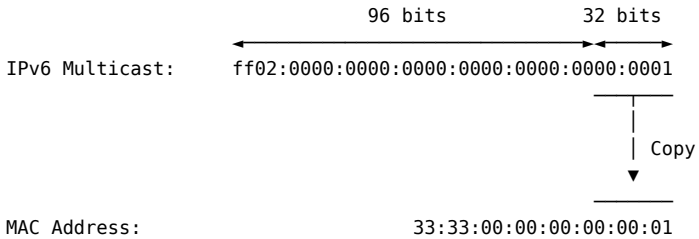


Figure 5: Calculation of the MAC address representation for the IPv6 multicast address ff02::1.

The IPv6 multicast addresses need to have corresponding MAC addresses in order to be insertable into the hardware's Forwarding Database (FDB). This is necessary to ensure that the multicast

traffic can be forwarded correctly on the network, when the multicast traffic is not flooded to all ports, e.g. when MLD snooping is enabled.

Configuration

Adding an IPv6 address to a WeOS device is quite straightforward and it is done in essentially the same way as for IPv4. Configuration of any type of IP address is always done in the interface context. When an IPv4 address is configured the `inet` command is used, for IPv6 the `inet6` command is used instead.

As such, the devices **S1**, **S2** and **S3** in the scenario presented in **Figure 1** are configured in the following way:

```
S1:/#> configure
S1:/config#> iface vlan1
S1:/config/iface-vlan1/#> inet6 static 2001:db8:1::101/64
```

```
S2:/#> configure
S2:/config#> iface vlan1
S2:/config/iface-vlan1/#> inet6 static 2001:db8:1::102/64
```

```
S3:/#> configure
S3:/config#> iface vlan1
S3:/config/iface-vlan1/#> inet6 static 2001:db8:1::103/64
```

> **NOTE**
>
> If we would not have specified the prefix length in the above commands, the default prefix length would have been `/64` anyway, since it is the default value. However, it is possibly good practice to always specify the prefix length, to ensure that it is always correct, since this behaviour could differ between different vendors.

By default IPv6 is globally disabled on the device. This means that even if an IPv6 address is configured on an interface, it will not be active until IPv6 is globally enabled. This option is located under the `ipv6` context. Therefore, to enable IPv6 on the device, the following command must be executed for each of the devices:

```
S1:/#> configure
S1:/config#> ipv6
S1:/config/ipv6/#> enable
```

```
S2:/#> configure
S2:/config#> ipv6
S2:/config/ipv6/#> enable
```

```
S3:/#> configure
S3:/config#> ipv6
S3:/config/ipv6/#> enable
```

At this point, when the configuration is applied on the devices, they should now have the IPv6 addresses added to the selected interfaces. In order to verify this the `show iface` command can be used in the exec context, it should look something like this:

```
S1:/#> show iface
INTERFACE          OPER  ADDRESS/LENGTH                SOURCE      MAC/PTP ADDRESS
lo                 UP    127.0.0.1/8                   static      00:00:00:00:00:00
vlan1              UP    2001:db8:1::101/64            static      52:54:00:12:34:50
                         fe80::5054:ff:fe12:3450/64    link-local
```

> **NOTE**
>
> Observe that we have also had a link-local address added to the interface as well, even though it was not explicitly configured. This will always be implicitly added to the interface for any `inet6` method that is configured. Of course it is possible to configure a link-local address explicitly as well, using `inet6 link-local`.

Ensure Snooping or Flooding of Unknown Multicast is Enabled

Currently, both of these settings should be enabled by default. However, it is still beneficial to know where to find these specific settings, in case they are disabled or need to be changed. For more information on why these settings are important, please refer to the section on IPv6 NeighborDiscovery below.

Enabling MLD Snooping

For some more detailed information why MLD snooping is important for IPv6, please refer to the subsection on MLD Snooping under the IPv6 Neighbour Discovery section.

MLD snooping shares the same configuration as IGMP snooping, i.e. they both use the same settings. Currently all of these shared settings are located under the `ip` context and not under the `ipv6` context. To enable MLD snooping, the following command must be executed:

```
S1:/#> configure
S1:/config#> ip
S1:/config/ip/#> multicast-snooping
```

In addition `multicast-snooping` can also be enabled/disabled on a per VLAN basis. For instance, to enable MLD snooping on VLAN 1, the following command must be executed:

```
S1:/#> configure
S1:/config#> vlan 1
S1:/config/vlan-1/#> multicast-snooping
```

For more information on other relevant multicast snooping settings, please refer to the IGMP/MLD Snooping documentation.

Enabling Flooding of Unknown IPv6 Multicast Addresses

For some more detailed information why flooding of unknown IPv6 multicast addresses is important for IPv6, please refer to the subsection on Flooding of Unknown IPv6 Multicast Addresses under the IPv6 Neighbour Discovery section.

The configuration for flooding of unknown IPv6 multicast addresses, shares the same configuration as flooding of unknown IPv4 multicast addresses. This setting is located under the `ip` context and not under the `ipv6` context. To enable flooding, the following command must be executed:

```
S1:/#> configure
S1:/config#> ip
S1:/config/ip/#> multicast-flood-unknown all
S1:/config/ip/#> show multicast-flood-unknown
ALL
S1:/config/ip/#>
```

The setting `multicast-flood-unknown` operates per port on the device. In the command above we did not specify any port, we instead used the keyword `all`, which means that the setting will be applied to all ports.

If we want to enable this setting for a specific port or a range of ports, we can do so by specifying the port or range of ports in the command. However, the command itself is additive, meaning that we must first remove any other ports that we do not want to have this setting enabled on, before adding the new ports. For instance, to enable flooding of unknown multicast on port eth1, the following command must be executed:

```
S1:/#> configure
S1:/config#> ip
S1:/config/ip/#> no multicast-flood-unknown
S1:/config/ip/#> multicast-flood-unknown eth1
S1:/config/ip/#> show multicast-flood-unknown
eth1
S1:/config/ip/#>
```

Or in order to apply it to a range of ports, for instance eth1 to eth5, the following command must be executed:

```
S1:/#> configure
S1:/config#> ip
S1:/config/ip/#> no multicast-flood-unknown
S1:/config/ip/#> multicast-flood-unknown eth1..eth5
S1:/config/ip/#> show multicast-flood-unknown
eth1..eth5
S1:/config/ip/#>
```

For some more information on the `multicast-flood-unknown` setting, please refer to the IGMP/MLD Snooping documentation page, the command description is located there.

Testing

At this point, the configured switches should be reachable via IPv6. We can test this by simply pinging the switches from any of the hosts on the network. For instance, we could try and ping the switches **S2**, **S3** and hosts **H4**, and **H7** from **S1**:

```
S1:/#> ping 2001:db8:1::102
Press Ctrl-C to abort PING 2001:db8:1::102(2001:db8:1::102) 56 data bytes
64 bytes from 2001:db8:1::102: icmp_seq=1 ttl=64 time=0.838 ms

S1:/#> ping 2001:db8:1::103
Press Ctrl-C to abort PING 2001:db8:1::103(2001:db8:1::103) 56 data bytes
```

```
64 bytes from 2001:db8:1::103: icmp_seq=1 ttl=64 time=0.838 ms

S1:/#> ping 2001:db8:1::4
Press Ctrl-C to abort PING 2001:db8:1::4(2001:db8:1::4) 56 data bytes
64 bytes from 2001:db8:1::4: icmp_seq=1 ttl=64 time=0.838 ms

S1:/#> ping 2001:db8:1::7
Press Ctrl-C to abort PING 2001:db8:1::7(2001:db8:1::7) 56 data bytes
64 bytes from 2001:db8:1::7: icmp_seq=1 ttl=64 time=0.838 ms

S1:/#>
```

It should at this point also be possible to access the switches via SSH, WEB and SNMP, if these management services are enabled on the devices. For instance, we can try to SSH into **S2** from **S1**:

```
S1:/#> ssh 2001:db8:1::102
admin@2001:db8:1::102's password:

S2:/#>
```

TroubleShooting

If the device has been configured with an IPv6 address, but it is not active on the configured interface, check the following:

- Is IPv6 globally enabled on the device, under the `ipv6` context?

If any of the configured devices on the network have issues reaching each other, check the following:

- Is MLD snooping enabled on the device?
- Is flooding of unknown IPv6 multicast addresses enabled on the device?

If MLD is in use, the joined groups can be checked on the device to ensure that the correct groups are joined. This can be done by using the `show ipv6 mld` command:

```
S1:/#> show ipv6 mld

Interface(s)
INTERFACE  STATE  ADDRESS                  V  QUERIER  QUERIER ADDRESS          QUERY TIMER   UPTIME
vlan1      up     fe80::edb:c6ff:fef3:0    2  other    fe80::e6f:eeff:fe8b:0    -             00:00:45.054

Group(s)
INTERFACE  GROUP            VERSION  UPTIME
vlan1      ff02::16         2        00:00:43.464
vlan1      ff02::1:ff00:1   2        00:00:42.220
vlan1      ff02::1:ff00:4   2        00:00:33.050
vlan1      ff02::1:ff00:7   2        00:00:02.217
vlan1      ff02::1:ff00:101 2        00:00:43.464
vlan1      ff02::1:ff00:102 2        00:00:29.075
vlan1      ff02::1:ff00:103 2        00:00:28.706
vlan1      ff02::1:ff15:0   2        00:00:42.220
vlan1      ff02::1:ff73:0   2        00:00:02.217
vlan1      ff02::1:ff85:0   2        00:00:33.050
vlan1      ff02::1:ff8b:0   2        00:00:29.075
vlan1      ff02::1:ffc8:0   2        00:00:28.706
vlan1      ff02::1:fff3:0   2        00:00:43.464

Join(s)
GROUP            SOURCE  STATE  LASTSEEN  NONTRKSEEN  CREATED
On interface vlan1:
ff02::16         *       JOIN   00:00:00  -           00:00:43
ff02::1:ff00:1   *       JOIN   00:00:07  -           00:00:42
ff02::1:ff00:4   *       JOIN   00:00:00  -           00:00:33
ff02::1:ff00:7   *       JOIN   00:00:02  -           00:00:02
```

```
ff02::1:ff00:101  *       JOIN   00:00:37  -          00:00:43
ff02::1:ff00:102  *       JOIN   00:00:00  -          00:00:29
ff02::1:ff00:103  *       JOIN   00:00:01  -          00:00:28
ff02::1:ff15:0    *       JOIN   00:00:07  -          00:00:42
ff02::1:ff73:0    *       JOIN   00:00:02  -          00:00:02
ff02::1:ff85:0    *       JOIN   00:00:00  -          00:00:33
ff02::1:ff8b:0    *       JOIN   00:00:00  -          00:00:29
ff02::1:ffc8:0    *       JOIN   00:00:01  -          00:00:28
ff02::1:fff3:0    *       JOIN   00:00:37  -          00:00:43
```

To check the currently identified IPv6 neighbours on the device, the `show ipv6 neighbors` command can be used:

```
S1:/#> show ipv6 neighbours

IPv6 Neighbours
ADDRESS                 INTERFACE  MAC                STATE
fe80::e12:6fff:fe73:0   vlan1      0c:12:6f:73:00:00  DELAY
2001:db8:1::102         vlan1      0c:6f:ee:8b:00:00  REACHABLE
fe80::e6f:eeff:fe8b:0   vlan1      0c:6f:ee:8b:00:00  REACHABLE
2001:db8:1::103         vlan1      0c:ad:25:c8:00:00  REACHABLE
2001:db8:1::7           vlan1      0c:12:6f:73:00:00  REACHABLE
fe80::ead:25ff:fec8:0   vlan1      0c:ad:25:c8:00:00  REACHABLE
```

IPv6 Neighbour Discovery

Same as for IPv4, IPv6 must have a way to discover the MAC address of a device on the local network. In IPv4, this is done using ARP (Address Resolution Protocol). IPv6 uses a similar mechanism called Neighbour Discovery Protocol (NDP).

In IPv4 ARP uses broadcast messages to discover the MAC address of other devices on the local network. Since IPv6 does not support broadcast, NDP uses multicast messages to discover the MAC address of other devices on the local network. However, the NDP messages are not sent to a common multicast address, but to a specific multicast address based on the target IPv6 address. This means that all of these different multicast addresses must be able to reach the intended device(s).

This calculated address is referred to as the **Solicited-Node Multicast Address**. It is calculated by taking the last 24 bits of the target IPv6 address and appending them to the prefix `ff02::1:ff00:0/104`. This will result in a multicast address that is, in most cases, unique for each target IPv6 address. As en example, given the IPv6 address of the devices `S1`, `S2` and `S3` in the scenario presented in **Figure 1**, the Solicited-Node Multicast Address would be:

Device	IPv6 Address	Solicited-Node Multicast Address
S1	2001:db8:1::101	ff02::1:ff00:101
S2	2001:db8:1::102	ff02::1:ff00:102
S3	2001:db8:1::103	ff02::1:ff00:103

Table 6: Solicited-Node Multicast Addresses for devices S1, S2 and S3 in the scenario presented in Figure 1.

Just to illustrate the calculation of the Solicited-Node Multicast Address, we use the IPv6 address of device `S1` from the above table, in **Figure 6** below.

```
                                  104 bits            24 bits
                          ◄──────────────────────────►◄──────►
S1 IPv6 Address:          2001:0db8:0001:0000:0000:0000:0000:0101
                                                      ──────┬──────
                                                            │
                                                            │ Copy
                                                            ▼
                                                      ──────┴──────
S1 ND Multicast:          ff02:0000:0000:0000:0000:0001:ff00:0101
```

Figure 6: Calculation of the Solicited-Node Multicast Address for device S1. The addresses have been expanded to their full 128 bits for clarity.

Therefore, in order to ensure that any device on the network can correctly perform a neighbour discovery, the switches must be able to forward any potential Solicited-Node Multicast Address correctly. To do this the devices must have at least one of the following features enabled:

- Flooding of Unknown IPv6 Multicast Addresses. This setting is enabled by default.
- MLD Snooping. This setting is also enabled by default.

One of the above must always be enabled to ensure that the devices can correctly perform neighbour discovery. If not, the required multicast addresses will not be reachable and the devices will not be able to communicate with each other. As mentioned, the default is that both of these features are enabled, meaning that there is no issue having them both enabled at the same time, it is actually recommended.

Flooding of Unknown IPv6 Multicast Addresses

With flooding of unknown IPv6 multicast addresses enabled, the switch will, as it sounds, flood all unknown IPv6 multicast addresses to all ports, or specific ports if wanted. Doing this will ensure that any ND messages will always be able to reach the intended device(s) on the local network.

The drawback of this is that all multicast traffic will always be flooded to all ports, or the specified individual flood ports. When we do this we essentially treat the multicast traffic as broadcast traffic, somewhat defeating the purpose of using multicast, in the sense that we are not limiting the traffic to only the intended devices. For this we should of course need MLD snooping enabled.

However, it is usually far more important that communication works, than to save bandwidth, as long as said bandwidth is enough to handle the intended traffic. But naturally, if the amount of multicast traffic on the network is increased, there is going to be a point where the network starts to get congested, and having this feature enabled could potentially make the congestion worse.

> **NOTE**
>
> As long as the above issue with bandwidth is not a problem, it is recommended to have this feature enabled. This will ensure that the devices can always perform neighbour discovery and communicate with each other.

If this feature is disabled, the switch will not flood unknown IPv6 multicast. In this scenario, we must instead ensure that MLD Snooping is enabled if we want to ensure that the devices can correctly perform neighbour discovery.

MLD Snooping

MLD snooping is the IPv6 equivalent of IGMP snooping for IPv4. If we do not flood unknown IPv6 multicast addresses, we must rely on MLD snooping to ensure that all the ND multicast addresses groups on the local network are joined correctly. Every single interface with an IPv6 address must

join the correct multicast group associated with its configured IPv6 address. As soon as an interface has an IPv6 address configured, it will automatically join the correct multicast group.

The above, goes for any IPv6 address, including link-local addresses. This means that if a static address is configured on an interface, a bare minimum of two ND multicast groups will be joined.

Even though any IPv6 enabled device must initially send an MLD report to join the correct multicast group, a MLD querier is necessary to ensure that the relevant groups will continue to be joined. The MLD querier will periodically send MLD queries to ensure that the groups are still required. If no querier is present, the groups will eventually time out and be removed. By default the WeOS devices are eligible to be elected as the MLD querier, when snooping is enabled.

For more information on MLD snooping, please refer to the IGMP/MLD Snooping.

> **TIP**
>
> Since the querier should be aware of all the multicast groups on the network, checking the joined groups on the querier can be a good way to verify that the MLD snooping is working correctly, and that all expected groups are joined.

> **WARNING**
>
> In very large network topologies that have a multitude of IPv6 addresses configured across many devices, the number of extra multicast groups that must be joined for all Solicited-Node Multicast Addresses can become quite large. Each of these groups will have their MAC equivalent address inserted into the hardware's Forwarding Database (FDB). The more entries that are present in the FDB, the more likely it is that a MAC address collision will occur. This can lead to a situation where the switch fails to add the MAC address to the FDB, resulting in the ND messages not being able to reach the intended device(s).
>
> Therefore, if IPv6 is to be utilised in a larger network with many IPv6 addresses present, it is recommended to also have flooding of unknown IPv6 multicast addresses enabled. This will ensure that the ND messages always reach the intended device(s), even if there is a MAC address collision in the FDB.

Combining MLD Snooping and Flooding of Unknown IPv6 Multicast Addresses

As mentioned, the default is that both of these features are enabled. This ensures that multicast traffic should always be able to reach the intended device(s), even if there is a MAC address collision in the FDB.

The resulting behaviour in this scenario is that unknown multicast traffic, i.e. multicast traffic that is not part of a joined group, will be flooded to all ports. Once an intended receiver joins a given multicast group, the group will now be "known" and the traffic should only be forwarded to the ports towards the intended receivers.

Therefore, in this scenario, the amount of unnecessarily flooded multicast traffic is essentially minimised when any intended receiver joins a given multicast group.

Address Resolution Example

As an example, let's say that device `S1` wants to ping device `S3`, on its statically configured IPv6 address **2001:db8:1::103**. The following steps will be taken to resolve the MAC address of device `S3`:

1. First S1 will check if the MAC address of S3 is already a known neighbour and if it is reachable. If it is, the ICMP echo request will be sent directly.
2. If the MAC address is not known, S1 will send an NDP Neighbour Solicitation message to the Solicited-Node Multicast Address of S3. This message will be sent to the multicast address **ff02::1:ff00:103**.
3. S3 will receive the NDP Neighbour Solicitation message and respond with an NDP Neighbour Advertisement message. This message will set the destination MAC address to the MAC that S1 used to send the NDP Neighbour Solicitation message. Similarly, the destination IPv6 address will be set to the source IPv6 address of the NDP Neighbour Solicitation message, **2001:db8:1::101**.
4. S1 will receive the NDP Neighbour Advertisement message and add the MAC address of S3 to its list of known neighbours.
5. S1 will now be able to send the ICMP echo request and insert the correct destination MAC address into the Ethernet frame.

4.5. Generic Maintenance

4.5.1. Access External Media Devices Howto

Introduction

This page aims to explain how external media devices are handled and made accessible for use in the system.

Currently it is only possible to utilise one external media device at a time. Most devices only have access to either a USB or a SD-Card, but some devices may have the possibility to connect both. In these situations only one of them will be accessible by the system, therefore it is recommended to only have one external media device connected at one time. In general, the system will try to prioritise SD-Cards, but it is more deterministic to only connect one device.

However, the system can support external media devices that have multiple partitions specified. In the eyes of the system each individual partition will be handled as a separate media. So if there is a need to have access to different devices this is how it can be achieved.

Each external media device that needs to be accessible on the device must be explicitly matched against a media definition. This definition specifies how a device will be known in the system and what physical external media it will be linked with. If no matching media definition is present for an external media or its individual partitions, they will not be mounted in the system.

This page will go through both how to format and partition a connected media device from the system itself, how to ensure that these medias are identified and made accessible for use.

For additional information on the Media tools and commands, refer to this page.

For additional information on the boot configuration commands, refer to this page.

Partition and Format a Connected Media Device

It is possible to create up to four partitions on an external media device utilising the systems built-in partitioning tool. This tool also allows the user to specify the wanted file system, selectable from the following file system types:

- ext2
- ext3
- ext4
- vfat, deprecated, see section filesystem.

In addition, the size of the partitions needs to be set and an optional label can also be specified for the device. Later when we will set up the media definitions on the device this label can be used as a match indicator for the media. It is also possible to match a device's partition number.

The dev names for a connected device will have the following naming convention when used in the media context for partition and formatting:

- Main disk device: **sd[a-z]** or **mmcblk1**.
- Partitions will be the same but with an added number for the partition instance: **sd[a-z][1-9]** or **mmcblk1p[0-4]**.

Partitioning a Device Using the Partition Tool

This section will provide an example how an external media can be partitioned and formatted in a very simple and easy way using the built-in media tools. In this example we want to partition our device, named sdb, in the following manner:

#	Device Name	Partition Number	Label	Size	Filesystem
1	sdb1	1	default	50%	ext3
2	sdb2	2	my-files	20%	ext3
3	sdb3	3	-	20%	ext2
4	sdb4	4	my-label	10%	ext4

In order to partition the device in the manner described we enter the media context from the top level in the CLI:

```
example:/#> media
example:/media/#>
```

In order to partition the device we use the `partition` command on the correct device (this command is tab completable) to enter the partition context:

> **TIP**
>
> We can also check all the connected physical devices using the `show devices` command in the current context. We can only partition the devices of the type `disk`.
>
> ```
> example:/media/#> show devices
> NAME       TYPE   NUM   LABEL                      FORMAT    SIZE      MOUNT-MEDIA
> sda        disk   -     -                          -         16.0M     -
> ```

```
example:/media/#> partition sdb
Entering partitioning mode with dev sdb as target ...

No changes will be performed until the start-partition command is
invoked. When invoked the entire device sdb will be fully re-partitioned
and reformatted, any files on the device will be LOST/REMOVED, keep
this in mind before proceeding.

example:/media/partition-sdb/#>
```

In this context we can define how we want the device to be partitioned entirely before we start the operation by calling the `start-partitioning` command. Nothing will happen to the device before this is invoked, so it is possible to just leave the context without anything changing on the device.

In this case, since we want the fully re-partition the device, we remove any current partition configuration:

```
example:/media/partition-sdb/#> delete all
```

Now we can add the definitions for the partitions we want to create, as we specified in the table above:

```
example:/media/partition-sdb/#> add label default size 50% format ext3
example:/media/partition-sdb/#> add label my-files size 20% format ext3
example:/media/partition-sdb/#> add size 20% format ext2
example:/media/partition-sdb/#> add label my-label size 10% format ext4
```

Before we initiate the partitioning and formatting of the device, it is best that we observe if the partitions we intend to create look as we expect:

```
example:/media/partition-sdb/#> show
Partition table for dev: sdb
#  DEV-NAME      PART-NUMB  LABEL                          SIZE      FS
1  /dev/sdb1     1          default                        50%       ext3
2  /dev/sdb2     2          my-files                       20%       ext3
3  /dev/sdb3     3          -                              20%       ext2
4  /dev/sdb4     4          my-label                       10%       ext4
```

When we have verified that the setup is what we intended we can start the actual partitioning and formatting of the device:

> **WARNING**
>
> When we start the partitioning we will also reformat the device, so any files will be removed. Ensure that the device does not contain any files that needs to be saved elsewhere.

```
example:/media/partition-sdb/#> start-partitioning

Understand that the content of media device will be lost during the
partitioning and formatting of the device. Make sure that the device
does not contain any files that needs to be saved elsewhere.

Start partitioning and formatting of the device (y/N)? y
Attempt to delete existing partitions on device /dev/sdb:
  1: /dev/sdb1 [Partition numb: 1]
  2: /dev/sdb2 [Partition numb: 2]
  3: /dev/sdb3 [Partition numb: 3]
  4: /dev/sdb4 [Partition numb: 4]

Attempt to partition device /dev/sdb in the following manner:
  1: /dev/sdb1 [Partition numb: 1], [Size: 50%]
  2: /dev/sdb2 [Partition numb: 2], [Size: 20%]
  3: /dev/sdb3 [Partition numb: 3], [Size: 20%]
  4: /dev/sdb4 [Partition numb: 4], [Size: 10%]

Initiate formatting of the partitions, this may take a while:
  1: /dev/sdb1 [Partition numb: 1] [Label: default], format with ext3 FS ...
Formatting: 100% \ [==================================================================]

  2: /dev/sdb2 [Partition numb: 2] [Label: my-files], format with ext3 FS ...
Formatting: 100% \ [==================================================================]

  3: /dev/sdb3 [Partition numb: 3] [Label: -], format with ext2 FS ...
Formatting: 100% \ [==================================================================]

  4: /dev/sdb4 [Partition numb: 4] [Label: my-label], format with ext4 FS ...
Formatting: 100% \ [==================================================================]

Device sdb successfully partitioned and formatted!
example:/media/#>
```

If no error indications appeared, the device should now have been partitioned and formatted according to the specified configuration. We can verify this using the `show devices` command:

```
example:/media/partition-sdb/#> show devices

Connected Media Devices
NAME       TYPE  NUM  LABEL                    FORMAT   SIZE     MOUNT-MEDIA
sdb        disk  -    -                        -        3824M    -
  sdb1     part  1    default                  ext3     1912M    external
  sdb2     part  2    my-files                 ext3     764M     -
  sdb3     part  3    -                        ext2     764M     -
  sdb4     part  4    my-label                 ext4     382M     -
```

We can also get a comprehensive overview of the connected external media device in relation to the media definitions specified on the system using the `show media` command in the *admin-exec* context:

```
example:/#> show media
MEDIA        DEVICE       LABEL          FORMAT   RW     SIZE  USED  DESCRIPTION
Active
internal     mtd:Config   -              jffs2    Yes   15.0M    5%  config,cert
external     sdb1         default        ext3     Yes    1.8G    0%  logging
Inactive
Available
-            sdb2         my-files       ext3     -      764M     -
-            sdb3         -              ext2     -      764M     -
-            sdb4         my-label       ext4     -      382M     -
```

As we can see only the first of our partitions that we created is listed under the `Active` tab. This is because it matches one of the default matching rules for media definitions on the system, the one named external.

> **NOTE**
>
> The default external media match rule, aptly named `external`, is set to match on the first partition of a device. In case a device have no partitions it will match on that as well (No partition means the first partition, in the context of the match rule).

Further, we can see the other three of our partitions are listed as `Available`. This means that a physical device exist, but they have no matching media definition. Hence, this device is available for matching.

Lastly, we currently have nothing listed under `Inactive`. Any media definition that have been created that finds no matching physical device will be listed here.

The next step in order to make the additional external physical media devices accessible, is to create media definitions that can correctly match them. This is covered in the following section on this page.

Format a Specific Device or Partition

It is possible to simply format a given device or partition as well, using the `format` command that is also located in the media context. In the example provided below we want to reformat one of our partitions, `sdb2`. It can be done in the following manner:

```
example:/media/#> format sdb2
example:/media/format-sdb2/#>
```

After that we select the file system we want to format the device with, along with an optional label:

```
example:/media/format-sdb2/#> filesystem ext3
example:/media/format-sdb2/#> label my-files
```

Before we start the formatting we can check our intended configuration:

```
example:/media/format-sdb2/#> show
Format settings for device sdb2
Filesystem : ext3
Label      : my-files
```

When we are satisfied with the planned format options, we can initiate the actual formatting:

> **NOTE**
>
> No changes have actually been made to the device before we use the `start-formatting` command, it is still possible to exit the formatting context without any changes to the device.

> **WARNING**
>
> Observe, that once the formatting starts all files on the device will be lost, so ensure that any files that should be saved have been so.

```
example:/media/format-sdb2/#> start-formatting


Warning: Formatting this dev will remove any files currently stored on it!
         Backup any data you want to keep before formatting the dev.

Do you still want to proceed (y/N)?y

Formatting: 100% \ [═══════════════════════════════════════════════════════]

Format successful!
example:/media/#>
```

If everything went well we can use the `show devices` command to verify that the device was formatted with the correct options:

```
example:/media/#> show devices

Connected Media Devices
NAME       TYPE  NUM  LABEL                    FORMAT   SIZE     MOUNT-MEDIA
sdb        disk  -    -                        -        3824M    -
  sdb1     part  1    default                  ext3     1912M    external
  sdb2     part  2    my-files                 ext3     764M     -
  sdb3     part  3    -                        ext2     764M     -
  sdb4     part  4    my-label                 ext4     382M     -
```

As we can see `sdb2` has been correctly formatted.

Make Specific Media Accessible on the System

In order for an external media device to become mounted and accessible on the system, a media definition needs to be created. This media definition will specify a name that will be how to refer to a specific media when using it in the system. Furthermore, any media definition also have an associated match rule that will specify how a physical external media device will be linked with said media definition.

Only one match rule can be specified per media definition, and can be set to match on the following parameters:

- `partition`: Used to indicate a match based on a specific partition number on the physical media device. If no partitions exist on the device, that will count as a match for partition 1 (The first).
- `label`: Used to indicate a match based on a specific label possibly present on the physical media device.

Default Media Definitions

By default the system will be configured with two media definitions. One is mean to represent the built-in flash memory and the other to serve as a general match rule for any potential connected media device. They are defined in the following manner:

Name	Match Rule	Description
internal	System Internal	Special definition and representation of the built-in flash, this media definition **CANNOT** be changed or removed.
external	Partition: 1	Base definition aimed to match the first partition located on any connected external physical media. If no partitions exist on the device this will also be a match in that case.

The `internal` media definition exist so that we can refer to the built-in flash in the same manner as we do any external media device.

The `external` media definition serve as the most simple way an external media can be matched and be made available. This definition is, in contrast to the previously mentioned, fully adjustable and even removable.

> **NOTE**
>
> Removing the `external` media definition in this case would effectively be disabling any connected external media device, since the device or any potential partition will not be mounted and be made available.

Creating New Media Definitions

In this case we want to create additional media definitions in order to match all of the partitions we created during the example in the earlier section on this page.

In this case we will keep the default `external` media definition and create three additional definitions to match the other partitions. In this case we want to have the following media definitions configured to match the specific partitions.

Name	Match Rule	Intended Physical Device to Match
external	Partition: 1	sdb1

Name	Match Rule	Intended Physical Device to Match
files	Label: my-files	sdb2
stuff	Partition: 3	sdb3
things	Label: my-label	sdb4

All the media definitions on the system are created in the boot configuration context. These settings will also be stored in the boot configuration.

> **NOTE**
>
> These media changes will not become active until the next reboot of the system. Therefore, making changes here without a reboot will *not* cause the media definitions to be mounted, even though the physical external devices currently exist.

In order to start configuring the new media definitions we need to enter the boot configuration context:

```
example:/#> boot
example:/boot/#>
```

First we can check so that we do not already have any other media definitions except for the default ones:

```
example:/boot/#> show media

Media Configuration
#  NAME                      R/O  TIMEOUT  MATCH
1  internal                  No   -------  System Internal
2  external                  No   30       Partition number: 1
```

If the output matches the above we currently only have unchanged default media definitions.

From here we can create our new media definitions in accordance with the table presented above:

```
example:/boot/#> media files
Creating new media definition ...
example:/boot/media-files/#> match label my-files
example:/boot/media-files/#> end
example:/boot/#>
```

```
example:/boot/#> media stuff
Creating new media definition ...
example:/boot/media-stuff/#> match partition 3
example:/boot/media-stuff/#> end
example:/boot/#>
```

```
example:/boot/#> media things
Creating new media definition ...
example:/boot/media-things/#> match label my-label
example:/boot/media-things/#> end
example:/boot/#>
```

Now with our new media definitions they should appear like this:

```
example:/boot/#> show media

Media Configuration
#  NAME                        R/O  TIMEOUT  MATCH
1  internal                    No   -------  System Internal
2  external                    No   30       Partition number: 1
3  files                       No   30       Label: my-files
4  stuff                       No   30       Partition number: 3
5  things                      No   30       Label: my-label
```

Now we exit the boot context so that we write our new settings into the boot configuration:

```
example:/boot/#> end
Saving bootloader configuration to FLASH
100% / [==================================================================]
example:/#>
```

In order for these new media definitions to have any effect and become active, we need to reboot the device:

```
example:/#> reboot
[ ... ]

Please press Enter to activate this console.

example login: admin
Password:

example:/#>
```

Now after we have rebooted the device our media definitions should be active, if they indeed properly matched against any connected external media device and its partitions. We can check this with the show media command, from the top level context (not in the boot context, that is only the configuration):

```
example:/#> show media
MEDIA       DEVICE      LABEL          FORMAT   RW     SIZE  USED  DESCRIPTION
Active
internal    mtd:Config  -              jffs2    Yes   15.0M    5%  config,cert
external    sdb1        default        ext3     Yes    1.8G    0%  logging
files       sdb2        my-files       ext3     Yes  763.3M    0%
stuff       sdb3        -              ext2     Yes  752.0M    0%
things      sdb4        my-label       ext4     Yes  348.3M    0%
Inactive
Available
```

4.5.2. Backup and Restore

The system provides the ability to save device configurations and restore them when needed. Saving a configuration creates a backup of current settings, which can later be applied to the same unit or another device of the same type. Restoring a saved configuration eliminates the need for manual reconfiguration, which can be time-consuming and prone to errors. This feature is particularly useful in scenarios such as:

- **Device Replacement:** When a unit is replaced, restoring the saved configuration ensures the new device operates with the same settings as the previous one.
- **Returning to a Known State:** If changes have been made and a stable, tested configuration is required, restoring from a backup provides a quick and reliable solution.

Using this functionality streamlines maintenance tasks and helps maintain a stable and predictable network environment.

Backup and Restore via Web Interface

> **NOTE**
>
> If the restored configuration includes a different IP address, the current session will not redirect automatically

To create a backup of the current running configuration or restore a previously saved configuration, navigate to **Maintenance** and open the **Backup & Restore** page.

- **Backup Configuration:** Click `Backup` to download a copy of the configuration.
- **Restore Configuration:** A previously saved configuration can be restored by uploading it to the device:
  1. Click `Choose File` and select the configuration file
  2. Click `Restore` to upload and activate the configuration. The restoration process begins immediately.

Backup and Restore via CLI

Commands specified in Filesystem can be used to backup and restore a configuration file.

- **Backup Configuration**: Backup the configuration file to a network location using `scp`:

```
example:/file/cfg/#> export backup.cfg proto scp user test_user password test_password 192.0.2.1
```

- **Restore Configuration**: Restore a configuration file from a network location using `scp`:

```
example:/file/cfg/#>
import proto scp user test_user password test_password 192.0.2.1 backup_remote.cfg
backup.cfg
```

Restore Validation

The system provides functionality to verify the integrity and correctness of a configuration file before restoring. This ensures that the file is compatible and known to the device, reducing the risk of configuration errors and service disruptions. The verification checks includes:

- **Integrity validation:** The file is compared with known created checksums, created and stored in the device, see Configuration Integrity.

- **Syntax validation:** Checks the file for any structural or syntax errors or inconsistencies. This is done by validating the file against the provided JSON schema.
- **Semantic validation:** Verifies that the dataset conforms to certain logical rules.

Auditable events will be created upon a restore event. Always check the audit log if there were any errors in applying the configuration or if the integrity check failed. These evens are described in Backup Restore Events and Configuration Integrity Events.

Manually check configuration integrity

If an unknown configuration file is restored, such as one older than the last five entries or created outside the device, it is recommended to manually verify its integrity. This can be done by comparing the hash of the running or startup configuration with the hash of the external backup file.

The startup and running configuration SHA-256 hashes are available in the Web interface under **Status -> System -> Details** or through the CLI command `show config-hash`:

```
example:/#> show config-hash
Running Configuration hash: 918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b
Startup Configuration hash: 918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b
```

To verify the backup configuration integrity, compare the resulting SHA-256 hash with backed up configuration file hash, which can be generated via the following commands:

- **Windows**:

```
C:\\> certutil -hashfile backup.cfg SHA256
SHA256 hash of backup.cfg:
918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b
```

- **Linux**:

```
$ sha256sum backup.cfg
918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b    backup.cfg
```

Manually check configuration integrity prior restore

Before performing a restore, it is possible to verify integrity by comparing the stored configuration hash on the device with the hash of the externally backed-up file. Backup events can be reviewed in the audit log. In this example, the internal log buffer is used. Compare the SHA-256 hash of the external backup file with the corresponding hash displayed in the table to verify its integrity.

The external backup file hash can be found via the following command:

- **Windows**:

```
C:\\> certutil -hashfile backup.cfg SHA256
SHA256 hash of backup.cfg:
918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b
```

- **Linux**:

```
$ sha256sum backup.cfg
918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b    backup.cfg
```

Compare the resulting hash sum with backed up hashes stored in the device via the audit `search` function. The configuration file hashes are stored under event ID `0x8000000000040000`, see Configuration Integrity Events:

```
example:/audit/#> search 0x8000000000040000
SEQ    Time                  Level/Facility     USER/ROLE               Event
EventID             Message

154    2025-11-23 10:42:14   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update running-config
[918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b]

157    2025-11-23 10:42:14   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update startup-config
[918319e1b9a8ba78ffb079d169e0a03bdda9ba676f738ecbb340a5a26bd64e4b]

200    2025-11-23 10:47:55   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update running-config
[9ba8b96f251d3d216874267ef60c60eec2b859e110cd73356ad85e0827fd81ff]

202    2025-11-23 10:47:56   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update startup-config
[9ba8b96f251d3d216874267ef60c60eec2b859e110cd73356ad85e0827fd81ff]

245    2025-11-23 10:48:19   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update running-config
[ed7fdf0f94eee30d92f61f48fd0a5be6a3faf0bdeb5824681c85cffd13f68918]

247    2025-11-23 10:48:20   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update startup-config
[ed7fdf0f94eee30d92f61f48fd0a5be6a3faf0bdeb5824681c85cffd13f68918]

418    2025-11-23 11:06:46   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update running-config
[110fee8e8714433279b9107d30a7e7429d79b74111f76039563e8050cecf313c]

420    2025-11-23 11:06:47   Notice/Security    root/system
Configuration Integrity - Update Integrity                  0x8000000000040000
Update startup-config
[110fee8e8714433279b9107d30a7e7429d79b74111f76039563e8050cecf313c]
```

Automate configuration integrity check

SNMP provides access to hashes for both the running and startup configurations. An external system can retrieve these hashes via SNMP and compare them with the hashes of stored backup files. This comparison verifies that the external backup matches the device's current configuration.

4.5.3. File Handling Howto

Introduction

This page aims to explain how files are handled, both locally and how files are moved to external media and network locations.

In this Howto configuration files are used as an example for other types of files see this page.

In all examples below it is assumed that the user has entered the configuration file context by executing:

```
example:/#> file cfg
example:/file/cfg/#>
```

General

All commands except import and export support TAB-completion of command arguments. For file arguments, actual present files are shown as well as symbolic names like running-config and startup-config.

In the top level of the CLI commands exist for specific file types.

- **follow** for log files
- **diff** for configuration files
- **copy** for configuration files

Viewing a file

To view a file the **show** command is used. This command is context aware meaning that is has specific features for different contexts. In this example of configuration files the **show** command is aware that the configuration files are json files.

In this example a subsection of the running-config is shown.

```
example:/file/cfg/#> show running-config meta vendor
Westermo
```

Whereas the full configuration can be viewed with:

```
example:/file/cfg/#> show running-config
```

A configuration file on external media can be viewed with:

```
example:/file/cfg/#> show media external example.cfg
```

Erasing a file

Erasing file on internal media:

```
example:/file/cfg/#> erase media internal example.cfg
```

Erasing file on external media:

```
example:/file/cfg/#> erase media external example.cfg
```

Erasing file on boot media:

```
example:/file/cfg/#> erase example.cfg/b>
```

Copy file to and from internal and external media

If an external media is mounted files can be copied from and to it.

Copying to internal media:

```
example:/file/cfg/#> copy running-config media internal example.cfg
```

Copying to external media:

```
example:/file/cfg/#> copy running-config media external example.cfg
```

Copying from external media to internal media:

```
example:/file/cfg/#> copy media external example.cfg media internal example.cfg
```

Copy file to and from network device

Exporting file to a network location via ftp:

```
example:/file/cfg/#> export example.cfg proto ftp user test_user password test_password 192.0.2.1
```

Exporting file to a network location via scp:

```
example:/file/cfg/#> export example.cfg proto scp user test_user password test_password 192.0.2.1
```

Importing file from a network location using ftp

```
example:/file/cfg/#> import proto ftp user test_user password test_password 192.0.2.1 example_remote.cfg example.cfg
```

4.5.4. Bootstrap System Configuration Over Network

About

This document aims to provide information on how to configure your device to obtain its configuration over the network.

For an overview of all available boot configuration options refer to the configuration guide:

- Boot Configuration Guide

For a more concrete use-case how to bootstrap the system configuration file over DHCP, see the following page:

- Bootstrap System Configuration Using DHCP

Introduction

Allowing the device to obtain its configuration over the network can help with centralizing the distribution of configurations for multiple devices on a larger network. Necessary configuration changes to devices on the network can simply be performed on the server containing the files, when the specific device reboots it will obtain the new configuration.

Another advantageous aspect is that it allows for easier substitution of devices, where a new device can simply obtain the same configuration as the last device that it replaced.

```
                                  .--------.
                                  |        |
                                  | Server |
                                  |        |
                                  '---+----'
                                      |
                                  .--.-.
                                 ( (    )__
                                (_,  \ ) ,_)  Internet/Intranet
                                  '-'--`--'
                                   |
---------+----------+----------+---------
         |          |          |
       .-+--.     .-+--.     .-+--.
       | H1 |     | H2 |     | H3 |
       '----'     '----'     '----'
```

Figure 1: The hosts are configured to obtain their configuration files from the server, somewhere on the network.

Configuration

In order to configure the system to attempt obtaining its startup configuration over the network, the boot `config-order` needs to be adjusted:

```
example:/#> boot
example:/boot/#> config-order internal:net
```

The first part of the configured option, **`internal`**, dictates that we should mount the internal media. The second part of the option, **`net`**, states that we should attempt to obtain the configuration over the network.

Additional settings specific to the net configuration can be controlled in its own sub-context:

```
example:/boot/#> net
example:/boot/net/#>
```

For more information on these options refer to the boot configuration options page.

Sync

The net sync option allows syncing any configuration file, obtained during the net boot, to be stored on the currently active configuration media. The specific media is determined based on how `net` was configured in the `config-order`, for example:

- `internal:net` -> If this config order option is active we will sync the obtained configuration file to the `internal` media definition.
- `internal:net` -> If this config order option is active we will sync the obtained configuration file to the `external` media definition.

Keep in mind that other media definitions can be added using the `media` command.

The net sync is enabled in the following manner:

```
example:/boot/#> net
example:/boot/net/#> sync
example:/boot/net/#> end
example:/boot/#>
```

This allows for configurations where we can use the last obtained configuration file from the server on the network, if it for some reason would become unavailable. In order to do this we set the next entry in the `config-order` to match the startup of the media used for the net boot configuration:

```
example:/boot/#> config-order internal:net, internal:startup
```

If the net boot time out, we bootstrap the internal startup file. Because we have enabled *sync* it will be the last obtained net configuration file.

> **NOTE**
>
> The `sync` option under the `net` context, is not the be mixed up with the `config-sync` option. These two are completely different functionalities.

Network Bootstrap Mode

When the device is configured to perform a network bootstrap of a configuration file, it will by default attempt to ask for a configuration file over active link connected to the device. The first file that it receive that it can safely apply to the device will be used.

Further, when the device in network boot mode it is still possible to receive a login prompt to the device, if connected with console cable. This makes it possible to log on to the device even tough the file have not yet been received, this can among other things be useful for debugging.

```
netboot-12-34-50 login: admin

.--.--.--.____.____.______.____.___.______.____.
|  |  |  |  -__|__  __|_     _|  -__|   _|       |  _  |
|_____|____|____| |__| |____|__| |__|__|__|____|
                                        www.westermo.com

.....................[ NOTICE ].......................
. Network boot in progress, awaiting a valid startup  .
.     configuration to be downloaded and applied      .
```

```
.......................................................

netboot-12-34-50:/#>
```

When logging into the device when net boot is in progress a **Notice** will be displayed informing the user. In addition the hostname can also be used to identify that the device is in net bootstrap mode. Set to *netboot* and the MAC address of the device without the Organizationally Unique Identifier. In the example above it is `netboot-12-34-50`.

Once a valid configuration is obtained or if the net boot times out and the next **`config-order`** option is applied, that configuration file will be seamlessly applied to the device.

While the system is in net boot mode, login credentials are the default admin/admin. The password to the device while in network bootstrap mode can also be changed in the following manner:

```
example:/boot/#> net
example:/boot/net/#> password Example
```

4.5.5. Bootstrap System Configuration with DHCP

About

This document presents a simple use-case where DHCP is used to bootstrap the system startup configuration.

For an overview on all boot configuration options in the system refer to the following documents:

- Boot Configuration Guide.
- Bootstrap configuration over network.

Introduction

This use-case aims to present a simple, yet effective, use-case where we obtain our startup configuration over the net using DHCP. When using DHCP for this purpose the client will request that it is provided with DHCP option 66 (TFTP Server Name) and 67 (Bootfile Name). This means that the DHCP server used must support these options.

Two different cases will be presented in this document. The major difference between these is the location of the server on the network. This is one of the strengths of using DHCP to bootstrap instead of BOOTP, is that it is possible to route the request to a server on another network from the client. This Requires a DHCP relay to be used as well.

Case 1: Server on the Same Subnet

In the case presented here, the server managing the configuration files will be located on the same subnet as the clients.

```
.--------.                  .--------.
| DHCP-  |                  | TFTP-  | Files: h1-config.cfg
| Server |                  | Server |        h2-config.cfg
|        |                  |        |
'---+----'                  '---+----'
    |.1                         |.5
    |                           |         192.168.1.0/24
----+----+----------------+----+-----------------------
         |                |
       .-+--.           .-+--.
       | H1 |           | H2 |
       '----'           '----'
```

Figure 1: Two host devices configured to bootstrap their configuration over the network using DHCP. Both the DHCP server and Tftp server are located on the same subnet.

We want host *H1* to obtain its configuration based on its Client ID, while H2 should obtain it based on its mac address.

> **NOTE**
>
> This example just assumes any arbitrary TFTP server.

Configuration

We need to configure our two hosts and DHCP server. The TFTP server configuration is assumed for this example.

Hosts

Configure host H1:

```
H1:/#> boot
H1:/boot/#> config-order internal:net
H1:/boot/#> net
H1:/boot/net/#> clientid host-h1
H1:/boot/net/#> mode dhcp
H1:/boot/net/#> no sync
H1:/boot/net/#> end
H1:/boot/#> end
H1:/#>
```

Configure host H2:

```
H2:/#> boot
H2:/boot/#> config-order internal:net
H2:/boot/#> net
H2:/boot/net/#> no clientid
H2:/boot/net/#> mode dhcp
H2:/boot/net/#> no sync
H2:/boot/net/#> end
H2:/boot/#> end
H2:/#>
```

DHCP Server

Configure the DHCP server to provide a specific bootfile for each of the hosts. Host H1 should match on its Client ID, host H2 should match on its MAC address:

```
dhcp-server:/#> configure
dhcp-server:/config/#> dhcp-server
dhcp-server:/config/dhcp-server/#> tftp-server 192.168.1.5
dhcp-server:/config/dhcp-server/#> subnet 192.168.1.0/24
Created new DHCP server for subnet 192.168.1.0.
dhcp-server:/config/dhcp-server/subnet-192.168.1.0/#> no pool
Disabling dynamic IP allocation for this subnet, only static hosts allowed.
dhcp-server:/config/dhcp-server/subnet-192.168.1.0/#>end
dhcp-server:/config/dhcp-server/#>
dhcp-server:/config/dhcp-server/#> host 1
dhcp-server:/config/dhcp-server/host-1/#> address 192.168.1.10
dhcp-server:/config/dhcp-server/host-1/#> match clientid string host-h1
dhcp-server:/config/dhcp-server/host-1/#> bootfile h1-config.cfg
dhcp-server:/config/dhcp-server/host-1/#> end
dhcp-server:/config/dhcp-server/#>
dhcp-server:/config/dhcp-server/#> host 2
dhcp-server:/config/dhcp-server/host-2/#> address 192.168.1.20
dhcp-server:/config/dhcp-server/host-2/#> match mac 52:54:00:12:34:56
dhcp-server:/config/dhcp-server/host-2/#> bootfile h2-confg.cfg
dhcp-server:/config/dhcp-server/host-2/#> end
dhcp-server:/config/dhcp-server/#>leave
dhcp-server:/#>
```

Case 2: Server on Another Subnet

In this case the DHCP and TFTP server is located on a separate network from the hosts. In order for this particular case to work, a DHCP Relay is necessary between the hosts and the servers. This allows us to route the traffic to the correct destinations.

```
.--------.                  .--------.
| DHCP-  | GW: .2           | TFTP-  | GW: .2
| Server |                  | Server | Files: h1-config.cfg
|        |                  |        |        h2-config.cfg
'---+----'                  '---+----'
    |.1                         |.5
    |                           |            10.0.0.0/24
----+------------+-------------+-----------------------
                 |
               |.2
           .---+----. vlan2
             | DHCP-  |
             | Relay  |
             |        |
             '---+----' vlan1
                 |.2
       |                       192.168.1.0/24
---------+-------+--------+---------------------------
         |                |
       .-+--.           .-+--.
       | H1 |           | H2 |
       '----'           '----'
```

Figure 2: Two host devices configured to bootstrap their configuration over the network using DHCP. Both the DHCP server and Tftp server are located on another subnet, requiring a DHCP-Relay to be used.

The configuration of the host devices will not be any different from before, the hosts should not need to care about the setup of the rest of the network. As long as the DHCP-Relay and the servers are correctly configured everything should work as before.

For this case to work, the DHCP server must also be able to distribute a default gateway in its offer to the hosts. This is because the hosts will need a route to the relay located at 192.168.1.2. Otherwise they will not be able to fetch the their configuration files from the server, since they wont know the route to it. But this requires no additional configuration on the hosts, only the DHCP Server.

> **NOTE**
>
> Both the DHCP and TFTP server will need to have a route to the DHCP-relay at 10.0.0.2, handling the 192.168.1.0/24 net. It does not really matter if this is done using static or dynamic routes or a default gateway, as long as it has a route back. In this example we use default gateways.

Configuration

The configuration of the Hosts and DHCP server is mostly the same, except for a few adjustments to the DHCP server. In addition for this case a DHCP relay is also needed.

Hosts

Both host **H1** and host **H2** are configured in the same manner as they were in the example in case 1.

DHCP Server

The DHCP Server is for the most part configured in the same manner as in case 1. There is one difference in the server configuration, and that is the location of the `tftp-server`. Since the DHCP server and TFTP server are not separate nets from the hosts, this needs to be changed. Further, we also change the DHCP `server-port` to the non standard 6767. This because we want the DHCP relay to relay the DHCP messages it receives to that port.

```
dhcp-server:/#> configure
dhcp-server:/config/#> dhcp-server
dhcp-server:/config/dhcp-server/#> tftp-server 10.0.0.5
dhcp-server:/config/dhcp-server/#> server-port 6767
dhcp-server:/config/dhcp-server/#>
```

In addition, we also need the DHCP server to provide a gateway to the hosts towards the relay located at 192.168.1.2. This is necessary for the hosts to be able to download their configurations from the TFTP server:

```
dhcp-server:/config/dhcp-server/#> host 1
dhcp-server:/config/dhcp-server/host-1/#> gateway 192.168.1.2
dhcp-server:/config/dhcp-server/host-1/#> end
dhcp-server:/config/dhcp-server/#> host 2
dhcp-server:/config/dhcp-server/host-2/#> gateway 192.168.1.2
dhcp-server:/config/dhcp-server/host-2/#> leave
dhcp-server:/#>
```

Lastly, the server is also in need of a route back to the DHCP Relay. In this example we handle this with a default gateway route:

```
dhcp-server:/#> configure
dhcp-server:/config/#> ip
dhcp-server:/config/ip/#> route default 10.0.0.2
dhcp-server:/config/ip/#>leave
dhcp-server:/#>
```

> **NOTE**
>
> This example assumes the same is done for the TFTP server.

DHCP Relay

We configure the relay to forward any dhcp messages it receives on vlan1 towards the DHCP server located at 10.0.0.1:6767:

```
dhcp-relay:/#> configure
dhcp-relay:/config/#> dhcp-relay
dhcp-relay:/config/dhcp-relay/#> server 10.0.0.1:6767
dhcp-relay:/config/dhcp-relay/#> iface vlan1
dhcp-relay:/config/dhcp-relay/#> leave
dhcp-relay:/#>
```

4.5.6. External Media HowTo

Introduction

The use-case in this document covers how to set up the system to use configuration files from external media, e.g. **USB** or **SD-card**. It also details how to ensure the configuration is protected from any changes when the system is running, i.e. a read-only mode.

Directory Layout

In order to boot from an external media it needs to have to correct directory layout, otherwise the system will not be able to obtain the necessary files and will fail-over to a fallback.

The operating system assumes an external media device has the following file system layout:

```
/media/
|- config/
|  |- config0.cfg                          # Example name
|  |- startup-config.cfg   --> config0.cfg   # Symlink
|  `- startup-config.lnk                   # For fat based fs
|
`- certs/
   |- aacerts/
   |- cacerts/
   |- certs/
   |- crls/
   |- ocspcerts/
   |- ovpnkeys/
   `- private/
```

The `startup-config.cfg` file is a symbolic link to the currently active configuration file (or slot), here `config0.cfg`. On file systems like FAT, which do not support symbolic links, a Windows-style `.lnk` file is used instead. The file must be named `startup-config.lnk` and contain the name of the file it refers to, without any leading path component, in this example the text `config0.cfg` followed by an optional newline.

Populate Media

The media can be set up either offline, on a workstation using the above directory layout, or using the built-in `media sync` feature:

```
example:/#> media sync internal external
Copying files to external media ...
```

Now we can configure the boot config order to external, when we reboot we will read the configuration file from that media.

If the device is configured to boot using external media and an empty media is used, the media will be populated with factory default files at boot, provided the media is not set to be `read-only`, see more on that below.

Configuration

This section details how to set up the device to use configuration files from external media. Appropriate fallback methods are also discussed. All available boot configuration options can be found here:

- Boot Configuration Guide

Configure the systems media definitions

In order for the system to be able to recognise and use any external media device, they need to be defined using the `media` command. This command will allow the user to set up how a media device

will be accessed on the device, and to what real physical device and/or partition on said device it will map to. By default two different media definitions exist:

- **`internal`:** Special definition and representation of the built-in flash, this media definition cannot be changed or removed.
- **`external`:** Base definition aimed to match the first partition located on any connected external physical media. If no partitions exist on the device this will also be a match in that case.

A more comprehensive example of how to make external media devices accessible on the system can be seen in the Access External Media Devices howto.

For additional information how to set up a media definition see the **`media`** command under the Boot Configuration Guide.

Set Up the Systems Configuration Order

The following commands define the order in which the system will attempt to obtain its startup configuration file.

```
example:/#> boot
example:/boot/#> config-order external:startup internal:startup
```

If an external media cannot be found the system will use the internal startup-config. Which can be useful to trigger a fail-over in case the external media is missing or damaged.

If the **`config-order`** setting only contains **`external`** and no media can be found, the system will trigger an alarm and start with the configured fallback:

```
example:/boot/#> config-fallback failsafe
```

The fail-safe option configured here starts the system with the `safe-config` configuration. Other fallback alternatives are: `factory` which uses the `factory-config` configuration, `none` which uses the `no-config` configuration, or `reboot` which causes the device to reboot until any of the media in **`config-order`** can be accessed.

Sync Configuration Files to Internal Media

The optional `config-sync` setting will when enabled automatically sync its configuration and certificates files to the systems built-in flash, represented by the **`internal`** media definition. The **`config-sync`** setting requires the user to provide the name of the external media to sync from. This action will then be performed during the initial part of the boot process.

This functionality allows for a configuration of the system where the external media would only need to be inserted once during the initial deployment. The easiest way to configure the system to act in this manner, can be achieved like this:

```
example:/boot/#> config-sync external
example:/boot/#> config-order internal:startup
```

Utilizing this setup would cause the system to copy any existing configuration from any inserted external media to the internal media, during the next reboot. It has also been configured with the **`config-order`** set to **`internal:startup`**. Therefore, the system will boot from the internal configuration, that now is the transferred configuration from the external media.

If the external media now is removed, the next time the system boots it will still boot according to the configuration defined by the `config-order` setting. Regardless if the `config-sync` flag is still set or not, this setting only cares about transferring the files, not how the system will actually boot.

Read-only Media

Set up any external media to be read only when booting:

```
example:/boot/#> media external
example:/boot/media-external/#> read-only
example:/boot/media-external/#> end
example:/boot/#>
```

Note: when a media has been configured read-only it must have the appropriate files present on the media, since no new information can be written to it when it has booted. This includes certificate files as well.

> **NOTE**
>
> The system does not allow any executable files, located on mounted external media, to be executed.

Customize Special Configuration Files

When defining the systems configuration order using the `config-order` command a media and a mode/file is always specified. The configuration files that are applied by the system in this case have a default that will be applied, unless the user override these. These are the names of the files mapped against the file/mode and the intended usage of their default files:

Mode/File	Corresponding Config File	Intended Usage
startup	startup-config.cfg	The normal startup configuration file of the system. This file is always updated anytime confiugration changes are saved normally on the device.
factory	factory-config.cfg	This configuration file contains the factory default configuration. Using this mode will cause the system to boot as if it were in its original factory state.
none	no-config.cfg	This configuration file is intended to be as the name indicates, as a *no config*. The default file will ensure that the system is started with ecentially no services enabled and all the ports of the device will be configured **disabled**. Hence, the device will only be accessible with a console connection.
safe	safe-config.cfg	This configuration file is intended to start the system in a *safe* state, where each port on the device will be individually separated from the rest. Each of these port interfaces will have DHCP enabled on them so that it is possible to assign them an IP address to make the device accessible. This so that the device should be recovrable since the user should be able to provide it with an accessible IP address.

Mode/File	Corresponding Config File	Intended Usage
net	net-config.cfg	This configuration file is applied on the system when it enters net boot mode. During the net boot process the device enters a special state where this configuration is applied in order to bring the system to a point where it is possible for it to obtain is startup configuration remotely. The default file will ensure that the device asks for a DHCP address on each port, independent of each other. Since it could be possible that the DHCP server is located on any of the ports, we need to send a discover on all of them.

In order to override these files it is possible to add a replacement manually in the intended configuration folder on the intended boot media. The folder structure was described earlier in this document. The replacement file simply needs to have the same name as the built-in default files described above. Anytime a replacement file is present on the selected configuration media, it will be used instead of the built-in default.

This could be especially useful for the net boot configuration file `net-config.cfg`, when the method of obtaining the configuration file needs to be more specific. Considering the default mentioned above, a DHCP discover will be sent on every port on the device. If you for instance want the initial discover to be sent on only a particular port, this could be achieved by adding an override version of the `net-config.cfg` under the configuration folder of the selected media.

Verify Setup

In order to see if the boot configuration looks like intended:

```
example:/boot/#> show
Bootstrap Configuration
Config-sync     : Disabled
Config-fallback : none

Config order
#  MEDIA                        MODE
1  internal                     startup


Certificate order
#  MEDIA
1  internal


Media Configuration
#  NAME                         R/O  TIMEOUT  MATCH
1  internal                     No   -------  System Internal
2  external                     No   30       Partition number: 1

Net Configuration
Sync            : Disabled
Timeout         : 300
Mode            : dhcp
Client ID       :

Bootloader Configuration
```

```
Login Password  : Disabled

Rescue Mode
    Address     : None
    Netmask     : None
    Peer        : None
    Port        : -2
```

4.5.7. Secure Boot unlock using tokens

About

This document provides instructions on configuring your device to use vendor-provided unsigned images.

For an overview of all available boot configuration options refer to the configuration guide:

- Boot Configuration Guide

Introduction

By allowing a device to ignore the security of the loaded OS, one can test various unsigned images. This can typically be needed in debug situations or when the vendor provides trial solutions as images.

The available unlock methods vary depending on the device's hardware security capabilities and are mutually exclusive:

1. **Unlock License Method**: For devices using a hardware anchor, i.e.: secure boot with a hardware based root-of-trust, only a cryptographically correct license file will unlock this bypass. This license token can be acquired from your vendor using credentials from the specific device in question. This is called the unlock-license function.
2. **Allow Untrusted Method**: Devices that do not use a hardware anchor can use the allow-untrusted function.

Configuration

In order to configure the system to ignore the security, the following parameters can be adjusted. For more information on these options refer to the boot configuration options page.

allow-untrusted

When a device without hardware anchor is used (hardware anchor indicates disabled), this option is used.

The allow-untrusted is enabled in the following manner:

```
example:/boot/#> loader
example:/boot/loader/#> allow-untrusted
example:/boot/loader/#> end
Saving bootloader configuration to FLASH
100% / [═══════════════════════════════════════════════════════════════════]
example:/boot/#>
```

After reboot a warning message should be displayed by the bootloader.

```
Found allow untrusted token. Ignoring image signatures .......... [WARN]
```

unlock-license

When a device with hardware anchor is used (hardware anchor indicates enabled), this option is used. For this to work, you need to a accuire a unlock license. To do so, you will need to send the product article number and the unique IDs presented in `show system-information`. Any mismatch with the unlock license will be displayed during the import and will result in a the function not being enabled.

```
[no] unlock-license [media <label>] [proto <ftp|tftp|scp|sftp|http|https>] [user
<username>] [password <password>] address[:port] FILENAME.json
```

Description

Starts an upgrade process based on the parameters given.

Options

Option	Description
`no`	Remove set license.
`media`	Label of the media (USB) to import from. Alternative to specifying a file transfer protocol and address.
`proto`	Protocol to use. ftp, tftp, scp, sftp, http or https.
`user`	Username if used by protocol.
`password`	Password if used by protocol.
`address[:port]`	IP-address and optional port of the server from which the image/package should be retrieved.
`FILENAME.json`	The name of the license file. It must be a JSON file and contain the correct format.

The unlock-license is enabled in the following manner:

```
example:/boot/#> loader
example:/boot/loader/#> unlock-license proto tftp 192.168.2.1 licensefile.json
example:/boot/loader/#> end
Saving bootloader configuration to FLASH
100% / [═══════════════════════════════════════════════════════════════════]
example:/boot/#>
```

After reboot a warning message should be displayed by the bootloader.

```
Found valid unlock license. Ignoring image signatures ........... [WARN]
```

4.6. Alarm, LEDs and Logging

4.6.1. Logging to Remote Machines HowTo

About

The goal of this document is to provide a comprehensive guide on how to configure client devices to send logging messages to a remote logging server. This process, known as remote logging, is a crucial aspect of maintaining and monitoring the health of distributed systems.

In this guide, we will walk you through the steps required to set up remote logging from a device. We will cover different protocols that can be used for transmitting log data, specifically UDP, TCP, and TLS. Each of these protocols has its own advantages and trade-offs in terms of speed, reliability, and security, and we will discuss these in the context of remote logging.

By the end of this guide, you will have a clear understanding of how to configure remote logging on a WeOS device, and how to choose the most appropriate protocol for your specific needs and environment.

> **SEE ALSO**
>
> Related Documentation
> - Logging Configuration
> - Logging and DHCP Option 7
> - Logging Relay

Introduction

The system generates a wealth of valuable data in the form of log messages. In a network comprising numerous devices, centralizing these log messages on a remote logging server can be beneficial. This consolidation of messages onto a single receiver simplifies the monitoring and analysis of the system's behaviour in a distributed setup.

Moreover, directing log messages to a remote server enables the storage of a larger volume of log messages over an extended period. This is particularly useful considering the internal log storage on the device is limited, both in terms of capacity and persistence, as log messages do not survive reboots.

For this use case, we consider the topology shown in Figure 1 below. There is a dedicated syslog server and a switch `S1` that is configured to send log messages to the syslog server. In this simplified view we focus on only one switch, but the same principles apply to multiple devices in a network.

```
                    .---------.
                    | Syslog  |
                    | Server  |
                    |         |
                    '----+----'
                         |.99
                         |
                         |
------------------+------------------ 192.168.1.0/24
                         |
                         |
                         |.10
                    .----+----.
                    |         |
                    |   S1    |
                    |         |
                    '---------'
```

Figure 1: Example network topology with a syslog server and a switch S1 configured to send log messages to the syslog server.

For the use case presented in this guide, how to configure each of the available transport protocols (UDP, TCP, and TLS) for remote logging will be presented. For your specific use case, you should choose the most appropriate protocol based on your requirements and constraints. Some considerations for choosing different transport types for remote logging include:

- **Security Requirements:** If security is not a primary concern for the logging data, or if it is handled through other means, UDP or TCP can be used. These protocols transmit messages in clear text. However, if security is a concern, TLS should be used. TLS encrypts the messages before they are sent over the network, providing an additional layer of security.
- **Server Capabilities:** The log server's capabilities may dictate the transport type. If the server does not support TLS, TCP or UDP may be the only options. It's important to verify the server's compatibility with the chosen transport protocol.
- **Network Reliability:** If the network is unreliable, TCP or TLS (which is built on top of TCP) should be used. These protocols provide mechanisms to ensure all log messages are delivered. UDP, on the other hand, does not guarantee delivery and may be more suitable for reliable networks or scenarios where occasional message loss is acceptable.
- **Performance Needs:** UDP generally has lower overhead than TCP or TLS, making it faster and more efficient in terms of network resources. If performance is a critical factor and security or guaranteed delivery is not a concern, UDP may be the preferred choice.
- **Regulatory Compliance:** Certain industries or applications may have specific regulatory requirements that dictate the use of secure transport protocols. In such cases, TLS would likely be the required choice.

In conclusion, the choice of transport protocol for remote logging depends on a variety of factors, including security requirements, server capabilities, network reliability, performance needs, and regulatory compliance. Understanding these factors will help you make an informed decision that best suits your specific use case.

In the next section, we will delve into the practical aspect of this guide. We will walk you through the step-by-step process of configuring the switch `S1` for remote logging using each of the transport protocols discussed in this section.

Configuration

In this section, we will cover the configuration steps required to set up remote logging on the switch `S1`. It is quite straightforward, we need to set up a logging source, destination and then lastly the sink, that will connect the source and destination components.

> **NOTE**
>
> For this configuration example, we assume that we already have the underlying interfaces with associated IP addresses configured on the switch.

In this example we do not use any logging filters. This means that all log messages generated by the switch will be sent to the remote syslog server.

Step 1: Source Configuration

We want the source of the log messages to be from the switch `S1` itself.. So we configure a source that we will call **local**:

> **NOTE**
>
> By default, from the factory default configuration, a logging source named **local** is already present on the switch. But for the sake of this example, we will create the source just to illustrate the configuration process.

```
S1:/#> configure
S1:/config/#> logging
S1:/config/logging/#> source local
S1:/config/logging/source-local/#> userspace
S1:/config/logging/source-local/#> kernel
S1:/config/logging/source-local/#> end
S1:/config/logging/#>
```

We specified that our source **local** should include both userspace and kernel messages in the logs generated by the source. The next step is to configure the destination for these log messages.

Step 2: Destination Configuration

We will configure a destination, that we will call **remote**, to send the log messages to the syslog server located at IP address **192.168.1.99**:

```
S1:/config/logging/#> destination remote
S1:/config/logging/destination-remote/#> type remote
S1:/config/logging/destination-remote/#> host address 192.168.1.99
```

> **TIP**
>
> It is also possible to specify a hostname instead of an IP address for the destination host. If we assume that the syslog server has a hostname **syslog-server**, we can specify it as follows:
>
> ```
> S1:/config/logging/destination-remote/#> host name syslog-server
> ```

> **NOTE**
>
> By default the IP-Protocol to use for the remote logging is **IPv4**. If you want to specify **IPv6** as the IP-Protocol, you can do so by setting the protocol:
>
> ```
> S1:/config/logging/destination-remote/#> ip-protocol ipv6
> ```
>
> Then we can specify the IPv6 address of the syslog server:
>
> ```
> S1:/config/logging/destination-remote/#> host address 2001:db8::1
> ```

Depending on which transport protocol to use for the remote logging, the configuration for each protocol is as follows:

- **UDP:** For UDP we simply specify the transport type as UDP:

  ```
  S1:/config/logging/destination-remote/#> transport udp
  ```

- **TCP:** For TCP we specify the transport type as TCP:

  ```
  S1:/config/logging/destination-remote/#> transport tcp
  ```

- **TLS:** For TLS we first specify the transport type as TLS:

  ```
  S1:/config/logging/destination-remote/#> transport tls
  ```

  Now, depending on a few factors, we may need to specify a certificate and/or a ca-certificate for the TLS connection.

  > **NOTE**
  >
  > We assume for the sake of this example that the appropriate certificates are already present on the switch. Refer to the Certificate Management documentation for more information on how to manage certificates on the switch.

  The following is how to handle the certificates for the TLS connection:

  - **Certificate:** If the syslog server requires a client certificate for authentication, we can specify the certificate to use:

    ```
    S1:/config/logging/destination-remote/#> certificate my-example-cert
    ```

    If the server does not require a client certificate, this step can be skipped. However, for optimal security, it is recommended to use a client certificate if the server supports it.

  - **CA Certificate:** Depending on how the the `peer-verify` option has been configured, we may need to specify a CA certificate to verify the server's certificate. If the `peer-verify` option is set to require a trusted CA certificate:

```
S1:/config/logging/destination-remote/#> peer-verify trusted
```

A certificate must be provided by the server during the TLS handshake and this certificate must be verifiable by the CA certificate that is specified on the client side:

```
S1:/config/logging/destination-remote/#> ca-certificate my-example-ca-cert
```

If the `peer-verify` option is set to `untrusted`, the CA certificate is not required. In this case, it is not necessary to verify the server's certificate against the CA certificate. However, for optimal security, it is recommended to use a CA certificate to validate the server's certificate.

> **NOTE**
>
> By default, the port used for remote logging will be set to the default for the selected transport protocol. If you want to specify a different port, you can do so by using the `port` command.

When we have configured the selected transport protocol, we can leave the destination configuration context.

```
S1:/config/logging/destination-remote/#> end
S1:/config/logging/#>
```

The final step is to configure the sink that will connect the source and destination components.

Step 3: Sink Configuration

Now we will configure our sink, that we will call **remote** to connect the source and destination components:

```
S1:/config/logging/#> sink remote
S1:/config/logging/sink-remote/#> source local
S1:/config/logging/sink-remote/#> destination remote
S1:/config/logging/sink-remote/#> end
S1:/config/logging/#>
```

The configuration overview should now look something like this:

```
S1:/config/logging/#> show
Console syslog   : Disabled
External logging : Disabled
Secure mode      : Enabled

 Sources
 NAME
 Type: local      USERSPACE  KERNEL  AUDIT
 local            Yes        Yes     No

 Destinations
 NAME
 Type: remote     TRANSPORT  TO-HOST       PORT   FORMAT
 remote           udp        192.168.1.99  514    rfc5424
```

```
Filters
NAME              INVERT        TYPE   FILTER

No filter(s) currently configured ...

Sinks
NAME              SOURCES       DESTINATIONS    OP FILTERS
remote            local         remote

S1:/config/logging/#>
```

This should be enough to get the remote logging up and running on the switch, so we exit the configuration context, to apply the configuration:

```
S1:/config/logging/#> leave
S1:/#>
```

Verification

At this point any log message generated by the switch `S1` should be sent to the syslog server located at IP address **192.168.1.99**. To verify that the configuration is working as expected, you can check the syslog server for any incoming log messages from **192.168.1.10**, or we can generate a test log message on the switch and check if it is sent to the syslog server:

> **NOTE**
>
> To access the system shell, you must connect to the device via the console using an administrative account with the appropriate permissions. While it is possible to access the shell via SSH, this feature must first be enabled through a console session.

```
S1:/#> shell
Domain shell started.
admin@example:~ # logger -t proc -I 123 -p user.notice "This is a test log message"
admin@example:~ #
```

When checking for the log message on the server side, it should look something like this:

```
May 22 13:34:04 192.168.1.10 proc[123]: This is a test log message
```

The logging status on the device can also be checked to see if the log messages are being processed correctly:

```
S1:/#> show logging destination
Destinations
TYPE    NAME         DESCRIPTION           PROCESSED  DROPPED  WRITTEN  QUEUED  MEMORY_USAGE
file    ntp          /var/log/ntp.log      2          0        2        0       0
file    syslog       /var/log/syslog       125        0        125      0       0
file    auth         /var/log/auth.log     3          0        3        0       0
file    alarm        /var/log/alarm        1          0        1        0       0
file    messages     /var/log/messages     88         0        88       0       0
file    kern         /var/log/kern.log     37         0        37       0       0
remote  sink-remote  udp,192.168.1.99:514  106        0        106      0       0
```

```
└──────────────────────────────────────────────────────────────────────────────┘
S1:/#>
```

As per the output above, the **sink-remote** destination, the number of **PROCESSED** and **WRITTEN** log messages should be the same, indicating that the log messages are being sent to the remote syslog server correctly.

If for instance the server would not be reachable, the number of **QUEUED** log messages would increase, indicating that the log messages are being queued up for transmission to the server when it becomes reachable again.

> **NOTE**
>
> If the syslog server is not reachable, when the switch's configuration is applied, the switch will not be able to send the log messages to the server. When the server becomes reachable it may take a short while before the switch attempts to connect to the server again.

Of course, it is always a good idea to simply check if the syslog server is reachable from the switch:

```
S1:/#> ping 192.168.1.99
Press Ctrl-C to abort PING 192.168.1.99 (192.168.1.99) 56(84) bytes of data.
64 bytes from 192.168.1.99: icmp_seq=1 ttl=64 time=0.852 ms
64 bytes from 192.168.1.99: icmp_seq=2 ttl=64 time=0.766 ms
^C
--- 192.168.1.99 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.766/0.809/0.852/0.043 ms
S1:/#>
```

If it is not reachable, you should check the network configuration on the switch, server, and the network in between.

Conclusion

In this HowTo, we have gone through the necessary steps to configure remote logging, in order to transmit all log messages generated by the device to a remote syslog server.

This is, of course, a simplified example, but it should give you a good starting point for setting up remote logging on your device. The configuration can be further customised to suit your specific requirements. For a comprehensive list of configuration options, refer to the Logging Configuration documentation.

4.6.2. Logging and DHCP Option 7

About

The goal of the use-case presented in this document is to demonstrate how to configure a network where WeOS devices are configured to send log messages to a remote syslog server, that is provided by a DHCP lease, that includes option 7 (log server).

We will walk through the steps to configure the necessary logging sinks on the client devices. Furthermore, we will also provide an example of how to configure a DHCP server on a WeOS device to provide the necessary DHCP option 7 in the lease.

> **SEE ALSO**
>
> Related Documentation
> - Logging Configuration
> - Logging to Remote Machines HowTo
> - DHCP Server Configuration

Prerequisites

This document assumes that you have a basic understanding of the WeOS logging system. If you are not familiar with the logging system, please refer to the Logging Configuration page.

In addition, it is good to have a basic understanding of how to configure a DHCP server on a WeOS device. If you are not familiar with the DHCP server configuration, please refer to the DHCP Server Configuration

Introduction

For more intricate detail when it comes to logging to remote machines, please refer to the Logging to Remote Machines HowTo. In this use-case, we will focus on configuring logging destinations that can be set to dynamically assign the IP address of the intended remote syslog server, using DHCP option 7 (log server).

Consider the simplified network topology in **Figure 1**, below.

```
  .---------.           .---------.
  |  DHCP-  |           | Syslog- |
  | Server  |           | Server  |
  |         |           |         |
  '----+----'           '----+----'
       |.10                  |.20
       |                     |
       |                     |
-------+----------+----------+------- 192.168.1.0/24
                  |
                  |
                  | vlan1: DHCP
             .----+----.
             |         |
             |   S1    |
             |         |
             '---------'
```

Figure 1: Example of a network with a DHCP server and a syslog server. The DHCP server is configured to provide the switch S1 with a lease that includes the syslog server's IP address, provided by DHCP option 7 (log server).

For this use-case, we will configure the switch S1 to send log messages using UDP, for the sake of simplicity, to the syslog server located at IP address **192.168.1.20**. We will configure the DHCP

server to provide the switch S1 with a lease that includes the syslog server's IP address, using DHCP option 7.

In the next section will will cover how to configure the devices **DHCP-Server** and **S1** to achieve this. We will start with the configuration of the DHCP server.

Configuration

We will focus on the configuration of both the DHCP server and the switch S1.

DHCP Server configuration

We will configure a DHCP server to provide an IP address to the switch from a pool of addresses within the subnet **192.168.1.0/24**. Additionally, we will configure the DHCP server to provide option 7, specifying **192.168.1.20** with every lease that is provided from the server, using the `log-server` command.

> **NOTE**
>
> We assume that the underlying network is already configured, with correct interfaces and addresses.

The configuration of the DHCP server is as follows:

```
DHCP-Server:/#> configure
DHCP-Server:/config/#> dhcp-server
DHCP-Server:/config/dhcp-server/#> subnet 192.168.1.0/24
DHCP-Server:/config/dhcp-server/subnet-192.168.1.0/#> pool 192.168.1.100 10
DHCP-Server:/config/dhcp-server/subnet-192.168.1.0/#> end
DHCP-Server:/config/dhcp-server/#> log-server 192.168.1.20
DHCP-Server:/config/dhcp-server/#> leave
DHCP-Server:/#>
```

> **TIP**
>
> The `log-server` can also be specified per `subnet` and `host` in addition to the *global* level.

For this simple example, we do not need to configure much more than this in order to provide the switch **S1** with the necessary lease.

Switch S1 configuration

Now we can configure the switch **S1** to set up a remote logging sink, which destination is set to obtain the IP address from the DHCP lease. We can configure the switch in the following way:

```
S1:/#> configure
S1:/config/#> iface vlan1
S1:/config/iface-vlan1/#> inet dhcp
S1:/config/iface-vlan1/inet-dhcp/#> end
S1:/config/iface-vlan1/#> end
S1:/config/#>
S1:/config/#> logging
S1:/config/logging/#> source local
S1:/config/logging/source-local/#> userspace
S1:/config/logging/source-local/#> kernel
S1:/config/logging/source-local/#> end
```

```
S1:/config/logging/#>
S1:/config/logging/#> destination remote
S1:/config/logging/destination-remote/#> type remote
S1:/config/logging/destination-remote/#> transport udp
S1:/config/logging/destination-remote/#> host dhcp vlan1
S1:/config/logging/destination-remote/#> end
S1:/config/logging/#>
S1:/config/logging/#> sink to-remote
S1:/config/logging/sink-to-remote/#> source local
S1:/config/logging/sink-to-remote/#> destination remote
S1:/config/logging/sink-to-remote/#> end
S1:/config/logging/#>
S1:/config/logging/#> show
Console syslog   : Disabled
External logging : Disabled
Secure mode      : Enabled

╒ Sources ═══════════════════════════════════════════════════════
NAME
Type: local      USERSPACE  KERNEL
local            Yes        Yes

╒ Destinations ══════════════════════════════════════════════════
NAME
Type: remote     TRANSPORT  TO-HOST  PORT   FORMAT
remote           udp        vlan1    514    rfc5424

╒ Filters ═══════════════════════════════════════════════════════
NAME             INVERT        TYPE  FILTER

No filter(s) currently configured ...

╒ Sinks ═══════════════════════════════════════════════════════
NAME             SOURCES       DESTINATIONS   OP FILTERS
to-remote        local         remote

dhcp:/config/logging/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
S1:/#>
```

> **TIP**
>
> A source called **local** should be present on the switch by default, from the default factory configuration. For the sake of this example, we explicitly configure the source to make it clear.

With this configuration, the switch **S1** will dynamically set the host IP address for the destination **remote**. This IP address is provided by a DHCP lease on the interface **vlan1**, if the lease includes the option 7 (log server).

> **NOTE**
>
> Currently, each logging destination that is configured in this manner can receive and apply up to two IP addresses per DHCP lease option 7.

Verification

With the current configuration, the switch **S1** should now be able to send log messages to the syslog server located at **192.168.1.20**, when it receives a lease from the DHCP server.

The first thing that we can verify is that the DHCP server has provided **S1** with a lease. If we check on the DHCP server, we should see that a lease has been provided to the switch:

```
DHCP-Server:/#> show dhcp-server
DHCP-Server:/#> show dhcp-server
DHCP server active, PID: 832

LEASE TIME  MAC ADDRESS       CLIENT ID            IP ADDRESS       HOSTNAME
864000      0c:b1:ec:09:00:00 01:0c:b1:ec:09:00:00 192.168.1.104    S1
```

If we check on **S1**, we should also see that the interface **vlan1** has received a lease:

```
S1:/#> show iface vlan1
Name         : vlan1
Admin. State : Up
Oper. Status : UP
MAC Address  : 0c:b1:ec:09:00:00
IP Addresses : 169.254.126.43/16  (link-local)
               192.168.1.104/24   (dhcp)      <------- IP address from DHCP
MTU          : 1500
Statistics   :
               RX: Packets    Bytes        Errors CsumErrs OutOfSeq Mcasts
                   64         8930         0      0        0        56
               TX: Packets    Bytes        Errors DeadLoop NoRoute  NoBufs
                   68         12341        0      0        0        0
```

Next, we can verify that our remote logging destination has the correct IP address as follows:

```
S1:/#> show logging destination remote
╒ Remote Destinations ═══════════════════════════════════════════════════
│NAME         REMOTE_PROTOCOL_IP_PORT  PROCESSED  DROPPED  WRITTEN  QUEUED  MEMORY_USAGE
│sink-remote  udp,192.168.1.20:514     13         0        13       0       0
└────────────────────────────────────────────────────────────────────────
```

If everything is configured correctly, the output should show that the destination **sink-remote** has been assigned the IP address of the Syslog server, which is **192.168.1.20**. In the output, we should also be able to see that our messages have been **WRITTEN** and **PROCESSED**.

If we have not yet received a lease, or if the provided lease does not include option 7, the status of the remote destinations will indicate that there are no active remote logging destinations:

```
S1:/#> show logging destination remote
╒ Remote Destinations ═══════════════════════════════════════════════════
│NAME  REMOTE_PROTOCOL_IP_PORT  PROCESSED  DROPPED  WRITTEN  QUEUED  MEMORY_USAGE
└────────────────────────────────────────────────────────────────────────
```

```
No entry currently found ...
```

This is because the remote logging destination instance will remain inactive until the switch receives a lease with option 7.

Finally, we can verify that the switch **S1** is able to send log messages to the syslog server, by generating a log message on the switch:

> **NOTE**
>
> To access the system shell, you must connect to the device via the console using an administrative account with the appropriate permissions. While it is possible to access the shell via SSH, this feature must first be enabled through a console session.

```
S1:/#> shell
Domain shell started.
admin@example:~ # logger -p notice "This is a test message"
admin@example:~ #
```

If the syslog server is correctly configured, the message should be received and logged by the syslog server.

Conclusion

The use of DHCP option 7 allows for dynamic configuration of logging destinations on WeOS devices. We demonstrated how to configure a DHCP server to provide a lease that includes the IP address of a syslog server. Additionally, we showed how to configure a WeOS device to dynamically use this IP address to specify the location of a remote syslog server.

4.6.3. Logging Relay HowTo

About

This document presents an example use-case for a logging relay setup. A logging relay acts as an intermediary, receiving log messages from hosts and forwarding them to another remote syslog server. This is particularly useful for hosts that can only send User Datagram Protocol (UDP) messages, which are not encrypted.

In our use-case, the logging relay receives these UDP messages and forwards them to a syslog server using Transport Layer Security (TLS). TLS provides encryption, ensuring that the log messages are secure during transit. This is an effective method for encrypting log messages when dealing with end devices that lack the capability to send encrypted messages themselves.

Furthermore, a logging relay can be a solution for transmitting messages over an untrusted network. By encrypting the messages, it mitigates the risk of sensitive log information being intercepted or tampered with.

> **SEE ALSO**
>
> Related Documentation
> - Logging Configuration
> - Logging to Remote Machines HowTo
> - Logging and Receiving From Remote Clients

Introduction

It is not uncommon for network devices to generate log messages that need to be stored and analysed. In some cases, these devices may not have the capability to send encrypted log messages. This is where a logging relay comes in. A logging relay acts as an intermediary, receiving log messages from hosts and forwarding them to the designated syslog server.

Consider a scenario similar to the one depicted in **Figure 1** below.

```
                    .---------.
                    | Syslog- |
                    | Server  |
                    |         |
                    '----+----'
                         |.99
                         |
                      .--.-.
                     ( (    )__
                    (_,  \ ) ,_)  Untrusted Network
                      '-'--`--'   172.16.1.0/24
                         |
                         |vlan2: .1
                    .----+----.
                    | Syslog- |
                    | Relay   |
                    |         |
                    '----+----'
                         |vlan1: .1
                         |
                         |
---+-------------+-------------+--- 192.168.1.0/24
   |           |             |
   |.10          |.20          |.30
.--+--.       .--+--.       .--+--.
| H1  |       | H2  |       | H3  |
'-----'       '-----'       '-----'
```

Figure 1: Example network topology with a syslog relay. The syslog relay receives log messages from hosts H1, H2, and H3, and forwards them to the syslog server.

In this example use-case we will focus on the configuration of a WeOS device to act as a the **Syslog-Relay** in the example topology. This entails configuring the device with a remote logging source, that will be combined with a remote logging destination.

The remote source will be configured to receive any UDP log message that reaches the device on the `vlan1` interface. The remote destination will be configured to send log messages to the **Syslog-Server** located at **172.16.1.99**, using TLS as the transport protocol.

> **NOTE**
>
> We assume that all the end devices are correctly configured to send their log messages to the **Syslog-Relay** at **192.168.1.1** using UDP as the transport protocol.

In the next section we will focus on the configuration of the **Syslog-Relay** device.

Configuration

We will focus on the configuration of the **Syslog-Relay** device. We take each step in turn, configuring the remote source, remote destination, and finally combining them in a sink configuration.

Prerequisites

We assume that the device is already configured with the necessary network settings, i.e. the IP addresses and the VLANs are correctly configured. For the example presented in the topology, we assume that the device has the following interface configuration:

- `vlan1`: Configured with an IP address in the **192.168.1.0/24** subnet.
- `vlan2`: Configured with an IP address in the **172.16.1.0/24** subnet.
- `hostname`: Is set to **Syslog-Relay**.

Remote Source

We start with the configuration of the remote source, that we will call **remote**:

```
Syslog-Relay:/#> configure
Syslog-Relay:/config/#> logging
Syslog-Relay:/config/logging/#> source remote
Syslog-Relay:/config/logging/source-remote/#> type remote
Syslog-Relay:/config/logging/source-remote/#> transport udp
Syslog-Relay:/config/logging/source-remote/#> listen vlan1
Syslog-Relay:/config/logging/source-remote/#> keep-hostname
Syslog-Relay:/config/logging/source-remote/#> end
Syslog-Relay:/config/logging/#>
```

This source configuration will listen for any UDP log messages that reach the device on the `vlan1` interface. We also enable the `keep-hostname` option to ensure that the hostname of the source is preserved in the log messages. Otherwise, when the messages are relayed to the syslog server, the hostname of the **Syslog-Relay** would be used instead. This option is useful for distinguishing between different sources when analysing the log messages.

When we want to have a remote source, we must also ensure that we disable the `secure-mode` option. While this is enabled, the device will not accept any remote log messages. Disable it in the top level logging configuration context:

```
Syslog-Relay:/config/logging/#> no secure-mode
```

This configuration is enough to receive the log messages from the devices on the **192.168.1.0/24** subnet. The next step is to configure the destination.

Remote Destination

Next, in order to relay the log messages to the **Syslog-Server** we need to configure a remote destination. We will call this destination **relay**:

```
Syslog-Relay:/config/logging/#> destination relay
Syslog-Relay:/config/logging/destination-relay/#> type remote
Syslog-Relay:/config/logging/destination-relay/#> transport tls
Syslog-Relay:/config/logging/destination-relay/#> host name Syslog-Server
Syslog-Relay:/config/logging/destination-relay/#> peer-verify trusted
Syslog-Relay:/config/logging/destination-relay/#> certificate my-cert
Syslog-Relay:/config/logging/destination-relay/#> ca-certificate my-ca-cert
Syslog-Relay:/config/logging/destination-relay/#> end
Syslog-Relay:/config/logging/#>
```

In this configuration we specify that we will reach the remote destination at the hostname **Syslog-Server** using TLS as the transport protocol. In addition, we specify a certificate to use and verify the client against the server.

We have also specified a CA certificate to use for verifying the server certificate, which will be necessary because the `peer-verify trusted` option has been set. This option requires that the server must also provide a certificate to the client, which will be validated against the specified CA certificate.

> **NOTE**
>
> Any necessary certificates must be imported into the device before configuring the remote destination. This includes the CA certificate, the client certificate. However, for the client the need for a certificate depends on how the syslog server is configured, if it requires a certificate for the client then it must be imported into the device.
>
> The client can also be configured the verify the server certificate, if the `peer-verify trusted` has been set. If this is the case the server also needs to provide a certificate to the client, that will be validated against the specified CA certificate.

We assume that we have already imported the necessary certificates into the device. Refer to the Certificate Management page for more information on how to import certificates into the device.

> **NOTE**
>
> Depending on how the certificates are generated, it may be necessary to specify the destination with its hostname. This is because the certificate may be generated with the hostname as the common name (CN). If this is the case, the hostname must be specified in the destination configuration.

In this example we have simply added the hostname as a static host entry in the device:

```
Syslog-Relay:/#> configure
Syslog-Relay:/config/#> dns
Syslog-Relay:/config/dns/#> host Syslog-Server 172.16.1.99
Syslog-Relay:/config/dns/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Syslog-Relay:/#>
```

This configuration should be enough to relay the log messages to the **Syslog-Server**.

Sink Configuration

The final step is to combine the source and destination in a sink configuration. We will call this sink **to-relay**:

```
Syslog-Relay:/config/logging/#> sink to-relay
Syslog-Relay:/config/logging/sink-to-relay/#> source remote
Syslog-Relay:/config/logging/sink-to-relay/#> destination relay
Syslog-Relay:/config/logging/sink-to-relay/#> end
Syslog-Relay:/config/logging/#>
```

The final configuration should look like this, when we check the overview from the top level logging configuration context:

```
Syslog-Relay:/config/logging/#> show
Console syslog   : Disabled
External logging : Disabled
Secure mode      : Disabled

 Sources
 NAME
 Type: local      USERSPACE  KERNEL
 local            Yes        Yes
 Type: remote     TRANSPORT  LISTEN                PORT   FORMAT
 remote           udp        vlan1 (192.168.1.1)   514    rfc5424

 Destinations
 NAME
 Type: remote     TRANSPORT  TO-HOST        PORT   FORMAT
 relay            tls        Syslog-Server  6514   rfc5424

 Filters
 NAME             INVERT        TYPE  FILTER

 No filter(s) currently configured ...

 Sinks
 NAME             SOURCES       DESTINATIONS   OP FILTERS
 to-relay         remote        relay

Syslog-Relay:/config/logging/#> leave
```

Now when we leave the configuration context, the configuration will be applied, and the device should start relaying the UDP log messages to the **Syslog-Server**.

```
Syslog-Relay:/config/logging/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Syslog-Relay:/#>
```

Verification

If everything has been configured correctly, we should be able to check the status of the logging source, with it looking something like this:

```
Syslog-Relay:/#> show logging source
 Sources
 TYPE        NAME             DESCRIPTION       PROCESSED
 remote      sink-remote      192.168.1.1       15
 unix-dgram  local-userspace  /dev/log          358
 file        local-kernel     /dev/kmsg         1

Syslog-Relay:/#>
```

We should see that we have a **remote** source active called **sink-remote**. If messages are being received, the **PROCESSED** counter should be increasing.

Next we can check the status of the logging destination:

```
Syslog-Relay:/#> show logging destination
 Remote Destinations
 NAME        REMOTE_PROTOCOL_IP_PORT  PROCESSED  DROPPED  WRITTEN  QUEUED  MEMORY_USAGE
 sink-relay  tls,Syslog-Server:6514   15         0        15       0       0

Syslog-Relay:/#>
```

Using `tcpdump` on the **Syslog-Relay** we can verify that the log messages are being received on the `vlan1` interface:

```
Syslog-Relay:/#> tcpdump iface vlan1 expr "proto 17 and port 514"
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on vlan1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
09:50:34.158829 IP 192.168.1.20.35038 > 192.168.1.1.syslog: SYSLOG user.notice, length: 76
09:50:37.414279 IP 192.168.1.30.45581 > 192.168.1.1.syslog: SYSLOG user.notice, length: 76
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
Syslog-Relay:/#>
```

If we see log messages being received on the `vlan1` interface, then the we can check that the messages are being relayed to the **Syslog-Server** by using `tcpdump` on the interface towards the **Syslog-Server**, interface `vlan2`:

```
Syslog-Relay:/#> tcpdump iface vlan2 expr "proto 6 and port 6514"
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on vlan2, link-type EN10MB (Ethernet), snapshot length 262144 bytes
09:54:26.278686 IP 172.16.1.1.41217 > server1.6514: Flags [P.], seq
1392486109:1392486134, ack 36688746, win 4005, length 25
09:54:26.279468 IP server1.6514 > 172.16.1.1.41217: Flags [.], ack 25, win 6560,
length 0
```

```
09:54:26.279493 IP 172.16.1.1.41217 > server1.6514: Flags [P.], seq 25:123, ack 1,
win 4005, length 98
09:54:26.280229 IP server1.6514 > 172.16.1.1.41217: Flags [.], ack 123, win 6560,
length 0
09:54:28.618872 IP 172.16.1.1.41217 > server1.6514: Flags [P.], seq 123:148, ack 1,
win 4005, length 25
09:54:28.620004 IP server1.6514 > 172.16.1.1.41217: Flags [.], ack 148, win 6560,
length 0
09:54:28.620042 IP 172.16.1.1.41217 > server1.6514: Flags [P.], seq 148:246, ack 1,
win 4005, length 98
09:54:28.620751 IP server1.6514 > 172.16.1.1.41217: Flags [.], ack 246, win 6560,
length 0
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
Syslog-Relay:/#>
```

If everything is working correctly, we should see something similar to the above.

Troubleshooting

The following is an example of some troubleshooting steps that can be taken if the configuration is not working as expected:

- On the **Syslog-Relay**, create a local file destination and add it to the same sink that we created to send messages to the **Syslog-Server**. This will allow us to verify that the messages are at least reaching the **Syslog-Relay** correctly, and are being processed by the sink.

  ```
  Syslog-Relay:/#> configure
  Syslog-Relay:/config/#> logging
  Syslog-Relay:/config/logging/#> destination local-file
  Syslog-Relay:/config/logging/destination-local-file/#> type file
  Syslog-Relay:/config/logging/destination-local-file/#> media internal
  Syslog-Relay:/config/logging/destination-local-file/#> file-name syslog-relay-debug.log
  Syslog-Relay:/config/logging/destination-local-file/#> end
  Syslog-Relay:/config/logging/#> sink to-relay
  Syslog-Relay:/config/logging/sink-to-relay/#> destination local-file
  Syslog-Relay:/config/logging/sink-to-relay/#> end
  Syslog-Relay:/config/logging/#> leave
  Applying configuration.
  Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
  ```

  Now any UDP log messages that are received by the **Syslog-Relay** should be written to the file **syslog-relay-debug.log**. We can check this file to see if the messages are being received correctly:

  ```
  Syslog-Relay:/#> show logfile syslog-relay-debug.log
  May 27 10:05:49 H2 admin: test 1
  May 27 10:05:51 H3 admin: test 2
\_\_\_
  ```

- Perform a similar check on the **Syslog-Server** to ensure that it is correctly configured to receive log messages over TLS. Verify if there are any errors or warnings in the log files that may indicate a problem with the TLS connection.
- Verify that the correct certificates have been imported into the device and that the correct certificates are being used in the configuration.
- Ensure that the correct port numbers are being used for the transport protocols (UDP and TLS), if either any of the clients or the server are using a different non-standard port number.
- Ensure that the configured host can be reached, using for example the `ping` command.

```
Syslog-Relay:/#> ping Syslog-Server
Press Ctrl-C to abort PING Syslog-Server (172.16.1.99) 56(84) bytes of data.
64 bytes from Syslog-Server (172.16.1.99): icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from Syslog-Server (172.16.1.99): icmp_seq=2 ttl=64 time=2.56 ms
^C
--- Syslog-Server ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.223/1.891/2.560/0.668 ms
client1:/#>
```

  If the host cannot be reached, then the device will not be able to relay the log messages to the syslog server.
- If a hostname is used to specify the destination, ensure that the hostname can be resolved correctly.

Conclusion

In this document we have presented an example use-case for a syslog relay setup. We have configured a WeOS device to act as a syslog relay, receiving log messages from hosts using UDP and forwarding them to a syslog server using TLS.

This setup is particularly useful for hosts that lack the capability to send encrypted log messages. By using a syslog relay, we can ensure that the log messages are secure during transit, even when transmitted over an untrusted network.

4.6.4. Logging and Receiving From Remote Clients

About

This document describes how to configure the logging system in WeOS to receive logs from remote clients. For this use case, remote log messages will be received and written to a file on the device. This is for the sake of demonstration and can be easily modified to suit your needs, e.g. sending logs to dedicated log servers, see the Logging Relay for an example of how to achieve something like that.

> **SEE ALSO**
>
> Related Documentation
> - Logging Configuration
> - Logging Relay

Prerequisites

This document assumes that you have a basic understanding of the WeOS logging system. If you are not familiar with the logging system, please refer to the Logging Configuration page.

Introduction

The WeOS logging system can be configured to receive log messages from remote clients. This can be useful for debugging purposes, monitoring the network, or relaying logs to a central log server.

In this example, we will configure a logging sink on the device to receive and write log messages from remote clients. Consider a scenario, similar to the one shown in Figure 1, where we have a number of hosts connected to a switch **S1**.

```
                    .---------.
                    |         |
                    |   S1    |
                    |         |
                    '----+----'
                         |vlan1: .1
                         |
                         |
---+-------------+-------------+--- 192.168.1.0/24
   |             |             |
   |.10          |.20          |.30
.--+--.       .--+--.       .--+--.
| H1  |       | H2  |       | H3  |
'-----'       '-----'       '-----'
```

Figure 1: Example network topology with a switch **S1** and a number of hosts. The switch S1 is connected to the hosts on the 192.168.1.0/24 subnet.

We will configure the switch **S1** to receive log messages from any host located in the **192.168.1.0/24** subnet, that are connected to its vlan1 interface.

In the next sections, we will cover the configuration steps required on the switch **S1**.

Configuration

The major focus of this example is to configure the switch to receive log messages from remote clients. We also need to specify a destination file where the logs will be written.

Step 1: Configure the Logging Sources

For this example scenario, we will configure the switch to be able to receive log messages from any host located on the subnet **192.168.1.0/24**, located on the **vlan1** interface. An individual source

definition will be created for each of the remote transport protocols UDP, TCP, and TLS. Then we can connect these three sources to a single sink.

UDP Source

We create a source called **all-udp** that listens on the **vlan1** interface for log messages from remote clients using UDP:

```
S1:/#> configure
S1:/config/#> logging
S1:/config/logging/#> source all-udp
S1:/config/logging/source-all-udp/#> type remote
S1:/config/logging/source-all-udp/#> transport udp
S1:/config/logging/source-all-udp/#> listen vlan1
S1:/config/logging/source-all-udp/#> end
S1:/config/logging/#>
```

TCP Source

We create a source called **all-tcp** that listens on the **vlan1** interface for log messages from remote clients using TCP:

```
S1:/config/logging/#> source all-tcp
S1:/config/logging/source-all-tcp/#> type remote
S1:/config/logging/source-all-tcp/#> transport tcp
S1:/config/logging/source-all-tcp/#> listen vlan1
S1:/config/logging/source-all-tcp/#> end
S1:/config/logging/#>
```

TLS Source

We create a source called **all-tls** that listens on the **vlan1** interface for log messages from remote clients using TLS:

```
S1:/config/logging/#> source all-tls
S1:/config/logging/source-all-tls/#> type remote
S1:/config/logging/source-all-tls/#> transport tls
S1:/config/logging/source-all-tls/#> listen vlan1
S1:/config/logging/source-all-tls/#> certificate my-cert
S1:/config/logging/source-all-tls/#> ca-certificate my-ca-cert
S1:/config/logging/source-all-tls/#> peer-verify trusted
S1:/config/logging/source-all-tls/#> end
S1:/config/logging/#>
```

> **NOTE**
>
> We assume that the certificates **my-cert** and **my-ca-cert** have been previously created and are available on the device. Refer to the Certificate Management page for more information on how to manage certificates.

When setting `peer-verify trusted`, any client that connects to the switch with TLS must present a certificate that is signed by the CA certificate **my-ca-cert**.

Step 2: Configure the Logging Destination

We will configure a logging destination called **file-1**. This destination will be set to write the log messages to a file called **from-remotes.log**, which will be stored on the internal RAM disk:

```
S1:/config/logging/#> destination file-1
S1:/config/logging/destination-file-1/#> type file
S1:/config/logging/destination-file-1/#> media internal
S1:/config/logging/destination-file-1/#> file from-remotes.log
S1:/config/logging/destination-file-1/#> end
S1:/config/logging/#>
```

Step 3: Configure the Logging Sink

Now that we have configured the source and destination components, we can connect them by creating a sink, that we will call **remote-sink**:

```
S1:/config/logging/#> sink remote-sink
S1:/config/logging/sink-remote-sink/#> source all-udp all-tcp all-tls
S1:/config/logging/sink-remote-sink/#> destination file-1
S1:/config/logging/sink-remote-sink/#> end
S1:/config/logging/#>
```

The overview of the configuration should now look something like this:

```
S1:/config/logging/#> show
Console syslog   : Disabled
External logging : Disabled
Secure mode      : Enabled

 Sources
 NAME
 Type: local      USERSPACE  KERNEL
 local            Yes        Yes
 Type: remote     TRANSPORT  LISTEN                PORT   FORMAT
 all-udp          udp        vlan1 (192.168.1.1)   514    rfc5424
 all-tcp          tcp        vlan1 (192.168.1.1)   601    rfc5424
 all-tls          tls        vlan1 (192.168.1.1)   6514   rfc5424

 Destinations
 NAME
 Type: file       MEDIA     PATH
 file-1           internal  from-remotes.log

 Filters
 NAME             INVERT        TYPE   FILTER

 No filter(s) currently configured ...

 Sinks
 NAME             SOURCES       DESTINATIONS   OP FILTERS
 remote-sink      all-udp       file-1
                  all-tcp
                  all-tls
```

```
Configuration Warnings: 1
#   Description
1   Secure Mode is currently enabled for logging, but at least one remote
    source is configured. This means that this source, or sources, will
    not be able to receive log messages. If a remote source is intended to
    receive log messages, Secure Mode must be disabled.
```

As we can see from the output above, we have a configuration warning that is displayed:

```
Configuration Warnings: 1
#   Description
1   Secure Mode is currently enabled for logging, but at least one remote
    source is configured. This means that this source, or sources, will
    not be able to receive log messages. If a remote source is intended to
    receive log messages, Secure Mode must be disabled.
```

This warning is displayed because the secure mode is enabled for logging, but we have configured remote sources. This means that the remote sources will not be able to receive log messages. If we want the remote sources to receive log messages, we must disable the secure mode:

```
S1:/config/logging/#> no secure-mode
```

With this configuration all remote clients sending log messages to **S1**, should be logged to the file **from-remotes.log**, located in the internal RAM log folder. Leave the configuration context to apply the changes:

```
S1:/config/logging/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
S1:/#>
```

Verification

If everything is configured correctly, we should be able to see the log messages from the remote clients in the **from-remotes.log** file. We can verify this by checking the contents of the file:

```
S1:/#> show logfile from-remotes.log
[ ... ]
May 27 14:43:56 192.168.1.10 admin: test-1
May 27 14:43:57 192.168.1.20 admin: test-2
May 27 14:43:57 192.168.1.30 admin: test-3

from-remotes.log                                Mon May 27 14:43:57 2024

S1:/#>
```

In the example above, we have generated log messages from three different hosts **H1**, **H2**, and **H3**.

TIP

If we want the log messages to be written to the file with their hostnames instead of their IP addresses, we can configure the source to use the `keep-hostname` option. This will keep the hostname of the remote client in the log message:

```
S1:/#> configure
S1:/config/#> logging
S1:/config/logging/#> source all-udp
S1:/config/logging/source-all-udp/#> keep-hostname
S1:/config/logging/source-all-udp/#> end
S1:/config/logging/#>
S1:/config/logging/#> source all-tcp
S1:/config/logging/source-all-tcp/#> keep-hostname
S1:/config/logging/source-all-tcp/#> end
S1:/config/logging/#> source all-tls
S1:/config/logging/source-all-tls/#> keep-hostname
S1:/config/logging/source-all-tls/#> end
S1:/config/logging/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
S1:/#>
```

We can also take a look at the logging source status output:

```
S1:/#> show logging source
╒ Sources ═══════════════════════════════════════════════════════════
│TYPE        NAME             DESCRIPTION       PROCESSED
│remote      sink-all-udp     192.168.1.1       61
│remote      sink-all-tcp     tcp,192.168.1.20  65
│remote      sink-all-tls     tls,192.168.1.30  63
│unix-dgram  local-userspace  /dev/log          298
│file        local-kernel     /dev/kmsg         50
└─────────────────────────────────────────────────────────────────

S1:/#>
```

We should be able to see that the **PROCESSED** column for the **remote** sources is increasing, which means that the log messages are being received and processed by the switch.

For the remote sources using a connection-oriented transport protocol, TCP and TLS (Uses TCP as the underlying transport), we can also check the number of connections that are currently established:

```
S1:/#> show logging source connection
╒ Source Connections ═══════════════════════════════════
│SOURCE        LISTEN_SOCKET     NUM_CONNECTIONS
│sink-all-tcp  192.168.1.1:601   1
│sink-all-tls  192.168.1.1:6514  1
└────────────────────────────────────────────────────────

S1:/#>
```

Since UDP is a connectionless protocol, we do not have any connection information for the UDP source.

Troubleshooting

The following is an example of a number of things to check if the logging system is not working as expected:

- Ensure that **secure-mode** is disabled if you want to receive log messages from remote clients, `no secure-mode`.
- Ensure that the clients can simply reach the switch, using ping for example. If the clients cannot reach the switch, they will not be able to send log messages.
- If TLS is used, ensure that the certificates are correctly configured and that the clients are presenting a certificate that is signed by the CA certificate.
- If non-standard ports are used, ensure that the clients are sending log messages to the correct port, and that the switch is listening on the correct port.

Conclusion

In this document, we've explored how to set up and manage remote logging in our application. We've seen how to configure our system to receive logs from a remote source, and how we can connect these sources to a sink to write the logs to a file, in this case.

4.6.5. Transmitting Auditable Events to Remote Machines HowTo

About

This document provides a description how the system can be configured to transmit the auditable events that are generated by the system to a remote machine, using syslog.

> **SEE ALSO**
>
> Related Documentation
> - Auditable Events
> - Logging Configuration
> - Logging to Remote Machines

Introduction

The system produces a great number of auditable events, to provide detailed and traceable information about the system's operation. For more details about auditable events, see the Auditable Events.

This document describes how to configure the system to ensure that auditable events are transmitted to remote machines, such as a central logging server. By default, every time an auditable event is generated by the system, it is also logged as a syslog message. Therefore, to send the auditable events to a remote machine, simply configure the system to transmit the generated syslog messages to the remote machine. In other words, sending the auditable events to a remote machine is the same as sending any other syslog message to a remote machine.

This entire HowTo expects the base configuration to be based on the configuration and setup described in the Logging to Remote Machines HowTo. Because of this, we will not repeat the configuration steps that are already described in that document. What we will detail here is simply the minor addition that is required to ensure that the auditable events are transmitted to the remote machine. Therefore, it is recommended that you read the Logging to Remote Machines HowTo first and base your initial configuration on that document.

Configuration

Essentially the only thing that differs from the configuration described in the Logging to Remote Machines HowTo is that the specific local source type `audit` should be set for the logging source that is used to log the auditable events.

In the above mentioned HowTo the configuration of a source called **local** is described. The configuration of the **local** source can be adjusted to also include the `audit` log messages:

```
S1:/#> configure
S1:/config/#> logging
S1:/config/logging/#> source local
S1:/config/logging/source-local/#> audit
S1:/config/logging/source-local/#> end
S1:/config/logging/#>
```

Or a new a new source can be created, that only specifies the `audit` log messages, if those are the only log messages that should be transmitted to the remote machine. So in this case, lets simply name it **audit**:

```
S1:/#> configure
S1:/config/#> logging
S1:/config/logging/#> source audit
S1:/config/logging/source-audit/#> type local
S1:/config/logging/source-audit/#> no userspace
S1:/config/logging/source-audit/#> no kernel
S1:/config/logging/source-audit/#> audit
S1:/config/logging/source-audit/#> end
S1:/config/logging/#>
```

If a new source is created, ensure that this is also attached to the relevant `sink` configuration, mapping the source, destination and the optional filter together, as described in the other HowTo's Step 3: Sink Configuration. If the new source **audit** should be added to the existing sink **remote**:

```
S1:/#> configure
S1:/config/#> logging
S1:/config/logging/#> sink remote
S1:/config/logging/sink-remote/#> source audit
S1:/config/logging/sink-remote/#> end
S1:/config/logging/#>
```

Where the configuration should now look something like this:

```
S1:/config/logging/#> show
example:/config/logging/#> show
Console syslog   : Disabled
External logging : Disabled
Secure mode      : Enabled

╒ Sources ═══════════════════════════════════════════════════════════
NAME
Type: local      USERSPACE  KERNEL  AUDIT
local            Yes        Yes     No
audit            No         No      Yes
═══════════════════════════════════════════════════════════════════

╒ Destinations ══════════════════════════════════════════════════════
NAME
Type: remote     TRANSPORT  TO-HOST       PORT   FORMAT
remote           udp        192.168.1.99  514    rfc5424
═══════════════════════════════════════════════════════════════════

╒ Filters ═══════════════════════════════════════════════════════════
NAME             INVERT        TYPE  FILTER
═══════════════════════════════════════════════════════════════════
No filter(s) currently configured ...
═══════════════════════════════════════════════════════════════════

╒ Sinks ═════════════════════════════════════════════════════════════
NAME             SOURCES       DESTINATIONS   OP FILTERS
remote           local         remote
                 audit
═══════════════════════════════════════════════════════════════════

example:/config/logging/#>
```

> **BY DEFAULT FROM FACTORY CONFIGURATION**
>
> From the factory configuration, a specific logging source named `audit` should exist, that simply specifies a type local that only provides the auditable events log messages. This is in addition to the default logging source `local`, that simply takes all regular log messages from userspace and the kernel. The `audit` source is used to separate the auditable events from the regular log messages, so that they can be treated differently, if needed.

Verification

For verification refer to the Logging to Remote Machines HowTo. The exact same steps can be used to verify that the auditable events are transmitted to the remote machine.

Conclusion

This document described how to configure the system to transmit the auditable events that are generated by the system to a remote machine, using syslog, with the same basic logging setup as described in the Logging to Remote Machines HowTo. Essentially the only thing that differs is that the specific local source type `audit` should be set for the logging source that is used to log the auditable events.

4.7. Network Services

4.7.1. DHCP Server HowTo

Introduction

Imagine a network topology with a number of devices connected to a router on two different VLANs. To avoid configuring static addresses for all devices on the network, we configure the router to also act as a DHCP server. We also want to inform the devices that the router is the default gateway and DNS server.

In some cases, predetermined roles on the network of certain hosts are preferred. For example, a given port on a switch, or a client which sends a certain hostname should always get a static IP address. For such cases the DHCP server can be configured with *Static Host Entires* that are individually configured. These entries are configured to be provided to host that meets the matching requirement of the entry, this can be either a Client Id, MAC address or DHCP Option 82 (switch port).

```
                                        |
                                    .--.-.
                                  ( (    )__
                                  (_,  \ ) ,_)  Internet/Intranet
                                   '-'--`--'
                                     |
                                .---+----.
                        vlan1 |  DHCP  | vlan2
                        +-----+ Server +-----+
                        |   .1|        |.1   |
                        |     '--------'     |
192.168.1.0/24          |                    |      192.168.2.0/24
----+----------+----+---               ---+----+----------+----
    |          |                             |          |
  .-+--.     .-+--.                        .-+--.     .-+--.
  | H1 |     | H2 |                        | H3 |     | H4 |
  '----'     '----'                        '----'     '----'
```

Figure 1: Example network with two LANs

Configuration

DHCP Server

Description

The DHCP Server is with two different subnet definitions, each with their own address pool, in order to distribute addresses to both VLANs.

Interface Configuration

Options

Option	Description
`VLAN1`	Address: 192.168.1.1
`VLAN2`	Address: 192.168.2.1

DHCP Server Configuration

continues on next page

continued

DHCP Server

Options

Option	Description
`Subnet 1`	• Pool: 192.168.1.100/24 - 192.168.1.150/24 • lease time: 1 hour • Gateway: 192.168.1.1 • Name-server: 192.168.1.1
`Subnet 2`	• Pool: 192.168.2.100/24 - 192.168.2.150/24 • lease time: 1 hour • Gateway: 192.168.2.1 • Name-server: 192.168.2.1

Address Pools

Enter the DHCP Server configuration Context:

```
example:/#> configure
example:/config/#> dhcp-server
example:/config/dhcp-server/#>
```

Configure the dynamic pool for subnet 192.168.1.0:

```
example:/config/dhcp-server/#> subnet 192.168.1.0/24
example:/config/dhcp-server/subnet-192.168.1.0/#> pool 192.168.1.100 192.168.1.149
example:/config/dhcp-server/subnet-192.168.1.0/#> lease-time 3600
example:/config/dhcp-server/subnet-192.168.1.0/#> gateway 192.168.1.1
example:/config/dhcp-server/subnet-192.168.1.0/#> name-server 192.168.1.1
example:/config/dhcp-server/subnet-192.168.1.0/#> end
example:/config/dhcp-server/#>
```

Configure the dynamic pool for subnet 192.168.2.0:

```
example:/config/dhcp-server/#> subnet 192.168.2.0/24
example:/config/dhcp-server/subnet-192.168.2.0/#> pool 192.168.2.100 192.168.2.149
example:/config/dhcp-server/subnet-192.168.2.0/#> lease-time 3600
example:/config/dhcp-server/subnet-192.168.2.0/#> gateway 192.168.2.1
example:/config/dhcp-server/subnet-192.168.2.0/#> name-server 192.168.2.1
example:/config/dhcp-server/subnet-192.168.2.0/#> end
example:/config/dhcp-server/#> leave
example:/#>
```

When we enter the configuration for the specific a few different variables are configured:

- First the `pool` of available addresses are configured to serve up to 50 hosts on that subnet (.100 - .149).
- The `lease-time` is configured so that each address are supplied with a validity of one hour (3600 seconds). When that time has elapsed any host that wants to retain its given address will need to consult the server again to renew it.
- Both a `gateway` and `name-server` are defined to be accessible on the DHCP server it self, since in this case it also serves as the router for the two local networks.

The server is now configured to serve two different subnets, all these addresses will be handled dynamically so no additional actions should be needed.

Static Leases

In this example we want to provide a specific configuration for **H3** based on its MAC address. We also want to provide a Specific address on the subnet 192.168.1.0/24 for any host with a specific Client Id.

Enter the DHCP Server configuration Context:

```
example:/#> configure
example:/config/#> dhcp-server
example:/config/dhcp-server/#>
```

Configure host 1:

```
example:/config/dhcp-server/#> host 1
example:/config/dhcp-server/host-1/#> match client string some_clientid_string
example:/config/dhcp-server/host-1/#> address 192.168.1.10
example:/config/dhcp-server/host-1/#> lease-time 10000
example:/config/dhcp-server/host-1/#> gateway 192.168.1.1
example:/config/dhcp-server/host-1/#> end
```

The `host 1` static lease is configured to `match` on a `clientid string`. If a host provides the matching Client Id it will be provided with the specific `address` 192.168.1.10, instead of an address from the dynamic address pool.

Configure host 2:

```
example:/config/dhcp-server/#> host 2
example:/config/dhcp-server/host-2/#> match mac 0c:50:e5:4f:d6:00
example:/config/dhcp-server/host-2/#> address 192.168.2.20
example:/config/dhcp-server/host-2/#> lease-time 10000
example:/config/dhcp-server/host-2/#> gateway 192.168.2.1
example:/config/dhcp-server/host-2/#> end
example:/config/dhcp-server/#> leave
example:/#>
```

The `host 2` static lease is configured to `match` on a specifc `mac` address. Any host associated with the matching MAC will be provided the specific `address` 192.168.2.20, and not another address from the dynamic pool of addresses.

Status

In order to verify if any DHCP leases are handed out to hosts on the subnet:

```
example:/#> show dhcp-server
LEASE TIME  MAC ADDRESS       CLIENT ID            IP ADDRESS       HOSTNAME
3600        0c:50:e5:8c:52:00 01:0c:50:e5:8c:52:00 192.168.1.136    H1
10000       0c:50:e5:7c:45:00 01:0c:50:e5:7c:45:00 192.168.1.10     H2
10000       0c:50:e5:4f:d6:00 01:0c:50:e5:4f:d6:00 192.168.2.20     H3
3600        0c:50:e5:33:a6:00 01:0c:50:e5:33:a6:00 192.168.2.121    H4
```

This shows the specific IP Addresses that have been handed out, and to which host (MAC Address) they have been provided for. This list should show all addresses that the server have provided, therefore this command is useful when debugging address distribution for your network.

4.7.2. DHCP Client Relay Server HowTo

Introduction

This document shows how to set up a DHCP server to provide leases to an IP subnet behind a DHCP relay agent.

The DHCP server is be configured with a default pool of addresses and a number of *Static Leases*. The DHCP relay is configured to append DHCP option 82 to each DHCP message on `vlan1` and forward it to the DHCP sever.

```
                                  |
                              .--.-.
                             ( (    )__
                            (_,  \ ) ,_)  Internet/Intranet
                              '-'--`--'
                                 |
                            .----+----.
                            |         |
                            | Server  |
                            |         |
                            '----+----'
                                 | .1
                     192.168.2.0/24 |
                                 | .2
                            .----+----.
                            |         | iface: vlan2
                            |  Relay  |
                            |         | iface: vlan1
                            '----+----'
                                 | .1
                     192.168.1.0/24 |
                                 |
            -----+-------+---+---+-------+-----
                 |       |       |       |
                .--+--. .--+--. .--+--. .--+--.
                |  H1 | |  H2 | |  H3 | |  H4 |
                '-----' '-----' '-----' '-----'
```

Figure 1: DHCP server and relay setup with multiple clients.

Relay Configuration

Configure the Relay to forward any DHCP traffic towards the server.

- Forward all DHCP messages to the `server` at 192.168.1.1 port 6767
- Append `option82`, to all DHCP messages that pass through the relay

```
Relay:/#> configure
Relay:/config/#> dhcp-relay
Relay:/config/dhcp-relay/#> server 192.168.1.1:6767
Relay:/config/dhcp-relay/#> iface vlan1
Relay:/config/dhcp-relay/iface-vlan1/#> end
Relay:/config/dhcp-relay/#>
Relay:/config/dhcp-relay/#> option82
Activating Option 82.
Relay:/config/dhcp-relay/option82/#> end
Relay:/config/dhcp-relay/#> leave
Relay:/#>
```

Server Configuration

The DHCP server is set up to provide a dynamic pool of addresses and a number of static leases for the **192.168.1.0** subnet. With each lease the server will also provide some additional information like: gateway and DNS server. Worth noting is the use of the non-standard port 6767, this means

clients cannot accidentally connect to the server for a pool address, but must always go through a relay agent.

```
Server:/config/#> dhcp-server
Server:/config/dhcp-server/#> server-port 6767
Server:/config/dhcp-server/#> name-server auto
Server:/config/dhcp-server/#> gateway auto
```

Next up is the subnet to serve:

```
Server:/config/dhcp-server/#> subnet 192.168.1.0/24
Created new DHCP server for subnet 192.168.1.0.
Server:/config/dhcp-server/subnet-192.168.1.0/#> pool 192.168.1.100 100
Server:/config/dhcp-server/subnet-192.168.1.0/#> end
Server:/config/dhcp-server/#>
```

We conclude the configuration by setting up a couple of *static leases*:

- The first lease use `match mac`, meaning only the client with that particular MAC address will be offered the `host 1` configuration, `address` 192.168.1.10
- The second lease use `match option82`, meaning any host connected to port *eth1* on the **Relay** (identified by its MAC address) will be offered the `host 2` configuration, `address` 192.168.1.20

```
Server:/config/dhcp-server/#> host 1
Server:/config/dhcp-server/host-1/#> match mac 0c:50:e5:07:24:00
Server:/config/dhcp-server/host-1/#> address 192.168.1.10
Server:/config/dhcp-server/host-1/#> end
Server:/config/dhcp-server/#> host 2
Server:/config/dhcp-server/host-2/#> match option82 circuit-id eth1
Server:/config/dhcp-server/host-2/#> match option82 remoteid-id 0c:50:e5:65:97:00
Server:/config/dhcp-server/host-2/#> address 192.168.1.20
Server:/config/dhcp-server/host-2/#> end
Server:/config/dhcp-server/#> leave
Server:/#>
```

Status

To verify that addresses are distributed as intended any provided lease can be viewed on the server:

```
Server:/#> show dhcp
LEASE TIME  MAC ADDRESS       CLIENT ID            IP ADDRESS       HOSTNAME
864000      0c:50:e5:07:24:00 01:0c:50:e5:07:24:00 192.168.1.10     H1
864000      0c:50:e5:a0:b3:00 01:0c:50:e5:a0:b3:00 192.168.1.20     H2
864000      0c:50:e5:b2:55:00 01:0c:50:e5:b2:55:00 192.168.1.114    H3
864000      0c:50:e5:c6:27:00 01:0c:50:e5:c6:27:00 192.168.1.120    H4
```

4.7.3. DHCP Relay/Server in Ring HowTo

Introduction

This document shows how to set up DHCP relay agents and a DHCP server setup in an FRNT ring topology. It is assumed that the ring topology has already been configured on the involved devices. If you need more information about each sub-system and its configurations options, see:

- DHCP Server
- DHCP Relay Agent
- FRNT (Fast Reconfiguration of Network Topology)

The use-case presented here shows how DHCP functionality can be combined with FRNT in order to create a redundant network setup with dynamic address allocation.

```
                                192.168.1.0/24
        .-----------------------------------------------------------.
        |                              FRNT                         |
        | .10                  .20                  .30              | .41
    .---+----.           .--------.           .--------.           .---+----.
    | DHCP   |           | DHCP   |           | DHCP   |           | DHCP   |
    | Relay1 +-----------+ Relay2 +-----------+ Relay3 +----||----+ Server |
    |        |           |        |           |        |           | Relay4 |
    '---+----'           '---+----'           '---+----'           '--------'
        |                    |                    |
      .-+--.               .-+--.               .-+--.
      | H1 |               | H2 |               | H3 |
      '----'               '----'               '----'
```

Figure 1: FRNT ring with three DHCP relay agents and one DHCP server/relay

The major benefit of this setup is the centralized handling of the DHCP configuration. All hosts are connected to a DHCP relay agent on the FRNT ring, and all DHCP traffic is relayed to the DHCP server, which is also configured to act as a relay on the ring. The relay on the server switch is configured to forward all DHCP traffic it receives to itself, since it acts as the server.

The setup has a few interesting advantages:

- It reduces the amount of broadcast messages on the ring. All hosts requesting a DHCP address go through one of the relays. Any DHCP message that would otherwise have been broadcast on the ring, is now sent as unicast from each relay to the server
- A DHCP relay on each switch with an access port enables setting up a static IP address per port using DHCP option 82. With this extra data from each relay agent, the DHCP server can identify each access port (using circuit-id and remote-id)
- Zero second fail-over in case of link break, thanks to FRNT. Worth noting is that the DHCP server has also been configured as the focal point in the FRNT ring, this is not mandatory but may be useful in some use-cases

Relay Agent Configuration

All DHCP relay agents have the same configuration:

- Forward all DHCP traffic to the `server` located at 192.168.1.41
- Listen to all DHCP traffic `iface` vlan1, except …
- Traffic on `port` eth3 and eth6 is ignored by the relay agent because these are the FRNT ring ports and no clients are ever directly connected here
- Enable `option82` to append DHCP option 82 on all packets forwarded to the server, consisting of additional information about the relay agent and port (circuit-id, remote-id)

```
RelayX:/#> configure
RelayX:/config/#> dhcp-relay
RelayX:/config/dhcp-relay/#> server 192.168.1.41
RelayX:/config/dhcp-relay/#> iface vlan1
RelayX:/config/dhcp-relay/iface-vlan1/#> port eth3,eth6
RelayX:/config/dhcp-relay/iface-vlan1/port-eth3,eth6/#> no enable
RelayX:/config/dhcp-relay/iface-vlan1/port-eth3,eth6/#> end
RelayX:/config/dhcp-relay/iface-vlan1/#> end
RelayX:/config/dhcp-relay/#> option82
Activating Option 82.
RelayX:/config/dhcp-relay/option82/#> end
RelayX:/config/dhcp-relay/#> leave
RelayX:/#>
```

Server Configuration

The DHCP server switch also acts as a relay to itself. This to handle directly connected hosts and append option 82 to their requests. It also allows the server to only accept DHCP messages from hosts connected to relays on the ring.

First, we configure the local DHCP relay on the server:

- Forward any DHCP messages to the `server` located on the local device (127.0.0.1)
- Since the device acts as both a DHCP Relay and DHCP Server, the server needs to be located at another port. The relay agent looks for DHCP messages on the default UDP port (67), so the server needs to be located on another. Hence, the `server` port is specified to 6767
- Enable `option82` on the relay so directly connected hosts are also forwarded with DHCP option 82 information to the server

```
Server:/#> configure
Server:/config/#> dhcp-relay
Server:/config/dhcp-relay/#> server 127.0.0.1:6767
Server:/config/dhcp-relay/#> iface vlan1
Server:/config/dhcp-relay/iface-vlan1/#> end
Server:/config/dhcp-relay/#> option82
Activating Option 82.
Server:/config/dhcp-relay/option82/#> end
Server:/config/dhcp-relay/#> end
```

Next, we need to configure the DHCP server itself:

- The DHCP `server-port` is set to 6767, as discussed previously
- A `subnet` is defined with a default `pool` in the range 192.168.1.100 - 192.168.1.199. This will be used by any client that does not have a matching static lease (below)

```
Server:/config/#> dhcp-server
Server:/config/dhcp-server/#> server-port 6767
Server:/config/dhcp-server/#> subnet 192.168.1.0/24
Created new DHCP server for subnet 192.168.1.0.
Server:/config/dhcp-server/subnet-192.168.1.0/#> pool 192.168.1.100 100
Server:/config/dhcp-server/subnet-192.168.1.0/#> end
Server:/config/dhcp-server/#> leave
```

Each DHCP message that reach the server has a DHCP option 82 tag from each DHCP relay. This makes it possible to set up *Static Lease* configurations for specific ports on any of the switches in the LAN.

In this example we want the port `eth1` on `Relay2` to receive a specific configuration from the server:

- Create a new `host` configuration with an IP address of 192.168.1.80.
- Set up `match` rules:
  - `circuit-id string` match for *eth1*, and
  - `remote-id hex` match for `0c:50:e5:65:97:00`, which is the MAC address of `Relay2`

```
Server:/#> configure
Server:/config/#> dhcp-server
Server:/config/#> host 1
Server:/config/dhcp-server/host-1/#> address 192.168.1.80
Server:/config/dhcp-server/host-1/#> match circuit-id string eth1
Server:/config/dhcp-server/host-1/#> match remote-id hex 0c:50:e5:65:97:00
Server:/config/dhcp-server/host-1/#> leave
```

Server Status

In order to verify that addresses are distributed as intended any provided lease can be viewed on the server:

```
Server:/#> show dhcp
LEASE_TIME  MAC_ADDRESS       CLIENT_ID            IP_ADDRESS       HOSTNAME
864000      0c:50:e5:07:24:00 01:0c:50:e5:07:24:00 192.168.1.114    H1
864000      0c:50:e5:a0:b3:00 01:0c:50:e5:a0:b3:00 192.168.1.10     H2
864000      0c:50:e5:b2:55:00 01:0c:50:e5:b2:55:00 192.168.1.115    H3
```

4.7.4. DHCP IP Per-Port HowTo

Introduction

This document shows how to set up the DHCP server to hand out a static IP address (lease) to clients connected to a specific physical port.

```
.---------------------.
|        Switch       | DHCP Server/Relay
|                     | iface: vlan1
| eth1           eth2 | 192.168.1.0/24
'--+---------------+--'
   |.132           |.55
   |               |
.--+--.         .--+--.
|  H1 |         |  H2 |
'-----'         '-----'
```

Figure 1: DHCP server setup with IP-per-port for port eth2.

In the case depicted above, clients connected to port `eth1` receive an address from the server's pool of addresses. Clients on port `eth2` on the other hand always receive a specific address, 192.168.1.55. Hence, the most obvious use-case for the latter is to have a single end-device connected to `eth2`.

For more information about any of the sub-systems used and their configurations options, see:

- DHCP Server
- DHCP Relay Agent

Configuration

Configuration of both DHCP server and DHCP relay services. First, we set up the relay to forward all DHCP traffic to the local DHCP server. The key setting is `option82`, which makes sure the relay agent appends DHCP *Option 82* port information for the server:

```
example:/#> configure
example:/config/#> dhcp-relay
example:/config/dhcp-relay/#> server 127.0.0.1
example:/config/dhcp-relay/#> iface vlan1
example:/config/dhcp-relay/iface-vlan1/#> port eth2
example:/config/dhcp-relay/iface-vlan1/port-eth2/#> circuitid-type portname
example:/config/dhcp-relay/iface-vlan1/port-eth2/#> end
example:/config/dhcp-relay/iface-vlan1/#> end
example:/config/dhcp-relay/#> option82
Activating Option 82.
example:/config/dhcp-relay/option82/#> remoteid-type mac
example:/config/dhcp-relay/option82/#> end
example:/config/dhcp-relay/#> end
example:/config/#>
```

We can now move to set up the DHCP server, starting with the default subnet and address pool.

```
example:/config/#> dhcp-server
example:/config/dhcp-server/#> subnet 192.168.1.0/24
Created new DHCP server for subnet 192.168.1.0.
example:/config/dhcp-server/subnet-192.168.1.0/#> pool 192.168.1.100 100
example:/config/dhcp-server/subnet-192.168.1.0/#> end
example:/config/dhcp-server/#>
```

Next, we set up the *static lease* for host(s) connected to port `eth2`. The `match option82` configuration is what the server use to identify the correct port (circuit-id) and its own MAC address for iface `vlan1` (remote-id).

```
example:/config/dhcp-server/#> host
example:/config/dhcp-server/host-1/#> match option82 circuit-id string eth2
example:/config/dhcp-server/host-1/#> match option82 remoteid-id string 0c:50:e5:65:97:00
example:/config/dhcp-server/host-1/#> address 192.168.1.55
example:/config/dhcp-server/host-1/#> end
example:/config/dhcp-server/#> leave
example:/#>
```

Status

To verify that addresses are distributed as intended any provided lease can be viewed on the server:

```
example:/#> show dhcp-clients
LEASE TIME  MAC ADDRESS       CLIENT ID            IP ADDRESS       HOSTNAME
864000      0c:50:e5:07:24:00 01:0c:50:e5:07:24:00 192.168.1.132    H1
864000      0c:50:e5:65:97:00 01:0c:50:e5:65:97:00 192.168.1.55     H2
```

4.7.5. PTP HowTo

This document, being a non-exhaustive guide, outlines some practical examples of how to configure and use Precision Time Protocol.

For an overview of what PTP is and how it may be configured, refer to:

- PTP Configuration

Introduction

PTP may be used in various networks of different structures and complexity levels. One of the reasons behind flexibility and scalability of this protocol lies in the concept of Transparent Clock (TC). While Ordinary Clock (OC) has more or less static configuration and function, acting as either Master (providing time) or Slave (receiving time), TC acts as a kind of port expander, as in Figure 1 below, allowing numerous Slave clocks to get synchronised to single time source, the so called Grand Master clock (GM). There may be many interconnected TCs in a network, each serving as a link to time source, and also, in the case of ring structure, providing failover function. The workflow of each TC may dynamically change over time, adapting to various cicrumstances and states occuring in the network.

```
                                        .-------.
                                        |       |
                           +----------+s OC   |
                           |          |       |
                           |          '---+---'
.-------.       .---+---.      .-------.
|       |       |   m   |      |       |
|  GM  m+-------+s TC  m+------+s OC   |
|       |       |   m   |      |       |
'-------'       '---+---'      '---+---'
                           |          .-------.
                           |          |       |
                           +----------+s OC   |
                                        |       |
                                        '-------'
```

Figure 1: Transparent clock as port expander. Letters ‘m’ and ‘s’ represent MASTER and SLAVE ports respectively.

General considerations

PTP could be used in two different ways: within a vlan, or outside any vlan.

If PTP is used in a vlan, make sure to add PTP ports to desired vlan. Suppose ports eth3 and eth4 are standalone interfaces, and PTP is supposed to run in vlan 100:

```
oc-gm:/#> conf vlan 100
oc-gm:/config/vlan-100/#> untagged eth3,eth4
oc-gm:/config/vlan-100/#> leave
oc-gm:/#>
```

If PTP is used outside any vlan, make sure to remove PTP ports from vlans and have them as standalone interfaces in WeOS. Suppose ports eth3 and eth4 are part of vlan 1:

```
oc-gm:/#> conf vlan 1
oc-gm:/config/vlan-1/#> no untagged eth3,eth4
oc-gm:/config/vlan-1/#> leave
oc-gm:/#>
```

Regardless of chosen configuration, consistency should be kept in all ptp devices.

PTP configuration for a network represented in Figure 1

To ensure correct operation of PTP, all devices should use the same profile. Additionally, it is a good practice to ensure protocol compatibility of all PTP devices in the network. It is known that some older equipment does not support the newer 2.1 protocol version. WeOS supports both 2.0 (default) and 2.1 protocol versions and it is possible to give preference to one or the other in CLI/Web.

Let us configure GM. This is done by configuring our device as an OC and giving it high priority, by setting *priority1* attribute to low value. In our little network, clock with highest priority will be automatically selected by BMCA as GM. Note that only one port may be selected for an OC.

```
oc-gm:/#> conf ptp
oc-gm:/config/ptp/#> enable
oc-gm:/config/ptp/#> description name "GM" location "howto"
oc-gm:/config/ptp/#> clock ordinary
oc-gm:/config/ptp/clock-ordinary/#> ports eth4
oc-gm:/config/ptp/clock-ordinary/#> priority1 100
oc-gm:/config/ptp/clock-ordinary/#> end
oc-gm:/config/ptp/#> profile power-v2
oc-gm:/config/ptp/profile-power-v2/#> leave
oc-gm:/#>
```

Our network is composed of 5 devices, out of which one device is our TC. Let us add 4 ports by which our TC is connected to other devices in the network. Note that setting of description attribute is not mandatory but will aid us in debugging and monitoring PTP network later on.

```
tc:/config/#> ptp
tc:/config/ptp#> enable
tc:/config/ptp#> description name "TC" location "howto"
tc:/config/ptp#> profile power-v2
tc:/config/ptp/profile-power-v2/#> end
tc:/config/ptp#> clock transparent
tc:/config/ptp/clock-transparent/#> ports eth5 eth9..eth11
tc:/config/ptp/clock-transparent/#> end
tc:/config/ptp#> leave
```

Slave devices are configured as OCs, with their *priority1* attribute left untouched. Note that all OC Slave devices will have identical configuration, with the sole exception of port attribute, which may differ from device to device.

```
oc-slave:/config/#> ptp
oc-slave:/config/ptp#> enable
oc-slave:/config/ptp#> description name "slave1" location "howto"
oc-slave:/config/ptp/#> clock ordinary
oc-slave:/config/ptp/clock-ordinary/#> ports eth4
oc-slave:/config/ptp/clock-ordinary/#> end
oc-slave:/config/ptp#> profile power-v2
oc-slave:/config/ptp/profile-power-v2/#>#> leave
```

NOTE: Only TC has official support in WeOS. OC may be used only for testing purposes.

Monitoring Status

Now that we have configured our PTP network, we want to know whether it is functioning as expected. We expect to see three slave devices, each synchronised to the GM.

Monitoring of PTP network is possible from any device in the network, by by issuing simple queries from adminexec. Let us issue the following queries from our TC:

`show ptp` command, see Listing 1, shows current configuration of our device as well as PTP states for each port. We may observe that all ports of our TC are operational, i.e, in MASTER or UNCALIBRATED state and peer delay for each link has been calculated.

`show ptp network` command may be used to monitor our PTP network, see Listing 2. We observe that in our case slave clocks 00077c.fffe.784fda, 00077c.fffe.78530b and 00077c.fffe.7bf4c5 are closely synchronised to the grandmaster clock as they are within the offset of 7.0, 2.0 and 0.0 nanoseconds respectively.

```
tc:/#> show ptp
Clock
ID                  : 00077c.fffe.785325
Type                : TC P2P
Domain              : 0
Mean Path delay     : 0.0
Master Offset       : -669033273898.0
Priority1           : 254
Priority2           : 128
Product             : Westermo;RedFox-5728-E-F16G-T12G-HVHV;00:07:7c:78:53:20
Distance            : 1
Description         : TC

Ports
ID                      INTERFACE        STATE          PEER DELAY
00077c.fffe.785325-1    eth9             MASTER         88
00077c.fffe.785325-2    eth10            UNCALIBRATED   94
00077c.fffe.785325-3    eth11            MASTER         11
```

Listing 1: Transparent clock status.

```
tc:/#> show ptp network

Clocks
ID                      TYPE    DOMAIN   MASTER OFFSET   DESCRIPTION
00077c.fffe.785325      TC P2P  0        -669033063522.0 TC
00077c.fffe.784fda      OC      0        7.0              slave1
00077c.fffe.78530b      OC      0        2.0              slave2
00077c.fffe.7bf4c5      OC      0        0.0              slave3
```

Listing 2: PTP network status. Collected from Transparent clock.

4.8. PKI

4.8.1. PKI - Enroll Certificates HowTo

About

This document aims to show how to enroll two certificates using the SCEP protocol: one from a server running OpenXPKI[1] as a SCEP server and certificate authority, and one from a server running Windows NDES. The configuration of OpenXPKI and Windows NDES servers is not part of this document.

For an overview of all SCEP configuration settings, see: SCEP

Prerequisites

- When dealing with certificates, it is important that the system clock on the device is correctly set, preferably using an NTP server.
- In these examples, it is assumed that a DNS server has been configured.

Enrollment Servers

This section covers the configuration of enrollment servers.

OpenXPKI server

To enroll a certificate, we first configure an enrollment server in the PKI config context. Let us start with the OpenXPKI server.

```
example:/#> configure
example:/config/#> pki
example:/config/pki/#> enroll 1
example:/config/pki/enroll-server-1#>
```

We give the server a meaningful `label` that is unique among local CA certificates and enrollment servers. The Root-CA certificate fetched from the server will be given that label. If there are any existing CA certificates with the same label, we recommend removing them first before proceeding with the configuration. Note: The label cannot be changed once the configuration is applied.

```
example:/config/pki/enroll-server-1#> label xpki-srv
```

Let us write a short `description` that fits and specify the complete `url` to the server, including scheme and port if applicable. Note that the SCEP protocol typically uses HTTP to communicate with the server.

```
example:/config/pki/enroll-server-1#> label xpki-srv
example:/config/pki/enroll-server-1#> description This is the OpenXPKI server
example:/config/pki/enroll-server-1#> url http://myurl.com:8080/scep/generic
```

The fingerprint is a hash of the Root-CA certificate in DER format. It is used for automatic integrity and authenticity checks before enrollment.

The fingerprint can be calculated as follows:

```
$ openssl x509 -outform der -in cert-ca.pem | sha256sum
ddaf9cb51e5bd1e882d0a7a3c1b3bc6cc1cb129071b06a3e6b27c06e503b04b3  -
```

[1] https://www.openxpki.org/

```
example:/config/pki/enroll-server-1#> fingerprint ddaf9cb51e5bd1e882d0a7a3c1b3bc6cc1cb129071b06a3e6b27c06e503b04b3
```

> **NOTE**
>
> Ensure that the `fingerprint` and `fingerprint-alg` match the values calculated for the server's Root-CA certificate. Using the wrong fingerprint will result in failure.

Be sure to use the Root-CA certificate to get the correct fingerprint.

Optionally, set `ra-dn` to the subject-hash of an RA certificate used for signing and encryption in SCEP communication. The certificate must have Key Usage values `digitalSignature` and `keyEncipherment`. Leave it unset to use the default Auto selection. Set `ra-dn` when a CA certificate is used as the RA and no dedicated RA certificate is provided. The selected certificate's subject hash must remain unchanged across CA rollover events.

Lastly, we set the `renew-request` to `PKCSReq`. The RFC 8894 standard introduced a dedicated RenewalReq message type. However, some SCEP servers (e.g., OpenXPKI) do not support it and perform renewal using PKCSReq instead of RenewalReq. Windows NDES handles it correctly.

```
example:/config/pki/enroll-server-1#> cipher aes128-cbc
example:/config/pki/enroll-server-1#> ra-dn 3f49c9d2
example:/config/pki/enroll-server-1#> renew-request pkcs-req
```

A complete configuration of the OpenXPKI enrollment server might look like this:

```
example:/config/pki/enroll-server-1#> show
Server #                      : 1
Label                         : xpki-srv
Description                   : This is the OpenXPKI server
Status                        : Enabled
URL                           : http://myurl.com:8080/scep/generic
RA-DN-Hash                    : 3f49c9d2
fingerprint                   : 0542DFC5F97A4812E18FBA0CAE092E82 ...
fingerprint-alg               : SHA256
Protocol                      : SCEP
Cipher                        : AES128-cbc
Renew-request                 : PKCSReq
example:/config/pki/enroll-server-1#>
```

Now we can apply the enrollment server configuration context with:

```
example:/config/pki/enroll-server-1#> leave
```

Windows NDES server

In a similar fashion, we configure a Windows NDES server:

```
example:/#> configure
example:/config/#> pki
example:/config/pki/#> enroll 2
example:/config/pki/enroll-server-2#> label win-srv
example:/config/pki/enroll-server-2#> description This is the Windows NDES server
example:/config/pki/enroll-server-2#> url http://myurl.com/certsrv/mscep/mscep.dll
```

```
example:/config/pki/enroll-server-2#> fingerprint 7BDFAB9BEDB200DE53F34DA2F2A9E8F9
example:/config/pki/enroll-server-2#> fingerprint-alg MD5
example:/config/pki/enroll-server-2#> cipher 3des-cbc
example:/config/pki/enroll-server-2#> ra-dn 7cbe3a1f
example:/config/pki/enroll-server-2#> show
Server #                        : 2
Label                           : win-srv
Description                     : This is the Windows NDES server
Status                          : Enabled
URL                             : http://myurl.com/certsrv/mscep/mscep.dll
RA-DN-Hash                      : 7cbe3a1f
fingerprint                     : 7BDFAB9BEDB200DE53F34DA2F2A9E8F9
fingerprint-alg                 : MD5
Protocol                        : SCEP
Cipher                          : 3DES-CBC
Renew-request                   : RenewalReq
example:/config/pki/enroll-server-2#>
```

Now we can step out of the enrollment server configuration context with:

```
example:/config/pki/enroll-server-2#> leave
```

Verification

Let's verify if our configuration was successful in the PKI exec context.

```
example:/pki/#> show

PKI
Daemon
Status                          : Running
Uptime                          : 1 min 12 sec

Enroll Servers
ID  LABEL      STATUS   TIME                 Message
1   xpki-srv   Enabled  2025-06-02 14:01:09  responding
2   win-srv    Enabled  2025-06-02 14:01:09  responding

[ ... ]
```

The PKI daemon responsible for enrollment is up and running and the two SCEP endpoints are responding. For other possible statuses see the troubleshooting section.

We see that CA/RA certificates were successfully downloaded. Notice Root-CA certificates `xpki-srv` and `win-srv` that can be referred to from other parts of the system.

```
example:/pki/cert/#> show full

TYPE     HASH     EXPIRES     NAME                 DIST LABEL
CA-local 684a8b4c Apr 29 2026 OpenXPKI:scep-ra       SCEP xpki-srv+ra
CA-local 8dff03d8 Apr 28 2026 OpenXPKI Issuing     SCEP xpki-srv-1.792a8246
CA-local 5be49178 Apr 28 2035 OpenXPKI Root~250425 SCEP xpki-srv
CA-local 12ec9d6d Apr 04 2026 WSE-S0307-MSCEP-RA   SCEP win-srv+ra-2
CA-local 12ec9d6d Apr 04 2026 WSE-S0307-MSCEP-RA   SCEP win-srv+ra
CA-local 627f7b0a Apr 04 2049 rndtestad-WSE~307-CA SCEP win-srv
Pub      52ff4f77 Jan 19 2038 zero-ee-00-00.local  web-default
Key      N/A      N/A         web-default          N/A  web-default
```

Certificate Enrollment Requests

Enrolling from OpenXPKI server

Now it is time to enroll our first certificate from the OpenXPKI server. Let us enter the enrollment context. We are greeted with errors and warnings indicating the need to fill empty fields of the request.

```
example:/#> pki
example:/pki/#> cert
example:/pki/cert/#> enroll mycert1
╒ Configuration Errors: 3 ═══════════════════════════════════════════════
│#   Description
│1   Server label missing.
├──────────────────────────────────────────────────────────────────────
│2   Server label not defined in the system.
├──────────────────────────────────────────────────────────────────────
│3   Certificate Distinguished Name missing.
└──────────────────────────────────────────────────────────────────────

╒ Configuration Warnings: 2 ═══════════════════════════════════════════
│#   Description
│1   The challenge password is missing. It is strongly recommended to use a
│    challenge password while enrolling a certificate.
├──────────────────────────────────────────────────────────────────────
│2   Certificate SubjectAlternativeName missing.
└──────────────────────────────────────────────────────────────────────
```

```
example:/pki/cert/enroll-mycert1/#> show
Server                        :
Password                      :
DN                            :
Renewal-threshold             : 80 %
SAN                           :
Keysize                       : 2048
Profile                       :
```

A Certificate Enrollment Request combines in one context parameters related to the enrollment process and parameters of the end-entity certificate we want to enroll.

First, we start by choosing the `server` to enroll from and the challenge `password` provided by the server administrator. We keep `renewal-threshold` to its default value of 80%.

```
example:/pki/cert/enroll-mycert1/#> server xpki-srv
example:/pki/cert/enroll-mycert1/#> password SecretChallange
```

> **SECURITY**
>
> It is RECOMMENDED that the challengePassword be a one-time authenticator value to limit the ability of an attacker who can capture the authenticator from the client or CA and reuse it to request further certificates. Read more here: RFC8894[1].

Second, we fill parameters of the certificate itself:

[1] https://www.rfc-editor.org/rfc/rfc8894.html#name-challengepassword-shared-se

```
example:/pki/cert/enroll-mycert1/#> dn CN=mycert1.co.uk, C=UK
example:/pki/cert/enroll-mycert1/#> san 192.168.100.101
example:/pki/cert/enroll-mycert1/#> profile tls-server
```

The `profile` and `san` are not mandatory but are recommended parameters. The profile depends on the server configuration and should be provided by the server administrator.

A complete configuration presents like this:

```
example:/pki/cert/enroll-mycert1/#> show
Cert label                      : mycert1
Server                          : xpki-srv
Password                        : SecretChallange
DN                              : CN=mycert1.co.uk, C=UK
Renewal-threshold               : 80 %
SAN                             : 192.168.100.101
Keysize                         : 2048
Profile                         : tls-server
example:/pki/cert/enroll-mycert1/#>
```

Now we can apply the configuration:

```
example:/pki/cert/enroll-mycert1/#> end
```

From now on, the device will send enrollment requests to the server until enrollment or failure. Regardless of whether requests reached the server or not, the certificate is considered PENDING. To show the status of the certificate:

```
example:/pki/cert/#> show
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
Pub               PENDING     N/A                  SCEP mycert1
Pub      4e3ca0d2 Jan 19 2038 zero-ee-00-00.local  web-default
```

While the certificate is in state PENDING, we can see its parameters with:

```
example:/#> pki cert show mycert1
Certificate request configuration for 'mycert1':
  Server label: xpki-srv
  Password:
  DN: CN=mycert1.co.uk, C=UK
  Renewal-threshold: 80
  SAN: 192.168.100.101
  Keysize: 2048
  Profile: tls-server
```

At this point, it is up to the CA administrator to either approve or reject our request. After approval, a certificate will be automatically downloaded to the device at the next certificate check done by the device (approximately within 2 minutes). The check can be manually triggered:

```
example:/#> pki trigger
Triggering PKI daemon.
```

After approval, the status of the certificate will show:

```
example:/#> pki cert show
TYPE     HASH     EXPIRES     NAME                DIST LABEL
Pub      5bddad69 Feb 17 2025 mycert1.co.uk       SCEP mycert1
Pub      4e3ca0d2 Jan 19 2038 zero-ee-00-00.local web-default
```

When rejected, the status would look like this:

```
example:/#> pki cert show
TYPE     HASH     EXPIRES     NAME                DIST LABEL
Pub               REJECTED    N/A                 SCEP mycert1
Pub      4e3ca0d2 Jan 19 2038 zero-ee-00-00.local web-default
```

There are many potential reasons for rejection, e.g.: the request contained invalid or incomplete information, failed policy checks (such as subject name or key usage restrictions), the requester is not authorized, or the CA administrator manually denies the request due to security concerns or mismatched credentials. Contact the server administrator for details.

In such a case, try re-editing the request and applying it again. A new private key will be generated and the request will get PENDING status again.

```
example:/pki/cert/#> enroll mycert1

[ *** modify the request ***]

example:/pki/cert/enroll-mycert1/#> end
example:/pki/cert/#> show

Certificates
TYPE     HASH     EXPIRES     NAME                DIST LABEL
Pub               PENDING     N/A                 SCEP mycert1
example:/pki/cert/#>
```

Enrolling from Windows NDES server

In a similar fashion, we can now enroll a certificate from the NDES server:

```
example:/#> pki
example:/pki/#> cert
example:/pki/cert/#> enroll mycert2
example:/pki/cert/enroll-mycert2/#> server win-srv
example:/pki/cert/enroll-mycert2/#> password CFFD9A59
example:/pki/cert/enroll-mycert2/#> dn CN=mycert2.co.uk, C=UK
example:/pki/cert/enroll-mycert2/#> san 192.168.100.102
example:/pki/cert/enroll-mycert2/#> profile RnDTestSCEPCertificate
example:/pki/cert/enroll-mycert2/#> show
Server                        : win-srv
Password                      : CFFD9A59
DN                            : CN=mycert2.co.uk, C=UK
Renewal-threshold             : 80 %
SAN                           : 192.168.100.102
Keysize                       : 2048
Profile                       : RnDTestSCEPCertificate
```

Applying the configuration starts the enrollment.

```
example:/pki/cert/enroll-mycert2/#> end
```

After the second certificate request is approved, the

```
example:/#> pki cert show
TYPE      HASH      EXPIRES      NAME                 DIST LABEL

Pub       a9b6da71 Feb 15 2025 mycert2.co.uk       SCEP mycert2
Pub       5bddad69 Feb 17 2025 mycert1.co.uk       SCEP mycert1
Pub       4e3ca0d2 Jan 19 2038 zero-ee-00-00.local      web-default
```

Now we have two certificates enrolled: one from an OpenXPKI server and the other from a Windows NDES server.

The device will start to send renewal requests for updated certificates when 80% of the validity period has elapsed, as specified in the 'Renewal-threshold' setting.

Troubleshooting

Enrollment Server status

```
example:/pki/#> show

PKI
Daemon
Status                          : Running
Uptime                          : 10 min 12 sec

Enroll Servers
ID  LABEL      STATUS   TIME                 Message
1   xpki-srv   Enabled  2025-06-02 14:01:09  CA fingerprint mismatch
2   win-srv    Enabled  2025-06-02 14:01:09  responding

[ ... ]
```

When server is not responding one of the following STATUS messages might show up:

- server not reachable - Indicates that server's end point is not reached, the server's url might be misspelled. Trying pinging the server to verify that the host is reachable.
- CA not downloaded - Attempt to download CA/RA certificates failed.
- CA fingerprint mismatch - CA certificate has been downloaded but its Root-CA's fingerprint does not match the one configured.
- no RA cert found - No RA certificate has been downloaded (see an issue below).

Examine syslog messages, verify configuration.

Certificate keeps PENDING

If a certificate keeps pending after applying the enrollment request there might be several reasons for that. To solve the issue:

- Check if the enrollment server is responding.
- Check if the correct server is associated with the certificate request.
- Contact the server administrator, the request might need manual approval.

No RA cert found for signing or encipherment

During enrollment attempt an error message “No RA cert found for signing or encipherment” appears. This indicates that there is no dedicated RA certificate found and the CA certificate at the bottom of the chain does not have required KeyUsage flags (digitalSignature and keyEncipherment) set. Try configuring the ra-dn option.

4.9. Routing

4.9.1. VRRP HowTo

About

This document provides a basic VRRP use-case, and how it is configured using the CLI.

For an overview of all available configuration settings for VRRP refer to the configuration guide:

- VRRP Configuration Guide

Introduction

The simplest use case for VRRP involves two redundant physical routers that together constitute a virtual router between two networks, as showcased in **Figure 1**.

```
                                   |
                                .--.-.
                               ( (    )__
                              (_,  \ ) ,_)  Network N2
                                '-'--`--'
                                 |
                  ---------+--------+--------+---------
                           |                 |
VRRP Setup             .---+---.         .---+---.         VRRP Setup
instance: 1            |       |         |       |         instance: 1
vrid: 1                |  R1   |         |  R2   |         vrid: 1
vaddr: 10.0.0.100      |       | vlan1   |       | vlan1   vaddr: 10.0.0.100
prio: 100              '---+---'         '---+---'         prio: 50
                         |.1                |.2
                           |                 |
                  --+------+---Network N1----+---------
                    |          10.0.0.0/24
                  .-+--.
                  | H  | route to N2 via 10.0.0.100
                  '----'
```

Figure 1: A VRRP setup with two routers forming a virtual router setup.

In this case, host *H* can reach network *N2* even though physical router *R1* is down. From the perspective of *H*, there is only one router - the virtual router - and which of the participating physical routers is the master router does not matter. Additional physical routers *R3* etc. may also participate in the virtual router to increase redundancy even further; all that is required is that all participating routers *R1*, *R2*, *R3* etc. run VRRP instances on their *N1*-facing interfaces, using the same virtual IP address and virtual router ID, and presumably different physical (real) addresses on those same interfaces.

Configuration

Configure both of the routers to be part of the same virtual router. We want R1 to be the default master router, therefore it needs to be assigned a higher priority.

It is assumed that both routers have two different VLANs configured for each of the networks described in **Figure 1**.

Router 1

Configure a VRRP instance:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> vrrp 1
Creating new VRRP instance: 1
R1:/config/router/vrrp-1/#> priority 100
```

```
R1:/config/router/vrrp-1/#> vrid 1
R1:/config/router/vrrp-1/#> iface vlan1
R1:/config/router/vrrp-1/#> address 10.0.0.100
R1:/config/router/vrrp-1/#> leave
R1:/#>
```

Notes on the configuration:

- R1 will be part of the virtual router with `vrid` 1.
- The `address` of the virtual router will be 10.0.0.100.
- List for traffic that enters `iface` vlan1.

Router 2

Configure a VRRP instance:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> vrrp 1
Creating new VRRP instance: 1
R2:/config/router/vrrp-1/#> priority 50
R2:/config/router/vrrp-1/#> vrid 1
R2:/config/router/vrrp-1/#> iface vlan1
R2:/config/router/vrrp-1/#> address 10.0.0.100
R2:/config/router/vrrp-1/#> leave
R2:/#>
```

Notes on the configuration:

- R2 will be part of a the virtual router with `vrid` 1.
- The `address` of the virtual router will be 10.0.0.100.
- Listen for traffic that enters `iface` vlan1.

Status

The status of the VRRP can be observed on each individual VRRP router. **Router 1:**

```
R1:/#> show vrrp

VRRP Instances
INSTANCE NAME    BASE INTERFACE   VRID ADDRESS         PRIORITY STATE   GROUP
1                vlan1            1    10.0.0.100      100(100) MASTER  -
```

Router 2:

```
R1:/#> show vrrp

VRRP Instances
INSTANCE NAME    BASE INTERFACE   VRID ADDRESS         PRIORITY STATE   GROUP
1                vlan1            1    10.0.0.100      50(50)   BACKUP  -
```

4.9.2. VRRP Load Balancing HowTo

About

This document provides a VRRP Load Balancing use-case, and how it is configured using the CLI.

For an overview of all available configuration settings for VRRP refer to the configuration guide:

- VRRP Configuration Guide

Introduction

It is possible to configure VRRP to perform some limited load balancing, an example of this can be observed in **Figure 1** below.

```
                                        |
                                    .--.-.
                                  ( (    )__
                                 (_,  \ ) ,_)  Network N2
                                   '-'--`--'
                                     |
                  ---------+--------+--------+---------
                           |                 |
VRRP Setup                 |                 |              VRRP Setup
instance: i1               |                 |              instance: i1
vrid: 1                    |                 |              vrid: 1
vaddr: 10.0.0.100          |                 |              vaddr: 10.0.0.100
prio: 100             .---+---.         .---+---.          prio: 50
                      |       |         |       |
                      |  R1   |         |  R2   |
VRRP Setup            |       | vlan1   |       | vlan1    VRRP Setup
instance: i2          '---+---'         '---+---'          instance: i2
vrid: 2                   |.1                |.2           vrid: 2
vaddr: 10.0.0.101         |                  |              vaddr: 10.0.0.101
prio: 50                  |                  |              prio: 100
                          |                  |
                          |                  |
                  --+------+---Network N1----+------+--
                    |      |   10.0.0.0/24   |      |
            |       |               .-+--. .-+--.
                    |      |                | H3 | | H4 | route to N2 via 10.0.0.101
                    |      |                '----' '----'
                  .-+--. .-+--.
                  | H1 | | H2 | route to N2 via 10.0.0.100
                  '----' '----'
```

Figure 1: A VRRP setup with two routers forming a virtual router that performs load balancing on traffic from Network N1 to Network N2.

In this case, two instances *i1* and *i2*, with VRIDs 1 and 2 respectively, have been set up on each of the participating routers *R1* and *R2*. The priorities are chosen so that by default *R1* is the master router for VRID 1 (the virtual router 10.0.0.100), and *R2* is the VRID 2 (10.0.0.101) master. Then, on network *N1*, hosts *H1* and *H2* are set up with a route to network *N2* via 10.0.0.100, while hosts *H3* and *H4* use 10.0.0.101 instead. Due to the (presumably static) configuration of routes in this manner, as long as both *R1* and *R2* are up, traffic from *N1* will be divided (though not actually *balanced*) between the two physical routers.

The need for the user to manually configure relevant hosts, and achieve a level of load splitting by statically moving hosts between virtual routers, shows some of the limitations of VRRP as a load-balancing system.

Configuration

Configure two VRRP instances on each of the routers. We configure the instances so that each physical router will become master for one of the virtual routers each.

It is assumed that both routers VLANs configured for each of the networks described in **Figure 1**.

Router 1

Configure VRRP **instance i1**:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> vrrp i1
Creating new VRRP instance: i1
R1:/config/router/vrrp-i1/#> vrid 1
R1:/config/router/vrrp-i1/#> address 10.0.0.100
R1:/config/router/vrrp-i1/#> prio 100
R1:/config/router/vrrp-i1/#> iface vlan1
R1:/config/router/vrrp-i1/#> preempt delay 0
R1:/config/router/vrrp-i1/#> end
R1:/config/router/#>
```

Configure VRRP **instance i2**:

```
R1:/config/router/#> vrrp i2
Creating new VRRP instance: i2
R1:/config/router/vrrp-i2/#> vrid 2
R1:/config/router/vrrp-i2/#> address 10.0.0.101
R1:/config/router/vrrp-i2/#> prio 50
R1:/config/router/vrrp-i2/#> iface vlan1
R1:/config/router/vrrp-i2/#> preempt delay 0
R1:/config/router/vrrp-i2/#> leave
R1:/#>
```

Router 2

Configure VRRP **instance i1**:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> vrrp i1
Creating new VRRP instance: i1
R2:/config/router/vrrp-i1/#> vrid 1
R2:/config/router/vrrp-i1/#> address 10.0.0.100
R2:/config/router/vrrp-i1/#> prio 50
R2:/config/router/vrrp-i1/#> iface vlan1
R2:/config/router/vrrp-i1/#> preempt delay 0
R2:/config/router/vrrp-i1/#> end
R2:/config/router/#>
```

Configure VRRP **instance i2**:

```
R2:/config/router/#> vrrp i2
Creating new VRRP instance: i2
R2:/config/router/vrrp-i2/#> vrid 2
R2:/config/router/vrrp-i2/#> address 10.0.0.101
R2:/config/router/vrrp-i2/#> prio 100
R2:/config/router/vrrp-i2/#> iface vlan1
R2:/config/router/vrrp-i2/#> preempt delay 0
R2:/config/router/vrrp-i2/#> leave
R2:/#>
```

Status

The status of the VRRP can be observed on each individual VRRP router.

Router 1:

```
R1:/#> show vrrp

VRRP Instances
INSTANCE NAME    BASE INTERFACE   VRID ADDRESS         PRIORITY STATE   GROUP
i1               vlan1            1    10.0.0.100      100(100) MASTER  -
i2               vlan1            2    10.0.0.101      50(50)   BACKUP  -
```

Router 2:

```
R2:/#> show vrrp

VRRP Instances
INSTANCE NAME    BASE INTERFACE   VRID ADDRESS         PRIORITY STATE   GROUP
i1               vlan1            1    10.0.0.100      50(50)   BACKUP  -
i2               vlan1            2    10.0.0.101      100(100) MASTER  -
```

As can be seen the master for the virtual router with **vrid 1** is router **R1** and the master for the virtual router with **vrid 2** is router **R2**.

4.9.3. VRRP Synchronisation Groups HowTo

About

This document provides a VRRP Synchronisation groups use-case, and how it is configured using the CLI.

For an overview of all available configuration settings for VRRP refer to the configuration guide:

- VRRP Configuration Guide

Introduction

A more complex use-case than a basic VRRP setup, involving synchronisation groups, can be constructed as show in **Figure 1**:

```
                                                  .---.
                                                  | H | route to N1 via 192.168.0.10
                                                  '-+-'
                           192.168.0.0/24           |
                 ---------+---Network N2----+------+--
                          |                 |
VRRP Setup                |                 |             VRRP Setup
instance: i2              |                 |             instance: i2
vrid: 1                   |                 |             vrid: 1
vaddr: 192.168.0.10       |.1               |.2           vaddr: 192.168.0.10
prio: 100            .---+---.         .---+---.         prio: 10
group: g1            |       | vlan2   |       | vlan2   group: g1
                     |  R1   |         |  R2   |
VRRP Setup           |       | vlan1   |       | vlan1   VRRP Setup
instance: i1         '---+---'         '---+---'         instance: i1
vrid: 1                  |.1               |.2           vrid: 1
vaddr: 10.0.0.100        |                 |             vaddr: 10.0.0.100
prio: 100                |                 |             prio: 50
group: g1                |                 |             group: g1
                         |                 |
                 --+------+---Network N1----+---------
                   |          10.0.0.0/24
                 .-+--.
                 | H1 | route to N2 via 10.0.0.100
                 '----'
```

Figure 1: A VRRP setup with two routers forming a virtual router setup with synchronisation groups.

From the perspective of host H1, this case is similar to the one presented here. Equivalently, the same applies to host *H2*, only that it has a redundant route to network *N1* instead.

However, in this scenario, it is possible for router *R1* to be the master router for traffic to *N1*, and for *R2* to be the master router towards *N2*. Assuming there is a requirement to avoid this kind of asymmetrical routing, both VRRP instances on each router *R1*, *R2* can be grouped together into synchronisation groups. This will ensure that one of *R1* or *R2* is always selected as the master router of both virtual routers. This example also illustrates that VRIDs can be reused across different interfaces on each router.

Configuration

Configure two VRRP instances per router and add those two instances to a synchronisation group.

It is assumed that both routers have two different VLANs configured for each of the networks described in **Figure 1**.

Router 1

Configure VRRP **instance i1**:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> vrrp i1
Creating new VRRP instance: i1
R1:/config/router/vrrp-i1/#> vrid 1
R1:/config/router/vrrp-i1/#> address 10.0.0.100
R1:/config/router/vrrp-i1/#> prio 100
R1:/config/router/vrrp-i1/#> iface vlan1
R1:/config/router/vrrp-i1/#> end
R1:/config/router/#>
```

Configure VRRP **instance i2**:

```
R1:/config/router/#> vrrp i2
Creating new VRRP instance: i2
R1:/config/router/vrrp-i2/#> vrid 1
R1:/config/router/vrrp-i2/#> address 192.168.0.10
R1:/config/router/vrrp-i2/#> prio 100
R1:/config/router/vrrp-i2/#> iface vlan2
R1:/config/router/vrrp-i2/#> end
R1:/config/router/#>
```

If we want to ensure that the router with highest priority is always selected when possible, we can enable **preemption**:

```
R2:/config/router/#> vrrp i1
R2:/config/router/vrrp-i1/#> preempt delay 0
R2:/config/router/vrrp-i1/#> end
R2:/config/router/#>
R2:/config/router/#> vrrp i2
R2:/config/router/vrrp-i2/#> preempt delay 0
R2:/config/router/vrrp-i2/#> end
R2:/config/router/#>
```

Configure the synchronisation **group g1**:

```
R1:/config/router/#> group g1
Creating new VRRP group: g1
R1:/config/router/vrrp-grp-g1/#> member i1
Added instance 'i1' as group member 1.
R1:/config/router/vrrp-grp-g1/#> member i2
Added instance 'i2' as group member 2.
R1:/config/router/vrrp-grp-g1/#> leave
R1:/#>
```

Router 2

Configure VRRP **instance i1**:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> vrrp i1
Creating new VRRP instance: i1
R2:/config/router/vrrp-i1/#> vrid 1
R2:/config/router/vrrp-i1/#> address 10.0.0.100
R2:/config/router/vrrp-i1/#> prio 50
```

```
R2:/config/router/vrrp-i1/#> iface vlan1
R2:/config/router/vrrp-i1/#> end
R2:/config/router/#>
```

Configure VRRP **instance i2**:

```
R2:/config/router/#> vrrp i2
Creating new VRRP instance: i2
R2:/config/router/vrrp-i2/#> vrid 1
R2:/config/router/vrrp-i2/#> address 192.168.0.10
R2:/config/router/vrrp-i2/#> prio 10
R2:/config/router/vrrp-i2/#> iface vlan2
R2:/config/router/vrrp-i2/#> end
R2:/config/router/#>
```

If we want to ensure that the router with highest priority is always selected when possible, we can enable **preemption**:

```
R2:/config/router/#> vrrp i1
R2:/config/router/vrrp-i1/#> preempt delay 0
R2:/config/router/vrrp-i1/#> end
R2:/config/router/#>
R2:/config/router/#> vrrp i2
R2:/config/router/vrrp-i2/#> preempt delay 0
R2:/config/router/vrrp-i2/#> end
R2:/config/router/#>
```

Configure the synchronisation **group g1**:

```
R2:/config/router/#> group g1
Creating new VRRP group: g1
R2:/config/router/vrrp-grp-g1/#> member i1
Added instance 'i1' as group member 1.
R2:/config/router/vrrp-grp-g1/#> member i2
Added instance 'i2' as group member 2.
R2:/config/router/vrrp-grp-g1/#> leave
R2:/#>
```

Status

The status of the VRRP can be observed on each individual VRRP router. **Router 1:**

```
R1:/#> show vrrp

VRRP Instances
INSTANCE NAME    BASE INTERFACE   VRID ADDRESS         PRIORITY STATE   GROUP
i1               vlan1            1    10.0.0.100      100(100) MASTER  g1
i2               vlan2            1    192.168.0.10    100(100) MASTER  g1
```

Router 2:

```
R2:/#> show vrrp

VRRP Instances
INSTANCE NAME    BASE INTERFACE   VRID ADDRESS         PRIORITY STATE   GROUP
```

```
i1              vlan1           1    10.0.0.100      50(100) BACKUP  g1
i2              vlan2           1    192.168.0.10    10(100) BACKUP  g1
```

4.9.4. OSPF HowTo

About

This document provides a basic OSPF use-case, and how it is configured using the CLI.

For an overview of all available configuration settings for OSPF refer to the configuration guide:

- OSPF Configuration Guide

Introduction

In Figure 1 below a simple example of a network is presented, with four routers and a number of networks connected to them. In order to ensure that the different networks can reach each other we want to utilise OSPF on the routers to dynamically set up the necessary routes.

```
   Net-A                                Net-B
  .--.-.                               .--.-.
 ( (    )__                          ( (    )__
(_,  \ ) ,_)                         (_,  \ ) ,_)
  '-'--`--'                            '-'--`--'
     |                                    |
   | 10.0.11.0/24                       | 10.0.14.0/24
   .1|                                  .1|
.----+-----.                         .----+-----.
|          |.1                     .2|          |
| Router-A +------------------------+ Router-B |
|          |      10.0.1.0/24       |          |
'----+-----'                         '----+-----'
   .2|                                  .1|
     | 10.0.2.0/24                        | 10.0.4.0/24
   .1|                                  .2|
.----+-----.                         .----+-----.
|          |.2                     .1|          |
| Router-C +------------------------+ Router-D |
|          |      10.0.3.0/24       |          |
'----+-----'                         '----+-----'
   .1|                                  .1|
     | 10.0.12.0/24                       | 10.0.13.0/24
     |                                    |
  .--.-.                               .--.-.
 ( (    )__                          ( (    )__
(_,  \ ) ,_)                         (_,  \ ) ,_)
  '-'--`--'                            '-'--`--'
   Net-C                                Net-D
```

Figure 1: Example OSPF network with IP addresses and subnets.

In a simple case like this, configuring an OSPF network is not something that is difficult. Essentially the only thing that needs to be configured is the networks that should be served by OSPF on each of the routers. In this case, for instance, Router-A would be configured to include and announce all its subnets in the OSPF domain (10.0.11.0/24, 10.0.1.0/24, 10.0.2.0/24). The other routers in the setup would be configured in the same manner.

Configuration

The individual routers should be configured to include all their network connections in the OSPF domain. The example presented in this use-case is a “flat” OSPF network, i.e. only one area is utilised, the backbone (area 0.0.0.0).

When configuring OSPF it is **very important** that the OSPF `router-id` is unique to each individual router part of the OSPF network. It is *common practice* to set the router-id to one of the IP addresses assigned to the router.

> **NOTE**
>
> In this example we do not explicitly configure the router-id, we let it be assigned automatically. In this case the router-id will be assigned the IP address of one of the router's interfaces (the one with the highest IP address). For example in this case, Router-A will be assigned a router-id of 10.0.11.1, since it is the highest IP address if its interfaces.

Router A

Configure OSPF on `Router-A` to announce all its network connections as part of the OSPF domain:

```
Router-A:/#> configure
Router-A:/config/#> router
Router-A:/config/router/#> ospf
Router-A:/config/router/ospf/#> network 10.0.1.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-A:/config/router/ospf/#> network 10.0.2.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-A:/config/router/ospf/#> network 10.0.11.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-A:/config/router/ospf/#> leave
Router-A:/#>
```

Router B

Configure OSPF on `Router-B` to announce all its network connections as part of the OSPF domain:

```
Router-B:/#> configure
Router-B:/config/#> router
Router-B:/config/router/#> ospf
Router-B:/config/router/ospf/#> network 10.0.1.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-B:/config/router/ospf/#> network 10.0.4.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-B:/config/router/ospf/#> network 10.0.14.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-B:/config/router/ospf/#> leave
Router-B:/#>
```

Router C

Configure OSPF on `Router-C` to announce all its network connections as part of the OSPF domain:

```
Router-C:/#> configure
Router-C:/config/#> router
Router-C:/config/router/#> ospf
Router-C:/config/router/ospf/#> network 10.0.2.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-C:/config/router/ospf/#> network 10.0.3.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-C:/config/router/ospf/#> network 10.0.12.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-C:/config/router/ospf/#> leave
Router-C:/#>
```

© 2026 Westermo Network Technologies AB

Router D

Configure OSPF on `Router-D` to announce all its network connections as part of the OSPF domain:

```
Router-D:/#> configure
Router-D:/config/#> router
Router-D:/config/router/#> ospf
Router-D:/config/router/ospf/#> network 10.0.3.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-D:/config/router/ospf/#> network 10.0.4.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-D:/config/router/ospf/#> network 10.0.13.0/24
Defaulting to OSPF backbone, area 0.0.0.0
Router-D:/config/router/ospf/#> leave
Router-D:/#>
```

Status

There are a few different ways we can observe the status of our OSPF network. The most important thing is that OSPF learns the routes. We can check the routes learnt by OSPF in the following way:

```
Router-A:/#> show ip ospf route
============ OSPF network routing table ============
N    10.0.1.0/24           [10] area: 0.0.0.0
                           directly attached to vlan1
N    10.0.2.0/24           [10] area: 0.0.0.0
                           directly attached to vlan2
N    10.0.3.0/24           [20] area: 0.0.0.0
                           via 10.0.2.1, vlan2
N    10.0.4.0/24           [20] area: 0.0.0.0
                           via 10.0.1.2, vlan1
N    10.0.11.0/24          [10] area: 0.0.0.0
                           directly attached to vlan3
N    10.0.12.0/24          [20] area: 0.0.0.0
                           via 10.0.2.1, vlan2
N    10.0.13.0/24          [30] area: 0.0.0.0
                           via 10.0.1.2, vlan1
                           via 10.0.2.1, vlan2
N    10.0.14.0/24          [20] area: 0.0.0.0
                           via 10.0.1.2, vlan1

============ OSPF router routing table ============

============ OSPF external routing table ==========
```

We can also check routes in the following manner, where routes learnt by OSPF is signified by `O`:

```
Router-A:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O   10.0.1.0/24 [110/10] is directly connected, vlan1, 01:40:21
C>* 10.0.1.0/24 is directly connected, vlan1
O   10.0.2.0/24 [110/10] is directly connected, vlan2, 01:40:21
C>* 10.0.2.0/24 is directly connected, vlan2
O>* 10.0.3.0/24 [110/20] via 10.0.2.1, vlan2, 01:39:41
O>* 10.0.4.0/24 [110/20] via 10.0.1.2, vlan1, 01:35:46
O   10.0.11.0/24 [110/10] is directly connected, vlan3, 01:40:21
```

```
C>* 10.0.11.0/24 is directly connected, vlan3
O>* 10.0.12.0/24 [110/20] via 10.0.2.1, vlan2, 01:39:41
O>* 10.0.13.0/24 [110/30] via 10.0.1.2, vlan1, 01:35:46
  *                       via 10.0.2.1, vlan2, 01:35:46
O>* 10.0.14.0/24 [110/20] via 10.0.1.2, vlan1, 01:35:46
```

4.9.5. OSPF Areas HowTo

About

This document aims to provide example OSPF use-cases where we configure OSPF areas to limit the amount of routing information each router needs to process and store.

For an overview of all available configuration settings for OSPF refer to the configuration guide:

- OSPF Configuration Guide

Introduction

In certain situations it can be beneficial to reduce the amount of routing information that each individual router in an OSPF network needs to handle and store in their internal database. In order to achieve this OSPF provides the concept of areas, these can help segment an OSPF network.

The following example use-cases are presented in this document:

- Case 1: Standard Area
- Case 2: Stub Area
- Case 3: Not So Stubby Area (NSSA)

Case 1: Standard Area

This case present a very simple example how to segment an OSPF networks into different area domains. The example in Figure 1 below, only showcase five different OSPF routers divided into three different areas. If more areas are needed they simply add them in the same manner as area 0.0.0.1 and 0.0.0.2 will be added in this example.

> **NOTE**
>
> Remember that any additional area that is added must be directly connected to the backbone (area 0.0.0.0). In cases where this for some reason is not possible, an OSPF feature called *Virtual Link* could be utilised. However, this is currently not supported.

```
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. Area 0.0.0.0                                                       .
.  (Backbone)                                                        .
.                          .-------.                                 .
.            10.0.1.0/24 |       | 10.0.2.0/24                      .
.      .-----------------+  BR   +-----------------.                .
.      |                .1|       |.1                |               .
.    .2|                  '-------'                  |.2             .
.  .---+---.                                     .---+---.           .
. .|       |. . . . . . . . . . . . . . . . . . .|       |. .
   | ABR1  |                                     | ABR2  |
. .|       |. . . . . . . .     . . . . . . . . .|       |. .
.  '---+---' Area 0.0.0.1 .     . Area 0.0.0.2 '---+---'           .
.    .1|                  .     .                  |.1             .
.      |                  .     .                  |               .
.      | 192.168.1.0/24   .     .   192.168.2.0/24 |               .
.      |                  .     .                  |               .
.    .2|                  .     .                  |.2             .
.  .---+---.              .     .              .---+---.           .
.  |       |              .     .              |       |           .
.  |  R1   |              .     .              |  R2   |           .
.  |       |              .     .              |       |           .
.  '-------'              .     .              '-------'           .
. . . . . . . . . . . . . .     . . . . . . . . . . . . . . . . . . .
```

Figure 1: Example of an OSPF network segmented into three different areas.

Segmenting an OSPF network into basic area domains is essentially the same as creating a “flat” OSPF network (only one area 0.0.0.0), as described here. The only addition needed to create simple area segmentation, is to include what area a network that OSPF should serve is to be part of.

For instance, in this case the router `ABR1` would be configured to serve the network `10.0.1.0/24` as part of area `0.0.0.0` and the network `192.168.1.0/24` as part of the area `0.0.0.1`. The same principle would be applied for the configuration of OSPF on the other routers in the entire OSPF network,

Configuration

We configure OSPF on each of the individual routers that, and ensure that the networks are added to the correct areas.

Backbone Router

The backbone router will serve two networks `10.0.1.0/24` and `10.0.2.0/24`, both being part of area 0.0.0.0:

```
BR:/#> configure
BR:/config/#> router
BR:/config/router/#> ospf
BR:/config/router/ospf/#> network 10.0.1.0/24 area 0
BR:/config/router/ospf/#> network 10.0.2.0/24 area 0
BR:/config/router/ospf/#> leave
BR:/#>
```

Area Border Router 1

The `ABR1` will serve two networks `10.0.1.0/24` and `192.168.1.0/24`, with the first being part of area `0.0.0.0` and the second area `0.0.0.1`:

```
ABR1:/#> configure
ABR1:/config/#> router
ABR1:/config/router/#> ospf
ABR1:/config/router/ospf/#> network 10.0.1.0/24 area 0
ABR1:/config/router/ospf/#> network 192.168.1.0/24 area 1
ABR1:/config/router/ospf/#> leave
ABR1:/#>
```

Router 1

The `R1` will serve one network `192.168.1.0/24`, with it being part of area `0.0.0.1`:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> ospf
R1:/config/router/ospf/#> network 192.168.1.0/24 area 1
R1:/config/router/ospf/#> leave
R1:/#>
```

Area Border Router 2

The `ABR2` will serve two networks `10.0.2.0/24` and `192.168.2.0/24`, with the first being part of area `0.0.0.0` and the second area `0.0.0.2`:

```
ABR2:/#> configure
ABR2:/config/#> router
ABR2:/config/router/#> ospf
ABR2:/config/router/ospf/#> network 10.0.2.0/24 area 0
ABR2:/config/router/ospf/#> network 192.168.2.0/24 area 2
ABR2:/config/router/ospf/#> leave
ABR2:/#>
```

Router 2

The `R2` will serve one network `192.168.2.0/24`, with it being part of area `0.0.0.2`:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> ospf
R2:/config/router/ospf/#> network 192.168.2.0/24 area 2
R2:/config/router/ospf/#> leave
R2:/#>
```

Case 2: Stub Area

The example use-case, presented in Figure 2, looks almost the same as Case 1, the difference is that we have a new router added to our OSPF network as part of area 0.0.0.0 (backbone). This is an *Autonomous System Boundary Router* (ASBR), which is connected to some external network that is distributed into the OSPF domain.

> **TIP**
>
> We do not cover configuration of route redistribution in this example use-case, for an example use-case of how to configure route redistribution see this.

For this example use-case we are interested in limiting the amount of external routing information that needs to be propagated throughout the network, by limiting the routing information that is passed into area 0.0.0.1 and area 0.0.0.2 even further. In order to limit external routing information that is passed into a certain area, OSPF provides a specific type of area known as a *stub area*.

> **NOTE**
>
> The backbone (area 0.0.0.0) cannot be declared a stub area.

```
                          External Network
                              .--.-.
                             ( (    )__
                            (_,  \ ) ,_)
                              '-'--`--'
                                  |
                             .---+---.
                             |       |
. . . . . . . . . . . . . .| ASBR  |. . . . . . . . . . . . . .
. Area 0.0.0.0               |       |                          .
.  (Backbone)          '---+---'                        .
.                               |                              .
.                               |                              .
.                             .---+---.                        .
.             10.0.1.0/24 |       | 10.0.2.0/24                .
.       .-----------------+  BR   +-----------------.          .
.       |                 .1|       |.1                |        .
.     .2|                   '-------'                  |.2     .
.   .---+---.                                       .---+---.   .
. .|       |. . . . . . . . . . . . . . . . . . .|       |. .
   | ABR1  |                                     | ABR2  |
. .|       |. . . . . . . .     . . . . . . . .|       |. .
.   '---+---' Area 0.0.0.1 .     . Area 0.0.0.2 '---+---'   .
.     .1|        (Stub)    .     .    (Stub)        |.1     .
.       |                  .     .                  |       .
.       | 192.168.1.0/24   .     .   192.168.2.0/24 |       .
.       |                  .     .                  |       .
.     .2|                  .     .                  |.2     .
.   .---+---.              .     .              .---+---.   .
.   |       |              .     .              |       |   .
.   |  R1   |              .     .              |  R2   |   .
.   |       |              .     .              |       |   .
.   '-------'              .     .              '-------'   .
. . . . . . . . . . . . . .     . . . . . . . . . . . . . . .
```

Figure 2: Example of an OSPF network segmented into three different areas, with two of the designated as stub areas. The backbone area also distributes some external network into the OSPF domain.

The routers present within the stub areas still have full routing information for the networks and routers within their own area, along with summary routing information to destinations within other areas. However, the routers within the stub area will harbour no routing information learnt from external routing sources.

For the routers inside the stub area, reaching the destinations outside the OSPF domain is based on *default routing* towards the area's ABR. All external routing information that is to be distributed into an area will be filtered out by the ABR, instead the ABR will inject a default route towards itself into the area.

Hence, if an external destination is to be reached from a router located inside a stub area the default route will be used to reach the area's ABR. In turn the ABR will have the routing information necessary to route traffic further on its way towards the intended destination.

Configuration

We assume the starting configuration of the devices is based on Case 1.

In order to create a stub area, all the routers that constitute the area (including the areas ABR) must configure the area as stub.

For routers **ABR1** and **R1** area 0.0.0.1 needs to be configured stub. And for the routers **ABR2** and **R2** area 0.0.0.2 needs to be configured stub.

ABR1

Go into the configuration for area 1 and set it to act as a stub area:

```
ABR1:/#> configure
ABR1:/config/#> router
ABR1:/config/router/#> ospf
ABR1:/config/router/ospf/#> area 1
ABR1:/config/router/ospf/area-0.0.0.1/#> stub
ABR1:/config/router/ospf/area-0.0.0.1/#> leave
ABR1:/#>
```

R1

Go into the configuration for area 1 and set it to act as a stub area:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> ospf
R1:/config/router/ospf/#> area 1
R1:/config/router/ospf/area-0.0.0.1/#> stub
R1:/config/router/ospf/area-0.0.0.1/#> leave
R1:/#>
```

ABR2

Go into the configuration for area 2 and set it to act as a stub area:

```
ABR2:/#> configure
ABR2:/config/#> router
ABR2:/config/router/#> ospf
ABR2:/config/router/ospf/#> area 2
ABR2:/config/router/ospf/area-0.0.0.2/#> stub
ABR2:/config/router/ospf/area-0.0.0.2/#> leave
ABR2:/#>
```

R2

Go into the configuration for area 2 and set it to act as a stub area:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> ospf
R2:/config/router/ospf/#> area 2
R2:/config/router/ospf/area-0.0.0.2/#> stub
R2:/config/router/ospf/area-0.0.0.2/#> leave
R2:/#>
```

Totally Stubby Area

Stub areas can be configured to limit the amount of routing information that is distributed into it even further by declaring it a *Totally stub area*.

What this will do is that summary routing information from other areas within the OSPF domain, will not be distributed to the routers inside the area. The behaviour is essentially the same as what a regular *stub* area does in limiting external routing information inside the area, but this is limiting routing information from other OSPF areas.

If an area is to be configured as a totally stub area, it only needs to be done on the area's ABR:

```
ABR:/#> configure
ABR:/config/#> router
ABR:/config/router/#> ospf
ABR:/config/router/ospf/#> area 1
ABR:/config/router/ospf/area-0.0.0.1/#> stub no-summary
ABR:/config/router/ospf/area-0.0.0.1/#> leave
ABR:/#>
```

Default route

As mentioned, the ABR will inject a default route towards itself into the area it serves. The cost of the default route injected into the area is by default set to 1. If this cost needs to be adjusted it can be done in the following manner:

```
ABR:/#> configure
ABR:/config/#> router
ABR:/config/router/#> ospf
ABR:/config/router/ospf/#> area 1
ABR:/config/router/ospf/area-0.0.0.1/#> default-cost 10
ABR:/config/router/ospf/area-0.0.0.1/#> leave
ABR:/#>
```

Case 3: Not So Stubby Area (NSSA)

This use-case builds further upon the one presented in Case 2, describing *stub areas*. As explained a stub area cannot redistribute routing information that has been learnt from external sources. What this entails is that a stub area cannot contain any *Autonomous System Border Router* (ASBR).

In the use-case presented in this example, as seen in Figure 3, we still want to limit the routing information that is distributed by utilising stub areas. However, in this case we have an ASBR router inside area 0.0.0.2. This is not possible with a stub area, but OSPF support another type of area that allows for this called a Not So Stubby Area (NSSA).

```
                          External Network
                              .--.-.
                             ( (    )__
                            (_,  \ ) ,_)
                              '-'--`--'
                                  |
                             .---+---.
                             |       |
. . . . . . . . . . . . . .| ASBR1 |. . . . . . . . . . . . . .
. Area 0.0.0.0               |       |                          .
.  (Backbone)           '---+---'                        .
.                               |                             .
.                               |                             .
.                             .---+---.                       .
.             10.0.1.0/24 |       | 10.0.2.0/24               .
.       .-----------------+  BR   +-----------------.         .
.       |                 .1|       |.1                |       .
.     .2|                   '-------'                  |.2     .
.   .---+---.                                     .---+---.   .
. .|       |. . . . . . . . . . . . . . . . . . .|       |. .
   | ABR1  |                                     | ABR2  |
. .|       |. . . . . . . .     . . . . . . . .|       |. .
.   '---+---' Area 0.0.0.1 .     . Area 0.0.0.2 '---+---'   .
.     .1|        (Stub)    .     .    (NSSA)        |.1     .
.       |                  .     .                  |       .
.       | 192.168.1.0/24   .     .   192.168.2.0/24 |       .
.       |                  .     .                  |       .
.     .2|                  .     .                  |.2     .
.   .---+---.              .     .              .---+---.   .
.   |       |              .     .              |       |   .
.   |  R1   |              .     .              |  R2   |   .
.   |       |              .     .              |       |   .
.   '-------'              .     .              '---+---'   .
. . . . . . . . . . . . . .     .                  |       .
                                .                  |       .
                                .              .---+---.   .
                                .              |       |   .
                  . . . . . . . .| ASBR2 |. .
                                 |       |
                             '---+---'
                                 |
                                                  .--.-.
                                                 ( (    )__
                                                (_,  \ ) ,_)
                                                  '-'--`--'
                                          External Network
```

Figure 3: Example of an OSPF network segmented into three different areas, with one area configured as a NSSA and distributing an external network into the OSPF domain.

In the example both area 0.0.0.1 and 0.0.0.2 are declared as stub areas to avoid routing information being propagated from an external network redistributed by the ASBR in the backbone. But area 0.0.0.2 is declared a NSSA in order to be able to redistribute the routes from its connected external network, while still limiting routing information from the backbones external networks as previously.

Configuration

We assume the starting configuration of the devices is based on Case 2.

In order to create a NSSA, all the routers that constitute the area (including the areas ABR and ASBR) must configure the area as NSSA.

ABR2

Go into the configuration for area 2 and set it to act as a NSSA:

```
ABR2:/#> configure
ABR2:/config/#> router
ABR2:/config/router/#> ospf
ABR2:/config/router/ospf/#> area 2
ABR2:/config/router/ospf/area-0.0.0.2/#> nssa
ABR2:/config/router/ospf/area-0.0.0.2/#> leave
ABR2:/#>
```

R2

Go into the configuration for area 2 and set it to act as a NSSA:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> ospf
R2:/config/router/ospf/#> area 2
R2:/config/router/ospf/area-0.0.0.2/#> nssa
R2:/config/router/ospf/area-0.0.0.2/#> leave
R2:/#>
```

ASBR2

Go into the configuration for area 2 and set it to act as a NSSA:

```
ASBR2:/#> configure
ASBR2:/config/#> router
ASBR2:/config/router/#> ospf
ASBR2:/config/router/ospf/#> area 2
ASBR2:/config/router/ospf/area-0.0.0.2/#> nssa
ASBR2:/config/router/ospf/area-0.0.0.2/#> leave
ASBR2:/#>
```

NSSA totally stub area

Same as for stub areas, NSSA areas can also be configured to limit the amount of routing information from other ares in the OSPF network, by declaring it a *NSSA totally stub area*.

This only needs to be configured on the ABR of the area:

```
ABR:/#> configure
ABR:/config/#> router
ABR:/config/router/#> ospf
ABR:/config/router/ospf/#> area 1
ABR:/config/router/ospf/area-0.0.0.1/#> nssa no-summary
ABR:/config/router/ospf/area-0.0.0.1/#> leave
ABR:/#>
```

4.9.6. OSPF Route Summarization and Filtering HowTo

About

This document aims to provide example OSPF use-cases where we configure OSPF areas to utilise route *summarisation* and *filtering*, to reduce the amount of routing information that the ABR propagates to the rest of the OSPF network.

For an overview of all available configuration settings for OSPF refer to the configuration guide:

- OSPF Configuration Guide

Introduction

In another howto, an example was provided how the amount routing information each router in an OSPF network need to handle can be reduced by utilising OSPF areas.

In addition to this the ABR can be configured to further limit the routing information that is propagated from an area into the rest of the OSPF network. This can be done by using route summarisation and route filtering.

> **NOTE**
>
> Being able to efficiently utilise this approach requires that the network have been segmented in a efficient manner.

The following example use-cases are presented in this document:

- Case 1: Route Summarization
- Case 2: Route Filtering

Case 1: Route Summarization

In this use-case an example will be provided how routing information from different areas can be reduced even more by utilising route summarisation. Route summarisation can be applied on ABRs, before route information is distributed into the other areas in the network.

What route summarisation entails, is that multiple different devices inside an OSPF area can be represented by a single route to devices outside the area. The reason the network segmentation is important in order to efficiently execute route summarisation, is because it is based on subnet sizing.

```
                                  . . . . . . . . . . . . . . . . . . . . . . . . . .
                                  .  Area 0.0.0.1                                   .
                                  .                .-------.   | 172.16.20.0/24  .
                                  .  172.16.1.0/24 |       |   | 172.16.21.0/24  .
. . . . . . . . . . . . . .       .  .--------------+  R2   +-->| 172.16.22.0/24  .
.     Area 0.0.0.0        .       .  |              |       |   | 172.16.23.0/24  .
.                         .       .  |              '-------'   | 172.16.24.0/24  .
. .-------.               .-------. |                                            .
. |       |               |       +-'                                            .
. |  R1   +---------------+  ABR  |                                              .
. |       | 10.0.1.0/24   |       +-.                                            .
. '-------'               '-------' |                                            .
.                         .       . |              .-------.   | 192.168.20.0/24 .
.                         .       . |              |       |   | 192.168.21.0/24 .
. . . . . . . . . . . . . .       . '--------------+  R3   |-->| 192.168.22.0/24 .
                                  .  192.168.1.0/24 |       |   | 192.168.23.0/24 .
                   .                 '-------'   | 192.168.24.0/24 .
                   .                                                .
                   . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Figure 1: Example of an OSPF network segmented into two areas, with route summarisation used on the ABR.

In Figure 1, an example of a OSPF network segmented into two different areas can be seen. Route summarisation will be added on the `ABR` router in order to represent the additional networks that can be accessed from router `R2` and router `R3`, as a single route for each of the routers.

The additional networks for `R2` can be summarised with `172.16.16.0/20` and the networks for `R3` can be summarised with `192.168.16.0/20`.

Configuration

The route summarisation only needs to be configured on the ABR, since the routes are summarised when they are advertised into the backbone area:

```
ABR:/#> configure
ABR:/config/#> router
ABR:/config/router/#> ospf
ABR:/config/router/ospf/#> area 1
ABR:/config/router/ospf/area-0.0.0.1/#> range 172.16.16.0/20 advertise
ABR:/config/router/ospf/area-0.0.0.1/#> range 192.168.16.0/20 advertise
ABR:/config/router/ospf/area-0.0.0.1/#> leave
ABR:/#>
```

Resulting Routes

Without any route summarisation on the ABR these are the routes that R1 will be assigned by OSPF:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O   10.0.1.0/24 [110/10] is directly connected, vlan1, 00:12:01
C>* 10.0.1.0/24 is directly connected, vlan1
O>* 172.16.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:08:31
O>* 172.16.20.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.21.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.22.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.23.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.24.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 192.168.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:08:31
O>* 192.168.20.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.21.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.22.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.23.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.24.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
```

After routes summarisation have been applied on the `ABR` the routes on `R1` are assigned like this:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O   10.0.1.0/24 [110/10] is directly connected, vlan1, 01:59:19
C>* 10.0.1.0/24 is directly connected, vlan1
O>* 172.16.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:01:43
O>* 172.16.16.0/20 [110/30] via 10.0.1.2, vlan1, 00:01:43
```

```
O>* 192.168.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:01:43
O>* 192.168.16.0/20 [110/30] via 10.0.1.2, vlan1, 00:01:43
```

As we can see the routes towards `172.16.20.0/24`, `172.16.21.0/24`, `172.16.22.0/24`, `172.16.23.0/24`, `172.16.24.0/24` are now represented by `172.16.16.0/20`.

In the same manner, the routes towards `192.168.20.0/24`, `192.168.21.0/24`, `192.168.22.0/24`, `192.168.23.0/24`, `192.168.24.0/24` are now represented by `192.168.16.0/20`.

Case 2: Route Filtering

In addition to being able to summarise routes, they can also be filtered out in order to not be advertised outside of the area.

Consider the same setup as in Case 1. Instead of summarising the routes towards the networks `R1` and `R2` are connected to, they will be filtered out, since we do not want that routing information in any other areas of the OSPF network.

Configuration

The route filtering needs to be configured on the ABR between the area and the backbone:

```
ABR:/#> configure
ABR:/config/#> router
ABR:/config/router/#> ospf
ABR:/config/router/ospf/#> area 1
ABR:/config/router/ospf/area-0.0.0.1/#> range 172.16.16.0/20 not-advertise
ABR:/config/router/ospf/area-0.0.0.1/#> range 192.168.16.0/20 not-advertise
ABR:/config/router/ospf/area-0.0.0.1/#> leave
ABR:/#>
```

Resulting Routes

Without the route filtering, the routes on `R1` will look like this:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O   10.0.1.0/24 [110/10] is directly connected, vlan1, 00:12:01
C>* 10.0.1.0/24 is directly connected, vlan1
O>* 172.16.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:08:31
O>* 172.16.20.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.21.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.22.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.23.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 172.16.24.0/24 [110/30] via 10.0.1.2, vlan1, 00:03:46
O>* 192.168.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:08:31
O>* 192.168.20.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.21.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.22.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.23.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
O>* 192.168.24.0/24 [110/30] via 10.0.1.2, vlan1, 00:00:06
```

After the route filtering have been applied on the `ABR`, the routes on `R1` will look like this:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O   10.0.1.0/24 [110/10] is directly connected, vlan1, 02:55:46
C>* 10.0.1.0/24 is directly connected, vlan1
O>* 172.16.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:01:50
O>* 192.168.1.0/24 [110/20] via 10.0.1.2, vlan1, 00:01:50
```

As can be seen, none of the networks connected to `R2` and `R3` have any routes towards them propagated outside of their area.

4.9.7. OSPF Route Redistribution HowTo

About

This document aims to provide example OSPF use-cases where route redistribution is applied in order to distribute routes from other sources into the OSPF domain.

For an overview of all available configuration settings for OSPF refer to the configuration guide:

- OSPF Configuration Guide

Introduction

Currently support exist to redistribute other RIP networks, connected routes and static routes into the OSPF domain. In addition routers are also able to inject a default route (0.0.0.0/0) into the the OSPF domain.

Any router who performs route redistribution into the OSPF domain is known as a *Autonomous System Boundary Router* (ASBR).

> **NOTE**
>
> Remember, if stub areas are used and it is necessary to redistribute routing information directly into that area, the type must be changed to NSSA. This is because a stub area cannot harbour any ASBRs.

The following example use-cases are presented in this document:

- Case 1: Redistribute RIP
- Case 2: Redistribute Connected Routes
- Case 3: Redistribute Static Routes
- Case 4: Redistribute Default Route

Case 1: Redistribute RIP

In Figure 1, an example setup consisting of one OSPF network and one RIP network is displayed. In order to redistribute the RIP network into the OSPF network, and vice versa, it needs to be configured on the `ASBR`.

```
. . . . . . . . . . . . . . . . . . .   . . . . . . . . . . . . . . . . . . .
.          OSPF Network         .   .          RIP Network          .
.                               .   .                               .
.  .-------.                    .   .                    .-------.  .
.  |       |                    .   .                    |       |  .
.  |  R1   |                    .   .                    |  R4   |  .
.  |       |                    .   .                    |       |  .
.  '---+---'                    .   .                    '---+---'  .
.    .2|                        .   .                        |.2    .
.      | 192.168.2.0/24         .   .          172.16.2.0/24 |      .
.    .1|                        .   .                        |.1    .
.  .---+---.                  .-------.                  .---+---.  .
.  |       |.1              .2|       |.2              .1|       |  .
.  |  R2   +------------------+ ASBR  +------------------+  R5   |  .
.  |       |  192.168.1.0/24  |       |  172.16.1.0/24   |       |  .
.  '---+---'                  '-------'                  '---+---'  .
.    .1|                        .   .                        |.1    .
.      | 192.168.3.0/24         .   .          172.16.3.0/24 |      .
.    .2|                        .   .                        |.2    .
.  .---+---.                    .   .                    .---+---.  .
.  |       |                    .   .                    |       |  .
.  |  R3   |                    .   .                    |  R6   |  .
.  |       |                    .   .                    |       |  .
.  '-------'                    .   .                    '-------'  .
.                               .   .                               .
. . . . . . . . . . . . . . . . . . .   . . . . . . . . . . . . . . . . . . .
```

Figure 1: Example of an OSPF network and a RIP network distributing its routes into each others domains.

When redistributed any destination will be reachable as if they were part of the same type of domain.

Configure

For this example we assume that both the OSPF network and the RIP network has been configured with the relevant internal networks. Refer to this example for basic OSPF network configuration, and to this for RIP network configuration.

Redistribution will be configured for both the OSPF and RIP networks, to redistribute each other.

ASBR

First we configure OSPF to redistribute the RIP network into its own domain:

```
ASBR:/#> configure
ASBR:/config/#> router
ASBR:/config/router/#> ospf
ASBR:/config/router/ospf/#> redistribute rip
ASBR:/config/router/ospf/#> leave
ASBR:/#>
```

Then we can also configure the RIP network to redistribute the OSPF network into its own domain:

```
ASBR:/#> configure
ASBR:/config/#> router
ASBR:/config/router/#> rip
ASBR:/config/router/rip/#> redistribute ospf
ASBR:/config/router/rip/#> leave
ASBR:/#>
```

> **NOTE**
>
> It may be necessary to also redistribute connected routes to be able to reach the networks `192.168.1.0/24` and `172.16.1.0/24` across the domains, because they are directly connected to the ASBR. Directly connected routes have administrative distance of 0 and will be selected above all other routes by default, so the RIP or OSPF route for those networks will not be active.

Routes

When redistributed we should be able to see routes to all parts of the OSPF domain, along with routes towards the networks present in the RIP domain. As an example, the routes on `R1`, in the OSPF domain, looks like this:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O>* 172.16.1.0/24 [110/20] via 192.168.2.1, vlan1, 00:11:06
O>* 172.16.2.0/24 [110/20] via 192.168.2.1, vlan1, 00:10:58
O>* 172.16.3.0/24 [110/20] via 192.168.2.1, vlan1, 00:10:58
O>* 192.168.1.0/24 [110/20] via 192.168.2.1, vlan1, 00:31:25
O   192.168.2.0/24 [110/10] is directly connected, vlan1, 00:36:25
C>* 192.168.2.0/24 is directly connected, vlan1
O>* 192.168.3.0/24 [110/20] via 192.168.2.1, vlan1, 00:31:25
```

In the same manner, the routes on `R4`, in the RIP domain, looks like this:

```
R4:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

R>* 172.16.1.0/24 [120/2] via 172.16.2.1, vlan1, 00:28:58
C>* 172.16.2.0/24 is directly connected, vlan1
R>* 172.16.3.0/24 [120/2] via 172.16.2.1, vlan1, 00:28:58
R>* 192.168.1.0/24 [120/3] via 172.16.2.1, vlan1, 00:12:06
R>* 192.168.2.0/24 [120/3] via 172.16.2.1, vlan1, 00:23:26
R>* 192.168.3.0/24 [120/3] via 172.16.2.1, vlan1, 00:23:26
```

Case 2: Redistribute Connected Routes

The case presented in Figure 2, shows a very simple example where a connected route is redistributed into an OSPF domain.

```
. . . . . . . . . . . . . . . . . . .
.          OSPF Network             .
.                                   .
.  .-------.                  .-------.                  .-------.
.  |       |.2              .1|       |.1              .2|       |
.  |  R1   +------------------+ ASBR  +------------------+  R2   |
.  |       |  192.168.1.0/24  |       |  172.16.1.0/24   |       |
.  '-------'                  '-------'                  '-------'
.                                   .
.                                   .
. . . . . . . . . . . . . . . . . . .
```

Figure 2: Example of an OSPF network redistributing a connected route into the OSPF domain.

> **NOTE**
>
> Remember that the router `R2` also needs to have a route towards the `ASBR` if it is to be reachable from inside the OSPF network. It could, for instance, be configured with a static route or a default route towards the `ASBR`.

Configure

As in the previous case, we assume that the OSPF network itself is already configured as intended. Configuring a router to redistribute a connected route will also turn it into a ASBR.

ASBR

The router is configured to redistribute connected routes in the following manner:

```
ASBR:/#> configure
ASBR:/config/#> router
ASBR:/config/router/#> rip
ASBR:/config/router/rip/#> redistribute connected
ASBR:/config/router/rip/#> leave
ASBR:/#>
```

Now routes to all connected links will be redistributed into the OSPF domain. In this example case we only have one connected link, but keep in mind that all connected links will be redistributed.

Routes

If the active routes are observed on `R1`, inside the OSPF domain, a route towards the connected network can be observed:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O>* 172.16.1.0/24 [110/20] via 192.168.1.1, vlan1, 00:23:47
O   192.168.1.0/24 [110/10] is directly connected, vlan1, 00:25:23
C>* 192.168.1.0/24 is directly connected, vlan1
```

Case 3: Redistribute Static Routes

In Figure 3, an example case is presented where an OSPF network is connected to some external network. We want some of these networks to be accessible from within the OSPF domain. The ASBR have static routes configured to reach these networks that we want to redistribute into the OSPF domain.

```
                                                                  Networks:
                                        172.16.10.0/24
. . . . . . . . . . . . . . . . . . .                             172.16.20.0/24
.          OSPF Network             .                             172.16.30.0/24
.                                   .
.  .-------.                  .-------.               .-------.     .--.-.
.  |       |.2              .1|       |.1           .2|       |    ( (    )
.  |  R1   +------------------+ ASBR  +---------------+  R2   +---(_,  \ ) ,_)
.  |       |  192.168.1.0/24  |       | 172.16.1.0/24 |       |     '-'--`--'
.  '-------'                  '-------'               '-------'  External Network
.                                   .
.                                   .
. . . . . . . . . . . . . . . . . . .
```

Figure 3: Example of an OSPF network redistributing static routes into the OSPF domain.

Configure

As before, we assume that the OSPF network itself is already configured as it is supposed to be. We will configure the static routes and their redistribution on the ASBR.

ASBR

First we add the static routes towards the intended networks, located in the connected external network, through `172.16.1.2`.

```
ASBR:/#> configure
ASBR:/config/#> ip
ASBR:/config/ip/#> route 172.16.10.0/24 172.16.1.2
ASBR:/config/ip/#> route 172.16.20.0/24 172.16.1.2
ASBR:/config/ip/#> route 172.16.30.0/24 172.16.1.2
ASBR:/config/ip/#> leave
ASBR:/#>
```

When the static routes have been set up, OSPF can be configured to redistribute those routes in the following way:

```
ASBR:/#> configure
ASBR:/config/#> router
ASBR:/config/router/#> rip
ASBR:/config/router/rip/#> redistribute static
ASBR:/config/router/rip/#> leave
ASBR:/#>
```

Routes

Looking at the routing table for `R1` we can see that the static routes have been distributed through the OSPF domain:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O>* 172.16.10.0/24 [110/20] via 192.168.1.1, vlan1, 00:00:01
O>* 172.16.20.0/24 [110/20] via 192.168.1.1, vlan1, 00:00:01
O>* 172.16.30.0/24 [110/20] via 192.168.1.1, vlan1, 00:00:01
O   192.168.1.0/24 [110/10] is directly connected, vlan1, 00:07:27
C>* 192.168.1.0/24 is directly connected, vlan1
```

Case 4: Redistribute Default Route

Distributing a default route cannot be done by redistributing a static route, it utilises its own command.

```
. . . . . . . . . . . . . . . . . . . .
.          OSPF Network               .
.                                     .
.  .-------.                  .-------.                  .-------.
.  |       |.2              .1|       |.1              .2|       |
.  |  R1   +------------------+ ASBR  +------------------+  GW   |
.  |       |  192.168.1.0/24  |       |  172.16.1.0/24   |       |
.  '-------'                  '-------'                  '-------'
.                                     .
.                                     .
. . . . . . . . . . . . . . . . . . . .
```

Figure 4: Example of an OSPF network distributing a default route into the OSPF domain.

The `ASBR` have a default route towards the `GW` router. The `ASBR` in turn injects a default route into the OSPF network that points to itself.

Configure

Same as before, we assume that the OSPF network itself is already configured as it is supposed to be. We configure the `ASBR` to advertise a default route into the OSPF domain.

ASBR

Configure OSPF to advertise a default route into the OSPF domain:

```
ASBR:/#> configure
ASBR:/config/#> router
ASBR:/config/router/#> ospf
ASBR:/config/router/ospf/#> distribute-default
ASBR:/config/router/ospf/#> leave
ASBR:/#>
```

> **NOTE**
>
> A default route will not be advertised by OSPF if it is not already present in the routing table. If the `always` keyword is added, to the `distribute-default` command, a default route will be advertised even if it is not present in the routing table.

Routes

When the a default route is being advertised the routing table on `R1` will look like this:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

O>* 0.0.0.0/0 [110/10] via 192.168.1.1, vlan1, 00:20:26
O   192.168.1.0/24 [110/10] is directly connected, vlan1, 00:20:37
C>* 192.168.1.0/24 is directly connected, vlan1
```

4.9.8. Policy-Based Routing (PBR)

About

This document aims to provide some example use cases and example scenarios how Policy-Based Routing (PBR) could be utilised to solve specific routing situations.

> **SEE ALSO**
>
> Related Documentation
> - Policy-Based Routing (PBR) Configuration Guide

Introduction

Utilisation of PBR allows an operator to configure routing setups that can perform decisions on additional criteria, rather than it just being based on the destination address, as with regular routes. This can allow for more agile routing solutions and a toolkit to solve specific situations.

The examples presented in this document are intended to provide ideas and inspiration as to how PBR can be applied and utilised, not necessarily general solutions applicable in every situation.

> **SEE ALSO**
>
> Use cases presented in this document
> - Case 1: Source-Based Routing
> - Case 2: Protocol-Based Routing
> - Case 3: Locally Originated Traffic

Case 1: Source-Based Routing

In regular routing the next hop is decided based on the destination IP address of a specific frame. However, situations may arise where it could be beneficial to base routing decisions on the source IP instead, performing source-based routing. There exist several reasons why this could be beneficial in different situations, here are just a few examples:

- Having specific ISPs handle frames from a select group of hosts
- Load-balancing of traffic towards different ISPs
- Prioritize traffic originating from specific hosts on higher-bandwidth links

In this use case scenario, policy-base routing will be utilised in order to direct traffic from different hosts towards different ISPs depending on their source IP, resulting in source-based routing.

> **TIP**
>
> It is possible to configure a policy-based route to perform source-based routing only for specific destination addresses. This can be done by combining the various match options when configuring the PBR instance, see this for all match options.

```
                         .--.-.
                        ( (    )__
                       (_,  \ ) ,_)  Internet
                         '-'--`--'
                          |     |
                          |     |
                    .-------'     '-------.
                    |                     |
                    |                     |
              .----+----.           .----+----.
              |         |           |         |
              |  ISP1   |           |  ISP2   |
              |         |           |         |
              '----+----'           '----+----'
               .99 |                     | .99
                   |                     |
     172.16.1.0/24 '-------.     .-------' 172.16.2.0/24
                          |     |
                       .1 |     | .1
                        .-+-----+-.
                        |         |
                        |   R1    |
                        |         |
                        '----+----'
                             | vlan1
                             |
                             |
                        .----+----.
     192.168.1.0/24     |         |      192.168.2.0/24
      .------+------+----+ Switch  +----+------+------.
      |      |      |    |         |    |      |      |
      |      |      |    '---------'    |      |      |
      |      |      |                   |      |      |
    .-+--. .-+--. .-+--.             .--+-. .--+-. .--+-.
    | H1 | | H2 | | H3 |             | H4 | | H5 | | H6 |
    '----' '----' '----'             '----' '----' '----'
```

Figure 1: Example of a setup with two different ISPs connected to the router R1. In addition, R1 is also connected to a switch with a number of hosts connected on two different VLANs.

The example topology presented in **Figure 1**, show a setup that have two different ISPs `ISP1` and `ISP2` that are both connected to the router `R1`, on the networks `172.16.1.0/24` and `172.16.2.0/24` respectively. Also connected to `R1` is a switch with a number of different hosts connected to it.

The main focus for this HowTo will be the configuration of router `R1`. The intent is to configure it so that any traffic originating from the network `192.168.1.0/24` is routed towards `ISP1` and any traffic originating from `192.168.2.0/24` is routed towards `ISP2`.

When this has been configured it should result in traffic originating from the hosts `H1`, `H2` and `H3` to be routed through `ISP1`. Similarly, the hosts `H4`, `H5` and `H6` should be routed through `ISP2`.

Configuration

The policy-based routes are configured on router `R1`. We assume that all the interfaces have already been configured in a suitable manner. The important thing for the sake of the example, is that vlan1 have been configured as the interface connected to the downstream switch.

First we configure the policy-based route responsible for routing the frames with a source within `192.168.1.0/24` towards `ISP1` located at `172.16.1.99`:

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> policy-route 1
R1:/config/ip/policy-route-1/#> match saddr 192.168.1.0/24
R1:/config/ip/policy-route-1/#> in-iface vlan1
```

```
R1:/config/ip/policy-route-1/#> next-hop 172.16.1.99
R1:/config/ip/policy-route-1/#> end
R1:/config/ip/#>
```

Next we configure the second policy route, responsible for handling routing of frames with sources within `192.168.2.0/24` towards `ISP2` located at `172.16.2.99`:

```
R1:/config/ip/#> policy-route 2
R1:/config/ip/policy-route-2/#> match saddr 192.168.2.0/24
R1:/config/ip/policy-route-2/#> in-iface vlan1
R1:/config/ip/policy-route-2/#> next-hop 172.16.2.99
R1:/config/ip/policy-route-2/#> end
R1:/config/ip/#>
```

If we check the overall policy routing configuration it should now look something like the following:

```
R1:/config/ip/#> show policy-route
ID   ENA  PREF  IN-IFACES   MATCH     VALUE              NEXT-HOP
1    Yes  Auto  vlan1       saddr:    192.168.1.0/24     172.16.1.99
2    Yes  Auto  vlan1       saddr:    192.168.2.0/24     172.16.2.99
```

If everything looks like expected, leave the configuration context to apply to configuration:

```
R1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#>
```

Status

When the policy routes have been configured they should be observable using the following command:

```
R1:/#> show ip policy-route
PRIO  SADDR           DADDR  SPORT  DPORT  IN-IFACE  NEXT-HOP
300   192.168.1.0/24  -      -      -      vlan1     172.16.1.99
301   192.168.2.0/24  -      -      -      vlan1     172.16.2.99
```

If the `NEXT-HOP` is reports `Unreachable` the link towards the next-hop may be down. It could also be the case that the network have not been configured correctly, since if there is no network in the same subnet as the next hop, the route will not be active.

> **NOTE**
>
> The status of the policy routes should be viewed using the `show ip policy-route` command, they will not be presented when using the regular `show ip route` command.

Case 2: Protocol-Based Routing

Imagine in this specific scenario that we want to route all WEB related traffic in direction towards a particular ISP, while all the other traffic is routed to another. We can achieve this by utilising PBR and perform matching on the port numbers used by the protocols.

```
                          .--.-.
                         ( (    )__
                        (_,  \ ) ,_)  Internet
                          '-'--`--'
                            |     |
                            |     |
                     .-------'     '-------.
            .-->     |                     |     <--.
            |        |                     |        |
Other       |   .----+----.           .----+----.   | HTTP/HTTPS
Traffic     |   |         |           |         |   | Traffic
            |   |  ISP1   |           |  ISP2   |   |
            |   |         |           |         |   |
            |   '----+----'           '----+----'   |
                 .99 |                     | .99
                     |                     |
   172.16.1.0/24 '-------.     .-------' 172.16.2.0/24
                            |     |
                        .1 |     | .1
                         .-+-----+-.
                         |         |
                         |   R1    |
                         |         |
                         '----+----'
                              | vlan1
                              |
                              |
                          .--.-.
                         ( (    )__
                        (_,  \ ) ,_)  LAN
                          '-'--`--'
```

Figure 2: Example of a setup with two different ISPs connected to the router R1. Where R1 is configured to send all the HTTP/HTTPS request traffic towards ISP2 and the rest towards ISP1.

The example scenario presented in **Figure 2** shows two different ISPs, `ISP1` and `ISP2`, connected to the router `R1`. The goal is that we want any HTTP and HTTPS traffic originating from the LAN network to be routed towards `ISP2`, located on next hop `172.16.2.99`. Any other traffic from the LAN should be sent towards `ISP1` located at the next hop `172.16.1.99`.

To achieve this desired behaviour a regular default gateway is configured on `R1` directing traffic towards `ISP1`. Afterwards, a policy-based route is configured on `R1` that matches on the destination ports, **80** for HTTP and **443** for HTTPS, with `ISP2` as the next hop address located at `172.16.2.99`.

This specific example focuses on requests originating somewhere on the LAN, for a web server located somewhere on the internet.

> **NOTE**
>
> Configuring the PBR in this way would only enforce outgoing traffic, as in trying to access a WEB page from the LAN located on the internet, since we are matching on the destination port.
>
> Say we would want to access a web server on the LAN from the internet, but retain the same routing, we would also need to add a similar rule that matches on source port instead of destination.

Configuration

The policy-based routes are configured on router `R1`. We assume that all the interfaces have already been configured in a suitable manner.

First we configure a regular default route towards `ISP1` that should handle all the other traffic:

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> route default 172.16.1.99
R1:/config/ip/#>
```

Now we can configure the policy-based routes. In this case we will need to configure two separate instances, one to handle HTTP (port 80), and another to handle HTTPS (port 443). We start with the configuration of the HTTP matching route:

```
R1:/config/ip/#> policy-route 1
R1:/config/ip/policy-route-1/#> match dport 80
R1:/config/ip/policy-route-1/#> in-iface vlan1
R1:/config/ip/policy-route-1/#> next-hop 172.16.2.99
R1:/config/ip/policy-route-1/#> end
R1:/config/ip/#>
```

Now we configure a second policy-based route instance for covering HTTPS traffic:

```
R1:/config/ip/#> policy-route 2
R1:/config/ip/policy-route-2/#> match dport 443
R1:/config/ip/policy-route-2/#> in-iface vlan1
R1:/config/ip/policy-route-2/#> next-hop 172.16.2.99
R1:/config/ip/policy-route-2/#> end
R1:/config/ip/#>
```

The configuration should now look something like this:

```
R1:/config/ip/#> show policy-route
ID   ENA  PREF  IN-IFACES   MATCH     VALUE                NEXT-HOP
1    Yes  Auto  vlan1       dport:    80                   172.16.2.99
2    Yes  Auto  vlan1       dport:    443                  172.16.2.99
R1:/config/ip/#>
```

If everything looks okay we can leave the configuration context, in order to apply the configuration:

```
R1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#>
```

Status

If everything has been set up correctly, the policy routes should now be active on the device:

```
R1:/#> show ip policy-route
PRIO  SADDR  DADDR  SPORT  DPORT  IN-IFACE  NEXT-HOP
300   -      -      -      80     vlan1     172.16.2.99
301   -      -      -      443    vlan1     172.16.2.99
```

If the `NEXT-HOP` reports `Unreachable`, the link towards the next-hop may be down. It could also be the case that the network have not been configured correctly, since if there is no network in the same subnet as the next hop, the route will not be active.

Case 3: Locally Originated Traffic

The scenario presented in this case serve to provide an example how a policy route can be setup to apply only on frames originating from the device itself, essentially direct access to the router itself.

```
                  .---------.                      .---------.
                  |         |     172.16.1.0/24    |         |
                  |   R1    +----------------------+   R2    |
                  |         | .10              .20 |         |
                  '----+----'                      '----+----'
                       | .10                        .20 |
192.168.10.0/24 |                                 | 192.168.20.0/24
                       | .1                          .2 |
                     .-+--.                          .-+--.
                     | H1 |                          | H2 |
                     '----'                          '----'

                GW: 192.168.10.10               GW: 192.168.20.20
```

Figure 3: Basic setup with two routers and two hosts.

Based on the topology presented in **Figure 3**, imagine that we for some reason would like both hosts to be able to reach both of the routers, but not each other. In order to achieve this we could use a simple policy-based route configuration, one on each of the routers.

However, first it is good to note why we may not be able to achieve this behaviour using regular static routes. For instance, if we want `H1` to be able to reach `R2` using regular static routes, we could configure `172.16.1.10` as the next hop for `192.168.10.0/24` on `R2`. Doing this would ensure that we could reach the net of host `H1` from `R2`.

In addition, we also want `H2` to be able to reach `R1`, so we add corresponding routes on `R1`, with `172.16.1.20` being the next hop for `192.168.20.0/24`.

At this point we would be able to reach `R2` from `H1` and correspondingly, `R1` from `H2`. However, the two hosts would also be able to reach each other, but we only wanted them to be able to reach the routers themselves.

Now, instead of adding static routes we want to configure policy-based routes on each of the routers, in a way that only allow the route to be utilised for traffic originating from the router itself. It will be setup so that `H2` can reach `R1` on `172.16.1.10` and so that `H1` can reach `R2` on `172.16.1.20`.

Configuration

Each of the two routers, `R1` and `R2`, needs to be configured with a policy route each. We also assume for this scenario that all the relevant interfaces have already been configured.

Each of the policy routes are intended to be usable for frames originating from the device itself. This can be achieved by specifying the interface `lo` as the `in-iface` for the policy route. In addition, the match rule itself will be configured to be the `saddr` of the device itself. So for `R1` the `saddr` will be set to 172.16.1.10 and for `R2` it will be set to 172.16.1.20.

R1

The policy-based route instance is configured on the device in the following manner:

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> policy-route 1
R1:/config/ip/policy-route-1/#> match saddr 172.16.1.10
R1:/config/ip/policy-route-1/#> in-iface lo
```

```
R1:/config/ip/policy-route-1/#> next-hop 172.16.1.20
R1:/config/ip/policy-route-1/#> end
R1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#>
```

This policy-based route will ensure that any traffic directed towards `R1` on `172.16.1.10`, will have the response routed towards `172.16.1.20`.

R2

The policy-based route instance is configured on the device in the following manner:

```
R2:/#> configure
R2:/config/#> ip
R2:/config/ip/#> policy-route 1
R2:/config/ip/policy-route-1/#> match saddr 172.16.1.20
R2:/config/ip/policy-route-1/#> in-iface lo
R2:/config/ip/policy-route-1/#> next-hop 172.16.1.10
R2:/config/ip/policy-route-1/#> end
R2:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R2:/#>
```

This policy-based route will ensure that any traffic directed towards `R2` on `172.16.1.20`, will have the response routed towards `172.16.1.10`.

Status

If both the routers have been configured correctly, the active policy-based routes should look something like this, starting with `R1`:

```
R1:/#> show ip policy-route
PRIO  SADDR           DADDR  SPORT  DPORT  IN-IFACE  NEXT-HOP
300   172.16.1.10/32  -      -      -      lo        172.16.1.20
```

For `R2` it should look something like this:

```
R2:/#> show ip policy-route
PRIO  SADDR           DADDR  SPORT  DPORT  IN-IFACE  NEXT-HOP
300   172.16.1.20/32  -      -      -      lo        172.16.1.10
```

Verify the Behaviour

At this point it should be possible for `H1` to reach `R2` on 172.16.1.20. We can test this by simply using ping:

```
H1:/#> ping count 3 172.16.1.20
Press Ctrl-C to abort PING 172.16.1.20 (172.16.1.20): 56 data bytes
64 bytes from 172.16.1.20: seq=0 ttl=63 time=3.379 ms
64 bytes from 172.16.1.20: seq=1 ttl=63 time=3.731 ms
64 bytes from 172.16.1.20: seq=2 ttl=63 time=3.655 ms

--- 172.16.1.20 ping statistics ---
```

```
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.379/3.588/3.731 ms
H1:/#>
```

However, we should not be able to reach anything in the 192.168.20.0/24 subnet, this includes `R2` on that address and `H2`:

```
H1:/#> ping count 3 192.168.20.20
Press Ctrl-C to abort PING 192.168.20.20 (192.168.20.20): 56 data bytes

--- 192.168.20.20 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
H1:/#>
H1:/#> ping count 3 192.168.20.2
Press Ctrl-C to abort PING 192.168.20.2 (192.168.20.2): 56 data bytes

--- 192.168.20.2 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
H1:/#>
```

The same verification could be done from `H1`, checking that `R1` on 172.16.1.10.

4.9.9. Equal-Cost Multi-Path Routing (ECMP) HowTo

About

This document aims to provide some example use cases and scenarios, how Equal-Cost Multi-Path (ECMP) routing can be applied and how it may behave in different situations.

> **SEE ALSO**
>
> Related Documentation
>
> - Static Routing
> - Equal-Cost Multi-Path Routing (ECMP) Overview
> - OSPF

Introduction

ECMP can allow for routes to be configured so that multiple paths can exist for the same destination, simultaneously. Meaning that different flows, aimed for the same destination, can potentially be routed on different paths. This can be useful for a number of different reasons, for example:

- Load balancing of traffic flows, bound for the same destination, over different paths.
- Distribution of frames on multiple links for better bandwidth utilisation.
- Allow traffic flows to select an alternate path on potential direct link failure, without the need for waiting on potential routing protocol updates.

This document provide a few different use cases and scenarios to help describe how ECMP can be applied and how it functions. They are intended to provide a basic understanding how they are used, so that they could possibly be applied to solve similar scenarios.

> **SEE ALSO**
>
> Use cases presented in this document
> - Case 1: Stateless Load Balancing
> - Case 2: Load Balancing and Redundant Routes

Case 1: Stateless Load Balancing

In this example use case, we aim to provide a router configuration that can provide some basic load balancing using ECMP, by providing two different default gateways. Refer to the example in **Figure 1** below, for an example setup.

```
                          .--.-.
                         ( (    )__
                        (_,  \ ) ,_)  Internet
                          '-'--`--'
                            |     |
                            |     |
                     .-------'     '-------.
                     |                     |
                     |                     |
               .----+----.           .----+----.
               |         |           |         |
               |  ISP1   |           |  ISP2   |
               |         |           |         |
               '----+----'           '----+----'
                .99 |                     | .99
                    |                     |
        172.16.1.0/24 '-------.     .-------' 172.16.2.0/24
                            |     |
                        .1 |     | .1
                         .-+-----+-.
                         |         |
                         |   R1    |
                         |         |  GW: 172.16.1.99
                         '----+----'      172.16.2.99
                              |
                              |
                          .--.-.
                         ( (    )__
                        (_,  \ ) ,_)  Lan
                          '-'--`--'
```

Figure 1: Load balancing over two different ISPs, by utilising two same-cost default gateways on R1.

On the device `R1` we have simply configured two different default routes, with the same cost, each pointing to one of the ISPs as the next hop gateway. The intention is that traffic flows from devices in the `Lan` will be distributed as much as possible between `ISP1` and `ISP2`.

> **NOTE**
>
> This could of course be scaled up by having even more next hops towards additional devices. This example uses only two next hops for the sake of simplicity.

Configure

No specific configuration in regards to ECMP needs to be made on router `R1`, we simply configure two default routes towards two different next hop addresses:

> **NOTE**
>
> Remember to **NOT** configure the routes with different distances.

```
R1:/#> configure ip
R1:/config/ip/#> route default 172.16.1.99
R1:/config/ip/#> route default 172.16.2.99
R1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#>
```

Now that the routes have been configured, the routing table should look something like the following:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 0.0.0.0/0 [1/0] via 172.16.1.99, vlan1, weight 1, 00:00:03
  *                 via 172.16.2.99, vlan2, weight 1, 00:00:03
C>* 172.16.1.0/24 is directly connected, vlan1, 00:01:32
C>* 172.16.2.0/24 is directly connected, vlan2, 00:01:32
```

Notice that the `0.0.0.0/0` (default) destination have two next hop addresses listed for it. Traffic bound for the Internet that enters the router `R1`, from any devices located in the `Lan` network, should now be provided a next hop towards either `ISP1` or `ISP2`, on a per flow level.

> **NOTE**
>
> The effectiveness of the load balancing depends how the hash mapping is performed. It is not a guarantee that there will be an exact 50/50 split across the two ISPs.

Case 2: Load Balancing and Redundant Routes

This use case will provide an example how we can configure multiple routers with ECMP routes to provide load balancing, while also providing several redundant routes. An example of this type of setup is provided in **Figure 2** below.

```
   Network-A                                                    Network-B
   .--.-.                                                        .--.-.
  ( (    )__                                                    ( (    )__
 (_,  \ ) ,_)                                                   (_,  \ ) ,_)
   '-'--`--'                                                     '-'--`--'
      |                 .---------.          .---------.                |
      |          NET-1  |         |   NET-3  |         |  NET-5         |
172.16.1.0/24    .-------+   R2    +----------+   R4    +-------.    172.16.2.0/24
      |        .´      .2|         |.2       .4|         |.4      `.        |
      |      .´          '----+----+           +----+----'          `.      |
 .----+----+´                |     `.       .´     |                 `+----+----.
 |         |.1               |       `.   .´       |                .6|         |
 |   R1    |            NET-9 |         `.´         | NET-10           |   R6    |
 |         |.1               |   NET-7.´ `.NET-8   |                .6|         |
 '---------+.                |      .´     `.      |                 .+---------'
             `.          .----+----+´         `+----+----.          .´
               `.      .3|         |.3       .5|         |.5      .´
                 `-------+   R3    +----------+   R5    +-------´
                  NET-2  |         |   NET-4  |         |  NET-6
                         '---------'          '---------'

                  Networks ==============================================
                  NET-1: 192.168.10.0/24     NET-6:  192.168.60.0/24
                  NET-2: 192.168.20.0/24     NET-7:  192.168.70.0/24
                  NET-3: 192.168.30.0/24     NET-8:  192.168.80.0/24
                  NET-4: 192.168.40.0/24     NET-9:  192.168.90.0/24
                  NET-5: 192.168.50.0/24     NET-10: 192.168.100.0/24
                  ====================================================
```

Figure 2: Routers with redundant connections, in order to achieve load balancing and route redundancy for traffic between Network-A and Network-B.

In the setup above, we want to ensure connectivity between the two networks `Network-A` and `Network-B`. On each of the routers routes are configured with the same distance, so that they are of equal cost, so that ECMP routing can be utilised. Because of the redundant paths, link failures should not impact the traffic sent between the two networks `Network-A` and `Network-B`.

With this setup many paths will exist at the same time that could be utilised, depending on how the multi-path hash is calculated for each traffic flow. Each of the routers will technically be able to send the traffic on any equal cost route. As an example, the following paths are of the same cost and could all be selected for different flows:

- `Network-A` -> `R1` -> `R2` -> `R4` -> `R6` -> `Network-B`
- `Network-A` -> `R1` -> `R2` -> `R5` -> `R6` -> `Network-B`
- `Network-A` -> `R1` -> `R3` -> `R5` -> `R6` -> `Network-B`
- `Network-A` -> `R1` -> `R3` -> `R4` -> `R6` -> `Network-B`

> **NOTE**
>
> Any route specified between `R2` and `R3`, and vice versa between `R4` and `R5`, should be configured with a higher `distance`, so that the cost will not be the same as the other routes. The reason for this is that we do not want any of those routes to be part of the ECMP routing, since that path requires an extra hop. Rather, that route should be utilised of the other is not usable, because of link failure.

In most link failures scenarios depending on the number of link failures and where they are situated, connectivity should still be possible between the two networks. However, depending on where the link failure is, it is possible that some traffic flows may use a path that is not the most efficient. As an example, if the link for `NET-3` and `NET-8` goes down, both of the following paths could be valid:

- `Network-A` -> `R1` -> `R3` -> `R5` -> `R6` -> `Network-B`
- `Network-A` -> `R1` -> `R2` -> `R3` -> `R5` -> `R6` -> `Network-B`

As we can see, in that specific scenario some traffic flow could use a path that has an extra hop. This should mostly be a problem if we use static routing, with a dynamic routing protocol it should select the better path, once the updated routing information have propagated to each of the routers.

> **WARNING**
>
> We could still end up in a situation with this setup where we can route traffic to a device that will not be able to forward it further, if we experience multiple link failures. As an example, if the links for networks `NET-3`, `NET-7` and `NET-10` on router `R4` are down, router `R6` could still forward traffic towards it, since it cannot know that those links are down, when we use static routing. If we would use a dynamic routing protocol this should not be an issue.

> **SEE ALSO**
>
> Configuration Examples
> - Configure With Static Routes
> - Configure With OSPF

Configure With Static Routes

This is an example how we can configure the devices with static routes. It is somewhat more cumbersome to do than to simply configure either RIP or OSPF. Nevertheless, it is more and more difficult the larger a topology becomes. However, since the topology in this example is not particularly extensive it can be a useful as an example.

For this configuration it is assumed that all of the interfaces have already been configured as intended. We need to configure each of the routers with the necessary static routes to be able to reach `Network-A` at `172.16.1.0/24` and `Network-B` at `172.16.2.0/24`. Since we want to utilise ECMP routing, all the possible next hops for each destination IP address must be considered.

We need to be aware when we set up the routes, over the networks `NET-9` and `NET-10`, to use a higher distance value. Setting a higher distance value for those routes will increase the cost, so that they will not be considered equal as part of the ECMP routing. Because providing a next hop over those two networks will always result in an additional hop to reach any destination in either `Network-A` or `Network-B`. We only want those routes to be considered if the outer routes are unavailable, due to link failure for instance. In essence, they should serve as a backup path, and not be part of the regular multi-path distribution of traffic flows that will be the result of the ECMP routing.

R1 Configuration:

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> route 172.16.2.0/24 192.168.10.2
R1:/config/ip/#> route 172.16.2.0/24 192.168.20.3
R1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

C>* 172.16.1.0/24 is directly connected, vlan7, 02:29:32
S>* 172.16.2.0/24 [1/0] via 192.168.10.2, vlan1, weight 1, 00:29:57
  *                     via 192.168.20.3, vlan2, weight 1, 00:29:57
C>* 192.168.10.0/24 is directly connected, vlan1, 01:02:26
C>* 192.168.20.0/24 is directly connected, vlan2, 00:29:57
```

R2 Configuration:

```
R2:/#> configure
R2:/config/#> ip
R2:/config/ip/#> route 172.16.1.0/24 192.168.10.1
R2:/config/ip/#> route 172.16.2.0/24 192.168.30.4
R2:/config/ip/#> route 172.16.2.0/24 192.168.80.5
R2:/config/ip/#> route 172.16.1.0/24 192.168.90.3 10
R2:/config/ip/#> route 172.16.2.0/24 192.168.90.3 10
R2:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R2:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 172.16.1.0/24 [1/0] via 192.168.10.1, vlan2, weight 1, 01:04:50
S   172.16.1.0/24 [10/0] via 192.168.90.3, vlan4, weight 1, 01:28:08
S>* 172.16.2.0/24 [1/0] via 192.168.30.4, vlan1, weight 1, 00:31:51
  *                     via 192.168.80.5, vlan3, weight 1, 00:31:51
S   172.16.2.0/24 [10/0] via 192.168.90.3, vlan4, weight 1, 01:22:15
C>* 192.168.10.0/24 is directly connected, vlan2, 01:04:50
C>* 192.168.30.0/24 is directly connected, vlan1, 00:31:51
```

```
C>* 192.168.80.0/24 is directly connected, vlan3, 00:32:10
C>* 192.168.90.0/24 is directly connected, vlan4, 01:28:08
```

R3 Configuration:

```
R3:/#> configure
R3:/config/#> ip
R3:/config/ip/#> route 172.16.1.0/24 192.168.20.1 1
R3:/config/ip/#> route 172.16.2.0/24 192.168.40.5 1
R3:/config/ip/#> route 172.16.2.0/24 192.168.70.4 1
R3:/config/ip/#> route 172.16.1.0/24 192.168.90.2 10
R3:/config/ip/#> route 172.16.2.0/24 192.168.90.2 10
R3:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R3:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 172.16.1.0/24 [1/0] via 192.168.20.1, vlan1, weight 1, 00:39:17
S   172.16.1.0/24 [10/0] via 192.168.90.2, vlan4, weight 1, 01:34:24
S>* 172.16.2.0/24 [1/0] via 192.168.40.5, vlan2, weight 1, 00:39:15
  *                     via 192.168.70.4, vlan3, weight 1, 00:39:15
S   172.16.2.0/24 [10/0] via 192.168.90.2, vlan4, weight 1, 01:29:50
C>* 192.168.20.0/24 is directly connected, vlan1, 00:39:17
C>* 192.168.40.0/24 is directly connected, vlan2, 00:39:15
C>* 192.168.70.0/24 is directly connected, vlan3, 01:25:02
C>* 192.168.90.0/24 is directly connected, vlan4, 01:34:24
```

R4 Configuration:

```
R4:/#> configure
R4:/config/#> ip
R4:/config/ip/#> route 172.16.1.0/24 192.168.30.2 1
R4:/config/ip/#> route 172.16.1.0/24 192.168.70.3 1
R4:/config/ip/#> route 172.16.2.0/24 192.168.50.6 1
R4:/config/ip/#> route 172.16.2.0/24 192.168.100.5 10
R4:/config/ip/#> route 172.16.1.0/24 192.168.100.5 10
R4:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R4:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 172.16.1.0/24 [1/0] via 192.168.30.2, vlan2, weight 1, 00:41:40
  *                     via 192.168.70.3, vlan3, weight 1, 00:41:40
S   172.16.1.0/24 [10/0] via 192.168.100.5, vlan4, weight 1, 01:27:57
S>* 172.16.2.0/24 [1/0] via 192.168.50.6, vlan1, weight 1, 00:40:34
S   172.16.2.0/24 [10/0] via 192.168.100.5, vlan4, weight 1, 01:27:57
C>* 192.168.30.0/24 is directly connected, vlan2, 00:41:40
C>* 192.168.50.0/24 is directly connected, vlan1, 00:40:34
C>* 192.168.70.0/24 is directly connected, vlan3, 01:27:53
C>* 192.168.100.0/24 is directly connected, vlan4, 01:27:57
```

R5 Configuration:

```
R5:/#> configure
R5:/config/#> ip
R5:/config/ip/#> route 172.16.1.0/24 192.168.40.3 1
R5:/config/ip/#> route 172.16.1.0/24 192.168.80.2 1
R5:/config/ip/#> route 172.16.2.0/24 192.168.60.6 1
R5:/config/ip/#> route 172.16.2.0/24 192.168.100.4 10
R5:/config/ip/#> route 172.16.1.0/24 192.168.100.4 10

R5:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R5:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 172.16.1.0/24 [1/0] via 192.168.40.3, vlan1, weight 1, 00:43:08
  *                     via 192.168.80.2, vlan3, weight 1, 00:43:08
S   172.16.1.0/24 [10/0] via 192.168.100.4, vlan4, weight 1, 01:29:03
S   172.16.2.0/24 [10/0] via 192.168.100.4, vlan4, weight 1, 01:29:03
S>* 172.16.2.0/24 [1/0] via 192.168.60.6, vlan2, weight 1, 01:30:28
C>* 192.168.40.0/24 is directly connected, vlan1, 00:43:12
C>* 192.168.60.0/24 is directly connected, vlan2, 01:30:28
C>* 192.168.80.0/24 is directly connected, vlan3, 00:43:08
C>* 192.168.100.0/24 is directly connected, vlan4, 01:29:03
```

R6 Configuration:

```
R6:/#> configure
R6:/config/#> ip
R6:/config/ip/#> route 172.16.1.0/24 192.168.50.4 1
R6:/config/ip/#> route 172.16.1.0/24 192.168.60.5 1
R6:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R6:/#> show ip route
S - Static | C - Connected | K - Kernel route   | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 172.16.1.0/24 [1/0] via 192.168.50.4, vlan2, weight 1, 00:42:20
  *                     via 192.168.60.5, vlan1, weight 1, 00:42:20
C>* 172.16.2.0/24 is directly connected, vlan7, 01:56:12
C>* 192.168.50.0/24 is directly connected, vlan2, 00:42:20
C>* 192.168.60.0/24 is directly connected, vlan1, 01:31:09
```

Configure With OSPF

In this example we will configure the routers with OSPF, instead of static routes. The outcome in the end should mostly be the same, in terms of the ECMP route selection. In addition we should also be more resistant to ending up in the scenarios where the static routes could end up not working.

> **WARNING**
>
> This setup is supposed to replace the configuration in the example Configure With Static Routes above, not to be combined with it.

As in the previous example we assume that all of the relevant interfaces have already been configured. The configuration example will simply focus on the set up of OSPF on the routers. In

this example we will use a very simple OSPF configuration, the only thing we really need to do is to specify to OSPF what networks it should know about.

Only for router `R1` and `R6` will we need to set an additional configuration option, by using the `redistribute connected` command. Because we want OSPF to know about the networks `Network-A` and `Network-B` that are directly connected to `R1` and `R6` respectively, but exist outside of the OSPF domain. Thus, if we want to be able to route traffic between those two networks, their existence must be made aware to the OSPF routing domain. Otherwise no routes towards those destinations will be inserted in any of the routers routing tables.

> **NOTE**
>
> While using OSPF will ensure a more optimal path during link failures, the failover time could in some cases be a bit longer. The reason for this is that the routing information from the protocol must propagate throughout the network.

R1 Configuration:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> ospf
R1:/config/router/ospf/#> network 192.168.10.0/24
R1:/config/router/ospf/#> network 192.168.20.0/24
R1:/config/router/ospf/#> redistribute connected
R1:/config/router/ospf/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#>
```

R2 Configuration:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> ospf
R2:/config/router/ospf/#> network 192.168.10.0/24
R2:/config/router/ospf/#> network 192.168.30.0/24
R2:/config/router/ospf/#> network 192.168.90.0/24
R2:/config/router/ospf/#> network 192.168.80.0/24
R2:/config/router/ospf/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R2:/#>
```

R3 Configuration:

```
R3:/#> configure
R3:/config/#> router
R3:/config/router/#> ospf
R3:/config/router/ospf/#> network 192.168.20.0/24
R3:/config/router/ospf/#> network 192.168.90.0/24
R3:/config/router/ospf/#> network 192.168.70.0/24
R3:/config/router/ospf/#> network 192.168.40.0/24
R3:/config/router/ospf/#> leave
Applying configuration.
```

```
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R3:/#>
```

R4 Configuration:

```
R4:/#> configure
R4:/config/#> router
R4:/config/router/#> ospf
R4:/config/router/ospf/#> network 192.168.30.0/24
R4:/config/router/ospf/#> network 192.168.50.0/24
R4:/config/router/ospf/#> network 192.168.100.0/24
R4:/config/router/ospf/#> network 192.168.70.0/24
R4:/config/router/ospf/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R4:/#>
```

R5 Configuration:

```
R5:/#> configure
R5:/config/#> router
R5:/config/router/#> ospf
R5:/config/router/ospf/#> network 192.168.100.0/24
R5:/config/router/ospf/#> network 192.168.60.0/24
R5:/config/router/ospf/#> network 192.168.80.0/24
R5:/config/router/ospf/#> network 192.168.40.0/24
R5:/config/router/ospf/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R5:/#>
```

R6 Configuration:

```
R6:/#> configure
R6:/config/#> router
R6:/config/router/#> ospf
R6:/config/router/ospf/#> network 192.168.50.0/24
R6:/config/router/ospf/#> network 192.168.60.0/24
R6:/config/router/ospf/#> redistribute connected
R6:/config/router/ospf/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R6:/#>
```

Status and Verification

If we have managed to configure everything correctly we should now be able to send traffic between the two networks. We could do some basic verification by simply pinging some device from one network to the other:

```
root@Network-A-Host1:/home/admin # ping 172.16.2.99 -c 3
PING 172.16.2.99 (172.16.2.99): 56 data bytes
64 bytes from 172.16.2.99: seq=0 ttl=60 time=4.063 ms
64 bytes from 172.16.2.99: seq=1 ttl=60 time=7.465 ms
64 bytes from 172.16.2.99: seq=2 ttl=60 time=6.815 ms
```

```
--- 172.16.2.99 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 4.063/6.114/7.465 ms
```

In order to check what specific path we are taking to reach the target, we could utilise traceroute:

```
root@Host-A:/home/admin # traceroute 172.16.2.99
traceroute to 172.16.2.99 (172.16.2.99), 30 hops max, 46 byte packets
 1  172.16.1.1 (172.16.1.1)  1.268 ms  0.956 ms  0.914 ms
 2  192.168.10.2 (192.168.10.2)  2.167 ms  2.722 ms  1.618 ms
 3  192.168.70.4 (192.168.70.4)  2.787 ms  1.928 ms  1.722 ms
 4  192.168.60.6 (192.168.60.6)  3.757 ms  1.791 ms  1.287 ms
 5  172.16.2.99 (172.16.2.99)  1.682 ms  2.385 ms  2.022 ms
```

The traceroute output corresponds to the following hops:

- `Network-A-Host1` -> `R1` -> `R2` -> `R4` -> `R6` -> `Network-B-Host1`

When we perform a traceroute on another host located at 172.16.2.222, in this case, we can see the because of the ECMP routing we selected a different path:

```
root@Host-A:/home/admin # traceroute 172.16.2.222
traceroute to 172.16.2.222 (172.16.2.222), 30 hops max, 46 byte packets
 1  172.16.1.1 (172.16.1.1)  1.522 ms  0.852 ms  0.884 ms
 2  192.168.20.3 (192.168.20.3)  1.532 ms  1.306 ms  1.409 ms
 3  192.168.80.5 (192.168.80.5)  2.445 ms  1.319 ms  1.289 ms
 4  192.168.60.6 (192.168.60.6)  2.082 ms  1.702 ms  1.622 ms
 5  172.16.2.222 (172.16.2.222)  5.326 ms  4.561 ms  3.892 ms
```

The traceroute output corresponds to the following hops:

- `Network-A-Host1` -> `R1` -> `R3` -> `R5` -> `R6` -> `Network-B-Host2`

Link Failure

If we have a link failure traffic should still continue flow. As an example, when we ping a host located in `Network-B` at `172.16.2.99` from `Network-A` it takes the following route:

- `Network-A-Host1` -> `R1` -> `R2` -> `R4` -> `R6` -> `Network-B-Host1`

Then a link failure occurs for `NET-3`, between `R2` and `R4`, when this occurs the traffic will instead be routed the following path:

- `Network-A-Host1` -> `R1` -> `R2` -> `R5` -> `R6` -> `Network-B-Host1`

We now hop from `R2` to `R5` over `NET-8` instead. This can clearly be seen with traceroute:

```
root@Host-A:/home/admin # traceroute 172.16.2.99
traceroute to 172.16.2.99 (172.16.2.99), 30 hops max, 46 byte packets
 1  172.16.1.1 (172.16.1.1)  1.017 ms  0.776 ms  1.826 ms
 2  192.168.10.2 (192.168.10.2)  2.072 ms  2.215 ms  2.577 ms
 3  192.168.80.5 (192.168.80.5)  3.365 ms  1.871 ms  1.691 ms
 4  192.168.60.6 (192.168.60.6)  2.324 ms  2.509 ms  2.301 ms
 5  172.16.2.99 (172.16.2.99)  2.441 ms  2.976 ms  2.020 ms
```

4.9.10. Multiple Upstream Failover HowTo

About

This document will cover how we can achieve automatic failover in a setup that utilises multiple upstream providers. This can be useful if we want to increase the redundancy for our upstream connections.

In order to achieve this we will need to utilise a few different functionalities, including:

- Static unicast routing.
- Policy-Based Routing (PBR).
- Ping trigger, part of the alarm system.

> **SEE ALSO**
>
> Related Documentation
> - Static Routing
> - Policy-Based Routing (PBR)
> - Alarm

Introduction

For this example we imagine a topology where we have a Router connected to two different upstream providers. This is done to improve our redundancy, by having one of them being the primary and the other used as a backup, in case we lose connection on the first mentioned. In **Figure 1** below a representation of this scenario is presented.

```
                          198.19.20.21
                                |
                             .--.-.
                            ( (    )__
                           (_,  \ ) ,_)  Internet
                             '-'--`--'
                               |     |
                               |     |
                        .-------'     '-------.
          .-->          |                     |     <--.
          |             |                     |        |
Primary   |   .----+----.           .----+----.   | Secondary
Uplink    |   |         |           |         |   | Uplink
          |   |  ISP1   |           |  ISP2   |   |
          |   |         |           |         |   |
          |   '----+----'           '----+----'   |
               .99 |                     | .99
                   |                     |
   172.16.1.0/24 '-------.     .-------' 172.16.2.0/24
                         |     |
                       .1 |     | .1
      Interface: vlan1 .-+-----+-. Interface: vlan2
                        |         |
                        |   R1    |
                        |         |  GW: 172.16.1.99 distance 1
                        '----+----'      172.16.2.99 distance 10
                             |
                             |
                            .--.-.
                           ( (    )__
                          (_,  \ ) ,_)  Lan
                            '-'--`--'
```

Figure 1: The router R1, connected to two different upstream providers ISP1 and ISP2, in order to achieve redundancy, by having ISP1 serve as the primary and ISP2 the secondary.

As can be seen in **Figure 1**, the router `R1` is connected to the two upstream providers `ISP1` and `ISP2`. Two different default gateways are configured on `R1` towards `ISP1` at `172.16.1.99` and `ISP2` at `172.16.2.99` respectively. In order to ensure that the first route will serve as the primary, the `distance` of this route is configured to 1 and for the other it is set to 10.

> **NOTE**
>
> A route with a lower distance will be considered the better route. Therefore by adjusting this value we can specify what route will be selected first in this case.

Handle the Failover

The next step will be to address how the failover is supposed to be carried out, but first its good to understand why we need special handling for this. As it stands right now, the only way for the secondary route to become selectable is if the upstream interface, `vlan1` on `R1`, gets a direct link failure. Then the network that the default route is associated with will go down and the route will become inactive, thus the secondary route will now be selected instead.

However, if the link would become broken somewhere along the path between `ISP1` and our target destination, as shown in **Figure 2**, it would not be possible for `R1` to be aware of that in the current situation. So traffic will be routed from the `Lan` towards `ISP1` that will not be able to reach our intended destination.

```
                    198.19.20.21
                          |
                       .--.-.
                     ( (    )__
                    (_,  \ ) ,_)  Internet
                       '-'--`--'
                         |     |
                         |     |
Link          .-------'     '-------.
Failure ----> X                     |
              |                     |
          .----+----.           .----+----.
          |         |           |         |
          |  ISP1   |           |  ISP2   |
          |         |           |         |
          '----+----'           '----+----'
           .99 |                     | .99
              |                     |
172.16.1.0/24 '-------.     .-------' 172.16.2.0/24
                         |     |
                      .1 |     | .1
  Interface: vlan1  .-+-----+-. Interface: vlan2
                     |         |
                     |   R1    |
                     |         |
                     '----+----'
                          |
                          |
                       .--.-.
                     ( (    )__
                    (_,  \ ) ,_)  Lan
                       '-'--`--'
```

Figure 2: A link failure has occurred somewhere between ISP1 and our target destination.

In order to handle this the `ping trigger` functionality will be used. It will track an upstream destination, in this example we will check if the address `198.19.20.21` is reachable. Two different ping triggers will be used, one for each of the upstream interfaces `vlan1` and `vlan2`.

Subsequently, the ping trigger is attached to the route as a remote tracker. If the ping trigger is not able to reach its provided destination address, it will adjust the distance of the route to 255. When this happens, the route with the next best distance will be used instead, in this specific example it will be the route to our Secondary provider `ISP2`.

> **NOTE**
>
> When a route is configured with a distance of 255, it is considered infinite, and the route will not be used, i.e. it will not be active in the routing table.

Ensure Correct Upstream Interface is Used for Ping Trigger

It is important to ensure that the requests sent by the ping triggers egress on the correct associated upstream interface, regardless of the state of the routing table. Since what we want to keep track of in this case is that we can reach our intended destination from `R1` on interface `vlan1` over the upstream provider `ISP1`, and vice versa for `vlan2` and `ISP2`.

In order to achieve this desired behaviour, policy-based routing will be utilised. The policy-based route instances that will be applied are only intended to ensure that traffic, originating from the device itself, will egress on the correct upstream interface. For each of the upstream interfaces we want to specify an associated policy-based route, that should be configured based on the following requirements:

- Match on the Source IP address of the upstream interface itself.
- Only consider for locally originated frames, i.e. no frames being routed through the device should be considered for the policy-based route.
- Set the next hop address to the upstream provider associated with the particular interface the route is intended to support.

Without the policy-based routes, all active ping trigger requests will be routed based on the currently active routes in the regular routing table. Hence, we will not be able to ensure that the requests are directed towards the intended interfaces. Depending on how the routes are set up on the rest of the network, this could lead to some of the following issues:

- The ping trigger reaching its destination across an unintended path. Therefore not actually verifying the path that was intended. This could lead to false positive scenarios, where we may think that a destination is reachable across a specific upstream provider.
- The ping trigger indicating that it cannot reach its destination, even though it actually would, if sent on the correct interface.

> **WARNING**
>
> When these policy routes are active, any frame originating from the device that happen to match the policy will also be routed towards the specified next hop. This is not just active for the ping trigger traffic itself.
>
> For instance, if the IP address of one of the upstream interfaces is pinged, and the response must be routed, the response will be routed based on the next hop configured for the associated policy-based route. This is done regardless if a “better” route may exist on the system, because policy-based routes always have priority over any other route that resides in the regular routing table.

SEE ALSO

Related Documentation
- Policy-Based Routing - Locally Originated Traffic Example

Configuration

All the configuration will be carried out on the representation of router `R1`. As described above we need to configure a number of different things, so this section will consist of a number of subsections focusing on the different components of the setup.

For the sake of this configuration example, we assume that the relevant interfaces have already been configured, along with their IP addresses.

Ping Trigger

First we start with the configuration of the two ping triggers, one to be associated with each of the upstream interfaces. We want both of the ping triggers to `peer` against the IP address `198.19.20.21`, but to ensure that this is done on different outbound interfaces.

We begin with the configuration of ping trigger 1, aimed to operate over `vlan1` across the upstream provider `ISP1`:

```
R1:/#> configure
R1:/config/#> alarm
R1:/config/alarm/#> trigger ping
R1:/config/alarm/trigger-1/#> peer 10.0.0.199
R1:/config/alarm/trigger-1/#> outbound 172.16.1.1
R1:/config/alarm/trigger-1/#> end
R1:/config/alarm/#>
```

Next we configure ping trigger 2, to operate over `vlan2` across the upstream provider `ISP2`:

```
R1:/config/alarm/#> trigger ping
R1:/config/alarm/trigger-2/#> peer 10.0.0.199
R1:/config/alarm/trigger-2/#> outbound 172.16.2.1
R1:/config/alarm/trigger-2/#> end
R1:/config/alarm/#> end
R1:/config/#>
```

Take note that we specify the `outbound` option for both of the ping triggers. Doing this is very important, since this will be needed to correctly match on the intended policy-based routes that we are going to add later. The IP addresses provided are the ones configured for `vlan1` and `vlan2` respectively.

WARNING

When configuring the `outbound` option we specify the IP address of the intended interface directly, instead of providing the interface name. This is done because some inconsistencies have been noted in regards to the matching behaviour of the policy-based routes, when matching on source IP address in this specific situation.

TIP

In order to adjust the reaction time of the ping trigger it is possible to adjust the transmission `interval` and the `number` of frames that must be lost.

```
R1:/config/alarm/trigger-1/#> interval 2
R1:/config/alarm/trigger-1/#> number 1
```

With the ping triggers configured, they can now be connected to the default routes that are going to be configured in the next step.

Default Routes

In this step the two different default routes are configured. The two routes will have their next hop set to the ip address of `ISP1` and `ISP2` respectively. The previously created ping triggers will be connected to the two routes, by providing the route configuration with the `track` option. The value provided to this option is the id of the specific ping trigger. If the associated ping trigger cannot reach the configured destination the distance of the route will be adjusted to 255, so that it will not be useable.

First we configure the route that will serve as the primary uplink:

```
R1:/config/#> ip
R1:/config/ip/#> route default 172.16.1.99 track 1
```

NOTE

When we provide no specific distance to a routes configuration, it will be given a distance of 1.

Next we will add the route that will act as the secondary uplink. This route will be configured to have a `distance` of 10, so that it will not be useable while the primary route is active:

```
R1:/config/ip/#> route default 172.16.2.99 10 track 2
```

In summary, we configured two different default gateways, each pointing to either `ISP1` at 172.16.1.199, or `ISP2` at 172.16.2.199. The route pointing to `ISP2` have a higher distance, and will therefore not be active while the primary route is. Both of the routes have been connected to a ping trigger using the `track` command. If the associated ping trigger fails to reach its destination, the route will have its distance increased to 255, making it inactive. This is the mechanism that allows for the failover to occur.

Policy-Based Routes

Lastly we configure the necessary policy-based routes that will ensure the ping trigger traffic egress towards the correct upstream. We configure two different policy-based route instances, one associated with each ping trigger.

First, we start with the route that should handle ping trigger 1, responsible for verifying the uplink towards `ISP1`:

```
R1:/config/ip/#> policy-route 1
R1:/config/ip/policy-route-1/#> in-iface lo
R1:/config/ip/policy-route-1/#> next-hop 172.16.1.99
R1:/config/ip/policy-route-1/#> match saddr 172.16.1.1/32
R1:/config/ip/policy-route-1/#> end
R1:/config/ip/#>
```

Next, we configure the route that should handle ping trigger 2, responsible for verifying the uplink towards `ISP2`:

```
R1:/config/ip/#> policy-route 2
R1:/config/ip/policy-route-2/#> in-iface lo
R1:/config/ip/policy-route-2/#> next-hop 172.16.2.99
R1:/config/ip/policy-route-2/#> match saddr 172.16.2.1/32
R1:/config/ip/policy-route-2/#> end
R1:/config/ip/#>
```

In summary, two policy-based routes have been configured, one associated with each ping trigger. This is the intention for each of the options set for the instance:

- The `in-iface` is set to the `lo` interface. Doing this will ensure that only traffic originating from the device itself can match the route.
- The `next-hop` is configured to be the address of the upstream next-hop, in this case either `ISP1` for `policy-route 1`, or `ISP2` for `policy-route 2`.
- The `match saddr` is set to the IP address of the local upstream interface. In this case the ip address of interface `vlan1` for `ISP1` and the IP address of interface `vlan2` for `ISP2`.

If everything has been configured correctly, everything should now be as we want it to be. We can now apply the configuration and verify if it works.

```
R1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#>
```

Status

Initially we expect everything to be up and running correctly, with no link failures anywhere. Therefore, we start with simply verifying that everything looks correct.

The first thing we can verify is that our policy-based routes look correct, so that we can expect our ping triggers to egress on the correct interfaces. It should look something like this.

```
R1:/#>  show ip policy-route
PRIO  SADDR          DADDR  SPORT  DPORT  IN-IFACE  NEXT-HOP
300   172.16.1.1/32  -      -      -      lo        172.16.1.99
301   172.16.2.1/32  -      -      -      lo        172.16.2.99
```

If the policy-based routes look correct, we can check the alarm status, to see if the ping triggers can correctly reach their destination address. It should look something like this, with both ping triggers reporting that the ping works.

```
R1:/#>  show alarm
NO TRIGGER          ENA ACT REASON
 1 Ping             YES  NO ping to 10.0.0.199 works
 2 Ping             YES  NO ping to 10.0.0.199 works
```

At this point we can check if our default gateways looks as we want them to. If everything is correct the route towards `ISP1` at `172.16.1.99` should be the active default route:

```
R1:/#>  show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S   0.0.0.0/0 [10/0] via 172.16.2.99, vlan2, weight 1, 07:03:24
S>* 0.0.0.0/0 [1/0] via 172.16.1.99, vlan1, weight 1, 07:03:24
C>* 172.16.1.0/24 is directly connected, vlan1, 07:03:24
C>* 172.16.2.0/24 is directly connected, vlan2, 07:03:24
```

If all the above checks out, we should be in a operating state that mirrors what we expected, based on the technical setup description.

Status On Link Failure

If we have a link failure, as shown in **Figure 2**, we expect the primary route to have its distance increased, so that the secondary route will now be the active route.

However, first we can check the status of our ping triggers. If the destination address is unreachable over interface `vlan1` through `ISP1`, trigger 1 should now indicate this:

```
R1:/#> show alarm
NO TRIGGER          ENA ACT REASON
 1 Ping             YES YES ping to 10.0.0.199 fails
 2 Ping             YES  NO ping to 10.0.0.199 works
```

If the ping trigger indicates that it fails to reach the destination, the associated route should now have had its distance increased to 255. If we check the routing table it should now show that the default route towards `ISP2` at `172.16.2.99` is the active default route:

```
R1:/#>  show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 0.0.0.0/0 [10/0] via 172.16.2.99, vlan2, weight 1, 00:00:02
S   0.0.0.0/0 [255/0] via 172.16.1.99, vlan1, weight 1, 00:00:02
C>* 172.16.1.0/24 is directly connected, vlan1, 07:10:13
C>* 172.16.2.0/24 is directly connected, vlan2, 07:10:13
```

If everything has behaved as above, we have now correctly had the upstream failover to the secondary uplink.

Possibilities When Combined With ECMP Routing

It is possible to combine this solution with ECMP routing, if we want to have failover, but utilise both of the uplinks at the same time. For an example how load balancing can be achieved, in a similar situation to the one presented in this document, refer to the following howto on ECMP load balancing.

In ECMP routing, we can have multiple routes, that are of equal cost, for the same destination active at the same time. This can allow for load balancing to occur between the Primary and Secondary uplinks, in this use case example. The failover scenario in this situation will be the same, the route will have its distance increased so that it is no longer selectable. Therefore, the failover scenario can be seen as simply disabling a specific route, instead of performing a failover to a secondary route.

In order to achieve this type of setup, the only thing that would need to change is the configuration of the routes themselves. Instead of providing the secondary route with a higher distance value, we simply ensure that all the upstream routes have the same distance. Configuring the two default routes with the same distance, will lead to them having the same cost, making ECMP routing possible. Therefore, we perform the route configuration from this step, in the following manner instead:

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> route default 172.16.1.99 track 1
R1:/config/ip/#> route default 172.16.2.99 track 2
R1:/config/ip/#> leave
Applying configuration.
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
R1:/#>
```

Now both of these routes should be active at the same time, and should be selectable at the same time, for individual flows of packets. In order to verify this, check the active routing table:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 0.0.0.0/0 [1/0] via 172.16.1.99, vlan1, weight 1, 00:00:01
  *                 via 172.16.2.99, vlan2, weight 1, 00:00:01
C>* 172.16.1.0/24 is directly connected, vlan1, 00:01:32
C>* 172.16.2.0/24 is directly connected, vlan2, 00:01:32
```

As we can see, both of our default routes will now be active at the same time. Again, if we have a link failure as described in **Figure 2**, it should still look similar to how it did in the case where we did not utilise ECMP routing:

```
R1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

S>* 0.0.0.0/0 [1/0] via 172.16.2.99, vlan2, weight 1, 00:00:02
S   0.0.0.0/0 [255/0] via 172.16.1.99, vlan1, weight 1, 00:00:02
C>* 172.16.1.0/24 is directly connected, vlan1, 00:01:32
C>* 172.16.2.0/24 is directly connected, vlan2, 00:01:32
```

SEE ALSO

Related Documentation

- Equal-Cost Multi-Path Routing (ECMP) Overview
- Equal-Cost Multi-Path Routing (ECMP) HowTo

4.9.11. Protocol Independent Multicast - Sparse Mode (PIM-SM) HowTo

About

This document aims to provide some example use cases and example scenarios how Protocol Independent Multicast - Sparse Mode (PIM-SM) could be utilised to simplify multicast routing.

> **SEE ALSO**
>
> Related Documentation
> - PIM-SM Configuration Guide
> - OSPF Configuration Guide
> - RIP Configuration Guide

Introduction

Multicast routing can be tricky and difficult to configure and scale, when using only static multicast routing. PIM simplifies the configuration of multicast routing in a network by allowing it to be configured in a dynamic manner.

PIM relies on the underlying unicast routing in order to operate. So it is still necessary to ensure that the network have a solid unicast routing setup. However, this is in many cases easier to configure and understand for most users. How the unicast routing is set up is not important, as long as it is correct. As the name suggests, PIM functions *independently* in regards to the underlying unicast protocol. Therefore, it functions really well together with a dynamic unicast routing protocol like OSPF or RIP.

PIM is a family consisting of a few different variants. This document focus solely on the set up and usage of the Sparse Mode (SM) variant. PIM-SM is best used for setups where the potential receivers of multicast are *sparsely* scattered throughout the intended network topology. In this scenario it also scales pretty well in somewhat larger topologies as well.

The examples in this document are somewhat simplified in order to try and convey the basics and to provide an understanding of how PIM is configured using the system.

> **SEE ALSO**
>
> Use cases presented in this document
> - Case 1: PIM-SM With Any Source Multicast

Case 1: PIM-SM With Any Source Multicast

This example covers how PIM-SM with the so called Any Source Multicast (ASM) mode can be configured. In the ASM mode the source of any given multicast stream does not need to be known by any of the potential receivers in the network. The receivers only need to know what multicast that they may want to receive.

The central aspect when using the ASM mode is the Rendezvous Point (RP). Since the receivers may be unaware of where the source of any requested multicast is located, the RP serves as a sort of meeting point for receivers and senders. For a more detailed explanation of this refer to the PIM-SM Configuration Guide.

```
                .-----. (Receiver)              .-----. (Receiver)
                | H1 |                          | H2 |
                '-+--'                          '--+-'
               .99|                                |.99
                  |                                |
192.168.10.0/24 |                                | 192.168.20.0/24
                  |                                |
                .1| vlan3                   vlan3 |.2
      (LHR) .---+----.                       .----+---. (LHR)
       (DR) |        |.1   10.0.1.0/24   .2|        | (DR)
            |   R1   +---------------------+   R2   |
            |        | vlan1         vlan2 |        |
            '---+----'                       '----+---'
                .1| vlan2                   vlan1 |.2
                  |                                |
    10.0.4.0/24 |                                | 10.0.2.0/24
                  |                                |
                .3| vlan1                   vlan2 |.4
      (FHR) .---+----.                       .----+---. (RP)
            |        | vlan2         vlan1 |        |
            |   R3   +---------------------+   R4   |
            |        |.3   10.0.3.0/24   .4|        |
            '---+----'                       '--------'
                .3| vlan3
                  |
192.168.30.0/24 |
                  |
               .99|
                .-+--. (Sender)
                | S1 |
                '----'
```

Figure 1: An example of a PIM-SM ASM setup in simple network topology.

In the example topology presented in **Figure 1** four routers are used to form the PIM domain. All these routers need to first be configured with a unicast routing protocol.

When the unicast routing have been set up, the main focus will be the configuration of PIM on all of the routers. The main configuration aspect is to ensure that the router `R4` will be the configured as the RP, located at the address `10.0.2.4`.

When everything has been configured the receivers `H1` and `H2` should be able to receive any multicast stream produced by the sender `S1`.

Configuration

There are a few different things that needs to be configured before we will be able to route multicast from the sender to the receivers:

- Configuration of the underlying unicast routing.
- Ensuring that multicast forwarding is enabled on all of the routers.
- Making sure that IGMP is enabled on the relevant interfaces, the ones facing the receivers.
- Configuration of PIM-SM specific settings.
  - Selection of PIM interfaces.
  - Location of the RP.

For this example it is assumed that the interfaces and their IP addresses have already been configured.

Underlying Unicast Routing

In this example we will use OSPF as our underlying routing protocol. It needs to be configured on each of the individual routers. All of the internal networks **10.0.1.0/24**, **10.0.2.0/24**, **10.0.3.0/24** and **10.0.4.0/24** will be added as OSPF networks. The connections to the sender and the receivers will be configured to be redistributed into the OSPF network with `redistribute connected`.

Configuration of router **R1**:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> ospf
R1:/config/router/ospf/#> network 10.0.1.0/24
R1:/config/router/ospf/#> network 10.0.4.0/24
R1:/config/router/ospf/#> redistribute connected
R1:/config/router/ospf/#> leave
R1:/#>
```

Configuration of router **R2**:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> ospf
R2:/config/router/ospf/#> network 10.0.1.0/24
R2:/config/router/ospf/#> network 10.0.2.0/24
R2:/config/router/ospf/#> redistribute connected
R2:/config/router/ospf/#> leave
R2:/#>
```

Configuration of router **R3**:

```
R3:/#> configure
R3:/config/#> router
R3:/config/router/#> ospf
R3:/config/router/ospf/#> network 10.0.3.0/24
R3:/config/router/ospf/#> network 10.0.4.0/24
R3:/config/router/ospf/#> redistribute connected
R3:/config/router/ospf/#> leave
R3:/#>
```

Configuration of router **R4**:

```
R4:/#> configure
R4:/config/#> router
R4:/config/router/#> ospf
R4:/config/router/ospf/#> network 10.0.2.0/24
R4:/config/router/ospf/#> network 10.0.3.0/24
R4:/config/router/ospf/#> leave
R4:/#>
```

This configuration should be enough to ensure that the underlying unicast routing, required for PIM to function, is in place.

IGMP

For PIM to function IGMP must be enabled on the interfaces that face any potential receiver. Therefore, ensure that IGMP is enabled on the receiver interfaces on router **R1** and **R2**.

```
R1:/#> configure
R1:/config/#> vlan 3
R1:/config/vlan-3/#> multicast-snooping
R1:/config/vlan-3/#> leave
R1:/#>
```

```
R2:/#> configure
R2:/config/#> vlan 3
R2:/config/vlan-3/#> multicast-snooping
R2:/config/vlan-3/#> leave
R2:/#>
```

Multicast Forwarding

Multicast forwarding is not enabled by default on the system. In order for PIM to be configurable on the system it must be enabled on every router that is to operate as a multicast router. We can enable it under the `ip` configuration context using the `multicast-forwarding`.

```
R1:/#> configure
R1:/config/#> ip
R1:/config/ip/#> multicast-forwarding
R1:/config/ip/#> leave
R1:/#>
```

```
R2:/#> configure
R2:/config/#> ip
R2:/config/ip/#> multicast-forwarding
R2:/config/ip/#> leave
R2:/#>
```

```
R3:/#> configure
R3:/config/#> ip
R3:/config/ip/#> multicast-forwarding
R3:/config/ip/#> leave
R3:/#>
```

```
R4:/#> configure
R4:/config/#> ip
R4:/config/ip/#> multicast-forwarding
R4:/config/ip/#> leave
R4:/#>
```

PIM-SM

At this point PIM-SM itself is the last thing to be configured. The minimum required is to specify the location of the RP at `10.0.2.4` router **R3**. This must be configured exactly the same on each of the PIM routers in the network.

Lastly the interfaces that PIM is to operate on also must be provided in the configuration. Any interface facing another PIM router, a multicast source or a multicast receiver must be specified.

Configuration of router **R1**:

```
R1:/#> configure
R1:/config/#> router
R1:/config/router/#> pim
R1:/config/router/pim/#> interface vlan1 vlan2 vlan3
R1:/config/router/pim/#> rendezvous-point 10.0.2.4
R1:/config/router/pim/#> leave
R1:/#>
```

Configuration of router **R2**:

```
R2:/#> configure
R2:/config/#> router
R2:/config/router/#> pim
R2:/config/router/pim/#> interface vlan1 vlan2 vlan3
R2:/config/router/pim/#> rendezvous-point 10.0.2.4
R2:/config/router/pim/#> leave
R2:/#>
```

Configuration of router **R3**:

```
R3:/#> configure
R3:/config/#> router
R3:/config/router/#> pim
R3:/config/router/pim/#> interface vlan1 vlan2 vlan3
R3:/config/router/pim/#> rendezvous-point 10.0.2.4
R3:/config/router/pim/#> leave
R3:/#>
```

Configuration of router **R4**:

```
R4:/#> configure
R4:/config/#> router
R4:/config/router/#> pim
R4:/config/router/pim/#> interface vlan1 vlan2
R4:/config/router/pim/#> rendezvous-point 10.0.2.4
R4:/config/router/pim/#> leave
R4:/#>
```

If everything has been configured correctly, it should now be possible to route a multicast stream thorough the network from `S1` to any of the receivers `H1` or `H2`.

Status

Before any multicast groups are requested by any of the receivers, or any streams transmitted by any source, we could take a quick look if everything initially seems to be correctly configured and running. The `show ip pim` command can be used to display a collection of a number of different pim status outputs, for more status see the following.

Initial PIM Status on router **R1** should look something like this:

```
R1:/#> show ip pim

PIM Rendezvous Point Info
RP-ADDRESS   GROUP/PREFIX-LIST   OIF       I-AM-RP   SOURCE   GROUP-TYPE
10.0.2.4     224.0.0.0/4         Unknown   no        Static   ASM

PIM Interfaces
INTERFACE   STATE   ADDRESS        PIM-NBRS   PIM-DR     FHR   IF-CHANNELS
pimreg      up      0.0.0.0        0          local      0     0
vlan1       up      10.0.1.1       1          10.0.1.2   0     0
vlan2       up      10.0.4.1       1          10.0.4.3   0     0
vlan3       up      192.168.10.1   0          local      0     0

PIM Neighbours
INTERFACE   NEIGHBOR   UPTIME     HOLDTIME   DR-PRIO
```

```
vlan1      10.0.1.2  00:00:03  00:01:42  1
vlan2      10.0.4.3  00:00:02  00:01:43  1

PIM Join
INTERFACE  ADDRESS  SOURCE  GROUP  STATE  UPTIME  EXPIRE  PRUNE

PIM Upstream
IIF  SOURCE  GROUP  STATE  UPTIME  JOINTIMER  RSTIMER  KATIMER  REFCNT
R1:/#>
```

For each of the PIM routers we should be able to see that the Interfaces are configured correctly. It is also good to see that the neighbours are detected correctly. Verifying that the RP is configured and reported correctly on each of the routers is also good, since if this is incorrect PIM may not function.

If everything looks okay so far we could now verify that we can route a multicast stream from the sender to any of the receivers.

> **NOTE**
>
> To access the system shell, you must connect to the device via the console using an administrative account with the appropriate permissions. While it is possible to access the shell via SSH, this feature must first be enabled through a console session.

> **TIP**
>
> For testing purposes the OS has access to the too `mcjoin` that can be used to both join a specific multicast group and transmit a stream towards that group.
>
> As an example we use it to join group 225.1.2.3 acting as device H1:
>
> ```
> H1:/#> shell
> Domain shell started.
> root@H1:/home/admin # mcjoin -i vlan1 -j 225.1.2.3
> ```
>
> Next, this is how we could create a stream using the same tool on S1:
>
> ```
> S1:/#> shell
> Domain shell started.
> root@S1:/home/admin # mcjoin -i vlan1 -s -t 5 225.1.2.3
> ```

If we have requested traffic for group `225.1.2.3` on H1 and we are transmitting it from S1 we should be able to check the installed multicast routes, and it should look something like this for each of the routers:

```
R1:/#> show ip pim mroute
IP Multicast Routing Table
Flags: S - Sparse, C - Connected, P - Pruned
       R - SGRpt Pruned, F - Register flag, T - SPT-bit set
SOURCE          GROUP      FLAGS  PROTO  INPUT  OUTPUT  TTL  UPTIME
*               225.1.2.3  SC     IGMP   vlan1  pimreg  1    00:05:50
```

```
                                  IGMP          vlan3   1
192.168.30.99  225.1.2.3  ST      STAR   vlan2  vlan3   1    00:05:35
```

```
R2:/#> show ip pim mroute
IP Multicast Routing Table
Flags: S - Sparse, C - Connected, P - Pruned
       R - SGRpt Pruned, F - Register flag, T - SPT-bit set
SOURCE         GROUP      FLAGS  PROTO  INPUT  OUTPUT  TTL  UPTIME
*              225.1.2.3  S      PIM    vlan1  vlan2   1    00:07:34
192.168.30.99  225.1.2.3  SRP    none   vlan1  none    0    --:--:--
```

```
R3:/#> show ip pim mroute
IP Multicast Routing Table
Flags: S - Sparse, C - Connected, P - Pruned
       R - SGRpt Pruned, F - Register flag, T - SPT-bit set
SOURCE         GROUP      FLAGS  PROTO  INPUT  OUTPUT  TTL  UPTIME
192.168.30.99  225.1.2.3  SFT    PIM    vlan3  vlan1   1    00:07:26
```

```
R4:/#> show ip pim mroute
IP Multicast Routing Table
Flags: S - Sparse, C - Connected, P - Pruned
       R - SGRpt Pruned, F - Register flag, T - SPT-bit set
SOURCE         GROUP      FLAGS  PROTO  INPUT  OUTPUT  TTL  UPTIME
*              225.1.2.3  S      none   vlan2  none    0    --:--:--
192.168.30.99  225.1.2.3  SRP    none   vlan1  none    0    --:--:--
```

By checking the routes we should see that the flow of traffic should be **S1** –> **R3** –> **R1** –> **H1**, by using the (192.168.30.99,225.1.2.3) mroutes.

4.10. Tunnels and VPN

4.10.1. Host to LAN Tunnel HowTo

About

This document provides an example of how to connect to a network using OpenVPN with certificates for authentication, and a username/password pair for an extra layer of security.

Introduction

Picture a road traffic monitoring system with a control centre and a number of mobile stations that should be able to connect to the control centre over a public network to report vehicle data and receive configuration information.

```
       .-------.  .-------.
       | HostA |  | HostB |
       '---+---'  '---+---'
         .2|          |.2
           '---.  .---'
10.0.1.0/24    |  |    10.0.2.0/24
             .1|  |.1
            .--+--+--.                          .---------.
            |        |    OpenVPN L3 Tunnel    |         |
            | Server +=========================+ Client1 |
            |        | ssl0               ssl0 |         |
            '----+---' 10.0.3.1                '----+----'
      Public IP: |                                 |
    198.18.19.20 |             .--.-.              |
                 |             ( (    )__          |
                 '------------(_,  \ ) ,_)----------'
Control Center                  '-'--`--'
                                    |       Mobile Stations
                                    |       .--------.
                                    `-------+        |-.
                                            | ClientN| |
                                            '--------' |
                                              '--------'
```

Client1 represents a mobile station, running the OpenVPN client who wants to reach HostA and HostB in the control centre. *Server* is the gateway to the control centre as well as the device where the OpenVPN server is running.

The example assumes that certificates and CA-certificates have already been imported to the client and server respectively.

Server Tunnel Setup

On the server device, begin by setting up the basic tunnel.

```
server:/#> configure
server:/config/#> tunnel ssl 0
server:/config/tunnel/ssl-0/#> server
server:/config/tunnel/ssl-0/#> method cert
server:/config/tunnel/ssl-0/#> certificate server1
server:/config/tunnel/ssl-0/#> ca-certificate server1
server:/config/tunnel/ssl-0/#> type tun
server:/config/tunnel/ssl-0/#> pool start 10.0.3.100 num 100 netmask 255.255.255.0
server:/config/tunnel/ssl-0/#> leave
server:/#>
```

A few comments on the above sequence:

- The certificate and CA-certificate both have the label *server1* but they are two different certificates. When importing a PKCS-bundle into WeOS, all included certificates and keys are assigned the same label. WeOS makes sure the correct certificate is used.
- The tunnel type is set to `tun` as we have no use for link layer (Ethernet) signalling in this scenario.
- The `pool` setting enables dynamic IP address assignment of connecting clients from the specified range. The tunnel is now ready and waiting for connections.

Ensure that the tunnel ssl0 is listed as *UP* using:

```
server:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
0        ssl0            UP       0 Days 0 Hours 0 Mins 7 Secs
```

Next, assign an IP address to the local tunnel interface.

```
server:/#> configure
server:/config/#> iface ssl0
server:/config/iface-ssl0/#> inet static 10.0.3.1/24
server:/config/iface-ssl0/inet-static-10.0.3.1/#> end
server:/config/iface-ssl0/#> no inet ssl
server:/config/iface-ssl0/#> leave
server:/#>
```

Verify that the interface named *ssl0* is UP and has the IP address/netmask which was just configured:

```
server:/#> show iface
INTERFACE         OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
ssl0              UP    10.0.3.1/24         static      N/A
vlan1             UP    10.0.1.1/24         static      00:07:7c:1c:cf:a0
                        169.254.71.83/16    link-local
vlan2             UP    10.0.2.1/24         static      00:07:7c:1c:cf:a0
vlan3             UP    198.18.19.20/24     static      00:07:7c:1c:cf:a0
server:/#>
```

Client Tunnel Setup

On the client device, begin by setting up the basic tunnel.

```
client1:/#> configure
client1:/config/#> tunnel ssl 0
client1:/config/iface-ssl0/#> no server
client1:/config/iface-ssl0/#> method cert
client1:/config/iface-ssl0/#> certificate client1
client1:/config/iface-ssl0/#> ca-certificate client1
client1:/config/iface-ssl0/#> type tun
client1:/config/iface-ssl0/#> peer 198.18.19.20
client1:/config/iface-ssl0/#> leave
client1:/#>
```

The client config is similar to that of the server. The main differences are:

- The `no server` command is used inform WeOS that this is a client.
- Different certificates are used for client (as compared to the server).

- A `peer` command identifies the public IP address of the server.

This should be enough to make the tunnel come up (it might take a few seconds). Run:

```
client1:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
0        ssl0            UP       0 Days 0 Hours 0 Mins 46 Secs
client1:/#> show iface
client1:/#> show iface
INTERFACE         OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
ssl0              UP    10.0.3.100/24       ssl0        N/A
vlan1             UP    198.18.19.55/24     static      00:07:7c:6e:3e:c0
client1:/#>
```

And verify that tunnel ssl0 is UP and the corresponding interface has received an IP address from the pool configured on the server.

It should now be possible to ping the server across the tunnnel:

```
client1:/#> ping count 3 10.0.3.1
Press Ctrl-C to abort PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: seq=0 ttl=64 time=1.124 ms
64 bytes from 10.0.3.1: seq=1 ttl=64 time=1.014 ms
64 bytes from 10.0.3.1: seq=2 ttl=64 time=0.888 ms

--- 10.0.3.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.888/1.008/1.124 ms
client1:/#>
```

Server Hardening

With the tunnel operational we can now “harden” it with additional authentication.

On the server device, start by defining a username and password to be used for client identification.

```
server:/#> configure
server:/config/#> aaa
server:/config/aaa/#> local-db 1
server:/config/aaa/local-db-1/#> username client1 password qwerty
server:/config/aaa/local-db-1/#> leave
server:/#>
```

This sets up username *client1* and associates it with the password *qwerty*. In this case the credentials are stored on the local device but other options are available. See WeOS AAA for details.

Now add a user authentication method to the tunnel configuration.

```
server:/#> configure
server:/config/#> tunnel ssl 0
server:/config/tunnel/ssl-0/#> aaa-method local-db 1
server:/config/tunnel/ssl-0/#> leave
server:/#>
```

This configuration change causes the OpenVPN server to restart. The client eventually detects that it has lost connection (it might take up to a minute) and try to re-connect, but will not be allowed to do so as it is not yet configured with the correct user credentials.

Client Hardening

On the client device, check tunnel status.

```
client1:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
0        ssl0            Down
```

The ssl0 tunnel should be DOWN. If it is still up, it means the broken connection due to the server restart has not been detected yet. (See the keepalive option for more details on this mechanism.)

Now add the username and password to the tunnel configuration:

```
client1:/#> configure
client1:/config/#> tunnel ssl 0
client1:/config/tunnel/ssl-0/#> identity client1 password qwerty
client1:/config/tunnel/ssl-0/#> leave
client1:/#>
```

This change causes another restart of the tunnel. Follow steps from part 1 to verify that the tunnel is UP and that pinging the other end works.

Server Push Routes

The final part is to configure the server to automatically push routes to the internal networks that the client needs to access. This simplifies client administration as the needed routes are created automatically on the client.

```
server:/#> configure
server:/config/#> tunnel ssl 0
server:/config/tunnel/ssl-0/#> push-network 10.0.1.0/24
server:/config/tunnel/ssl-0/#> push-network 10.0.2.0/24
server:/config/tunnel/ssl-0/#> leave
server:/#>
```

The OpenVPN server restarts again, dropping all connections.

Client Push Routes

The OpenVPN client eventually detects the connection loss and restarts the tunnel once more. If you want to speed up the process, you can manually force a tunnel restart by executing:

```
client1:/#> tunnel ssl restart
```

> **NOTE**
>
> If you have multiple tunnels, this restarts all of them.

Wait a few seconds for the tunnel to be re-established. Then check the routing table:

```
client1:/#> show ip route
S - Static | C - Connected | K - Kernel route  | > - Selected route
O - OSPF   | R - RIP       | [Distance/Metric] | * - FIB route

s>* 10.0.1.0/24 [1/0] via 10.0.3.1, ssl0, weight 1, 00:00:18
s>* 10.0.2.0/24 [1/0] via 10.0.3.1, ssl0, weight 1, 00:00:18
C>* 10.0.3.0/24 is directly connected, ssl0, 00:00:18
C>* 198.18.19.0/24 is directly connected, vlan1, 00:31:23
client1:/#>
```

There should be routes to the 10.0.1.0 and 10.0.2.0 networks with the tunnel interface as gateway. With the routes in place it should be possible to ping HostA and HostB on the control centre internal network (example for HostA below):

```
client1:/#> ping count 3 10.0.1.2
Press Ctrl-C to abort PING 10.0.1.2 (10.0.1.2): 56 data bytes
64 bytes from 10.0.1.2: seq=0 ttl=63 time=1.552 ms
64 bytes from 10.0.1.2: seq=1 ttl=63 time=1.416 ms
64 bytes from 10.0.1.2: seq=2 ttl=63 time=1.643 ms

--- 10.0.1.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.416/1.537/1.643 ms
```

> **NOTE**
>
> If you are trying this example in a real network, you must configure Host A and B to reach the 10.0.3.0 network (for example by assigning the server as their default gateway), otherwise they will not be able to respond to the ping.

4.10.2. LAN Tunnel Bridging HowTo

About

This document provides an example of how to set up a LAN to LAN tunnel bridge using OpenVPN with a pre-shared key (PSK). For simplicity, only basic configuration options are used.

Introduction

Company X has two sites A and B in different geographical locations, connected by a public network. They want to connect both sites to the same LAN and use a DHCP server on Site A to serve Site B.

```
          Site A                                Site B

        .--------.                            .--------.
        |  DHCP  |                            |        |
        | Server |                            |  Host  |
        |   A1   |                            |   B1   |
        '----+---'                            '---+----'
             | IP:192.168.1.1                     | IP:DHCP(A1)
             |                                    |
             |                                    |
             | vlan100                            | vlan100
        eth1 | IP:DHCP(A1)                   eth1 | IP:DHCP(A1)
        .----+---.                             .---+----.
        |        |                             |        |
        |  GW-A  |                             |  GW-B  |
        |        |                             |        |
        '----+---'                             '---+----'
      vlan2  |                .--.-.                | vlan2
  Public IP: |               ( (    )__             | Public IP:
198.19.20.21 '-------------(_,  \ ) ,_)-----------' 198.19.20.22
                              '-'--`--'
                     Public Network/Internet
```

WeOS devices GW-A and GW-B both have a port on VLAN 2 connected to the WAN (Public network/Internet) and eth1 connected to the local network.

The example assumes that an OpenVPN key has been generated offline and imported into both GW-A and GW-B under the label *psk1*.

DHCP Server A1 and Host B1 are not configured in this example. The gateways GW-A and GW-B are assumed to be able to reach each other via public IPv4 addresses.

Server Configuration

In this setup *GW-A* is the OpenVPN server node. To configure the server side of the tunnel, enter the following commands in the CLI:

```
server:/#> configure
server:/config/#> tunnel ssl 0
server:/config/tunnel/ssl-0/#> method psk
server:/config/tunnel/ssl-0/#> secret psk1
server:/config/tunnel/ssl-0/#> server
server:/config/tunnel/ssl-0/#> type tap
server:/config/tunnel/ssl-0/#> leave
server:/#>
```

The OpenVPN server is now up and running, waiting for a connection. You can verify this with the following admin-exec commands:

```
server:/#> show port ssl0
ssl0
Link            : Up
Oper. status    : Enabled
Admin status    : Enabled
Description     : ssl0
Type            : SSL
State           : ──────────
Alarm           : N/A
VLAN            :
MAC             : a2:67:57:f4:0a:42
server:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
0        ssl0            Down
```

Since the tunnel is a layer 2 (`tap`) tunnel, its endpoint *ssl0* is treated as a virtual port. As such, port-type commands can be used to operate on it. The first command displays *ssl0* as Enabled with Link UP, and with a MAC address assigned to it.

The second command displays the tunnel as DOWN, which is expected. This changes to UP once a connection with a client has been established.

Next, we bridge *ssl0* with *eth1* to include the tunnel in the broadcast domain of the local subnet. We also set the bridge interface *vlan100* to request a local IP address via DHCP.

```
server:/#> configure vlan 100
server:/config/vlan-100/#> untagged eth1,ssl0
server:/config/vlan-100/#> end
server:/config/#> iface vlan100
server:/config/iface-vlan100/#> inet dhcp
server:/config/iface-vlan100/inet-dhcp/#> leave
server:/#>
```

Verify that the an IP address to interface *vlan100* was assigned by running:

```
server:/#> show iface
INTERFACE         OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             DOWN  169.254.71.83/16    link-local  00:07:7c:1c:cf:a0
vlan2             UP    198.18.19.20/24     static      00:07:7c:1c:cf:a0
vlan100           UP    192.168.1.189/24    dhcp        00:07:7c:1c:cf:a0
server:/#>
```

Note: this requires that a DHCP server is available on subnet. It may take a couple of seconds for the IP address to be assigned.

Client Configuration

GW-B is the OpenVPN client. To configure the client side of the tunnel, enter the following commands in the CLI:

```
client:/#> configure
client:/config/#> tunnel ssl 0
client:/config/tunnel/ssl-0/#> method psk
client:/config/tunnel/ssl-0/#> secret psk1
client:/config/tunnel/ssl-0/#> no server
```

```
client:/config/tunnel/ssl-0/#> peer 198.18.19.20
client:/config/tunnel/ssl-0/#> type tap
client:/config/tunnel/ssl-0/#> leave
client:/#>
```

The IP address for the peer is the public IP address of the server.

Again, verify the operational status:

```
client:/#> show port ssl0
ssl0
Link            : Up
Oper. status    : Enabled
Admin status    : Enabled
Description     : ssl0
Type            : SSL
State           : Forwarding
Alarm           : N/A
VLAN            :
MAC             : a2:3f:63:78:89:7d
client:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
0        ssl0            UP       0 Days 0 Hours 0 Mins 18 Secs
```

The first command displays *ssl0* as being Enabled with Link UP, and with a MAC address assigned to it.

The second command should now display the tunnel as UP. It may, however, take a few seconds for the configuration to be applied and for the tunnel to be established before it is displayed as UP.

Once the tunnel is UP, we bridge *ssl0* with *eth1* to include the tunnel in the broadcast domain of the local subnet, like we did for the server side. This effectively connects the two LAN segments (Site A and Site B) together.

Also set bridge interface *vlan100* to request a local IP address via DHCP, i.e., over the tunnel:

```
client:/#> configure
client:/config/#> vlan 100
client:/config/vlan-100/#> untagged eth1,ssl0
client:/config/vlan-100/#> end
client:/config/#> iface vlan100
client:/config/iface-vlan100/#> inet dhcp
client:/config/iface-vlan100/#> leave
client:/#>
```

Verify that an IP address was assigned to *vlan100* by running:

```
client:/#> show iface
INTERFACE         OPER  ADDRESS/LENGTH      SOURCE      MAC/PTP ADDRESS
lo                UP    127.0.0.1/8         static      00:00:00:00:00:00
vlan1             DOWN  169.254.63.249/16   link-local  00:07:7c:1c:d2:80
vlan2             UP    198.18.19.21/24     static      00:07:7c:1c:d2:80
vlan100           UP    192.168.1.153/24    dhcp        00:07:7c:1c:d2:80
client:/#>
```

It should be a DHCP-assigned IP address. It may take some time for the IP address to appear – keep in mind that all DHCP traffic has to be passed across the tunnel, which in turn means encrypting the traffic, encapsulating it in UDP packets, routing them across the public network, and doing the reverse procedure at the other end.

Final Topology

The following figure illustrates the resulting topology, with an L2 tunnel connecting the two LANs via the virtual port ssl0 on each GW device.

```
          Site A                              Site B

        .--------.                          .--------.
        |  DHCP  |                          |        |
        | Server |                          |  Host  |
        |   A1   |                          |   B1   |
        '----+---'                          '---+----'
             | IP:192.168.1.1                   | IP:DHCP(A1)
             |                                  |
             |                                  |
             | vlan100                          | vlan100
             | IP:DHCP(A1)                      | IP:DHCP(A1)
        eth1 | Untagged:eth1,ssl0           eth1 | Untagged:eth1,ssl0
        .----+---.                          .---+----.
        |     `. |     OpenVPN L2 Tunnel     | .´     |
        |  GW-A `+===========================+´ GW-B  |
        |        | ssl0                 ssl0 |        |
        '----+---'                          '---+----'
             |                                  |
       vlan2 |              .--.-.              | vlan2
  Public IP: |             ( (    )__           | Public IP:
198.18.19.20 '------------(_,  \ ) ,_)-----------' 198.18.19.21
                            '-'--`--'
                     Public Network/Internet
```

4.10.3. Internal Routes HowTo

About

This document provides an example of how to set up OpenVPN internal routes. For simplicity, only basic configuration options are used.

Introduction

Company X has two sites A and B in different geographical locations, connected by a public network. They want to connect both sites to the same LAN and and have the networks behind the VPN nodes communicate.

The OpenVPN server node needs to know which networks are behind each of the client nodes. The internal routes associated with each client must be specified in the server configuration, as it is not able to infer this information from the routing table.

```
          Site A                              Site B

        .--------.                          .--------.
        |        |                          |        |
        |  Host  |                          |  Host  |
        |   A1   |                          |   B1   |
        '----+---'                          '---+----'
             | IP:10.70.0.2                     | IP:20.60.0.2
             |                                  |
             |                                  |
             |                                  |
        eth1 | IP:10.70.0.1                eth1 | IP:20.60.0.1
        .----+---.                          .---+----.
        |        |                          |        |
        |  GW-A  |                          |  GW-B  |
        | server |                          | client |
        '----+---'                          '---+----'
      vlan2  |               .--.-.              | vlan2
  Public IP: |              ( (    )__           | Public IP:
192.168.0.21 '-------------(_,  \ ) ,_)----------' 192.168.0.22
                             '-'--`--'
                     Public Network/Internet
```

WeOS devices GW-A and GW-B both have a port on VLAN 2 connected to the WAN (Public network/Internet) and eth1 connected to the local network.

The example assumes that certificates have been generated offline and imported into both GW-A and GW-B.

Hosts A1 and B1 are not configured in this example. The gateways GW-A and GW-B are assumed to be able to reach each other via public IPv4 addresses.

Server Configuration

In this setup *GW-A* is the OpenVPN server node. To configure the server side of the tunnel, enter the following commands in the CLI:

First, create the tunnel:

```
server:/#> configure tunnel ssl 200
Creating new SSL tunnel 200, check your settings before activating the tunnel!
server:/config/tunnel/ssl-200/#> type tun
server:/config/tunnel/ssl-200/#> method cert
server:/config/tunnel/ssl-200/#> ca-certificate server1
server:/config/tunnel/ssl-200/#> certificate server1
server:/config/tunnel/ssl-200/#> server
```

```
server:/config/tunnel/ssl-200/#> client-to-client
server:/config/tunnel/ssl-200/#> leave
```

Then add a static ip address to the server's end of the tunnel:

```
server:/#> configure iface ssl200
server:/config/iface-ssl200/#> no inet
server:/config/iface-ssl200/#> inet static 192.168.254.1/24
server:/config/iface-ssl200/inet-static-192.168.254.1/#> leave
```

Then create a client configuration:

```
server:/#> configure tunnel ssl 200
server:/config/tunnel/ssl-200/#> client-config 1
Creating new client config: 1!
server:/config/tunnel/ssl-200/client-config-1/#> common-name client1
server:/config/tunnel/ssl-200/client-config-1/#> address 192.168.254.21/24
server:/config/tunnel/ssl-200/client-config-1/#> iroute 20.60.0.0/24
server:/config/tunnel/ssl-200/client-config-1/#> end
server:/config/tunnel/ssl-200/#> push-network 10.70.0.0/24
server:/config/tunnel/ssl-200/#> leave
```

We have created a configuration for a new client, we have assigned a static ip address to its end of the tunnel (192.168.254.21), we have pushed the server's network to the client (10.70.0.0/24), and we have also configured the tunnel to know about the internal routes to network behind the client's node (20.60.0.0/24).

The final step is to add the routes to the client's network.

```
server:/#> configure ip
server:/config/ip/#> route 20.60.0.0/24 ssl200
server:/config/ip/#> leave
```

Now we can check the tunnel and ssl interface status:

```
server:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
200      ssl200          Down

server:/#> sh iface ssl200
Name         : ssl200
Admin. State : Up
Oper. Status : DOWN
MAC Address  : N/A
IP Addresses : 192.168.254.1/24   (static)
MTU          : 1500
Statistics   :
               RX: Packets    Bytes        Errors CsumErrs OutOfSeq Mcasts
                   0          0            0      0        0        0
               TX: Packets    Bytes        Errors DeadLoop NoRoute  NoBufs
                   0          0            0      0        0        0
```

The tunnel as DOWN, which is expected. This changes to UP once a connection with a client has been established.

Client Configuration

GW-B is the OpenVPN client node. To configure the client side of the tunnel, enter the following commands in the CLI:

```
client:/#> configure tunnel ssl 200
Creating new SSL tunnel 200, check your settings before activating the tunnel!
client:/config/tunnel/ssl-200/#> type tun
client:/config/tunnel/ssl-200/#> method cert
client:/config/tunnel/ssl-200/#> ca-certificate client1
client:/config/tunnel/ssl-200/#> certificate client1
client:/config/tunnel/ssl-200/#> no server
client:/config/tunnel/ssl-200/#> peer 192.168.0.21
client:/config/tunnel/ssl-200/#> leave
```

The IP address for the peer is the public IP address of the server.

Again, we can check the tunnel and ssl interface status:

```
client:/#> show tunnel ssl
TUNNEL   DESCRIPTION     STATUS   UPTIME
200      ssl200          UP       0 Days 0 Hours 0 Mins 10 Secs

client:/#> sh iface ssl200
Name         : ssl200
Admin. State : Up
Oper. Status : UP
MAC Address  : N/A
IP Addresses : 192.168.254.21/24  (UNKNOWN)
MTU          : 1500
Statistics   :
               RX: Packets    Bytes        Errors CsumErrs OutOfSeq Mcasts
                   0          0            0      0        0        0
               TX: Packets    Bytes        Errors DeadLoop NoRoute  NoBufs
                   0          0            0      0        0        0
```

Verifying connectivity

From host A1 to host B1:

```
a1:/#> ping count 5 20.60.0.2
Press Ctrl-C to abort PING 20.60.0.2 (20.60.0.2): 56 data bytes
64 bytes from 20.60.0.2: seq=0 ttl=62 time=5.482 ms
64 bytes from 20.60.0.2: seq=1 ttl=62 time=5.229 ms
64 bytes from 20.60.0.2: seq=2 ttl=62 time=5.225 ms
64 bytes from 20.60.0.2: seq=3 ttl=62 time=2.512 ms
64 bytes from 20.60.0.2: seq=4 ttl=62 time=5.215 ms

--- 20.60.0.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 2.512/4.732/5.482 ms
a1:/#>
```

From host B1 to host A1:

```
b1:/#> ping count 2 10.70.0.2
Press Ctrl-C to abort PING 10.70.0.2 (10.70.0.2): 56 data bytes
64 bytes from 10.70.0.2: seq=0 ttl=62 time=4.953 ms
```

```
64 bytes from 10.70.0.2: seq=1 ttl=62 time=5.440 ms

--- 10.70.0.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 4.953/5.196/5.440 ms
```

NOTE

Do not forget to add the route from host B1 to the tunnel so that the ping replies are routed correctly.

```
b1:/#> configure ip route 192.168.254.0/24 20.60.0.1
b1:/config/#> leave
```

Final Topology

The following figure illustrates the resulting topology, with an L3 tunnel connecting the two LANs via the virtual interface ssl200 on each GW device.

```
          Site A                                  Site B

        .--------.                              .--------.
        |        |                              |        |
        |  Host  |                              |  Host  |
        |   A1   |                              |   B1   |
        '----+---'                              '---+----'
             | IP:10.70.0.2                         | IP:20.60.0.2
             |                                      |
             |                                      |
             |                                      |
        eth1 | IP:10.70.0.1                    eth1 | IP:20.60.0.1
        .----+---.                              .---+----.
        |     `. |         OpenVPN Tunnel       | .´     |
        |  GW-A `+==============================+´ GW-B  |
        |        | ssl200                ssl200 |        |
        |        |192.168.254.1   192.168.254.21|        |
        '----+---'                              '---+----'
      vlan2  |                .--.-.                | vlan2
  Public IP: |               ( (    )__             | Public IP:
192.168.0.21 '--------------(_,  \ ) ,_)------------' 192.168.0.22
                              '-'--`--'
                     Public Network/Internet
```

4.10.4. IPsec Site-To-Site

About

This is a configuration example of an IPsec tunnel in a *Site-To-Site* scenario. In this setup, two separate networks are securely connected over the Internet using dedicated VPN gateways. The configuration demonstrates how to establish a reliable and encrypted communication channel between the sites.

Introduction

A common use case for IPsec VPNs is to connect two networks via a secure tunnel over the Internet. We refer to this scenario as *Site-To-Site* VPN. It is accomplished by having two VPN gateways, one at each site, negotiate and establish a secure tunnel, and to forward all traffic between the two networks through this tunnel. By creating VPN tunnels you establish a secure overlay network on top of your regular Internet connections.

```
                                  .--.-.
                                 ( (    )__
                                (_,  \ ) ,_)  Internet/Public Network
                                  '-'--`--'
                                    |    |
                     .------------------'    '--------------------.
                     |                                            |
                     |Public IP:                                  |Public IP:
                     |192.168.0.1                                 |192.168.0.2
                 .---+----.                                   .---+----.
                 |  Alice |           IPsec Tunnel             |  Bob   |
                 |  GW-1  |===================================|  GW-2  |
                 |        |Responder                 Initiator|        |
                 '---+----'                                   '---+----'
                     |.1     (Site A)           (Site B)          |.1
                     |     10.1.0.0/16         10.2.0.0/16        |
           ---+------+---+-------------         ---+----------+----+----
              |          |                         |          |
              |.11       |.5                       |.11       |.5
            .-+--.     .-+--.                   .-+--.     .-+--.
            | H1 |     | H2 |                   | H3 |     | H4 |
            '----'     '----'                   '----'     '----'
```

Figure 1: Site-To-Site tunnel using IPsec.

In Figure 1 Alice is the responder and Bob is the initiator. They both configured their identities to allow for IKE authentication. Each peer defined the local and remote subnet. They also use a pre shared key chosen randomly with sufficient entropy.

To secure all traffic between Site-A and B, Alice would define 10.1.0.0/16 as local subnet, and 10.2.0.0/16 as remote subnet in the tunnel configuration. Bob would do the opposite, i.e., define 10.2.0.0/16 as local subnet, and 10.1.0.0/16 as remote subnet. Note that when subnets don't match IKE might be established (Phase 1) but the tunnel won't be created. In certain scenarios, it is possible to establish a tunnel where the remote subnet of one peer resides within the local subnet of another. In such cases, only the traffic that falls within the shared IP address range of the two subnets will be secured.

Responder Setup

```
alice:/#> config
alice:/config/#> tunnel
alice:/config/tunnel/#> ipsec 1
alice:/config/tunnel/ipsec-1/#> description "Site-to-site"

alice:/config/tunnel/ipsec-1/#> local-address 192.168.0.1
```

```
alice:/config/tunnel/ipsec-1/#> local-subnet 10.1.0.0/16
alice:/config/tunnel/ipsec-1/#> local-id alice

alice:/config/tunnel/ipsec-1/#> peer 192.168.0.2
alice:/config/tunnel/ipsec-1/#> remote-subnet 10.2.0.0/16
alice:/config/tunnel/ipsec-1/#> remote-id bob

alice:/config/tunnel/ipsec-1/#> secret This_is_a_strong_sec$%,^@d
alice:/config/tunnel/ipsec-1/#> leave
```

Initiator Setup

```
bob:/#> config
bob:/config/#> tunnel
bob:/config/tunnel/#> ipsec 1
bob:/config/tunnel/ipsec-1/#> description "Site-to-site"
bob:/config/tunnel/ipsec-1/#> initiator

bob:/config/tunnel/ipsec-1/#> local-address 192.168.0.2
bob:/config/tunnel/ipsec-1/#> local-subnet 10.2.0.0/16
bob:/config/tunnel/ipsec-1/#> local-id bob

bob:/config/tunnel/ipsec-1/#> peer 192.168.0.1
bob:/config/tunnel/ipsec-1/#> remote-subnet 10.1.0.0/16
bob:/config/tunnel/ipsec-1/#> remote-id alice

bob:/config/tunnel/ipsec-1/#> secret This_is_a_strong_sec$%,^@d

bob:/config/tunnel/ipsec-1/#> leave
```

Verification

After applying the configuration Alice can check if Bob succeeded with connection:

```
alice:/#> show tunnel ipsec
ID  DESCRIPTION      STATUS              TIME (since last rekey)
1   ipsec1           Up(Installed)       0 Days 1 Hours 8 Mins 39 Secs
```

```
alice:/#> show tunnel ipsec 1
Current state for tunnel ipsec1 is UP(Installed).

Security-Association(SA) details:
ipsec1: #6, ESTABLISHED, IKEv2, 20759114c78bf131_i* 4fb9e973f3067b48_r
  local  'alice' @ 192.168.0.1[500]
  remote 'bob' @ 192.168.0.2[500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/ECP_256
  established 5s ago, rekeying in 14304s
  tunnel1: #5, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA2_256_128
    installed 5s ago, rekeying in 3267s, expires in 3955s
    in  c2032981,      0 bytes,     0 packets
    out cc173955,      0 bytes,     0 packets
    local  192.168.0.1/32
    remote 192.168.0.2/32

Loaded configuration for the connection:
ipsec1: IKEv2, no reauthentication, rekeying every 14400s, dpd delay 30s
  local:  192.168.0.1
```

```
  remote: 192.168.0.2
  local pre-shared key authentication:
    id: alice
  remote pre-shared key authentication:
    id: bob
  tunnel1: TUNNEL, rekeying every 3600s, dpd action is start
    local:  dynamic
    remote: dynamic
```

4.10.5. IPsec Host-To-Host

About

This is a minimal configuration example of an IPsec tunnel in a *Host-To-Host* scenario. It demonstrates how to securely connect two individual hosts over an untrusted network using a pre-shared key. This setup is ideal for environments where only direct communication between two endpoints is required, without exposing additional subnets or devices.

Introduction

In this minimal setup, host-2-host communication is enabled. In Figure 1 Alice is the responder, while Bob is the initiator. Both peers configure their respective local and peer IP addresses. Since neither of them exposes a subnet behind their addresses, only traffic between the two hosts will be secured and allowed through the tunnel. They use a pre shared key chosen randomly with sufficient entropy.

```
                        .--.-.
                       ( (    )__
                      (_,  \ ) ,_)  Internet/Public Network
                        '-'--`--'
                          |    |
    .------------------'    '--------------------.
    |                                             |
    |Public IP:                                   |Public IP:
    |192.168.0.1                                  |192.168.0.2
.---+----.                                    .---+----.
|        |           IPsec Tunnel             |        |
| Alice  |=====================================|  Bob   |
|        | Responder               Initiator  |        |
'--------'                                    '--------'
```

Figure 1: Host-To-Host tunnel using IPsec.

Responder Setup

```
alice:/#> config
alice:/config/#> tunnel
alice:/config/tunnel/#> ipsec 10
alice:/config/tunnel/ipsec-1/#> description "Alice to Bob"
alice:/config/tunnel/ipsec-1/#> local-address 192.168.0.1
alice:/config/tunnel/ipsec-1/#> peer 192.168.0.2
alice:/config/tunnel/ipsec-1/#> secret This_is_a_strong_sec$%,^@d
```

Initiator Setup

```
bob:/#> config
bob:/config/#> tunnel
bob:/config/tunnel/#> ipsec 10
bob:/config/tunnel/ipsec-1/#> description "Bob to Alice"
bob:/config/tunnel/ipsec-1/#> initiator
bob:/config/tunnel/ipsec-1/#> local-address 192.168.0.2
bob:/config/tunnel/ipsec-1/#> peer 192.168.0.1
bob:/config/tunnel/ipsec-1/#> secret This_is_a_strong_sec$%,^@d
```

Verification

Both Alice and Bob omitted configuring IKE identities, resulting in the substitution of local and remote IDs with the local address and peer address, respectively. Additionally, a default subnet of /32 is employed for traffic selection.

```
alice:/#> show tunnel ipsec 10
Current state for tunnel ipsec10 is UP(Installed).

Security-Association(SA) details:
ipsec10: #6, ESTABLISHED, IKEv2, 20759114c78bf131_i* 4fb9e973f3067b48_r
  local  'alice' @ 192.168.0.1[500]
  remote 'bob' @ 192.168.0.2[500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/ECP_256
  established 5s ago, rekeying in 14304s
  tunnel1: #5, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA2_256_128
    installed 5s ago, rekeying in 3267s, expires in 3955s
    in  c2032981,      0 bytes,     0 packets
    out cc173955,      0 bytes,     0 packets
    local  192.168.0.1/32
    remote 192.168.0.2/32

Loaded configuration for the connection:
ipsec10: IKEv2, no reauthentication, rekeying every 14400s, dpd delay 30s
  local:  192.168.0.1
  remote: 192.168.0.2
  local pre-shared key authentication:
    id: alice
  remote pre-shared key authentication:
    id: bob
  tunnel1: TUNNEL, rekeying every 3600s, dpd action is start
    local:  dynamic
    remote: dynamic
```

4.10.6. IPsec in Legacy Mode (IKEv1)

About

This document provides a configuration example for establishing an IPsec tunnel in a *Site-to-Site* scenario between WeOS 5 and WeOS 4 devices using the IKEv1 version of the IPsec protocol. It is intended for users who need to interoperate with legacy WeOS 4 installations or third-party devices that only support IKEv1.

Introduction

Internet Key Exchange version 1 (IKEv1) is the original protocol for negotiating IPsec security associations. While IKEv2 is recommended for new deployments due to its improved security and efficiency, IKEv1 is still required for compatibility with older devices and some third-party equipment.

Site-to-Site Scenario

A typical site-to-site VPN connects two separate networks securely over the Internet. For example, a company may want to connect two offices using a tunnel, where Site A uses a new WeOS 5 device and Site B uses an older WeOS 4 device as their respective gateways. In this example, WeOS 5 acts as the initiator and WeOS 4 as the responder. Each site has its own local subnet, and the tunnel is established between two VLAN interfaces.

Network Topology

```
  [Site A: WeOS 5]                      [Site B: WeOS 4]
192.168.15.1/24 (vlan1000)          192.168.45.1/24 (vlan1000)
10.0.1.15/24   (vlan100)            10.0.1.45/24   (vlan100)
          |                                |
          |--------- IPsec Tunnel----------|
          |             (IKEv1)            |
```

Figure 1: Site-to-site tunnel using IPsec IKEv1.

Both gateways have two interfaces configured: first, `vlan1000`, connected to the respective internal subnet, and second, `vlan100`, exposed to the outside network.

The IPsec tunnel (using IKEv1, supported by both versions of WeOS) securely connects the two sites over the Internet or another untrusted network. This allows devices on the `192.168.15.1/24` network at Site A to communicate securely with devices on the `192.168.45.1/24` network at Site B, as if they were on the same local network.

Configuration

WeOS 5 Configuration (Initiator)

Enter configuration mode and set up the VLANs and interfaces. Then configure the IPsec tunnel as follows:

```
siteA:/conf/#> tunnel ipsec 1
siteA:/conf/tunnel/ipsec-1/#> initiator
siteA:/conf/tunnel/ipsec-1/#> version 1
siteA:/conf/tunnel/ipsec-1/#> peer 10.0.1.45
siteA:/conf/tunnel/ipsec-1/#> local-address 10.0.1.15
siteA:/conf/tunnel/ipsec-1/#> remote-subnet 192.168.45.0/24
siteA:/conf/tunnel/ipsec-1/#> local-subnet 192.168.15.0/24
siteA:/conf/tunnel/ipsec-1/#> ike auth sha256 cipher aes256 dh modp2048
siteA:/conf/tunnel/ipsec-1/#> esp auth sha256 cipher aes256 dh modp2048
siteA:/conf/tunnel/ipsec-1/#> secret westermo
siteA:/conf/tunnel/ipsec-1/#> leave
```

Explanation:

- `initiator` specifies that this device will initiate the tunnel.
- `version 1` selects IKEv1.
- `peer` is the IP address of the remote WeOS 4 device.
- `local-address` and `local-subnet` define the local interface and protected network.
- `remote-subnet` is the network behind the peer.
- The `ike` and `esp` lines set the cryptographic parameters. Note that you cannot leave them at the default `Auto` setting because WeOS 4 does not support that; therefore, select the most secure proposal supported.
- `secret` sets the pre-shared key.

WeOS 4 Configuration (Responder)

On the WeOS 4 device, configure the VLANs, interfaces, and IPsec tunnel. Use the provided Management Guide for additional help. Below are the essential commands for IPsec configuration:

```
siteB:/conf/#> tun ip 1
siteB:/conf/tunnel/ipsec-1/#> peer 10.0.1.15
siteB:/conf/tunnel/ipsec-1/#> outbound vlan100
siteB:/conf/tunnel/ipsec-1/#> remote-subnet 192.168.15.0/24
siteB:/conf/tunnel/ipsec-1/#> local-subnet 192.168.45.0/24
siteB:/conf/tunnel/ipsec-1/#> secret westermo
siteB:/conf/tunnel/ipsec-1/#> leave
```

Explanation:

- `peer` is the WeOS 5 device.
- `outbound vlan100` specifies the outgoing interface for the tunnel.
- `remote-subnet` and `local-subnet` define the protected networks.
- `secret` must match the pre-shared key on WeOS 5.

Verification

After applying the configuration, you can verify the tunnel status on both devices.

On WeOS 5:

```
siteB:/#> sh tun ipsec
ID  DESCRIPTION      STATUS              TIME (since last rekey)
1   ipsec1           Up(Installed)       5 min 25 sec

siteB:/#> sh tun ipsec 1
Current state for tunnel ipsec1 is UP(Installed).

Security-Association(SA) details:
ipsec1: #33834, ESTABLISHED, IKEv1, d3309517c6364582_i* 4e6a8ddc2932d8ce_r
  local  '10.0.1.15' @ 10.0.1.15[500]
  remote '10.0.1.45' @ 10.0.1.45[500]
  AES_CBC-256/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_2048
  established 327s ago, rekeying in 13097s
  tunnel1: #55, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-256/HMAC_SHA2_256_128/MODP_2048
    installed 327s ago, rekeying in 3173s, expires in 3633s
    in  ccc4b9b2,      0 bytes,     0 packets
    out ba951356,      0 bytes,     0 packets
    local  192.168.15.0/24
    remote 192.168.45.0/24

Loaded configuration for the connection:
```

```
ipsec1: IKEv1, reauthentication every 14400s, dpd delay 30s
  local:  10.0.1.15
  remote: 10.0.1.45
  local pre-shared key authentication:
    id: 10.0.1.15
  remote pre-shared key authentication:
    id: 10.0.1.45
  tunnel1: TUNNEL, rekeying every 3600s, dpd action is start
    local:  192.168.15.0/24
    remote: 192.168.45.0/24
```

The output should indicate that the tunnel is “Up” and “Installed,” confirming successful negotiation and establishment of the IPsec tunnel.

Similarly verify the status on WeOS 4:

```
siteA:/#> show tunnel ipsec
```

Troubleshooting

- Ensure that the pre-shared key (`secret`) matches on both devices.
- Verify that the local and remote subnets are correctly configured and do not overlap incorrectly.
- Check that the cryptographic parameters (cipher, auth, DH group) are supported and match on both sides.
- If the tunnel does not come up, review the system logs for errors related to IKE negotiation.

Summary

This guide demonstrates how to configure a site-to-site IPsec tunnel between WeOS 5 and WeOS 4 using IKEv1. While IKEv1 is considered legacy, it is still required for interoperability with older devices. For new deployments, consider using IKEv2 for improved security and reliability.

4.10.7. IPsec with Certificates

About

This is a configuration example of an IPsec tunnel that uses X.509 certificates for authentication instead of pre-shared keys. In this setup, two devices are securely connected over the Internet using an IPsec tunnel. Each device is provisioned with its own end-entity certificate signed by a trusted CA certificate. The configuration demonstrates how to establish a reliable and encrypted communication channel using certificate-based authentication.

Introduction

In this example, the devices use IKEv2 with certificate-based authentication. Each peer presents its own certificate and validates the peer certificate using a trusted CA certificate. Identities configured on the tunnel are matched against the subject (or subjectAltName) of the presented certificates.

The topology is the same as in the PSK-based example:

```
                          .--.-.
                         ( (    )__
                        (_,  \ ) ,_)  Internet/Public Network
                          '-'--`--'
                           |    |
        .------------------'    '--------------------.
        |                                            |
        |Public IP:                                  |Public IP:
        |192.168.0.1                                 |192.168.0.2
    .---+----.                                   .---+----.
    |  Alice |           IPsec Tunnel            |  Bob   |
    |        |===================================|        |
    |        |Responder                 Initiator|        |
    '--------'                                   '--------'
```

Figure 1: IPsec tunnel using certificate-based authentication.

```
   Alice PKI                                   Bob PKI
+----------------------+                  +----------------------+
| Root CA (root-a.pem) |                  | Root CA (root-b.pem) |
+------+---------------+                  +------------+---------+
       |                                               |
       v                                               v
+-------------------------+            +-------------------------+
| Issuing CA (int-a.pem)  |            | Issuing CA (int-b.pem)  |
+------+------------------+            +--------------+----------+
       |                                               |
       v                                               v
+--------------------+                     +--------------------+
| alice-cert.pem (EE)|                     | bob-cert.pem (EE)  |
| CN=alice           |                     | CN=bob             |
+--------------------+                     +--------------------+
```

Figure 2: Alice and Bob are using two separate PKI hierarchies. They need to import the root CA certificate of their respective peer before establishing connection.

In Figure 1, Alice is the responder and Bob is the initiator. They both configured their identities to allow for IKE authentication. Instead of a shared secret, each peer is configured with a local certificate and a trusted CA certificate used to validate peer's certificate.

Before configuring the tunnels, make sure that:

- Each peer has a local certificate (with private key) installed (imported or enrolled) in the device certificate store.
- Both peers trust the CA that issued the peer certificates; the CA certificate is installed in the Local CA Store.

In our example, Alice must have at least the following certificates installed. The “A-chain” (root-a, int-a, alice) will be sent to Bob. Root of “B-chain” will validate Bob’s chain.

```
alice:/pki/#> show cert full

Certificates
TYPE     HASH     EXPIRES     NAME                 DIST LABEL
CA-local 9db57dbe Sep 21 2035 Root-A CA                 root-a
CA-local 8a09dd4b Sep 21 2028 Issuing-A CA              int-a
CA-local 36a0943a Sep 22 2035 Root-B CA                 root-b
Pub      db1c90fc Dec 10 2026 alice                     alice
Key      N/A      N/A         alice                N/A  alice
```

Responder Setup (Alice)

In this example, Alice is configured as the responder. She uses a certificate with label `alice-cert` signed by a issuing CA, `issuing-a` signed by a Root-CA certificate with label `root-a`. We assume that all certificates from Alice’s PKI chain and Bob’s Root-CA `root-b`, have already been imported or enrolled to the system.

```
alice:/#> config
alice:/config/#> tunnel
alice:/config/tunnel/#> ipsec 1
alice:/config/tunnel/ipsec-1/#> description "Site-to-site"

alice:/config/tunnel/ipsec-1/#> local-address 192.168.0.1
alice:/config/tunnel/ipsec-1/#> peer 192.168.0.2
alice:/config/tunnel/ipsec-1/#> local-id CN=alice
alice:/config/tunnel/ipsec-1/#> remote-id CN=bob

alice:/config/tunnel/ipsec-1/#> method cert
alice:/config/tunnel/ipsec-1/#> certificate alice-cert
alice:/config/tunnel/ipsec-1/#> ca root-b

alice:/config/tunnel/ipsec-1/#> leave
```

The `method cert` command selects certificate-based authentication. The `certificate` command selects which local certificate (and private key) to use for this tunnel. The `ca` command selects the CA certificate that is used to validate the certificate chain presented by the peer. **NOTE**: the CA certificate must be a standalone file (not a bundle). The `local-id` and `remote-id` values configured on the tunnel must match exactly the identities in the certificates (for example, the certificate subject).

In this example, the revocation policy is kept default, `relaxed`, meaning that the tunnel will not be established if Bob’s certificate is explicitly revoked in a CRL. In other cases, like CRLDP not available the revocation checking will not be

Initiator Setup (Bob)

Bob is configured as the initiator. He uses a certificate with label `bob-cert` signed `issuer-b` signed by `root-b`. Again, all certificates must be present in the system.

```
bob:/#> config
bob:/config/#> tunnel
bob:/config/tunnel/#> ipsec 1
bob:/config/tunnel/ipsec-1/#> description "Site-to-site (cert)"
bob:/config/tunnel/ipsec-1/#> initiator

bob:/config/tunnel/ipsec-1/#> local-address 192.168.0.2
```

```
bob:/config/tunnel/ipsec-1/#> peer 192.168.0.1
bob:/config/tunnel/ipsec-1/#> local-id CN=bob
bob:/config/tunnel/ipsec-1/#> remote-id CN=alice

bob:/config/tunnel/ipsec-1/#> method cert
bob:/config/tunnel/ipsec-1/#> certificate bob-cert
bob:/config/tunnel/ipsec-1/#> ca root-a

bob:/config/tunnel/ipsec-1/#> leave
```

Again, `method cert` enables certificate-based authentication on the tunnel, `certificate` selects the local certificate to present to the peer, and `ca` selects the CA certificate used to validate the remote certificate. The revocation policy is `relaxed`.

Revocation

In context of IPsec the CRL revocation checking is always enabled and does not require any additional configuration. It relies on CRL Distribution Points defined in the incoming certificate chain. Revocation cannot be disabled instead one of the two policies can be selected: *strict* ro *relaxed* (default).

Revocation status checking is done during initial authentication phase and at every next re-authentication attempt. The `reauth-time` parameter controls how often the IKE_SA is re-authenticated and the peer certificate and revocation status are revalidated. Please align re-authentication time to CRL refresh time. An example alignment is `0.5 x (CRL-refresh-time)`.

Given that CRL updates every week, the following example shows how to configure IKE re-authentication to 3 days on Alice. Bob should use compatible values.

```
alice:/config/tunnel/ipsec-1/#> reauth-time 3d
```

Strict Revocation

If you need a stricter, security-first policy, enable `revocation strict` on both peers. With strict mode:

- Authentication fails if a current CRL cannot be fetched, its signature is invalid, or the CRL is expired.
- The tunnel will not be established without valid, up-to-date revocation data.

Before enabling strict mode, ensure:

- Reliable network access to CRL distribution points.
- Accurate system time via NTP.
- Appropriate CRL cache lifetimes aligned with the CRL refresh schedule.
- Regular re-authentication (reauth-time) set to a value compatible with CRL updates.

Verification

After applying the configuration, Alice can check if Bob succeeded with connection:

```
alice:/#> show tunnel ipsec
ID  DESCRIPTION      STATUS              TIME (since last rekey)
1   ipsec1           Up(Installed)       0 Days 1 Hours 8 Mins 39 Secs
```

```
alice:/#> show tunnel ipsec 1
Current state for tunnel ipsec1 is UP(Installed).

Security-Association(SA) details:
ipsec1: #6, ESTABLISHED, IKEv2, 20759114c78bf131_i* 4fb9e973f3067b48_r
  local  'alice' @ 192.168.0.1[500]
  remote 'bob' @ 192.168.0.2[500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/ECP_256
  established 5s ago, rekeying in 14304s
  tunnel1: #5, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA2_256_128
    installed 5s ago, rekeying in 3267s, expires in 3955s
    in  c2032981,      0 bytes,     0 packets
    out cc173955,      0 bytes,     0 packets
    local  192.168.0.1/32
    remote 192.168.0.2/32

Loaded configuration for the connection:
ipsec1: IKEv2, reauthentication every 14400s, rekeying every 14400s, dpd delay 30s
  local:  192.168.0.1
  remote: 192.168.0.2
  local certificate authentication:
    id: alice, certificate: alice-cert, ca: ipsec-ca
  remote certificate authentication:
    id: bob, ca: ipsec-ca
  tunnel1: TUNNEL, rekeying every 3600s, dpd action is start
    local:  dynamic
    remote: dynamic
```

If the tunnel does not come up, check the following:

- The local and remote subnets match between the peers.
- The `local-id` and `remote-id` values correspond to the identities in the certificates.
- The correct local certificate and CA certificate labels are configured.
- The revocation policy is suitable for your environment and valid revocation information is available when using `revocation strict`.
- Check that when using a multi-tier certificate hierarchy all certificates from the chain, including intermediates are present in the system.

4.10.8. IPsec Through a Firewall

About

This guide explains how to configure the WeOS firewall to allow IPsec traffic. Two common scenarios are covered: a WeOS device that acts as both an IPsec gateway and a firewall, and a WeOS device that acts as an intermediate firewall between two IPsec peers.

Introduction

IPsec relies on specific protocols and ports to establish and maintain secure tunnels. When a firewall is present on the path between two IPsec peers, it must be configured to permit the following traffic:

Traffic	Protocol	Port	Purpose
IKE	UDP	500	Key exchange and tunnel setup
NAT-Traversal	UDP	4500	IPsec through NAT devices
ESP	proto 50	—	Encrypted tunnel data

If any of these are blocked, the tunnel will fail to establish or will drop unexpectedly.

> **NOTE**
>
> NAT-Traversal (NAT-T) automatically kicks in when the peers detect a NAT device between them. It encapsulates ESP inside UDP port 4500. Always allow UDP 4500 alongside UDP 500 unless you are certain that no NAT is present.

> **NOTE**
>
> IPsec traffic is exempted from NAT when the device acts as both the IPsec gateway and NAT gateway at the same time.

Scenario 1: IPsec Gateway with Integrated Firewall

In this scenario the WeOS device acts as both the IPsec VPN gateway and runs the firewall. The firewall must allow inbound IKE and ESP on the WAN interface so that the local IPsec daemon can receive and respond to tunnel negotiations.

```
                                  .--.-.
                                ( (    )__
                               (_,  \ ) ,_)  Internet/Public Network
                                 '-'--`--'
                                   |    |
                     .------------------'    '--------------------.
                     |                                            |
                     |WAN (vlan2):                                |WAN (vlan2):
                     |192.168.0.1                                 |192.168.0.2
                 .---+-------.                             .----+------.
                 |   Alice   |         IPsec Tunnel        |    Bob    |
                 | Responder |=============================| Initiator |
                 |  GW + FW  |                             |  GW + FW  |
                 '-----------'                             '-----------'
                     |LAN (vlan1):                                |LAN (vlan1):
                     |10.1.0.1/16                                 |10.2.0.1/16
                     |                                            |
           ---+-----+---+---                          ---+------+---+---
              |         |                                |         |
            .-+--.    .-+--.                          .-+--.    .-+--.
            | H1 |    | H2 |                          | H3 |    | H4 |
            '----'    '----'                          '----'    '----'
```

Figure 1: Each gateway runs both the IPsec daemon and the firewall. Firewall INPUT rules on the WAN interface are required to allow IKE and ESP.

Firewall Configuration on Alice

Configure the firewall to allow the IPsec control and data plane traffic on the WAN interface (`vlan2`). Setting both policies to `drop` ensures that only explicitly permitted traffic is accepted.

```
alice:/#> configure
alice:/config/#> ip firewall
alice:/config/ip/firewall/#> input accept in vlan2 dport 500 proto udp comment "IKE"
alice:/config/ip/firewall/#> input accept in vlan2 proto esp comment "IPsec ESP"
alice:/config/ip/firewall/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#> copy run start
```

```
bob:/#> configure
bob:/config/#> ip firewall
bob:/config/ip/firewall/#> input accept in vlan2 proto esp comment "IPsec ESP"
bob:/config/ip/firewall/#> input accept saddr 10.1.0.0/16 proto icmp comment "IPsec ESP"
bob:/config/ip/firewall/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
bob:/#> copy run start
```

Explanation:

- `input accept in vlan2 dport 500 proto udp` allows IKE negotiations to reach the local IPsec daemon running on the responder.
- `input accept in vlan2 proto esp` allows ESP packets to reach the local IPsec daemon. This is required for the tunnel to function on both sides.
- `input accept saddr 10.1.0.0/16 proto icmp` allows ICMP responses from Alice's LAN to reach Bob and allow him to respond. This is optional but useful for testing connectivity. This is needed because Bob's firewall is configured to drop all other inbound traffic by default, so without this rule the ping responses from Alice's LAN would be blocked.

IPsec Tunnel Configuration

Configure the IPsec tunnel as described in IPsec Site-To-Site. The firewall rules above are independent of the tunnel configuration.

Verification

Check that the tunnel comes up and that the firewall is not dropping IKE or ESP packets:

```
alice:/#> show tunnel ipsec
ID  DESCRIPTION      STATUS              TIME (since last rekey)
1   ipsec1           Up(Installed)       0 Days 0 Hours 2 Mins 10 Secs
```

```
alice:/#> ping 10.2.0.1
Press Ctrl-C to abort PING 10.2.0.1 (10.2.0.1) 56(84) bytes of data.
64 bytes from 10.2.0.1: icmp_seq=1 ttl=64 time=0.645 ms
64 bytes from 10.2.0.1: icmp_seq=2 ttl=64 time=0.412 ms
64 bytes from 10.2.0.1: icmp_seq=3 ttl=64 time=0.490 ms
^C
--- 10.2.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 0.412/0.515/0.645/0.096 ms
alice:/#>
```

Scenario 2: Standalone Firewall Between IPsec Peers

In this scenario the WeOS device acts as a dedicated firewall/router that sits between an IPsec gateway and the Internet. The firewall itself does not terminate IPsec; it only relays IKE and ESP traffic between the peers. PORT-FORWARD rules are required to pass IKE and ESP through.

```
                                  .---+----.
                     .-----------------+  WeOS  +------------------.
                     |     LAN (vlan1) |Firewall| WAN (vlan2)      |
                     |     192.168.1.1 '--------' 61.123.187.141   |
                     |                                             |
                     |                                             |
                     |WAN (vlan2):                                 |WAN (vlan2):
                     |192.168.1.2                                  |61.123.234.102
                .----+----.                                    .----+----.
                |  Alice  |         IPsec Tunnel               |   Bob   |
                |Responder|===================================|Initiator|
                '----+----'                                    '---------'
                     |10.1.0.1/16 (vlan1)                          |10.2.0.1/16 (vlan1)
                     |                                             |
           ---+-----+---+---                             ---+------+---+---
              |         |                                   |          |
            .-+--.    .-+--.                              .-+--.     .-+--.
            | H1 |    | H2 |                              | H3 |     | H4 |
            '----'    '----'                              '----'     '----'
```

Figure 2: A dedicated WeOS firewall forwards IPsec traffic between the two gateways. A tunnel is established between Alice and Bob, but the firewall does not terminate it. PORT-FORWARD rules are required to allow IKE and ESP to pass through the firewall.

Firewall Configuration on the Intermediate Firewall

The intermediate firewall needs PORT-FORWARD rules to pass IKE, NAT-T, and ESP between Alice's gateway (`192.168.0.2`) and Bob's gateway (`192.168.1.2`).

```
fw:/#> configure
fw:/config/#> ip firewall
fw:/config/ip/firewall/#> forward accept in vlan1 out vlan2
fw:/config/ip/firewall/#> forward accept in vlan2 out vlan1 dport 500 proto udp
fw:/config/ip/firewall/#> forward accept in vlan2 out vlan1 dport 4500 proto udp
fw:/config/ip/firewall/#> forward accept in vlan2 out vlan1 proto esp
fw:/config/ip/firewall/#> nat out vlan2
```

```
fw:/config/ip/firewall/#> port-forward in vlan2 to-daddr 192.168.1.2 dport 500 proto udp
fw:/config/ip/firewall/#> port-forward in vlan2 to-daddr 192.168.1.2 dport 4500 proto udp
fw:/config/ip/firewall/#> port-forward in vlan2 to-daddr 192.168.1.2 proto esp
fw:/config/ip/firewall/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
fw:/#> copy run start
```

Explanation:

- `forward accept in vlan1 out vlan2` allows all traffic from the internal LAN (`vlan1`) to be forwarded toward the WAN (`vlan2`). This covers Alice's gateway responding to Bob.
- `forward accept in vlan2 out vlan1 dport 500 proto udp` allows Bob's IKE requests to reach Alice's gateway.
- `forward accept in vlan2 out vlan1 dport 4500 proto udp` allows NAT-T traffic if Bob is behind a NAT.
- `forward accept in vlan2 out vlan1 proto esp` allows ESP packets from Bob to reach Alice's gateway.
- `nat out vlan2` masquerades outbound traffic behind the firewall's WAN address, so the IPsec peers see the firewall's public IP rather than Alice's private address. When NAT is active, NAT-T (UDP 4500) will be used automatically between the peers to keep the ESP packets routable.
- `port-forward in vlan2 to-daddr 192.168.1.2 dport 500 proto udp` redirects inbound IKE (UDP 500) arriving on the WAN to Alice's gateway. This is required when the firewall performs NAT and Bob uses the firewall's public IP as the peer address.
- `port-forward in vlan2 to-daddr 192.168.1.2 proto esp` redirects inbound ESP to Alice's gateway for the same reason.

Troubleshooting

- **Tunnel stuck in `Down(Connecting)`**: The initiator cannot reach the responder. Verify that UDP 500 (and UDP 4500 if NAT is present) is permitted both inbound and outbound on all firewalls on the path.
- **Tunnel reaches `Down(IKE Established)` but no data flows**: ESP is being blocked. Confirm that `proto esp` rules exist and are not shadowed by a broader `drop` rule.
- **Tunnel establishes but drops after a short time**: NAT state entries may be timing out. Ensure UDP 4500 is allowed so NAT-T keepalives can maintain the NAT mapping. DPD keepalives also require that IKE and ESP responses can return.
- **Checking dropped packets**: Add the `log` flag to a drop rule to capture what is being silently discarded:

```
fw:/config/ip/firewall/#> log limit 100/minute
fw:/config/ip/firewall/#> forward drop log comment "catch-all log"
fw:/#> show ip firewall conntrack
```

Related Topics

- IPsec Site-To-Site
- IPsec Host-To-Host
- IPsec with Certificates
- Firewall HowTo
- IPsec Tunnelling Guide

4.10.9. WireGuard Site-To-Site

About

This is a configuration example of a WireGuard tunnel in a *Site-To-Site* scenario. In this setup, two separate networks are securely connected over a untrusted network using WireGuard VPN. The configuration demonstrates how to establish a fast, secure, and efficient communication channel between the sites.

Introduction

WireGuard provides a modern and simple approach to Site-To-Site VPN connectivity. Unlike traditional VPN solutions, WireGuard uses cryptokey routing where traffic is routed based on public keys and allowed IP addresses. This creates a secure overlay network with minimal configuration complexity.

```
                                .--.-.
                               ( (    )__
                              (_,  \ ) ,_)  Internet/Public Network
                                '-'--`--'
                                 |    |
              .------------------'    '--------------------.
              |                                            |
              |Public IP:                                  |Public IP:
              |172.16.1.1:51820                            |172.16.1.2:51820
              |Tunnel IP:                                  |Tunnel IP:
              |10.0.0.1                                    |10.0.0.2
          .---+----.                                   .---+----.
          |  Alice |        WireGuard Tunnel           |  Bob   |
          |  GW-1  |===================================|  GW-2  |
          |  wg1   |                                   |  wg1   |
          '---+----'                                   '---+----'
              |.1   (Site A)             (Site B)          |.1
              |     192.168.1.0/24       192.168.2.0/24    |
   ---+------+---+-------------        ---+----------+----+----
      |          |                         |          |
      |.11       |.5                       |.11       |.5
    .-+--.     .-+--.                   .-+--.     .-+--.
    | H1 |     | H2 |                   | H3 |     | H4 |
    '----'     '----'                   '----'     '----'
```

Figure 1: Site-To-Site tunnel using WireGuard.

In Figure 1, Alice and Bob each configure a WireGuard interface (wg1) with their own private/public key pairs. They exchange public keys and configure each other as peers. The tunnel IP addresses (10.0.0.1 and 10.0.0.2) are used for direct communication between the gateways, while the allowed IPs configuration enables routing between the site networks (192.168.1.0/24 and 192.168.2.0/24).

Prerequisites

Before configuring the tunnel, each site needs:

1. A public IP address or hostname (172.16.1.1 for Alice, 172.16.1.2 for Bob)
2. UDP port 51820 accessible through any firewalls
3. Generated WireGuard key pairs (private and public keys)

Key Generation

Generate key pairs and preshared key externally using the `wg` command-line tool.

```
$> wg genkey | tee privatekey | wg pubkey > publickey
$> wg genpsk > presharedkey
```

```
Private, Alice: OCGibrbxiumJsgHpQLb+oEl/KctdfkNVnV+sdn+ty3s=
Public, Alice: IWhTmzGIFNQOb9kGBVnuKaVQzamEhnyJkcPesrsGewU=
Private, Bob: EG8gc5SWzHqwwdmUWQm9xFb4tGd+1Uwm1Po9RjvHRko=
Public, Bob: wq0ouxXud1N0X3rYUEOTY2uWwmFA6YpFLo/PKPud9jY=
Preshared key: kHvZAEfyOj4B5Qs/z/hUwHMurmJeE1UQtXedKDEeDno=
```

Note: The keys shown above are examples. Always generate your own unique keys.

Alice Setup (Site A)

Configure Alice's WireGuard tunnel instance:

```
alice:/#> config
alice:/config/#> tunnel
alice:/config/tunnel/#> wireguard 1
alice:/config/tunnel/wireguard-1/#> description "Site-A-to-Site-B"
alice:/config/tunnel/wireguard-1/#> private-key OCGibrbxiumJsgHpQLb+oEl/KctdfkNVnV+sdn+ty3s=
alice:/config/tunnel/wireguard-1/#> listen-port 51820

alice:/config/tunnel/wireguard-1/#> peer 1
alice:/config/tunnel/wireguard-1/peer-1#> public-key wq0ouxXud1N0X3rYUEOTY2uWwmFA6YpFLo/PKPud9jY=
alice:/config/tunnel/wireguard-1/peer-1#> endpoint 172.16.1.2
alice:/config/tunnel/wireguard-1/peer-1#> listen-port 51820
alice:/config/tunnel/wireguard-1/peer-1#> allowed-ip 10.0.0.2/32
alice:/config/tunnel/wireguard-1/peer-1#> allowed-ip 192.168.2.0/24
alice:/config/tunnel/wireguard-1/peer-1#> persistent-keepalive 25
alice:/config/tunnel/wireguard-1/peer-1#> end
alice:/config/tunnel/wireguard-wg0/#> end
alice:/config/tunnel/#> end
alice:/config/#> end
```

Set IP on Alice's WireGuard interface:

```
alice:/config/iface wg1
alice:/config/iface-wg1/inet static 10.0.0.1/24
alice:/config/iface-wg1/inet-static-10.0.0.1/#>leave
```

Bob Setup (Site B)

Configure Bob's WireGuard tunnel instance:

```
bob:/#> config
bob:/config/#> tunnel
bob:/config/tunnel/#> wireguard 1
bob:/config/tunnel/wireguard-1/#> description "Site-B-to-Site-A"
bob:/config/tunnel/wireguard-1/#> private-key EG8gc5SWzHqwwdmUWQm9xFb4tGd+1Uwm1Po9RjvHRko=
bob:/config/tunnel/wireguard-1/#> listen-port 51820

bob:/config/tunnel/wireguard-1/#> peer 1
bob:/config/tunnel/wireguard-1/peer-1#> public-key IWhTmzGIFNQOb9kGBVnuKaVQzamEhnyJkcPesrsGewU=
bob:/config/tunnel/wireguard-1/peer-1#> endpoint 172.16.1.1
bob:/config/tunnel/wireguard-1/peer-1#> listen-port 51820
bob:/config/tunnel/wireguard-1/peer-1#> allowed-ip 10.0.0.1/32
bob:/config/tunnel/wireguard-1/peer-1#> allowed-ip 192.168.1.0/24
bob:/config/tunnel/wireguard-1/peer-1#> persistent-keepalive 25
bob:/config/tunnel/wireguard-1/peer-1#> end
bob:/config/tunnel/wireguard-wg0/#> end
```

```
bob:/config/tunnel/#> end
bob:/config/#> end
```

Set IP on Bob's WireGuard interface:

```
bob:/config/iface wg1
bob:/config/iface-wg1/inet static 10.0.0.2/24
bob:/config/iface-wg1/inet-static-10.0.0.2/#>leave
```

Configuration Explanation

Private Key: Each gateway's private key, used for encryption and authentication. Never share this key.

Listen Port: UDP port 51820 (default for WireGuard). Must be accessible through firewalls.

Address: The tunnel IP address for the WireGuard interface. Used for direct gateway-to-gateway communication.

Peer Public Key: The public key of the remote gateway. This identifies and authenticates the peer.

Endpoint: The remote gateway's public IP address and port. Required for both peers in a Site-To-Site setup.

Allowed IPs: Defines which IP ranges can be routed through this peer:
- `10.0.0.X/32`: The remote gateway's tunnel IP
- `192.168.X.0/24`: The remote site's network

Persistent Keepalive: Sends keepalive packets every 25 seconds to maintain NAT traversal and connection state. Essential for peers behind NAT or with stateful firewalls.

Verification

Check the tunnel status on Alice:

```
alice:/#> show tunnel wireguard

 WireGuard Tunnels
Iface  Public Key                                      Listen Port
wg1    IWhTmzGIFNQOb9kGBVnuKaVQzamEhnyJkcPesrsGewU=    51820

 WireGuard Peers
Iface  Public Key                                      Endpoint
wg1    wq0ouxXud1N0X3rYUEOTY2uWwmFA6YpFLo/PKPud9jY=    172.16.1.2:51820
```

View detailed information for a specific interface:

```
alice:/#> show tunnel wireguard wg1
Iface      : wg1
Public key : IWhTmzGIFNQOb9kGBVnuKaVQzamEhnyJkcPesrsGewU=
Listen port: 51820

Peer                : wq0ouxXud1N0X3rYUEOTY2uWwmFA6YpFLo/PKPud9jY=
Endpoint            : 172.16.1.2:51820
Latest handshake    : 72 seconds
Persistent keepalive: off seconds
Rx bytes            : 1372
Tx bytes            : 1460
```

```
 Allowed IPs
CIDR
10.0.0.2/32
192.168.2.0/24
```

Key indicators of a healthy tunnel:

- **Latest Handshake:** Should be recent (within the last 2 minutes for active tunnels)
- **Rx/Tx bytes:** Shows data being transmitted through the tunnel
- **Endpoint:** Should show the remote peer's address

Testing Connectivity

Test connectivity between the sites:

```
alice:/#> ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=3.92 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.817 ms
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss
```

Test access to Site B's network:

```
alice:/#> ping 192.168.2.5
PING 192.168.2.5 (192.168.2.5) 56(84) bytes of data.
64 bytes from 192.168.2.5: icmp_seq=1 ttl=63 time=2.15 ms
64 bytes from 192.168.2.5: icmp_seq=2 ttl=63 time=0.783 ms
--- 192.168.2.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss
```

Troubleshooting

No Handshake

If the latest handshake shows "(none)" or is very old:

1. Verify that both peers have correct public keys configured
2. Check that the endpoint addresses are correct and reachable
3. Ensure UDP port 51820 is not blocked by firewalls
4. Verify that private keys are correctly configured

Traffic Not Flowing

If handshake succeeds but traffic doesn't flow:

1. Verify allowed-ip configurations on both sides:
   - Alice should allow 10.0.0.2/32 and 192.168.2.0/24
   - Bob should allow 10.0.0.1/32 and 192.168.1.0/24
2. Check routing tables on both gateways
3. Verify that IP forwarding is enabled
4. Check firewall rules

Connection Drops

If the connection drops frequently:

1. Ensure persistent-keepalive is configured (25 seconds recommended)
2. Check for network stability issues

3. Verify that no aggressive firewall timeouts are interfering

Advanced Configuration

Using Preshared Keys for Additional Security

Add a preshared key for post-quantum resistance:

```
alice:/config/tunnel/wireguard-wg0/peer-1/#> preshared-key kHvZAEfyOj4B5Qs/z/hUwHMurmJeE1UQtXedKDEeDno=
```

Both peers must use the same preshared key.

Multiple Site Networks

To allow access to multiple networks at Site B, add more allowed-ip entries:

```
alice:/config/tunnel/wireguard-wg0/peer-1/#> allowed-ip 10.2.0.0/24
alice:/config/tunnel/wireguard-wg0/peer-1/#> allowed-ip 10.3.0.0/24
alice:/config/tunnel/wireguard-wg0/peer-1/#> allowed-ip 10.4.0.0/24
```

Security Considerations

1. **Key Protection:** Never expose private keys. They should only be configured on their respective gateways.
2. **Firewall Rules:** Limit access to UDP port 51820 to only the peer's IP address if possible.
3. **Regular Monitoring:** Check handshake times regularly to ensure connectivity.
4. **Key Rotation:** Consider rotating keys periodically for enhanced security.
5. **Allowed IPs:** Use specific network ranges rather than 0.0.0.0/0 to limit access to only necessary networks.

Summary

This HowTo demonstrated:

- Generating WireGuard keys for both sites
- Configuring WireGuard interfaces with tunnel IPs
- Setting up peers with public keys and endpoints
- Configuring allowed IPs for site-to-site routing
- Enabling persistent keepalive for NAT traversal
- Verifying tunnel status and testing connectivity

For more details on WireGuard configuration options, see the WireGuard Guide.

4.11. Firewall and NAT

4.11.1. Firewall HowTo

Introduction

This document covers common Firewall use cases and step-by-step configuration examples using WeOS.

For more information, see the Firewall Configuration Guide.

Private Network With Web Server

Figure 1 shows a common network setup. Two LAN subnets and a WEB subnet are connected through a gateway and Firewall to the Internet. The following example shows how the Firewall can be configured to allow devices on both LANs to communicate with each other, access Gateway services, and connect to the web server. Devices on the Internet should also be able to access the web server, while all other traffic is dropped.

Desired outcome:

- All private subnets can communicate with each other.
- All private subnets can access Gateway services such as DHCP.
- The web server is accessible by the LANs and Internet.
- All other external traffic is dropped.

```
                                  .--.-.
                                 ( (    )__
                                (_,  \ ) ,_)  Internet/Intranet
                                  '-'--`--'
                                      |  WAN (vlan4)                        ^
                                 .---+----.                                 |
                                 |        |                              Public
                          +-----+ GW/FW  +-----+                          ------
                          |  .99|        |.99  |                          Private
LAN (vlan1)               |     '----+---'     |      LAN (vlan2)           |
192.168.1.0/24            |          |.99      |      192.168.2.0/24        V
                 ----+----+          |         +----+----
                     |.1             |              |.1
                  .-+--.             |            .-+--.
                  | H2 |             |            | H3 |
                  '----'             |            '----'
                                     |
                                     |
                                ----+---- WEB (vlan3)
                                     |.1   192.168.3.0/24
                                   .-+--.
                                   | H1 |
                                   '----'
```

Figure 1: Showing the example network setup using a Firewall.

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> firewall
example:/config/ip/firewall/#> policy input drop
example:/config/ip/firewall/#> policy forward drop
example:/config/ip/firewall/#> forward accept in vlan1
example:/config/ip/firewall/#> forward accept in vlan2
example:/config/ip/firewall/#> forward accept in vlan3
example:/config/ip/firewall/#> input accept in vlan1
example:/config/ip/firewall/#> input accept in vlan2
example:/config/ip/firewall/#> input accept in vlan3
example:/config/ip/firewall/#> port-forward in vlan4 dport 80 to-daddr 192.168.3.1
to-dport 8080 proto tcp
example:/config/ip/firewall/#> forward accept in vlan4 daddr 192.168.3.1 dport 8080 proto tcp
```

```
example:/config/ip/firewall/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
```

Example Input Rules

When enabling the Firewall, all incoming packets will be blocked unless the policy for *input* filters is changed from **drop** to **accept** or individual *accept* rules are created for each service.

> **NOTE**
>
> If you do not allow any management access (*SSH*, *HTTP*, *HTTPS*) on any interface, you will not be able to reach the device after enabling the Firewall. Be careful and do not lock yourself out!

Allow SSH access only on vlan1

```
example:/config/ip/firewall/#> input accept in vlan1 dport ssh proto tcp
```

Allow HTTP access on any interface

```
example:/config/ip/firewall/#> input accept dport http proto tcp
```

Allow HTTPS access only on vlan1

```
example:/config/ip/firewall/#> input accept in vlan1 dport https proto tcp
```

Allow DNS access (tcp, port 53) on any interface

```
example:/config/ip/firewall/#> input accept dport 53 proto tcp
```

Allow DNS access (udp, port 53) on any interface

```
example:/config/ip/firewall/#> input accept dport 53 proto udp
```

Allow DHCP access on any interface (udp, port 67)

```
example:/config/ip/firewall/#> input accept dport 67 proto udp
```

Allow NTP on any interface (udp, port 123)

```
example:/config/ip/firewall/#> input accept dport 123 proto udp
```

Allow ping (ICMP) only on vlan2

```
example:/config/ip/firewall/#> input accept in vlan2 proto icmp
```

Allow traceroute on any interface

```
example:/config/ip/firewall/#> input accept proto udp dport 33434-33534
```

Allow TFTP only on vlan1 (udp, port 69)

```
example:/config/ip/firewall/#> input accept in vlan1 dport 69 proto udp
```

Allow SNMP access on vlan1 (udp, port range 161-162)

```
example:/config/ip/firewall/#> input accept in vlan1 dport 161-162 proto udp
```

Allow VRRP on interface vrrp1

```
example:/config/ip/firewall/#> input accept in vrrp1 proto vrrp
```

Allow OSPF on interface vlan1

```
example:/config/ip/firewall/#> input accept in vlan1 proto ospf
```

In a similar way, other services can be reached by accepting the following protocol/port numbers:

- Modbus - tcp, port 502
- MlkII - udp, port 60000
- RIP - udp, port 520
- Telnet - tcp, port 23
- FTP - tcp, port 20-21 (not supported through NAT)

More port numbers can be found on Wikipedia[1].

Example Forward Rules

Forward rules control what traffic is allowed to pass through the device between interfaces. When the default forward policy is *drop*, explicit *accept* rules must be created for each desired traffic flow.

Allow all traffic from vlan1 to vlan2

```
example:/config/ip/firewall/#> forward accept in vlan1 out vlan2
```

Allow traffic from vlan1 to a specific host on vlan2

```
example:/config/ip/firewall/#> forward accept in vlan1 out vlan2 daddr 192.168.2.10/32
```

Allow HTTP and HTTPS traffic from vlan1 to vlan3

```
example:/config/ip/firewall/#> forward accept in vlan1 out vlan3 dport 80 proto tcp
example:/config/ip/firewall/#> forward accept in vlan1 out vlan3 dport 443 proto tcp
```

Allow traffic from a private subnet to the WAN interface

```
example:/config/ip/firewall/#> forward accept in vlan1 out vlan4 saddr 192.168.1.0/24
```

Drop traffic from a specific host regardless of destination

```
example:/config/ip/firewall/#> forward drop saddr 192.168.1.50/32
```

[1] https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Example Prerouting Rules

Prerouting rules are evaluated before the routing decision. In WeOS, prerouting supports DSCP mangling, so packets can be marked with a DSCP value to influence QoS treatment along the forwarding path.

Mark all traffic arriving on vlan1 with DSCP 46 (Expedited Forwarding)

```
example:/config/ip/firewall/#> prerouting dscp 46 in vlan1
```

Mark HTTP traffic from vlan2 with DSCP 10 (AF11)

```
example:/config/ip/firewall/#> prerouting dscp 10 in vlan2 dport 80 proto tcp
```

Mark traffic destined for a specific host with DSCP 34 (AF41)

```
example:/config/ip/firewall/#> prerouting dscp 34 daddr 10.0.0.1/32
```

Example NAT Rules

NAT (NAPT/masquerading) rules allow hosts on a private subnet to share a single public IP address on an outbound interface. NAT rules apply during postrouting.

Enable masquerading on the WAN interface (vlan4)

```
example:/config/ip/firewall/#> nat out vlan4
```

Enable masquerading on the WAN interface for a specific source subnet only

```
example:/config/ip/firewall/#> nat out vlan4 saddr 192.168.1.0/24
```

Enable masquerading for multiple private subnets on the WAN interface

```
example:/config/ip/firewall/#> nat out vlan4 saddr 192.168.1.0/24
example:/config/ip/firewall/#> nat out vlan4 saddr 192.168.2.0/24
```

Example Port Forward Rules

Port forwarding allows a service on a private host to be reached from an external network by mapping an external port on the gateway to an internal address and port. Port forwarding rules apply during prerouting and typically require a corresponding forward rule to allow the redirected traffic through.

Forward incoming HTTP (port 80) on vlan4 to an internal web server

```
example:/config/ip/firewall/#> port-forward in vlan4 dport 80 to-daddr 192.168.3.1 proto tcp
example:/config/ip/firewall/#> forward accept in vlan4 daddr 192.168.3.1 dport 80 proto tcp
```

Forward a non-standard external port to an internal service port

```
example:/config/ip/firewall/#> port-forward in vlan4 dport 2222 to-daddr 192.168.1.10 to-dport 22 proto tcp
example:/config/ip/firewall/#> forward accept in vlan4 daddr 192.168.1.10 dport 22 proto tcp
```

© 2026 Westermo Network Technologies AB

Forward a port range to an internal device

```
example:/config/ip/firewall/#> port-forward in vlan4 dport 5000-5010 to-daddr 192.168.2.20 proto udp
example:/config/ip/firewall/#> forward accept in vlan4 daddr 192.168.2.20 dport 5000-5010 proto udp
```

Example of ALG Helpers

This is an example of how a D-2 device can access files on a TFTP/FTP server running TFTP over the default port 69 and FTP over a custom port 2222. The device D-1 is acting as a firewall.

```
┌──────────┐        ┌──────────┐        ┌──────────┐
│   D-1    │        │   D-2    │        │   D-3    │
│ TFTP/FTP │        │   FW     │        │ CLIENT   │
│   SRV    │        │          │        │          │
│     eth1├────────┤eth1  eth2├────────┤eth1      │
└──────────┘        └──────────┘        └──────────┘
            vlan21              vlan20
```

Part of the firewall configuration on D-1:

```
FW:/config/#> ip firewall
FW:/config/ip/firewall/#> policy forward drop
FW:/config/ip/firewall/#> forward accept in vlan20 out vlan21 dport 2222 proto tcp
FW:/config/ip/firewall/#> forward accept in vlan20 out vlan21 dport 69 proto udp
FW:/config/ip/firewall/#> alg ftp 2222
FW:/config/ip/firewall/#> alg tftp
FW:/config/ip/firewall/#> show


Forward Rules
ID  Rule
001 forward accept in vlan20 out vlan21 dport 2222 proto tcp
002 forward accept in vlan20 out vlan21 dport 69 proto udp
N/A drop


Application Level Gateway
Alg Helper
FTP  ENABLED  PORT 2222
TFTP ENABLED  PORT 69
```

4.11.2. Stateless NAT HowTo

Introduction

This document details some common use-cases for stateless NAT and how to configure them using WeOS.

For more information, see the Stateless NAT Configuration Guide.

Single Private Network

The below figure shows a sample network where the NAT GW translates the private network (192.168.1.0/24) to the 10.0.2.0/24 network on the public side and vice versa.

```
                 .--.-.
                ( (    )__
               (_,  \ ) ,_)  Internet/Intranet
                 '-'--`--'
                   |
                 .-+-.
                 | R |
                 '-+-'
                   |.1
                   |
  ^                |.33       10.0.0.0/24
  |            .---+--.
Public         |      | vlan2
------         |  GW  |
Private        |      | vlan1
  |            '---+--'
  V                |.1        (10.0.2.0/24)
                   |          192.168.1.0/24
     ---+------+---+----------+---------
        |          |          |
        |.11       |.5        |.139
      .-+--.     .-+--.     .-+--.
      | H1 |     | H2 |     | H3 |
      '----'     '----'     '----'
```

Figure 1: Basic stateless NAT setup

Assumptions:

- The NAT gateway is already set up with two VLAN interfaces, with relevant addresses and IP forwarding enabled
- The upper router (R) has a route to the 10.0.2.0/24 network via 10.0.0.33, the IP address of the NAT GW.

The additional configuration for setting up stateless NAT is as follows:

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> stateless-nat
example:/config/ip/stateless-nat/#> dnat in vlan2 dst 10.0.2.0/24 to-dst 192.168.1.0
Would you like routes to be added for NAT rules (y/N)? n

You will have to manually add routes for your NAT rules!
example:/config/ip/stateless-nat/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> copy run start
example:/#>
```

Multiple Private Networks

The network below illustrates a case where a private network (192.168.1.0/24) is used in one setup and then replicated. When connecting these networks together, stateless NAT can be used to achieve connectivity across all private networks.

```
                    .--.-.
                   ( (    )__
                  (_,  \ ) ,_)                          Internet
                    '-'--`--'
                       |
                    .-+-.
                    | R |                               (optional)
                    '---'
                       |.100
                       |                                10.0.0.0/24
            ---+-----+------------+-----------------+---------------
  ^            |.1                |.2                |.3
  |         .--+--. vlan2       .--+--. vlan2      .--+--. vlan2
Public      |     |             |     |            |     |
-------     | GW1 |             | GW2 |            | GW3 |
Private     |     |             |     |            |     |
  |         '--+--' vlan1       '--+--' vlan1      '--+--' vlan1
  V            |.1                |.1                |.1
        -------+---------  -------+---------  ------+---------
        192.168.1.0/24     192.168.1.0/24     192.168.1.0/24
        => 10.0.1.0/24     => 10.0.2.0/24     => 10.0.3.0/24
        Private Net #1     Private Net #2     Private Net #3
```

Figure 2: Multiple similar private networks

All private networks use subnet 192.168.1.0/24 internally. From the *outside*, above their respective NAT GW, they are:

- Private Net #1 is known as 10.0.1.0/24
- Private Net #2 is known as 10.0.2.0/24
- Private Net #3 is known as 10.0.3.0/24

For full reachability NAT GW1 must have routes for to *10.0.2.0/24* via *10.0.0.2* and *10.0.3.0/24* via 10.0.0.3, etc. In this example there is also an additional router towards the Internet (R), meaning NAT GW1 has its *default route* pointing to *10.0.0.100*. Similar routes must be set up for NAT GW2, NAT GW3, and the (optional) Upper Router (R).

```
nat-gw1:/config/ip/#> route 10.0.2.0/24 10.0.0.2
nat-gw1:/config/ip/#> route 10.0.3.0/24 10.0.0.3
nat-gw1:/config/ip/#> route default 10.0.0.100
```

The stateless NAT configuration at NAT GW1 is as follows:

```
nat-gw1:/config/ip/#> stateless-nat
nat-gw1:/config/ip/stateless-nat/#> dnat in vlan2 dst 10.0.1.0/24 to-dst 192.168.1.0
Would you like routes to be added for NAT rules (y/N)? n

You will have to manually add routes for your NAT rules!
nat-gw1:/config/ip/stateless-nat/#>
```

After configuring all NAT GWs, and optionally Upper Router (R), a host on private network #1 can *ping* a remote host (192.168.1.33) on private network #3, using its *public* address (10.0.3.33).

```
host-on-net1:~$ ping -c 3 10.0.3.33
PING 10.0.3.33 (10.0.3.33) 56(84) bytes of data.
```

```
64 bytes from 10.0.3.33: icmp_seq=1 ttl=62 time=1.25 ms
64 bytes from 10.0.3.33: icmp_seq=2 ttl=62 time=1.11 ms
64 bytes from 10.0.3.33: icmp_seq=3 ttl=62 time=1.12 ms

--- 10.0.3.33 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.115/1.164/1.250/0.072 ms
host-on-net1:~$
```

Multiple Private Networks using Proxy ARP

The network below illustrates *almost* the exact same case, except that all NAT gateways and the Upper Router (R) reside on a /16 subnet rather than a /24 subnet. Note that the 10.0.0.0/16 subnet covers all of the /24 networks (10.0.1.0/24, etc). used for translation of the private networks.

```
                    .--.-.
                   ( (    )__
                  (_,  \ ) ,_)                          Internet
                    '-'--`--'
                      |
                    .-+-.
                    | R |                               (optional)
                    '---'
                      |.100
                      |                                 10.0.0.0/16
            ---+-----+------------+-----------------+---------------
  ^            |.1                |.2               |.3
  |         .--+--. vlan2      .--+--. vlan2     .--+--. vlan2
Public      |     |            |     |           |     |
-------     | GW1 |            | GW2 |           | GW3 |
Private     |     |            |     |           |     |
  |         '--+--' vlan1      '--+--' vlan1     '--+--' vlan1
  V            |.1                |.1               |.1
        -------+---------  -------+---------  ------+---------
        192.168.1.0/24     192.168.1.0/24     192.168.1.0/24
        => 10.0.1.0/24     => 10.0.2.0/24     => 10.0.3.0/24
        Private Net #1     Private Net #2     Private Net #3
```

Figure 3: Multiple similar private networks with simplified routing

As before, all private networks use subnet 192.168.1.0/24 internally. From the outside it looks as if each private network represents a *slice* of the larger 10.0.0.0/16 subnet:

- Private Net #1 is known as 10.0.1.0/24
- Private Net #2 is known as 10.0.2.0/24
- Private Net #3 is known as 10.0.3.0/24

The *purpose* here is to avoid configuring individual routes to every Private Network at the NAT Gateways. With many private networks this can otherwise be a real hassle. Instead, the NAT gateways need to use *Proxy ARP* on their *upper* interface to ensure the traffic to their private network will reach them. In order for NAT GW1 to proxy-ARP for traffic toward 10.0.1.0/24, it needs a route to that subnet. Such a route can be added automatically, to the *loopback* interface, by answering *Yes* when adding the DNAT rule (see below).

The following example shows the route, stateless NAT and proxy ARP configuration at NAT GW1.

```
nat-gw1:/config/ip/#> route default 10.0.0.100
nat-gw1:/config/ip/#> stateless-nat
nat-gw1:/config/ip/stateless-nat/#> dnat in vlan2 dst 10.0.1.0/24 to-dst 192.168.1.0
Would you like routes to be added for NAT rules (y/N)? y
Added routes for NAT rules!
nat-gw1:/config/ip/stateless-nat/#> end
```

```
nat-gw1:/config/ip/#> show route
Network          Netmask          Gateway          Interface        Distance
═══════════════════════════════════════════════════════════════════════════════
10.0.1.0         255.255.255.0    *                lo               254
0.0.0.0          0.0.0.0          10.0.0.100       *                0
nat-gw1:/config/ip/#> end
nat-gw1:/config/#> iface vlan2
nat-gw1:/config/iface-vlan2/#> proxy-arp
nat-gw1:/config/iface-vlan2/#>
```

Having configured all NAT GWs, and optionally Upper Router (R), a host on private network #1 can *ping* a remote host (192.168.1.33) on network #3, using its *public* address (10.0.3.33).

```
host-on-net1:~$ ping -c 3 10.0.3.33
PING 10.0.3.33 (10.0.3.33) 56(84) bytes of data.
64 bytes from 10.0.3.33: icmp_seq=1 ttl=62 time=1.29 ms
64 bytes from 10.0.3.33: icmp_seq=2 ttl=62 time=1.09 ms
64 bytes from 10.0.3.33: icmp_seq=3 ttl=62 time=1.09 ms

--- 10.0.3.33 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.097/1.164/1.296/0.093 ms
host-on-net1:~$
```

4.12. Train

4.12.1. One Consist Network with One ETB Node HowTo

This document describes how to configure a consist with one consist network managed by one ETBN. The consist is depicted in Figure 1.

The result of the inauguration will be a network where ETBN1 has the ETBN ID 1, the IP-address 10.128.0.1 (based on the ETBN ID and since the network is configured as TCMS). CN1 will attain the network IP 10.128.64.0/18.

```
     .------------.
    /     cst1     \
    '--------------'
       .-------.
     --+       +--     dir1 = (ethX5, ethX6)
<-dir1| ETBN1 |dir2-> dir2 = (ethX7, ethX8)
     --+       +--
       '---+---'
           |
    -------+----- CN1 vlan10 (ethX3)
```

Figure 1: A consist with one consist network managed by one ETB node.

Configuration

```
etbn1:#> configure
etbn1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1:/config/vlan-10/#> untagged ethX3
etbn1:/config/vlan-10/#> end
etbn1:/config/#> iface vlan10
etbn1:/config/iface-vlan10/#> no inet
etbn1:/config/iface-vlan10/#> inet static 10.0.0.1/18
etbn1:/config/iface-vlan10/inet-static-10.0.0.1/#> end
etbn1:/config/iface-vlan10/#> end
etbn1:/config/#> ttdp
Activating TTDP with default settings.
etbn1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1:/config/ttdp/#> node 1 ecn 1
etbn1:/config/ttdp/#> ecn 1 vlan10
etbn1:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn1:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn1:/config/ttdp/#> local-id 1
etbn1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1:/config/vlan-2/#> no multicast-snooping
etbn1:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> speed-duplex 100-full
```

```
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> speed-duplex 100-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> speed-duplex 100-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> speed-duplex 100-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink
etbn1:/config/port-ethX8/#> mdix-mode mdi
etbn1:/config/port-ethX8/#> end
etbn1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1:/#>
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

Inauguration Result

Using the `show ttdp` command we now verify that we have recevied the intended network configuration. Highlighted in yellow are the values that are of interest.

```
etbn1:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 1
etbn-ip      : 10.128.0.1
local-id     : 1
own-mac      : 0c:65:8c:93:00:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0x32752ae8
EtbTopoCnt   : 0x80ea82ac
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.1/18     1
```

```
NODE ID      MAC ADDRESS            ORIENTATION
       1      0c:65:8c:93:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111  1       Direct     10.128.64.0/18

etbn1:/#>
```

4.12.2. One Consist Network with Two Redundant ETB Nodes HowTo

This document describes how to configure a consist with a single consist network managed by two redundant ETBNs. The consist is depicted in Figure 1.

The result of the inauguration will be a network where ETBN1 has the ETBN ID 1, the IP-address 10.128.0.1 (based on the ETBN ID and since the network is configured as TCMS). ETBN2 will attain ETBN-ID 2 and IP 10.128.1.2. CN1 will attain the network IP 10.128.64.0/18.

```
                .---------------------------.
               /            cst1             \
               '-----------------------------'
                 .-------.    .-------.
                --+       +----+       +--     dir1 = (ethX5, ethX6)
            <-dir1| ETBN1 |    | ETBN2 |dir2-> dir2 = (ethX7, ethX8)
                --+       +----+       +--
                  '---+---'    '---+---'
                      |            |
                  ----+------------+---- CN1 vlan10 (ethX3)
```

Figure 1: A single consist with one consist network managed by two redundant ETB nodes.

> **NOTE**
>
> The two redundant nodes must have the same configuration with respect to `uuid` and `node`. In this example we also have the same `ecn` configuration. They must, however, have different values for the `local-id` setting.

Both nodes also need to have VRRP configured and enabled on the CN side.

Configuration

Start with ETBN1 and configure VLANs and TTDP.

```
etbn1:#> configure
etbn1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1:/config/vlan-10/#> untagged ethX3
etbn1:/config/vlan-10/#> end
etbn1:/config/#> iface vlan10
etbn1:/config/iface-vlan10/#> no inet
etbn1:/config/iface-vlan10/#> inet static 10.0.0.1/18
etbn1:/config/iface-vlan10/inet-static-10.0.0.1/#> end
etbn1:/config/iface-vlan10/#> end
etbn1:/config/#> ttdp
Activating TTDP with default settings.
etbn1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1:/config/ttdp/#> node 1 ecn 1
etbn1:/config/ttdp/#> node 2 ecn 1
etbn1:/config/ttdp/#> ecn 1 vlan10
etbn1:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn1:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn1:/config/ttdp/#> local-id 1
etbn1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1:/config/#> vlan 2
```

```
Creating new VLAN vid:2 with name: vlan2
etbn1:/config/vlan-2/#> no multicast-snooping
etbn1:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> speed-duplex 100-full
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> speed-duplex 100-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> speed-duplex 100-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> speed-duplex 100-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink
etbn1:/config/port-ethX8/#> mdix-mode mdi
etbn1:/config/port-ethX8/#> end
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

The next step is to configure VRRP on each node. Start with ETBN1 which will have the MASTER role. The MASTER role is gained by setting the priority for the VRRP instance higher than on ETBN2 (see VRRP).

```
etbn1:/config/#> router
etbn1:/config/router/#> vrrp cn1
Creating new VRRP instance: cn1
Invalid settings: Interface not set.
etbn1:/config/router/vrrp-cn1/#> iface vlan1
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn1:/config/router/vrrp-cn1/#> address 10.0.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn1:/config/router/vrrp-cn1/#> vrid 132
etbn1:/config/router/vrrp-cn1/#> version 3
etbn1:/config/router/vrrp-cn1/#> priority 150
```

```
etbn1:/config/router/vrrp-cn1/#> interval 1
etbn1:/config/router/vrrp-cn1/#> preempt delay 1
etbn1:/config/router/vrrp-cn1/#> end
etbn1:/config/router/#> end
etbn1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1:/#>
```

Now, repeat the exact steps above in the second ETB node. The only difference is the static address for VLAN 10 and the local id.

```
etbn2:#> configure
etbn2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn2:/config/vlan-10/#> untagged ethX3
etbn2:/config/vlan-10/#> end
etbn2:/config/#> iface vlan10
etbn2:/config/iface-vlan10/#> no inet
etbn2:/config/iface-vlan10/#> b>inet static 10.0.0.2/18
etbn2:/config/iface-vlan10/inet-static-10.0.0.2/#> end
etbn2:/config/iface-vlan10/#> end
etbn2:/config/#> ttdp
Activating TTDP with default settings.
etbn2:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn2:/config/ttdp/#> node 1 ecn 1
etbn2:/config/ttdp/#> node 2 ecn 1
etbn2:/config/ttdp/#> ecn 1 vlan10
etbn2:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn2:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn2:/config/ttdp/#> local-id 2
etbn2:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn2:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn2:/config/vlan-2/#> no multicast-snooping
etbn2:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn2:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn2:/config/#> port ethX5
etbn2:/config/port-ethX5/#> speed-duplex 100-full
etbn2:/config/port-ethX5/#> no auto-negotiate
etbn2:/config/port-ethX5/#> fastlink
etbn2:/config/port-ethX5/#> mdix-mode mdi
etbn2:/config/port-ethX5/#> end
etbn2:/config/#> port ethX6
etbn2:/config/port-ethX6/#> speed-duplex 100-full
etbn2:/config/port-ethX6/#> no auto-negotiate
etbn2:/config/port-ethX6/#> fastlink
```

```
etbn2:/config/port-ethX6/#> mdix-mode mdi
etbn2:/config/port-ethX6/#> end
etbn2:/config/#> port ethX7
etbn2:/config/port-ethX7/#> speed-duplex 100-full
etbn2:/config/port-ethX7/#> no auto-negotiate
etbn2:/config/port-ethX7/#> fastlink
etbn2:/config/port-ethX7/#> mdix-mode mdi
etbn2:/config/port-ethX7/#> end
etbn2:/config/#> port ethX8
etbn2:/config/port-ethX8/#> speed-duplex 100-full
etbn2:/config/port-ethX8/#> no auto-negotiate
etbn2:/config/port-ethX8/#> fastlink
etbn2:/config/port-ethX8/#> mdix-mode mdi
etbn2:/config/port-ethX8/#> end
```

Repeat the same steps for ETBN2 with a lower priority to make it backup. We set the priority to 100.

```
etbn2:/config/#> router
etbn2:/config/router/#> vrrp cn1
Creating new VRRP instance: cn1
Invalid settings: Interface not set.
etbn2:/config/router/vrrp-cn1/#> iface vlan1
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn2:/config/router/vrrp-cn1/#> address 10.0.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn2:/config/router/vrrp-cn1/#> vrid 132
etbn2:/config/router/vrrp-cn1/#> version 3
etbn2:/config/router/vrrp-cn1/#> priority 100
etbn2:/config/router/vrrp-cn1/#> interval 1
etbn2:/config/router/vrrp-cn1/#> preempt delay 1
etbn2:/config/router/vrrp-cn1/#> end
etbn2:/config/router/#> end
etbn2:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn2:/#>
```

Communication Profile Configuration

Finally, for proper ECSP operation, a *CSTINFO data file* is needed on the device. See the Communication Profile documentation for more information about this file and its purpose.

For this HowTo, we will be using the following file: cst1 communication profile configuration.

On each node, the *CSTINFO data file* must be placed in the directory **/cfg/** with the name **cstinfo.json**. This can be done by using any available administrative interface, or by putting the file on the external USB media (in which case it will be available in the **/cfg/** directory after rebooting the device).

> **NOTE**
>
> Ensure the filenames are correct. That is, the file should be named **cstinfo.json** and placed in **/cfg/** on each device.

> **NOTE**
>
> Both devices use identical CSTINFO data files. If nodes in the same consist have different CSTINFO data, or if only one of them has a data file while the other does not, results will be unpredictable.

Inauguration Result

Using the `show ttdp` command we now verify that we have recevied the intended network configuration. Highlighted in yellow are the values that are of interest.

```
etbn1:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 1
etbn-ip      : 10.128.0.1
local-id     : 1
own-mac      : 0c:65:8c:93:00:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0×b81af94
EtbTopoCnt   : 0×6f002b4
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.1/18     1,2

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111  1..2    Direct     10.128.64.0/18

etbn1:/#>
```

```
etbn2:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 2
etbn-ip      : 10.128.0.2
local-id     : 2
own-mac      : 0c:7a:40:ee:00:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0×b81af94
EtbTopoCnt   : 0×6f002b4
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.2/18     1,2

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
```

```
      2      0c:7a:40:ee:00:00      Direct

CONSIST                                  NODES    DIRECTION   NETWORKS
11111111-1111-1111-1111-111111111111  1..2     Direct      10.128.64.0/18

etbn2:/#>
```

4.12.3. Two Consist Networks with One ETB Node HowTo

This document describes how to configure a consist with one consist network managed by one ETBN in a. The consist is depicted in Figure 1.

The result of the inauguration will be a network where ETBN1 has the ETBN ID 1, the IP-address 10.128.0.1 (based on the ETBN ID and since the network is configured as TCMS). CN1 will attain the network IP 10.128.64.0/18 and CN2 10.128.128.0/18.

```
     .------------.
    /     cst1     \
    '--------------'
       .-------.
     --+       +--     dir1 = (ethX5, ethX6)
<-dir1| ETBN1 |dir2-> dir2 = (ethX7, ethX8)
     --+       +--
       '-+---+-'
         |   |
    -----+----- CN1 vlan10 (ethX3)
             |
         -----+----- CN2 vlan20 (ethX4)
```

Figure 1: A consist with two consist networks managed by one ETB node.

Configuration

```
etbn1:#> configure
etbn1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1:/config/vlan-10/#> untagged ethX3
etbn1:/config/vlan-10/#> end
etbn1:/config/#> vlan 20
Creating new VLAN vid:20 with name: vlan20
etbn1:/config/vlan-20/#> untagged ethX4
etbn1:/config/vlan-20/#> end
etbn1:/config/#> iface vlan10
etbn1:/config/iface-vlan10/#> no inet
etbn1:/config/iface-vlan10/#> inet static 10.10.0.1/18
etbn1:/config/iface-vlan10/inet-static-10.10.0.1/#> end
etbn1:/config/iface-vlan10/#> end
etbn1:/config/#> iface vlan20
etbn1:/config/iface-vlan20/#> no inet
etbn1:/config/iface-vlan20/#> inet static 10.20.0.1/18
etbn1:/config/iface-vlan20/inet-static-10.20.0.1/#> end
etbn1:/config/iface-vlan20/#> end
etbn1:/config/#> ttdp
Activating TTDP with default settings.
etbn1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1:/config/ttdp/#> node 1 ecn 1,2
etbn1:/config/ttdp/#> ecn 1 vlan10
etbn1:/config/ttdp/#> ecn 2 vlan20
etbn1:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1:/config/ttdp/#> local-id 1
etbn1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
```

```
etbn1:/config/vlan-2/#> no multicast-snooping
etbn1:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> speed-duplex 100-full
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> speed-duplex 100-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> speed-duplex 100-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> speed-duplex 100-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink
etbn1:/config/port-ethX8/#> mdix-mode mdi
etbn1:/config/port-ethX8/#> end
etbn1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1:/#>
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

Inauguration Result

Using the `show ttdp` command we now verify that we have recevied the intended network configuration. Highlighted in yellow are the values that are of interest.

```
etbn1:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 1
etbn-ip      : 10.128.0.1
local-id     : 1
own-mac      : 0c:65:8c:93:00:00
uuid         : 11111111-1111-1111-1111-111111111111
```

```
connTableCnt : 0x32752ae8
EtbTopoCnt   : 0x3d2e40c0
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.10.0.1/18    1
   2     vlan20    10.20.0.1/18    1

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1       Direct     10.128.64.0/18
                                                          10.128.128.0/18
etbn1:/#>
```

Here it is possible to verofy the inaugurated ETBN ID and IP, the CN numbering and CN network IP. To highlight the values, the values are marked as bold in the example.

4.12.4. Two Consist Networks with Two Redundant ETB Nodes HowTo

This document describes how to configure a consist with two consist networks managed by two redundant ETBNs. The consist is depicted in Figure 1.

The result of the inauguration will be a network where ETBN1 has the ETBN ID 1, the IP-address 10.128.0.1 (based on the ETBN ID and since the network is configured as TCMS). ETBN2 will attain ETBN-ID 2 and IP 10.128.1.2. CN1 will attain the network IP 10.128.64.0/18, and CN2 IP 10.128.128.0/18.

```
                .---------------------------.
               /            cst1             \
               '-----------------------------'
                  .-------.    .-------.
                --+       +----+       +--     dir1 = (ethX5, ethX6)
            <-dir1| ETBN1 |    | ETBN2 |dir2-> dir2 = (ethX7, ethX8)
                --+       +----+       +--
                  '-+--+--'    '--+--+-'
                    |  |          |  |
                  --+----------------+-- CN1 vlan10 (ethX3)
                      |          |
                    ---+----------+-------- CN2 vlan20 (ethX4)
```

Figure 1: A consist with two consist networks managed by two redundant ETB nodes.

> **NOTE**
>
> In order to get two ETB nodes to manage two consist networks, they both need to have the exact same configuration with respect to `uuid`, `node` and, in this setup, `ecn` configuration. Also, the ETBN must have different `local-id`.

Configuration

Start with ETBN 1 in the consist and configure VLANs and the TTDP part. Both nodes also need to have VRRP configured and enabled.

```
etbn1:#> configure
etbn1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1:/config/vlan-10/#> untagged ethX3
etbn1:/config/vlan-10/#> end
etbn1:/config/#> vlan 20
Creating new VLAN vid:20 with name: vlan20
etbn1:/config/vlan-20/#> untagged ethX4
etbn1:/config/vlan-20/#> end
etbn1:/config/#> iface vlan10
etbn1:/config/iface-vlan10/#> no inet
etbn1:/config/iface-vlan10/#> inet static 10.10.0.1/18
etbn1:/config/iface-vlan10/inet-static-10.10.0.1/#> end
etbn1:/config/iface-vlan10/#> end
etbn1:/config/#> iface vlan20
etbn1:/config/iface-vlan20/#> no inet
etbn1:/config/iface-vlan20/#> inet static 10.20.0.1/18
etbn1:/config/iface-vlan20/inet-static-10.20.0.1/#> end
etbn1:/config/iface-vlan20/#> end
etbn1:/config/#> ttdp
Activating TTDP with default settings.
etbn1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1:/config/ttdp/#> node 1 ecn 1,2
etbn1:/config/ttdp/#> node 2 ecn 1,2
```

```
etbn1:/config/ttdp/#> ecn 1 vlan10
etbn1:/config/ttdp/#> ecn 2 vlan20
etbn1:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1:/config/ttdp/#> local-id 1
etbn1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1:/config/vlan-2/#> no multicast-snooping
etbn1:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> speed-duplex 100-full
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> speed-duplex 100-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> speed-duplex 100-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> speed-duplex 100-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink
etbn1:/config/port-ethX8/#> mdix-mode mdi
etbn1:/config/port-ethX8/#> end
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

Next step is to configure one VRRP instance for each CN on each node. Start with the node that will have the MASTER role for both CN's. In this example, ETBN1 is selected as MASTER (see VRRP).

First, configure the VRRP instance for CN1:

```
etbn1:/config/#> router
etbn1:/config/router/#> vrrp cn1
Creating new VRRP instance: cn1
Invalid settings: Interface not set.
etbn1:/config/router/vrrp-cn1/#> iface vlan10
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn1:/config/router/vrrp-cn1/#> address 10.10.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn1:/config/router/vrrp-cn1/#> vrid 132
etbn1:/config/router/vrrp-cn1/#> version 3
etbn1:/config/router/vrrp-cn1/#> priority 150
etbn1:/config/router/vrrp-cn1/#> interval 1
etbn1:/config/router/vrrp-cn1/#> preempt delay 1
etbn1:/config/router/vrrp-cn1/#> end
```

Configure the VRRP instance for CN2:

```
etbn1:/config/router/#> vrrp cn2
Creating new VRRP instance: cn2
Invalid settings: Interface not set.
etbn1:/config/router/vrrp-cn2/#> iface vlan20
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn1:/config/router/vrrp-cn2/#> address 10.20.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn1:/config/router/vrrp-cn2/#> vrid 132
etbn1:/config/router/vrrp-cn2/#> version 3
etbn1:/config/router/vrrp-cn2/#> priority 150
etbn1:/config/router/vrrp-cn1/#> interval 1
etbn1:/config/router/vrrp-cn1/#> preempt delay 1
etbn1:/config/router/vrrp-cn2/#> end
etbn1:/config/router/#> end
etbn1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1:/#>
```

Now, repeat the exact steps above in the second ETB node. The only differences are the IP addresses on the CNs and the local id of the ETBN.

```
etbn2:#> configure
etbn2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn2:/config/vlan-10/#> untagged ethX3
etbn2:/config/vlan-10/#> end
etbn2:/config/#> vlan 20
Creating new VLAN vid:20 with name: vlan20
etbn2:/config/vlan-20/#> untagged ethX4
etbn2:/config/vlan-20/#> end
etbn2:/config/#> iface vlan10
etbn2:/config/iface-vlan10/#> no inet
etbn2:/config/iface-vlan10/#> inet static 10.10.0.2/18
etbn2:/config/iface-vlan10/inet-static-10.10.0.2/#> end
```

```
etbn2:/config/iface-vlan10/#> end
etbn2:/config/#> iface vlan20
etbn2:/config/iface-vlan20/#> no inet
etbn2:/config/iface-vlan20/#> inet static 10.20.0.2/18
etbn2:/config/iface-vlan20/inet-static-10.20.0.2/#> end
etbn2:/config/iface-vlan20/#> end
etbn2:/config/#> ttdp
Activating TTDP with default settings.
etbn2:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn2:/config/ttdp/#> node 1 ecn 1,2
etbn2:/config/ttdp/#> node 2 ecn 1,2
etbn2:/config/ttdp/#> ecn 1 vlan10
etbn2:/config/ttdp/#> ecn 2 vlan20
etbn2:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn2:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn2:/config/ttdp/#> local-id 2
etbn2:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn2:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn2:/config/vlan-2/#> no multicast-snooping
etbn2:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn2:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn2:/config/#> port ethX5
etbn2:/config/port-ethX5/#> speed-duplex 100-full
etbn2:/config/port-ethX5/#> no auto-negotiate
etbn2:/config/port-ethX5/#> fastlink
etbn2:/config/port-ethX5/#> mdix-mode mdi
etbn2:/config/port-ethX5/#> end
etbn2:/config/#> port ethX6
etbn2:/config/port-ethX6/#> speed-duplex 100-full
etbn2:/config/port-ethX6/#> no auto-negotiate
etbn2:/config/port-ethX6/#> fastlink
etbn2:/config/port-ethX6/#> mdix-mode mdi
etbn2:/config/port-ethX6/#> end
etbn2:/config/#> port ethX7
etbn2:/config/port-ethX7/#> speed-duplex 100-full
etbn2:/config/port-ethX7/#> no auto-negotiate
etbn2:/config/port-ethX7/#> fastlink
etbn2:/config/port-ethX7/#> mdix-mode mdi
etbn2:/config/port-ethX7/#> end
etbn2:/config/#> port ethX8
etbn2:/config/port-ethX8/#> speed-duplex 100-full
etbn2:/config/port-ethX8/#> no auto-negotiate
etbn2:/config/port-ethX8/#> fastlink
etbn2:/config/port-ethX8/#> mdix-mode mdi
etbn2:/config/port-ethX8/#> end
```

Repeat the same steps for ETBN2 with a lower priority to make it backup for both CN's. In this example the priority for both instances on ETBN2 is set to 100.

First, configure the VRRP instance for CN1:

```
etbn2:/config/#> router
etbn2:/config/router/#> vrrp cn1
Creating new VRRP instance: cn1
Invalid settings: Interface not set.
etbn2:/config/router/vrrp-cn1/#> iface vlan10
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn2:/config/router/vrrp-cn1/#> address 10.10.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn2:/config/router/vrrp-cn1/#> vrid 132
etbn2:/config/router/vrrp-cn1/#> version 3
etbn2:/config/router/vrrp-cn1/#> priority 100
etbn2:/config/router/vrrp-cn1/#> interval 1
etbn2:/config/router/vrrp-cn1/#> preempt delay 1
etbn2:/config/router/vrrp-cn1/#> end
```

Now configure the VRRP instance for CN2:

```
etbn2:/config/router/#> vrrp cn2
Creating new VRRP instance: cn2
Invalid settings: Interface not set.
etbn2:/config/router/vrrp-cn2/#> iface vlan20
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn2:/config/router/vrrp-cn2/#> address 10.20.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn2:/config/router/vrrp-cn2/#> vrid 132
etbn2:/config/router/vrrp-cn2/#> version 3
etbn2:/config/router/vrrp-cn2/#> priority 100
etbn2:/config/router/vrrp-cn1/#> interval 1
etbn2:/config/router/vrrp-cn1/#> preempt delay 1
etbn2:/config/router/vrrp-cn2/#> end
etbn2:/config/router/#> end
etbn2:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn2:/#>
```

Inauguration Result

Using the `show ttdp` command we now verify that we have recevied the intended network configuration. Highlighted in yellow are the values that are of interest.

```
etbn1:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 1
etbn-ip      : 10.128.0.1
local-id     : 1
own-mac      : 0c:65:8c:93:00:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0×b81af94
EtbTopoCnt   : 0×de33f1c2
```

```
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.10.0.1/18    1,2
   2     vlan20    10.20.0.1/18    1,2

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1..2    Direct     10.128.64.0/18
                                                          10.128.128.0/18

etbn1:/#>
```

```
etbn2:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 2
etbn-ip      : 10.128.0.2
local-id     : 2
own-mac      : 0c:7a:40:ee:00:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0×b81af94
EtbTopoCnt   : 0×de33f1c2
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.10.0.2/18    1,2
   2     vlan20    10.20.0.2/18    1,2

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1..2    Direct     10.128.64.0/18
                                                          10.128.128.0/18

etbn2:/#>
```

4.12.5. Multicast Between End Devices in Two Consists HowTo

This document describes how to configure two consits with one consist network each. Each consist network is managed by one ETBN. The topology used in this example is depicted in Figure 1.

```
  .--------------.     .--------------.
 /     cst1       \   /      cst2      \
 '----------------'   '----------------'
     .-------.             .-------.
   --+       +-------------+       +--     dir1 = (ethX5, ethX6)
<-dir1| ETBN1 |dir2-> <-dir1| ETBN1 |dir2-> dir2 = (ethX7, ethX8)
   --+       +-------------+       +--
     '-+-----'             '---+---'
       |                       |
     --+-+                   --+-+ CN1 vlan10 (ethX3)
         |                       |
     end-device             end-device
     sender                 receciver
```

Figure 1: Two consists with one ETBN managing one consist network each.

> **NOTE**
>
> The IEC 61375 standard defines both static multicast groups (train wide) and dynamic multicast groups (configurable). Static multicast groups are predefined in WeOS while dynamic groups are added via the CLI.

Configuration

To test both of these multicast types a dynamic group 239.193.0.1 is configured using the `mroute` command in the TTDP context.

```
etbn1cst1:#> configure
etbn1cst1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1cst1:/config/vlan-10/#> untagged ethX3
etbn1cst1:/config/vlan-10/#> end
etbn1cst1:/config/#> iface vlan10
etbn1cst1:/config/iface-vlan10/#> no inet
etbn1cst1:/config/iface-vlan10/#> inet static 10.10.0.1/18
etbn1cst1:/config/iface-vlan10/inet-static-10.10.0.1/#> end
etbn1cst1:/config/iface-vlan10/#> end
etbn1cst1:/config/#> ttdp
Activating TTDP with default settings.
etbn1cst1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1cst1:/config/ttdp/#> node 1 ecn 1
etbn1cst1:/config/ttdp/#> ecn 1 vlan10
etbn1cst1:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst1:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst1:/config/ttdp/#> local-id 1
etbn1cst1:/config/ttdp/#> mroute group 239.193.0.1 out 1,etb
Adding multicast route for 239.193.0.1
etbn1cst1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1cst1:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
```

```
etbn1cst1:/config/vlan-2/#> no multicast-snooping
etbn1cst1:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1cst1:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1cst1:/config/#> port ethX5
etbn1cst1:/config/port-ethX5/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX5/#> no auto-negotiate
etbn1cst1:/config/port-ethX5/#> fastlink
etbn1cst1:/config/port-ethX5/#> mdix-mode mdi
etbn1cst1:/config/port-ethX5/#> end
etbn1cst1:/config/#> port ethX6
etbn1cst1:/config/port-ethX6/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX6/#> no auto-negotiate
etbn1cst1:/config/port-ethX6/#> fastlink
etbn1cst1:/config/port-ethX6/#> mdix-mode mdi
etbn1cst1:/config/port-ethX6/#> end
etbn1cst1:/config/#> port ethX7
etbn1cst1:/config/port-ethX7/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX7/#> no auto-negotiate
etbn1cst1:/config/port-ethX7/#> fastlink
etbn1cst1:/config/port-ethX7/#> mdix-mode mdi
etbn1cst1:/config/port-ethX7/#> end
etbn1cst1:/config/#> port ethX8
etbn1cst1:/config/port-ethX8/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX8/#> no auto-negotiate
etbn1cst1:/config/port-ethX8/#> fastlink
etbn1cst1:/config/port-ethX8/#> mdix-mode mdi
etbn1cst1:/config/port-ethX8/#> end
etbn1cst1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1cst1:/#>
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

Now, repeat the exact steps above in ETB node 1 in consist 2. The only difference in the configuration is the `uuid`, which is 22222222-2222-2222-2222-222222222222 for consist 2.

```
etbn1cst2:#> configure
etbn1cst2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1cst2:/config/vlan-10/#> untagged ethX3
etbn1cst2:/config/vlan-10/#> end
etbn1cst2:/config/#> iface vlan10
etbn1cst2:/config/iface-vlan10/#> no inet
etbn1cst2:/config/iface-vlan10/#> inet static 10.10.0.1/18
etbn1cst2:/config/iface-vlan10/inet-static-10.10.0.1/#> end
etbn1cst2:/config/iface-vlan10/#> end
```

```
etbn1cst2:/config/#> ttdp
Activating TTDP with default settings.
etbn1cst2:/config/ttdp/#> uuid 22222222-2222-2222-2222-222222222222
etbn1cst2:/config/ttdp/#> node 1 ecn 1
etbn1cst2:/config/ttdp/#> ecn 1 vlan10
etbn1cst2:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst2:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst2:/config/ttdp/#> local-id 1
etbn1cst2:/config/ttdp/#> mroute group 239.193.0.1 out 1,etb
Adding multicast route for 239.193.0.1
etbn1cst2:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1cst2:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1cst2:/config/vlan-2/#> no multicast-snooping
etbn1cst2:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1cst2:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1cst2:/config/#> port ethX5
etbn1cst2:/config/port-ethX5/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX5/#> no auto-negotiate
etbn1cst2:/config/port-ethX5/#> fastlink
etbn1cst2:/config/port-ethX5/#> mdix-mode mdi
etbn1cst2:/config/port-ethX5/#> end
etbn1cst2:/config/#> port ethX6
etbn1cst2:/config/port-ethX6/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX6/#> no auto-negotiate
etbn1cst2:/config/port-ethX6/#> fastlink
etbn1cst2:/config/port-ethX6/#> mdix-mode mdi
etbn1cst2:/config/port-ethX6/#> end
etbn1cst2:/config/#> port ethX7
etbn1cst2:/config/port-ethX7/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX7/#> no auto-negotiate
etbn1cst2:/config/port-ethX7/#> fastlink
etbn1cst2:/config/port-ethX7/#> mdix-mode mdi
etbn1cst2:/config/port-ethX7/#> end
etbn1cst2:/config/#> port ethX8
etbn1cst2:/config/port-ethX8/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX8/#> no auto-negotiate
etbn1cst2:/config/port-ethX8/#> fastlink
etbn1cst2:/config/port-ethX8/#> mdix-mode mdi
etbn1cst2:/config/port-ethX8/#> end
etbn1cst2:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1cst2:/#>
```

NOTE

Validate connection between ETBNs by using ping command. Ping the inaugurated IP address of the neighbouring ETBN.

4.12.6. Port Settings for ETBN Backbone Ports HowTo

This document describes how to configure the port settings of the ETBN backbone ports.

See Figure 1 below

```
                    <-dir1           dir2->
         .-------.         .-------.         .-------.
Line A --+       +---------+       +---------+       +--
         | ETBN1 |         | ETBN2 |         | ETBN3 |
Line B --+       +---------+       +---------+       +--
         '-------'         '-------'         '-------'
Line A
       MDI      MDI      MDI      MDI      MDI      MDI
       Master   Follower Master   Follower Master   Follower

Line B
       MDI      MDI      MDI      MDI      MDI      MDI
       Follower Master   Follower Master   Follower Master

  dir1 = (ethX5, ethX6)
  dir2 = (ethX7, ethX8)
```

Figure 1: Mdix and master/follower configuration for backbone nodes.

> **NOTE**
>
> It is **strongly** recommended that all cables used to connect ETB nodes are crossover cables.

> **NOTE**
>
> Master follower configuration only applies when using Gbit ports.

Port configuration parameters

Port configuration parameters that may be used to improve link up and link down time for backbone ports.

Mdix-mode

Manual configuration allows for steps in connection phase to be skipped. MDIX configuration should be the same for all ports; either MDI for all ports or MDIX for all ports. IEC61375-2-5 specifies that one crossover shall occur on a line between two ports. This means that when a bypass-relay relays traffic, the relay needs to cancel out the crossover of one of the cables. This is done by making an additional crossover in the relay. With crossover cables, MDIX configuration should always be the same on both sides.

Gbit clock sync master/follower

Manual configuration allows for steps in connection phase to be skipped for 1000Mbit/s copper ethernet. Master/follower is not affected by crossover, hence every pair of ports that could be connected should include one master and one follower. This includes features such as bypass relays and rotation of ETBNs. Due to possible rotation of ETBNs, the direction of master/follower on line A and line B are opposite.

Speed-duplex

Setting fixed speed duplex forces the port to only communicate using this setting. This is required for disabling auto negotiation.

Auto negotiation

Disabling auto negotiation allows for faster connection times.

Fastlink

Configuring fastlink allows for faster link up/down times.

Configuration

This section show examples of how backbone ports could be configured for 100Mb/s and 1000Mb/s.

100Mb

- Set fixed speed 100Mb full duplex.
- Disable auto negotiation.
- Set MDIX-mode to MDI.
- Set fastlink enable.

Syntax

Example for ETBN1, the same configuration should be applied to all ETBNs.

```
etbn1:/#> configure
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> speed-duplex 100-full
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> speed-duplex 100-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> speed-duplex 100-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> speed-duplex 100-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink
etbn1:/config/port-ethX8/#> mdix-mode mdi
etbn1:/config/port-ethX8/#> end
etbn1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
```

Gbit

- Ports in dir1 line A should be master.
- Ports in dir1 line B should be follower.
- Ports in dir2 line A should be follower.
- Ports in dir2 line B should be master.
- Set fixed speed 1000Mb full duplex.
- Disable auto negotiation.

- Set fastlink 1000-t-up.
- Set MDIX-mode to MDI.

Syntax

Example for ETBN1, the same configuration should be applied to all ETBNs.

```
etbn1:/#> configure
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> gbit-cu-master
etbn1:/config/port-ethX5/#> speed-duplex 1000-full
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink 1000-t-up
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> no gbit-cu-master
etbn1:/config/port-ethX6/#> speed-duplex 1000-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink 1000-t-up
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> no gbit-cu-master
etbn1:/config/port-ethX7/#> speed-duplex 1000-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink 1000-t-up
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> gbit-cu-master
etbn1:/config/port-ethX8/#> speed-duplex 1000-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink 1000-t-up
etbn1:/config/port-ethX8/#> mdix-mode mdi
etbn1:/config/port-ethX8/#> end
etbn1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
```

4.12.7. Redundant ECSP Services HowTo

This document describes how to configure the ECSP services in a redundant train setup. The consists are depicted in Figure 1.

The result of the inauguration will be a network where ETBN1 in consist 1 has ETBN-ID 1, the IP-address 10.128.0.1 (based on the ETBN-ID and since the network is configured as TCMS). ETBN1 in consist 2 will attain ETBN-ID 2 and IP 10.128.0.2. ETBN2 in consist 2 will attain ETBN-ID 3 and IP 10.128.0.3. CN1 in consist 1 will attain the network IP 10.128.64.0/18, and CN1 in consist 2 IP 10.128.128.0/18.

Note that in the diagram below, the nodes are named ETBN1, ETBN2 and ETBN1, while in the configuration examples below, the device hostnames are `etbn1cst1`, `etbn1cst2` and `etbn2cst2`. This is to differentiate between a few concepts: the names in the diagram refer to the *local ETBN ID* (set using the `local-id` command), while the hostnames refer to both the *local ETBN ID* as well as the *consist* that the node is in.

```
     .-------.    .------------------------.
    /  cst1   \  /           cst2           \
    '---------'  '--------------------------'
     .-------.      .-------.    .-------.
   --+       +------+       +----+       +--      dir1 = (ethX5, ethX6)
<-dir1| ETBN1 |      | ETBN1 |    | ETBN2 |dir2->  dir2 = (ethX7, ethX8)
   --+       +------+       +----+       +--
     '---+---'      '---+---'    '---+---'
         |              |            |
         |          --+-+------------+---- cst2-cn1 vlan10 (ethX3)
         |            |
         |          ECSC2
         |
       --+---+----- cst1-cn1 vlan10 (ethX3)
         |
       ECSC1
```

Figure 1: Acceessing ECSP in a redundant consist setup.

> **NOTE**
>
> The two redundant nodes in consist 2 must have the same configuration with respect to `uuid` and `node`. In this example we also have the same `ecn` configuration. They must, however, have different values for the `local-id` setting.

> **NOTE**
>
> Both nodes in consist 2 need to have VRRP configured and enabled on the CN side.

Configuration

The first configuration example shows how to configure ETBN1 in consist 1.

```
etbn1cst1:#> configure
etbn1cst1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1cst1:/config/vlan-10/#> untagged ethX3
etbn1cst1:/config/vlan-10/#> end
etbn1cst1:/config/#> iface vlan10
etbn1cst1:/config/iface-vlan10/#> no inet
etbn1cst1:/config/iface-vlan10/#> inet static 10.0.0.10/18
```

```
etbn1cst1:/config/iface-vlan10/inet-static-10.0.0.10/#> end
etbn1cst1:/config/iface-vlan10/#> end
etbn1cst1:/config/#> ttdp
Activating TTDP with default settings.
etbn1cst1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1cst1:/config/ttdp/#> node 1 ecn 1
etbn1cst1:/config/ttdp/#> ecn 1 vlan10
etbn1cst1:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst1:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst1:/config/ttdp/#> local-id 1
etbn1cst1:/config/ttdp/#> ecsp-iface vlan10
etbn1cst1:/config/ttdp/#> ecsp-addr 10.0.0.10
etbn1cst1:/config/ttdp/#> ecsc-addr 10.0.0.42
etbn1cst1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1cst1:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1cst1:/config/vlan-2/#> no multicast-snooping
etbn1cst1:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1cst1:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1cst1:/config/#> port ethX5
etbn1cst1:/config/port-ethX5/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX5/#> no auto-negotiate
etbn1cst1:/config/port-ethX5/#> fastlink
etbn1cst1:/config/port-ethX5/#> mdix-mode mdi
etbn1cst1:/config/port-ethX5/#> end
etbn1cst1:/config/#> port ethX6
etbn1cst1:/config/port-ethX6/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX6/#> no auto-negotiate
etbn1cst1:/config/port-ethX6/#> fastlink
etbn1cst1:/config/port-ethX6/#> mdix-mode mdi
etbn1cst1:/config/port-ethX6/#> end
etbn1cst1:/config/#> port ethX7
etbn1cst1:/config/port-ethX7/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX7/#> no auto-negotiate
etbn1cst1:/config/port-ethX7/#> fastlink
etbn1cst1:/config/port-ethX7/#> mdix-mode mdi
etbn1cst1:/config/port-ethX7/#> end
etbn1cst1:/config/#> port ethX8
etbn1cst1:/config/port-ethX8/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX8/#> no auto-negotiate
etbn1cst1:/config/port-ethX8/#> fastlink
etbn1cst1:/config/port-ethX8/#> mdix-mode mdi
etbn1cst1:/config/port-ethX8/#> end
etbn1cst1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1cst1:/#>
```

Next, configure the ETBNs for consist 2. It contains of one consist network managed by two nodes.

Start with ETBN1 in consist 2 and configure vlans and the TTDP part.

```
etbn1cst2:#> configure
etbn1cst2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1cst2:/config/vlan-10/#> untagged ethX3
etbn1cst2:/config/vlan-10/#> end
etbn1cst2:/config/#> iface vlan10
etbn1cst2:/config/iface-vlan10/#> no inet
etbn1cst2:/config/iface-vlan10/#> inet static 10.0.0.10/18
etbn1cst2:/config/iface-vlan10/inet-static-10.0.0.10/#> end
etbn1cst2:/config/iface-vlan10/#> end
etbn1cst2:/config/#> ttdp
Activating TTDP with default settings.
etbn1cst2:/config/ttdp/#> uuid 22222222-2222-2222-2222-222222222222
etbn1cst2:/config/ttdp/#> node 1 ecn 1
etbn1cst2:/config/ttdp/#> node 2 ecn 1
etbn1cst2:/config/ttdp/#> ecn 1 vlan10
etbn1cst2:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn1cst2:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn1cst2:/config/ttdp/#> local-id 1
etbn1cst2:/config/ttdp/#> ecsp-iface vlan10
etbn1cst2:/config/ttdp/#> ecsp-addr 10.0.0.10
etbn1cst2:/config/ttdp/#> ecsp-leader-addr 10.0.0.100
etbn1cst2:/config/ttdp/#> ecsc-addr 10.0.0.52
etbn1cst2:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1cst2:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1cst2:/config/vlan-2/#> no multicast-snooping
etbn1cst2:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1cst2:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1cst2:/config/#> port ethX5
etbn1cst2:/config/port-ethX5/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX5/#> no auto-negotiate
etbn1cst2:/config/port-ethX5/#> fastlink
etbn1cst2:/config/port-ethX5/#> mdix-mode mdi
etbn1cst2:/config/port-ethX5/#> end
etbn1cst2:/config/#> port ethX6
etbn1cst2:/config/port-ethX6/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX6/#> no auto-negotiate
etbn1cst2:/config/port-ethX6/#> fastlink
etbn1cst2:/config/port-ethX6/#> mdix-mode mdi
etbn1cst2:/config/port-ethX6/#> end
etbn1cst2:/config/#> port ethX7
etbn1cst2:/config/port-ethX7/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX7/#> no auto-negotiate
etbn1cst2:/config/port-ethX7/#> fastlink
etbn1cst2:/config/port-ethX7/#> mdix-mode mdi
etbn1cst2:/config/port-ethX7/#> end
etbn1cst2:/config/#> port ethX8
```

```
etbn1cst2:/config/port-ethX8/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX8/#> no auto-negotiate
etbn1cst2:/config/port-ethX8/#> fastlink
etbn1cst2:/config/port-ethX8/#> mdix-mode mdi
etbn1cst2:/config/port-ethX8/#> end
```

The next step is to configure VRRP on each node in the redundant consist, i.e., consist 2. Start with ETBN1 which will have the MASTER role. The MASTER role is gained by setting the priority for the VRRP instance higher than on ETBN2 (see VRRP).

```
etbn1cst2:/config/#> router
etbn1cst2:/config/router/#> vrrp cn1
Creating new VRRP instance: cn1
Invalid settings: Interface not set.
etbn1cst2:/config/router/vrrp-cn1/#> iface vlan1
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn1cst2:/config/router/vrrp-cn1/#> address 10.0.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn1cst2:/config/router/vrrp-cn1/#> vrid 132
etbn1cst2:/config/router/vrrp-cn1/#> version 3
etbn1cst2:/config/router/vrrp-cn1/#> priority 150
etbn1cst2:/config/router/vrrp-cn1/#> interval 1
etbn1cst2:/config/router/vrrp-cn1/#> preempt delay 1
etbn1cst2:/config/router/vrrp-cn1/#> end
etbn1cst2:/config/router/#> end
etbn1cst2:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1cst2:/#>
```

Now, repeat the exact steps above in the ETBN2 in consist 2.

```
etbn2cst2:#> configure
etbn2cst2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn2cst2:/config/vlan-10/#> untagged ethX3
etbn2cst2:/config/vlan-10/#> end
etbn2cst2:/config/#> iface vlan10
etbn2cst2:/config/iface-vlan10/#> no inet
etbn2cst2:/config/iface-vlan10/#> inet static 10.0.0.20/18
etbn2cst2:/config/iface-vlan10/inet-static-10.0.0.20/#> end
etbn2cst2:/config/iface-vlan10/#> end
etbn2cst2:/config/#> ttdp
Activating TTDP with default settings.
etbn2cst2:/config/ttdp/#> uuid 22222222-2222-2222-2222-222222222222
etbn2cst2:/config/ttdp/#> node 1 ecn 1
etbn2cst2:/config/ttdp/#> node 2 ecn 1
etbn2cst2:/config/ttdp/#> ecn 1 vlan10
etbn2cst2:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
etbn2cst2:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
  in port settings for all agg ports.
```

```
etbn2cst2:/config/ttdp/#> local-id 2
etbn2cst2:/config/ttdp/#> ecsp-iface vlan10
etbn2cst2:/config/ttdp/#> ecsp-addr 10.0.0.10
etbn2cst2:/config/ttdp/#> ecsp-leader-addr 10.0.0.100
etbn2cst2:/config/ttdp/#> ecsc-addr 10.0.0.52
etbn2cst2:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn2cst2:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn2cst2:/config/vlan-2/#> no multicast-snooping
etbn2cst2:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn2cst2:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn2cst2:/config/#> port ethX5
etbn2cst2:/config/port-ethX5/#> speed-duplex 100-full
etbn2cst2:/config/port-ethX5/#> no auto-negotiate
etbn2cst2:/config/port-ethX5/#> fastlink
etbn2cst2:/config/port-ethX5/#> mdix-mode mdi
etbn2cst2:/config/port-ethX5/#> end
etbn2cst2:/config/#> port ethX6
etbn2cst2:/config/port-ethX6/#> speed-duplex 100-full
etbn2cst2:/config/port-ethX6/#> no auto-negotiate
etbn2cst2:/config/port-ethX6/#> fastlink
etbn2cst2:/config/port-ethX6/#> mdix-mode mdi
etbn2cst2:/config/port-ethX6/#> end
etbn2cst2:/config/#> port ethX7
etbn2cst2:/config/port-ethX7/#> speed-duplex 100-full
etbn2cst2:/config/port-ethX7/#> no auto-negotiate
etbn2cst2:/config/port-ethX7/#> fastlink
etbn2cst2:/config/port-ethX7/#> mdix-mode mdi
etbn2cst2:/config/port-ethX7/#> end
etbn2cst2:/config/#> port ethX8
etbn2cst2:/config/port-ethX8/#> speed-duplex 100-full
etbn2cst2:/config/port-ethX8/#> no auto-negotiate
etbn2cst2:/config/port-ethX8/#> fastlink
etbn2cst2:/config/port-ethX8/#> mdix-mode mdi
etbn2cst2:/config/port-ethX8/#> end
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

Repeat the same steps for ETBN2 with a lower priority to make it backup. In this example the priority for the instance on ETBN2 is set to 100.

```
etbn2cst2:/config/#> router
etbn2cst2:/config/router/#> vrrp cn1
Creating new VRRP instance: cn1
Invalid settings: Interface not set.
etbn2cst2:/config/router/vrrp-cn1/#> iface vlan1
```

```
Warning: No ports have been added to the configured VLAN.
VRRP will not work as expected with this configuration.
Invalid settings: Invalid IPv4 address.
etbn2cst2:/config/router/vrrp-cn1/#> address 10.0.0.100
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn2cst2:/config/router/vrrp-cn1/#> vrid 132
etbn2cst2:/config/router/vrrp-cn1/#> version 3
etbn2cst2:/config/router/vrrp-cn1/#> priority 100
etbn2cst2:/config/router/vrrp-cn1/#> interval 1
etbn2cst2:/config/router/vrrp-cn1/#> preempt delay 1
etbn2cst2:/config/router/vrrp-cn1/#> end
etbn2cst2:/config/router/#> end
etbn2cst2:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn2cst2:/#>
```

Communication Profile Configuration

Finally, for the ECSPs to function properly, a *CSTINFO data file* is needed on each device. See the Communication Profile documentation for more information about this file and its purpose.

Two such files are provided as examples for this HowTo: CSTINFO data file for consist 1 and CSTINFO data file for consist 2. They are, like the remainder of this document, representative but non-exhaustive examples.

On each node, the *CSTINFO data file* must be placed in the directory **/cfg/** with the name **cstinfo.json**. This can be done by using any available administrative interface, or by putting the file on the external USB media (in which case it will be available in the **/cfg/** directory after rebooting the device).

> **NOTE**
>
> The two devices in consist 2 - *etbn1cst2* and *etbn2cst2* - use identical CSTINFO data files. In this example, this is CSTINFO data file for consist 2. If nodes in the same consist have different CSTINFO data, or if only one of them has a data file while the other does not, results will be unpredictable.

> **NOTE**
>
> Ensure the filenames are correct. That is, the file should be named **cstinfo.json** and placed in **/cfg/** on each device.

When the above configuration is complete, the ECSP in cst1 uses IP address 10.0.0.10 on iface vlan10, while the ECSP in cst2 uses IP address 10.0.0.100 on iface vlan10.

To use the ECSP services on cst1, an ECSC client needs to be running on a host with IP address 10.0.0.42, and to use the ECSP services on cst2, an ECSC client needs to be running on a host with IP address 10.0.0.52.

Inauguration Result

Using the `show ttdp` command we now verify that we have recevied the intended network configuration. Highlighted in yellow are the values that are of interest.

```
etbn1cst1:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 1
etbn-ip      : 10.128.0.1
local-id     : 1
own-mac      : 0c:65:8c:93:00:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0×1d491819
EtbTopoCnt   : 0×80ea82ac
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.10/18    1

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct
      3      0c:c8:64:98:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1       Direct     10.128.64.0/18

etbn1cst1:/#>
```

```
etbn1cst2:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 2
etbn-ip      : 10.128.0.2
local-id     : 1
own-mac      : 0c:7a:40:ee:00:00
uuid         : 22222222-2222-2222-2222-222222222222
connTableCnt : 0×1d491819
EtbTopoCnt   : 0×f4b2eb52
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.10/18    2,3

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct
      3      0c:c8:64:98:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1       Direct     10.128.64.0/18
22222222-2222-2222-2222-222222222222   2..3    Direct     10.128.128.0/18

etbn1cst2:/#>
```

```
etbn2cst2:/#> show  ttdp
TTDP NODE
```

```
backbone-id  : 0 (tcms)
etbn-id      : 3
etbn-ip      : 10.128.0.3
local-id     : 2
own-mac      : 0c:c8:64:98:00:00
uuid         : 22222222-2222-2222-2222-222222222222
connTableCnt : 0x1d491819
EtbTopoCnt   : 0xf4b2eb52
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.2/18     2,3

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct
      3      0c:c8:64:98:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1       Direct     10.128.64.0/18
22222222-2222-2222-2222-222222222222   2..3    Direct     10.128.128.0/18

etbn2cst2:/#>
```

Additionally, we can use the `nslookup` command to verify that the communication profile configuration has been read. A set of URIs has been chosen to verify that the DNS entries has been created.

```
etbn1cst1:/#> nslookup devDCU2.anyVeh.lCst.lTrn
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

Name:      devDCU2.anyVeh.lCst.lTrn
Address 1: 10.0.0.111
etbn1cst1:/#> nslookup devHVAC.anyVeh.lCst.lTrn
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

Name:      devHVAC.anyVeh.lCst.lTrn
Address 1: 10.0.0.100
etbn1cst1:/#> nslookup devCCUO2.anyVeh.lCst.lTrn
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

Name:      devCCUO2.anyVeh.lCst.lTrn
Address 1: 10.0.0.11
etbn1cst1:/#>
```

4.12.8. Non Redundant ECSP Services HowTo

This document describes how to configure the ECSP services in a non redundant train setup. The consists are depicted in Figure 1.

The result of the inauguration will be a network where ETBN1 in consist 1 has ETBN-ID 1, the IP-address 10.128.0.1 (based on the ETBN-ID and since the network is configured as TCMS). ETBN1 in consist 2 will attain ETBN-ID 2 and IP 10.128.0.2. CN1 in consist 1 will attain the network IP 10.128.64.0/18, and CN1 in consist 2 IP 10.128.128.0/18.

Note that in the diagram below, the nodes are named ETBN1 and ETBN2 respectively, while in the configuration examples below, the device hostnames are `etbn1cst1` and `etbn1cst2`. This is to differentiate between a few concepts: the names in the diagram refer to the *local ETBN ID* (set using the `local-id` command), while the hostnames refer to both the *local ETBN ID* as well as the *consist* that the node is in.

```
      .-------.    .-------.
     /  cst1   \  /  cst2   \
     '---------'  '---------'
      .-------.    .-------.
    --+       +----+       +--     dir1 = (ethX5, ethX6)
<-dir1| ETBN1 |    | ETBN2 |dir2-> dir2 = (ethX7, ethX8)
    --+       +----+       +--
      '---+---'    '---+---'
          |            |
          |       --+--+---- cst2-cn1 vlan10 (ethX3)
          |         |
          |       ECSC2
          |
    --+---+----- cst1-cn1 vlan10 (ethX3)
      |
    ECSC1
```

Figure 1: Accessing the ECSP's in two consists.

Configuration

The first configuration example shows how to configure ETBN1 in consist 1.

```
etbn1cst1:#> configure
etbn1cst1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1cst1:/config/vlan-10/#> untagged ethX3
etbn1cst1:/config/vlan-10/#> end
etbn1cst1:/config/#> iface vlan10
etbn1cst1:/config/iface-vlan10/#> no inet
etbn1cst1:/config/iface-vlan10/#> inet static 10.0.0.10/18
etbn1cst1:/config/iface-vlan10/inet-static-10.0.0.10/#> end
etbn1cst1:/config/iface-vlan10/#> end
etbn1cst1:/config/#> ttdp
Activating TTDP with default settings.
etbn1cst1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1cst1:/config/ttdp/#> node 1 ecn 1
etbn1cst1:/config/ttdp/#> ecn 1 vlan10
etbn1cst1:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst1:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst1:/config/ttdp/#> local-id 1
etbn1cst1:/config/ttdp/#> mroute group 239.193.0.1 out 1,etb
```

```
Adding multicast route for 239.193.0.1
etbn1cst1:/config/ttdp/#> ecsp-iface vlan10
etbn1cst1:/config/ttdp/#> ecsp-addr 10.0.0.10
etbn1cst1:/config/ttdp/#> ecsc-addr 10.0.0.42
etbn1cst1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1cst1:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1cst1:/config/vlan-2/#> no multicast-snooping
etbn1cst1:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1cst1:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1cst1:/config/#> port ethX5
etbn1cst1:/config/port-ethX5/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX5/#> no auto-negotiate
etbn1cst1:/config/port-ethX5/#> fastlink
etbn1cst1:/config/port-ethX5/#> mdix-mode mdi
etbn1cst1:/config/port-ethX5/#> end
etbn1cst1:/config/#> port ethX6
etbn1cst1:/config/port-ethX6/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX6/#> no auto-negotiate
etbn1cst1:/config/port-ethX6/#> fastlink
etbn1cst1:/config/port-ethX6/#> mdix-mode mdi
etbn1cst1:/config/port-ethX6/#> end
etbn1cst1:/config/#> port ethX7
etbn1cst1:/config/port-ethX7/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX7/#> no auto-negotiate
etbn1cst1:/config/port-ethX7/#> fastlink
etbn1cst1:/config/port-ethX7/#> mdix-mode mdi
etbn1cst1:/config/port-ethX7/#> end
etbn1cst1:/config/#> port ethX8
etbn1cst1:/config/port-ethX8/#> speed-duplex 100-full
etbn1cst1:/config/port-ethX8/#> no auto-negotiate
etbn1cst1:/config/port-ethX8/#> fastlink
etbn1cst1:/config/port-ethX8/#> mdix-mode mdi
etbn1cst1:/config/port-ethX8/#> end
etbn1cst1:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1cst1:/#>
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

Now, repeat the exact steps above in the second ETB node. The only difference is the uuid which now will be **uuid 22222222-2222-2222-2222-222222222222**.

```
etbn1cst2:#> configure
etbn1cst2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1cst2:/config/vlan-10/#> untagged ethX3
etbn1cst2:/config/vlan-10/#> end
etbn1cst2:/config/#> iface vlan10
etbn1cst2:/config/iface-vlan10/#> no inet
etbn1cst2:/config/iface-vlan10/#> inet static 10.0.0.10/18
etbn1cst2:/config/iface-vlan10/inet-static-10.0.0.10/#> end
etbn1cst2:/config/iface-vlan10/#> end
etbn1cst2:/config/#> ttdp
Activating TTDP with default settings.
etbn1cst2:/config/ttdp/#> uuid 22222222-2222-2222-2222-222222222222
etbn1cst2:/config/ttdp/#> node 1 ecn 1
etbn1cst2:/config/ttdp/#> ecn 1 vlan10
etbn1cst2:/config/ttdp/#> dir1 ethX5,ethX6
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst2:/config/ttdp/#> dir2 ethX7,ethX8
Remember to manually set mdi/mdi-x mode
 in port settings for all agg ports.
etbn1cst2:/config/ttdp/#> local-id 1
etbn1cst2:/config/ttdp/#> mroute group 239.193.0.1 out 1,etb
Adding multicast route for 239.193.0.1
etbn1cst2:/config/ttdp/#> ecsp-iface vlan10
etbn1cst2:/config/ttdp/#> ecsp-addr 10.0.0.10
etbn1cst2:/config/ttdp/#> ecsc-addr 10.0.0.42
etbn1cst2:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically
  if TTDP is deactivated.
etbn1cst2:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1cst2:/config/vlan-2/#> no multicast-snooping
etbn1cst2:/config/vlan-2/#> untagged lag-dir1,lag-dir2
etbn1cst2:/config/vlan-2/#> end
```

Continue by configuring the ports.

```
etbn1cst2:/config/#> port ethX5
etbn1cst2:/config/port-ethX5/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX5/#> no auto-negotiate
etbn1cst2:/config/port-ethX5/#> fastlink
etbn1cst2:/config/port-ethX5/#> mdix-mode mdi
etbn1cst2:/config/port-ethX5/#> end
etbn1cst2:/config/#> port ethX6
etbn1cst2:/config/port-ethX6/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX6/#> no auto-negotiate
etbn1cst2:/config/port-ethX6/#> fastlink
etbn1cst2:/config/port-ethX6/#> mdix-mode mdi
etbn1cst2:/config/port-ethX6/#> end
etbn1cst2:/config/#> port ethX7
etbn1cst2:/config/port-ethX7/#> speed-duplex 100-full
etbn1cst2:/config/port-ethX7/#> no auto-negotiate
etbn1cst2:/config/port-ethX7/#> fastlink
etbn1cst2:/config/port-ethX7/#> mdix-mode mdi
etbn1cst2:/config/port-ethX7/#> end
etbn1cst2:/config/#> port ethX8
etbn1cst2:/config/port-ethX8/#> speed-duplex 100-full
```

```
etbn1cst2:/config/port-ethX8/#> no auto-negotiate
etbn1cst2:/config/port-ethX8/#> fastlink
etbn1cst2:/config/port-ethX8/#> mdix-mode mdi
etbn1cst2:/config/port-ethX8/#> end
etbn1cst2:/config/#> end
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1cst2:/#>
```

Communication Profile Configuration

Finally, for the ECSPs to function properly, a *CSTINFO data file* is needed on each device. See the Communication Profile documentation for more information about this file and its purpose.

Two such files are provided as examples for this HowTo: CSTINFO data file for consist 1 and CSTINFO data file for consist 2. They are, like the remainder of this document, representative but non-exhaustive examples.

On each node, the *CSTINFO data file* must be placed in the directory **/cfg/** with the name **cstinfo.json**. This can be done by using any available administrative interface, or by putting the file on the external USB media (in which case it will be available in the **/cfg/** directory after rebooting the device).

> **NOTE**
>
> Ensure the filenames are correct. That is, the file should be named **cstinfo.json** and placed in **/cfg/** on each device.

When the above configuration is complete, the ECSP in cst1 uses IP address 10.0.0.10 on iface vlan10, while the ECSP in cst2 uses IP address 10.0.0.20 on iface vlan10.

To use the ECSP services on cst1, an ECSC client (ECSC1 in figure 1) needs to be running on a host with IP address 10.0.0.42 To use the ECSP services on cst2, an ECSC client (ECSC2 in figure 1) needs to be running on a host with IP address 10.0.0.42 (but connected to the correct consist net of course).

Inauguration Result

Using the `show ttdp` command we now verify that we have recevied the intended network configuration. Highlighted in yellow are the values that are of interest.

```
etbn1cst1:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 1
etbn-ip      : 10.128.0.1
local-id     : 1
own-mac      : 0c:65:8c:93:00:00
uuid         : 11111111-1111-1111-1111-111111111111
connTableCnt : 0×b81af94
EtbTopoCnt   : 0×ec518201
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK
```

```
CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.10/18    1

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1       Direct     10.128.64.0/18
22222222-2222-2222-2222-222222222222   2       Direct     10.128.128.0/18

etbn1cst1:/#>
```

```
etbn1cst2:/#> show ttdp
TTDP NODE
backbone-id  : 0 (tcms)
etbn-id      : 2
etbn-ip      : 10.128.0.2
local-id     : 1
own-mac      : 0c:7a:40:ee:00:00
uuid         : 22222222-2222-2222-2222-222222222222
connTableCnt : 0×b81af94
EtbTopoCnt   : 0×ec518201
Inauguration : INAUGURATED
Inhibition   : Self: Off,  Others: Off,  Train: Off
Diagnostic   : OK

CST-NET  IFACE     IP              CONNECTED ETBNs
   1     vlan10    10.0.0.10/18    2

NODE ID      MAC ADDRESS            ORIENTATION
      1      0c:65:8c:93:00:00      Direct
      2      0c:7a:40:ee:00:00      Direct

CONSIST                                NODES   DIRECTION  NETWORKS
11111111-1111-1111-1111-111111111111   1       Direct     10.128.64.0/18
22222222-2222-2222-2222-222222222222   2       Direct     10.128.128.0/18

etbn1cst2:/#>
```

Additionally, we can use the `nslookup` command to verify that the communication profile configuration has been read. A set of URIs has been choosen to verify that the DNS entries has been created.

```
etbn1cst1:/#> nslookup devDCU2.anyVeh.lCst.lTrn
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

Name:      devDCU2.anyVeh.lCst.lTrn
Address 1: 10.0.0.111
etbn1cst1:/#> nslookup devHVAC.anyVeh.lCst.lTrn
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

Name:      devHVAC.anyVeh.lCst.lTrn
Address 1: 10.0.0.100
etbn1cst1:/#> nslookup devCCUO2.anyVeh.lCst.lTrn
```

```
Server:    127.0.0.1
Address 1: 127.0.0.1 localhost

Name:      devCCUO2.anyVeh.lCst.lTrn
Address 1: 10.0.0.11
etbn1cst1:/#>
```

4.12.9. Split ECN over ETB HowTo (advanced)

This document describes a specific use case, where two ECNs and a non-ECN network are overlaid on top of the ETB, thus not needing dedicated ECN cabling between the ETBNs.

This can be useful in certain cases where adding another physical cable (in addition to the regular ETB cabling) is difficult between the ETBNs, or in cases where a direct connection between the ETBNs is prevented by the ECN architecture.

In the case presented below, there are two ETBNs (ETBN1, ETBN2) connected to the same ECN, without dedicated ECN cabling. There is also a redundant pair of externally managed firewall devices (GW1, GW2), splitting the ECN into two subnets. As the firewall devices are considered third-party devices, only configuration hints for them will be provided. There are also two end devices, or hosts (H1, H2) added for illustration.

The VLAN used for the ECN in this scenario will be **vlan10**. When using the “ECN over ETB” functionality, it is important that the same VID be used for each ECN across all participating ETBNs in the consist, since the ECN traffic is sent across the ETB using VLAN tags.

This use case utilises the “ECN subnetting” functionality. While not strictly needed to use the “ECN over ETB” function, the two can be used together for greater effect. In effect, this feature lets the ECN be split into two distinct subnets locally, while still presented as one ECN to the rest of the train.

There are three VRRP virtual routers used in this example, the ETBNs being one of them (this is the usual setup for redundant ECNs when using TTDP: see the redundant ETBN HowTo), with the VIP address **10.0.0.10**. This address is used by every device on the combined ECN as the default route, in order to facilitate traffic via the ETB, as in a normal TTDP scenario. The other two VRRP routers are handled by the firewall devices, one instance on each half of the ECN. The VIP addresses of each of these is then used by devices to reach the other half of the ECN. These are **10.0.0.100** for the upper part of the ECN, and **10.0.32.100** for the lower part.

Schematically the network topology looks as follows.

```
        .--------------------.
       /         cst1         \
       '----------------------'
        .-------.    .-------.
      --+       +----+       +--     dir1 = (ethX5, ethX6)
<-dir1| ETBN1 |    | ETBN2 |dir2-> dir2 = (ethX7, ethX8)
      --+       +----+       +--
        '---+---'    '---+---'
       ethX3|.1     ethX3|.2
            |            |
     .----. |            | .----.
     | H1 +-+ .11    .12 +-+ H2 |             ^
     '----' |            | '----'             |
            |            |                    |
        eth0|.21     eth0|.22          "Upper" part of ECN
        .---+---.    .---+---.         (disjoint)
        |       |    |       |         10.0.0.0/19
        |  GW1  |    |  GW2  |         -------------------
        |       |    |       |         "Lower" part of ECN
        '---+---'    '---+---'         (joint)
        eth1|.21     eth1|.22          10.0.32.0/19
            |            |                    |
        ----+-----+------+----                |
                  | .99                       V
               .--+---.
               | ECSC |
               '------'
```

Figure 1: Consist network layout.

> **NOTE**
>
> As far as the IEC61375 stack is concerned, there is just one ECN in the consist. This is what TTDP communicates to any other coupled consists. This entire ECN has the address space 10.0.0.0/18, and it will be mapped to one /18 subnet in the train scope, as usual. Internally, however, this ECN is split up into two /19 networks, on either side of the FW devices. This requires some special configuration on the ETBNs, and is made possible by the “ECN subnetting” feature.

> **NOTE**
>
> Note also that the upper part of the ECN is disjoint - there is no direct path between ED1 and ED2 on the 10.0.0.0/19 network. For the two to communicate directly, the network needs to be overlaid on top of the ETB cabling. This is made possible by the “ECN over ETB” feature.

Configuration

Non-WeOS configuration

The network shown in Figure 1 contains some non-WeOS devices, for illustration as well as context. It is assumed that they are configured as follows. Note that link-local routes are not explicitly specified (e.g., it is assumed that a device with an address of a subnet has a link-local route to that subnet). Note also that ED1 and ED2 can be configured using DHCP, including static routes - see the DHCP Server HowTo and the DHCP Server documentation, in particular the `route` subnet setting.

```
ED1: address 10.0.0.11/19
     static route to 10.0.32.0/19 via 10.0.0.100
     default gateway 10.0.0.10

ED2: address 10.0.0.12/19
     static route to 10.0.32.0/19 via 10.0.0.100
     default gateway 10.0.0.10

GW1: eth0 (upper) address 10.0.0.21/19
     eth1 (lower) address 10.0.32.21/19
     IP forwarding enabled between eth0 and eth1
     default gateway 10.0.0.10
     VRRP configuration:
       VRRP instance 1:
         base interface: eth0 (upper)
         address 10.0.0.100
       VRRP instance 2:
         base interface: eth1 (lower)
         address 10.0.32.100

GW2: eth0 (upper) address 10.0.0.22/19
     eth1 (lower) address 10.0.32.22/19
     IP forwarding enabled between eth0 and eth1
     default gateway 10.0.0.10
     VRRP configuration:
       VRRP instance 1:
         base interface: eth0 (upper)
         address 10.0.0.100
       VRRP instance 2:
         base interface: eth1 (lower)
         address 10.0.32.100

ECSC: address 10.0.32.99/19
      default gateway 10.0.32.100
      ECSP address 10.0.0.10
```

ETBN1 configuration

Start by configuring the required VLAN and VRRP settings on ETBN1. Also, add a route to 10.0.32.0/19 (the lower part of the ECN), using the upper VIP address of the GW devices as the router.

```
etbn1:/#> config
etbn1:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn1:/config/vlan-10/#> untagged ethX3
Moving untagged port eth4 from vid 1 to vid 10.
etbn1:/config/vlan-10/#> end
etbn1:/config/#> iface vlan10
etbn1:/config/iface-vlan10/#> no inet
etbn1:/config/iface-vlan10/#> inet static 10.0.0.1/19
etbn1:/config/iface-vlan10/inet-static-10.0.0.1/#> end
etbn1:/config/iface-vlan10/#> end
etbn1:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn1:/config/vlan-2/#> no multicast-snooping
etbn1:/config/vlan-2/#> end
etbn1:/config/#> router
etbn1:/config/router/#> vrrp ecn1
Creating new VRRP instance: ecn1
Invalid settings: Interface not set.
etbn1:/config/router/vrrp-ecn1/#> iface vlan10
Invalid settings: Invalid IPv4 address.
etbn1:/config/router/vrrp-ecn1/#> address 10.0.0.10
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn1:/config/router/vrrp-ecn1/#> vrid 1
etbn1:/config/router/vrrp-ecn1/#> priority 100
etbn1:/config/router/vrrp-ecn1/#> end
etbn1:/config/router/#> end
etbn1:/config/#> ip
etbn1:/config/ip/#> route 10.0.32.0/19 10.0.0.100
etbn1:/config/ip/#> end
etbn1:/config/#>
```

Then, set up the necessary ETB port settings.

```
etbn1:/config/#>
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> speed-duplex 100-full
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> speed-duplex 100-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> speed-duplex 100-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
```

```
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> speed-duplex 100-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink
etbn1:/config/port-ethX8/#> mdix-mode mdi
etbn1:/config/port-ethX8/#> end
etbn1:/config/#>
```

> **NOTE**
>
> In this HowTo, it is assumed that the ETB uses crossover cabling. For this reason, all ETB ports on both ETBNs use the same MDI-X mode. If non-crossover cabling is used, these settings must be changed for two of the inward-facing ETB ports.

Finally, set up the TTDP configuration and add the ETB link aggregates to the ETB VLAN. Note specifically the use of the `ecn-over-etb` setting, where we indicate that VID 10 is to be forwarded over the ETB cabling.

> **NOTE**
>
> There is an extra parameter, `18`, given to the `ecn` command. This is normally omitted in the TTDP configuration, but is required here for the "ECN subnetting" functionality. What this does is to let TTDP know that even though the device uses a /19 address on `vlan10`, the ECN, as presented in TTDP, should use the entire /18 space (the other /19 net is reachable via the GW devices).

```
etbn1:/config/#>
etbn1:/config/#> ttdp
Activating TTDP with default settings.
etbn1:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn1:/config/ttdp/#> node 1 ecn 1
etbn1:/config/ttdp/#> node 2 ecn 1
etbn1:/config/ttdp/#> ecn 1 vlan10 18
etbn1:/config/ttdp/#> dir1 ethX5 ethX6
etbn1:/config/ttdp/#> dir2 ethX7 ethX8
etbn1:/config/ttdp/#> local-id 1
etbn1:/config/ttdp/#> ecsp-iface vlan10
etbn1:/config/ttdp/#> ecsp-leader-addr 10.0.0.3
etbn1:/config/ttdp/#> ecsc-addr 10.0.32.99
etbn1:/config/ttdp/#> ecn-over-etb 10
etbn1:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically if TTDP is deactivated
etbn1:/config/#> vlan 2
etbn1:/config/vlan-2/#> untagged lag-dir1 lag-dir2
etbn1:/config/vlan-2/#> leave
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn1:/#> cp ru st
etbn1:/#>
```

ETBN2 configuration

The configuration for ETBN2 is similar. The changed values have been marked with **yellow** text. The static interface address and VRRP priority are different.

```
etbn2:/#> config
etbn2:/config/#> vlan 10
Creating new VLAN vid:10 with name: vlan10
etbn2:/config/vlan-10/#> untagged ethX3
Moving untagged port eth4 from vid 1 to vid 10.
etbn2:/config/vlan-10/#> end
etbn2:/config/#> iface vlan10
etbn2:/config/iface-vlan10/#> no inet
etbn2:/config/iface-vlan10/#> inet static 10.0.0.2/19
etbn2:/config/iface-vlan10/inet-static-10.0.0.2/#> end
etbn2:/config/iface-vlan10/#> end
etbn2:/config/#> vlan 2
Creating new VLAN vid:2 with name: vlan2
etbn2:/config/vlan-2/#> no multicast-snooping
etbn2:/config/vlan-2/#> end
etbn2:/config/#> router
etbn2:/config/router/#> vrrp ecn1
Creating new VRRP instance: ecn1
Invalid settings: Interface not set.
etbn2:/config/router/vrrp-ecn1/#> iface vlan10
Invalid settings: Invalid IPv4 address.
etbn2:/config/router/vrrp-ecn1/#> address 10.0.0.10
Invalid settings: VRID invalid. Valid values: 1 to 255.
etbn2:/config/router/vrrp-ecn1/#> vrid 1
etbn2:/config/router/vrrp-ecn1/#> priority 80
etbn2:/config/router/vrrp-ecn1/#> end
etbn2:/config/router/#> end
etbn2:/config/#> ip
etbn2:/config/ip/#> route 10.0.32.0/19 10.0.0.100
etbn2:/config/ip/#> end
etbn2:/config/#>
```

The port configuration for ETBN2 is identical to that of ETBN1.

```
etbn1:/config/#>
etbn1:/config/#> port ethX5
etbn1:/config/port-ethX5/#> speed-duplex 100-full
etbn1:/config/port-ethX5/#> no auto-negotiate
etbn1:/config/port-ethX5/#> fastlink
etbn1:/config/port-ethX5/#> mdix-mode mdi
etbn1:/config/port-ethX5/#> end
etbn1:/config/#> port ethX6
etbn1:/config/port-ethX6/#> speed-duplex 100-full
etbn1:/config/port-ethX6/#> no auto-negotiate
etbn1:/config/port-ethX6/#> fastlink
etbn1:/config/port-ethX6/#> mdix-mode mdi
etbn1:/config/port-ethX6/#> end
etbn1:/config/#> port ethX7
etbn1:/config/port-ethX7/#> speed-duplex 100-full
etbn1:/config/port-ethX7/#> no auto-negotiate
etbn1:/config/port-ethX7/#> fastlink
etbn1:/config/port-ethX7/#> mdix-mode mdi
etbn1:/config/port-ethX7/#> end
etbn1:/config/#> port ethX8
etbn1:/config/port-ethX8/#> speed-duplex 100-full
etbn1:/config/port-ethX8/#> no auto-negotiate
etbn1:/config/port-ethX8/#> fastlink
etbn1:/config/port-ethX8/#> mdix-mode mdi
```

```
etbn1:/config/port-ethX8/#> end
etbn1:/config/#>
```

For the TTDP configuration, only the local ETBN id must be changed. Save and commit the configuration.

```
etbn2:/config/#>
etbn2:/config/#> ttdp
Activating TTDP with default settings.
etbn2:/config/ttdp/#> uuid 11111111-1111-1111-1111-111111111111
etbn2:/config/ttdp/#> node 1 ecn 1
etbn2:/config/ttdp/#> node 2 ecn 1
etbn2:/config/ttdp/#> ecn 1 vlan10 18
etbn2:/config/ttdp/#> dir1 ethX5 ethX6
etbn2:/config/ttdp/#> dir2 ethX7 ethX8
etbn2:/config/ttdp/#> local-id 2
etbn2:/config/ttdp/#> ecsp-iface vlan10
etbn2:/config/ttdp/#> ecsp-leader-addr 10.0.0.3
etbn2:/config/ttdp/#> ecsc-addr 10.0.32.99
etbn2:/config/ttdp/#> ecn-over-etb 10
etbn2:/config/ttdp/#> end
TTDP: Created VLAN 492. Note that this VLAN will not be deleted automatically if TTDP is deactivated
etbn2:/config/#> vlan 2
etbn2:/config/vlan-2/#> untagged lag-dir1 lag-dir2
etbn2:/config/vlan-2/#> leave
Applying configuration.
Configuration activated.
Remember "copy run start" to save to flash (NVRAM).
etbn2:/#> cp ru st
etbn2:/#>
```

Result

Once the configuration is applied and saved, the TTDP stack should inaugurate, and once that happens the “ECN over ETB” function will overlay ECN traffic between ETBN1 and ETBN2 on top of the ETB. To ensure that this is the case, verify connectivity between H1 and H2 (e.g., via `ping`).

Ensure that VRRP is operating correctly - exactly one of the ETBNs should be in MASTER state (use `show vrrp` on the ETBNs to verify this), and the VRRP instances on the GW devices should also behave correctly.

Also, verify connectivity between the ECSC and the ETBNs, using `ping` and/or the ECSP interfaces.

4.13. Policy

4.13.1. Policy Traffic Filtering HowTo

This document details an example and use-case for how the policy traffic filter can be used.

For an overview of what Policy Traffic Filtering is and how it may be configured, refer to:

- Policy Traffic Filtering Configuration Guide.

```
iface: vlan10, 10.0.0.0/16                                iface: vlan20, 192.168.1.0/24
.-------------------------------------.                   .------.
|                 S1                  | eth4     eth4 |      |
| eth1           eth2            eth3 +---------------+  S2  |
|VLAN10         VLAN10          VLAN20|  VLAN Trunk   |VLAN20|
'--+---------------+---------------+--'                  '--+---'
   |.1             |.1             |.2                      |.1
   |               |               |                        |
   |.11            |.12            |.20                     |.25
.--+--.         .--+--.         .--+--.                  .--+--.
|  H1 |         |  H2 |         |  H3 |                  |  H4 |
'-----'         '-----'         '-----'                  '-----'
```

Figure 1: Policy Traffic Filtering setup with four end devices connected through two switches.

In an individual network there might be a desire to accept or deny individual hosts or devices. To ensure that applied rules always are valid it is recommended to match on the MAC address of the host. Matching on the IP address of the host can cause rules to become invalid if dynamic address allocation is used (DHCP), unless static IP address allocation is used.

In the following example the desire is to:

- Only traffic with a destination IP to H3 and a source IP address originating from vlan 20, may ingress interface eth4 on S1.
- Only allow MAC address of H2 to egress interface eth4 on S1 from VLAN 10.

Configuration

First we start of by creating our policy groups and filters.

```
example:/#> configure
example:/config/#> policy
example:/config/policy/#> ingress-policy net
example:/config/policy/ingress-policy-1/#> filter 1
example:/config/policy/ingress-policy-1/filter-1/#> match saddr 192.168.1.0/24
example:/config/policy/ingress-policy-1/filter-1/#> match daddr 192.168.1.20
example:/config/policy/ingress-policy-1/filter-1/#> action accept
example:/config/policy/ingress-policy-1/filter-1/#> exit
example:/config/policy/ingress-policy-1/#> description Allow VLAN 20 to H3
example:/config/policy/ingress-policy-1/#> exit
example:/config/policy/#> egress-policy management
example:/config/policy/egress-policy-1/#> filter 1
example:/config/policy/egress-policy-1/filter-1/#> match smac 00:11:b4:39:8e:20
example:/config/policy/egress-policy-1/filter-1/#> action accept
example:/config/policy/egress-policy-1/filter-1/#> exit
example:/config/policy/egress-policy-1/#> description Allow management from H2
example:/config/policy/egress-policy-1/#> exit
example:/config/policy/#>
```

Now our policies have been created. However, they are yet to be applied to any port.

```
example:/config/policy/#> port eth4
example:/config/policy/port-eth4/#> ingress-chain
```

```
example:/config/policy/port-eth4/ingress-chain/#> ingress net vid 20
example:/config/policy/port-eth4/ingress-chain/#> default-action drop
example:/config/policy/port-eth4/ingress-chain/#> exit
example:/config/policy/port-eth4/#> exit
example:/config/policy/#> port eth4
example:/config/policy/port-eth4/#> egress-chain
example:/config/policy/port-eth4/egress-chain/#> egress management
example:/config/policy/port-eth4/egress-chain/#> default-action drop
example:/config/policy/port-eth4/egress-chain/#> exit
example:/config/policy/port-eth4/#> exit
example:/config/policy/#> leave
```

Status

In order to see the currently applied policies and rules, the following command can be used:

```
example:/#> show policy

Enabled : Yes

╒ Ingress-policies ═══════════════════════════════════════════════════
│NAME                  FILTER  POS     ACTION      MATCH    VALUE
│1: net
│                      1       1       Accept      saddr:   192.168.1.0/24
│                                                  daddr:   192.168.1.20/32
│Description:
│ Allow VLAN 20 to H3
└──────────────────────────────────────────────────────────────────────

╒ Egress-policies ═══════════════════════════════════════════════════
│NAME                  FILTER  POS     ACTION      MATCH    VALUE
│1: management
│                      1       1       Accept      smac:    00:11:b4:39:8e:20
│Description:
│ Allow management ...
└──────────────────────────────────────────────────────────────────────

╒ Ingress-chains ═══════════════════════════════════════════════════
│PORTS             POS     VLAN    POLICY
│eth4
│                  1       20      1: net
│                          all     Default-Drop
└──────────────────────────────────────────────────────────────────────

╒ Egress-chains ═══════════════════════════════════════════════════
│PORTS             POS     VLAN    POLICY
│eth4
│                  1       all     1: management
│                          all     Default-Drop
└──────────────────────────────────────────────────────────────────────
```

4.14. Application Hosting

4.14.1. Application Hosting HowTo

This document outlines some practical examples of how to:

- Enable the feature in WeOS
- How to prepare the device
- How to import application images
- How to configure application hosting and applications
- How to control applications & monitor status

For a general overview of the feature, refer to the configuration guide:

- Application Hosting

Enable Application Hosting

To enable Application Hosting feature in WeOS, a special license package must be obtained and installed.

To obtain installation package, contact Westermo Network Technologies AB.

Having obtained the package, install it in WeOS. After successful installation, reboot the system for the effect to take place.

```
example:/package/#> install proto tftp 198.19.10.1 app-hosting-license-signed.wmopkg
Attempting to download file app-hosting-license-signed.wmopkg, using TFTP

Connecting to 198.19.10.1 (198.19.10.1:69)
app-hosting-license-signed.wmopkg:

0%       25%       50%       75%       100%
├─────────+─────────+─────────+─────────┤
├═══════════════════════════════════════┤
    PKG: Signed
Installed ... Changes are applied after reboot.
example:/package/#> show
╒ Packages ═══════════════════════════════════════════════════════════════════╕
│NAME                        VERSION  PERSISTENT  VERIFICATION  DESCRIPTION  │
│app-hosting-license-signed  0.1      False       Signed        App-hosting  │
└──────────────────────────────────────────────────────────────────────────────┘

example:/package/#> leave
example:/#> reboot
```

Prepare the Device

Application Hosting is dependent on external media storage for storing application images and files associated with hosted application.

Applications have access to external media for storing their configuration files, logs, etc. The external media can be setup with specific partitions for different storage. The suggestion is to not mix the app-image storage with the shares.

First and foremost, external media device needs to be formatted to `vfat`, `ext2`, `ext3` or `ext4` filesystem type. This can be done using multiple tools in Linux, Windows, and even WeOS.

In Linux, a GUI tool called 'GParted' could be used.

In Windows, a build-in feature can be used:

1. Open File Explorer → Right-click the external media → Format.

2. File System: Select desired type.
3. Allocation unit size: Default.
4. Click Start.

To format and partition external media in WeOS, do the following:

```
example:/#> show media
MEDIA       DEVICE      LABEL          FORMAT   RW     SIZE  USED  DESCRIPTION
Active
internal    mtd:Config  -              jffs2    Yes   14.0M    6%  config,cert
Inactive
external    -           -              -        -         -     -
Available
-           mmcblk1     -              -        -     29.7G     -

example:/media/#> partition mmcblk1
Entering partitioning mode with dev mmcblk1 as target ...

No changes will be performed until the start-partition command is
invoked. When invoked the entire device mmcblk1 will be fully re-partitioned
and reformatted, any files on the device will be LOST/REMOVED, keep
this in mind before proceeding.

example:/media/partition-mmcblk1/#> add part-numb 1 label app_images size 10G format ext4
example:/media/partition-mmcblk1/#> show
Partition table for dev: mmcblk1
#  DEV-NAME      PART-NUMB  LABEL                        SIZE      FS
1  /dev/mmcblk1p1 1         app_images                   10G       ext4
example:/media/partition-mmcblk1/#> start-partitioning

Understand that the content on the device will be lost during the
partitioning and formatting of the device. Make sure that the device
does not contain any files that needs to be saved elsewhere.

Start partitioning and formatting of the device (y/N)? y

Attempt to partition device /dev/mmcblk1 in the following manner:
  1: /dev/mmcblk1p1 [Partition numb: 1], [Size: 10G]
Command (m for help): Partition type
   p   primary partition (1-4)
   e   extended
Command (m for help):
Initiate formatting of the partitions, this may take a while:
  1: /dev/mmcblk1p1 [Partition numb: 1] [Label: app_images], format with ext4 FS ...
Formatting: 100% \ [==================================================================]

Device mmcblk1 successfully partitioned and formatted!
example:/media/#> show
MEDIA       DEVICE      LABEL          FORMAT   RW     SIZE  USED  DESCRIPTION
Active
internal    mtd:Config  -              jffs2    Yes   14.0M    6%  config,cert
Inactive
external    -           -              -        -         -     -
Available
-           mmcblk1p1   app_images     ext4     -     10.0G     -

example:/media/#> exit
example:#>
```

After this, for the partitions to be mounted to named Medias the device boot settings needs to be modified. These changes are read at boot time.

```
example:/#>
example:/#> boot
example:/boot/#> no media external
example:/boot/#> media images
Creating new media definition ...
example:/boot/media-images/#> match label app_images
example:/boot/media-images/#> exit
example:/boot/#> exit
Saving bootloader configuration to FLASH
100% / [═══════════════════════════════════════════════════════════════════]
example:/#>
```

Restart the device and check that you have been provided with the correct medias and names.

```
example:/#> show media
MEDIA       DEVICE      LABEL          FORMAT   RW     SIZE  USED  DESCRIPTION
Active
internal    mtd:Config  -              jffs2    Yes   14.0M    6%  config,cert
images      mmcblk1p1   app_images     ext4     Yes    9.7G    0%
Inactive
Available
example:/#>
```

Now it is time to place application image onto the formatted drive. By default, WeOS expects to find application images in `/app/images` folder. Create this folder structure and copy images there.

Alternatively, any folder structure can be used, but make sure to change external storage settings in WeOS and set the correct path (see example below). The path set is relative to the root folder of the external media.

Enter the storage config context for application hosting and set the selected media to our formatted `images` media and, optionally, set a path within the external media for storage of application images. This example will continue using the default path `app/images`.

```
example:/#> config app-hosting
example:/config/app-hosting/#> storage
example:/config/app-hosting/storage/#> media images
example:/config/app-hosting/storage/#> path my/custom/folder/structure/
example:/config/app-hosting/storage/#> leave
```

Make sure you connect external media to your device before you proceed.

When the external drive is connected, WeOS will either look for images in the default catalogue (`app/images`) or any other configured folder.

Importing Application Images

Images can be imported in two ways:

1. Manual

   Place application image directly onto the external media. Note that by default, WeOS expects images to be placed in `app/images` folder on external media, unless configured otherwise.

2. Import from a remote server

   Place application image into the folder used by remote file server, head to `/file/app-hosting` context in WeOS and import the image using `import` command, as shown below:

```
example:/#> file app-hosting
example:/file/app/#> import proto tftp 192.0.2.1 /tftpboot/zeroimage-1.0.wmopkg
example:/file/app/#> show
/media/images/app/images/     MOD. DATE/LINK    SIZE      MD5
coronetimage-1.0.wmopkg       2025-11-27 22:35  2704786
zeroimage-1.0.wmopkg          2025-11-27 22:35  2872718

Free: 1013.9 MiB | Total: 1021.10 MiB
example:/file/app/#>
```

Imported application images will be placed into `app/images` folder on external media. As previously mentioned this path can be configured in application hosting context.

Label the Image

For images to be usable by applications, labels need to be set. Head to application hosting admin exec and configure labels:

```
example:/#> app-hosting
example:/app-hosting/#> label zeroimage-1.0 zero
example:/app-hosting/#> show

Applications
No applications found.

Application Images
NAME                                LABEL           VALID  VERIFIED  VERSION
zeroimage-1.0                       zero            Yes    Yes       1.0
coronetimage-1.0                    coronet         Yes    Yes       1.0

Image storage
Storage media    : images
Storage path     : app/images/
example:/app-hosting/#>
```

Configure Applications and System Resource Allocation

Several options can be configured for the overarching application hosting context, including storage which was configured previously:

- CPU & RAM - a percentage of the total CPU power and RAM capacity can be specified. All applications will share the allocated amount of CPU and RAM.

```
example:/#> config app-hosting
example:/config/app-hosting/#> cpu 25
example:/config/app-hosting/#> ram 25
example:/config/app-hosting/#> show

Allocated Hardware Resources
CPU(%)           RAM(%)
25               25

Image storage
Storage media    : images
Storage path     : app/images/
```

Now that application hosting is ready, create and configure hosting application:

```
example:/#> config app-hosting
example:/config/app-hosting/#> app app1
┌ Configuration Errors: 1 ═══════════════════════════════════════════════════
│#   ERROR  Description
│1   0860   Missing image label
└────────────────────────────────────────────────────────────────────────────

example:/config/app-hosting/app-app1/#> image zero
example:/config/app-hosting/app-app1/#> share port eth4
example:/config/app-hosting/app-app1/#> share veth lan0 as wan0
example:/config/app-hosting/app-app1/#> capability CAP_NET_ADMIN
example:/config/app-hosting/app-app1/#> show
Name           : app1
Status         : Enabled
Autostart      : Yes
Description    :
Image label    : zero
Command        : /sbin/init
Shell          : sh
Net. Namespace : user-ns
UID            : 0
GID            : 0

Shared Resources
TYPE  HOST                              GUEST               OPTS
port  eth4                              eth4
veth  lan0                              wan0

Environment Variables
NAME                    VALUE

Capabilities
NAME
CAP_NET_ADMIN
example:/config/app-hosting/app-app1/#> leave
```

For application to run successfully, an application image must be specified. Applications may utilise system resources, such as physical ports, and create virtual ethernet pairs (vEth). In the case above, app1 takes ownership of port eth4. Essentially, this means that eth4 will be moved to app1's network namespace, and WeOS will lose access to this port. app1 also creates a vEth, in the example above. The nature of vEth is that it has two ends: in our case, wan0 is the name of the interface seen and owned by app1, while lan0 is the other end, seen and owned by WeOS. The diagram below shows relationships of ports and vEth.

We can see that eth4 now belongs to app1 and resides in its private network namespace. wan0 belongs to app1, while lan0 belongs to WeOS. Additionally, lan0 interface can be added to a VLAN available in WeOS, in this case VLAN v1. This is not a necessity, but a convenience. lan0 interface can simply stay as a standalone interface in WeOS.

```
+-------------------------------------------------------------------------+
| WeOS                                        +-----------------+         |
|                                             |       app1      |         |
|                                             +---------+-------+         |
|                                                       |                 |
|     +-------------------------------+       +---------+-------+         |
|     |                               |       |                 |         |
|     |       Host-OS netns           |       |    app1-netns   |         |
|     |                               |       |                 |         |
```

```
|     +--+-------------------+--------+        +-+--------+------+         |
|        |                   |                  |        |                |
|        |                   |                  |        |                |
|        |                +--+-+                |        |                |
|        |         +------+ v1 +-------+        |        |                |
|        |         |      +--+-+       |        |        |                |
|        |         |         |         |        |        |                |
|        |         |         |         |        |        |                |
|     +--+-+    +--+-+    +--+-+    +--+-+    +--+-+   +--+-+             |
|     |    |    |    |    |    |    |    +----+    |   |    |             |
|     +----+    +----+    +----+    +----+    +----+   +----+             |
|      Eth1      Eth2      Eth3      Lan0      Wan0     Eth4               |
+-----------------------------------------------------------------------+
```

Control Applications and Monitor Status

Applications can be controlled and monitored from the application hosting admin exec context.

To see the status of applications, run the following command:

```
example:/#> app-hosting
example:/app-hosting/#> show

Applications
NAME             STATUS           NET.NAMESPACE    IMG.LABEL
app1             Running          app1             zero

Application Images
NAME                                LABEL            VALID  VERIFIED  VERSION
zeroimage-1.0                       zero             Yes    Yes       1.0
coronetimage-1.0                    coronet          Yes    Yes       1.0

Image storage
Storage media    : images
Storage path     : app/images
example:/app-hosting/#>
```

Use the following commands for starting, stopping restarting or attaching to the application:

```
example:/app-hosting/#> stop app1
example:/app-hosting/#> start app1
example:/app-hosting/#> restart app1
example:/app-hosting/#> attach app1
/ # exit
example:/app-hosting/#>
```

If application image label ever needs to be changed, use the following commands:

```
example:/app-hosting/#> label zeroimage-1.0 ze
example:/app-hosting/#> reload ze
example:/app-hosting/#>
```

When image label is changed, applications using the old label need to be restarted for the new changes to take place. `reload` command does that.

To monitor application logs, use the following commands:

```
example:/app-hosting/#> show log app1
example:/app-hosting/#> follow log app1
```

4.14.2. Use Case: Firewall

Introduction

This document shows an example use case of how to use application hosting to create an advanced router and firewall using nftables.

```
WeOS                                # Firewall Application
     .-------.  .-------.            # .-------.  .-------.
     | vlan1 |  | vlan2 |            # | vlan1 |  | vlan2 |
     '---+---'  '---+---'            # '---+---'  '---+---'
         '-.      .-'                #     '---.  .---'
       .---+------+---.              #       .-+--+-.
       |      br0     +-------.      #       | eth0 |
       '-+---+--+-----'       |      #       '--+---'
   .-----' .-'  '-.           |      #          |
.--+-.  .--+-.  .-+--.    .---+---.  #          |             .----.
|eth1|  |eth2|  |eth3|    |  fw0  |  #          |             |eth4|
'----'  '----'  '----'    '---+---'  #          |             '----'
                              '-----------------'
```

Our application will take exclusive control of the `eth4` port, which is connected to our ISP, i.e. it is our WAN interface. We will then connect one end of a vEth pair (`fw0`) to the host system bridge (`br0`) and the other end (`eth0`) will be placed in the application's network namespace.

We will then set up the application to configure the necessary network interfaces and load our firewall configuration, when the application starts.

Prerequisites

To complete this howto you will need:

- An application image with `iproute2`, `iputils` and `nftables` installed.

Also, before we can configure our application *instance*, there are a few things that need to be prepared first. Follow the 4 steps below in the links and then continue this guide.

1. Enable Application Hosting.
2. Prepare an external media (USB/SD-Card) for *persistent storage* of the the application *image*. This is described here in Prepare the Device.
3. Import your image (remember that your image needs to have `iproute2`, `iputils` and `nftables` installed).
4. Label your image.

Given that you have now enabled application hosting, prepared an external media, imported your application image and labeled it we can proceed with configuring our application.

Application Configuration

Our application needs three files backed by persistent storage:

- `app-interfaces`: For BusyBox's version of `ifupdown` to configure our local VLAN interfaces
- `app-sysctl.conf`: For enabling IP forwarding on startup
- `app-fw.nft`: Configuration for nftables[1]

Create the files in the root folder of the external media with the following content:

app-interfaces

[1] https://nftables.org/projects/nftables/index.html

```
auto lo
iface lo inet loopback

auto vlan1
iface vlan1 inet static
  address 192.168.1.1
  netmask 255.255.255.0
  pre-up ip link set dev eth0 up
  pre-up ip link add dev vlan1 link eth0 type vlan id 1

auto vlan2
iface vlan2 inet static
  address 192.168.2.1
  netmask 255.255.255.0
  pre-up ip link set dev eth0 up
  pre-up ip link add dev vlan2 link eth0 type vlan id 2

auto wan
iface wan inet dhcp
pre-up nft -f /etc/fw.nft
```

app-sysctl.conf

```
net.ipv4.ip_forward=1
```

app-fw.nft

```
flush ruleset

table filter {
  chain input {
    type filter hook input priority 0; policy drop;

    iif lo accept
    meta l4proto icmp accept
    goto common
  }

  chain forward {
    type filter hook forward priority 0; policy drop;

    iif vlan1 accept
    iif vlan2 accept
    goto common
  }

  chain common {
    ct state invalid drop
    # Always allow the return traffic related to flows that were
    # initiated by a local host.
    ct state established,related accept
  }

  chain output {
    type filter hook output priority 0; policy accept;
  }

  chain postrouting {
    type nat hook postrouting priority 0;
    # Hide local addresses for traffic going out over the Internet.
    oifname wan masquerade
  }
}
```

The `nftables` configuration above is an example of a very basic masquerading firewall that allows all outgoing connections and the generated return traffic. Edit the configuration to suit your preferences.

WeOS Configuration

First, here is the output of `show media` to display the name used for our external media: `images`. Also, the output of `show app` to display the name used for our label, `mylabel`, that was set during the steps of the Prerequisites section.

```
example:/#> show media
MEDIA       DEVICE      LABEL          FORMAT    RW     SIZE  USED  DESCRIPTION
Active
internal    mtd:Config  -              jffs2     Yes   14.0M    5%  config,cert
images      mmcblk1p1   app_images     ext4      Yes    9.7G    0%
Inactive
Available
```

```
example:/#> show app

Applications
No applications found.

Application Images
NAME                                LABEL           VALID  VERIFIED  VERSION
myimage-1.0                         mylabel         Yes    Yes       1.0

Image storage
Storage media     : images
Storage path      : app/images/
```

If you have chosen other names for your external media and image label, make sure to use those from here on.

- Prepared the Device
- Imported your image
- Labeled your image

Now, we are ready to configure our application instance and connect it to the WeOS networking stack:

```
example:/#> config app-hosting
example:/config/app-hosting/#> app fw
┌ Configuration Errors: 1 ═══════════════════════════════════
│ #   ERROR  Description
│ 1   0865   Missing image label
└────────────────────────────────────────────────────────────

example:/config/app-hosting/app-fw/#> image mylabel
example:/config/app-hosting/app-fw/#> apparmor relaxed
example:/config/app-hosting/app-fw/#> seccomp relaxed
example:/config/app-hosting/app-fw/#> share veth fw0 as eth0 none
example:/config/app-hosting/app-fw/#> share port eth4 as wan none
example:/config/app-hosting/app-fw/#> share media images app-fw.nft as /etc/fw.nft
example:/config/app-hosting/app-fw/#> share media images app-sysctl.conf as /etc/sysctl.conf
example:/config/app-hosting/app-fw/#> share media images app-interfaces as /etc/network/interfaces
example:/config/app-hosting/app-fw/#> capability cap_net_admin
example:/config/app-hosting/app-fw/#> capability cap_net_raw
example:/config/app-hosting/app-fw/#> show
Name            : fw
Status          : Enabled
Autostart       : Yes
Description     :
Image label     : mylabel
Command         : /sbin/init
Shell           : sh
```

```
Net. Namespace : user-ns
AppArmor       : relaxed
Seccomp        : relaxed
UID            : 0
GID            : 0

Shared Resources
TYPE  HOST                              GUEST              OPTS
media images/app-fw.nft                 /etc/fw.nft
media images/app-sysctl                 /etc/sysctl.conf
media images/app-interfaces             /etc/network/interfaces
port  eth4                              wan
veth  fw0                               eth0

Environment Variables
NAME                    VALUE

Capabilities
NAME
CAP_NET_ADMIN
CAP_NET_RAW
example:/config/app-hosting/app-fw/#> end
example:/config/app-hosting/#> end
```

> **NOTE**
>
> This example uses the `relaxed` AppArmor and seccomp profiles because the application configures network interfaces, creates VLANs, and enables IP forwarding during the applications startup. The strict AppArmor profile does not permit the `CAP_NET_ADMIN` capability required for these network operations.

Now we set up our VLANs, ensuring that vEth port `fw0` is a tagged member of both VLANs:

```
example:/config/#> vlan 1
example:/config/vlan-1/#> untagged eth1
example:/config/vlan-1/#> tagged eth3,fw0
example:/config/vlan-1/#> end
example:/config/#> vlan 2
example:/config/vlan-2/#> untagged eth2
example:/config/vlan-2/#> tagged eth3,fw0
example:/config/vlan-2/#> end
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> inet static 192.168.1.2/24
example:/config/iface-vlan1/inet-static-192.168.1.2/#> end
example:/config/iface-vlan1/#> end
example:/config/#> iface vlan2
example:/config/iface-vlan2/#> inet static 192.168.2.2/24
example:/config/iface-vlan2/inet-static-192.168.2.2/#> end
example:/config/iface-vlan2/#> end
```

Finally, we will configure WeOS to act as a DHCP server on VLAN 1 and 2, making sure to advertise our application as being the default gateway; also add a default route to the application for the local device:

```
example:/config/#> dhcp-server
example:/config/dhcp-server/#> subnet 192.168.1.0/24
example:/config/dhcp-server/subnet-192.168.1.0/#> gateway 192.168.1.1
example:/config/dhcp-server/subnet-192.168.1.0/#> name-server 8.8.8.8
```

```
example:/config/dhcp-server/subnet-192.168.1.0/#> end
example:/config/dhcp-server/#> subnet 192.168.2.0/24
example:/config/dhcp-server/subnet-192.168.2.0/#> gateway 192.168.2.1
example:/config/dhcp-server/subnet-192.168.2.0/#> name-server 8.8.8.8
example:/config/dhcp-server/subnet-192.168.2.0/#> end
example:/config/dhcp-server/#> end
example:/config/#> ip
example:/config/ip/#> route default 192.168.1.1
example:/config/ip/#> route default 192.168.2.1
example:/config/ip/#> leave
example:/#> copy running-config startup-config
```

Connectivity Test

Now we are ready to attach to our application verify connectivity:

```
example:/#> app attach fw
#
```

We should be able to reach WeOS on both VLANs as well as external resources on the Internet:

```
# ifup -a
ifup: interface lo already configured
# ip -br addr
lo               UNKNOWN        127.0.0.1/8 ::1/128
gre0@NONE        DOWN
gretap0@NONE     DOWN
erspan0@NONE     DOWN
ip_vti0@NONE     DOWN
ip6tnl0@NONE     DOWN
vlan1@eth0       UP             192.168.1.1/24 fe80::144a:32ff:fe1f:6eb/64
vlan2@eth0       UP             192.168.2.1/24 fe80::144a:32ff:fe1f:6eb/64
wan@gretap0      UP             198.18.221.120/24 fe80::207:7cff:fe65:f844/64
eth0@if21        UP             fe80::144a:32ff:fe1f:6eb/64
# ping -c1 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: seq=0 ttl=64 time=0.181 ms

--- 192.168.1.2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.181/0.181/0.181 ms
# ping -c1 192.168.2.2
PING 192.168.2.2 (192.168.2.2): 56 data bytes
64 bytes from 192.168.2.2: seq=0 ttl=64 time=0.266 ms

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.266/0.266/0.266 ms
# ping -c1 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=52 time=4.923 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 4.923/4.923/4.923 ms
#
```

We can also verify that we cannot reach vlan1 or vlan2 on the WeOS side by connecting a host pc to `eth4` (`wan` on the application side). But first we need to let that pc know how to reach that vlan:

```
###
### From the host pc
###
user@hostpc $ sudo ip route add 192.168.1.0/24 via <ip of wan interface>
user@hostpc $ sudo ip route add 192.168.2.0/24 via <ip of wan interface>

###
### Try to ping the application side of vlan1 and vlan2 (should succeed)
###
user@hostpc ping -c1 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.319 ms

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.319/0.319/0.319/0.000 ms

user@hostpc $ ping -c1 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.306 ms

--- 192.168.2.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.306/0.306/0.306/0.000 ms

###
### Try to ping the WeOS side of vlan1 and vlan2 (should fail)
###
user@hostpc $ ping -c1 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.

--- 192.168.1.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

user@hostpc $ ping -c1 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

5. Appendix

5.1. Application Hosting

5.1.1. AppArmor Profiles

The system comes with two AppArmor profiles, one for strict mode and one for relaxed mode. The strict profile is used by default, and the relaxed profile can be used by applications that require additional permissions. The following profiles are included with the system.

Strict

```
#include <tunables/global>

profile weos-app-apparmor-strict flags=(attach_disconnected,mediate_deleted) {
  signal (send,receive) peer=weos-app-apparmor-strict,

network inet stream,
network inet dgram,
network inet6 stream,
network inet6 dgram,
network unix stream,
network unix dgram,

audit deny network raw,
audit deny network packet,
audit deny network alg,

capability chown,
capability dac_override,
capability kill,
capability setgid,
capability setuid,
capability net_bind_service,

/ r,
/** r,

/{usr/,}{bin,sbin}/** ix,
/lib/** ix,
/lib64/** ix,
/opt/**/bin/** ix,

/tmp/** rwl,
/var/tmp/** rwl,
/run/** rwkl,
/dev/shm/** rwl,
/app/data/** rwl,
/opt/app/data/** rwl,
/dev/null rw,
audit deny /dev/zero w,
/dev/zero r,
/dev/urandom r,
/dev/random r,
/dev/full rw,
```

continues on next page

continued

Strict

```
/dev/tty rw,
/dev/pts/** rw,
/mnt/** rw,
/root/** rw,

audit deny /tmp/** x,
audit deny /var/tmp/** x,
audit deny /run/** x,
audit deny /dev/shm/** x,
audit deny /app/data/** x,
audit deny /opt/app/data/** x,

signal (receive) peer=unconfined,
signal (receive) set=(term,int,hup,chld) peer=runc,
signal (receive) set=(term,int,hup,chld) peer=crun,

audit deny @{PROC}/* w,
audit deny @{PROC}/{[^1-9],[^1-9][^0-9],[^1-9s][^0-9y][^0-9s],[^1-9][^0-9][^0-9]
[^0-9]*}/** w,
audit deny @{PROC}/sys/** rw,
audit deny @{PROC}/sysrq-trigger rwklx,
audit deny @{PROC}/kcore rwklx,
audit deny @{PROC}/mem rwklx,
audit deny @{PROC}/kmem rwklx,

audit deny mount,
audit deny umount,

audit deny /sys/** rwklx,

audit deny ptrace,
}
```

Relaxed

```
#include <tunables/global>

profile weos-app-apparmor-relaxed flags=(attach_disconnected,mediate_deleted) {
  signal (send,receive) peer=weos-app-apparmor-relaxed,
  ptrace (trace,read) peer=weos-app-apparmor-relaxed,

network,

capability,

file,
umount,

signal (receive) peer=unconfined,

# Override the generic /proc pseudo-directory write deny below.
priority=1 @{PROC}/sys/kernel/shm** w,
```

continues on next page

continued

Relaxed

```
audit deny @{PROC}/* w,
audit deny @{PROC}/{[^1-9],[^1-9][^0-9],[^1-9s][^0-9y][^0-9s],[^1-9][^0-9][^0-9]
[^0-9]*}/** w,
audit deny @{PROC}/sysrq-trigger rwklx,
audit deny @{PROC}/kcore rwklx,

audit deny mount,

audit deny /sys/[^f]*/** wklx,
audit deny /sys/f[^s]*/** wklx,
audit deny /sys/fs/[^c]*/** wklx,
audit deny /sys/fs/c[^g]*/** wklx,
audit deny /sys/fs/cg[^r]*/** wklx,
audit deny /sys/firmware/** rwklx,
audit deny /sys/devices/virtual/powercap/** rwklx,
audit deny /sys/kernel/security/** rwklx,
}
```

5.1.2. Seccomp Profiles

The system comes with two seccomp profiles, one for strict mode and one for relaxed mode. The strict profile is used by default, and the relaxed profile can be used by applications that require additional system calls. The following profiles are included with the system.

Strict

```
{
  "defaultAction": "SCMP_ACT_ERRNO",
  "defaultErrnoRet": 38,
  "flags": [
    "SECCOMP_FILTER_FLAG_LOG"
  ],
  "architectures": [],
  "syscalls": [
    {
      "names": [
        "accept",
        "accept4",
        "access",
        "alarm",
        "bind",
        "brk",
        "capget",
        "chdir",
        "chmod",
        "chown",
        "chown32",
        "clock_getres",
        "clock_getres_time64",
        "clock_gettime",
        "clock_gettime64",
        "clock_nanosleep",
        "clock_nanosleep_time64",
        "close",
        "close_range",
        "connect",
        "copy_file_range",
        "creat",
        "dup",
        "dup2",
        "dup3",
        "epoll_create",
        "epoll_create1",
        "epoll_ctl",
        "epoll_pwait",
        "epoll_pwait2",
        "epoll_wait",
        "eventfd",
        "eventfd2",
        "execve",
        "execveat",
        "exit",
        "exit_group",
        "faccessat",
```

continues on next page

continued

Strict

```
        "faccessat2",
        "fadvise64",
        "fallocate",
        "fchdir",
        "fchmod",
        "fchmodat",
        "fchmodat2",
        "fchown",
        "fchown32",
        "fchownat",
        "fcntl",
        "fcntl64",
        "fdatasync",
        "fgetxattr",
        "flistxattr",
        "flock",
        "fork",
        "fremovexattr",
        "fsetxattr",
        "fstat",
        "fstat64",
        "fstatat64",
        "fstatfs",
        "fstatfs64",
        "fsync",
        "ftruncate",
        "ftruncate64",
        "futex",
        "futex_requeue",
        "futex_time64",
        "futex_wait",
        "futex_waitv",
        "futex_wake",
        "futimesat",
        "getcpu",
        "getcwd",
        "getdents",
        "getdents64",
        "getegid",
        "getegid32",
        "geteuid",
        "geteuid32",
        "getgid",
        "getgid32",
        "getgroups",
        "getgroups32",
        "getitimer",
        "getpeername",
        "getpgid",
        "getpgrp",
        "getpid",
        "getppid",
        "getpriority",
```

continues on next page

continued

Strict

```
        "getrandom",
        "getresgid",
        "getresgid32",
        "getresuid",
        "getresuid32",
        "getrlimit",
        "get_robust_list",
        "getrusage",
        "getsid",
        "getsockname",
        "getsockopt",
        "get_thread_area",
        "gettid",
        "gettimeofday",
        "getuid",
        "getuid32",
        "getxattr",
        "inotify_add_watch",
        "inotify_init",
        "inotify_init1",
        "inotify_rm_watch",
        "io_cancel",
        "ioctl",
        "io_destroy",
        "io_getevents",
        "io_pgetevents",
        "io_pgetevents_time64",
        "ioprio_get",
        "ioprio_set",
        "io_setup",
        "io_submit",
        "ipc",
        "kill",
        "landlock_add_rule",
        "landlock_create_ruleset",
        "landlock_restrict_self",
        "lchown",
        "lchown32",
        "lgetxattr",
        "link",
        "linkat",
        "listen",
        "listxattr",
        "llistxattr",
        "_llseek",
        "lremovexattr",
        "lseek",
        "lsetxattr",
        "lstat",
        "lstat64",
        "madvise",
        "membarrier",
        "memfd_create",
```

continues on next page

continued

Strict

```
        "mincore",
        "mkdir",
        "mkdirat",
        "mknod",
        "mknodat",
        "mlock",
        "mlock2",
        "mlockall",
        "mmap",
        "mmap2",
        "mprotect",
        "mseal",
        "mq_getsetattr",
        "mq_notify",
        "mq_open",
        "mq_timedreceive",
        "mq_timedreceive_time64",
        "mq_timedsend",
        "mq_timedsend_time64",
        "mq_unlink",
        "mremap",
        "msgctl",
        "msgget",
        "msgrcv",
        "msgsnd",
        "msync",
        "munlock",
        "munlockall",
        "munmap",
        "nanosleep",
        "newfstatat",
        "_newselect",
        "open",
        "openat",
        "openat2",
        "pause",
        "pidfd_getfd",
        "pidfd_open",
        "pidfd_send_signal",
        "pipe",
        "pipe2",
        "pkey_alloc",
        "pkey_free",
        "pkey_mprotect",
        "poll",
        "ppoll",
        "ppoll_time64",
        "prctl",
        "pread64",
        "preadv",
        "preadv2",
        "prlimit64",
        "process_mrelease",
```

continues on next page

continued

Strict

```
        "pselect6",
        "pselect6_time64",
        "pwrite64",
        "pwritev",
        "pwritev2",
        "read",
        "readahead",
        "readlink",
        "readlinkat",
        "readv",
        "recv",
        "recvfrom",
        "recvmmsg",
        "recvmmsg_time64",
        "recvmsg",
        "remap_file_pages",
        "removexattr",
        "rename",
        "renameat",
        "renameat2",
        "restart_syscall",
        "rmdir",
        "rseq",
        "rt_sigaction",
        "rt_sigpending",
        "rt_sigprocmask",
        "rt_sigqueueinfo",
        "rt_sigreturn",
        "rt_sigsuspend",
        "rt_sigtimedwait",
        "rt_sigtimedwait_time64",
        "rt_tgsigqueueinfo",
        "sched_getaffinity",
        "sched_getattr",
        "sched_getparam",
        "sched_get_priority_max",
        "sched_get_priority_min",
        "sched_getscheduler",
        "sched_rr_get_interval",
        "sched_rr_get_interval_time64",
        "sched_setaffinity",
        "sched_setattr",
        "sched_setparam",
        "sched_setscheduler",
        "sched_yield",
        "seccomp",
        "select",
        "semctl",
        "semget",
        "semop",
        "semtimedop",
        "semtimedop_time64",
        "send",
```

continues on next page

continued

Strict

```
        "sendfile",
        "sendfile64",
        "sendmmsg",
        "sendmsg",
        "sendto",
        "setfsgid",
        "setfsgid32",
        "setfsuid",
        "setfsuid32",
        "setgid",
        "setgid32",
        "setgroups",
        "setgroups32",
        "setitimer",
        "setpgid",
        "setpriority",
        "setregid",
        "setregid32",
        "setresgid",
        "setresgid32",
        "setresuid",
        "setresuid32",
        "setreuid",
        "setreuid32",
        "setrlimit",
        "set_robust_list",
        "setsid",
        "setsockopt",
        "set_thread_area",
        "set_tid_address",
        "setuid",
        "setuid32",
        "setxattr",
        "shmat",
        "shmctl",
        "shmdt",
        "shmget",
        "shutdown",
        "sigaltstack",
        "signalfd",
        "signalfd4",
        "sigprocmask",
        "sigreturn",
        "socketpair",
        "splice",
        "stat",
        "stat64",
        "statfs",
        "statfs64",
        "statmount",
        "statx",
        "symlink",
        "symlinkat",
```

continues on next page

continued

Strict

```
        "sync",
        "sync_file_range",
        "syncfs",
        "sysinfo",
        "tee",
        "tgkill",
        "time",
        "timer_create",
        "timer_delete",
        "timer_getoverrun",
        "timer_gettime",
        "timer_gettime64",
        "timer_settime",
        "timer_settime64",
        "timerfd_create",
        "timerfd_gettime",
        "timerfd_gettime64",
        "timerfd_settime",
        "timerfd_settime64",
        "times",
        "tkill",
        "truncate",
        "truncate64",
        "ugetrlimit",
        "umask",
        "uname",
        "unlink",
        "unlinkat",
        "utime",
        "utimensat",
        "utimensat_time64",
        "utimes",
        "vfork",
        "wait4",
        "waitid",
        "waitpid",
        "write",
        "writev"
      ],
      "action": "SCMP_ACT_ALLOW"
    },
    {
      "names": [
        "capset"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SETPCAP",
      "includes": {
        "caps": [
          "CAP_SETPCAP"
        ]
      }
    },
```

continues on next page

continued

Strict

```
    {
      "names": [
        "socket"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 2,
          "op": "SCMP_CMP_EQ"
        }
      ],
      "comment": "AF_INET only"
    },
    {
      "names": [
        "socket"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 10,
          "op": "SCMP_CMP_EQ"
        }
      ],
      "comment": "AF_INET6 only"
    },
    {
      "names": [
        "socket"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 1,
          "op": "SCMP_CMP_EQ"
        }
      ],
      "comment": "AF_UNIX only"
    },
    {
      "names": [
        "socket"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 16,
          "op": "SCMP_CMP_EQ"
        }
```

continues on next page

continued

Strict

```
      ],
      "comment": "AF_NETLINK only"
    },
    {
      "names": [
        "clone"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 2114060288,
          "valueTwo": 0,
          "op": "SCMP_CMP_MASKED_EQ"
        }
      ],
      "comment": "Allow clone only when CLONE_NEW* namespace flags (0x7E020000)
are all clear"
    },
    {
      "names": [
        "clone3"
      ],
      "action": "SCMP_ACT_ERRNO",
      "errnoRet": 38,
      "comment": "Force fallback to clone(2) so the CLONE_NEW* arg filter
applies"
    },
    {
      "names": [
        "personality"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 0,
          "op": "SCMP_CMP_EQ"
        }
      ]
    },
    {
      "names": [
        "personality"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 4294967295,
          "op": "SCMP_CMP_EQ"
        }
      ]
```

continues on next page

continued

Strict

```
    },
    {
      "names": [
        "arch_prctl"
      ],
      "action": "SCMP_ACT_ALLOW",
      "includes": {
        "arches": [
          "amd64"
        ]
      }
    },
    {
      "names": [
        "arm_fadvise64_64",
        "arm_sync_file_range",
        "breakpoint",
        "cacheflush",
        "set_tls"
      ],
      "action": "SCMP_ACT_ALLOW",
      "includes": {
        "arches": [
          "arm",
          "arm64"
        ]
      }
    },
    {
      "names": [
        "switch_endian"
      ],
      "action": "SCMP_ACT_ALLOW",
      "includes": {
        "arches": [
          "ppc64"
        ]
      }
    }
  ]
}
```

Relaxed

```
{
  "defaultAction": "SCMP_ACT_ERRNO",
  "defaultErrnoRet": 38,
  "flags": [
    "SECCOMP_FILTER_FLAG_LOG"
  ],
  "architectures": [],
  "syscalls": [
```

continues on next page

continued

Relaxed

```
{
  "names": [
    "accept",
    "accept4",
    "access",
    "add_key",
    "alarm",
    "bind",
    "brk",
    "cachestat",
    "capget",
    "chdir",
    "chmod",
    "chown",
    "chown32",
    "clock_getres",
    "clock_getres_time64",
    "clock_gettime",
    "clock_gettime64",
    "clock_nanosleep",
    "clock_nanosleep_time64",
    "clone",
    "clone3",
    "close",
    "close_range",
    "connect",
    "copy_file_range",
    "creat",
    "dup",
    "dup2",
    "dup3",
    "epoll_create",
    "epoll_create1",
    "epoll_ctl",
    "epoll_pwait",
    "epoll_pwait2",
    "epoll_wait",
    "eventfd",
    "eventfd2",
    "execve",
    "execveat",
    "exit",
    "exit_group",
    "faccessat",
    "faccessat2",
    "fadvise64",
    "fallocate",
    "fanotify_init",
    "fanotify_mark",
    "fchdir",
    "fchmod",
    "fchmodat",
    "fchmodat2",
```

continues on next page

continued

Relaxed

```
        "fchown",
        "fchown32",
        "fchownat",
        "fcntl",
        "fcntl64",
        "fdatasync",
        "fgetxattr",
        "flistxattr",
        "flock",
        "fork",
        "fremovexattr",
        "fsetxattr",
        "fstat",
        "fstat64",
        "fstatat64",
        "fstatfs",
        "fstatfs64",
        "fsync",
        "ftruncate",
        "ftruncate64",
        "futex",
        "futex_requeue",
        "futex_time64",
        "futex_wait",
        "futex_waitv",
        "futex_wake",
        "futimesat",
        "getcpu",
        "getcwd",
        "getdents",
        "getdents64",
        "getegid",
        "getegid32",
        "geteuid",
        "geteuid32",
        "getgid",
        "getgid32",
        "getgroups",
        "getgroups32",
        "getitimer",
        "getpeername",
        "getpgid",
        "getpgrp",
        "getpid",
        "getppid",
        "getpriority",
        "getrandom",
        "getresgid",
        "getresgid32",
        "getresuid",
        "getresuid32",
        "getrlimit",
        "get_robust_list",
```

continues on next page

continued

Relaxed

```
        "getrusage",
        "getsid",
        "getsockname",
        "getsockopt",
        "get_thread_area",
        "gettid",
        "gettimeofday",
        "getuid",
        "getuid32",
        "getxattr",
        "inotify_add_watch",
        "inotify_init",
        "inotify_init1",
        "inotify_rm_watch",
        "io_cancel",
        "ioctl",
        "io_destroy",
        "io_getevents",
        "io_pgetevents",
        "io_pgetevents_time64",
        "ioprio_get",
        "ioprio_set",
        "io_setup",
        "io_submit",
        "ipc",
        "keyctl",
        "kill",
        "landlock_add_rule",
        "landlock_create_ruleset",
        "landlock_restrict_self",
        "lchown",
        "lchown32",
        "lgetxattr",
        "link",
        "linkat",
        "listen",
        "listxattr",
        "llistxattr",
        "_llseek",
        "lremovexattr",
        "lseek",
        "lsetxattr",
        "lstat",
        "lstat64",
        "listmount",
        "madvise",
        "map_shadow_stack",
        "membarrier",
        "memfd_create",
        "memfd_secret",
        "mincore",
        "mkdir",
        "mkdirat",
```

continues on next page

continued

Relaxed

```
        "mknod",
        "mknodat",
        "mlock",
        "mlock2",
        "mlockall",
        "mmap",
        "mmap2",
        "mprotect",
        "mseal",
        "mq_getsetattr",
        "mq_notify",
        "mq_open",
        "mq_timedreceive",
        "mq_timedreceive_time64",
        "mq_timedsend",
        "mq_timedsend_time64",
        "mq_unlink",
        "mremap",
        "msgctl",
        "msgget",
        "msgrcv",
        "msgsnd",
        "msync",
        "munlock",
        "munlockall",
        "munmap",
        "name_to_handle_at",
        "nanosleep",
        "newfstatat",
        "_newselect",
        "nice",
        "open",
        "openat",
        "openat2",
        "pause",
        "pidfd_getfd",
        "pidfd_open",
        "pidfd_send_signal",
        "pipe",
        "pipe2",
        "pkey_alloc",
        "pkey_free",
        "pkey_mprotect",
        "poll",
        "ppoll",
        "ppoll_time64",
        "prctl",
        "pread64",
        "preadv",
        "preadv2",
        "prlimit64",
        "process_mrelease",
        "pselect6",
```

continues on next page

continued

Relaxed

```
        "pselect6_time64",
        "pwrite64",
        "pwritev",
        "pwritev2",
        "read",
        "readahead",
        "readlink",
        "readlinkat",
        "readv",
        "recv",
        "recvfrom",
        "recvmmsg",
        "recvmmsg_time64",
        "recvmsg",
        "remap_file_pages",
        "removexattr",
        "rename",
        "renameat",
        "renameat2",
        "request_key",
        "restart_syscall",
        "rmdir",
        "rseq",
        "rt_sigaction",
        "rt_sigpending",
        "rt_sigprocmask",
        "rt_sigqueueinfo",
        "rt_sigreturn",
        "rt_sigsuspend",
        "rt_sigtimedwait",
        "rt_sigtimedwait_time64",
        "rt_tgsigqueueinfo",
        "sched_getaffinity",
        "sched_getattr",
        "sched_getparam",
        "sched_get_priority_max",
        "sched_get_priority_min",
        "sched_getscheduler",
        "sched_rr_get_interval",
        "sched_rr_get_interval_time64",
        "sched_setaffinity",
        "sched_setattr",
        "sched_setparam",
        "sched_setscheduler",
        "sched_yield",
        "seccomp",
        "select",
        "semctl",
        "semget",
        "semop",
        "semtimedop",
        "semtimedop_time64",
        "send",
```

continues on next page

continued

Relaxed

```
        "sendfile",
        "sendfile64",
        "sendmmsg",
        "sendmsg",
        "sendto",
        "setfsgid",
        "setfsgid32",
        "setfsuid",
        "setfsuid32",
        "setgid",
        "setgid32",
        "setgroups",
        "setgroups32",
        "setitimer",
        "setpgid",
        "setpriority",
        "setregid",
        "setregid32",
        "setresgid",
        "setresgid32",
        "setresuid",
        "setresuid32",
        "setreuid",
        "setreuid32",
        "setrlimit",
        "set_robust_list",
        "setsid",
        "setsockopt",
        "set_thread_area",
        "set_tid_address",
        "setuid",
        "setuid32",
        "setxattr",
        "shmat",
        "shmctl",
        "shmdt",
        "shmget",
        "shutdown",
        "sigaltstack",
        "signalfd",
        "signalfd4",
        "sigprocmask",
        "sigreturn",
        "socketpair",
        "splice",
        "stat",
        "stat64",
        "statfs",
        "statfs64",
        "statmount",
        "statx",
        "symlink",
        "symlinkat",
```

continues on next page

continued

Relaxed

```
        "sync",
        "sync_file_range",
        "syncfs",
        "sysinfo",
        "tee",
        "tgkill",
        "time",
        "timer_create",
        "timer_delete",
        "timer_getoverrun",
        "timer_gettime",
        "timer_gettime64",
        "timer_settime",
        "timer_settime64",
        "timerfd_create",
        "timerfd_gettime",
        "timerfd_gettime64",
        "timerfd_settime",
        "timerfd_settime64",
        "times",
        "tkill",
        "truncate",
        "truncate64",
        "ugetrlimit",
        "umask",
        "uname",
        "unshare",
        "unlink",
        "unlinkat",
        "utime",
        "utimensat",
        "utimensat_time64",
        "utimes",
        "vfork",
        "wait4",
        "waitid",
        "waitpid",
        "write",
        "writev"
      ],
      "action": "SCMP_ACT_ALLOW"
    },
    {
      "names": [
        "capset"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SETPCAP",
      "includes": {
        "caps": [
          "CAP_SETPCAP"
        ]
      }
```

continues on next page

continued

Relaxed

```
    },
    {
      "names": [
        "adjtimex",
        "clock_adjtime",
        "clock_adjtime64"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_TIME",
      "includes": {
        "caps": [
          "CAP_SYS_TIME"
        ]
      }
    },
    {
      "names": [
        "setns"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_ADMIN",
      "includes": {
        "caps": [
          "CAP_SYS_ADMIN"
        ]
      }
    },
    {
      "names": [
        "umount",
        "umount2"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_ADMIN",
      "includes": {
        "caps": [
          "CAP_SYS_ADMIN"
        ]
      }
    },
    {
      "names": [
        "vmsplice"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_ADMIN",
      "includes": {
        "caps": [
          "CAP_SYS_ADMIN"
        ]
      }
    },
    {
```

continues on next page

continued

Relaxed

```
      "names": [
        "ptrace",
        "process_vm_readv",
        "process_vm_writev"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_PTRACE",
      "includes": {
        "caps": [
          "CAP_SYS_PTRACE"
        ]
      }
    },
    {
      "names": [
        "bpf"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_ADMIN",
      "includes": {
        "caps": [
          "CAP_SYS_ADMIN"
        ]
      }
    },
    {
      "names": [
        "chroot"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_CHROOT",
      "includes": {
        "caps": [
          "CAP_SYS_CHROOT"
        ]
      }
    },
    {
      "names": [
        "mount",
        "pivot_root",
        "sethostname",
        "setdomainname",
        "quotactl",
        "quotactl_fd",
        "lookup_dcookie"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_ADMIN",
      "includes": {
        "caps": [
          "CAP_SYS_ADMIN"
        ]
```

continues on next page

continued

Relaxed

```
      }
    },
    {
      "names": [
        "reboot"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_BOOT",
      "includes": {
        "caps": [
          "CAP_SYS_BOOT"
        ]
      }
    },
    {
      "names": [
        "iopl",
        "ioperm"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_RAWIO",
      "includes": {
        "caps": [
          "CAP_SYS_RAWIO"
        ]
      }
    },
    {
      "names": [
        "init_module",
        "finit_module",
        "delete_module"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_MODULE",
      "includes": {
        "caps": [
          "CAP_SYS_MODULE"
        ]
      }
    },
    {
      "names": [
        "settimeofday",
        "clock_settime",
        "clock_settime64"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_TIME",
      "includes": {
        "caps": [
          "CAP_SYS_TIME"
        ]
```

continues on next page

continued

Relaxed

```
      }
    },
    {
      "names": [
        "acct"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYS_PACCT",
      "includes": {
        "caps": [
          "CAP_SYS_PACCT"
        ]
      }
    },
    {
      "names": [
        "open_by_handle_at"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_DAC_READ_SEARCH",
      "includes": {
        "caps": [
          "CAP_DAC_READ_SEARCH"
        ]
      }
    },
    {
      "names": [
        "syslog"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_SYSLOG",
      "includes": {
        "caps": [
          "CAP_SYSLOG"
        ]
      }
    },
    {
      "names": [
        "perf_event_open"
      ],
      "action": "SCMP_ACT_ALLOW",
      "comment": "Allowed only with CAP_PERFMON",
      "includes": {
        "caps": [
          "CAP_PERFMON"
        ]
      }
    },
    {
      "names": [
        "socket"
```

continues on next page

continued

Relaxed

```
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 38,
          "op": "SCMP_CMP_LT"
        }
      ],
      "comment": "Allow all socket families below AF_ALG (38)"
    },
    {
      "names": [
        "socket"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 39,
          "op": "SCMP_CMP_EQ"
        }
      ],
      "comment": "AF_NFC (39) - between ALG and VSOCK"
    },
    {
      "names": [
        "socket"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 40,
          "op": "SCMP_CMP_GT"
        }
      ],
      "comment": "Allow socket families above AF_VSOCK (40)"
    },
    {
      "names": [
        "personality"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 0,
          "op": "SCMP_CMP_EQ"
        }
      ]
    },
    {
```

continues on next page

continued

Relaxed

```
      "names": [
        "personality"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 8,
          "op": "SCMP_CMP_EQ"
        }
      ]
    },
    {
      "names": [
        "personality"
      ],
      "action": "SCMP_ACT_ALLOW",
      "args": [
        {
          "index": 0,
          "value": 4294967295,
          "op": "SCMP_CMP_EQ"
        }
      ]
    },
    {
      "names": [
        "arch_prctl"
      ],
      "action": "SCMP_ACT_ALLOW",
      "includes": {
        "arches": [
          "amd64"
        ]
      }
    },
    {
      "names": [
        "arm_fadvise64_64",
        "arm_sync_file_range",
        "breakpoint",
        "cacheflush",
        "set_tls"
      ],
      "action": "SCMP_ACT_ALLOW",
      "includes": {
        "arches": [
          "arm",
          "arm64"
        ]
      }
    },
    {
```

continues on next page

continued

Relaxed

```
      "names": [
        "switch_endian"
      ],
      "action": "SCMP_ACT_ALLOW",
      "includes": {
        "arches": [
          "ppc64"
        ]
      }
    }
  ]
}
```

5.2. Train

5.2.1. ECN Subnetting

Within this document supplemental information is provided that aims to aid users in understanding the concept of spitting IEC61375 ECN subnets into smaller subnets within the same ECN. This document assumes a deep understanding of the −2-3 and −2-5 sections of the standard.

Background and use cases

Consider the following, standard, IEC61375 train topology:

```
+---------------------------+       +---------------------------+
|         Consist 1         |  ETB  |         Consist 2         |
|       +-----------+  +----|-------|----+  +-----------+       |
|       |+---------+|  |+---|-------|---+|  |+---------+|       |
|       ||         ||  ||   |       |   ||  ||         ||       |
| +-----++-+     +-++--++-+ |       | +-++--++-+     +-++-----+ |
| |  ETBN  |     |  ETBN  | |       | |  ETBN  |     |  ETBN  | |
| |   #1   |     |   #2   | |       | |   #3   |     |   #4   | |
| |        |     |        | |       | |        |     |        | |
| +-+--+---+     +---+--+-+ |       | +----+---+     +----+---+ |
|   |  |             |  |   |       |      |              |     |
|   |  | 10.0.0.0/21 |  |   |       |      | 10.0.0.0/18  |     |
|   |  |    ECN 1    |  |   |       |      |    ECN 1     |     |
|   |  +---------+---+  |   |       |      +--+-----------+     |
|   |            |      |   |       |         |                 |
|   |       +----+---+  |   |       |  +------+-+               |
|   |       |        |  |   |       |  |        |               |
|   |       |  ED A  |  |   |       |  |  ED C  |               |
|   |       |        |  |   |       |  |        |               |
|   |       +--------+  |   |       |  +--------+               |
|   | 10.0.8.0/21       |   |       +---------------------------+
|   |    ECN 2          |   |
|   +-----------+------+   |
|               |          |
|           +----+---+     |
|           |        |     |
|           |  ED B  |     |
|           |        |     |
|           +--------+     |
+---------------------------+
```

Assume that the train reference direction is to the left; the ETBNs will then get the train scope ETBN IDs indicated in the figure. Consequently, ECNs 1 and 2 in Consist 1 will become train scope subnets 1 and 2, while the ECN in Consist 2 becomes subnet 3. Let us assume that ETBNs 1 and 3 are the master routers for their corresponding ECNs (ETBN 1 for both its ECNs). For the purposes of this discussion, assume also that all end devices use address .100 on their local network.

The topology above will result in the end devices being reachable at the following addresses after −2-5 inauguration:

Device	Local scope	Train scope	Subnet
ED A	10.0.0.100	10.128.64.100	1
ED B	10.0.8.100	10.128.128.100	2
ED C	10.0.0.100	10.128.192.100	3

Now consider if the ECN in Consist 2 was to be divided into two parts, with a router in between. This need could be motivated for instance by certain cyber security requirements, such that one part of the ECN could communicate with the other part, but not necessarily with the remainder of the train. While this would be policed by the aforementioned router (also acting as a firewall), the IEC61375 software stack will need to accommodate such a setup.

ECN subnetting

In this topology it can be seen how within **Consist 2** *ECN 1* has been split into two smaller subnets.

```
+---------------------------+       +---------------------------+
|         Consist 1         |  ETB  |         Consist 2         |
|       +-----------+  +----|-------|----+  +-----------+       |
|       |+---------+|  |+---|-------|---+|  |+---------+|       |
|       ||         ||  ||   |       |   ||  ||         ||       |
| +-----++-+     +-++--++-+ |       | +-++--++-+     +-++-----+ |
| |  ETBN  |     |  ETBN  | |       | |  ETBN  |     |  ETBN  | |
| |   #1   |     |   #2   | |       | |   #3   |     |   #4   | |
| |        |     |        | |       | |        |     |        | |
| +-+--+---+     +---+--+-+ |     _ | +----+---+     +----+---+ | _
|   |  |            |  |   |    /  |      |             |     |  \
|   |  | 10.0.0.0/21 |  |   |    |  |      |             |     |  |
|   |  |    ECN 1    |  |   |    |  |      |    ECN 1    |     |   >
|   |  +---------+---+  |   |    |  |      +--+--------+--+     |  | \
|   |            |      |   |    |  |         |        |        |  |  \
|   |       +----+---+  |   |    |  |  +------+-+      |        |  |   \
|   |       |        |  |   |    |  |  |        |      |        |  |    |
|   |       |  ED A  |  |   |    |  |  |  ED C  |      |        |  |    |
|   |       |        |  |   |   <   |  |        |   +--+-----+  |  |    |
|   |       +--------+  |   |  / |  |  +--------+   | ECN    |  | _/    |
|   | 10.0.8.0/21       |   |  | |  |               | router |  | _     |
|   |    ECN 2          |   |  | |  |               |        |  |  \    |
|   +------------+------+   |  | |  |               +--+-----+  |  |    |
|                |          |  | |  |                  |        |   >   |
|           +----+---+      |  | |  |               +--+-----+  |  | \  |
|           |        |      |  | |  |               |        |  |  | |  |
|           |  ED B  |      |  | |  |               |  ED D  |  |  | |  |
|           |        |      |  | |  |               |        |  |  | |  |
|           +--------+      |  | \_ |               +--------+  | _/ |  |
+---------------------------+  |    +---------------------------+    |  |
                               |                                     |  |
                               |                      ECN 1 part b --/  |
                               |                      10.0.8.0/21       |
                               |                                        |
                               \--- ECN 1             ECN 1 part a -----/
                                    10.0.0.0/18       10.0.0.0/21
```

The difference here, as can be seen, is that the ECN in Consist 2 has been split into two subnets, named parts *a* and *b*. While the ETBNs in this consist will still report there being one local ECN in the consist, the parts have disjoint address ranges - two /21 subnets of the /18 ECN. Note that in this specific case, part of the /18 ECN does not exist, even though it is addressible (specifically, the address range **10.0.16.0** through **10.0.63.255**, inclusive), which leads to the addresses being effectively wasted. This could be solved by, for instance, making the two /21 subnets larger (**10.0.0.0/19** and **10.0.32.0/19** respectively) or by adding several more ECN parts and routers (in parallel to part *b*), but this has been omitted here for simplicity.

The new ECN router needs two interfaces, one facing each part of the ECN. Let us assume that it uses address .200 on both its interfaces. From the perspective of the ETBNs in Consist 2, the entirety of the ECN is still reachable through the same interface as before; the only change from before is that an additional route is needed, to **10.0.8.0/21** (part *b* of the ECN) via **10.0.0.200** (the IP address of the ECN router, facing part *a*).

The entirety of the ECN will be mapped onto one train scope subnet, leading to the following addresses being used:

Device	Local scope	Train scope	Subnet
ED A	10.0.0.100	10.128.64.100	1
ED B	10.0.8.100	10.128.128.100	2

Device	Local scope	Train scope	Subnet
ED C	10.0.0.100	10.128.192.100	3
ECN router	10.0.0.200, 10.0.8.200		
ED D	10.0.8.100	10.128.200.100	3

The network configuration in this case is a little bit interesting. The ETBNs see the ECN as one large **/18** network, but they have a better (more specific) route to part *b* (a subnet of the ECN). End devices on part *a*, such as **ED C**, however see the network as a **/21**, with the ETBN as their default gateway. This also applies to the ECN router. While this seems to work when tested, it could pose problems in certain cases (e.g. due the broadcast addresses being different). The ECN router itself will also use the ETBNs as the default gateway, in addition to having link scope routes to both parts of the ECN. As discussed previously, a new route will need to be added on the ETBNs in the consist, to allow for part b of the ECN to be reachable via the ECN router. The remainder of the ECN will be reachable via the regular link scope ECN interface route. In redudant setups this route is manually installed in all ETBNs in the consist.

Additionally, for −2-3 purposes the static consist configuration on the ETBNs must be adjusted. Since EDs **C** and **D** are in the *same* subnet for TTDP purposes, they must both have the same *cnId* value in the static consist configuration. **ED D** will however need to have its *fctId* value added to 2048 (8<<8), to make the IP address resolution work. This is may seem convoluted, but creates the correct behaviour.

An advantage of this solution is that the IEC61375 RNAT rules can remain as they are; **10.128.200.100** in the table above will be NATed to **10.0.8.100** as usual, and in turn be routed through the ECN router. There will however be a need for a new configuration setting in the ETBNs, to allow for the new required route.

These routes is not added in TTDP but instead something that is statically configured as would be done in the normal route management for each ETBN, if needed.

The consist can contain several ECNs, just as before, with some or all of them using this new concept. There is also no change in regards to redundant routers. The concept should work the same regardless of whether the ECN is served by one ETBN or a set of redundant ETBNs. However, it would not be possible to have the union of several different ECN parts served by different ETBNs (or sets of redundant ETBNs) be portrayed as a single train scope subnet. Fundamentally, this would be confusing considering how the ETBN-to-CN map works in TTDP, but this can be worked around, as will be discussed later.

6. File Appendix

`cst1-cstinfo.json`: CSTINFO data file for consist 1


```
{
    "version": "0x0100",
    "cstId": "UIC000000000100",
    "cstType": "EMU",
    "cstOwner": "Westermo",
    "etbInfoList": [
  {
      "etbId": 0,
      "cnCnt": 1
  }
    ],
    "vehInfoList": [
  {
      "vehId": "UIC000000000101",
      "vehType": "cab",
      "vehOrient": 1,
      "cstVehNo": 1,
      "tractVeh": true,
      "fctInfoList": [
    {
        "fctName": "devCCU01",
        "fctId": 10,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devECSC",
        "fctId": 50,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devHMI1",
        "fctId": 20,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devGED1",
        "fctId": 81,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "grpCCU0",
        "fctId": 10,
        "grp": true,
        "etbId": 0,
        "cnId": 0
```


© 2026 Westermo Network Technologies AB

```
    },
    {
        "fctName": "grpHMI",
        "fctId": 20,
        "grp": true,
        "etbId": 0,
        "cnId": 0
    }
      ]
  },
  {
      "vehId": "UIC000000000102",
      "vehType": "coach",
      "vehOrient": 1,
      "cstVehNo": 2,
      "tractVeh": false,
      "fctInfoList": [
    {
        "fctName": "devDCU1",
        "fctId": 110,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devDCU2",
        "fctId": 111,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devHVAC",
        "fctId": 100,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    }
      ]
  },
  {
      "vehId": "UIC000000000103",
      "vehType": "cab",
      "vehOrient": 2,
      "cstVehNo": 3,
      "tractVeh": false,
      "fctInfoList": [
    {
        "fctName": "devCCUO2",
        "fctId": 11,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devHMI2",
```


© 2026 Westermo Network Technologies AB

```
        "fctId": 21,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devGED2",
        "fctId": 82,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    }
      ]
  }
    ]
}
```


`cst2-cstinfo.json`: CSTINFO data file for consist 2


```
{
    "version": "0x0100",
    "cstId": "UIC000000000200",
    "cstType": "EMU",
    "cstOwner": "Westermo",
    "etbInfoList": [
  {
      "etbId": 0,
      "cnCnt": 1
  }
    ],
    "vehInfoList": [
  {
      "vehId": "UIC000000000201",
      "vehType": "cab",
      "vehOrient": 1,
      "cstVehNo": 1,
      "tractVeh": false,
      "fctInfoList": [
    {
        "fctName": "devCCU01",
        "fctId": 10,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devECSC",
        "fctId": 50,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devHMI1",
        "fctId": 20,
        "grp": false,
        "etbId": 0,
```


© 2026 Westermo Network Technologies AB

```
        "cnId": 1
    },
    {
        "fctName": "devGED1",
        "fctId": 81,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "grpCCUO",
        "fctId": 10,
        "grp": true,
        "etbId": 0,
        "cnId": 0
    },
    {
        "fctName": "grpHMI",
        "fctId": 20,
        "grp": true,
        "etbId": 0,
        "cnId": 0
    }
      ]
  },
  {
      "vehId": "UIC000000000202",
      "vehType": "coach",
      "vehOrient": 1,
      "cstVehNo": 2,
      "tractVeh": false,
      "fctInfoList": [
    {
        "fctName": "devDCU1",
        "fctId": 110,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devDCU2",
        "fctId": 111,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    },
    {
        "fctName": "devHVAC",
        "fctId": 100,
        "grp": false,
        "etbId": 0,
        "cnId": 1
    }
      ]
  }
```


© 2026 Westermo Network Technologies AB

```
    ]
}
```

`cstinfo-config-schema.json`: comm profile configuration schema

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "version": "",
    "type": "object",
    "properties": {
        "version": {
            "type": "string",
            "pattern": "^0x[0-9A-Fa-f]{4}$",
            "description": "Version of the CSTINFO structure as per -2-3. Note that
this is not the same as the version of this JSON file - the value written here is
sent on the wire."
        },
        "cstId": {
            "type": "string",
            "description": "Consist identifier, e.g. UIC number. User defined.",
            "maxLength": 16
        },
        "trnId": {
            "type": "string",
            "description": "Train identifier, e.g. UIC number. User defined and
optional. Note that this doesn't get sent out on the wire in CSTINFO telegrams, but
is part of the Operational Train Directory structure anyway. Using different values
across different consists may prevent them from reaching the SHARED Operational Train
Directory state together.",
            "maxLength": 16
        },
        "trnOperator": {
            "type": "string",
            "description": "Train operator, e.g. company name. User defined and
optional. Note that this doesn't get sent out on the wire in CSTINFO telegrams, but
is part of the Operational Train Directory structure anyway. Using different values
across different consists may prevent them from reaching the SHARED Operational Train
Directory state together.",
            "maxLength": 16
        },
        "cstType": {
            "type": "string",
            "description": "Consist type, e.g. EMU, DMU, etc. User defined.",
            "maxLength": 16
        },
        "cstOwner": {
            "type": "string",
            "description": "Consist owner, e.g. company name. User defined.",
            "maxLength": 16
        },
        "dns-local-address": {
            "type": "boolean",
            "description": "True if the local (consist scope) address is returned by
TCN-DNS in default. False if the train-scope address is used. Optional."
        },
        "etbInfoList": {
            "type": "array",
```

© 2026 Westermo Network Technologies AB

```
            "items": {
                "type": "object",
                "properties": {
                    "etbId": {
                        "type": "integer",
                        "description": "ETB ID, determines the address space of
related ECNs and their functions/groups.",
                        "minimum": 0,
                        "maximum": 3
                    },
                    "cnCnt": {
                        "type": "integer",
                        "description": "Number of CNs within the consist connected to
this ETB. CNs are always numbered sequentially starting with 1, so this count is
sufficient.",
                        "minimum": 1,
                        "maximum": 16
                    }
                },
                "required": ["etbId", "cnCnt"]
            },
            "minItems": 1,
            "maxItems": 4,
            "uniqueItems": true,
            "description": "List of ETBs. Each ETB is identified by its ID (0-3). The
cnCnt indicates the number of CNs in the ETB. The list must contain at least one ETB
and can contain up to four ETBs."
        },
        "fctInfoList": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "fctName": {
                        "type": "string",
                        "description": "Function or group name. Becomes the host part
of the URI. -2-3 gives this some additional semantic meaning, as described in the
TCN-DNS guide.",
                        "maxLength": 16
                    },
                    "fctId": {
                        "type": "integer",
                        "description": "Function ID. Must be unique within the
consist. Becomes the host part of the IP address.",
                        "minimum": 1,
                        "maximum": 32768
                    },
                    "grp": {
                        "type": "boolean",
                        "description": "Indicates if the function is a group and will
resolve to a multicast address. Optional."
                    },
                    "etbId": {
                        "type": "integer",
                        "description": "ETB ID to which this function is related. 255
for all-train groups.",
```

© 2026 Westermo Network Technologies AB

```
                        "minimum": 0,
                        "maximum": 255
                    },
                    "cnId": {
                        "type": "integer",
                        "description": "CN on which this function/group exists.
Becomes the 'subnet' after inauguration, which then becomes the network part of the
IP address.",
                        "minimum": 0,
                        "maximum": 32
                    }
                },
                "required": ["fctName", "fctId", "grp", "etbId", "cnId"],
                "additionalProperties": false
            },
            "uniqueItems": true,
            "minItems": 0,
            "maxItems": 3072,
            "description": "List of functions/groups. Each function/group is
identified by its name and ID. See -2-3 and the TCN-DNS guide for descriptions of
these fields."
        },
        "vehInfoList": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "vehId": {
                        "type": "string",
                        "description": "Vehicle identifier, e.g. UIC number. User
defined.",
                        "maxLength": 16
                    },
                    "vehType": {
                        "type": "string",
                        "description": "Vehicle type, e.g. cab, trailer, etc. User
defined.",
                        "maxLength": 16
                    },
                    "vehOrient": {
                        "type": "integer",
                        "description": "Vehicle orientation. 1 for direct, 2 for
inverse.",
                        "minimum": 1,
                        "maximum": 2
                    },
                    "cstVehNo": {
                        "type": "integer",
                        "description": "Consist vehicle number (index), starting with
1.",
                        "minimum": 1,
                        "maximum": 32
                    },
                    "tractVeh": {
                        "type": "boolean",
                        "description": "True if the vehicle is a traction vehicle,
```


© 2026 Westermo Network Technologies AB

```
false if it is a trailer."
                    },
                    "vehPropSlot": {
                        "type": "integer",
                        "description": "Vehicle property slot number, starting with
1. This needs to match a 'slot' attribute of an element in 'vehPropList' below.",
                        "minimum": 1,
                        "maximum": 32
                    },
                    "fctInfoList": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "fctName": {
                                    "type": "string",
                                    "description": "Function or group name, user
defined. Becomes the host part of the URI.",
                                    "maxLength": 16
                                },
                                "fctId": {
                                    "type": "integer",
                                    "description": "Function ID, user defined. Must
be unique within the consist. Becomes the host part of the IP address.",
                                    "minimum": 1,
                                    "maximum": 16382
                                },
                                "grp": {
                                    "type": "boolean",
                                    "description": "Indicates if the function is a
group and will resolve to a multicast address. Optional."
                                },
                                "etbId": {
                                    "type": "integer",
                                    "description": "ETB ID to which this function is
related. 255 for all-train groups.",
                                    "minimum": 0,
                                    "maximum": 255
                                },
                                "cnId": {
                                    "type": "integer",
                                    "description": "CN on which this function/group
exists. Becomes the 'subnet' after inauguration, which then becomes the network part
of the IP address.",
                                    "minimum": 0,
                                    "maximum": 32
                                }
                            },
                            "required": ["fctName", "fctId", "grp", "etbId", "cnId"],
                            "additionalProperties": false
                        },
                        "uniqueItems": true,
                        "minItems": 0,
                        "maxItems": 3072,
                        "description": "List of functions/groups for this vehicle.
See above."
```

© 2026 Westermo Network Technologies AB

```
                    }
                },
                "required": ["vehId", "vehType", "vehOrient", "cstVehNo",
"tractVeh"],
                "additionalProperties": false
            },
            "uniqueItems": true,
            "minItems": 1,
            "maxItems": 32,
            "description": "List of vehicles. A consist must include at least one
vehicle."
        },
        "cstProp": {
            "type": "object",
            "properties": {
                "version": {
                    "type": "string",
                    "pattern": "^0x[0-9A-Fa-f]{4}$",
                    "description": "Version of the properties data. User defined."
                },
                "data": {
                    "type": "array",
                    "items": {
                        "type": "string",
                        "minLength": 1,
                        "pattern": "^[A-Za-z0-9+\\/]*={0,2}$"
                    },
                    "description": "Opaque properties data. User defined. Represented
here as a list of base64 strings, which will be concatenated in order before being
decoded for transmission. 43692 is the length of a base64-encoded string of 32769
bytes.",
                    "minItems": 1,
                    "maxItems": 43692,
                    "uniqueItems": false
                }
            },
            "required": ["version", "data"]
        },
        "vehPropList": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "version": {
                        "type": "string",
                        "pattern": "^0x[0-9A-Fa-f]{4}$",
                        "description": "Version of the properties data. User
defined."
                    },
                    "slot": {
                        "type": "integer"
                    },
                    "data": {
                        "type": "array",
                        "items": {
                            "type": "string",
```

© 2026 Westermo Network Technologies AB

```
                            "pattern": "^[A-Za-z0-9+\\/]*={0,2}$"
                        },
                        "description": "Opaque properties data. User defined.
Represented here as a list of base64 strings, which will be concatenated in order
before being decoded for transmission. 43692 is the length of a base64-encoded string
of 32769 bytes.",
                        "minItems": 1,
                        "maxItems": 43692,
                        "uniqueItems": false
                    }
                },
                "required": ["version", "slot", "data"]
            }
        }
    },
    "required": ["version", "cstId", "cstType", "cstOwner", "etbInfoList",
"vehInfoList"]
}
```


© 2026 Westermo Network Technologies AB