RedFox Series
Wolverine Series
Lynx Series
Falcon Series
Viper Series
6101-3201

Westermo OS Management Guide

WeOS™

RedFox

Wolverine

Lynx

Falcon

Viper

www.westermo.com

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: http://www.westermo.com

Contents

Part I

Introduction to WeOS and its Management Methods

Chapter 1

Introduction

1.1 Westermo and its WeOS products

Westermo provides an extensive set of network products for robust industrial data communications, *managed* as well as *unmanaged* products. Westermo's products are found in diverse set of harsh environment applications, and where robustness and reliability are vital properties.

This guide describes the extensive functionality of managed Westermo products running the *Westermo OS* (WeOS).

1.2 Getting Started

Please see www.westermo.com for the latest updated version of this document – the *WeOS Management Guide*. There you can also find product User Guides, and other support information for your product.

The dedicated *User Guide* of your product includes information on how to get started with WeOS on your specific product. That is a good place to start if you wish to do the least possible configuration of your switch (i.e., assign appropriate IP settings) before putting it into your network infrastructure.

If the User Guide of your specific product lacks a section on how to get started with WeOS, please visit the chapter 2 (*Quick Start*) of this document.

1.3 Introduction to WeOS

Westermo OS (WeOS) is a network operating system delivering an extensive set of functionality including layer-2 (basic switching, VLAN, IGMP snooping, etc.), layer-3 (routing, firewall, NAT, etc.), and higher-level services (DHCP, DNS, etc.). Furthermore, WeOS provides easy management via a Web interface, via the associated WeConfig tool, and via a USB stick. To satisfy even more advanced customer needs, WeOS provides flexible management via a command line interface (CLI), as well as via SNMP.

WeOS provides two levels of functionality, WeOS *Standard* and WeOS *Extended*. Products running WeOS Standard are outstanding layer-2 switches suitable to build reliable LAN infrastructures. Products running WeOS Extended extends the WeOS functionality by adding routing capabilities and a rich set of related higher level services (NAT, firewall, VPN, etc.).

1.4 How to read this document

This guide is structured in the following parts:

- Part I: This part gives general information on WeOS, and introduces the main methods to *manage* a WeOS unit (WeConfig, Web, CLI and SNMP)[1].

  The information in Part I applies both to products running WeOS Standard and WeOS Extended.

  - Chapter 1 is this chapter.
  - Chapter 2 describes how to *get started* with your WeOS product.
  - Chapters 3 gives an overview of the different ways to manage a WeOS unit. If you need recommendations of which method to use, please read chapter 3.
  - Chapters 4-5 present the WeOS Web and CLI support. Detailed information for Web and CLI Management is provided in the later parts of the document.
  - Chapters 6 is the main source of information for WeOS SNMP support.

[1] For information on how to configure a WeOS unit using a USB memory stick, see Chapter 7.

- Part II: Each of the chapters in this part covers services and features in common software levels *Standard* and *Extended*.
  - Chapter 7 handles general maintenance task (firmware upgrade, configuration file handling, factory reset, etc.) and tools such as *ping*, *traceroute*, which be useful when troubleshooting your network.
  - Chapters 8-12 cover management of Ethernet, SHDSL and xDSL (ADSL/VDSL) ports.
  - Chapters 13-18 concern various layer-2 services in WeOS (VLANs, layer-2 redundancy (FRNT, RSTP, Link Aggregation), and IGMP Snooping).
  - Chapter 19 covers network interface configuration including IP address, netmask, etc., as well system wide network settings such as default gateway and DNS.
  - Chapter 20-25 handles various general settings (System Identity), AAA services, DHCP (Server and Relay), and status maintenance (Alarm, Digital I/O, Front Panel LEDs, and logging).
- Part III covers WeOS router/gateway services. These features are only applicable to WeOS Extended products.
  - Chapters 26-30 describes static and dynamic routing, and VRRP support in WeOS.
  - Chapter 31 concerns NAT and Firewall support.
- Part IV covers WeOS VPN and tunneling services. These features are only provided for WeOS Extended products.
  - Chapters 32 gives an overview to VPN and tunneling services.
  - Chapter 33 covers PPP support (PPP over serial port and PPPoE).
  - Chapter 34 describes GRE tunneling support.
  - Chapters 35 and 36 presents VPN support using IPsec and SSL (OpenVPN).
- Part V contains information on serial port configuration (chapter 38) and applications. These features apply to WeOS products with serial ports, both for WeOS Standard and WeOS Extended.
  - Chapter 39 describes Serial Over IP and Modem Replacement functionality

– Chapter 40-41 cover Modbus Gateway and Microlok Gateway support.

1.5 Westermo products running WeOS

Below you find the list of Westermo products running WeOS, as well as references to their respective *User Guide*:

- Falcon: User Guide [41] (FDV-206-1D1S). (”Basis” platform)
- Lynx: User Guides [46] (Lynx-L110/210) and [42] (Lynx-L106/206-F2G). (”Basis” platform)
- Lynx-DSS: User Guides [43] (L108/208-F2G-S2), [44] (L105/205-S1), and [45] (L106/206-S2). (”Basis” platform)
- RedFox Industrial (RFI): User Guides [48] (”Corazon” platform) and [47] (”Atlas” platform)
- RedFox Industrial Rack (RFIR): User Guide [49] (”Corazon” platform)
- RedFox Rail (RFR): User Guide [50] (RFR-212-FB (”Corazon” platform), and RFR-12-FB (”Atlas” platform)).
- Wolverine: User Guides [37] (DDW-142), [38] (DDW-142-485), [39] (DDW-225) and [40] (DDW-226). (”Basis” platform)
- Viper: User Guides [51] (Viper-112/212 and Viper-112/212-T3G) and [52] (Viper-112/212-P8 and Viper-112/212-T3G-P8) (”Basis” platform)

Note
Atlas, Basis and Corazon denote HW platforms used by different products. Products utilising the same HW platform use the same kind of CPU, and have the same amount of RAM and flash memory.

1.5.1 Product hardware details affecting WeOS functionality

The WeOS functionality described in the Management Guide generally applies to all Westermo products running WeOS of the appropriate software level (Standard or Extended). However, where functionality assumes the presence of certain hardware (such as a USB port), those functions are limited to products including

that hardware. The table below provides a summary of hardware differences affecting the availability of certain WeOS functions. For a more definite description of hardware specifications you are referred to the dedicated *User Guide* of each product (see section 1.5).

	Ethernet Ports	SHDSL Ports	xDSL Port	Serial Port(s)	Console port	Digital In/Out	USB Port	Failover Relay	PoE Ports
Falcon									
FDV-206-1D1S	X		X	X	X	X	X		
Lynx									
L106/206-F2G	X				X	X	X		
L110/210	X				X	X			
Lynx-DSS									
All Lynx-DSS models	X			X	X	X	X		
RedFox Industrial & RedFox Industrial Rack									
All RFI and RFIR models	X				X	X	X		
RedFox Rail									
All RFR models	X						X	X[1]	
Viper									
All ”non-PoE’ models	X				X		X		
All ”PoE” models	X				X		X		X
Wolverine									
DDW-142	X	X[2]		X	X	X	X		
DDW-142-485	X	X[2]		X	X	X	X		
DDW-225	X	X			X	X	X		
DDW-226	X	X		X	X	X	X		

[1] Failover Relay is available on RedFox Rail models ”RFR-12 FB” and ”RFR-212 FB”. See the related User Guide[50] for more information on failover relay functionality.

[2] The DDW-142 and DDW-142-485 SHDSL ports have support for PAF (SHDSL link bonding).

Chapter 2

Quick Start

This section provides a guide to quickly get started with your switch. Only simple configuration procedures will be covered[1]. The steps covered concern:

- Get familiar with the factory default setting
- Configuring an appropriate IP address

2.1 Starting the Switch for the First Time

When booting the switch for the first time the switch will use the factory default setting.

The factory default setting makes the switch operate as a manageable layer-2 switch, where all Ethernet ports belong to the same virtual LAN (VLAN)[2].

- Manageable: The switch is manageable via any of the Ethernet ports. To manage the switch via an Ethernet port you need to know the IP address of the switch (see table 2.1). For switches equipped with a console port, the switch can as well be managed via that port without knowing the IP address of the switch.

[1] For more advanced settings, we refer to the remaining chapters of this guide as well as the online help provided via the Web configuration tool and the Command Line Interface (CLI).

[2] On Falcon series of switches, all Ethernet ports belong to the default VLAN (VLAN 1), while the xDSL port belongs to a separate VLAN (VLAN 1006). That is, by factory default Falcon operates as a router. See chapter 11 for more details.

- Single VLAN: By default all ports on the switch will belong to the same VLAN. Thus, devices connected to different ports of the switch should be able to communicate with each other right away. For more advanced setups, the ports of the switch can be grouped into different VLANs. In the factory default setting all ports belong to VLAN 1.

The default IP setting for the switch is as shown in table 2.1.

	Address	**Netmask**	**Gateway**
Primary IP address	Dynamic (DHCP)	(Dynamic)	(Dynamic)
Secondary IP address	192.168.2.200	255.255.255.0	Disabled

Table 2.1: Factory Default IP settings.

Thus, when you power up your WeOS unit with the factory configuration, you can connect to it via two addresses:

- The *static* IP address *192.168.2.200*: This address is simplest to use if you are setting up a single unit.
- A *dynamic* address assigned by a DHCP server[3] (if present): This address may be simplest to use if you want to connect and configure multiple new WeOS units simultaneously.

Note
Before you put your switch into your production network you should change its IP setting according to your network topology. How you change your IP setting is described in the next section.

2.2 Modifying the IP Setting

The switch can be configured with a static IP setting, or it can get its IP address dynamically via DHCP. The latter case is useful if you are running a DHCP server on the same LAN as the switch will be located.

WeOS provides several management tools, which will be presented further in later chapters of this guide. In this chapter we limit the scope to describe how these tools can be used to update the IP settings of the switch.

[3] In addition, the unit will autoconfigure itself with a *link-local* address in the *169.254.x.x* range, where 'x' is in interval 0-255. See section 19.2.6 for more information.

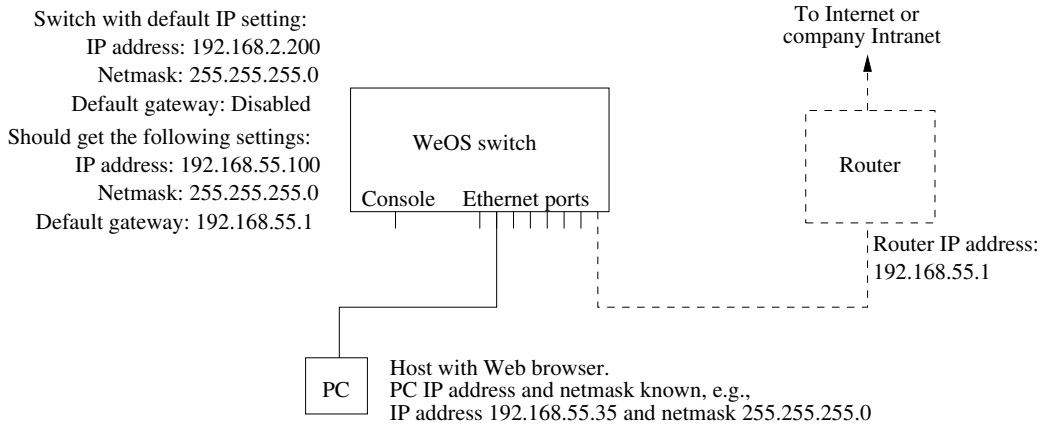
- *WeConfig*: is Westermo's Network configuration management tool (NCM) made for commissioning and maintenance of components in a network. It replaces the former Westermo tool known as *IPConfig*. For further information on WeConfig's features and how to use the tool, see the WeConfig User Guide[54].
- *Web*: Configuration of IP settings via the Web interface is described in section 2.2.1.
- *CLI*: Configuration of IP settings via the Command Line Interface (CLI) is described in section 2.2.2.

Hint

If you are not sure what IP address your switch has, use the *WeConfig* tool, or the *CLI via console* method (section 2.2.2.1). If neither of these methods work, please visit section 7.1.3 for information on how to conduct a factory reset.

2.2.1 Using the Web Interface to Update the Switch IP Settings

To configure the IP settings via web your switch is required to be located on the same IP subnet as your PC.

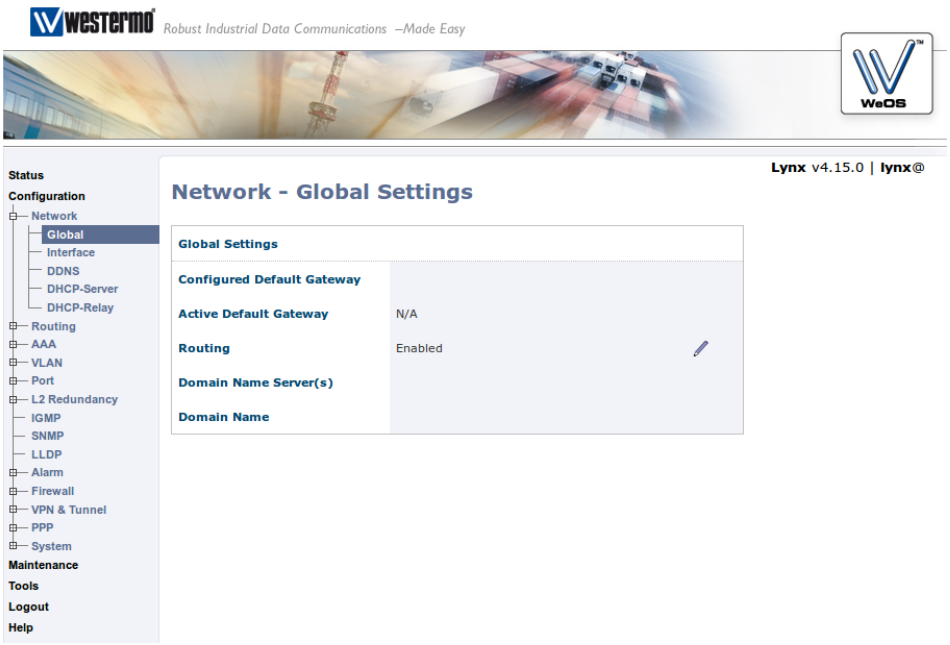


In this *example* the switch shall be assigned the IP address 192.168.55.100, netmask 255.255.255.0 and default gateway 192.168.55.1. To achieve this you must (temporarily) change the IP address of the PC in order to be able to communicate with the switch.

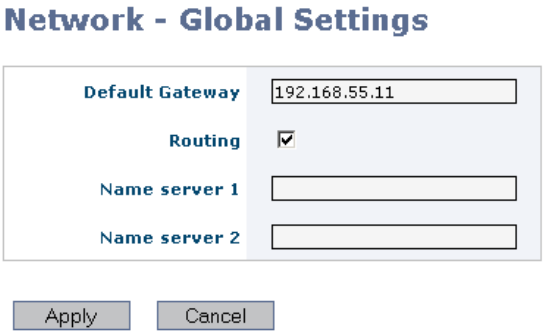
The steps to configure the IP settings via the web interface are as follows:

1. *Connect your PC to the switch*: Connect your PC to the switch as shown in the figure above.
2. *Modifying IP Settings on PC*: The IP settings on the PC must be updated to match the default settings on the switch, i.e., the PC should be assigned an IP address on the 192.168.2.0/24 network, e.g.,
   - PC IP address: 192.168.2.1
   - PC Netmask: 255.255.255.0
3. *Access switch via web browser*: Open your web browser and enter URL **`http://192.168.2.200`** in the browser's address field. You will be asked to enter a *username* and a *password*. Use the factory default account settings shown below:
   - Login username: **admin**
   - Password: **westermo**

4. *Open the Network configuration page*: Click on the **Configuration** top-menu and then on the **Network** sub-menu and then the **Global settings** menu.



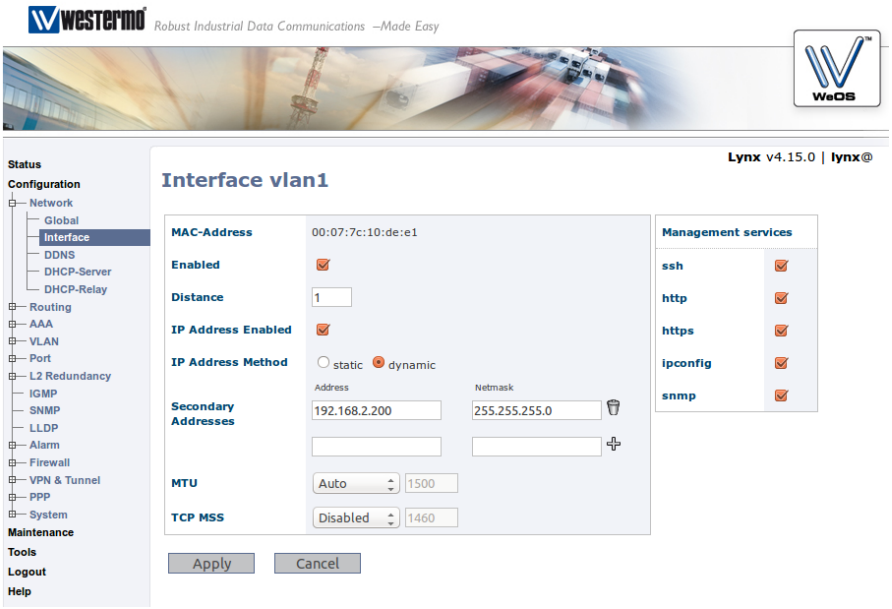
5. *Configure Default Gateway*: Now click the edit icon (✎) in the **Global Settings** frame. The following page should appear.



   Fill in the appropriate address in the **Default Gateway** field. In this example, the default gateway is 192.168.55.1. Click the **Apply** button. Your switch is configured with a new default gateway.

6. *Open Interface Configuration Page*: Click on the **Configuration** top-menu and then on the **Network** sub-menu and then the **Interface** sub menu. In

the **Interface** page, click the *edit* icon (✎) on the row for the interface named **vlan1**. The *Interface Configuration Page* will appear:



7. *Configure Interface IP Settings*: Enter the appropriate IP settings for your switch. In this example we would:

   (a) Set **IP Address Method** to **static** (radio button).

   (b) Set **Primary Address** to **192.168.55.100** with **255.255.255.0** in the **Netmask** field.

   (c) Remove Secondary Address (**192.168.2.200**) using the *trash* icon (🗑).

   Click the **Apply** button and your switch is configured with a new IP address.

8. *Reconfigure PC's IP Settings*: As the IP address is changed on the switch, you cannot reach it from your PC any longer. To access the switch from the PC, the PC's IP settings must be changed again. In this case, we assume it is changed back to its original settings:

   - PC IP address: 192.168.55.35
   - PC Netmask: 255.255.255.0
   - PC Default Gateway: 192.168.55.1

Further management of the switch can be performed via any of the available management tools - WeConfig, Web, SSH/Telnet/CLI or SNMP.

2.2.2 Using the CLI to Update the Switch IP Settings

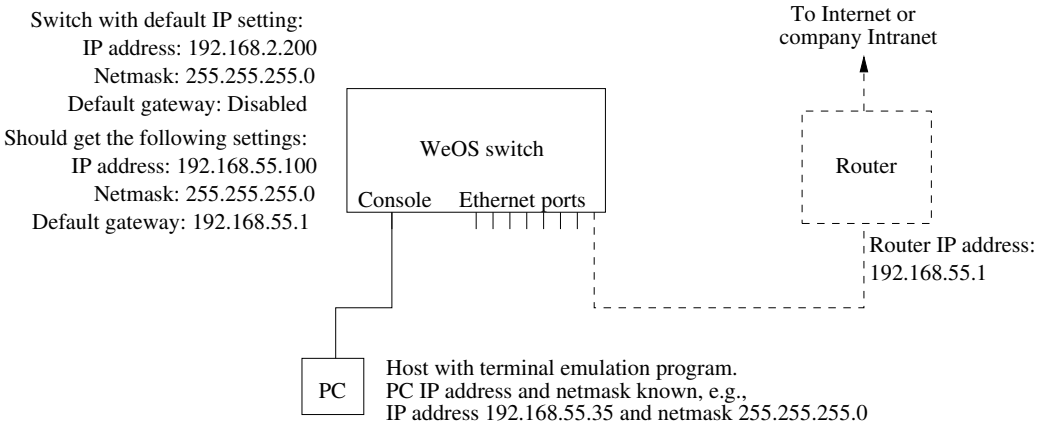
The CLI can be accessed in three ways: via the console port (given that the switch is equipped with a console port) or via the Ethernet ports using the Secure Shell (SSH) or the Telnet protocol. Section 2.2.2.1 explains how to access the CLI via the console port, and how to update the IP settings. Section 2.2.2.2 explains how to access the CLI via SSH.

Access with Telnet is also possible, but this is **not enabled** by default on the switch, and to use it you will first have to access it with one of the other methods and enable this protocol for management. See Section 7.3.49 (CLI) for information on how to *enable the Telnet service* on the unit, and then Section 19.4 (Web) or Section 19.6.6 for information on how to enable Telnet configuration via interface **"vlan1"**.

2.2.2.1 Accessing the CLI via the console port

For WeOS switches equipped with a console port, this port can be used to change IP address of the switch.

1. *Connect your PC to the switch*: Connect your PC to the switch as shown in the figure below.



Important notice for WeOS Switches equipped with a console port

See the User Guide of your specific product (section 1.5) for information on what Diagnostic Cable to use when connecting to the console port of your specific product.

2. *Terminal program*: To communicate with the switch via the console port, you need to use a terminal emulation program on your PC, such as *Hyperterminal*. Ask your system administrator if you need help to *install* or *configure* your terminal emulation program.

   The following settings should be used when connecting to the console port:

Console Port Parameter	Setting
Data rate	115200 bits/s
Data bits	8
Stop bits	1
Parity	Off
Flow control	Off

3. *Activating the console*: When the switch has finished booting, you will be asked to press the **Enter** key on your keyboard to activate the console.

4. *Logging in*: Now you will be asked to enter a *username* and thereafter a *password*. For a switch using the factory default settings, use the following login username and password:

   - Login username: **admin**
   - Password: **westermo**

   Below you see a sample printout when logging in on a WeOS switch. (The password is not ”echoed” back to the screen.)

Example

```
example login:  admin
Password:
.--.--.--.-----.-----.-----.-----.--.--------.-----.
|  |  |  |  -__|__ --|_    _|  -__|   _|  .  .  |  _  | http://www.westermo.com
 \__/\__/|_____._____| |__| |_____|__| |__|__|__|_____|        info@westermo.se
   Robust Industrial Data Communications -- Made Easy

\\/ Westermo WeOS v4.15.0 4.15.0 -- Jun 16 19:10 CEST 2014
Type: ’help’ for help with commands, ’exit’ to logout or leave a context.

example:/#>
```

5. *Listing IP address*: Use the CLI command **”show iface”** to list information about network interfaces.

Example

```
example:/#> show iface
Press Ctrl-C or Q(uit) to quit viewer, Space for next page, <CR> for next line.

Interface Name    Oper  Address/Length      MTU    MAC/PtP Address
---------------   ----  ------------------  -----  --------------------------
lo                UP    127.0.0.1/8         16436  N/A
vlan1             UP    192.168.2.200/24    1500   00:07:7c:10:de:e1
                        169.254.145.230/16
------------------------------------------------------------------------------

example:/#>
```

6. *Changing IP address and netmask*: To change the switch IP addressing mode (”static” instead of ”DHCP”), set a static address and netmask, and to skip secondary addresses, use CLI commands **”configure”**, **”iface vlan1”**, **”inet static”**, **”address <IPV4ADDRESS/LEN>”**, **”no address secondary”** and **”end”** as shown below. This example is based on the setup in step 1, and configures the switch with an address (192.168.55.100/24) on the same IP subnet as the PC.

Example

```
example:/#> configure
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> inet static
example:/config/iface-vlan1/#> address 192.168.55.100/24
example:/config/iface-vlan1/#> no address secondary
Remove all secondary IP addresses, are you sure (y/N)? y
Removing all secondary IPs!
example:/config/iface-vlan1/#> end
example:/config/#> end
Stopping DHCP Clients ..................................... [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> show iface
Press Ctrl-C or Q(uit) to quit viewer, Space for next page, <CR> for next line.

Interface Name    Oper  Address/Length      MTU    MAC/PtP Address
---------------   ----  ------------------  -----  --------------------------
lo                UP    127.0.0.1/8         16436  N/A
vlan1             UP    192.168.55.100/24   1500   00:07:7c:10:de:e1
------------------------------------------------------------------------------

example:/#>
```

7. *Set default gateway IP address*: The figure below shows the same network setup, but with a router attached to the IP subnet.

   With this setup you would like to configure a *default gateway* IP address to allow management of the switch from outside the local network. This

can be achieved using CLI commands **”configure”**, **”ip”**, **”route default 192.168.55.1 <IPADDRESS>”**, and **”end”** as shown below.

Example

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> route default 192.168.55.1
example:/config/ip/#> end
example:/config/#> end
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

8. *Save configuration*: Although the configuration changes has been activated, the running configuration must be stored to the startup configuration. Otherwise the changes will be lost if the switch is rebooted.

Example

```
example:/#> copy running-config startup-config
example:/#>
```

9. You are now done setting the IP address, subnet mask and default gateway of your switch. Logout from the CLI using the **”logout”** command.

Further management of the switch can be performed via any of the available management tools - WeConfig, Web, SSH/Telnet/CLI or SNMP.

2.2.2.2 Accessing the CLI via SSH

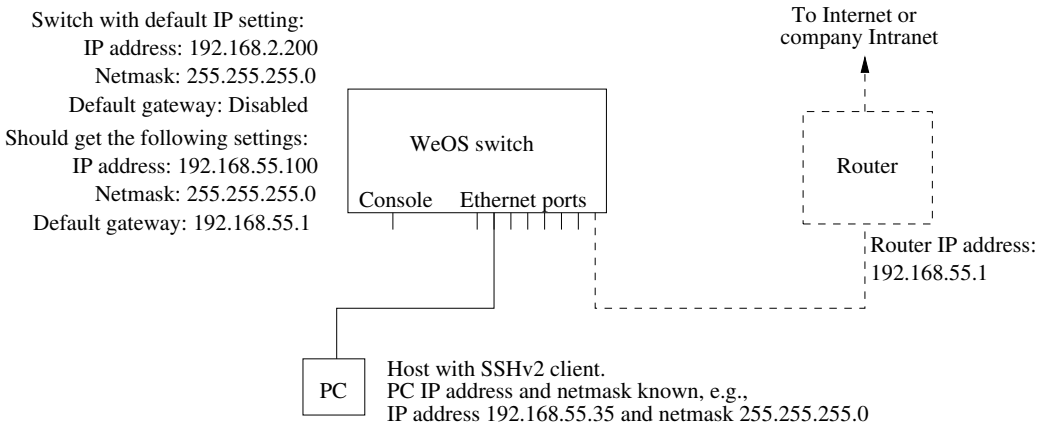
Configuring the IP settings via SSH/CLI is very similar to configuring them via the console port. The major differences are:

- The IP address of the PC must (temporarily) be changed in order to be able to communicate with the switch, i.e., the PC should have an address on network 192.168.2.0/24, e.g., 192.168.2.1/24.
- After the IP settings have been changed on the switch, the PC is likely to loose contact with the switch. The PC must therefore change its IP address again, and login to the switch again in order to copy the running configuration to the startup configuration.

The steps to configure the IP settings via SSH/CLI are as follows:

1. *Connect your PC to the switch*: Connect your PC to the switch as shown in the figure below. In this example we assume the switch will get IP address

192.168.55.100, netmask 255.255.255.0 and default gateway 192.168.55.1.



2. *Modifying IP Settings on PC*: The IP settings on the PC must be updated to match the default settings on the switch, i.e., the PC should be assigned an IP address on the 192.168.2.0/24 network, e.g.,

   - PC IP address: 192.168.2.1
   - PC Netmask: 255.255.255.0
   - PC Default Gateway: Not needed

3. *Connecting and Logging in*: When connecting via SSH you will be asked to enter a *username* and thereafter a *password*. For a switch using the factory default settings, use the following login username and password:

   - Login username: **admin**
   - Password: **westermo**

   The procedure to connect may vary slightly depending on what SSH client you are using. The example below show the connection procedure using Unix OpenSSH[4]. (On Windows one can use Putty[5].)

[4] OpenSSH, http://www.openssh.com
[5] Putty, http://www.chiark.greenend.org.uk/~sgtatham/putty/

Example

```
user@pc:~$ ssh admin@192.168.2.200
The authenticity of host '192.168.2.200 (192.168.2.200)' can't be established.
RSA key fingerprint is 6d:0c:f3:d3:28:d6:d8:43:bc:69:f8:d0:d6:a2:27:87.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.200' (RSA) to the list of known hosts.
admin@192.168.2.200's password:
.--.--.--.-----.-----.------.-----.-.--.--------.-----.
|  |  |  |  -__|__ --|_    _|  -__|   _|  .  .  |  _  | http://www.westermo.com
 \__/\__/|_____._____| |__| |_____|__| |__|__|__|_____|        info@westermo.se
   Robust Industrial Data Communications -- Made Easy

\\/ Westermo WeOS v4.15.0 4.15.0 -- Jun 16 19:10 CEST 2014
Type: 'help' for help with commands, 'exit' to logout or leave a context.

example:/#>
```

4. *Changing IP settings*: The switch IP settings are changed with the same commands as described when accessing the CLI via the console port (section 2.2.2.1). In this example we assign IP address, netmask and default gateway.

Example

```
example:/#> configure
example:/config/#> iface vlan1
example:/config/iface-vlan1/#> inet static
example:/config/iface-vlan1/#> address 192.168.55.100/24
example:/config/iface-vlan1/#> no address secondary
Remove all secondary IP addresses, are you sure (y/N)? y
Removing all secondary IPs!
example:/config/iface-vlan1/#> end
example:/config/#> ip
example:/config/ip/#> route default 192.168.55.1
example:/config/ip/#> end
example:/config/#> end
```

   The configuration is now changed, but not yet saved to the startup configuration. However, as the IP address is changed, the SSH connection will be broken.

5. *Logging in again to save configuration*: To login again, the PC's IP settings must be changed again. In this case, we assume it is changed back to its original settings:

   - PC IP address: 192.168.55.35
   - PC Netmask: 255.255.255.0
   - PC Default Gateway: 192.168.55.1

We can then login again to copy the running configuration to startup configuration.

Example

```
user@pc:~$  ssh admin@192.168.55.100
The authenticity of host '192.168.55.100 (192.168.55.100)' can't be established.
RSA key fingerprint is 6d:0c:f3:d3:28:d6:d8:43:bc:69:f8:d0:d6:a2:27:87.
Are you sure you want to continue connecting (yes/no)?  yes
Warning: Permanently added '192.168.55.100' (RSA) to the list of known hosts.
admin@192.168.55.100's password:
.--.--.--.-----.-----.------.-----.-.--.--------.-----.
|  |  |  |  -__|__ --|_    _|  -__|   _|  .  .  |  _  | http://www.westermo.com
 \__/\__/|_____._____| |__| |_____|__| |__|__|__|_____|        info@westermo.se
   Robust Industrial Data Communications -- Made Easy

\\/ Westermo WeOS v4.15.0 4.15.0 -- Jun 16 19:10 CEST 2014
Type: 'help' for help with commands, 'exit' to logout or leave a context.

example:/#>  copy running-config startup-config
example:/#>
```

You are now done setting the IP address, subnet mask and default gateway of your switch. Logout from the CLI using the **"logout"** command.

Further management of the switch can be performed via any of the available management tools - WeConfig, Web, SSH/CLI or SNMP.

Chapter 3

Overview of Management Methods

WeOS is managed and monitored using the following tools and interfaces:

- **WeConfig**: is Westermo's Network configuration management tool (NCM) made for commissioning and maintenance of components in a network. It replaces the former Westermo tool known as *IPConfig*. For further information on WeConfig's features and how to use the tool, see the WeConfig User Guide[54].

- **Web:** The WeOS Web interface provides management of essential features. The Web interface should satisfy the needs of all common use cases.

- **CLI:** The WeOS Command Line Interface is an industry standard CLI, and provides the most complete management support. The CLI is intended for advanced users requiring fine grain control of the system.

In addition, WeOS provides device management via SNMP (v1/v2c/v3). A set of standard MIBs and the WeOS private MIB are supported, as described in chapter 6.

Task	WeConfig	Web	CLI	SNMP
Discover WeOS Devices	X	(X)	(X)	
Set Device IP Address	X	X	X	X
Upgrade firmware	X	X	X	
Common management tasks		X	X	X
All management tasks			X	
Secure management		X	X	X

In the following sections the properties of the WeConfig tool, the Web Interface, and the CLI are presented further. These sections give information about what management tool to use for a specific need. For more information on SNMP we refer to chapter 6.

3.1 When to use the WeConfig tool

The Westermo configuration management tool, WeConfig, is used for basic configuration and maintenance of WeOS products. It is an ideal tool to *upgrade firmware* and *manage configuration files* (backup and restore) of a *large set of WeOS devices*. With WeConfig you to scan, discover and draw maps of the WeOS devices in your network, and you can also conduct some basic configuration of WeOS units, such as setting the IP address and the default gateway.

For further information on WeConfig's features and how to use the tool, see the WeConfig User Guide[54].

3.2 When to use the Web Interface

The Web interface would be the management interface of choice for most users. The main advantages of the Web Interface are:

- *Easy to use*: The Web management interface provides an *easy to use* method to manage the switch.
- *All common features*: The web interface includes support for all essential management features, and should therefore meet the needs of most users.
- *Secure management:* The web interface can be accessed via regular HTTP and secure HTTP (HTTPS). Secure management is also possible via the CLI (SSHv2) and and SNMP (SNMPv3).

- *Discover other Westermo Switches*: The Web contains a discovery service (IPconfig) similar to what WeConfig provides. (Note, you must still be able to login to one switch in order to make use of this service.)

To use the Web interface, you must know the IP address of your switch. To find out the switch IP address you may need to use the WeConfig tool[1], but once you know it you can do the rest of the management via the Web interface.

The Web interface is introduced in chapter 4.

3.3 When to use the Command Line Interface (CLI)

The WeOS CLI aims to serve advanced users. Furthermore, the CLI is the only management tool which cannot be disabled.

Below we list the situations where the CLI is the most suitable management tool.

- *Complete set of management features*: The CLI includes all the management features available on the switch. If you cannot accomplish your task with any of the other management tools, the CLI may provide the feature you need.
- *Discover other Westermo Switches*: The CLI contains a discovery service similar to what WeConfig provides, but more rudimentary.

  Note
  You must still be able to login to one switch in order to make use of this service.

- *Secure management:* To access the CLI you must either have physical access to the switch (console port), or use the Secure Shell (SSHv2) application to access the CLI remotely. Secure management is also possible via the Web interface (HTTPS) and SNMP (SNMPv3).
- *Configuration scripting:* With a CLI it is possible to develop automatic configuration scripts, e.g., using the *Expect* automation and testing tool. *Expect* extensions exist for many common scripting languages (Ruby, Perl, Tcl).

As with the Web interface, you must know the IP address of your switch before you can access the CLI remotely via SSH (access via the console port is possible

[1] For more information about finding the IP address of your switch we refer to the *Getting Started* guide in chapter 2.

without knowing the switch IP address). To find out the switch IP address you may need to use the WeConfig tool, but once you know it you can do the rest of the management via SSH/CLI.

The WeOS CLI is introduced in chapter 5.

Chapter 4

Management via Web Interface

WeOS supports device management via web interface. Both HTTP and HTTPS[1] are supported. The design is optimised for style sheet and JavaScript[2] capable web browsers. In addition, the design allows users to access the web interface and all settings *without* a style sheet and JavaScript capable browser, but then with less guidance and support from the user interface.

When using the Web Management Tool you have to be aware of the following:

- Only one user can be logged in at a time (see section 4.2 for more information).
- You are automatically logged out after ten (10) minutes of inactivity (see section 4.2 for more information).
- When you click **`Apply`** on a page, the settings on that page are immediately activated.
- When you click **`Apply`** on a page, all settings are stored in the *startup configuration* and therefore survive a reboot (see chapter 7 for more information).

Section 4.2 explains how to access the Web Management Tool and section 4.3 describes the web menu hierarchy. In section 4.3 the *system overview* web pages are presented. Other pages and settings are described per topic in chapter 20 and following chapters.

[1] For HTTPS server authentication, a self-signed certificate is used as of WeOS v4.17.1.
[2] JavaScript is a trademark of Oracle Corporation.

4.1 Document Conventions

Specific conventions for the web part of this document.

`Button Text`	Buttons are indicated by use of bold type-writer style.
Menu path: Top Item ⇒ Sub Item	For each page the menu path to the page is described with this syntax. It means: First click the *Top Item* menu item and in the sub-menu revealed, click the *Sub Item* menu item. See also section 4.3.
Menu path: Top Item ⇒ Sub Item ⇒ **`Button Text`**	This is an extension to the *Menu path: Top Item ⇒ Sub Item* version described above. It tells you to click a button with the text *Button Text* on the page navigated to by *Top Item ⇒ Sub Item*.
Top Item ⇒ Sub Item ⇒  (ctx)	The button may be an icon. In this case the icon is shown. Additionally in parenthesis a sub-context (ctx) may be described which will identify a context on the page, normally identified by its header.

4.2 Logging in

To access the switch through the web interface, enter the appropriate URL (e.g., the factory default IP-address http://192.168.2.200) in the address field of your web-browser. You will then be presented to the login page where you fill in the *username* and *password*, see figure 4.1.

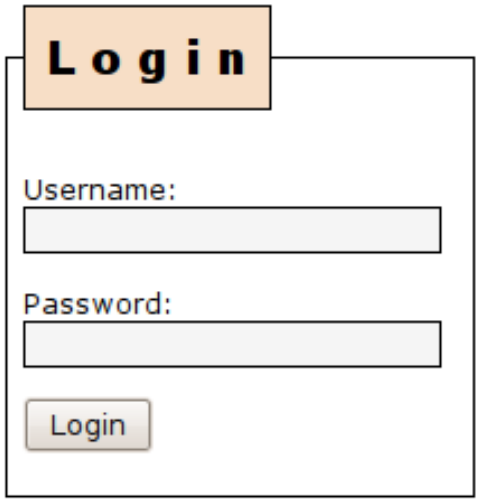


Figure 4.1: Web login window

Currently there is only a single user account defined, the *administrator* user account. Note that it is the same user account used for login in CLI. Factory default user account and password are as follows :

- Login: **admin**
- Password: **westermo**

Your web session will last for ten (10) minutes after your latest ”web action”. Clicking a link or button at least every 10 minutes will let you keep the session

forever. The same goes for pages with an automatic refresh option, given that a refresh interval of 10 minutes or shorter is selected.

Only *one user at a time* can be logged into the switch Web Management Tool. If a new user tries to log in the currently logged in user will automatically be logged out.

4.3 Navigation

After logging in you will be redirected to the *start page*, see fig. 4.2. In the page header you find the menus used to navigate between different tasks. The menu consists of two rows, the *top-menu* row, and the *sub-menu*. For some items you will be presented to a third level sub-menu below the second level sub-menu. Its function is analogously to the second level sub-menu .

To navigate in the menu, click on the *top-menu* to reveal the associated *sub-menu*. Then click on the desired *sub-menu* item. For example, fig. 4.2 shows the selection of top-menu *Status* and sub-menu *Summary* (i.e., Status ⇒ Summary).

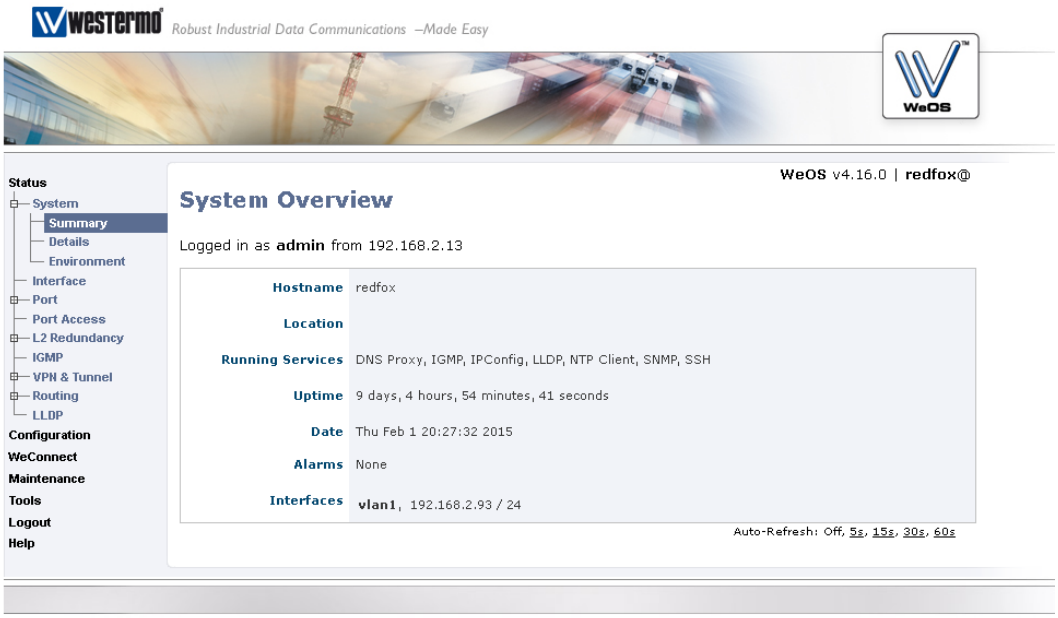


Figure 4.2: Unit Summary - the first page after logging in.

The top-level menu structure is described below:

- Status - This is where you find status information of the running system (port status, protocol status, etc.)
- Configuration - This is where you configure the unit
- Maintenance - This is where you do firmware upgrades, configuration file backups, view log files, manage port monitoring, etc.

- Tools - Here you find various tools for trouble-shooting and other purposes (e.g., ”ping”).

Pages where you can change settings generally contains an **Apply** and a **Cancel** button, as shown in fig. 4.3. The semantics of the **Apply** and **Cancel** buttons are provided below:

Apply	Applies the changes on the current page. Changes are applied immediately (i.e., no reboot needed), and are also stored in the startup configuration.
Cancel	Discards changes and either returns to an overview page for the context, or reloads current page and thus shows the current settings.

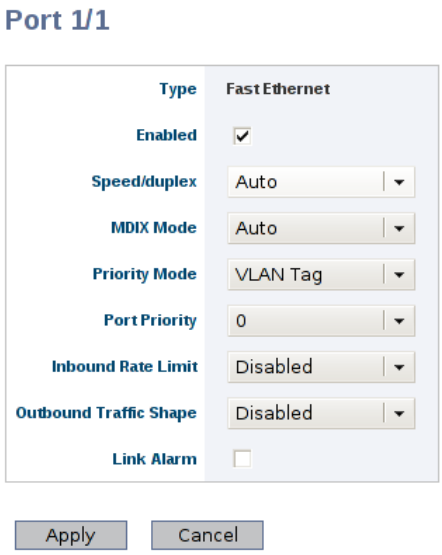


Figure 4.3: Sample web page containing **Apply** and **Cancel** buttons.

Pages with lists of ports may have additional information to display, e.g. if the port is included in a port aggregate or bonded with PAF. This is indicated by the background behind the port label is highlighted as shown in fig. 4.4. When hovering a highlighted port the additional information is displayed in a pop-up. Inside a drop-down menu, the ports are also highlighted, but no pop-ups are presented.

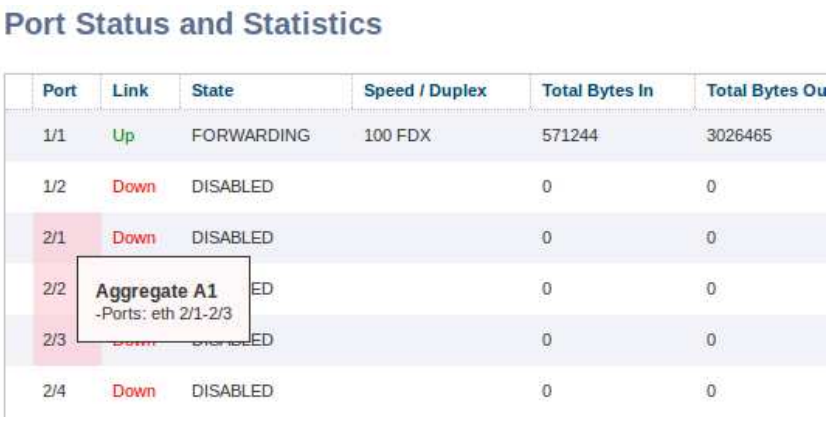


Figure 4.4: Sample web page with port information pop-up.

4.4 System Overview

There are two levels of system information, *summary* and *detailed*.

4.4.1 System Overview - Summary

Menu path: Status ⇒ Summary

Fig. 4.5 shows the first page you will be presented to after logging into the switch. It provides a quick overview of the system, including a list of current alarms.

Figure 4.5: The basic system overview page.

Hostname	An arbitrary name to identify this unit.
Location	An arbitrary description to identify where the unit is located.
ADSL/VDSL Status	Current ADSL/VDSL connection status. Displays negotiation status, IP-address, up/down speed and DSL uptime.

Continued on next page

	Continued from previous page
Uptime	The time passed since last reboot of the unit.
Date	The current date and time. System time is configured manually or set by using a NTP-server.
Running Services	A list of services currently running on the unit.
Alarms	Currently active port and FRNT alarms. *Link alarms* are only shown for ports where link alarm is enabled and when the link is down. *FRNT alarms* are only shown for FRNT ports with link down.
Interfaces	Displays the interfaces and their primary addresses.

4.4.2 System Overview - Detailed

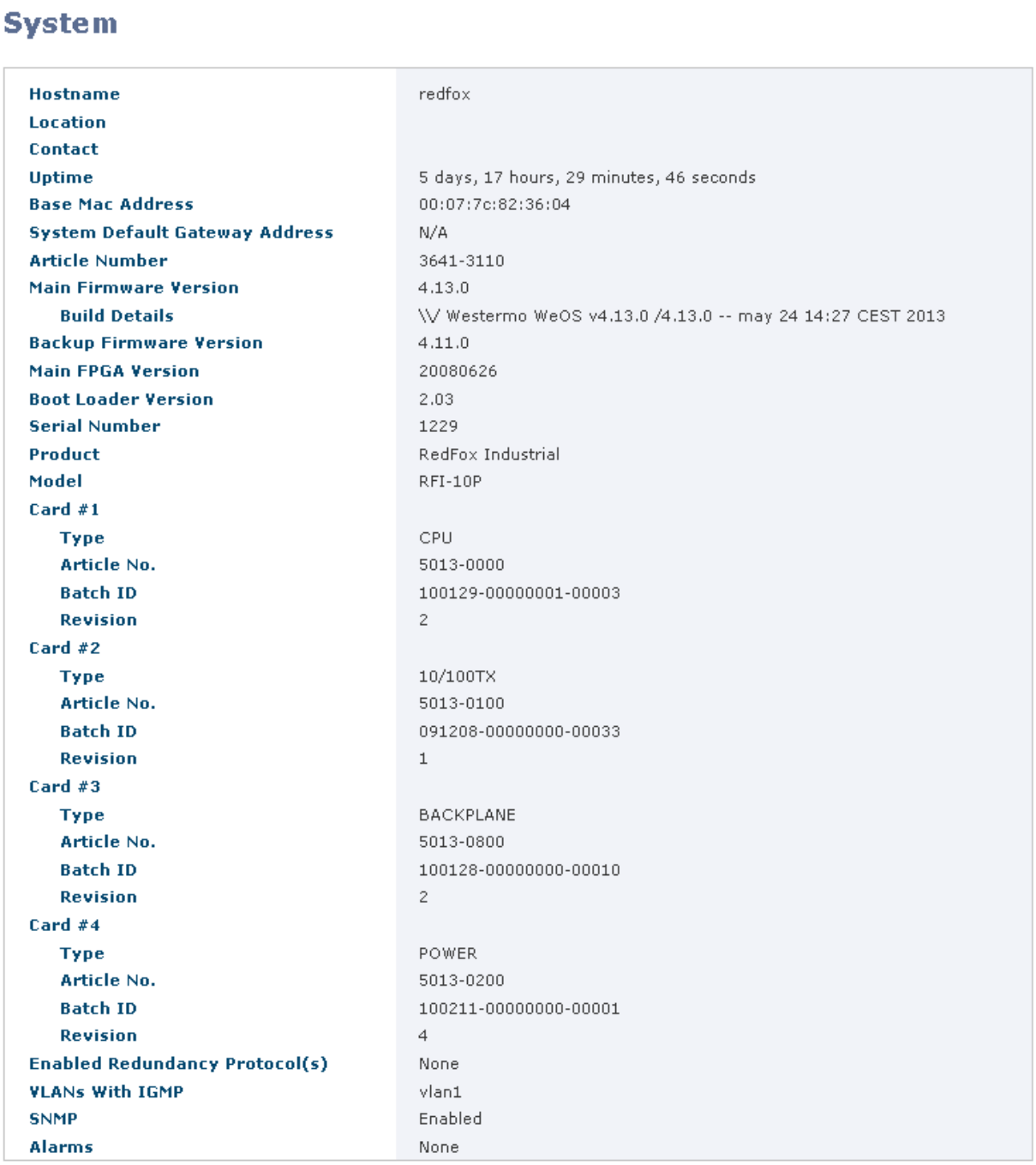
Menu path: Status ⇒ System

To get more information about the switch you go to the detailed page shown in fig. 4.6. This page contains more information on hardware (e.g. versions, article number, etc.) and system status (e.g. memory usage and CPU load).

Hostname	An arbitrary name to identify this unit.
Location	An arbitrary description to identify unit location.
Contact	An arbitrary description to identify a contact person who has more information about management of the unit and the network.
Uptime	The time passed since last reboot of the unit.
Base MAC Address	The base MAC address defines the starting point of the MAC address range used within the unit. This is a unique number assigned to each unit.
System Default Gateway Address	The operational default gateway for all VLANs on the unit. Either retrieved dynamically or set statically.
Article Number	The article number for the unit.
Main Firmware Version	The version number of the main firmware.
Build Details	The build string of the currently running firmware.
Backup Firmware Version	The version number of the backup firmware.
Main FPGA Version	The version number of the FPGA software.
Boot Loader Version	The version number of the boot loader software.
Serial Number	The units serial number.
Product	The product name.
Model	The product model.
Type	Description for the card in the specified slot.
Article No.	The article number of the card in the specified slot.
Batch ID	The batch identification of the card in the specified slot.
Revision	The revision of the card in the specified slot.
Enabled Redundancy Protocol(s)	A list of the redundancy protocols currently enabled on the unit.
VLANs With IGMP	A list of VLANs on which IGMP is enabled.

Continued on next page

	Continued from previous page
SNMP	Shows if SNMP support is enable or disabled.
Alarms	Currently active port and FRNT alarms. *Link alarms* are only shown for ports where link alarm is enabled and link is down.*FRNT alarms* are only shown for FRNT ports where link alarm is enabled and when the link is down.



System

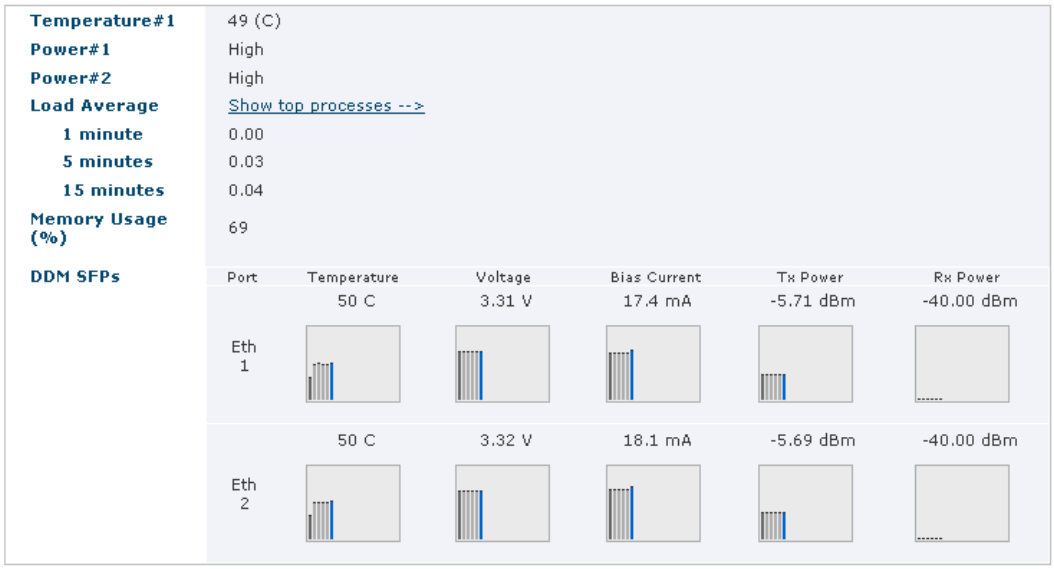
Hostname	redfox
Location	
Contact	
Uptime	5 days, 17 hours, 29 minutes, 46 seconds
Base Mac Address	00:07:7c:82:36:04
System Default Gateway Address	N/A
Article Number	3641-3110
Main Firmware Version	4.13.0
Build Details	\\/ Westermo WeOS v4.13.0 /4.13.0 -- may 24 14:27 CEST 2013
Backup Firmware Version	4.11.0
Main FPGA Version	20080626
Boot Loader Version	2.03
Serial Number	1229
Product	RedFox Industrial
Model	RFI-10P
Card #1	
Type	CPU
Article No.	5013-0000
Batch ID	100129-00000001-00003
Revision	2
Card #2	
Type	10/100TX
Article No.	5013-0100
Batch ID	091208-00000000-00033
Revision	1
Card #3	
Type	BACKPLANE
Article No.	5013-0800
Batch ID	100128-00000000-00010
Revision	2
Card #4	
Type	POWER
Article No.	5013-0200
Batch ID	100211-00000000-00001
Revision	4
Enabled Redundancy Protocol(s)	None
VLANs With IGMP	vlan1
SNMP	Enabled
Alarms	None

Figure 4.6: Detailed system overview page.

4.4.3 System Environment

Menu path: Status ⇒ Environment

To get more information about the system environment variables you go to the environment page.



Temperature	Shows system temperature i Celsius(C).
Load Average	The load average is a standard Linux way of measuring system load.
Memory Usage (%)	A snapshot of RAM (Random Access Memory) usage as percentage of total RAM.
DDM/DOM SFPs [1]	Shows DDM/DOM diagnostics for each SFP. The black bar for each graph represents the first value which was read after boot up, and the blue bar is current value. The DDM/DOM information will be polled for each SFP every twelfth hour. Each graph will then be updated and can consist of up to 20 polled entries. By positioning the mouse over a graph, the user will be presented with startup, max and min value. Please note that each graph shows trend over time and not the absolute value, graphs for different SFP should not be compared.

[1] DDM/DOM diagnostic information is only available for Westermo DDM SFPs, see the SFP Transceiver Datasheet of your WeOS product (www.westermo.com).

Chapter 5

Management via Command Line Interface (CLI)

This chapter introduces the command line interface (CLI) tool. Switches running WeOS include a CLI similar to what is provided by other major vendors of network equipment. The CLI provides a more complete set of management features than the Web interface, the WeConfig tool or SNMP. Thus, when advanced management operations are required, the CLI is the management interface of choice.

The CLI can be accessed via the console port, or remotely via secure shell (SSHv2) and Telnet[1].

Section 5.1 introduces the CLI hierarchy and its various contexts. Section 5.2 explains how to access the CLI interface, and section 5.3 provides general information on how to use the CLI.

The last section (section 5.4) presents CLI commands available in *all* CLI contexts as well as their syntax. Other CLI commands are described per topic in the chapters to follow.

5.1 Overview of the WeOS CLI hierarchy

The WeOS CLI is organised in a hierarchical structure. For management purposes, the use of a hierarchical structure limits the available commands to those relevant for a certain topic. This in turn simplifies switch operation.

[1] Telnet server is by default disabled, see also section 7.3.49.

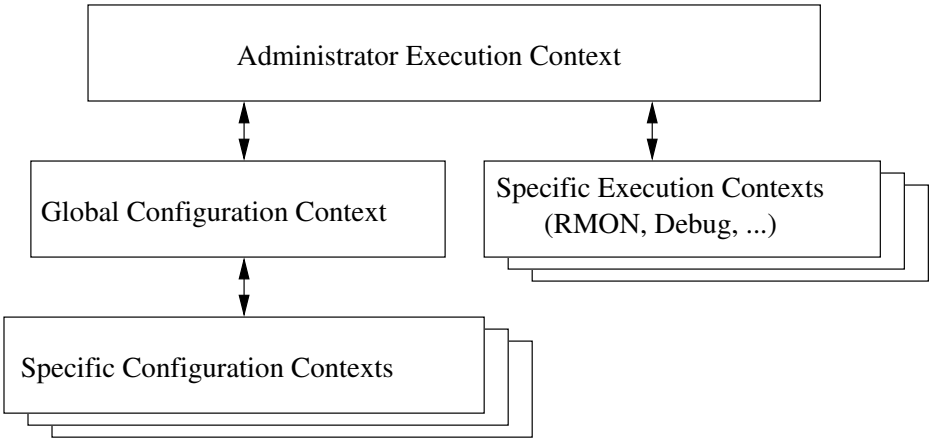


Figure 5.1: CLI hierarchy

Fig. 5.1 shows an overview of the CLI hierarchy. When the user logs in as ”admin” the user will enter the CLI with ”administrator” privileges in Admin Exec context. (In addition to the ”admin” user, future versions of WeOS are likely to support a ”guest” account with limited privileges.)

Admin Exec context In Admin Exec context the user can execute a set of general monitoring and diagnostic functions, and also manage configuration files and firmware versions. From Admin Exec context the user can enter a set of specific execution contexts, e.g., to view RMON statistics.

Global Configuration context From the Admin Exec context the user can enter the Global Configuration context. In Global Configuration the user can configure device parameters of global significance, such as *hostname* and *location* of the device. From Global Configuration the user can reach contexts specific to certain protocols or device entities such as *port*, *vlan*, *interface*, and *FRNT* contexts.

A simple example on CLI usage is given below. There you can see how the CLI prompt changes to match the current context.

Example

```
example:/#> configure
example:/config/#> vlan 100
example:/config/vlan-100/#> untagged 1,2
example:/config/vlan-100/#> end
example:/config/#> end
example:/#>
```

5.2 Accessing the command line interface

To login via the console port you need the username and password. Currently there is only a single user account defined, the *administrator* user account. Factory default account and password:

- Login: **admin**
- Password: **westermo**

The same account is used for management via CLI and Web (see section 4). To reset the *administrator* password to the default setting, see chapter 7.

5.2.1 Accessing CLI via console port

For WeOS switches equipped with a console port, that port can be used to access the CLI. (For information on which WeOS devices that have a console port, see section 1.5.1).

Console cable
See the User Guide of your specific product (section 1.5) for information on what Diagnostic Cable to use when connecting to the console port of your specific product.

Recommended Terminal Emulation programs:

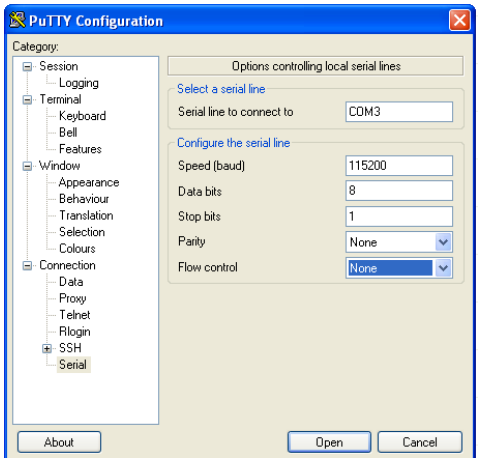
- **Win32**: *PuTTY*, http://www.chiark.greenend.org.uk/~sgtatham/putty/
- **UNIX**: There are different terminal emulation programs for different Unix dialects. On Linux *minicom* is recommended.

The following console port settings are used:

Data rate	115200 bits/s
Data bits	8
Stop bits	1
Parity	None
Flow control	None

The example in below shows how to login via the console port using the *PuTTY* application. Once you have installed and started *PuTTY*, configure the appropriate

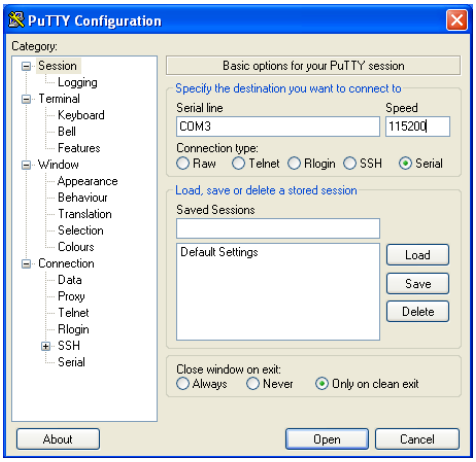
Serial settings.



Hint

In this example, the switch is accessible via the logical port ”COM3”, but the USB/serial adapter may be mapped to a different COM port on your PC. Please check ”Ports (COM and LPT)” in the Windows ”Device Manager” to get information on what COM port to specify.

When the appropriate serial settings have been configured, select the ”Session” view. Select *Serial* as *Connection type* as shown in the figure below.



To start the serial connection, press the **Open** button. The figure below shows the console prompt when logging in to the CLI via the console on a unit named *example*.

```
example login: admin
Password:
.---.--.--.-----.-----.------.-----.-.--.--------.-----.
|   |  |  |  -__|__ --|_     _|  -__|   _|  .  .  |  _  | http://www.westermo.com
 \__/\__/|_____._____| |__| |_____|__| |__|__|__|_____|        info@westermo.se
   Robust Industrial Data Communications -- Made Easy

\\/ Westermo WeOS v4.15.0 4.15.0 -- Jun 16 19:10 CEST 2014
Type: 'help' for help with commands, 'exit' to logout or leave a context.

example:/#>
```

5.2.2 Accessing the CLI via SSH or Telnet

To gain access to the CLI via SSH you need a *SSH client*, the switch IP address, and the account information (username and password).

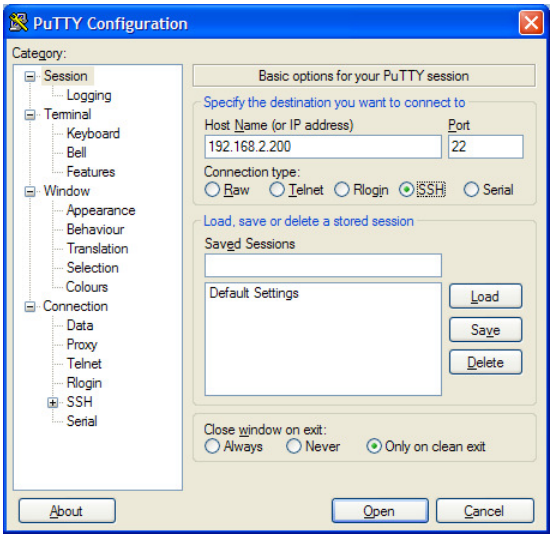
Recommended SSH Clients:

- **Win32**: PuTTY, http://www.chiark.greenend.org.uk/~sgtatham/putty/
- **UNIX** OpenSSH, http://www.openssh.com

The switch IP address can be found using the WeConfig tool, see the WeConfig User Guide[54] (additional methods are listed in section 7.1.3).

The following example illustrates how to login to the switch using PuTTY from a Windows based host system as user *admin*. In this example, the switch is a WeOS switch with IP address 192.168.2.200 (the factory default IP address). See section 5.2 for information about user accounts and passwords.

In the PuTTY session view, select *SSH* as *Connection type*, and enter the IP address of the switch (here 192.168.2.200).



Click the **0pen** button to start the SSH session. You will be presented to a login prompt (see below), and enter login *admin* and the associated password.

```
example login: admin
Password:
.--.--.--.-----.-----.------.-----.-.--.--------.-----.
|  |  |  |  -__|__ --|_     _|  -__|   _|  .  .  |  _  | http://www.westermo.com
 \__/\__/|_____._____| |__| |_____|__| |__|__|__|_____|        info@westermo.se
   Robust Industrial Data Communications -- Made Easy

\\/ Westermo WeOS v4.15.0 4.15.0 -- Jun 16 19:10 CEST 2014
Type: 'help' for help with commands, 'exit' to logout or leave a context.

example:/#>
```

The CLI can be accessed remotely by using a *Telnet* client, in the same way as using SSH. Of security reasons, use of Telnet is discouraged and therefore disabled by default. In order to manage the unit via Telnet, you must first:

- Enable the Telnet server via the CLI, see section 7.3.49.
- Enable telnet management for the desired network interface(s) via the CLI (see section 19.6.6).

5.3 Using the CLI

5.3.1 Starting out with the CLI

When first entering the CLI you end up in the *Admin Exec* context. In the *Admin Exec* you can view system *status* information using various **"show"** commands, upgrade system firmware, etc., as well as other functions, which do not affect the system *configuration*.

To be able to modify the switch configuration you should enter the *Global Configuration* context, by using the **"configure"** command as shown below. From the *Global Configuration* you are able to configure system parameters such as its **"hostname"** or its **"date"**.

Example

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

As described in section 5.3.2 you can reach other, specific configuration contexts from the *Global Configuration* context.

Example

```
example:/#> configure
example:/config/#> vlan 100
example:/config/vlan-100/#> untagged 1/1,1/2
example:/config/vlan-100/#> end
example:/config/#> end
example:/#>
```

To get help on what commands are available in the current context, use the **"help"** command (see example in fig. 5.2). First the context specific configuration commands are shown, followed by the commands to *show* the current configuration settings. At the end, commands available in all contexts are shown (see also section 5.4.).

Example

```
example:/config/vlan-100/#> help
Available Commands
================================================================================
enable                    Enable, or disable this VLAN
name <ARG>                Set name of VLAN
tagged <ARG>              Set tagged ports
untagged <ARG>            Set untagged ports
channel <ARG>             Set VLAN channel interface
priority <ARG>            Set VLAN priority, overrides port priority
igmp                      Enable, or disable IGMP Snooping

show enable               Show if VLAN is active or not
show name                 Show name of VLAN
show tagged               Show tagged ports
show untagged             Show untagged ports
show channel              Show VLAN channel interface
show priority             Show VLAN priority setting
show igmp                 Show IGMP Snooping status

no <ARG>                  Prefix, used to disable services or settings.
do                        Shortcut to EXEC mode, e.g. do ping <IP>.
end                       Save settings and return to previous mode.
leave                     Save settings and return to EXEC mode.
abort                     Cancel all changes and leave this mode.
show <ARG>                Show summary, or status.
repeat <ARG>              Repeat next command every second, until Ctrl-C
help <ARG>                This help text.
tutorial                  Brief introduction to the CLI
================================================================================
<ARG> - Command takes argument(s), see help <command> for further information.
Short forms of commands are possible, see the tutorial for more help.
example:/config/vlan-100/#>
```

Figure 5.2: Use of the **”help”** command to list available commands (here in the VLAN context).

The **”help”** command can also be used to get information on a specific command as shown below.

Example

```
example:/config/vlan-100/#> help igmp
Syntax:
        [no] igmp

Description:
        Enable, or disable IGMP Snooping

================================================================================
The [no] keyword is when you want to disable a service or remove a property.
example:/config/vlan-100/#>
```

The CLI supports basic *TAB-completion*, which can come in handy when you do not know the exact command name, e.g., writing **"fi[TAB]"** within the *IP* context will expand to **"firewall"**.

TAB-completion is only able to expand the full command when there is no ambiguity. Otherwise the available alternatives will be listed.

Example

```
example:/#> d[TAB]
do      debug   date    dir     delete
example:/#> d
```

Furthermore, when there is no ambiguity it is possible to use an abbreviation of a command instead of the full command (i.e., without using *TAB-completion*).

Example

```
example:/#> con
example:/config/#>
```

5.3.2 Entering and leaving CLI contexts

Fig. 5.3 gives a general overview of how to enter and leave the various context in the CLI hierarchy. The commands to move between contexts are further discussed in the text below.

To enter Global Configuration context from Admin Exec context, the **"configure"** command is used. From Global Configuration context one can reach several specific configuration contexts, and the command to enter them is context specific, e.g.,:

vlan <VID>	Manage VLAN settings for VLAN with given VID.
port <PORT>	Manage port settings for port with given PORT identifier.
interface <IFNAME>	Manage settings for the given network interface.

By entering the Global Configuration context the user is able to interactively change the device configuration, however, configuration changes will not take effect until the user leaves the configuration contexts and returns to the Admin Exec context via the **"end"** or **"leave"** commands.

When the user returns to Admin Exec context, the *running-configuration* of the switch will be updated. To make the configuration changes permanent the *running-*

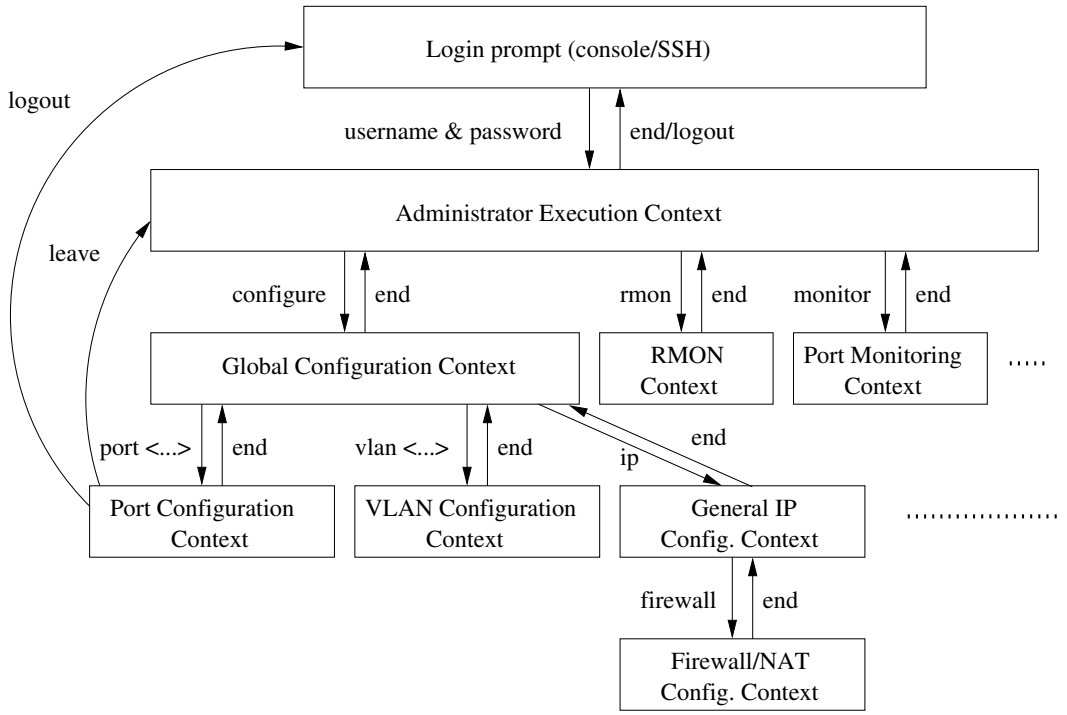


Figure 5.3: Moving between CLI contexts. Only a subset of the available contexts is shown. Although not shown, the *leave* and *logout* commands can be used from all contexts.

configuration should be saved to the *startup-configuration* using the **"copy"** command, see also chapter 7.

It is also possible to leave the configuration contexts without updating the *running-configuration*. The commands to leave a context are listed below. More information on these and other general CLI commands can be found in section 5.4.

end	Confirms configuration changes conducted in this context and returns to the context immediately above. If issued within the Global Configuration context, the user returns to the Admin Exec context and the *running-configuration* is updated.
leave	Confirms configuration changes made and returns to Admin Exec context. The *running-configuration* is updated.
Ctrl-Z	An alias for leave. Ends your configuration session and returns to Admin Exec context.

Continued on next page

	Continued from previous page
abort	Discards configuration changes conducted in this context and returns to the context immediately above. If issued within the Global Configuration context, the user returns to the Admin Exec context without updating the *running-configuration*. If issued in Admin Exec context it works the same as logout.
exit	An alias for abort.
Ctrl-D	An alias for abort. Blocked if any text is already input on the command line.
logout	Log out from the CLI. If conducted from within any of the configuration contexts, all configuration changes are discarded (i.e., the *running configuration* is not updated).

5.3.3 CLI command conventions

This section describes the CLI command conventions used within this guide. The syntax for a sample set of CLI commands is shown below:

- `[no] default-gw <ADDRESS>`
- `igmp-interval <12|30|70|150>`
- `show iface [IFNAMELIST]`

Convention	Description
`command syntax`	Command syntax is generally written in typewriter style (fixed width)
`"command syntax"`	Commands described in running text use bold typewriter style enclosed by quotation marks.
`UPPERCASE`	A variable parameter. Enter value according to the description that follows.
`lowercase`	A keyword parameter. Enter value according to the given syntax.
`|`	Vertical bar. Used to separate alternative (mutually exclusive) parameters.
`< >`	Angle brackets. Encloses a mandatory parameter.
`[ ]`	Squared brackets. Encloses an optional parameter.
	Continued on next page

Continued from previous page

Convention	Description
[< >]	Angle brackets within squared brackets. Encloses a mandatory parameter within an optional choice.

5.4 General CLI commands

The majority of the CLI commands are specific to a certain context, however, there is a set of CLI commands available in all contexts. These commands are explained further here. The **"configure"** command used to enter the Global Configuration context from the Admin Exec context, is also covered.

Command	Section
no <COMMAND>	Section 5.4.1
do	Section 5.4.2
end	Section 5.4.3
leave	Section 5.4.4
abort	Section 5.4.5
logout	Section 5.4.6
repeat <COMMAND>	Section 5.4.7
help [COMMAND]	Section 5.4.8
tutorial	Section 5.4.9
configure [terminal]	Section 5.4.10

5.4.1 Negate/disable a setting

Syntax `no <COMMAND>`

Context All contexts

Usage Depending on context the **"no"** command disables or resets a setting to default.

Primarily used within configuration contexts to negate or disable a configuration setting, e.g., in *port* context **"no flow-control"** disables flow control. For some commands, "no" is used to reset to a default value, e.g., **"no polling-interval"** (NTP client context) sets the NTP polling-interval to its default value (600 seconds).

The **"no"** command can also be used to negate/disable certain commands outside the *configuration* context, e.g., to disable debugging or port monitoring.

Default values Not applicable

5.4.2 Execute (do) command from Admin Exec context

Syntax `do <COMMAND>`

Context All contexts

Usage Use the **”do <COMMAND>”** to execute a COMMAND available in Admin Exec context from any context.

For example, when located in Global Configuration context, the user could run **”do show running-config”** to see the *running configuration*, or run **”do ping 192.168.1.1”** to ”ping” IP address 192.168.1.1.

Default values Not applicable

5.4.3 End context

Syntax end

Context All contexts

Usage Leave this context and return to the context immediately above. If this command is issued within any of the configuration contexts, the command implies that the configuration changes conducted within that context are confirmed. If the command is issued in the Global Configuration context, the user returns to the Admin Exec context, and the *running-configuration* is updated.

Default values Not applicable

5.4.4 Leave context

Syntax `leave`

Context All contexts

Usage Leave this context and return to the Admin Exec context. If this command is issued within any of the configuration contexts, the command implies that the configuration changes conducted are confirmed, and the *running-configuration* is updated.

Default values Not applicable

5.4.5 Abort context

Syntax `abort`

Context All contexts

Usage Leave this context and return to the context immediately above. If this command is issued within any of the configuration contexts, the command implies that the configuration changes conducted within that context are discarded. If the command is issued in the Global Configuration context, the user returns to the Admin Exec context without updating the *running-configuration*.

Default values Not applicable

5.4.6 Logout

Syntax `logout`

Context All contexts

Usage Logout from system. If this command is issued within any of the configuration contexts, the command implies that the configuration changes conducted are discarded, i.e., the *running-configuration* is not updated.

Default values Not applicable

5.4.7 Repeat a command

Syntax `repeat <COMMAND>`

Context Admin Exec context

Usage Repeat COMMAND every second until Ctrl-C is pressed.

Default values Not applicable

5.4.8 On-line help

Syntax `help <COMMAND>`

Context All contexts

Usage Show help information specific to a certain context, or a specific command.

Default values If no COMMAND is specified, help information related to the current context is shown.

5.4.9 CLI tutorial

Syntax `tutorial`

Context All contexts

Usage Show CLI tutorial text.

Default values Not applicable

5.4.10 Entering Global Configuration Context

When a user logs in to the CLI the user will enter the Admin Exec context. In Admin Exec context the user can view status information and have access to tools such as *ping* and *traceroute*, but is not able to perform any configuration. To configure the device, the user can use the *configure* command to enter the Global Configuration context.

Syntax `configure [terminal]`

Context Admin Exec context

Usage Enter global Configuration Context.

The optional `terminal` argument is a compatibility keyword, for advanced users. It disables all safe guards (yes-or-no questions), making it possible to paste-in configuration files into the terminal.

Pasting in configuration files can also be done with the `copy` command as `copy con run` to copy *console* to *running-config*.

Default values Interactive mode (i.e. the **`"terminal"`** argument does not apply by default)

Chapter 6

WeOS SNMP Support

The Simple Network Management Protocol (SNMP) provides a standardised method to manage and monitor IP devices remotely. The WeOS SNMP agent supports SNMP v1, v2c and v3.

6.1 Introduction and feature overview

Table 6.1 shows WeOS SNMP control features for the Web and CLI interfaces. Further description of the SNMP support is presented in the sections 6.1.1-6.1.6. If you are only interested in knowing how to manage SNMP features via the Web or CLI, please visit sections 6.2 or 6.3 directly.

6.1.1 SNMP introduction

The Simple Network Management Protocol (SNMP) provides a standardised method to manage and monitor IP devices remotely. In SNMP a *manager station* can manage a set of status and configuration objects via an SNMP *agent* on the management unit. The WeOS SNMP agent supports SNMP v1, v2c and v3.

An SNMP manager:

- can send SNMP *GET* messages to poll status and configuration information from an SNMP *agent*.

Feature	Web (Sec. 6.2)	CLI (Sec. 6.3)	General Description
General			
Enable/disable SNMP	X	X	
SNMPv1/v2c			
Read Community	X	X	Sec. 6.1.2
Write Community	X	X	”
Trap Community	X	X	Sec. 6.1.2-6.1.3
Trap Host	X	X	Sec. 6.1.3
SNMPv3			
Read-Only SNMPv3 User	X	X	Sec. 6.1.4
Read/Write SNMPv3 User	X	X	”

Table 6.1: WeOS control of SNMP features.

- can send SNMP *SET* messages to the SNMP *agent* to modify the device settings (or issue commands such as ’reboot’).
- can get notified by an agent when specific events occur, such as link down event, via SNMP *TRAP* messages.

The objects manageable via SNMP are defined in a management information base (MIB). The WeOS MIB support aims at providing SNMP management primarily via standard MIBs to enable easy integration with existing SNMP management tools. In addition, WeOS includes an enterprise MIB (private MIB) to provide access to MIB objects not available via the standard MIBs.

6.1.2 SNMP Communities

An SNMP *community* is a relationship between the manager and managed station. It can be seen as a (very) basic authentication and authorisation mechanism for SNMP v1 and v2c[1]. Three types of communities are supported:

- *Read community*: The SNMP read community is used by a manager to read SNMP MIB objects from a managed station.

  Default read community: `public`

[1] See section 6.1.4 for secure management using SNMPv3.

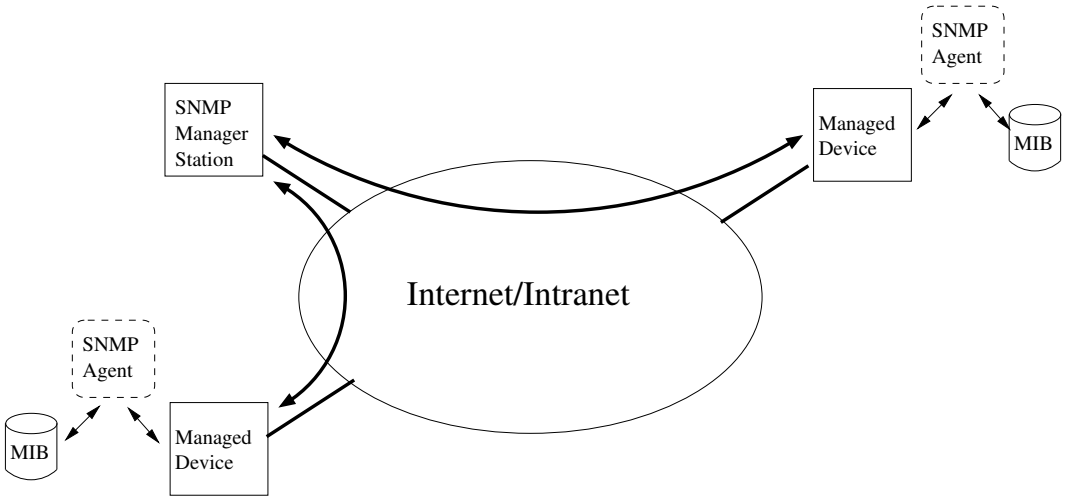


Figure 6.1: Sample SNMP setup, where one manager station controls two devices by communicating with SNMP agents running on the managed devices.

- *Write community*: The SNMP write community can be used to write (and read) SNMP MIB objects to (from) a managed station. Thus, if the agent has its write community enabled, it is possible to configure the switch via SNMP. The write community is typically named **”private”**.

  Default write community: `Disabled`

- *Trap community*: The SNMP trap community is used when an agent wants to send a notification to the manager (SNMP Trap). The trap community is typically named **”public”**.

  Default trap community: `trap`

Warning
Using the well-known community strings ”public” and ”private” could pose a serious security problem.

6.1.3 Trap Support

SNMP traps are only generated if there is at least one *Trap Host* (i.e., SNMP management station) defined. Up to three *Trap Hosts* can be defined. If two or more Trap Hosts are configured, traps will be sent to all of them.

The WeOS SNMP trap support is integrated with the WeOS alarm handling system (see section 24.1). This means that you as an operator have fine-grained control of which traps to send. All traps in the list below, except *Coldstart* and *lldpRemTablesChange*, can be controlled via the alarm handling system.

- *Link Alarm*: A trap is generated on *link up* or *link down*, given that *Link Alarm* is enabled on that specific port (see sections 24.1.3 and 8.1.5).

  Link Down OID: iso(1).org(3).dod(6).internet(1).snmpV2(6).snmpModules(3).snmpMIB(1).snmpMIBObjects(1).snmpTraps(5).linkDown(3)

  Link Up OID: iso(1).org(3).dod(6).internet(1).snmpV2(6).snmpModules(3).snmpMIB(1).snmpMIBObjects(1).snmpTraps(5).linkUp(4)

  > **Note**
  > When a port is being reconfigured, link down and link up events are likely to occur. If *link-alarm* is enabled on that port, a couple of SNMP traps are likely to be generated as a side-effect of the port reconfiguration.

- **Cold Start**: A trap is generated when a system comes up.

  OID: *iso(1).org(3).dod(6).internet(1).snmpV2(6).snmpModules(3).snmpMIB(1).snmpMIBObjects(1).snmpTraps(5).coldStart(1)*

- **LLDP Remote System Update**: A trap is generated when a remote system has updated.

  OID: *iso(1).std(0).iso8802(8802).ieee802dot1(1).ieee802dot1mibs(1).lldpMIB(2).lldpNotifications(0).lldpNotificationPrefix(0).lldpRemTablesChange(1)*

- **Digital-In**: A trap is generated when the voltage level on the pins of a digital-in sensor changes from *high* to *low*, or *low* to *high*.

  Digital-In High OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).sensorNotifications(1).sensorNotificationPrefix(0).digitalInHigh(1)*

  Digital-In Low OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).sensorNotifications(1).sensorNotificationPrefix(0).digitalInLow(2)*

- **Power Supply**: A trap is generated when the voltage level on any of the power feeds changes from *high* to *low*, or *low* to *high*.

Power Supply High OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).sensorNotifications(1).sensorNotificationPrefix(0).powerSupplyHigh(3)*

Power Supply Low OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).sensorNotifications(1).sensorNotificationPrefix(0).powerSupplyLow(4)*

- **Temperature**: A trap is generated when the temperature measured by a built-in temperature sensor reaches the configured rising or falling thresholds.

  Temperature High OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).sensorNotifications(1).sensorNotificationPrefix(0).temperatureHigh(5)*

  Temperature Low OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).sensorNotifications(1).sensorNotificationPrefix(0).temperatureLow(6)*

- **FRNT Ring Status**: A trap is generated when a unit detects a change of FRNT ring status, i.e., ring up (ring mode) or ring down (bus mode).

  FRNT Ring Up OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).frntNotifications(2).frntNotificationPrefix(0).frntRingUp(1)*

  FRNT Ring Down OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).frntNotifications(2).frntNotificationPrefix(0).frntRingDown(2)*

- **SNR-margin**: On units with a SHDSL/xDSL port traps are generated when the SNR margin falls below (or rises above) a configurable threshold.

  OID: *iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).transmission(10).hdsl2ShdslMIB(48).hdsl2ShdslNotifications(0).hdsl2ShdslSNRMarginCrossing(2)*

- **LFF Status**: On units with SHDSL ports, a trap is generated when a unit detects a change in the Link Fault Forward (LFF) status on a SHDSL port, i.e., if the remote end reports that its Ethernet port is up or down.

  LFF Remote Up OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).lffNotifications(3).lffNotificationPrefix(0).lffRemoteUp(1)*

LFF Remote Fail OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).lffNotifications(3).lffNotificationPrefix(0).lffRemoteFail(2)*

- **PoE total power consumption**: On units with Ethernet ports supporting Power over Ethernet, traps are generated with the total consumed power rises above (or falls below) a configurable threshold.

  Power consumption above threshold OID: *iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).powerEthernetMIB(105).pethNotifications(0).pethMainPowerUsageOnNotification(2)*

  Power consumption below threshold OID: *iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).powerEthernetMIB(105).pethNotifications(0).pethMainPowerUsageOffNotification(3)*

- **Summary Alarm Status**: The summary alarm status (*summaryAlarmStatus*) follows the status of the ON LED:

  - when the ON LED turns *red*, the *summaryAlarmStatus* has value *Warning (1)*.

  - when the ON LED turns *green*, the *summaryAlarmStatus* has value *OK (2)*.

  It is possible to get SNMP traps when the summary Alarm Status changes state (see section 24.3.16 for information of how to enable summary alarm traps). When enabled, a *summaryAlarmOK* trap is sent when the ON LED turns *green*, and a *summaryAlarmWarning* trap is sent when it turns *red*.

  Summary Alarm OK OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).genericNotifications(4).genericNotificationPrefix(0).summaryAlarmOK(1)*

  Summary Alarm Warning OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).notifications(6).genericNotifications(4).genericNotificationPrefix(0).summaryAlarmWarning(2)*

  The summary alarm status can be read at the following OID: *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).westermo(16177).common(2).weos(1).system(5).eventSystem(2).summaryAlarmStatus(1)*

6.1.4 Secure management using SNMPv3

To manage a unit securely via SNMP, SNMPv3 should be used. SNMPv3 provides privacy and integrity (per packet authentication) to the SNMP messages.

SNMPv3 introduces the notion of a SNMPv3 *user*, as opposed to the *community* concept used in SNMPv1/v2c. The following parameters can be configured for an SNMPv3 user.

- Read-Only or Read-Write access: Defines whether the *user* should have *read* access to the SNMP variables, or be able to *read* and *modify* them.
- Security Mode: Three security modes are available:
  - *noAuthnoPriv*: No security (i.e., neither authentication, nor encryption)
  - *authNoPriv*: Authentication, but no privacy.
  - *authPriv*: Authentication and Encryption

  Note
  As of WeOS v4.17.1, the WeOS SNMP agent accepts SNMP requests of security level *authNoPriv* also for SNMPv3 users created at level *authPriv*. This feature is likely to be removed in future WeOS releases.
- Encryption protocol: WeOS offers SNMPv3 data encryption using DES and AES-128.
- Authentication protocol: WeOS offers SNMPv3 data integrity using using MD5 and SHA1.
- Scope: A user can be restrained to only access a part of the MIB tree supported by the unit.

The encryption and authentication passwords are strings of 8-16 characters. ASCII characters 33-126 except '#' (ASCII 35) are allowed.

A maximum of 8 SNMPv3 users can be defined, each with their own parameter set.

6.1.4.1 SNMPv3 example

This example illustrates the configuration of an SNMPv3 user on the a WeOS switch. The user *alice* is grated *read-only* access to the full MIB tree. Security

level *authNoPriv* is used where SHA1 is used as authentication protocol.

Example

```
example:/#> configure
example:/config/#> snmp-server
example:/config/snmp/#> rouser alice auth sha1 alicepwd
example:/config/snmp/#> leave
example:/#> cp running start
```

Section 6.1.6 lists recommended SNMP management software. Those tools have graphical user interfaces and should be straight forward to use. For a simple test you could also use the (Unix) Net-SNMP **"snmpwalk"** command. (Here it is assumed that the switch is accessible on IP address *192.168.2.200* and the "walk" is limited to the mib-2 system's group).

Example

```
mypc:~$ snmpwalk -v3 -u alice -l authNoPriv -a SHA -A alicepwd 192.168.2.200 system
SNMPv2-MIB::sysDescr.0 = STRING: Westermo RedFox Industrial, primary: v4.4.0, backup: v4.
bootloader: v2.01, fpga: v20080626
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.16177
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (94018) 0:15:40.18
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: redfox
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 79
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
mypc:~$
```

6.1.5 Supported MIBs

6.1.5.1 Standard MIBs

As of WeOS v4.17.1 the following standard MIBs are supported:

- RFC1213 MIB-2: The original MIB-2 standard MIB.
- RFC2863 Interface MIB: The *ifXTable* of the IF-MIB is supported.
- RFC2819 RMON MIB: RMON Ethernet statistics (*etherStatsTable*) is supported.
- RFC4188 Bridge MIB
- RFC4318 RSTP MIB
- RFC4363 Q-BRIDGE MIB: The *dot1qVlan* group and *dot1qVlanStaticTable* are supported, enabling support for static VLAN configuration.

- RFC4836 MAU MIB: The *dot3IfMauBasicGroup* and *dot3IfMauAutoNegGroup* of the MAU MIB are supported.
- RFC3635 Ether-like Interface MIB: The *dot3StatsTable* is supported, enabling monitoring of various error counters for Ethernet ports.
- RFC4133 Entity MIB: The *entityPhysical* group of the Entity MIB is supported. It can be used to read unit serial number, firmware version, etc.
- RFC3433 Entity Sensor MIB: The Entity Sensor MIB can be used to monitor the status of unit sensors for temperature, power supply, and ”digital-in”, etc.
- RFC 4319 HDSL2/SHDSL MIB: On products with SHDSL ports, the *hdsl2ShdslSpanConfTable*, *hdsl2ShdslSpanStatusTable*, *hdsl2ShdslInventoryTable* and *hdsl2ShdslSpanConfProfileTable* are supported (read-only).
- RFC 3621 Power Ethernet MIB: The PoE MIB is supported on products with PoE ports.
- IEEE 802.1AB LLDP MIB
- RFC2787 VRRPv2 MIB: The *vrrpOperations* group is supported (read-only).
- RFC6527 VRRPv3 MIB: The *vrrpv3Operations* group is supported (read-only).

6.1.5.2 Private MIB

To use the WeOS private MIB, two Westermo specific MIB files should be loaded into your SNMP management software (see section 6.1.6 for information on recommended management software):

- WESTERMO-MIB: Defines the top level objects of the Westermo Private MIB name space.
- WESTERMO-WEOS-MIB: Defines the WeOS branch of the Westermo Private MIB.

6.1.6 Recommended Management Software

The following SNMP managers are recommended:

- OidView from ByteSphere[2].
- MG-SOFT MIB Browser Pro. from MG-SOFT[3].
- SNMPc from Castlerock Computing[4].

[2] http://www.oidview.com/oidview.html. OidView is a trademark of BYTESPHERE TECHNOLOGIES LLC.
[3] http://www.mg-soft.com/mgMibBrowserPE.html.
[4] http://www.castlerock.com/. SNMPc is a trademark of Castlerock Computing.

6.2 Managing SNMP via the web interface

Menu path: Configuration ⇒ SNMP

On the SNMP configuration page you will be presented to the current settings for SNMP on your switch, see below. You may change the settings by editing the page.

On the lower part of the page there is a list of SNMP v3 Users.

Enabled	Check the box to enable SNMP. If you have a JavaScript enabled browser the other settings will not be displayed unless you check this box.
Read Community	A community identifier for read access. Leave blank to disable read community.
Write Community	A community identifier for read/write access. Leave blank to disable write community.
Trap Community	A community identifier for traps. Defaults to community identifier **`trap`**.
Trap Host Address 1/2/3	IP address of SNMP trap management station. None, one , two or three addresses may be filled in. Leave all blank to disable SNMP traps.

6.2.1 Manage SNMP v3 Users

On the lower part of the SNMP configuration page you will be presented to the list of currently configured SNMP v3 users.

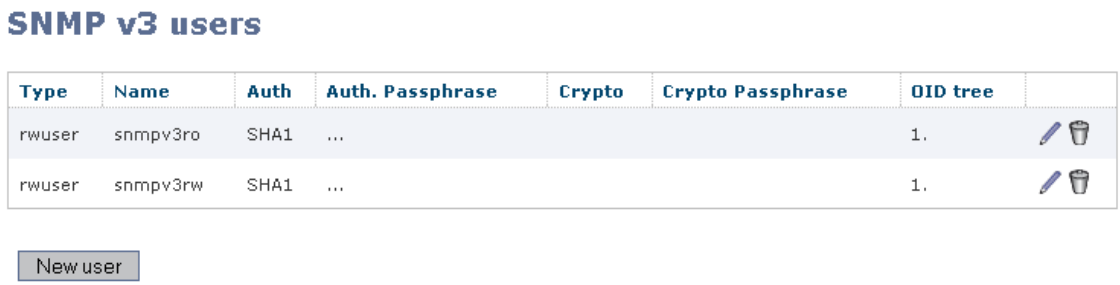


Figure 6.2: Listing of SNMP v3 users.

Type	Access rights for the user. **rwuser** User has read and write access. **rouser** User has read access only.
Name	A text string defining the user. Max 32 characters. Valid characters are ASCII 33-126 except '#' (ASCII 35).
Auth	Achieve message integrity protection by specifying MD5 or SHA1 message authentication.
Auth. Passphrase	The authentication password is a string of 8-16 characters. ASCII characters 33-126 except '#' (ASCII 35) are allowed.
Crypto	Achieve message privacy by specifying DES or AES128 message encryption.
Crypto Passphrase	The encryption password is a string of 8-16 characters. ASCII characters 33-126 except '#' (ASCII 35) are allowed.
OID Tree	Limit access to a certain branch of the supported MIB. Defaults to the whole tree ('1.')
Edit	Click this icon to edit the SNMP v3 user in that table row.
Delete	Click this icon to remove a the SNMP v3 user in that table row.
New User	Click on this button to create a new SNMP v3 user.

When clicking the *New User* button, the SNMP v3 user edit page will be displayed.

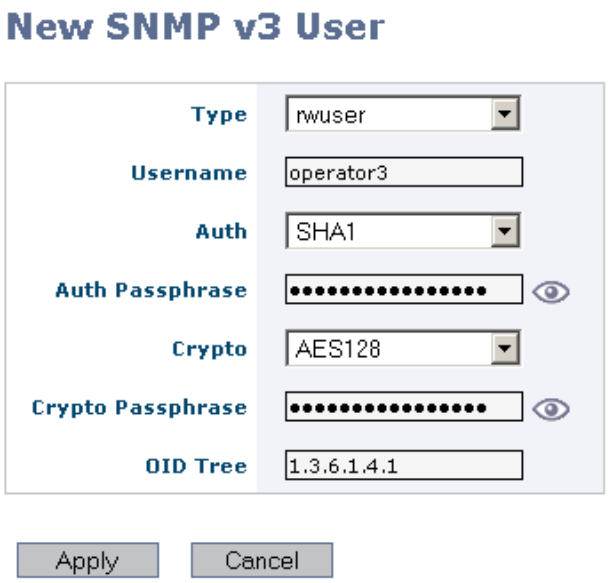


Figure 6.3: New SNMP v3 user.

See table above for description of fields.

6.3 Manage SNMP Settings via the CLI

Command	Default	Section
SNMP Server Configuration		
[no] snmp-server	Enabled	Section 6.3.1
[no] rocommunity <COMMUNITY>	public	Section 6.3.2
[no] rwcommunity <COMMUNITY>	Disabled	Section 6.3.3
[no] trapcommunity <COMMUNITY>	trap	Section 6.3.4
[no] host <IPADDR>	Disabled	Section 6.3.5
[no] rouser <USERNAME>	Disabled	Section 6.3.6
[auth <md5|sha1> <PASSPHRASE>		
[crypto <des|aes128> <PASSPHRASE>]]		
[OIDTREE]		
[no] rwuser <USERNAME>	Disabled	Section 6.3.7
[auth <md5|sha1> <PASSPHRASE>		
[crypto <des|aes128> <PASSPHRASE>]]		
[OIDTREE]		
SNMP Server Status		
show snmp-server		Section 6.3.8

6.3.1 Manage SNMP Server

Syntax `[no] snmp-server`

Context Global Configuration context.

Usage Enter SNMP Server Configuration context. If the SNMP server is disabled, it will be enabled when issuing the **”snmp-server”** command. Use **”no snmp-server”** to disable the SNMP server.

Use **”show snmp-server”** to show all SNMP server settings. (Also available as **”show”** command within the *snmp-server* context.)

Default values Enabled.

6.3.2 Manage SNMP Read Community

Syntax `[no] rocommunity <COMMUNITY_STRING>`

Context SNMP Server Configuration context.

Usage Configure the SNMP Read Community string. Use **"no rocommunity"** to disable the SNMP Read Community.

Use **"show rocommunity"** to show the SNMP Read Community setting.

Default values `rocommunity public`

6.3.3 Manage SNMP Write Community

Syntax `[no] rwcommunity <COMMUNITY_STRING>`

Context SNMP Server Configuration context.

Usage Configure the SNMP Write Community string. Use **"no rwcommunity"** to disable the SNMP Read Community.

Use **"show rwcommunity"** to show the SNMP Write Community setting.

Default values Disabled.

6.3.4 Manage SNMP Trap Community

Syntax `[no] trapcommunity <COMMUNITY_STRING>`

Context SNMP Server Configuration context.

Usage Configure the SNMP Trap Community string. **"no trapcommunity"** will reset the trap community to the default string (**"trapcommunity trap"**).

Use **"show trapcommunity"** to show the SNMP Trap Community setting.

Default values `trap`

6.3.5 Manage SNMP Trap Hosts

Syntax `[no] host <IPV4ADDRESS>`

Context SNMP Server Configuration context.

Usage Configure a SNMP Trap Host. Up to three trap hosts can be configured (issue the **"trap-host"** command multiple times with different IP addresses). Use **"no host <IPV4ADDRESS>"** to remove a trap-host and **"no host"** to remove all trap hosts.

Without any defined trap host, SNMP traps will not be sent.

Use **"show host"** to show the configured SNMP Trap Hosts.

Default values Disabled.

6.3.6 Manage SNMPv3 Read-Only User

Syntax `[no] rouser <USERNAME> [auth <md5|sha1> <PASSPHRASE> [crypto <des|aes128> <PASSPHRASE>]] [OIDTREE]`

Context SNMP Server Configuration context.

Usage Configure a SNMP read-only user.

- *USERNAME:* A text string defining the user. Max 32 characters. Valid characters are ASCII 33-126 except '#' (ASCII 35).
- *Authentication:* Achieve message integrity protection by specifying MD5 or SHA1 message authentication. The authentication password is a string of 8-16 characters. ASCII characters 33-126 except '#' (ASCII 35) are allowed.
- *Encryption:* Achieve message privacy by specifying DES or AES128 message encryption. The encryption password is a string of 8-16 characters. ASCII characters 33-126 except '#' (ASCII 35) are allowed.
- *OIDTREE:* Limit access to a certain branch of the supported MIB. Defaults to the whole tree ('1.')

Use **"no rouser <USERNAME>"** to remove a specific *read-only* user, or **"no rouser"** to remove all read-only users.

Use **"show rouser"** show settings for configured SNMPv3 read-only users.

Default values Disabled.

Examples

- Authentication and encryption:
  ”rouser alice auth sha1 alicepwd1 crypto aes128 alicepwd2”
- Authentication with access to dot1dBridge subtree:
  ”rouser bob auth md5 bobspwd1 1.3.6.1.2.1.17”

6.3.7 Manage SNMPv3 Read-Write User

Syntax `[no] rwuser <USERNAME> [auth <md5|sha1> <PASSPHRASE> [crypto <des|aes128> <PASSPHRASE>]] [OIDTREE]`

Context SNMP Server Configuration context.

Usage Configure a SNMP read-write user. For more information, see section 6.3.6.

Use **”show rwuser”** show settings for configured SNMPv3 read-write users.

Default values Disabled.

Examples See section 6.3.6.

6.3.8 Show SNMP server status

Syntax `show snmp-server`

Context Admin Exec context.

Usage Show whether SNMP server is running or not.

Examples

SNMP server enabled

Example

```
example:/#> show snmp-server
SNMP server running as PID: 540
example:/#>
```

SNMP server disabled (see **”no snmp-server”** in section 6.3.1).

Example

```
example:/#> show snmp-server
No SNMP server currently running
example:/#>
```

Part II

Common Switch Services

Chapter 7

General Switch Maintenance

7.1 Overview

The table below summarises maintenance features available for the different management tools. General descriptions of these features are presented in sections 7.1.1-7.1.10. If you are only interested in knowing how to manage maintenance features via the Web or CLI, please visit sections 7.2 or 7.3 directly.

Feature	Web	CLI	General Description
Firmware Upgrade			
Upgrade primary firmware	X	X	Section 7.1.1
Upgrade backup firmware	X	X	-”-
Upgrade bootloader		X	-”-
View firmware versions	X	X	-”-
Bootstrap Options			
Configuration File Media		X	Section 7.1.2.2
BOOTP Bootstrap Settings		X	-”-
USB Bootstrap Settings		X	-”-
Login Account management			
Set Admin Password	X	X	Section 21.1.1
Recover from lost Admin Password			Section 7.1.3

Continued on next page

Continued from previous page

Feature	Web	CLI	General Description
Configuration Files and Reboot			
Reset to Factory Default	X	X	Section 7.1.3
Reboot	X	X	Section 7.1.4
View Configuration Files	(X)	X	-”-
Alternate Configuration Files		X	Sections 7.1.4 and 7.1.5
Configuration Backup	X	X	Sections 7.1.4 and 7.1.5
Configuration Upload	X	X	Sections 7.1.4 and 7.1.5
Auto-Backup and Restore (USB)		X	Section 7.1.6
Configuration Deployment (USB)			Section 7.1.7
Virtual File System			
Maintenance of Configuration		X	Section 7.1.5
Log and USB files	(X)	X	-”-
Certificate and Key Management			
Upload PKCS#12 Bundle	X	X	Section 7.1.8
Upload PEM file	X	X	-”-
Public Certificate	X	X	-”-
Private Key	X	X	-”-
CA Certificate	X	X	-”-
Upload OpenVPN static key file	X	X	-”-
Set (non-default) Label	X		-”-
Controlling Management Services			
Enable/disable LLDP	X	X	Section 7.1.9
Enable/disable Web		X	
Enable/disable IPConfig		X	
Enable/disable SSH		X	
Enable/disable Telnet		X	
Enable/disable SNMP	X	X	(See chapter 6)
Maintenance and diagnostic tools			
Ping	X	X	Section 7.1.10
Traceroute	X	X	-”-

Continued on next page

Continued from previous page

Feature	Web	CLI	General Description
IPConfig Client	X	X	-”-
Port Monitoring	X	X	-”-
Wake-On-Lan	X	X	-”-
SSH Client		X	
Telnet Client		X	
Tech Support	X		
Other maintenance features			
Show System Environment Sensors	X	X	
Show System Uptime	X	X	
Show Memory Usage	X	X	
Show Running Processes		X	
Show Flash Table		X	
Update Flash Table[1]		X	-”-

7.1.1 WeOS Firmware

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 image 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.

[1] Ability to update the flash partition table is only available on early RedFox units (Industrial and Rail), where the flash partition table needs to be modified before upgrading to WeOS 4.3.0 or later. See section 7.1.11 for details.

> **Hint**
> It is strongly recommended to use the same system firmware *version* for the primary and backup image. Thereby you ensure that the backup firmware interprets the configuration file the same way the primary firmware does.

For information on how to keep the primary and backup firmware synchronised, see section 7.1.1.2.

- *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. As of WeOS v4.17.1, the *system firmware* can be upgraded via the Web or via the CLI, while the *bootloader* is only possible to upgrade via the CLI.

> **Warning**
> 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 for example from version 4.16.0 to 4.15.1, it is recommended to reboot the switch once the old firmware has been installed. When the switch comes up with the old firmware (here 4.15.1), copy the *factory default configuration* to the *running configuration*. See section 7.1.4 for more information on configuration files.

7.1.1.1 Upgrading firmware and bootloader

Firmware and bootloader for WeOS products can be downloaded from www.westermo.com.

The method to upgrade firmware and bootloader differs somewhat if the unit to upgrade is running WeOS 4.13.1 (or later), as compared to units running releases before 4.13.1.

- *Units running WeOS 4.13.1 or later:* The WeOS firmware and bootloader can be upgraded using a common ”pkg” file in WeOS 4.13.1 and later. This is explained further in section 7.1.1.1.2.

Note
WeOS releases older than 4.13.1 (e.g., WeOS 4.13.0 or 4.11.2) are unable to handle ”pkg” files.

- *Units running releases earlier than WeOS 4.13.1:* When upgrading WeOS units running older versions than WeOS 4.13.1 (e.g., WeOS 4.13.0 or 4.11.2), there are individual firmware and bootloader files per WeOS product. This is described in section 7.1.1.1.1.

Hint
If your unit is running a WeOS, e.g., WeOS4.12.0, and you wish to upgrade using a ”pkg” installation file (e.g., ”WeOS-4.14.0.pkg”) you first need to upgrade to WeOS 4.13.1 using the old method in section 7.1.1.1.1.

Hint
If the switch reports lack of free memory when trying to upgrade the firmware, try to disable non-essential services on the switch.

7.1.1.1.1 Upgrading when running older firmware than WeOS 4.13.1

Before WeOS 4.13.1 the firmware installation file to use differed per product family. Similarly, there were different bootloader installation files per product. A summary of name conventions is given in the table below:

Product	Primary and secondary FW	Bootloader FW
RedFox	rwXXXX.img (e.g., rw4112.img)	xscale-redboot-YYY.bin (e.g., xscale-2.03.bin)
Lynx and Viper	lwXXXX.img (e.g., lw4112.img)	imx27-redboot-ZZZ.bin (e.g., imx27-redboot-4.11.bin)
Wolverine	wwXXXX.img (e.g., ww4112.img)	” ”
Falcon	fwXXXX.img (e.g., fw4112.img)	” ”

If you run a release older than 4.13.1, and wish to upgrade to 4.14.0 or later, where only ”pkg” files are supported, you must first upgrade to 4.13.1 (or some

later 4.13.x release) using ”img” files[1].

Hint

Although any 4.13.x release from 4.13.1 and later can be used as intermediate release when upgrading to pkg files, it is recommended that you use the most recent 4.13.x release. See www.westermo.com for download of WeOS 4.13 releases.

Below there are examples showing how to upgrade the primary firmware to a WeOS 4.13 release with support for ”pkg” files (here ”4.13.4” is used) and bootloader via a FTP server (or TFTP server) at 192.168.3.10 on a WeOS Lynx unit.

- Upgrading primary firmware via CLI on a Lynx (before WeOS 4.13.1). Here we upgrade to WeOS 4.13.4 from a FTP server at 192.168.3.10.

  Example

  ```
  example:/#> upgrade primary 192.168.3.10 lw4134.img ...
  ```

- Upgrading bootloader via CLI on a Lynx (before WeOS 4.13.1). Here we upgrade the bootloader to ”imx27-redboot-4.11.bin” from a FTP server at 192.168.3.10.

  Example

  ```
  example:/#> upgrade boot 192.168.3.10 imx27-redboot-4.11.bin ...
  ```

7.1.1.1.2 Upgrading when running WeOS 4.13.1 (or later) If you have WeOS 4.13.1 or later installed, upgrading firmware or bootloader is simplified in the sense that the same installation file (a ”pkg” file) is used for all types of upgrades (bootfile or firmware) on any type of WeOS product.The table below lists the firmware used upgrade system firmware and bootloader.

Product Family	System Firmware (Primary/Secondary Image)	Bootloader Firmware
All WeOS products	WeOS-X.X.X.pkg (e.g., WeOS-4.17.1.pkg)	WeOS-X.X.X.pkg (e.g., WeOS-4.17.1.pkg)

[1] WeOS 4.13.1 and later 4.13.x releases are available both as ”img” and ”pkg” files, while only ”pkg” files are available from WeOS4.14.0 and onwards.

Thus, upgrading the primary (or secondary) system firmware image, or the boot-loader will be done using the same (pkg) installation file.

Note
If you use TFTP for upgrading with ”pkg” files, make sure your TFTP server supports large files as defined in RFC2347[22].

Note
Be aware that upgrade using TFTP may be much slower compared to the FTP or HTTP methods. This is of particular concern if the link you are transfering data through has high latency. Some examples are: ADSL/VDSL/SHDSL links, 3G/4G links or accessing via VPN tunnel.

This is an effect of how the TFTP protocol works. Every data block that is sent is ACKed by the other end, and the sender will wait for this ACK before sending the next piece of data. FTP and HTTP use TCP for transfer, and TCP has its sliding window algorithm that is much better suited for high latency scenarios.

An example calculation of approximate transfer time for a high latency link:
Let’s say the data is 50 Mbyte (PKG files are often larger than this) and the latency, or round-trip-delay, is: 50 ms.
The standard TFTP block size is 512 bytes.
50 Mbyte divided in 512 byte sized blocks means 102400 blocks.
This translates to 5120 seconds at 50 ms per block, or 1 hour and 25 minutes!

Below you find CLI examples to illustrate upgrading *firmware* and *bootloader* using ”pkg” files:

- *Upgrading firmware* via CLI: Here we upgrade the primary firmware to ’WeOS 4.17.1 from a FTP server (or TFTP server) at 192.168.3.10.:

  Example
  ```
  example:/#> upgrade primary 192.168.3.10 WeOS-4.17.1.pkg
  ...
  ```

- *Upgrading bootloader* via CLI: Here we upgrade to the bootloader from a FTP server (or TFTP server) at 192.168.3.10.):

Example

```
example:/#> upgrade boot 192.168.3.10 WeOS-4.17.1.pkg
...
```

Note

If your unit has an older version than WeOS 4.13.1 (e.g., WeOS 4.12.1), you are not able to upgrade using WeOS ”pkg” installation files directly. You first need to upgrade to WeOS 4.13.1 (or a later 4.13.x release) using the methods described in section 7.1.1.1.1.

7.1.1.2 Keeping Primary and Backup Firmware Synchronised

It is recommended to use the same version for primary and backup firmware. This ensures that your unit will have same functionality if it boots on the backup firmware as on the primary firmware.

Therefore, when upgrading the primary firmware, you are recommended to upgrade the backup firmware too. This section includes a 4-step example, where it is assumed you wish to upgrade the primary firmware on a WeOS unit from WeOS 4.13.4 to WeOS 4.14.1, i.e., from image **”WeOS-4.13.4.pkg”**[2] to **”WeOS-4.14.1.pkg”**.

1. *Prepare:* (This step is not necessary if you did steps 3 and 4 during an earlier upgrade, or if you have never upgraded your unit.)

   Before upgrading the primary firmware, check that the backup firmware is of the same version as the primary (here WeOS 4.13.4), and that the startup configuration file is matching the firmware version.

   (a) *Startup Configuration file matching current firmware version (here WeOS 4.13.4):* The simplest way to ensure that your startup configuration file is in-line with the current firmware version is to click an **Apply** ”button” in the Web (e.g., **Apply** in the IGMP configuration page, see section 18.2), or to run **”copy running-config startup-config”** in the CLI (see section 7.3.22).

   Note

   From WeOS 4.15.0 and onwards, this step is no longer necessary, as the startup configuration will then automatically be updated in-line with the current firmware version. See also section 7.1.4.

[2] WeOS 4.13.1 and later 4.13.x releases are available both in ”pkg” and ”img” format.

(b) *Verify that version of backup image is the same as the primary firmware:* To find out what firmware version you are using, see *Detailed System Overview* page in the Web (see section 4.4.2) or use the **"show system-information"** in the CLI (see section 7.3.2). In the example below the primary firmware version is 4.13.4 and the backup is 4.9.2.

Example

```
example:/#> show system-information

System Information
================================================================================

System Name         : example
System Contact      :
System Location     :
System Timezone     : Etc/UTC

Product Family      : Lynx              Model               : L210
Architecture        : mxc               Base MAC Address    : 00:07:7c:10:de:80
Article number      : 3643-0105-007     Serial Number       : 16975
Boot loader ver.    : 4.11              Active firmware     : Main
Main firmware ver. : 4.13.4             Backup firmware ver: 4.9.2
... (More info follows)
example:/#>
```

If the backup image is of a different version (as in the example above), you should upgrade the backup firmware (to WeOS 4.13.4) before moving to step 2. To upgrade the backup firmware (to WeOS 4.13.4), either use the Web upgrade facility, see section 7.2.1, or use the CLI **"upgrade"** command, see section 7.3.1. The example below shows an upgrade of the backup firmware from a FTP/TFTP server at 192.168.3.10.

Example

```
example:/#>  upgrade secondary 192.168.3.10 WeOS-4.13.4.pkg

==> Upgrade in progress, console disabled.  Please stand by ... <==

Connecting to 192.168.3.10:21 (192.168.3.10:21)
WeOS-4.13.4.pkg      100% |*******************************| 57747k  0:00:00 ETA

Checking download ...
Unpacking weos (from /upgrade/download)...
Setting up weos (4.13.4-1)...

Checking lw4134.img ...
   Type: CramFS
     ID: OK (Lnx2)
   Size: OK
    CRC: OK 0xDC73D8CD

Flashing /dev/mtd2 ...
100% - [=================================================================]

Updating RedBoot directory with new CRC ...
100%  [=================================================================]

Done.
example:/#>
```

2. *Upgrade primary:* To upgrade the primary firmware to WeOS 4.14.1, either use the Web upgrade facility (see section 7.2.1), or use the CLI **”upgrade”** command from the CLI (see section 7.3.1). E.g., use **”upgrade primary 192.168.3.10 WeOS-4.14.1.pkg”** to upgrade the primary firmware from a FTP/TFTP server at 192.168.3.10. Compare with the example in step 1b.

> **Note**
> As you are running your unit on a primary firmware, upgrading the primary firmware implies that the unit will automatically be rebooted when the upgrade finishes.

3. *Login and confirm configuration:* At the end of the upgrade process, the unit will reboot, using the new primary image if the upgrade procedure succeeded. After logging in again, do the following steps:

   (a) *Verify configuration:* Verify that the unit works as expected, doing whatever tests you find necessary for your use case. If the unit does not work as excepted, you should either consider downgrading to the previous version (here WeOS 4.13.4) or to inspect the running configuration to find and correct the cause of your problems.

> **Note**
> If you decide to downgrade, it is recommended to do that *before* changing or saving startup configuration for the new version (WeOS 4.14.1), as there are no general guarantees that the older WeOS version can interpret a later configuration file in exact the same way.

(b) *Make Startup Configuration file match the new firmware version (here WeOS 4.14.1):* (This is similar to step 1a, but now for the new firmware.) If the unit works as expected, store the configuration in-line with the new firmware (WeOS 4.14.1). The simplest way is to click an **Apply** ”button” in the Web (e.g., **Apply** in the IGMP configuration page, see section 18.2), or to run **”copy running-config startup-config”** in the CLI (see section 7.3.22).

> **Note**
> From WeOS 4.15.0 and onwards, this step is no longer necessary, as the startup configuration will then automatically be updated in-line with the current firmware version. See also section 7.1.4.

4. *Upgrade backup firmware:* The last step is to upgrade the backup firmware to the new WeOS version (here 4.14.1). For this you can use the Web upgrade facility, see section 7.2.1, or the CLI **”upgrade”** command, e.g., **”upgrade secondary 192.168.3.10 WeOS-4.14.1.pkg”** to upgrade the secondary firmware from a FTP/TFTP server at 192.168.3.10. Compare with the example in step 1b.

7.1.2 System bootstrap

During system bootstrap, the *bootloader* firmware is responsible for loading the *system* firmware. This is described further in section 7.1.2.1.

As part of the bootstrap, the WeOS unit is also capable of conducting a *cable factory reset* (section 7.1.3.3. The configuration is typically read from flash (startup-configuration file), but it is possible to retrieve the configuration from USB (section 7.1.6-7.1.7), or via BOOTP. Options for controlling these and other bootstrap related settings is covered in section 7.1.2.2.

7.1.2.1 Loading System Firmware (WeOS)

The bootloader attempts to load the *primary* system firmware image, with fall-back to loading the secondary system firmware if fails to load the primary firmware.

As described further below, different WeOS products use different bootloaders (Barebox, U-boot or RedBoot).

The Barebox bootloader enables you to stop the bootstrap process (from console port, press *Ctrl-C* at system startup), and enter an interactive *boot-menu*.

Example

```
Barebox Boot Menu
       1: Primary Partition
       2: Secondary Partition
       3: Network (BOOTP)
       4: System Recovery
       5: Shell
```

Access to the Barebox boot-menu can be password protected (section 7.1.2.2). From the boot-menu you can select which system firmware image (WeOS) to load (primary or secondary image on flash), but you can also choose to download a firmware remotely via TFTP into RAM, by entering the *rescue-mode* (System Recovery).

Note

As of WeOS v4.17.1, use of BOOTP in the Barebox boot-menu (alternative ”3.”) is a *technology preview*. Use of TFTP (rescue mode) or BOOTP is limited to Ethernet ports with ”internal PHY”; SFP ports can for example not be used.

Warning

Do not enter the bootloader shell (option ”3.”) unless you know what you are doing. Use of the bootloader shell is unsupported and can result in a broken unit.

If Barebox fails to load both the primary and secondary firmware, it will enter the rescue-mode, which you can access via the console port. As when entering rescue-mode from the regular boot-menu, you can download a new firmware into RAM via TFTP. Once the unit has booted, you can login and conduct a regular firmware upgrade (storing the firmware to flash).

In rescue-mode, Barebox also provides a rescue console service (UDP network console), which is useful if you do not have access to a console cable, or if your WeOS product lacks a console port. The rescue console can be accessed using any tool that can open a UDP socket, e.g., *netcat* on a Unix system **”nc -u -p 6000 192.168.2.200 6000”** if the default IP and UDP port numbers are used; this assumes your PC has IP address 192.168.2.1. Section 7.1.2.2 gives more information on configuration options related to the rescue console.

WeOS units run different types of bootloaders (Barebox, U-boot or RedBoot), and the boot-menu and rescue-mode features described above only apply to Barebox. The following bootloaders are used by different the different WeOS product platforms.

- Atlas: Products based on the Atlas use the *RedBoot* bootloader
- Basis: Products based on the Basis also use the *RedBoot* bootloader
- Corazon: Products based on the Corazon use the *U-boot* or *Barebox* bootloader. Barebox is supported from WeOS 4.15.2, and is now the *preferred* bootloader for Corazon products.

For information about what platform your product has, see section 4.4.2 (Web), or section 7.3.2) (CLI), or see the product list in section 1.5.

If you wish to check what *type* of bootloader (Barebox, U-boot or RedBoot) your unit runs, use the **”show partitions”** command as described in section 7.3.55.

See section 7.1.1.1 for information on how to upgrade your bootloader.

7.1.2.2 Bootstrap options

- *Configuration Boot Media:* WeOS supports two methods to retrieve configu-

ration file(s): from the on-board flash (default), from TFTP server (by use of BOOTP), and there are also options to deploy or restore configuration from a USB stick.

  - *Flash:* By default the WeOS unit boots using configuration files (startup-configuration, VPN certificates, etc.) from the (on-board) flash. The configuration on flash is also used as fall-back when other methods fail.
  - *BOOTP:* It is possible to bootstrap the configuration using BOOTP. For this you need a DHCP/BOOTP Server (section 22), and a TFTP Server, holding the unit's configuration file. As of WeOS v4.17.1, it is only possible to use BOOTP/TFTP to download the WeOS configuration file (certificates for IPsec, etc., can not be downloaded).

    Note
    Bootstrapping the configuration file using BOOTP is only possible over the WeOS unit's Ethernet ports. DSL ports (SHDSL, ADSL, VDSL) can not be used.

  - *USB:* It is possible to retrieve the configuration from a USB stick[3] by utilising WeOS USB Auto-Backup & Restore (section 7.1.6) or WeOS USB Deployment (section 7.1.6) functions[4]. These services have precedence over bootstrapping from Flash and BOOTP, but can be disabled (see USB Bootstrap Settings below).

- *BOOTP Bootstrap Settings:* When using BOOTP as configuration boot media, you can specify the BOOTP timeout (default 5 minutes), i.e., the maximum time to wait for the BOOTP/TFTP configuration file download to succeed. Fall-back is to use configuration on on-board flash.

  By default, the downloaded configuration file is only stored in RAM. You can manually store it to flash (e.g., by **"cp running-config startup-config"**), but you can also configure the WeOS to store the file to *startup-config* on flash automatically after download.

- *USB Bootstrap Settings:* During bootstrap, a WeOS unit checks if there is a USB stick attached in order to *restore* section 7.1.6) or *deploy* (section 7.1.6) a configuration from the USB stick.

[3] See section 1.5.1 for WeOS products with USB interfaces, and section 7.1.5.1 for list of USB sticks verified for use with WeOS.

[4] As a technology preview feature, there is also a boot media option referred to as "boot from USB". See WeOS release notes for more information on WeOS technology previews in general and for specific information on the "boot from USB" function.

– *Timings:* There are two timings related to Bootstrap and USB services:

  * *Delayed USB backup/restore and USB deploy:* (Non-configurable) A USB media not plugged in (or detected) when the device boots up can still be used to backup/restore or deploy the device configuration up to 30 seconds after power on.

  * *USB bootstrap timeout:* (Configurable) The USB bootstrap timeout halts boot for specified number of seconds, waiting for USB media to settle and be detected by the device. Before the timeout has elapsed and no media has been detected the device is unreachable with all ports remaining in blocking. Default: **Disabled** (i.e., zero delay)

Hint

Setting a ”USB bootstrap timeout” is useful to avoid a situation where the unit first applies the configuration from on-board flash, and afterwards detects the USB stick and applies USB *restore* or *deploy* (”Delayed USB backup/restore and USB deploy”).

– *Enable/Disable:* USB bootstrap services can be disabled. Disabling USB bootstrap services implies disabling USB Deployment and *automatic* USB Backup & Restore features. Manual backup and restore to/from a USB stick is still possible. Default: **Enabled**

Warning

USB bootstrap services are enabled by default for ease of use and robustness. However, it gives users with physical access to the switch the opportunity to modify or retrieve the configuration without logging in. If unauthorised personnel have physical access to the unit it is *recommended* to disable USB bootstrap services for security purposes.

Below is an example of how to disable USB Bootstrap services.

Example

```
example:/#> boot
example:/boot/#> usb
example:/boot/usb/#> no enable
example:/boot/usb/#> show
   Status        : Disabled
   Timeout       : Disabled
example:/boot/usb/#> leave
example:/#>
```

- *Barebox boot-menu options:* Boot options related to the Barebox boot-menu (boot-menu password, rescue console settings, etc.) are described in sections 7.3.15-7.3.20.

7.1.3 What to do if you cannot access your switch

Occasionally you may end up in a situation where you cannot access your switch:

- *Forgetting IP address*: If you have forgotten what IP address you assigned to your switch, you will no longer be able to access it remotely (Web, SSH, Telnet, SNMP). Section 7.1.3.1 presents different methods to find the IP address of your switch.

- *Forgetting password*: If you have forgotten the **admin** password you assigned to your switch, you should conduct either a *factory reset* or a *password reset*. Both alternatives require that you have *physical access* to the switch.

  - *Factory Reset*: By resetting the switch to the factory default setting the whole[5] switch configuration (including the **”admin”** password)) will be reset to its default values. That is, the **”admin”** password will be reset to **”westermo”**, thus enabling you to login again.

    The way to accomplish a factory reset may differ if the switch has a console port (section 7.1.3.2) or if it lacks a console port (section 7.1.3.3).

  - *Password Reset*: On switches with a console port there is a possibility to reset the **”admin”** password to its default value (**”westermo”**) without affecting the rest of the configuration, see section 7.1.3.2.

- *Misconfiguration*: You may also lose the ability to access your switch remotely (Web, SSH, Telnet, SNMP, WeConfig) due to *misconfiguration*, e.g., by disabling all Ethernet ports, or moving them to a VLAN where the switch has no IP address assigned. This case can be resolved by logging into the switch via the console port, and change the configuration appropriately via the CLI (see chapter 5 on information of how to access the CLI via the console port).

  However, if the switch does not have a console port, you may need to conduct a *factory reset* as described in section 7.1.3.3.

[5] Only configuration files on unit flash will be affected. Files on an attached USB stick (if present) will not be affected.

7.1.3.1 Discovering the IP address of your switch

The factory default IP setting enables you to access your switch via IP address 192.168.2.200, as well as via an address assigned via a DHCP server[6] (see table 7.4).

	Address	**Netmask**	**Gateway**
Primary IP address	Dynamic (DHCP)	(Dynamic)	(Dynamic)
Secondary IP address	192.168.2.200	255.255.255.0	Disabled

Table 7.4: Factory Default IP settings.

If you have forgotten what IP address you assigned your switch there are several methods to find it out:

1. *WeConfig (from PC)*: The WeConfig tool is designed to scan for (Westermo) switches on the local network. See the WeConfig User Guide[54] for details on how to use the WeConfig tool. This option is probably the simplest method to find the IP address of a switch, but will not work if the IPConfig service has been disabled on your switch (see section 7.3.46 for information on how to enable/disable IPConfig on your switch).
2. *IPConfig client (from switch)*: The WeOS CLI and the Web contain an IPConfig *client* scanning facility, thus if you are logged into a switch you are to scan for neighbour switches. As in the previous step, switches can only be discovered this way if they have the IPConfig *service* enabled.
3. *Via console port*: On switches equipped with a console port, the IP address of the switch can be found using the switch Command Line Interface (CLI). See chapter 5 for more information of how to use the CLI. (If you have forgotten the **admin** password, please see section 7.1.3.2).
4. *LLDP*: If LLDP is enabled (section 7.1.9), WeOS announces its presence (including its IP address) in LLDP messages. Thus, an LLDP client (or simply a network sniffer such as Wireshark[7]) can be used to discover the IP address of the switch.

In case you are not able to discover the IP address by any of these methods, conducting a factory reset will take the switch back to its original IP configuration

[6] In addition, the unit will autoconfigure itself with a *link-local* address in the *169.254.x.x* range, where 'x' is in interval 0-255. See section 19.2.6 for more information.
[7] Wireshark network protocol analyser, http://www.wireshark.org.

(as shown in table 7.4). See sections 7.1.3.2 and 7.1.3.3 for information on how to conduct a factory reset.

7.1.3.2 Password or Factory Reset via Console Port

For WeOS switches *equipped with a console port*, it is possible to conduct a *factory reset* or just a *password reset* using the special accounts (**factory** or **password**). *For security reasons*, these special accounts can *only be used via the console port*. For security hardening purposes, these two special accounts can be disabled in the device's boot context, in the CLI (see sections 7.3.10 and 7.3.11).

- Admin password reset: It is possible to recover from a lost **admin** password by using the following login and password from the console port. The **admin** password will be reset to its default value (**westermo**), and thereby enable you to login to the switch again.
  - Login: **password**
  - Password: **reset**
- Factory reset: It is possible to reset the switch to factory default settings by using the following login and password from the console port. The whole[8] switch configuration (including the **admin** password) will be reset to its factory default setting.
  - Login: **factory**
  - Password: **reset**

7.1.3.3 Factory Reset without using Console Port

There is a mechanism to conduct a factory reset without using the console port or being logged into the unit – this method is referred to as "cable factory reset".

Note
Depending on the type of product, cable factory reset is conducted by connecting *one pair* of Ethernet ports (single cable) **or** *two pairs* of Ethernet ports (two cables) as shown in the table below.

[8] Only configuration files on unit flash will be affected. Files on an attached USB stick (if present) will not be affected.

1. Power off the switch and disconnect *all* Ethernet cables (including copper and fiber cables) and DSL cables.
2. Connect one pair (or two pairs) of Ethernet ports as described in the table below. The ports need to be connected directly, i.e., **not** via a hub or switch. Use a *straight* cable - not *cross-over* cable - when connecting a port pair.

Product/Model	Ethernet Port Pair 1	Ethernet Port Pair 2
Falcon		
FDV-206-1D1S	port 1 ⇔ port 4	port 2 ⇔ port 3
Lynx		
L106/206-F2G	port 3 ⇔ port 6	port 4 ⇔ port 5
L110/210	port 3 ⇔ port 10	port 6 ⇔ port 7
Lynx-DSS		
L105/205-S1	port 1 ⇔ port 4	port 2 ⇔ port 3
L106/206-S2	port 1 ⇔ port 4	port 2 ⇔ port 3
L108/208-F2G-S2	port 3 ⇔ port 6	port 4 ⇔ port 5
RedFox Industrial		
All RFI models	port 1/1 ⇔ port 1/2	Not applicable
RedFox Industrial Rack		
All RFIR models	port 1 ⇔ port 2	Not applicable
RedFox Rail		
RFR-12-FB	port X1 ⇔ port X6	port X2 ⇔ port X5
Viper		
All Viper-12 models	port X1 ⇔ port X6	port X2 ⇔ port X5
Wolverine		
DDW-142	port 1 ⇔ port 2	Not applicable
DDW-142-485	port 1 ⇔ port 2	Not applicable
DDW-225/226	port 2/1 ⇔ port 2/4	port 2/2 ⇔ port 2/3

3. Power on the unit.
4. Wait for the unit to start up. Control that the ON LED is *flashing red*. The ON LED flashing indicates that the unit is now *ready* to be reset to factory default. You now have the choice to go ahead with the factory reset, or to skip factory reset and boot as normal.
   - *Go ahead with factory reset*: Acknowledge that you wish to conduct the factory reset by unplugging (one of) the Ethernet cable(s). The ON LED will stop flashing.

This initiates the factory reset process, and the unit will restart with factory default settings.

- *Skip the factory reset*: To skip the factory reset process, just wait for approximately 30 seconds after the ON LED starts flashing RED without unplugging (any of) the Ethernet cable(s). The switch will conduct a normal boot with the existing settings.

7.1.4 Configuration Files and Reboot

The system keeps three special configuration files:

- *Startup Configuration*: The configuration file used by the switch after system boot or reboot. The *startup configuration* is stored in non-volatile memory (flash)[9].

  Note
  From WeOS 4.15.0 and onwards, the startup configuration is verified to be in-line with the syntax of the current firmware version upon system boot. If there are deviations (which may be the case after a firmware upgrade), the startup configuration is automatically updated.

- *Running Configuration*: The configuration currently used by the switch. The running configuration is kept in volatile memory (RAM).

  The *running configuration* is identical to the *startup configuration* when configuration changes are made via the Web interface, the WeConfig tool or SNMP. That is, when using these methods to manage the switch, a change in the *running configuration* is immediately copied to the *startup configuration*.

  In contrast, when managing the switch via the CLI, configuration changes only affect the *running configuration*. Thus, to make CLI changes survive a reboot, you must explicitly copy the running configuration to the startup configuration.

- *Factory Default Configuration*: The system keeps a factory default configuration file. The factory default file is kept in non-volatile memory (flash) and cannot be overwritten. When the switch is shipped, and after factory reset,

[9] As described in section 7.1.5, it is possible to keep several configuration files on flash. The startup configuration file is actually a symbolic name for one of the stored configuration files.

the startup configuration file is identical to the factory default configuration file.

In addition to these configuration files, it is possible (via CLI) to keep a set of additional configuration files on the switch, which enables easy swapping between alternate configurations.

Warning

Configuring the switch via multiple management interfaces in parallel is discouraged, since it may lead to unexpected behaviour.
For example, consider the case when two users are accessing the switch at the same time, one user via the CLI and another user via the Web interface: Assume the ”CLI user” makes changes to the running configuration, but of some reason do not wish to copy these changes to the startup configuration (yet).
If the *another* user, the ”Web user”, applies a single change using the web management tool, all the changes done to the running configuration (by the ”CLI user”) will be saved to the startup configuration. (Actually clicking the **`Apply`** button, even without changing any values has the same affect.)

7.1.4.1 Account password when loading a configuration file

Configuration files contain information on user account and (hashed) passwords, e.g., for the **”admin”** account. Thus, when loading a configuration file to the switch (i.e., overwriting the *startup-configuration* or *running-configuration*), the account passwords will also be replaced according to the setting in the new configuration file.

Warning

To copy a new configuration file to the *running-config* or *startup-config* while keeping the existing user names and passwords, the lines in the new configuration file containing the **”username”** command should be removed before installing the new configuration file.

If you unintentionally happen to loose the *admin* password because you copied a configuration file including an unknown **admin** password, see section 7.1.3 for information on how to regain access to the switch.

7.1.5 Virtual File System

WeOS keeps various files of interest for the operator:

- Configuration files: By default there is only one configuration file (named *config0.cfg* stored on the switch. However, it is possible to create and keep multiple configuration files on the switch, both for backup purposes of for easy shifting between configuration setups. Configuration files are commonly named with the prefix *config* and will always have *'.cfg'* as extension.

  As mentioned in section 7.1.4 there are also three special configuration files:

  - *Running Configuration:* The running configuration is only stored in RAM, thus, it is not kept over a reboot.
  - *Startup Configuration:* The startup config is *mapped* to one of the stored configurations. By default it points to *config0.cfg*, but the mapping can be changed (using the CLI **"copy"** command as described in section 7.3.22).
  - *Factory Default Configuration:* The factory default configuration file cannot be modified (except through a firmware upgrade). Its available for the purpose of conducting a factory reset.

- Log files: Events are logged in various log files, e.g.:
  - auth.log
  - kern.log
  - messages
  - mgmt.log
  - snmpd
  - ppp.log

For units equipped with a USB port, the operator is also able to access files on a mounted USB stick.

The files are organised in a virtual file system, and are made available both for local and remote access.

	Local File Path	Remote File Path
Configuration files	cfg://	/cfg/
Log files	log://	/log/
USB files	usb://	/usb/

Section 7.1.5.1 gives general information on the use of USB memory sticks in WeOS products. Section 7.1.5.2 describes available methods for file maintenance when logged into the switch, while section 7.1.5.3 covers methods available for maintaining files remotely.

7.1.5.1 General information on using USB memory sticks

In order to copy files to/from a USB memory stick attached to USB port of the WeOS product[10], the USB memory stick *must*:

- be partitioned
- be formatted as VFAT or FAT32 on the first partition

As of WeOS v4.17.1 the following USB stick(s) are verified for use with WeOS products:

Westermo USB stick 3641-0190 (Serial number 1195 or higher)[50, 51, 52]

If a factory reset is conducted on the WeOS unit, only files on unit flash (configuration, IPsec certificates, etc.) will be affected by the factory reset. Files on an attached USB stick (if present) will not be affected.

7.1.5.2 File access when logged into the switch

An operator logged in to a switch can copy, download or upload files using the CLI **”copy”** command. Services available when logged into the system include:

- Making local backup copies of files, e.g.,
  ”copy log://messages log://messages.5”
- Upload or download to/from a remote server via TFTP, FTP, and SCP. (Downloading is also available via HTTP.)

[10] For information on WeOS products equipped with a USB port, see section 1.5.1, or the User Guide of your WeOS product (see section 1.5).

Upload example using TFTP:
”copy cfg://config0.cfg tftp://server.example.com/myswitchconfig.txt”

- Copying between systems: The CLI *copy* command can be used to copy files between remote systems via TFTP, FTP, SCP, and HTTP (HTTP can only be used as source, not destination).

  Example copying from HTTP server to TFTP server:
  ”copy http://server1.example.com/original.txt tftp://server2.example.com/backup.txt”

7.1.5.3 Remote file access

An operator is able to upload and download files to/from the switch remotely via *SCP*. This feature is convenient and saves time, since files can be maintained without the need to log into each switch.

Example with remote file upload:

Example

```
unix> scp config1.cfg admin@myswitch.example.com:/cfg/
Password for admin@myswitch.example.com:
unix>
```

Example with remote file download:

Example

```
unix> scp admin@myswitch.example.com:/log/messages .
Password for admin@myswitch.example.com:
unix>
```

7.1.6 Automatic Backup and Restore to/from USB

On WeOS units equipped with a USB port, a USB memory stick can be used for automatic backup and restore. The intended application for the auto-backup function is to **simplify unit replacement** in case of unit failure.

Once activated, it works seamlessly. If a stick is already prepared nothing else is needed. If a unit fails you simply replace it, moving the USB stick to the replacement unit. Which must be of same mark and model. At first boot, the replacement unit automatically restores all necessary files from the faulty unit.

> **Note**
> The auto-backup and restore function only handles configuration. It does **not** handle backup/restore of WeOS firmware images. You must not only ensure that your replacement unit is of the same model as the original unit. It should also have same WeOS firmware version loaded as the original unit.

Details of how to activate auto-backup, and how to perform restore are provided in sections 7.1.6.1-7.1.6.2. Section 7.1.6.3 contains information on USB directories for auto-backup and restore.

7.1.6.1 Procedure for activating auto-backup

- *Basic preparations the USB stick:* See section 7.1.5.1 for formatting and partitioning requirement for USB memory sticks used with WeOS units.
- *Insert USB stick:* Insert the USB stick into WeOS unit and power it up.
- *Log in to CLI:* Log into the unit (CLI), either via console port or remotely via SSH (see section 5.2).
- *Activate auto-backup:* Run the CLI **”backup”** command.

Example

```
example:/#> backup
WeOS Auto Backup & Restore for USB Media
================================================================================
This command initializes a USB media, usually a memory stick, to be used for
automatic backup and restore of configuration files (including certificates).

Intended use-case is to have one memory stick for each device in the network
to ease replacement of faulty units.

The replacement WeOS unit will at boot automatically restore the backup and
seamlessly pick up where the faulty unit left off.

Configuration and certificate files, including private keys (!) are backed up
to /usb/westermo/backup/

Activate WeOS auto-backup & restore on this USB stick, are you sure (y/N)? y
Performing initial backup...
Backup done.
example/#>
```

The configuration files (including certificates and private keys) are now backed up to sub-directories under **"/usb/westermo/backup/"** (see section 7.1.6.3).

- *Keep USB inserted:* The USB memory stick should stay attached to the WeOS unit. Any changes to the configuration files on unit flash will be continuously backed-up to USB.

An alternative method to initialise auto-backup is to create the (empty) directory on the USB stick */westermo/backup/* (see section 7.1.6.3) before inserting it to the WeOS unit. When attached, either when inserting it, or when the unit is powered up, all configuration files (including certificates and private keys) will be backed up on the USB automatically.

7.1.6.2 Restoring configuration from USB to replacement unit

When booting a WeOS unit checks if a USB stick is attached. If a USB stick is found with *auto-backup* activated, the WeOS unit checks if a restore operation should take place or not. This automatic *restore* operation only takes place at boot-up (configuration file is copied from USB to on-board flash, and used as startup configuration), or within an interval of 30 seconds after boot-up. In the latter case, which can occur if the USB stick is not ready at system boot time, the WeOS unit starts with and runs the configuration on on-board flash for a short while; *restore* operation then updates both the startup-configuration and running configuration.

Note

While replacing a WeOS unit using the USB auto-backup and restore support, it is recommended that the unit is disconnected from the network (see step 5 in the procedure below), and therefore there should be no problem if the replacement unit runs with the configuration on the on-board flash for a short while. Still, if it is important that the restore operation takes place before the WeOS reads its startup configuration, an additional boot delay can be added (see section 7.1.2.2 as well as step 1 in the procedure below).

1. *Prepare replacement unit:* The replacement should be of the same model as the original unit (e.g., a Lynx L210-F2G should be replaced by another Lynx L210-F2G), and ensure that it has the same WeOS firmware version loaded as the original unit.

   Hint

   If you are unsure of what firmware version your original unit was running, you can inspect the configuration file on your USB stick – at the top of the configuration file used as **"startup-configuration"** you should see the WeOS version, e.g., WeOS 4.15.2.

   It is recommended that the replacement unit has **not** had the auto-backup feature activated already. If unsure, please do a factory reset[11] of the replacement unit before proceeding. Use either of the methods described in section 7.1.3.2 (factory reset via console port), section 7.1.3.3 (cable factory reset), or section 7.2.4 (factory reset via web interface).

   Optionally, you can then login to the replacement unit and set a *USB delay* in the *boot* context. For example, to extend the time to discover a USB stick at boot with up to 10 seconds, use the following commands:

   Example

   ```
   example:/#> boot
   example:/boot/#> usb
   example:/boot/usb/#> timeout 10
   ```

   This gives the USB stick more time to settle at boot, and be ready for use when configuration is activated (see remark at the start of this section). Suitable USB delay differs depending on what WeOS product you are using

[11] Only files on unit flash (configuration file(s), IPsec certificates, etc.) will be affected by the factory reset. Files on an attached USB stick (if present) will not be affected.

(boot time differs) and what USB stick you are using (see section 7.1.5.1 for information on USB sticks verified for WeOS).

2. *Unplug power of replacement unit:* Before inserting the USB memory stick holding the backup configuration you should unplug the power of the replacement unit.
3. *Insert USB stick in replacement unit*
4. *Power up the replacement unit:* When the replacement unit boots, the configuration files on USB will automatically be restored to unit flash.
5. *Connect network cables:* It is recommended to connect the network cables *after* powering up the replacement unit. You may also connect them *before* powering up the unit (see comments on timings for detecting USB stick at the start of this section).
6. *Keep USB attached:* The USB memory stick should be stay attached to the WeOS unit. Any changes to the configuration files on unit flash will be continuously backed up to USB.

The automatic restore operation is only done when booting the WeOS unit, or within 30 seconds after boot-up[12]. If the USB stick (holding backup information) is inserted into a running unit need to reboot the unit for the auto-restore operation to occur. Alternatively, you can run the CLI **"restore"** command to manually trigger it.

Example

```
example:/#> restore
Restore backup from USB stick and activate to running-config, are you sure (y/N)? y
Stopping DHCP/DNS Server ................................... [ OK ]
Starting DHCP/DNS Server ................................... [ OK ]
example:/#>
```

7.1.6.3 Backup files in USB directory tree

Backup files will be stored on the USB in the following directory tree.

```
/usb/
  +-- westermo/
       +-- backup/       <-- Automatic Backup & Restore directory
```

[12] The restore operation is **not** conducted if "auto-backup" is already activated on the WeOS unit **and** the "gen.id" counter on the USB and unit flash have the same value, see also section 7.1.6.3.

```
+-- cfg/     <-- Configuration files
+-- crt/     <-- Certificates and keys
```

Additional details: The **”/usb/westermo/backup/cfg/”** directory will contain some additional files: **”startup-config.lnk”** specifies which config file is used as **”startup-configuration”**, and **”gen.id”** contains a counter. The corresponding **”gen.id”** file on unit flash is incremented every time a change on unit flash is detected. For every change the unit flash is synchronised to USB.

During the boot procedure, the **”gen.id”** values on USB and unit flash are compared. If equal, it is assumed that the configuration files are synchronised (no restore conducted). This is the case when rebooting a unit with auto-backup activated.

7.1.7 Configuration Deployment via USB

The *USB configuration deployment* function can be used for several purposes:

- *Easy configuration deployment of one or more WeOS units:* The USB stick is only attached during unit configuration, and can then be moved to the next unit to be configured.
- *To ensure a WeOS unit always boots up with a pre-defined configuration:* In this case, the USB stick will always be attached to the WeOS unit. The configuration on USB is copied to unit flash on every boot.

> **Note**
> For this use case, you may consider setting a boot delay (section 7.1.2.2) to avoid the risk that your unit starts with and temporarily uses the configuration on the on-board flash, see below for more explanations.

This ”USB configuration deployment” function differs from ”USB auto-backup and restore” described in section 7.1.6 in that configuration changes applied after boot only apply to the WeOS unit’s on-board flash – the configuration files on the USB memory stick are not affected.

- The model and WeOS version of the unit to be configured should match the intended configuration file(s) on the USB memory stick.
- The USB memory stick (prepared for deployment) is inserted before the unit is powered up. When the unit boots up configuration files will be copied from

USB to unit flash, and used during startup configuration.

- The deployment function is also automatically activated if a USB stick (prepared for deployment) is detected up to 30 seconds after boot-up. In the latter case, which can occur if the USB stick is not ready at system boot time, the WeOS unit starts with and runs the configuration on on-board flash for a short while; *deployment* operation then updates both the startup-configuration and running configuration.

  Note
  To prohibit that the unit first boots using configuration stored on the unit's on-board flash, you can setting a boot-delay (e.g., **"boot wait 10"** to extend the boot time with 10 seconds). By setting the delay large enough, the USB stick gets enough time to be ready when startup configuration is applied. Suitable boot delay differs depending on what WeOS product you are using (boot time differs) and what USB stick you are using (see section 7.1.5.1 for information on USB sticks verified for WeOS)

- The *USB configuration deployment* function is activated if the directory *"westermo/deploy/"* is detected on an attached USB during boot-up. USB configuration deployment has *precedence* over USB auto-backup and restore. That is, if the USB memory stick contains both a *"westermo/deploy/"* and a *"westermo/backup/"* directory, the configuration deployment function will be activated.

  Section 7.1.7.1 provides information on the file structure and format of the files in the *"westermo/deploy/"* directory.

7.1.7.1 Deployment files in USB directory tree

Deployment configuration files should reside on the USB in the following directory tree.

```
/usb/
  +-- westermo/
       +-- deploy/                        <-- USB Deploy
            +-- cfg/
            |    +-- <FILE>.cfg           <-- Actual configuration file, e.g., config0.cfg
            |    +-- startup-config.lnk   <-- Windows style .lnk file
            +-- crt/
                 +-- ...                 <-- Certificates and keys
```

The *startup-config.lnk* file holds the file name of the startup configuration file. The format of this file is:

- No leading directories, to avoid any / or \ confusion
- No end-of-line after file name, to avoid any DOS/UNIX/Mac confusion
- File name stored at first position in file, e.g., *config0.cfg*

As of WeOS v4.17.1 there is no CLI or Web function for setting up a USB configuration deployment memory stick for use with WeOS. Meanwhile the easiest way might be to

1. perform a USB auto-backup (see section 7.1.6.1), and
2. plug the USB stick into a PC and rename the *backup* directory to *deploy*.

7.1.8 Certificate and Key Management

WeOS supports upload and management of certificate and key files. As of WeOS v4.17.1, use of certificates is limited to IPsec VPNs and SSL VPNs (OpenVPN), see chapters 35 and 36.

It is possible to upload/import PKCS#12 bundles containing *public certificate*, *private key* and the certificate of the issuing certificate authority (*CA certificate*). The PKCS bundle can be password protected (recommended).

It is also possible to upload individual certificate files in PEM format or OpenVPN static key files. For further information on certificate management, see sections 7.2.6 (Web) and 7.3 (CLI).

7.1.9 Managing LLDP

The Link Layer Discovery Protocol (LLDP) is a standardised layer 2 protocol (IEEE 802.1AB[12]), which advertises information about the device itself and its capabilities to other devices within a LAN. 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[12]), which can be used by NMS-systems to draw a topology map of the network.

Examples of information advertised by LLDP:

- Remote port number
- Port capabilities
- IP address (see note below)
- Hostname
- MAC-address
- VLAN ID

In WeOS LLDP frames are advertised every 30 second. If an interface stops receiving frames, the neighbour information is expired after 120 seconds.

Note
The advertised IP address is the address of the ports default VLAN, see section 13.1.2.

Note
As of WeOS v4.17.1 LLDP is enabled/disabled globally for all ports.

7.1.10 Maintenance and diagnostic tools

The switch supports a set of maintenance and diagnostic tools:

Ping and Traceroute The standard Ping and Traceroute commands are available via the CLI and the Web, and are useful as basic troubleshooting tools.

Port monitoring The switch supports *port monitoring*, thus the user can monitor the traffic exchanged on one or more Ethernet ports on a dedicated monitor port. Only *correct* Ethernet packets will be forward onto the monitor destination port. To monitor occurrence of packet drops due to bad CRC, etc., we refer to the RMON statistics counters, see chapter 9.

Note
To observe all traffic on the monitor source ports, the total amount of traffic on the monitor source ports should not exceed the capacity of the monitor destination port.

WeOS IPConfig Client As mentioned in chapter 3 WeOS provides the *WeConfig PC tool* for discovery and rudimentary management of Westermo switches.

The CLI and the Web provides a similar mechanism (IPConfig client), i.e., once logged into the switch, it is possible to scan for other Westermo units on the same LAN.

Wake-On-Lan A Wake on Lan (WOL) client is available via the CLI and the Web. This allows a computer to be turned on or woken up by a network message (magic packet).

Additional features relevant for maintenance and diagnostics are described in chapter 9 (RMON Statistics), chapter 25 (Event and Alarm Logging), chapter 6 (SNMP), and chapter 24 (Alarm handling, Digital I/O and Front-panel LEDs).

7.1.11 Upgrading early RedFox Units to 4.3.0 or later

Early RedFox units (Industrial and Rail) delivered with WeOS 4.0.0, comes with a flash memory partition unsuitable for the larger firmware image size of WeOS 4.3.x[13] and later.

- **How to determine if your RedFox has an old partition table:**
  For RedFox Industrial, only products shipped with WeOS 4.0.0 came with the old partition table. You can determine if your product has the old partition table by inspecting the product's *model* (or the *article number*) and *serial number* – if the serial number is lower than the ones listed below, your product was shipped with the old partition table.

  You find information on your product's type of *model*, *article number*, and *serial number* via the Web interface (Menu path: Status ⇒ System, see section 4.4.2), or via the CLI **"show system-information"** command, see section 7.3.2).

Model	(Article number)	Serial number
RFI-18-F4G-T4G	3641-3300	< 1190
RFI-14P-F4G	3641-3200	< 1180
RFI-10P	3641-3110	< 1220
RFI-18P	3641-3100	< 1111

  If you are unsure whether your flash table is already updated, you can use the CLI **"show flash-table"** command available on WeOS 4.2.0 and later (see section 7.3.54) to list information on the flash partition table:

[13] WeOS 4.3.x refers to all patch releases (4.3.0, 4.3.1, . . .) of the WeOS 4.3 feature branch.

  - Main and backup partitions of size 12.5 MB (hex 0x00c80000) means the new partition table is used.
  - Main partition of size 8.5 MB (hex 0x00880000) and backup of size 7 MB (hex 0x00700000) means the old partition table is used.

- **Do you need to update your partition table?**
  It is possible to upgrade the *primary* firmware to WeOS 4.3.x even if your RedFox has an old partition table. If your RedFox has an old partition table, you must update it in order to:

  1. Upgrade your *backup firmware* (i.e., the firmware on the backup partition) to WeOS 4.3.x or later.
  2. Upgrade your *primary firmware* to a WeOS image larger than 8.5 MByte. The WeOS 4.3.x image for RedFox is below this limit, but later firmware versions (4.4.x and later) may be larger than 8.5 MB, and then the flash table needs to be updated.

- **How to update your flash table:**

  Warning
  Updating the flash partition table will corrupt your system! Configuration files, certificates and backup image will be destroyed.

  Although this update facility has been tested extensively by Westermo there are no guarantees it will work flawlessly for all use cases.

  Therefore, we do not recommend this action on active running units in the field. Instead, replace the unit with a spare one first.

  1. Backup your .cfg files, startup-config and any certificate files to a USB stick or remote (T)FTP/SCP server (see section 7.3.22).
  2. If you are running WeOS version 4.2.0 or later proceed directly to the next step. If you are running WeOS 4.0.0, you must first upgrade your primary firmware to a later release, e.g., 4.2.0 or 4.3.0 (see section 7.2.1 or 7.3.1).
  3. Access the CLI via the **console port**, run the **”`flash-table-update`”** command (see section 7.3.56), and wait for it to finish. The unit reboots when it has completed the update.

> **Note**
> The **"flash-table-update"** command is only available on WeOS 4.2.0 and later, and is only visible if you have a RedFox with the old partition table.

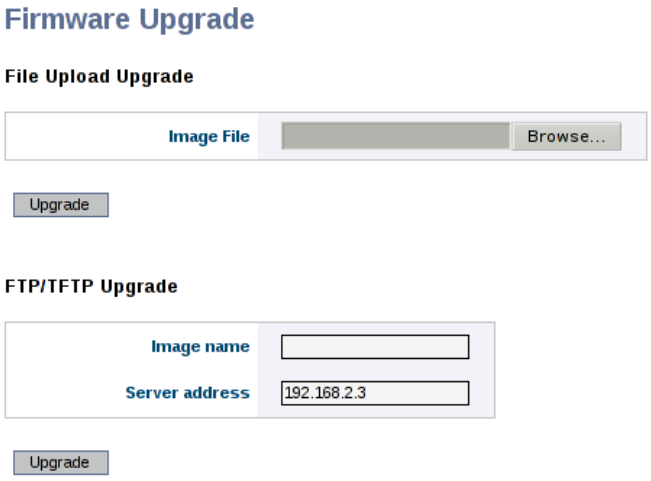
4. Restore configuration files, any necessary certificates and the system backup image.

7.2 Maintenance via the Web Interface

7.2.1 Managing switch firmware via the Web Interface

Menu path: Maintenance ⇒ F/W Upgrade

On the firmware upgrade page you are able to upgrade firmware by downloading an image using FTP/TFTP or by direct upload via the Web browser.



7.2.1.1 Firmware Upgrade Using File Upload

Image File	Select the file to upload (browser dependent).
Upgrade	Click the **Upgrade** button to initiate firmware upgrade.

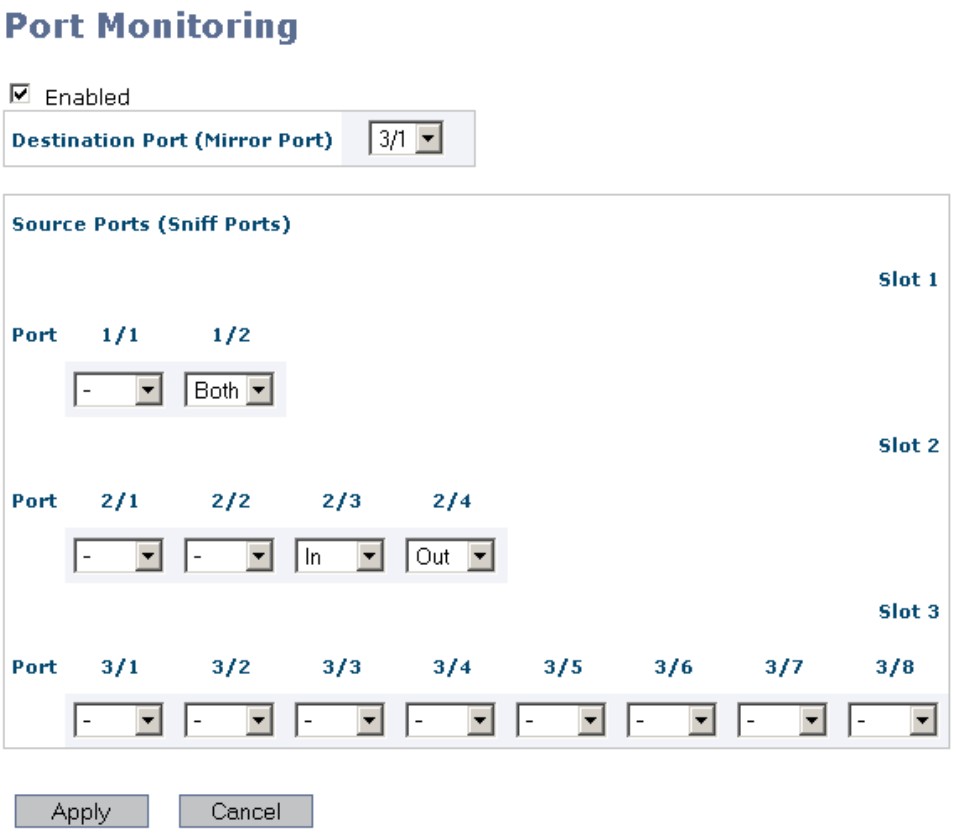
7.2.1.2 Firmware Upgrade Using TFTP/FTP Server

Image name	The file name of the image file on the FTP/TFTP server.
Server address	The IP address of the FTP/TFTP server.
Upgrade	Click the **Upgrade** button to initiate firmware upgrade.

Note
If you use TFTP for upgrading with ”pkg” files, make sure your TFTP server supports large files as defined in RFC2347[22].

7.2.2 Port Monitoring

Menu path: Tools ⇒ Port Monitoring

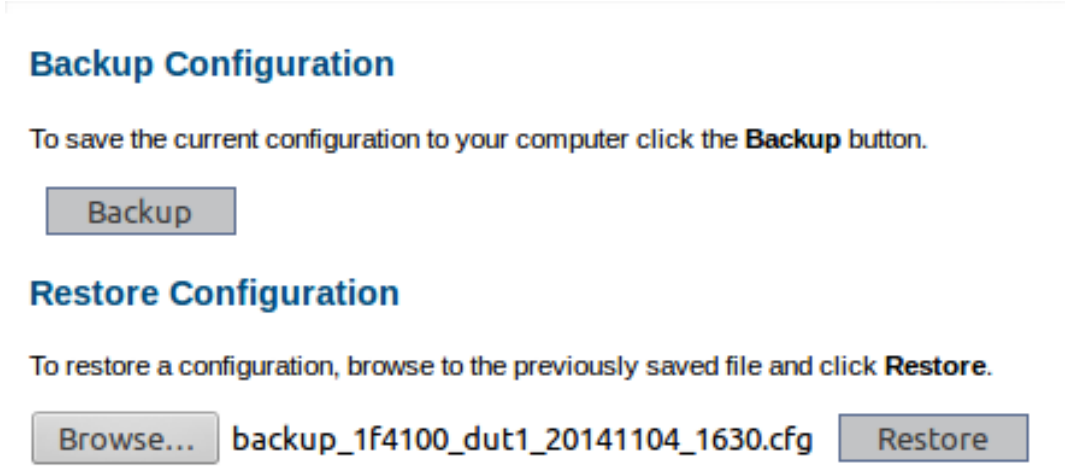


Enabled	Check the box to enable port monitoring. If you have a JavaScript enabled browser the other settings will not be displayed unless you check this box.
Destination Port (Mirror)	Select one port to which data from source ports will be copied (mirrored).
Source Ports (Sniff Ports)	Select one or more ports to monitor by selecting the ports desired sniff mode. Available modes are: **In** Inbound (ingress) traffic. **Out** Outbound (egress) traffic. **Both** Both inbound and outbound traffic.

7.2.3 Backup and Restore

Menu path: Maintenance ⇒ Backup&Restore

To create a backup of your switch configuration on your host, visit the *backup and restore* page.



Backup	Click this button to download a copy of the running configuration on your switch. You will be asked to open or save the file. Normally chose *save* to save the file to your host. The behaviour is web browser specific and may also depend on your current browser settings. See Fig. 7.1 for an example.
File Path	Click the **Browse** button to browse for the file. The behaviour of the file selection is browser specific.
Restore	Click this button to restore the configuration the configuration described in the file you selected in *File Path*.

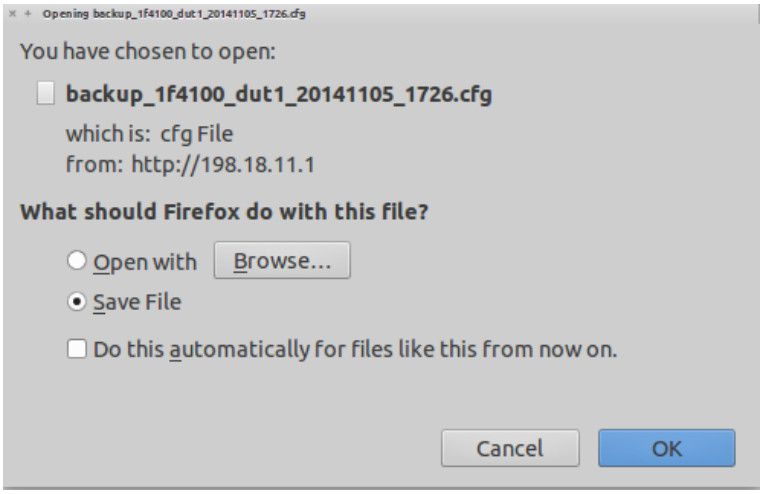


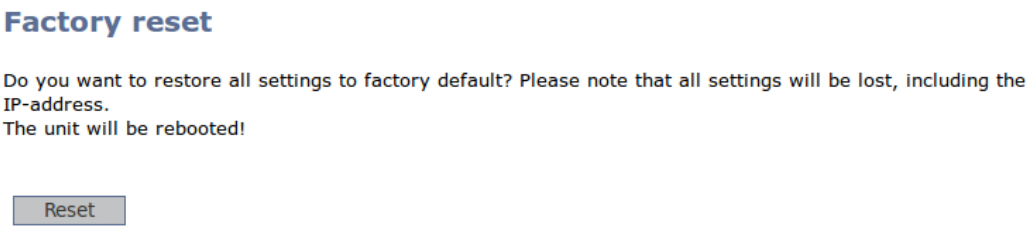
Figure 7.1: Example save dialogue (this example is from a Firefox browser)

7.2.4 Factory Reset

Menu path: Maintenance ⇒ Factory reset

To conduct a factory reset, press the *Reset* button.

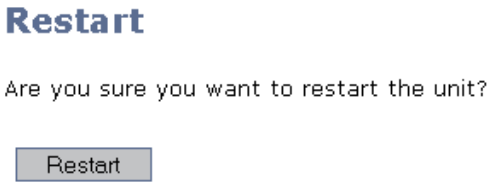
Only configuration files on unit flash will be affected by a factory reset. Files on an attached USB stick (if present) will not be affected.



7.2.5 Restart

Menu path: Maintenance ⇒ Restart

To restart the switch press the *Restart* button.



7.2.6 Managing certificates and keys

Menu path: Management⇒Certificates

When entering the certificates page you will be presented to a list of all certificates and keys available on your switch. Here you can import or delete certificates/keys.

Type	The type of certificate/key: **Public** (regular certificate), **Private** (a private key belonging to a regular certificate), **CA** (a CA certificate), or **OpenVPN** (an OpenVPN static key).
Label	A label identifying the certificate/key. Unique per certificate file type (Public, Private, CA and OpenVPN).

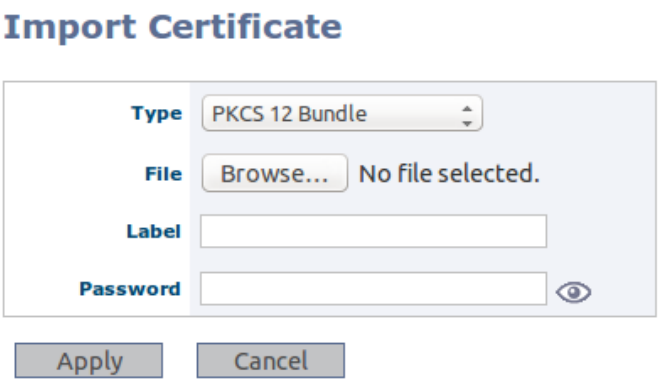
Continued on next page

	Continued from previous page
Common Name (CN)	The common name (CN) part of the distinguished name (DN) found in the imported certificate's subject.
Expires	The date of expiration for the certificate.
Delete	Click this icon to remove a certificate/key. You will be asked to acknowledge the removal before it is actually executed.
Details	Click this icon to display details regarding a certificate.
`Import`	Click this button to import a certificate or key.

7.2.6.1 Import Certificates

Menu path: Management ⇒ Certificates ⇒ **`Import`**

When clicking the **`Import`** button you will be presented to the certificate import page where you can import PKCS#12 certificate bundles, certificates and private key files in PEM format, or an OpenVPN static key.

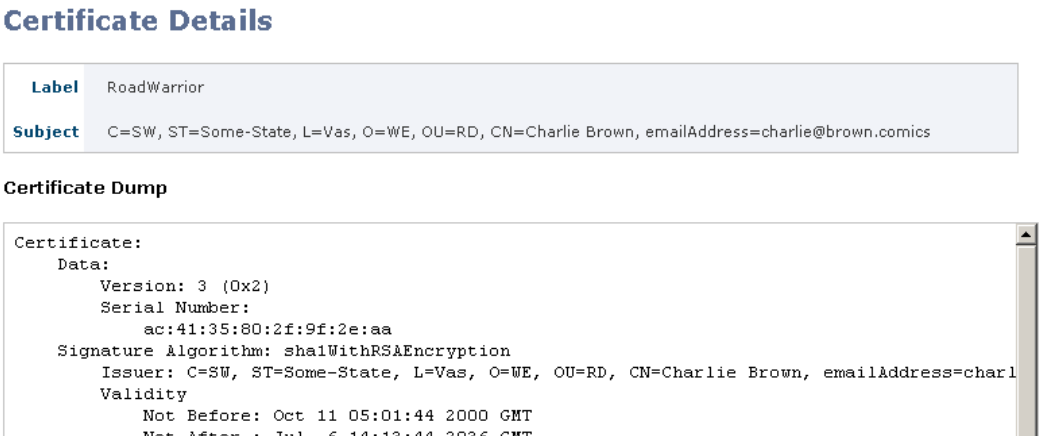


Type	Select the type of file to import (PKCS#12 bundle, PEM file or OpenVPN static key file).
File	Browse your file system for the file to import by clicking the **`Browse ...`** button.
Mode	(Only for PEM files) Declare the type of PEM file to upload: **`Public`** (regular certificate), **`Private`** (a private key), or **`CA`** (a CA certificate).
	Continued on next page

	Continued from previous page
Label	Enter a label for identification of the certificate/key. The filename (base part) will be used as label if left empty. E.g. if uploaded file name is *mycert.p12*, the label will be *mycert*
Password	(Only for PKCS#12 bundles) If your certificate bundle is password protected, you have to enter the password or the import will fail.

7.2.6.2 Certificate Details

Menu path: Management ⇒ Certificates ⇒ 



Label	A unique label identifying the certificate.
Common Name (CN)	The common name (CN) part of the distinguished name (DN) found in the imported certificate subject.
Certificate Dump	A raw dump of the certificate.

To exit the details page, select a menu option in the navigation menu.

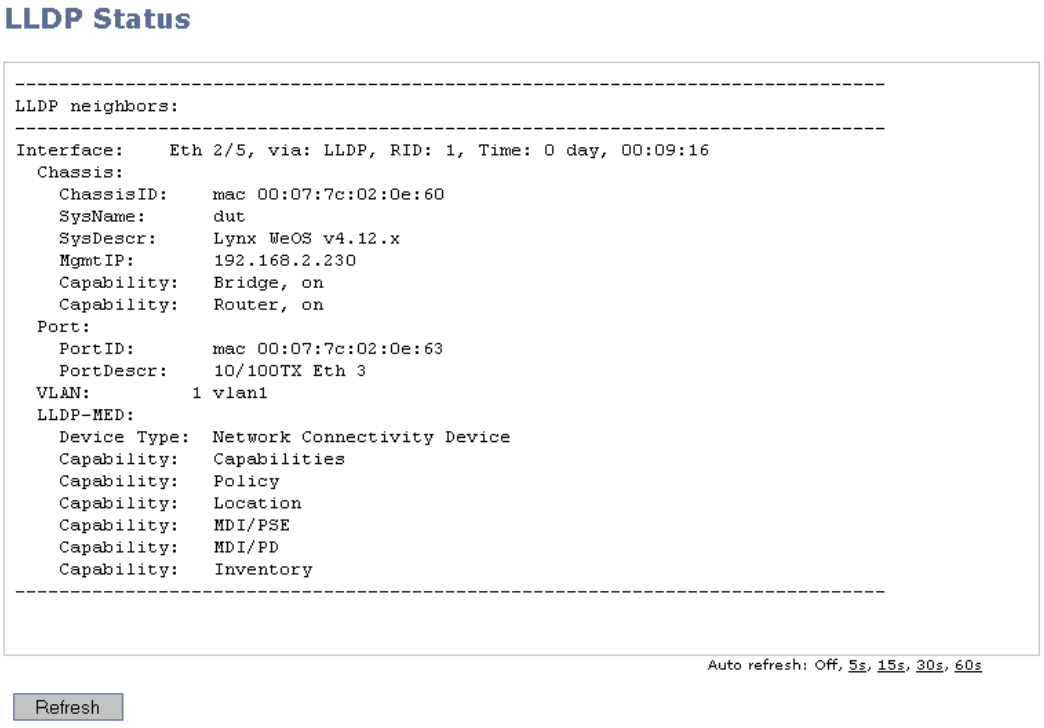
7.2.7 Enable/disable LLDP via the web interface

Menu path: Configuration ⇒ LLDP

Enabled	Check this box and click **Apply** to enable LLDP support on the unit.

7.2.8 Show LLDP Status via the web interface

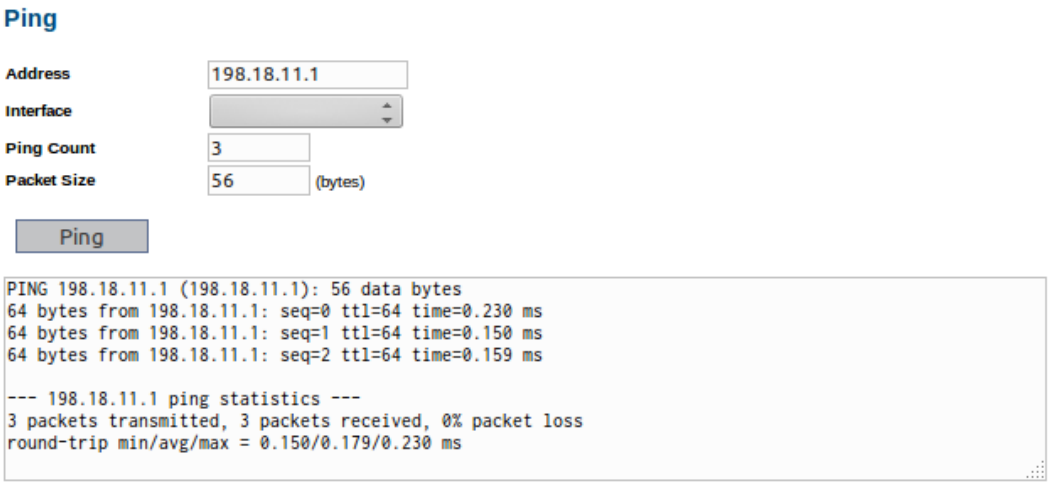
Menu path: Status ⇒ LLDP



7.2.9 Ping tool

Ping is useful as a basic diagnostic tool. The output on the web is displayed once the ping command has completed. If the command takes too long to execute the web page may time out.

Menu path: Tools ⇒ Ping

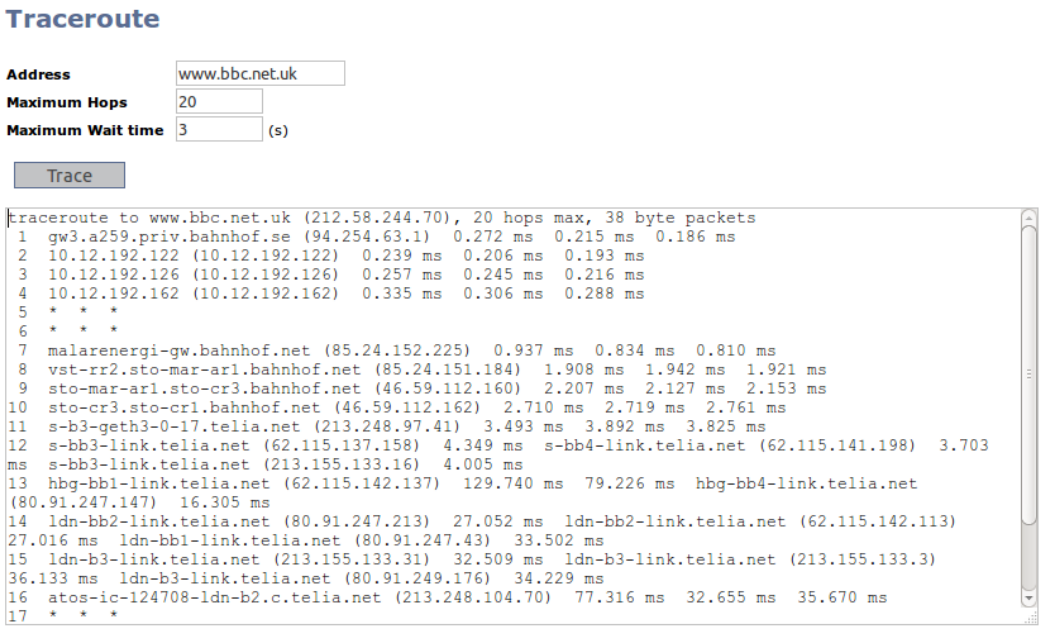


Address	The network host to send ICMP ECHO REQUEST packets to
Ping Count	Defines the number of ICMP packets to send.
Packet Size	Alters the default size of the ICMP packets. This only only increases the empty payload of the packet

7.2.10 Traceroute tool

Trace the route packets take to a network host. The output on the web is displayed once the ping command has completed. If the command takes too long to execute the web page may time out.

Menu path: Tools ⇒ Trace

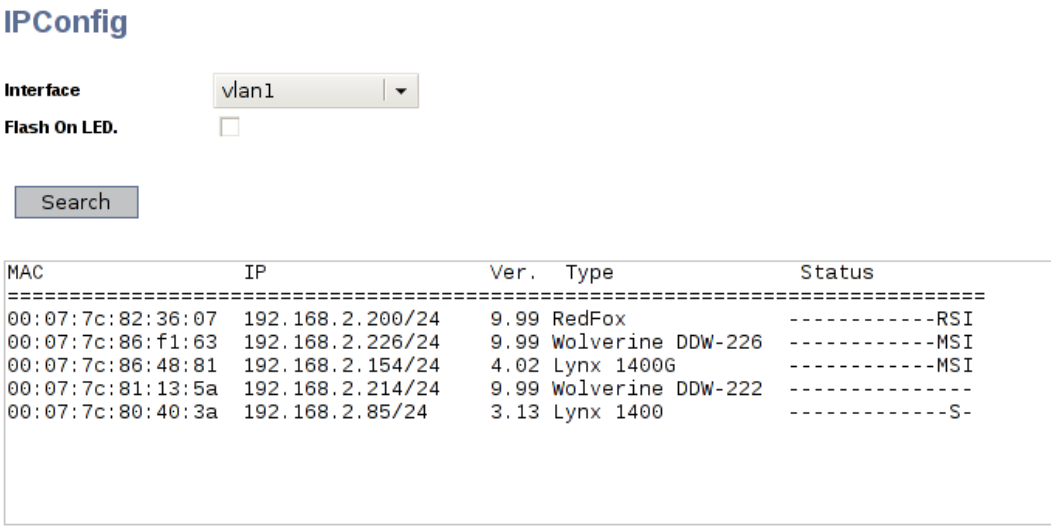


Address	The network host
Maximum Hops	Max time-to-live (number of hops).
Maximum Wait time	Set the delay, in seconds, before timing out a probe packet

7.2.11 IPConfig scan tool

Scan network for IPConfig neighbours. The output on the web is displayed once the ping command has completed. If the command takes too long to execute the web page may time out.

Menu path: Tools ⇒ IPConfig

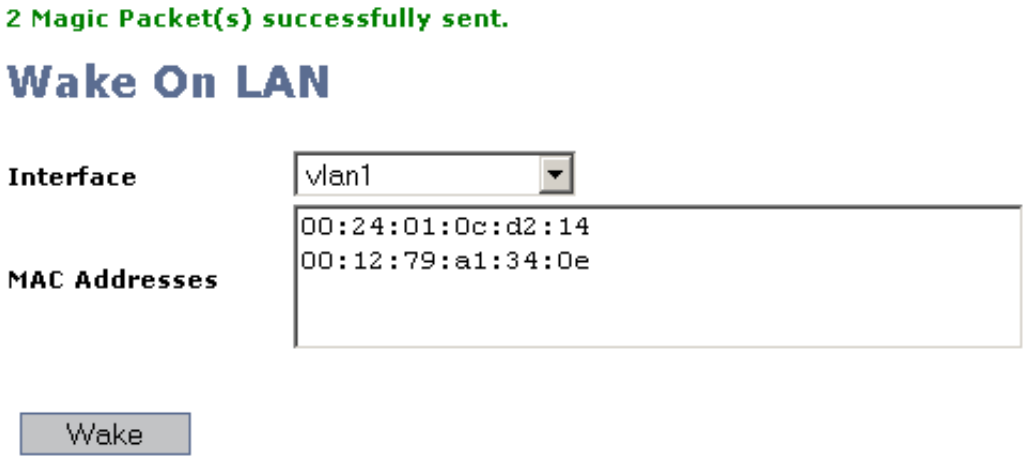


Interface	The interface to scan
Flash On LED.	If enabled, this unit will flash the on LED, while scanning

7.2.12 Wake on Lan

The Wake on Lan (WOL) allows computers to be turned on or woken up by a network message (magic packet).

Menu path: Tools ⇒ WOL



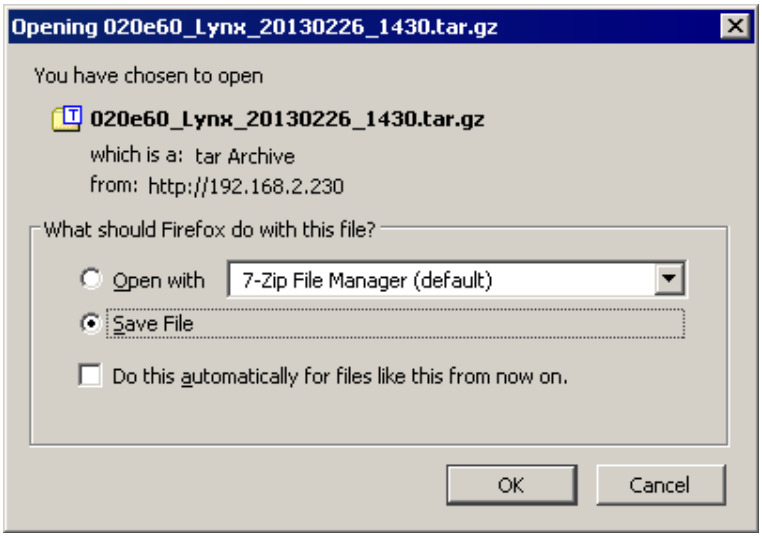
Interface	The interface to send the magic packet on.
MAC Addresses	The MAC Addresses of the computers to wake

7.2.13 Tech support

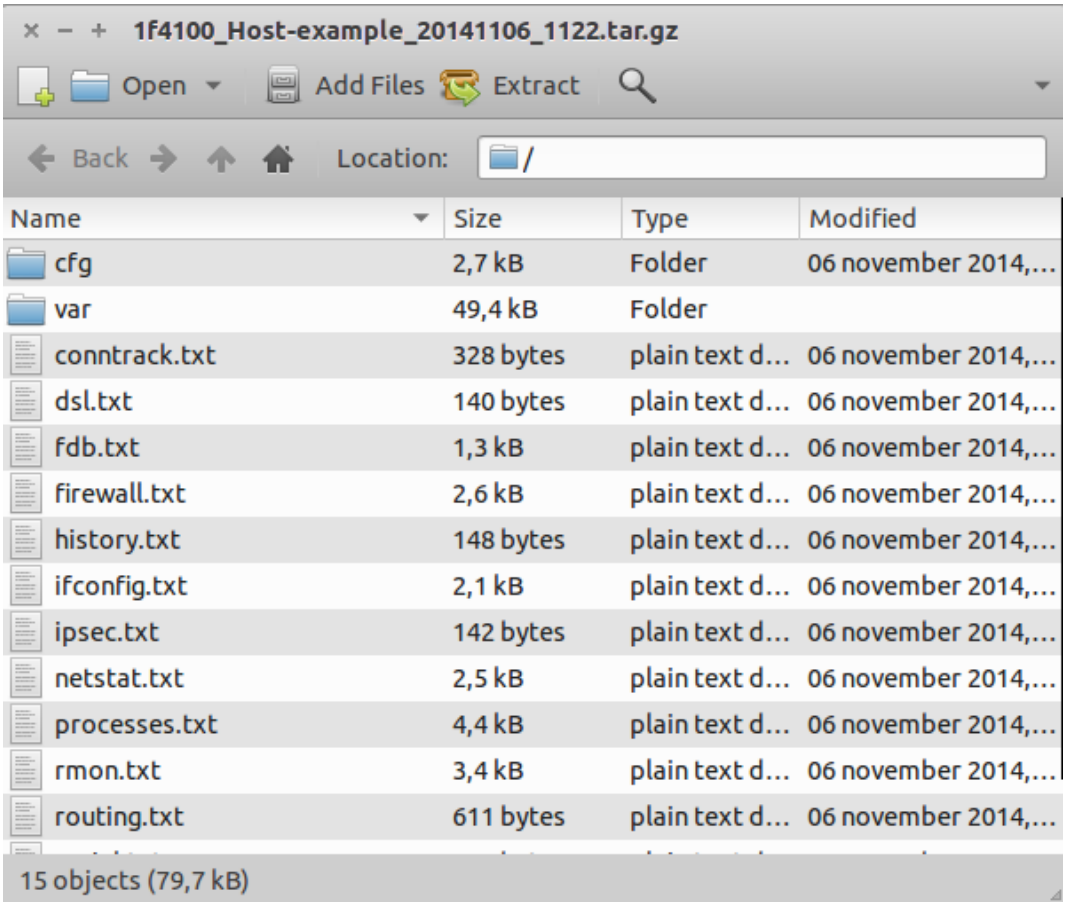
The Tech support collects system information (hardware, status and configuration) and delivers it as a compressed file. Note: The configuration is included with passwords. The file format is compressed tar archive(tar.gzip).The filename has the format of
<LOCATION>_<HOSTNAME>_<YYYYMMDD>_<HHMMSS>.tar.gz, if the location field is not set, the last three octets of the mac-address will be used.

Menu path: Tools ⇒ Tech Support

Clicking **Create** will create a Tech support file. Once the file is created you will be presented with the following dialogue.



The Tech support file consist of a number of text files. Configuration files can be found in the /cfg directory of the archive, and log files under the /var/log subdirectory.



7.3 Maintenance via the CLI

Command	Default	Section
Firmware Upgrade		
upgrade <pri|sec|boot> <IPADDR FILENAME | URI://. . . >		Section 7.3.1
show system-information		Section 7.3.2
System Boot Options		
boot	N/A	Section 7.3.3
[no] boot-order <flash|bootp>	Flash	Section 7.3.4
[no] bootp	N/A	Section 7.3.5
[no] timeout <0-1800>	300	Section 7.3.6
[no] mac <offset <num> | address <MACADDRESS>>	offset 1[14]	Section 7.3.7
[no] vfs-target <flash|usb>	Disabled	Section 7.3.8
[no] console	N/A	Section 7.3.9
[no] password-reset	Enabled	Section 7.3.10
[no] factory-reset	Enabled	Section 7.3.11
[no] usb	N/A	Section 7.3.12
[no] enable	Enabled	Section 7.3.13
[no] timeout <1-60>	Disabled	Section 7.3.14
[no] loader	N/A	Section 7.3.15
[no] login <password|hash> <STRING>	Disabled	Section 7.3.16
[no] rescue-port <UDPPORT>	6000	Section 7.3.17
[no] rescue-address <IPADDR>	192.168.2.200	Section 7.3.18
[no] rescue-netmask <NETMASK>	255.255.255.0	Section 7.3.19
[no] rescue-peer <IPADDR>	192.168.2.1	Section 7.3.20
File handling (Configuration, Log, etc.) and Reboot		
dir <cfg:// | log:// | usb://>		Section 7.3.21
copy <FROM_FILE> <TO_FILE>		Section 7.3.22
erase <file>		Section 7.3.23
show <running-config | startup-config | factory-config | [<filesys>://]FILENAME>		Section 7.3.24

Continued on next page

[14] See command description for details and exceptions.

Continued from previous page

Command	Default	Section
backup		Section 7.3.25
restore		Section 7.3.26
reboot		Section 7.3.27
Certificate and Key Management		
cert import <pkcs|pem|ovpn> [. . .] <URI>		Section 7.3.28
no cert [force] [LABEL]		Section 7.3.28
show cert [LABEL]		Section 7.3.29
Maintenance and Diagnostic tools		
ping <IPADDR>		Section 7.3.30
traceroute <IPADDR>		Section 7.3.31
ssh [USER@]<IPADDR|DNAME>[/PORT]	admin/22	Section 7.3.32
telnet <IPADDR|DNAME> [PORT]	23	Section 7.3.33
show ipconfig <IFNAME>		Section 7.3.34
[no] monitor		Section 7.3.35
[no] enable	Disabled	Section 7.3.36
destination <PORT>		Section 7.3.37
source <PORTLIST>		Section 7.3.38
LLDP Management		
[no] lldp		Section 7.3.39
[no] enable	Enabled	Section 7.3.40
Show LLDP status		
show lldp		Section 7.3.41
Configure/View Management Service Settings		
[no] web	Enabled	Section 7.3.42
[no] session-timeout <TIMEOUT>	10 Min	Section 7.3.43
port <PORT>	80	Section 7.3.44
ssl-port <PORT>	443	Section 7.3.45
[no] ipconfig	Enabled	Section 7.3.46
[no] read-only	Disabled	Section 7.3.47

Continued on next page

Continued from previous page

Command	Default	Section
[no] ssh	Enabled	Section 7.3.48
[no] telnet	Disabled	Section 7.3.49
[no] snmp-server	Enabled	Section 6.3.1
Other maintenance commands		
date [[YYYY-MM-DD]hh:mm[:ss]]		Section 20.2.7
[no] timezone <TIMEZONE>		Section 20.2.5
show timezone [QUERY|SUBSTRING]		Section 20.2.8
show env		Section 7.3.50
show uptime		Section 7.3.51
show memory		Section 7.3.52
show processes		Section 7.3.53
show flash-table		Section 7.3.54
show partitions		Section 7.3.55
flash-table-update		Section 7.3.56

7.3.1 Upgrading firmware

Syntax `upgrade <pri|sec|boot> <IPADDR> <FILENAME>`
`upgrade <pri|sec|boot> URI://<ADDRESS>/PATH/<FILENAME>`

Context Admin Exec

Usage Upgrade primary, secondary, or bootloader firmware via FTP, TFTP or USB stick. In the first form, upgrade attempts to download and install *FILENAME* via FTP from a server at *IPADDR*. If no FTP server is available, the command tries to download the file using TFTP instead.

> **Note**
> If you use TFTP for upgrading with ”pkg” files, make sure your TFTP server supports large files as defined in RFC2347[22].

The second form uses a URI based format. The same format used in the copy command, not all URIs are supported though, only ftp://, tftp:// and

usb://. In the usb:// case there is of course no need to give an ADDRESS, and PATH is optional. Also, some units may not have a USB port.

In the second form of the command it is also possible use an Internet name (FQDN), instead of just an IP address. For this to work you need to have first setup a valid name server in the configuration.

Before the actual ”Flashing” starts, i.e. when upgrade is still downloading or checking the downloaded image CRC, it is possible to abort the upgrade using Ctrl-C (BREAK). However, once the actual flashing starts the BREAK signal, and other blockable signals, is completely disabled to prevent accidental destruction of the device partition and image contents.

After installing a *primary firmware*, the switch will automatically be rebooted. (More precisely: after installing a *primary firmware*, the switch will automatically be rebooted given that the system booted from the primary image. Similarly, after installing a *secondary firmware*, the switch will automatically be rebooted given that the system booted from the secondary image.)

Caution! Only conduct upgrades over a stable network connection. Ensure that the switch is not powered off while the downloaded firmware is being installed.

Default values N/A

Examples **”upgrade primary 192.168.1.1 WeOS-4.15.1.pkg”** will download and install a new primary image named *WeOS-4.15.1.pkg*, from FTP/TFTP server at *192.168.1.1*.

”upgrade boot 192.168.1.1 WeOS-4.15.1.pkg” will download and install a new bootloader image included in the pkg file (*WeOS-4.15.1.pkg*) from a FTP/TFTP server with *192.168.1.1*.

”upgrade pri usb://WeOS-4.15.1.pkg” upgrades primary firmware on a WeOS unit using pkg file *WeOS-4.15.1.pkg* present on a USB stick. Check if the USB stick has been *mounted* first using the **”dir usb://”** command.

7.3.2 Show System Information

Syntax `show system-information`

Context Admin Exec

Usage List general system information such as serial number, firmware version, contained hardware, etc.

Default values Not applicable

Example

```
example:/#> show system-information

System Information
================================================================================

System Name        : example
System Contact     :
System Location    :
System Timezone    : Etc/UTC

Product Family     : RedFox            Model              : RFIR-219-F4G-T7G-AC
Architecture       : mpc85xx           Base MAC Address   : 00:07:7c:15:5f:20
Platform           : Corazon           Class              : Extended
Article number     : 3641-4015         Serial Number      : 1037
Boot loader ver.   : 2014.06.0-1       Active firmware    : Main
Main firmware ver. : 4.15.2            Backup firmware ver: 4.15.2
Manufacturing date : Sep 24, 2014

Card #1  ======================================================================
Type               : CPU
Chipset            : MV88E6352 r1
Article no         : 5013-1010
Revision           : 0
Batch id           : 140915-01274960-00001
Channel interfaces : 2
Bandwidth limit    : Disabled (for CPU channels)
... (More info follows)
example:/#>
```

7.3.3 Manage Boot Options

Syntax `boot`

Context Admin Exec context

Usage Enter System Bootstrap context to configure device specific boot settings. These settings are stored separately, i.e., outside the regular config-

uration file.

Use **”show boot”** to view a summary of the boot option settings.

Default values N/A

Example

```
example:/#> show boot
Boot order      : flash
example:/#>
```

7.3.4 Set Boot Order

Syntax `[no] boot-order <flash|bootp|usb>`

Context System Bootstrap context

Usage Select Boot Order for *configuration file*[15].

As of WeOS v4.17.1 the **”boot-order”** has the following limitations:

- **”boot-order”** can only be used to select a single boot media, not a list. That is, you can select either **”flash”** or **”bootp”**, but not both.

  Note
  The WeOS unit will fall-back to find its startup-configuration from on-board flash when other methods such as **”bootp”** fails.

- The alternative **”boot-order usb”** (referred to as ”boot from USB”) is only available as *technology preview*. See WeOS release notes for more information on WeOS technology previews in general and for specific information on the ”boot from USB” function.

Use **”no boot-order”** to reset the boot-order to the default setting.

Use **”show boot-order”** to view the configured boot order. Flash will listed as second choice if **”boot-order bootp”** is set.

Default values Flash

[15] Future versions of WeOS may include support for boot order of software image files.

Example

```
example:/#> boot
example:/boot/#> show boot-order
flash
example:/boot/#> boot-order bootp
example:/boot/#> show boot-order
bootp, flash
example:/boot/#> end
example:/#>
```

7.3.5 Manage BOOTP Bootstrap Settings

Syntax `[no] bootp`

Context System Bootstrap context

Usage Enter System Bootstrap BOOTP context to configure settings for BOOTP boot services.

"no bootp" will reset the BOOTP bootstrap settings to default.

Use **"show bootp"** to list BOOTP bootstrap settings (also available as **"show"** command within the System Bootstrap BOOTP context.

Default values N/A

7.3.6 BOOTP timeout

Syntax `[no] timeout <0-1800>`

Context System Bootstrap BOOTP context

Usage Set timeout in seconds to wait for BOOTP server response.

If no BOOTP response is received from the BOOTP/DHCP server, new BOOTP Requests will be re-transmitted up to the given timeout interval.

To avoid congestion, the Requests are re-transmitted randomised around an exponential back-off interval; the back-off interval is doubled for each request up to 60 seconds.

The BOOTP client will wait one extra back-off interval after the last transmitted request, thus the actual timeout can be roughly 60 seconds longer than configured.

Use **”no timeout”** to reset the timeout to default.

Default values 300 (seconds)

7.3.7 BOOTP source MAC address

Syntax `[no] mac <offset <num> | address <MACADDRESS>>`

Context System Bootstrap BOOTP context

Usage Set MAC address for BOOTP request. The source MAC-address used in BOOTP request can be:

- offset relative to system base MAC: Typically used this if you wish your product to use a MAC match the MAC of a specific LAN interface on your unit.
- a statically configure MAC: Assign a specific MAC address to use for BOOTP for this unit.

By default the source MAC is an offset to system base MAC, which would match the MAC assigned to interface *vlan1*. On most WeOS products this would mean **”mac offset 1”** (exceptions are products with more than one CPU channel; the offset equals the number of CPU channels by default).

> **Note**
> See sec. 7.3.2 and 13.4.14 for information on CPU base MAC and CPU channels. For more information on how a LAN interface is assigned its MAC address, see section 19.2.4.

Use **”no mac”** to reset the BOOTP MAC setting to default.

Use **”show mac”** to show the BOOTP MAC setting.

Default values `offset 1` (or more generally, the offset equals the number of CPU channels of the product.)

Example

```
example:/#> show iface
Press Ctrl-C or Q(uit) to quit viewer, Space for next page, <CR> for next line.

Interface Name    Oper  Address/Length      MTU    MAC/PtP Address
---------------  ----  ------------------  -----  --------------------------
lo                UP    127.0.0.1/8         16436  N/A
vlan1             UP    192.168.2.200/24    1500   00:07:7c:84:91:65
-------------------------------------------------------------------------------

example:/#> boot
example:/boot/#> bootp
example:/boot/bootp/#> show mac
00:07:7c:84:91:65 (offset 1)
example:/boot/bootp/#>
```

7.3.8 Storage of BOOTP configuration file (VFS target)

Syntax `[no] vfs-target <flash|usb>`

Context System Bootstrap BOOTP context

Usage Set virtual file system (VFS) target for configuration file.

Use this setting to save the retrieved file in a non-volatile location. By default all configuration files retrieved over BOOTP are temporary, and will be lost when rebooting the system, unless an operator saves a copy with an explicit **"`copy running-config cfg://mybackup.cfg`"** or similar (e.g., Web 'Apply' or SNMP Set).

Set to **"`vfs-target flash`"** to automatically save to built-in flash (*startup-config*) , or **"`vfs-target usb`"** to save to an external USB stick.

Use **"`no vfs-target`"** to disable the setting to get the default behaviour where the file is stored in RAM only.

Use **"`show vfs-target`"** to show the VFS target setting.

Default values Disabled (i.e., store in RAM only)

7.3.9 Manage Console Settings

Syntax `[no] console`

Context System Bootstrap context

Usage Enter System Bootstrap Console context to configure settings related to the console, or functions only available from the console.

”no console” will reset all console settings to default.

Use **”show console”** to list all console settings (also available as **”show”** command within the System Bootstrap Console context.

Default values N/A

7.3.10 Enable/Disable Console Password Reset

Syntax `[no] password-reset`

Context System Bootstrap Console context

Usage Enable or disable the function to reset the admin user’s password from the console port.

Use **”no password-reset”** to disable the password/reset login.

Use **”show password-reset”** to show whether it is enabled or disabled.

Default values Enabled

Example

```
example:/#> boot
example:/boot/#> show console
   Password reset : Enabled
   Factory reset  : Disabled
example:/boot/#> console
example:/boot/console/#> no password-reset
example:/boot/console/#> show
   Password reset : Disabled
   Factory reset  : Disabled
example:/boot/console/#>
```

7.3.11 Enable/Disable Console Factory Reset

Syntax `[no] factory-reset`

Context System Bootstrap Console context

Usage Enable or disable the function to reset the device to factory defaults from the console port.

Use **”no factory-reset”** to disable the factory/reset login.

Use **”show factory-reset”** to show whether it is enabled or disabled.

Default values Enabled

Example

```
example:/#> boot
example:/boot/#> show console
   Password reset : Disabled
   Factory reset  : Enabled
example:/boot/#> console
example:/boot/console/#> no factory-reset
example:/boot/console/#> show
   Password reset : Disabled
   Factory reset  : Disabled
example:/boot/console/#>
```

7.3.12 Manage USB Bootstrap Settings

Syntax `[no] usb`

Context System Bootstrap context

Usage Enter System Bootstrap USB context to configure settings for USB boot services.

”no usb” will reset the USB settings to default.

Use **”show usb”** to list configured USB settings (also available as **”show”** command within the System Bootstrap USB context.

Default values N/A

7.3.13 Enable/disable USB Bootstrap Services

Syntax `[no] enable`

Context System Bootstrap USB context

Usage Enable or disable USB bootstrap services.

Use **”no enable”** to disable USB bootstrap services: *USB automatic backup/restore*

and *USB deployment*[16]. It is still possible to perform manual **”backup”** (see section 7.3.25) and manual **”restore”** see section 7.3.26).

Use **”show enable”** to show whether USB bootstrap functionality is enabled or disabled.

Default values Enabled

Example

```
example:/#> boot
example:/boot/#> show usb
   Status       : Enabled
   Timeout      : Disabled
example:/boot/#> usb
example:/boot/usb/#> no enable
example:/boot/usb/#> show
   Status       : Disabled
   Timeout      : Disabled
example:/boot/usb/#>
```

7.3.14 USB wait timeout

Syntax `[no] timeout <1-60>`

Context System Bootstrap USB context

Usage Set timeout in seconds for USB stick to settle at boot.

Some USB sticks cannot be accessed immediately at power-up. This setting can be used to fine tune the time the system waits for a USB stick to settle.

The system bootup time will be prolonged up to the given timeout, unless the system discovers the USB stick before.

Default values Disabled (no timeout)

[16] **”no enable”** also disables the *technology preview* feature ”boot from USB”, see also section 7.3.4

Example

```
example:/#> boot
example:/boot/#> usb
example:/boot/usb/#> timeout 10
example:/boot/usb/#> show
   Status       : Enabled
   Timeout      : 10 second(s)
example:/boot/usb/#> leave
example:/#>
```

7.3.15 Manage bootloader settings (Barebox)

Syntax `[no] loader`

Context System Bootstrap context

Usage Enter System Bootloader context to configure settings related to the (Barebox) bootloader boot-menu. (You enter the *boot-menu* by pressing *Ctrl-C* on the console port when a unit boots.

> **Note**
> The System Bootloader context is only available for products running the Barebox bootloader.

"no loader" will reset all bootloader settings to default.

Use **"show loader"** to list all bootloader settings (also available as **"show"** command within the System Bootloader context.

Default values N/A

Example

```
example:/boot/#> show loader
Device Bootloader Configuration:

  Login Password: Disabled

  Rescue Mode Settings:
    Address: 192.168.2.200
    Netmask: 255.255.255.0
    Peer   : 192.168.2.1
    Port   : 6000
example:/boot/#>
```

7.3.16 Setting boot-menu password (Barebox)

Syntax `[no] login <password|hash> <STRING>`

Context System Bootloader context

Usage Configure a boot-menu login password. Setting a boot-menu password is recommended to improve security. When a password is configured, a user must provide the correct password to enter the boot-menu at system bootstrap.

When setting the password, you can either enter it as is (**”login password <STRING>”**), or provide a SHA1 hash of the password (**”login hash <STRING>”**).

Use **”no login”** to disable the boot-menu login password.

Use **”show login”** to see if a boot-menu login password is set or not.

Default values Disabled (`no login`)

Example

```
example:/boot/loader/#> login password TopSecret
example:/boot/loader/#> end
Saving bootloader configuration to FLASH
100% / [=================================================================]
example:/boot/#>
```

7.3.17 Setting rescue console UDP port (Barebox)

Syntax `[no] rescue-port <UDPPORT>`

Context System Bootloader context

Usage Configure UDP port for rescue-mode netconsole, e.g., **”rescue-port 12345”**. This is used as the local and remote port number for the UDP rescue console. Defaults to UDP port *6000*.

Use **”no rescue-port”** to reset UDP port to the default (6000). Use **”show rescue-port”** to show the configured UDP port.

Default values 6000

7.3.18 Setting rescue console local IP address (Barebox)

Syntax `[no] rescue-address <IPADDR>`

Context System Bootloader context

Usage Configure local IP address for rescue-mode netconsole, e.g., **"rescue-address 10.0.1.1"**. This is used as the local IP for rescue console. Defaults to address *192.168.2.200*.

This address is also used as default local IP address when selecting TFTP boot-image download (technology preview) within the boot-menu (at startup).

Use **"no rescue-address"** to reset local IP for rescue console to 192.168.2.200. Use **"show rescue-address"** to show the configured address.

Default values 192.168.2.200

7.3.19 Setting rescue console netmask (Barebox)

Syntax `[no] rescue-netmask <IPADDR>`

Context System Bootloader context

Usage Configure local IP address netmask for rescue-mode netconsole, e.g., **"rescue-netmask 255.255.0.0"**. Defaults to netmask *255.255.255.0*.

Use **"no rescue-netmask"** to reset netmask for rescue console interface to 255.255.255.0 Use **"show rescue-netmask"** to show the configured netmask.

This netmask is also used as default rescue interface netmask when selecting TFTP boot-image download (technology preview) within the boot-menu (at startup).

Default values 255.255.255.0

7.3.20 Setting rescue console peer IP address (Barebox)

Syntax `[no] rescue-peer <IPADDR>`

Context System Bootloader context

Usage Configure peer IP address for rescue-mode netconsole, e.g., **"rescue-peer 10.0.1.2"**. This is used as the peer IP for rescue console. Defaults to address *192.168.2.1*.

This address is also used as default peer IP address when selecting TFTP boot-image download (technology preview) within the boot-menu (at startup).

Use **"no rescue-peer"** to reset local IP for rescue console to 192.168.2.1. Use **"show rescue-peer"** to show the configured address.

Default values 192.168.2.1

7.3.21 List Configuration and Log Files

Syntax `dir [<cfg:// | log:// | usb://>]`

Context Admin Exec

Usage List files in the configuration file directory, log file directory, or files on a mounted USB memory. When listing configuration files you should be able to see which of the present configuration files that is used as startup file. To map a different configuration file as startup configuration, see the **"copy"** command (section 7.3.22).

Default values `cfg://`

Example

```
example:/#> dir
================================================================================
Contents of Config File System
================================================================================
              config0.cfg --> startup-config
              config1.cfg

example:/#>
```

7.3.22 Copy, Store, Restore or Paste Files

Syntax `copy <FROM_FILE> <TO_FILE>`

Several methods are available to specify <FROM_FILE> and <TO_FILE>. Local file access methods are listed below:

- Configuration files (default): **"cfg://<FILENAME>"**

- Special configuration files: **”console”**, **”running-config”**, **”startup-config”**, and **”factory-config”**.
- Log files: **”log://<FILENAME>”**
- USB memory: **”usb://[DIRECTORY/]<FILENAME>”**

Remote file access methods:

- TFTP: **”tftp://location[/directory]/filename”**
- FTP: **”ftp://[username[:password]@]location[:PORT] [directory]/filename”**

  If no username is provided, anonymous ftp login will be used. Default password is **”guest@default”**.
- SCP: **”scp://[username@]location[:PORT][/directory] /filename”**
  By default username **”admin”** will be used.
- HTTP: **”http://location[:PORT][/directory]/filename”**

Context Admin Exec

Usage Copy files, save config, transfer to/from network locations. Copy local-to-local, local-to-network and network-to-network. Special files are console, running-config, startup-config and factory-config.

The variant **”copy <FROM> startup-config”**, where **”FROM”** is a file of the form **”configN[.cfg]”** or **”cfg://file.cfg”**, changes which configuration file is used as the startup-config. In effect only changing which file startup-config points to. The contents of the previous file it pointed to remains untouched.

This also means that you can **not** copy a file directly to startup-config from any VFS. I.e., when copying a file from (T)FTP or USB you must first copy the file to a configN[.cfg] file in the cfg:// VFS.

Please note, the use of the special file **”console”** is very similar to the old DOS style usage. Albeit limited to the usage: **”copy console <FILE>”**. When issuing this command you are presented with a *paste area* where you can safely type in or paste parts of, or full, configuration files. However, when pasting in partial ”.cfg” file snippets the system will use WeOS defaults for unspecified settings.

Also, the destination file in **”copy console <FILE>”** cannot be the console

itself or factory-config, which is read-only. Hence we recommend using: **"copy console config<N>"** or **"copy console running-config"**.

Default values N/A

Examples

1. Restore factory default (to running configuration)

Example

```
example:/#> copy factory-config running-config
Using default factory.cfg found in firmware image.
Stopping Syslog daemon ..................................... [ OK ]
Starting Syslog daemon ..................................... [ OK ]
example:/#>
```

2. Store running configuration to startup configuration

Example

```
example:/#> copy running-config startup-config
example:/#>
```

3. Copy configuration file from USB to local configuration file *config3*.

Example

```
example:/#> copy usb://myconfig.cfg config3
Copying myconfig.cfg to config3 ...
Done.
example:/#>
```

4. Copy configuration file onto remote server using FTP.

Example

```
example:/#> copy cfg://config0.cfg ftp://mylogin:mypw@192.168.2.99/myconfig
example:/#>
```

7.3.23 Delete a Configuration File

Syntax `erase [filesys://]<FILENAME>`

filesys can be **"cfg"**, **"log"**, or **"usb"**, with **"cfg"** as default.

Context Admin Exec

Usage Delete a configuration file, log file or a file on a mounted USB memory.

Default values **"cfg"** is the default file system.

Example

```
example:/#> dir
================================================================================
                        Existing Configurations on System
================================================================================
                  config0 --> startup-config
                  config1

example:/#> erase config1
example:/#> dir
================================================================================
                        Existing Configurations on System
================================================================================
                  config0 --> startup-config

example:/#>
```

7.3.24 Show Configuration File (or other files)

Syntax `show <running-config|startup-config|factory-config|[<filesys>://]<FILENAME>`

filesys can be **"cfg"**, **"log"**, or **"usb"**, with **"cfg"** as default.

Context Admin Exec

Usage Show content of a configuration file, log file, or file on a mounted USB memory. Special files are *running-config*, *startup-config* and *factory-config*. Use the **"dir"** command to list files (section 7.3.21).

Default values **"cfg"** is the default file system.

7.3.25 Activate Auto-Backup

Syntax backup (applicable on units with USB port)

Context Admin Exec

Usage This command activates WeOS automatic backup and restore for USB media. The directory **"/usb/westermo/backup"** is used for this purpose.

See section 7.1.6 for details.

Default values Not applicable.

7.3.26 Manual Restore from USB

Syntax `restore` (applicable on units with USB port)

Context Admin Exec

Usage Force restore from USB to running-config.

This command can be used to force an auto-restore of backup files from a USB stick to **"cfg://"** and also activate the new startup-config in the system running-config.

See section 7.1.6 for details.

Default values Not applicable.

7.3.27 Rebooting the Device

Syntax `reboot`

Context Admin Exec

Usage Reboot the device. The switch will boot up with its *startup-config*.

Default values Not applicable.

7.3.28 Import Certificate/Key

Syntax (for PKCS#12)

```
cert import pkcs [password <PASSWORD>] <URI> [label <LABEL>]
```

Syntax (for PEM)

```
cert import pem type <private|public|ca> <URI> [label <LABEL>]
```

Syntax (for OpenVPN key)

```
cert import ovpn <URI> [label <LABEL>]
```

Context Admin Exec

Usage Import PKCS#12 certificate bundle, individual certificate files in PEM format, or an OpenVPN static key. An optional label name can be specified. By default the label name is set from the file name.

Examples:

- **”cert import pkcs password "secret" ftp://1.2.3.4/bundle.p12”**
- **”cert import pem type public usb://remote.crt”**
- **”cert import ovpn ftp://1.2.3.4/tls-auth.key label tls”**

To remove/delete a certificate by label, use ’force’ to avoid questions:

- **”no cert remote”** (Remove certificate file with label ”remote”. There can be different certificate files (of different types) with the same label. If so, a separate question will be asked for each file before removal.)
- **”no cert force remote”**

Default values Not applicable.

7.3.29 List and show details of Certificates

Syntax `show cert [LABEL]`

Context Admin Exec

Usage List all certificates, or show details of a specific certificate.

Example to show all certificates, or display/dump a given label:

- **”show cert”** (lists all certificates)
- **”show cert remote”** (list details of certificate with label ”remote”. There can be different certificate files (of different types) with the same label. Then all are shown.

Default values Not applicable.

7.3.30 Ping

Syntax `ping [-i <IFACE|IPADDR>] [-s <size>] [-c <count>] [-t <TTL>] [-M <hint>] <HOST>`

Context Admin Exec context

Usage Ping a remote host.

Ping is useful as a basic diagnostic tool.

The -i option can be used to select the interface to send ICMP_ECHO on, which is useful in, e.g., VPN setups. The -i option can also be used with an IP address to spoof the source IP address.

The -M option is used to control where to set the DF (don't fragment) bit in the ICMP packet. If this bit is set, no one will be allowed to fragment this packet and an error will be generated if the packet is to big to fit in the MTU. Valid options for hint:

- do: Set the don't fragment bit, prohibit all fragmentation.
- dont: Never set the don't fragment bit.
- want: Make a MTU discovery and fragment packet if it is too large to fit in the MTU.

You can use use the domain name or IP address as the host argument, but you need a valid name server setup for domain names to work, see section 19.7.5.

Default values Not applicable.

Example

```
example:/#> ping 192.168.131.1
Ctrl-C to abort PING 192.168.131.1 (192.168.131.1): 56 data bytes
64 bytes from 192.168.131.1: seq=0 ttl=64 time=4.832 ms
64 bytes from 192.168.131.1: seq=1 ttl=64 time=0.836 ms
64 bytes from 192.168.131.1: seq=2 ttl=64 time=0.810 ms
64 bytes from 192.168.131.1: seq=3 ttl=64 time=0.823 ms

--- 192.168.131.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.810/1.825/4.832 ms
example:/#>
```

7.3.31 Traceroute

Syntax `traceroute <HOST>`

Context Admin Exec context

Usage Trace the path the packets take to a remote host.

Traceroute is useful as a basic diagnostic tool.

You can use use the domain name or IP address as the host argument, but you need a valid name server setup for domain names to work, see section 19.7.5.

Default values Not applicable.

Example

```
example:/#> traceroute 192.168.130.41
traceroute to 192.168.130.41 (192.168.130.41), 30 hops max, 40 byte packets
 1  192.168.131.1  1.116 ms  0.755 ms  0.806 ms
 2  192.168.130.41  0.824 ms  0.705 ms  0.742 ms
example:/#>
```

7.3.32 Remote Login to another device (SSH Client)

Syntax `ssh [USER@]<IPADDR|DOMAINNAME>[/PORT]`

Context Admin Exec context.

Usage Login to remote device using SSH.

Default values Default user **”admin”**, default (TCP) port number **”22”**.

7.3.33 Remote Login to another device (Telnet Client)

Syntax `telnet <IPADDR|DOMAINNAME>[:PORT]`

Context Admin Exec context.

Usage Login to remote device using Telnet.

Default values Default (TCP) port number **”23”**.

7.3.34 Show IPConfig Neighbours

Syntax `show ipconfig [IFNAME]`

Context Admin Exec context.

Usage The command has two purposes:

- Scan the network for IPConfig neighbours on the given interface, i.e., scan for other Westermo devices with the *IPConfig service* enabled (see section 7.3.46).
- Show status of the IPConfig process on the own device, if enabled.

Note: There is another **"show ipconfig"** command available in the Global Configuration context, which shows IPConfig *server configuration* settings, see section 7.3.46.

Default values If no interface is given, a scan for IPConfig neighbours is tried on interface *vlan1* (if existing).

Example

```
example:/#> show ipconfig
Using default interface vlan1
MAC               IP                  Ver.  Type              Status
================================================================================
00:07:7c:87:85:23  192.168.2.100/24    4.03 Lynx+             ------------SI
00:07:7c:87:85:13  192.168.2.200/24    4.03 Lynx+             ------------RSI
00:07:7c:87:57:a3  192.168.2.201/24    4.03 Lynx+             FOC:RING:MN:RSI
00:07:7c:87:85:d3  192.168.2.225/24    4.03 Lynx+             MEM:RING:MN:RSI
================================================================================
Process ipconfigd running as PID 475
example:/#>
```

Explanations to the output:

- MAC: The *base MAC* address of the discovered device.
- IP: The IP address of the discovered device.
- Version: Software version on the discovered unit. In the example above, all discovered devices are running some variant of 4.3.x software. The *platform generation* number (4) and *feature release* (03) number are shown, but we cannot determine if those units are running 4.3.0, 4.3.1 or some other 4.3.x patch revision.
- Type: The type of Westermo device discovered.
- Status:
  - If FRNT is enabled, the role is displayed as **"FOC"** (focal point) or **"MEM"** (member switch), and one can also see whether the FRNT ports are up or down: **"M"** - FRNT port M is up, **"m"** - FRNT port M is down, and so on. Note: the ports "M" and "N" refers to the operational state of the FRNT port, which can differ from their configured role if the ports are connected in the wrong order (swapped).

- If RSTP/STP is enabled on the discovered device, the letter **”R”** is shown.
- If SNMP is enabled on the discovered device, the letter **”S”** is shown.
- If IGMP Snooping is enabled on the discovered device, the letter **”I”** is shown.

7.3.35 Manage Port Monitoring

Syntax `[no] monitor`

Context Admin Exec context

Usage Use the **”monitor”** command to enter the Port Monitoring context.

”no monitor” will disable port monitoring (in the same way as **”no enable”** within the Port Monitoring, see section 7.3.36).

Use **”show monitoring”** to show port monitoring settings (also available as **”show”** command within the Port Monitoring context).

Default values Not applicable.

7.3.36 Enable/disable Port Monitoring

Syntax `[no] enable`

Context Port Monitoring context

Usage Enable port monitoring. Use **”no enable”** to disable port monitoring.

Use **”show enable”** to list whether port monitoring is enabled or disabled.

Default values `no enable` (disabled)

7.3.37 Set Mirror Port

Syntax `[no] destination <PORT>`

Context Port Monitoring context

Usage Set the monitor destination port, i.e., the *mirror* port.

Use **”show destination”** to show currently configured port monitoring destination port.

Default values By default there is no destination port.

7.3.38 Set Monitored Ports

Syntax `[no] source <PORTLIST> [ingress] [egress]`

Context Port Monitoring context

Usage Add/delete/update monitor source port(s), i.e., the ports being *monitored*.

Use **”show source”** to show current set of ports being monitored.

Default values By default there are no source ports. Commands apply both to **”ingress”** and **”egress”** if neither is specified.

7.3.39 Manage LLDP settings

Syntax `[no] lldp`

Context Global Configuration context.

Usage Enter LLDP Configuration context. Use **”no lldp”** to disable lldp.

Use **”show lldp”** to view the current configuration. Alternatively, you can enter the LLDP Configuration context and run **”show”** (see example in section 7.3.40).

Default values LLDP is enabled by default.

7.3.40 Enable/disable LLDP

Syntax `[no] enable`

Context LLDP Configuration context.

Usage Enable/disable LLDP. Use **”enable”** to enable and **”no enable”** to disable LLDP on all LAN ports. (As of WeOS v4.17.1 **”no enable”** will be stored as **”no lldp”**, see section 7.3.39.)

Default values LLDP is enabled by default.

Example

```
example:/config/#> lldp
example:/config/lldp/#> enable
example:/config/lldp/#> show
LLDP is enabled
example:/config/lldp/#>
```

7.3.41 Show LLDP Status

Syntax `show lldp`

Context Admin Exec context.

Usage Show LLDP information about neighbouring devices.

Default values Not applicable.

Example

```
example:/#> show lldp
--------------------------------------------------------------------------------
LLDP neighbors:
--------------------------------------------------------------------------------
Interface:    Eth 10, via: LLDP, RID: 1, Time: 0 day, 01:32:31
  Chassis:
    ChassisID:    mac 00:07:7c:84:d7:44
    SysName:      wolverine
    SysDescr:     Wolverine WeOS v4.9.x
    MgmtIP:       192.168.2.2
    Capability:   Bridge, off
    Capability:   Router, on
    Capability:   Wlan, off
  Port:
    PortID:       mac 00:07:7c:84:d7:47
    PortDescr:    10/100TX Eth 2/1
  VLAN:         1 vlan1
  LLDP-MED:
    Device Type:  Network Connectivity Device
    Capability:   Capabilities
    Capability:   Policy
    Capability:   Location
    Capability:   MDI/PSE
    Capability:   MDI/PD
    Capability:   Inventory
--------------------------------------------------------------------------------
```

7.3.42 Enable/disable Web Management Interface

Syntax `[no] web`

Context Global Configuration context.

Usage Enable web management interface, and enter Web Configuration context. Use **"no web"** to disable the web server.

> **Warning**
> Then the switch cannot be managed via the Web interface.

Use **"show web"** to list current Web configuration settings (also available as **"show"** command within the Web Configuration context).

Default values Enabled (**"web"**)

7.3.43 Set Web Management Session Timeout

Syntax `[no] session-timeout <TIMEOUT>`

Context web context.

Usage Configures the session timeout. (**"no session timeout"**) disables timeout.

Default values 10 min

7.3.44 Set Web Management HTTP port

Syntax `[no] port <PORT>`

Context web context.

Usage Configures the HTTP port.

Default values 80

7.3.45 Set Web Management HTTPS port

Syntax `[no] ssl-port <PORT>`

Context web context.

Usage Configures the HTTPS (SSL) port.

Default values 443

7.3.46 Enable/disable IPConfig Service

Syntax `[no] ipconfig`

Context Global Configuration context.

Usage Enable IPConfig service interface (management of the unit via the Westermo IPConfig protocol), and enter IPConfig Configuration context. Use **"no ipconfig"** to disable the IPConfig server

> **Warning**
> After this the switch cannot be managed (or detected) via the IPConfig protocol, used by the WeConfig tool.

Use **"show ipconfig"** to list whether IPConfig is enabled or disabled. **Note:** There is another **"show ipconfig"** command available in the Admin Exec context, which is used (1) to scan for neighbour Westermo units, and (2) to list *status* information on the IPConfig server running on this device, see section 7.3.34.

Default values Enabled (**"ipconfig"**)

Examples

1. How to check whether IPConfig service is enabled on my switch:

   Example

   ```
   example:/#> config
   example:/config/#> show ipconfig
   Ipconfig is enabled
   Read only mode : Disabled
   example:/config/#> end
   ```

2. How to enable/disable IPConfig service:

   Enter Global Configuration context, check the current IPConfig configuration, and modify it if desired. Below is an example of how to disable IPConfig.

Example

```
example:/#> config
example:/config/#> show ipconfig
Ipconfig is enabled
Read only mode : Disabled
example:/config/#> no ipconfig
Deactivating ipconfig service.
example:/config/#> end
```

7.3.47 Enable/Disable configuration and upgrade via IPConfig service

Syntax `[no] read-only`

Context IPConfig Configuration context.

Usage The IPConfig service (used by the WeConfig tool) can be used to discover and view status of a unit, but also for some simple configuration (IP address, netmask and default gateway) and firmware upgrade (primary firmware). By setting IPConfig in **"read-only"** mode, no configuration or firmware upgrade is possible via IPConfig service.

Use **"show read-only"** to list whether 'read-only' is enabled or disabled. Use **"read-only"** to activate 'read-only' mode, and **"no read-only"** to set the mode to 'read/write'.

Default values Disabled (**"no read-only"**, i.e., configuration and upgrading via IPconfig service is possible.)

Examples How to limit IPConfig service to 'read-only'. That is, disabling configuration and upgrading of the unit via IPConfig, while allowing use of IPConfig to discover the unit and status information retrieval.

Example

```
example:/#> config
example:/config/#> show ipconfig
Ipconfig is enabled
Read only mode : Disabled
example:/config/ipconfig/#> read-only
Setting IPconfig read only mode Enabled
example:/config/ipconfig/#> end
```

7.3.48 Enable/disable SSH Service

Syntax `[no] ssh`

Context Global Configuration context.

Usage Enable SSHv2 management service, and enter SSH Configuration context. Use **"no ssh"** to disable the SSHv2 server.

> **Warning**
> Then the switch cannot be managed via SSHv2.

Use **"show ssh"** to list current SSH configuration settings (also available as **"show"** command within the SSH Configuration context).

Default values Enabled (**"ssh"**)

7.3.49 Enable/disable Telnet Service

Syntax `[no] telnet`

Context Global Configuration context.

Usage Enable Telnet management service, and enter Telnet Configuration context. Use **"no telnet"** to disable the Telnet server.

> **Warning**
> Then the switch cannot be managed via Telnet.

Use **"show telnet"** to list current Telnet configuration settings (also available as **"show"** command within the Telnet Configuration context).

Default values Disabled (**"no telnet"**)

7.3.50 Show System Environment Sensors

Syntax `show env`

Context Admin Exec context.

Usage List available environment sensors, their index, and their current value. Examples of sensors are *power* (DC1 and DC2), Digital In, and Temperature sensors.

If the unit is equipped with DDM/DOM capable SFPs[17], the voltage, bias current, Tx power, Rx power and temperature parameters will be listed for each SFP.

Default values Not applicable.

7.3.51 Show System Uptime

Syntax `show uptime`

Context Admin Exec context.

Usage Show system uptime.

Default values Not applicable.

7.3.52 Show Memory Usage

Syntax `show memory`

Context Admin Exec context.

Usage Show system memory usage.

Default values Not applicable.

7.3.53 Show Running Processes

Syntax `show processes`

Context Admin Exec context.

Usage Show a list of currently running processes.

Default values Not applicable.

[17] DDM/DOM diagnostic information is only available for Westermo DDM SFPs, see the SFP Transceiver Datasheet of your WeOS product (www.westermo.com).

7.3.54 Show Flash Partition Table

Syntax `show flash-table`

Context Admin Exec context.

Usage Show information on the flash partition table.

Default values Not applicable.

7.3.55 Show Partition table

Syntax `show partitions`

Context Admin Exec context.

Usage Show information on the flash partition table. The **"`show partitions`"** is similar to the **"`show flash-table`"** command (section 7.3.54), but presents the partition table somewhat differently.

Default values Not applicable.

Examples

- Example with a WeOS unit (Basis platform) with RedBoot boot-loader (see partition *mtd0*).

Example

```
example:/#> show partitions
Partition  Name                 Size
================================================================================
mtd0       RedBoot              512.0 KiB
mtd1       Linux_main            12.5 MiB
mtd2       Linux_backup          12.5 MiB
mtd3       JFFS2                  4.0 MiB
mtd4       Branding               2.1 MiB
mtd5       RedBoot config         4.0 KiB
mtd6       FIS directory        128.0 KiB
example:/#>
```

- Example with WeOS unit (Corazon platform) with U-boot boot-loader (see partition *mtd4*).

Example

```
example:/#> show partitions
Partition  Name                 Size
================================================================================
mtd0       Linux_main            56.0 MiB
mtd1       Linux_backup          56.0 MiB
mtd2       Config                15.0 MiB
mtd3       U-Boot Config        512.0 KiB
mtd4       U-Boot               512.0 KiB
example:/#>
```

- Example with WeOS unit (Corazon platform) with Barebox boot-loader (see partition *mtd4*).

Example

```
example:/#> show partitions
Partition  Name                 Size
================================================================================
mtd0       Linux_main            56.0 MiB
mtd1       Linux_backup          56.0 MiB
mtd2       Config                15.0 MiB
mtd3       BareboxEnv           512.0 KiB
mtd4       Barebox              512.0 KiB
example:/#>
```

7.3.56 Update Flash Partition Table

Syntax `flash-table-update`

Context Admin Exec context.

Usage This command is used to update the flash partition table on early RedFox units, in order to allow firmware upgrades to WeOS release 4.3.0 or later. For details, see section 7.1.11.

Default values Not applicable.

Chapter 8

Ethernet Port Management

By default all ports on the switch are enabled. Section 8.1 provides general information about the available port settings. Section 8.2 covers port settings via the Web interface and section 8.3 port settings via the CLI.

8.1 Overview of Ethernet Port Management

The table below presents available port settings. The features are presented further in the following sections.

Feature	Web	CLI	General Description
Enable/disable port	X	X	
Speed-duplex mode	X	X	Section 8.1.2
Flow control	X	X	Section 8.1.3
Port priority (level)	X	X	Section 8.1.4
Port priority mode	X	X	Section 8.1.4
Link alarm	X	X	Section 8.1.5
Inbound rate limit	X	X	Section 8.1.6
Rate Selection	X	X	-”-
Traffic Selection		X	-”-
Outbound traffic shaping	X	X	Section 8.1.7
MDI/MDIX	X	X	Section 8.1.8
Fastlink	(X)	(X)	Section 8.1.9

Continued on next page

Continued from previous page

Feature	Web	CLI	General Description
fallback default-VID		X	Section 8.1.10
PHY fine tuning		X	
Shielded/Unshielded TP cable		X	
TX power mode		X	
View port configuration	X	X	
View port status	X	X	
View SFP DDM/DOM diagnostics	X	X	Section 8.1.11

8.1.1 Port naming conventions

The convention to name communication ports such as Ethernet ports, DSL ports, and Serial ports differs between WeOS products and product families.

8.1.1.1 Simple numbering

Lynx, Falcon, DDW-142 (Wolverine), RedFox Industrial Rack, RedFox Rail, and Viper all use a simple *port ID* to refer to their ports.

- *Lynx[46] and RedFox Industrial Rack[49]:* Ethernet ports on Lynx and RedFox Industrial Rack are named *1*, *2*, *3*, . . .
- *Falcon[41], Lynx-DSS[43] and some Wolverine units (DDW-142[37]):* These units have multiple port types; Ethernet, serial port(s) and xDSL/SHDSL (Falcon/Wolverine), which are numbered individually per port type. For example, Falcon is equipped with:
  - four Ethernet ports (numbered *1*, *2*, *3* and *4*),
  - one xDSL port (numbered *1*), and
  - one serial port (numbered *1*).

  As Ethernet and xDSL ports can be used in overlapping contexts, e.g., they can be associated with the same VLAN, a port *qualifier* (**”eth”** or **”dsl”**) is

sometimes used to distinguish Ethernet port 1 (**”eth 1”**) from xDSL port 1 (**”dsl 1”**).

- *RedFox Rail and Viper:* Ethernet ports on RedFox Rail and Viper are named *X1*, *X2*, *X3*, . . .

8.1.1.2 Slot based numbering

RedFox Industrial[47, 48] and some Wolverine products (DDW-225[39] and DDW-226[40]) use a slotted architecture, and ports are named according to the *slot ID* and the *port's position* within that slot. For example, port *1/2* would denote the second port in the first slot.

This name convention is used irrespective of port type, e.g., DDW-226 (Wolverine) has two SHDSL ports (1/1-1/2), 4 Ethernet ports (2/1-2/4), and one Serial port (1/1). Details on the name convention and the slotted architecture is described further below, using RedFox Industrial as example.

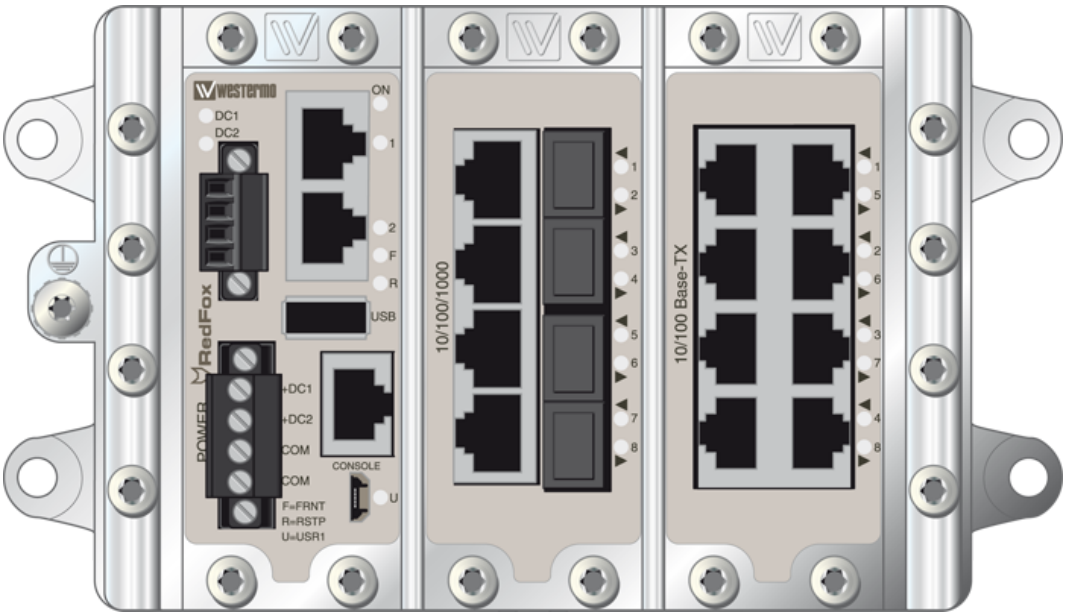


Figure 8.1: Three-slot RedFox Industrial switch equipped with a 8-port Gigabit/SFP card (middle slot), and an 8-port 10/100BaseTX card (right slot).

The *RedFox Industrial* switches come in a two-slot and a three-slot version. Fig. 8.1 shows a sample three-slot RedFox Industrial equipped with a 4-port Gigabit/SFP

card (middle slot) and an 8-port 10/100BaseTX card (right slot). The leftmost slot contains the Power/CPU card, which is present on all RedFox Industrial switches.

RedFox Industrial makes use of a slotted architecture with different combinations of interface modules. As mentioned above WeOS numbers the ports based on *slotID/portID*, where the

- the *slotID* denotes the slot's position within the rack (left to right), and
- the *portID* denotes the port's position within the slot (left to right, up to down).

For example, the three Ethernet ports in the leftmost slot (slot 1) are named *1/1* (top), *1/2* (middle). and *1/3* (bottom). The ports in the second slot are named *2/1-2/4* (left side) and *2/5-2/8* (right side), and ports in slot 3 are named *3/1-3/4* (left side) and *3/5-3/8* (right side).

8.1.2 Port speed and duplex modes

By default ports are configured to auto-negotiate speed (10/100/1000 Mbit/s) and duplex modes (half/full) to the "best" common mode when a link comes up. When configured for auto-negotiation, the resulting speed and duplex mode agreed is shown as part of the port status information.

It is possible to disable auto-negotiation and instead use a static speed and duplex mode setting. When using a static speed and duplex setting, the operator should ensure that the ports on both ends of the link are configured with the same static speed and duplex settings.

Depending on Ethernet port *type*, the available port speeds will differ:

- Fast Ethernet copper ports: Fast Ethernet copper ports are capable to operate at 10 or 100 Mbit/s.
- Gigabit Ethernet copper ports: Gigabit Ethernet copper ports are capable to operate at 10, 100 or 1000 Mbit/s.
- Gigabit Ethernet fibre ports: Gigabit Ethernet fibre ports are capable to operate at 1000 Mbit/s.

8.1.3 Flow control

The ports can be configured to use *flow control*, i.e., to dynamically limit inbound traffic to avoid congestion on outbound ports.

When flow control is enabled on a *full duplex* port, the switch will send *pause frames* (IEEE 802.3x) to limit inbound traffic on this port, if that traffic is causing congestion when sent out on another switch port.

When flow control is enabled on a *half duplex* port, the switch will use a technique known as *back-pressure* to limit inbound traffic on this port, if that traffic is causing congestion when sent out on another switch port. (The *back-pressure* technique enables a switch to force its neighbour to slow down by sending *jamming signals* on that port, thus emulating a packet collision.)

8.1.4 Layer-2 priority support

Each Ethernet port has four output queues, enabling layer-2 priority support with four traffic classes. The queues are serviced according to *strict priority scheduling*, i.e., when there are traffic in multiple queues, the packets in the queue with higher priority is serviced first.

A packet's priority is determined when it enters on a port, and can be classified based on:

- VLAN ID: The switch can be configured to give specific priority to certain VLANs. This can be useful to, e.g., when providing IP telephony via a dedicated VLAN. Priority based on VLAN ID has precedence over all priority classifications described below.

  VLAN ID priority settings are further described in chapter 13.

- VLAN tag: For packets carrying a VLAN tag, the packet's priority can be based on content of the priority bits inside the VLAN tag. The VLAN tag is useful to carry packet priority information on inter-switch links.

  Use of VLAN tag priority can be configured per port (see sections 8.2 and 8.3).

- IP ToS/DiffServ: For IP packets the priority can be classified based on the content of the IP ToS bits (IPv4) or the IP TC bits (IPv6). Classification based on the IP ToS/Diffserv bits can be useful to provide higher priority to delay sensitive applications, such as *IP telephony* and *remote login*, than to

bulk data applications, such as *file transfer*, however, it requires that those applications can set the IP ToS/Diffserv bits appropriately.

Use of IP ToS/DiffServ priority can be configured per port (see sections 8.2 and 8.3).

- Port Priority: Priority can be classified based on the inbound port.

  Use of port priority can be configured per port (see sections 8.2 and 8.3). Furthermore, when priority classification is configured to be based on VLAN tag (or IP ToS/DiffServ), priority will be based on the port priority for untagged (or non-IP respectively) packets.

When priority is classified based on VLAN ID, VLAN tag, or port priority, the priority assigned to a packet will take a value in range 0-7, and be represented by 3 bits (IEEE 802.1p). The mapping of 802.1p priority (8 values) to traffic class (4 output queues) is shown in table 8.2. The rationale behind this mapping is described in IEEE 802.1Q-2005 (Annex G).

IEEE 802.1p priority	Queue number/ Traffic Class
0	0 (lowest)
1	0
2	1
3	1
4	2
5	2
6	3
7	3 (highest)

Table 8.2: Mapping of IEEE 802.1p priority to Queue/Traffic Class.

When priority is classified based on IP ToS/DiffServ, the priority assigned to a packet will take a value in range 0-63, and be represented by 6 bits (DSCP - Differentiated Services Code Point). The mapping of DSCP priority (64 values) to traffic class (4 output queues) is shown in table 8.3. This mapping is in line with the use of IP Precedence fields (RFC 1349), and IP DiffServ for *best effort* and *control* traffic (RFC 2474), *assured forwarding* (RFC 2597) and *expedited forwarding* (RFC 3246).

Packets sent out on a port *with* a VLAN tag will carry priority information (802.1p) within their VLAN tag.

IP Priority bits						Queue bits		Queue number/ Traffic class
5	4	3	2	1	0	1	0	
0	0	-	-	-	-	0	0	0 (lowest)
0	1	-	-	-	-	0	1	1
1	0	-	-	-	-	1	0	2
1	1	-	-	-	-	1	1	3 (highest)

Table 8.3: Mapping of IP priority bits to Queue/Traffic Class.

- For packets where priority was classified based on VLAN ID, VLAN tag, or port priority, the outbound priority (3 bits) will be equal to the determined inbound priority (3 bits).
- When priority is classified based on IP ToS/DiffServ, determining the outbound priority (3 bits) is more complex: the two most significant bits of the outbound priority will be equal to the queue number (i.e., queue bits in table 8.3), while the least significant bit of the outbound priority is equal to the least significant bit of the inbound port's configured port priority.

  E.g., if the packet is put in priority queue 2 (binary '10'), and the port priority of the inbound port has an odd value (least significant bit is '1'), the packet will carry priority value 5 ('101') in its VLAN tag when sent on the outbound port.

Warning

Configuration of layer-2 priority should be handled with care. In particular, mapping user traffic to the highest priority queue is discouraged, since that may affect time critical control traffic, such as FRNT traffic, already mapped to the highest priority queue.

8.1.5 Link alarm

Each Ethernet port on the switch can be configured to indicate alarm when the link comes up or goes down. The alarm is indicated in multiple ways:

- *SNMP trap:* An SNMP trap will be sent when a link changes state, i.e., both when the link comes up, or when it goes down. This assumes that SNMP is

enabled, and that a trap host is configured. See chapter 6 for more information.

- *Front panel LEDs:* A link alarm may effect both the individual LED of the port, as well as the common status LED for the switch (for definite information about what functions affect the common status LED, see chapter 24):
  - *Individual LED:* Each Ethernet port has a LED, which generally indicates 'green' if the link is up. If there is no link, the LED will indicate 'yellow' when link alarm is configured.
  - *Common status LED:* The switch has a common status LED, labelled 'ON' on the front panel. This LED will generally indicate 'green' if all associated functions are OK, and 'red' if one or more of the associated alarm sources are 'NOT OK'. E.g., if one of the ports configured with link alarm indicates link down, the common status LED will be 'red'.
- *Web interface:* Link alarms (link down) are indicated on the *main* Web page, and the *port configuration/status* page.
- *CLI:* A link alarm (link down) is indicated by an exclamation mark ('!') when displaying the port's status in the CLI.
- *Digital I/O:* A link alarm can affect the output level of the digital I/O port in the same way as it will affect the common status LED.

  For more information on the functionality of the Digital I/O port, see chapter 24.

8.1.6 Inbound/Ingress rate limiting

The switch can be configured to limit the rate of a port's incoming traffic - *inbound rate limiting* (also referred to as *ingress rate limiting*). By default a port will accept packets at a rate up to the link speed, but with inbound rate limiting activated the switch will start dropping packets when data arrives above the given rate threshold.

The inbound rate limiting feature can be useful as a complement to layer-2 priority handling (see section 8.1.4) when congestion within the network is to be avoided.

There are two configuration settings for inbound rate limiting:

- *Rate:* Defines the threshold data rate. The web interface provides a predefined set of rates (drop-down list). The CLI allows for more fine-grain rate settings:
  - in steps of 64 kbit/s in range 64-1000 kbit/s
  - in steps of 1 Mbit/s in range 1-100 Mbit/s
  - in steps of 10 Mbit/s in range 100-1000 Mbit/s (on Gigabit Ethernet ports.)

  Rate limiting calculations consider the layer-2 bits, i.e., from Ethernet destination MAC address to CRC (interframe gap and preamble bits are not counted).
- *Traffic Type:* Defines the kind of traffic subject to inbound rate limiting. By default, a configured rate limit will apply to all traffic, however, it is possible to restrain the rate limit to specific (layer-2) traffic types: broadcast, multicast and/or unknown[1] unicast. As of WeOS v4.17.1 selection of traffic types can only be done via the CLI.

[1] Unknown unicast traffic is traffic with a unicast destination MAC address not present in the switch forwarding database (see section 13.4.19). Unknown unicast traffic is flooded onto all ports within the (V)LAN.

8.1.6.1 Restrictions on inbound rate limiting

On RedFox units, some of the interface modules have hardware dependent restrictions regarding the inbound rate limit function. These restrictions are described in this section.

Which Ethernet ports on RedFox have the restrictions described here?

The restrictions apply to Ethernet ports of switchcores MV88E6095 and MV88E6185. Please see *Detailed System Overview* page in the Web (section 4.4.2) or use the **"show system-information"** in the CLI (section 7.3.2) to find *definite* information about what switchcore(s) is used in your product. An informative list of products/modules where the restrictions apply is given below:

- RedFox Industrial (RFI) with *Corazon* platform[48]: Only Ethernet ports on modules "F4G" and "F4G-T4G" (MV88E6185) have these restrictions.
- RedFox Industrial (RFI) with *Atlas* platform[47]: Ethernet ports on all modules except module "F8" have these restrictions.
- RedFox Industrial Rack (RFIR)[49]: Only Ethernet ports in the 8-port group/module with Gbit/s ports (4 Gbit/s SFP and 4 Gbit/s Copper ports; MV88E6185) have these restrictions.
- RedFox Rail (RFR) with *Corazon* platform[50]: No Ethernet ports on the RFR-212 have these restrictions.
- RedFox Rail (RFR) with *Atlas* platform (not for sale): All Ethernet ports on the RFR-12 have these restrictions.

- *TCP traffic:* When the data rate rises above the given threshold on these Ethernet ports, packets will be dropped in a manner "punishing" TCP traffic relatively hard. Thus, activating inbound rate limiting applicable to unicast traffic may have an undesired impact on your TCP traffic,.
- *Traffic types:* When restricting the inbound rate limit to a certain traffic type (broadcast, multicast and/or unknown unicast) on these Ethernet ports, there are dependencies between the settings:
  - *Unknown unicast:* Selecting "unknown unicast" implies that "unknown unicast", "multicast" and "broadcast" traffic will be subject to inbound rate limiting.
  - *Multicast:* Selecting "multicast" implies that "multicast" and "broadcast" traffic will be subject to inbound rate limiting.

  - *Broadcast:* Selecting ”broadcast” simply means that ”broadcast” traffic will be subject to inbound rate limiting.

- *Rate limiting on Gigabit ports:* Maximum rate limit on (MV88E6185) Gigabit ports is 250 Mbit/s. Setting a higher rate limit (e.g., 300 Mbit/s) will result in a rate limit of 250 Mbit/s.

Due to these restrictions, it is recommended that *inbound rate limiting* is primarily used as a means of storm prevention, on the ports where these restrictions apply.

8.1.7 Outbound/Egress traffic shaping

The switch can be configured to limit the outbound data rate on a port (outbound traffic shaping). By default each port will send at the maximum speed of the link, but with outbound traffic shaping activated the switch will limit the outbound rate to a given threshold. Above that threshold the switch will buffer packets - *bursty* traffic will be *shaped*. In case the output buffer is full, additional packets destined for that port will be dropped.

When configuring the *threshold rate* for outbound traffic shaping, the same settings as for inbound rate limiting (see section 8.1.6) applies. For outbound traffic shaping it is also possible to specify rate in frames per second. The web interface provides a predefined set of rates (drop-down list). The CLI allows for more fine-grain rate settings:

- Bits per second:
  - in steps of 64 kbit/s in range 64-1000 kbit/s
  - in steps of 1 Mbit/s in range 1-100 Mbit/s
  - in steps of 10 Mbit/s in range 100-1000 Mbit/s (on Gigabit Ethernet ports)
- Frames per second: in range 7700-1488000 frames per second

Traffic shaping calculations consider the layer-2 bits, i.e., from Ethernet destination MAC address to CRC (interframe gap and preamble bits are not counted).

Note
Outbound traffic shaping in *frames per second* mode is available for Ethernet ports on all WeOS products, with exceptions for ports on some RedFox and RedFox Industrial Rack models. The Ethernet ports listed to have restrictions for ingress rate limiting (see section 8.1.6.1) also lack support for the *frames per second* mode.
Furthermore, outbound traffic shaping in *frames per second* mode is *not* available available for DSL ports (ADSL/VDSL or SHDSL) ports.

8.1.8 MDI/MDIX crossover

By default a switch is able to sense which pin to use for reception and which to use for transmission (auto MDI/MDIX crossover), thus no external crossover cable is necessary. In addition, a port can be configured statically in MDI (Media Dependent Interface) or MDIX (crossover) mode.

8.1.9 Fastlink - Fast link-up/link-down on fixed 10/100 Ethernet copper ports

Default port settings in WeOS are aimed at being conformant and compatible with as many devices as possible. Therefore the ports are setup to auto-negotiate speed, duplex and automatically agree with the link partner on which end should cross RX and TX when a straight cable is used. Naturally this takes a bit of time, despite all current products today do this in dedicated PHY circuitry.

To speed things up considerably, a feature called ”Fastlink” can be activated on fixed 10/100 Mbit/s Ethernet copper ports[2]. This feature basically disables any IEEE back-offs and timeouts in place to protect against glitches and temporary link loss that otherwise prevent the port from going UP or DOWN. Westermo has put a great deal of effort into making sure that, when enabling Fastlink, glitches and link loss still do *not* occur.

Enable Fastlink by configuring the port(s) with the following two settings:

- *Fixed* speed/duplex mode, preferably *100 Mbit full-duplex*. That is, auto-negotiation of speed/duplex mode is disabled. See section 8.1.2 for information on port speed/duplex.

[2] Fastlink does not apply to Gigabit Ethernet ports, or to any SFP ports.

- *Fixed* MDI/MDIX crossover mode, i.e., auto-MDI/MDIX is disabled. See section 8.1.8 for information on port crossover mode.

In most use-cases auto-negotiation of speed-duplex and MDI/MDIX is still preferable, but enabling Fastlink can improve failover performance in some redundancy applications – we refer to this as the *fastlink* mode:

- RedFox Rail [50] bypass relay ports[50]: RedFox Rail routers equipped with a bypass relay are typically used in train backbones. The four backbone ports, two in each direction, are controlled by a relay, ensuring connectivity between routers on the backbone when one or more routers experience power-loss. The *fastlink* mode minimises disruption when the bypass relay changes state.
- Layer-2 redundancy: the *fastlink* mode can improve failover performance for various layer-2 redundancy mechanisms, e.g., when using static link aggregation (section 17).

Note
The *fastlink* mode requires more precise knowledge of cabling and devices used because all automatic detection is disabled. E.g., on the RedFox Rail[50] bypass relay ports, make sure to setup 100 Mbps Full-Duplex, with MDI/MDIX mode set to either:
- MDIX in both directions and crossover cables between switches, or
- MDI in one direction, MDIX in the other, with a straight cable

The latter case does however not work when a train car is turned 180°, but may be useful in other setups since straight cables are more commonplace.

8.1.10 Fallback default VID

The fallback default VLAN ID is generally unnecessary to configure.

The purpose of the fallback default-VID is to control what should happen with ”untagged” packets entering a port only configured ”tagged” on a set of VLANs. For more information on VLAN features and the VLAN related terms used throughout this section, see chapter 13.

Every port needs to have a ”default VID”. The default VID specifies the VLAN ID an ”untagged” packet should be associated with as it enters that port. A port’s default VID is determined as follows:

- If a port is associated ”untagged” with a VLAN, that VID will be the port’s default VID. E.g., if a port is associated ”untagged” to VID 10, the port will have VID 10 as its ”default VID”.
- If a port is *not* associated ”untagged” with any VLAN, the port’s default VID is determined as:
  - the port’s fallback default VID, given that a fallback default-VID is configured, or
  - the default VLAN (VID 1), if no fallback default-VID is configured

The fallback default VID can be used to control whether ”untagged” packets should be accepted on a port (only) associated ”tagged” with a set of VLANs. If the port’s default VID is represented within that set of VLANs, the packet will be accepted. Otherwise it will be dropped.

8.1.11 SFP DDM/DOM Diagnostics

Digital diagnostics monitoring (DDM), also known as digital optics monitoring (DOM), is a function enabling the user to monitor diagnostic parameters of the SFP.

WeOS provides diagnostic information for the following DDM parameters.

- Optical TX power
  Measures the optic power when transmitting, which can be used for detecting a deteriorating link[3]. The accuracy is better than +/-3dB and the total range of -40 to +8.2 dBm (0–6.5535 mW).
- Optical RX power
  Measures the optic power when receiving, which can be used for detecting a deteriorating link. The accuracy is better than +/-3dB and the total range of -40 to +8.2 dBm (0–6.5535 mW).
- Temperature
  The temperature of the SFP should be very close or equal to the temperature of the unit. The temperature accuracy is better than 3 degrees Celsius (°C) and the total range of -128 °C to +128 °C.

[3] By comparing the TxPower on a unit with the RxPower on the unit it is connected to, the user can find out the amount of "signal strength" that is lost over the optic link. When the gap between TxPower and RxPower is increasing, the optic link’s capability to transfer the signal decreases.

- Bias current
  The transmitting bias current can be used to determine how fast an SFP is aging. The accuracy is better than +/- 10% and the total range of 0 - 131 mA.
- Voltage The voltage should always be 3.3V since the SFP's power supply line is the same as the unit. The accuracy is better than +/-3% and the total range of 0–6.55 V.

DDM/DOM information will only be listed for enabled ports.

Note
DDM support in WeOS is limited to Westermo DDM SFPs, see the SFP Transceiver Datasheet of your WeOS product (www.westermo.com).

8.2 Managing port settings via the web interface

8.2.1 List Port Settings

Menu path: Configuration ⇒ Port ⇒ Port

When entering the port configuration page you will be presented to a list of all ports available on your switch, see fig. 8.2. Here you get an overview of the settings for all ports, and in addition two items of dynamic information - alarms and link status.

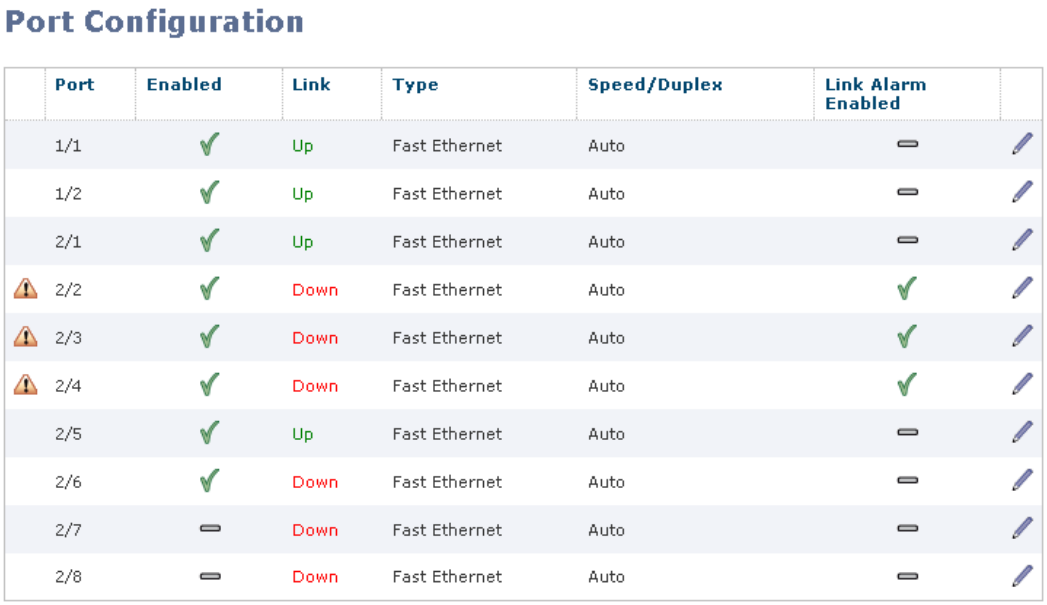


Figure 8.2: Port configuration settings overview (this example is from a RedFox Industrial switch)

Alarm	There is an active link alarm associated with the port. Only shown if link alarm is enabled and the link is down.
Port	The port label.
Enabled	Shows if the port is enable or disabled
Link	Link status for the port. Up or down.
Type	The port type: Gigabit Ethernet Fibre optic, Gigabit Ethernet, Fast Ethernet Fibre optic or Fast Ethernet.

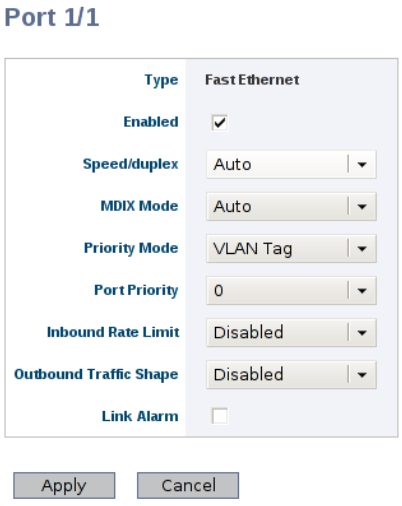
Continued on next page

	Continued from previous page
Speed/Duplex	The speed duplex setting. Auto means speed and duplex will be automatically negotiated. Otherwise the current setting will be shown as speed in Megabit and duplex as FDX for full duplex and HDX for half duplex. Note! This is not the negotiated speed, it is the configuration setting!
Link Alarm Enabled	When link alarm is enabled an alarm will be generated if port link is down. Alarms trigger an SNMP trap message to be sent and alarms to be shown on the administration web. In the ports overview table a green check-mark means enabled, and a dash means disabled.
Edit	Click this icon to edit a port's settings.

To change the settings for a specific port you will have to click the edit icon which will take you to the port setting edit page see section 8.2.2.

8.2.2 Edit Port Settings

Menu path: Configuration ⇒ Port ⇒ Port ⇒



On this page you can change the settings for the port.

Type	The port type: Gigabit Ethernet Fibre optic, Gigabit Ethernet, Fast Ethernet Fibre optic or Fast Ethernet.
Enable	Enable/disabled the port
Speed/Duplex	The speed duplex setting. Auto means speed and duplex will be automatically negotiated. Otherwise the current setting will be shown as speed in Megabit and duplex as FDX for full duplex and HDX for half duplex. Note! This is not the negotiated speed, it is the configuration setting!
MDIX mode	How to handle crossover cables. If you connect two units with different port settings (one with mdi and one with mdix) you need a straight-through twisted pair cabling. If you connect two units with the same setting you will need a crossover cabling. **Auto** Automatic detection **mdi** Medium dependent interface **mdix** mdi crossover
Priority Mode	Here you select on what information priority will be based: **Port Based** Based on the port's priority. See the next item (**Priority**). **IP** Based on the content of the IP ToS bits (IPv4) or the IP TC bits (IPv6). **VLAN Tag** Based on the content of the (802.1p) priority field inside the received packet's VLAN tag.
Priority	The port's priority level. Zero (0) is low priority and seven (7) high priority.
Inbound Rate Limit	Bandwidth limit for inbound traffic. *Disabled* means no limiting.
Outbound Traffic Shape	Bandwidth limit for outbound traffic. *Disabled* means no limiting.
Link Alarm	When link alarm is enabled an alarm will be generated if port link is down. Alarms trigger an SNMP trap message to be sent and alarms to be shown on the administration web.

8.2.3 List SFP DDM/DOM diagnostics

For information on how to view SFP DDM/DOM diagnostics, see section 4.4.3.

8.3 Managing port settings via the CLI

The *port* configuration context can be entered using the **”port <PORT|PORTLIST>”** command from the *Global Configuration* context. When providing a list of ports, the scope of the configuration commands becomes all ports in the list. There is also a specific command, **”ports”**, to enter the port context with the scope of *all Ethernet ports* of the device.

Command	Default	Section
port [eth|. . .] <PORTLIST>	Ethernet	Section 8.3.1
ports [eth|. . .]	Ethernet	Section 8.3.2
[no] enable	Enabled	Section 8.3.3
[no] speed-duplex <auto|10-half|10-full| 100-half|100-full|. . . >	auto	Section 8.3.4
[no] flow-control	Disabled	Section 8.3.5
[no] priority <0-7>	0	Section 8.3.6
[no] priority-mode <tag|ip|port>	tag	Section 8.3.7
[no] link-alarm	Disabled	Section 8.3.8
[no] rate-limit <64-1000000> [match <TYPE>[,<TYPE>,...]]	Disabled	Section 8.3.9
[no] traffic-shaping <<64-1000000>| <7700-1488000> fps>	Disabled	Section 8.3.10
[no] mdix-mode <auto|mdi|mdix>	auto	Section 8.3.11
[no] unshielded	Unshielded	Section 8.3.12
[no] low-power	Low Power	Section 8.3.13
[no] default-vid <VLAN_ID>	Disabled	Section 8.3.14
Show port status		
show ports		Section 8.3.15
Show SFP DDM/DOM diagnostics		
show environment		Section 7.3.50

8.3.1 Managing Ports

Syntax `port [eth|...] <PORT|PORTLIST>`
The **"port"** command is used for many port types, thus the full command syntax is
"port [eth|dsl|shdsl|xdsl|serial] <PORT|PORTLIST>".

Context Global Configuration context

Usage Enter Port Configuration context of the given PORT (or PORTLIST) and port type.

A **"PORTLIST"** is a comma separated list of ranges of ports without intermediate spaces, e.g., **"1/1,1/2"** on a *slotted* product, or **"1-3,5"** on a *non-slotted* product.

The port qualifier keyword **"eth|..."** is not needed if the numbers in the **"PORTLIST"** are unique to a single type of port. If there are multiple port with the same number (but different types), the port qualifier is needed, e.g., **"port eth 1"** and **"port dsl 1"**.

For information on using the **"port"** command to enter:

- xDSL Port Configuration context, see section 11.3.1.
- SHDSL Port Configuration context, see section 10.3.1.
- Serial port context, see section 38.3.1.

Use **"show port [eth|...] [PORT|PORTLIST]"** to list port configuration information on one or more ports. Also available as **"show"** command within thePort Configuration.

Default values Not applicable for configuration. For listing configuration **"show port"** information on all ports are listed by default.

Entering port configuration context for Ethernet ports 1-4:

Example

```
example:/config/#> port 1-4
example:/config/port-eth1-4/#>
```

This unit has two ports with number 1 (**"eth 1"** and **"dsl 1"**) thus the port qualifier is needed to determine which port to configure:

Example

```
example:/config/#> port 1
Ambiguous or bad port range or port type: 1
example:/config/#> port dsl 1
example:/config/port-dsl1/#>
```

8.3.2 Managing all Ports

Syntax `ports [eth|dsl|shdsl|xdsl|serial]`

Context Global Configuration context

Usage Enter *Port Configuration* context with the scope of all ports of a specific type (Ethernet, xDSL, etc.).

Default values ”Ethernet” for *configuration* (i.e., **”ports”** will enter Ethernet Port Configuration for all Ethernet ports), and ”All” for *showing configuration* (i.e., **”show ports”** will list information on all port types).

Listing information on all ports.

Example

```
example:/config/#> show ports
Ethernet ------------------------------------ Priority ---- Limit - Default
Port    Ena Aneg Speed  DPX  Flow  MDI/X  Alarm  Mode Level    In | Out     Vid
================================================================================
Eth 1   YES  YES  ---    -     NO   auto     NO  tag      0   None None    Auto
Eth 2   YES  YES  ---    -     NO   auto     NO  tag      0   None None    Auto
Eth 3   YES  YES  ---    -     NO   auto     NO  tag      0   None None    Auto
Eth 4   YES  YES  ---    -     NO   auto     NO  tag      0   None None    Auto
================================================================================
xDSL ----------------------------------------- Priority ---- Limit - Default
Port    Ena Mode Filter Encap PVC   Annex Alarm  Mode Level    In | Out     Vid
================================================================================
DSL 1   YES adsl    YES   llc 8/35      A    NO  tag      0   None None    Auto
================================================================================
Serial -------------------- Data ------- Stop RTS XON -----------------------
Port    Ena Type    Speed    bits Parity  bits CTS XOFF  Terminate
================================================================================
Ser 1   YES rs232   115200   8    None    1    OFF OFF   N/A
================================================================================
example:/config/#>
```

Listing information on a all ports of a specific type

Example

```
example:/config/#> show ports dsl
xDSL ------------------------------------------ Priority ---- Limit - Default
Port    Ena Mode Filter Encap PVC   Annex Alarm  Mode Level    In | Out     Vid
================================================================================
DSL 1   YES adsl    YES   llc 8/35      A    NO  tag      0   None None    Auto
================================================================================
example:/config/#>
```

8.3.3 Port enabling and disabling

Syntax `[no] enable`

Context Ethernet Port Configuration context (also available in SHDSL Port Configuration and xDSL Port Configuration for products with DSL ports)

Usage Use **"enable"** to enable and **"no enable"** disable a port.

Use **"show enable"** to show whether the port is enabled or disabled.

Default values Ports are enabled by default.

8.3.4 Speed and duplex setting

Syntax `[no] speed-duplex <auto|10-half|10-full|100-half|100-full|1000-half|1000-full>`

Context Ethernet Port Configuration context.

Usage Set port speed and duplex modes. **"auto"** means auto-negotiate, other modes are static configurations specifying 10, 100 or 1000 Mbit/s, and half or full duplex.

"no speed-duplex" will revert to default configuration for the speed-duplex setting, i.e., **"speed-duplex auto"**.

Use **"show speed-duplex"** to show the port's speed/duplex setting.

Default values `auto`

Error messages An attempt to set a port speed not available for this specific port type will render an error message, including information of available port speeds.

8.3.5 Flow-control setting

Syntax `[no] flow-control`

Context Ethernet Port Configuration context.

Usage Enable or disable IEEE 802.3 flow-control. For full duplex links, flow control will utilise IEEE 802.3 *pause frames*, and for half duplex links a technique known as *back-pressure* is used.

The flow control setting is only valid when the speed-duplex mode is set to **"auto"**, see section 8.1.2.

Use **"show flow-control"** to show the port's flow-control setting.

Default values Disabled (`no flow-control`)

8.3.6 Port priority setting

Syntax `[no] priority <0-7>`

Context Ethernet Port Configuration context (also available in SHDSL Port Configuration and xDSL Port Configuration for products with DSL ports)

Usage Set the (IEEE 802.1p) priority associated with the port. Packets coming in on this port will receive this priority unless priority is based on VLAN ID, VLAN tag or IP ToS/DiffServ bits.

"no priority" will revert to default configuration for the port priority setting, i.e., **"priority 0"** (zero).

Use **"show priority"** to show the port's priority setting.

Default values `0` (zero)

8.3.7 Set port priority mode

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

Context Ethernet Port Configuration context (also available in SHDSL Port Configuration and xDSL Port Configuration for products with DSL ports)

Usage Base priority classification for this port on content of VLAN tag (IEEE 802.1p priority bits), content of IP ToS/Diffserv bits, or the port priority configured for this port.

> **Note**
> VLAN priority settings (see section 13.4) will have precedence over port priority mode settings.

tag (Default) The packet's priority is based on the content of the VLAN tag (802.1p priority bits) of the incoming packet. For packets coming in *untagged*, the priority is based on the priority associated with the port, see section 8.3.6.

ip The packet's priority is based on the content of the IP ToS/Diffserv bit of the incoming packet. For non-IP packets coming in on the port (e.g., ARP packets), the priority is based on the priority associated with the port, see section 8.3.6.

port The packet's priority is based on the priority associated with the port, see section 8.3.6.

Use **"show priority-mode"** to show the port's "priority mode" setting.

Default values `tag`

8.3.8 Link alarm

Syntax `[no] link-alarm`

Context Ethernet Port Configuration context (also available in SHDSL Port Configuration and xDSL Port Configuration for products with DSL ports)

Usage Use **"link-alarm"** to enable and **"no link-alarm"** disable link-alarm for this port. When enabled, an alarm indication is activated when the link is down.

"show link-alarm" to show the port's link-alarm setting.

Default values Disabled (**"no link-alarm"**)

8.3.9 Inbound rate limiting

Syntax `[no] rate-limit <64-1000000> [match <TYPE>[,<TYPE>,...]]`

Context Ethernet Port Configuration context (also available in SHDSL Port Configuration and xDSL Port Configuration for products with DSL ports)

Usage Configure inbound rate limit in kbit/s. It is also possible use ISO modifiers k/M/G, e.g., 256k or 10M as specifiers for kbit/s and Mbit/s.

> **Note**
> Set values are rounded off to the nearest possible HW setting.

Optionally packet TYPE may be specified using one or more of the specifiers **"all"** (all types), **"bc"** (broadcast), **"mc"** (multicast) or **"u-uni"** (unknown unicast) in any combination. If no TYPE is specified (or if the specifier **"all"** is given) all packets will be rate limited.

> **Note**
> All WeOS products except RedFox and RedFox Industrial Rack support any combination of types. As stated in section 8.1.6.1, the traffic type selection on RedFox and RedFox Industrial Rack implicitly adds **"bc"** if **"mc"** is specified, and adds both **"bc,mc"** if **"u-uni"** is specified.

Use **"no rate-limit"** to disable inbound rate limiting.

Use **"show rate-limit"** to show the port's inbound rate limit setting.

Default values Disabled (**"no rate-limit"**)

8.3.10 Outbound traffic shaping

Syntax `[no] traffic-shaping <<64-1000000>|<7700-1488000> fps>`

Context Ethernet Port Configuration context (also available in SHDSL Port Configuration and xDSL Port Configuration for products with DSL ports, albeit not fps)

Usage Configure outbound traffic shaping in kbit/s or frames per second. It is also possible use ISO modifiers k/M/G, e.g., 256k or 10M as specifiers for kbit/s and Mbit/s.

> **Note**
> Set values are rounded off to the nearest possible HW setting.

Use **”no traffic-shaping”** to disable outbound traffic shaping.

Use **”show traffic-shaping”** to show the port’s outbound traffic shaping setting.

Default values Disabled (**”no traffic-shaping”**)

8.3.11 Cable crossover setting

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

Context Ethernet Port Configuration context.

Usage Configuration of Cable Crossover setting. **”auto”** means automatic crossover mode, **”mdix”** sets port to crossover mode (MDIX) and **”mdi”** sets port to MDI mode. This command is not valid for *fibre* ports.

”no mdix-mode” resets the MDIX mode to the default setting (**”auto”**).

Use **”show mdix-mode”** to show the port’s cable crossover setting.

Default values `auto`.

8.3.12 Adapting PHY Receiver to Shielded or Unshielded Cable

Syntax `[no] shielded`

Context Ethernet Port Configuration context.

Usage Fine tune the PHY receiver to the cable characteristics of shielded or unshielded TP cables. This setting applies to 10/100 Base-TX ports, excluding SFP/SFF ports as well as ports also capable of 1000 Mbit/s speeds.

Use **”shielded”** to adapt the PHY receiver to the use of shielded TP cables. Use **”no shielded”** to adapt the PHY receiver to the use of unshielded TP cables.

> **Note**
> This setting is only expected to be used by customers with special requirements - the default setting should be sufficient for most use cases.

Use **”show shielded”** to show the port’s ”shielded” setting.

Default values Unshielded (`no shielded`).

8.3.13 Enable/disable Low Power Mode on TX Data Signalling

Syntax `[no] low-power`

Context Ethernet Port Configuration context.

Usage It possible to select between two signal power modes on the Ethernet data signalling pins for 10/100 Base-TX ports. (This setting applies to 10/100 Base-TX ports, excluding SFP/SFF ports as well as ports also capable of 1000 Mbit/s speeds.)

The *low-power* mode is sufficient in most use cases, but for long cables or cables with specific characteristics it may be necessary to *disable low-power mode*.

Use **”low-power”** and **”no low-power”** respectively to enable/disable low-power mode on this Ethernet port.

> **Note**
> This setting is only expected to be used by customers with special requirements - the default setting should be sufficient for most use cases.

Use **”show low-power”** to show whether the PHY (TX Data Signalling) low-power mode is enabled or disabled.

Default values Low-Power (`low-power`).

8.3.14 Fallback default VLAN

Syntax `[no] default-vid <VLAN_ID>`

Context Ethernet Port Configuration context (also available in SHDSL Port Configuration and xDSL Port Configuration for products with DSL ports)

Usage Configuration of (fallback) default-VID for this port. The default-VID configuration is only valid when this port is not configured ”untagged” on any VLAN.

Use **”no default-vid”** to clear the (fallback) default VID setting (the default-VID setting will also be cleared whenever the port is associated ”untagged” with any VLAN).

When cleared (**”no default-vid”**), VLAN ID 1 will be used as the port’s fallback default-VID.

For more information see section 8.1.10.

Use **”show default-vid”** to show the port’s ”fallback default-VID” setting.

Default values Disabled/cleared (`no default-vid`).

8.3.15 Show port status (all ports)

Syntax `show ports`

Context Admin Exec context

Usage Show Port status information for all ports.

Default values Not applicable.

Chapter 9

Ethernet Statistics

A set of per port Ethernet statistic counters are available via the Web and via the CLI. Most of these counters correspond to standard SNMP MIB Ethernet statistics counters from the RMON MIB (RFC 2819), the Interface MIB (RFC 2863) and the Ether-Like MIB (RFC 3635)[1]. For more information about WeOS SNMP support, see chapter 6.

Section 9.1 gives a general introduction to the Ethernet statistic counters available via Web and CLI. Sections 9.2 and 9.3 present use of Ethernet statistics via the Web and CLI respectively.

9.1 Ethernet Statistics Overview

The table below provides a summary of the available Ethernet statistics counters. Sections 9.1.1-9.1.8 give more detailed information on their meaning.

Feature	Web	CLI	Description
Inbound			
Total Bytes	X	(X)[2]	Section 9.1.1
Bytes Good		X	-”-
Bytes Bad		X	-”-
Mean rate		X	-”-

Continued on next page

[1] The Ether-Like MIB is currently not supported in WeOS.

Continued from previous page

Feature	Web	CLI	Description
Total Good Packets		(X)[2]	Section 9.1.2
Unicast	X	X	-”-
Multicast	X	X	-”-
Broadcast	X	X	-”-
Pause frames		X	-”-
Size statistics	X		-”-
Dropped	X	X	Section 9.1.3
Filtered		X	-”-
Discarded		X	-”-
Erroneous		(X)[2]	Section 9.1.4
Undersize	X	X	-”-
Oversize	X	X	-”-
Fragments	X	X	-”-
Jabber	X	X	-”-
Checksum	X	X	-”-
PHY Error		X	-”-
Outbound			
Total Bytes	X	X	Section 9.1.5
Mean rate		X	”
Total Packets	(X)[2]	(X)[2]	Section 9.1.6
Unicast	X	X	-”-
Multicast	X	X	-”-
Broadcast	X	X	-”-
Pause frames		X	-”-
Dropped			Section 9.1.7
Filtered		X	-”-
Collisions and Busy Medium	X	(X)[2]	Section 9.1.8
Single		X	-”-
Multiple		X	-”-
Excessive		X	-”-
Late	X	X	-”-
Other collisions		X	-”-
Deferred		X	-”-

9.1.1 Inbound Byte Counters

A set of byte counters (i.e., octet counters) are provided. The number of *good* bytes is also used to compute a rough estimation of the current inbound data rate.

Bytes Good The number of *good bytes/octets* received on a port, i.e., the sum of the length of all good Ethernet frames received.

Bytes Bad The number of *bad bytes/octets* received on a port, i.e., the sum of the length of all bad Ethernet frames received.

Total Bytes The sum of good and bad bytes received on a port (see above). This would correspond to the RMON MIB *etherStatsOctets* and the Interface MIB *ifHCInOctets* objects.

Mean Rate Rough estimation of the current data rate based on the number of good bytes received during a time interval (2 seconds).

9.1.2 Inbound Counters of Good Packets

The following per port counters for *good* inbound Ethernet packets are provided.

Unicast packets The number of *good* packets with a unicast MAC address received on the port.

This would correspond to the Interface MIB *ifInUcastPkts* object.

Multicast packets The number of *good* packets with a group MAC address (excluding broadcast) received on the port.

This would correspond to the RMON MIB *etherStatsMulticastPkts* and the Interface MIB *ifInMulticastPkts* objects, except that *Pause frames* (see below) are not included.

Broadcast packets The number of *good* packets with a broadcast MAC address received on the port.

This would correspond to the RMON MIB *etherStatsBroadcastPkts* and the Interface MIB *ifInBroadcastPkts* objects.

Pause Frames The number of *good* flow control packets received.

[2] Counters listed within parenthesis (i.e., as '(X)') are provided implicitly.

Packet Size Statistics Counters for good Ethernet packet of the following size intervals are provided: 64 bytes, 65-127 bytes, 128-255 bytes, 256-511 bytes, 512-1023 bytes, and 1024-MAXPKTSIZE bytes, where MAXPKTSIZE is 1632.

These size intervals match the corresponding RMON statistics counters, except for the MAXPKTSIZE (1632 instead of 1518).

9.1.3 Dropped Inbound Packets

Counters for two types of dropped inbound packets are provided. Note, these packets are *good* Ethernet packets, but are dropped due to the reasons given below.

Filtered Inbound packets dropped due to VLAN mismatch or because the port was in LEARNING, LISTENING or BLOCKING state.

Discarded Packets dropped due to lack of buffer space.

9.1.4 Erroneous Inbound Packets

The following counters for received erroneous packets are provided:

Undersized packet Number of packets smaller than 64 bytes, and with a valid FCS.
This corresponds to the RMON MIB *etherStatsUndersizePkts* object.

Oversized packet Number of packets larger than 1632 bytes, and with a valid FCS.
This corresponds to the RMON MIB *etherStatsOversizePkts* object, except for the used MAXPKTSIZE (1632 instead of 1518 bytes).

Fragmented packet Number of packets smaller than 64 bytes, with an *invalid* FCS.
This corresponds to the RMON MIB *etherStatsFragments* object.

Jabber Number of packets larger than 1632 bytes, and with an *invalid* FCS.
This corresponds to the RMON MIB *etherStatsJabbers* object, except for the used MAXPKTSIZE (1632 instead of 1518 bytes).

Checksum/FCS Error Packets of valid length (64-1632), but with an incorrect FCS.

This corresponds to the RMON MIB *etherStatsCRCAlignErrors* object, except for the used MAXPKTSIZE (1632 instead of 1518 bytes).

PHY Error Signal Number of received packets generating a *receive error* signal from the Ethernet PHY. (Referred to as *InMacRcvErr* in the CLI port statistics list)

9.1.5 Outbound Byte Counters

A single outbound byte/octet counter, **Outbound Bytes**, is provided. It represents the sum of the length of all Ethernet frames sent on the port. This would correspond to the Interface MIB *ifHCOutOctets* object.

The number of **Outbound bytes** is also used to calculate a rough estimation of the current sending data rate (**Mean Rate**, i.e., the number of bytes sent during a time interval (2 seconds).

9.1.6 Outbound Packets Counters

The following per port counters for outbound Ethernet packets are provided.

Unicast packets The number of packets with a unicast destination MAC address sent on the port.

This would correspond to the Interface MIB *ifOutUcastPkts* object.

Multicast packets The number of packets with a group destination MAC address (excluding broadcast) sent on the port.

This would correspond to the Interface MIB *ifOutMulticastPkts* objects, except that *Pause frames* (see below) are not included.

Broadcast packets The number of packets with a broadcast destination MAC address sent on the port.

This would correspond to the Interface MIB *ifOutBroadcastPkts* objects.

Pause Frames The number of flow control packets sent.

9.1.7 Dropped Outbound Packets

The counter for a single type of dropped outbound packets is described here (there is also a second kind, see *excessive collisions* in section 9.1.8).

Filtered Outbound packets dropped outbound policy rules or because the port was in LEARNING, LISTENING or BLOCKING state.

9.1.8 Outbound Collision and Busy Medium Counters

The collision and busy medium counters described here are only relevant for half-duplex links.

Single Collisions The number of packets involved in a single collision, but then sent successfully.
This would correspond to the Ether-like MIB *dot3StatsSingleCollisionFrames* object.

Multiple Collisions The number of packets involved in more than one collision, but finally sent successfully.
This would correspond to the Ether-like MIB *dot3StatsMultipleCollisionFrames* object.

Excessive Collisions The number of packets failing (i.e., dropped) due to excessive collisions (16 consecutive collisions).
This would correspond to the Ether-like MIB *dot3StatsExcessiveCollisions* object.

Late Collisions The number of collisions detected later than a *512-bits time* into the packet transmission.
This would correspond to the Ether-like MIB *dot3StatsLateCollisions* object.

Other Collisions Other collisions than *single*, *multiple*, *excessive* or *late* collisions discovered on a port.

Total Collisions Computed as the sum of *single*, *multiple*, *excessive*, *late* and *other* collisions.

Deferred (busy medium) The number of packets experiencing a busy medium on its first transmission attempt, and which is later sent successfully, and without experiencing any collision.
This would correspond to the Ether-like MIB *dot3StatsDeferredTransmissions* object.

9.2 Statistics via the web interface

Statistics shown in the web administration tool has two views. An *overview* with a selection of statistics for all ports, including some status information (e.g. if port is blocking or forwarding), and a *detailed* page with a larger set of statistics.

Note that collection of statistics is started by the first access to the statistics page, and will be halted after a short period of time (to save resources) if no one requests the statistic data. This has the effect that you may need to enter the page once again, by e.g. clicking the menu item, to ensure you are presented to updated statistics data.

9.2.1 Statistics Overview

Menu path: Status⇒Port

On the port statistics overview page you will be presented to a selection of static data for each port. Additional statistic numbers are presented on the detailed view page.

Port Status and Statistics

Port	Link	State	Speed / Duplex	Total Bytes In	Total Bytes Out	FCS Errors	Details
1/1	Up	FORWARDING	100 FDX	111435680	304967	0	
1/2	Up	BLOCKING	100 FDX	113430677	1275	0	
2/1	Down	DISABLED		0	0	0	
2/2	Down	DISABLED		0	0	0	
2/3	Down	DISABLED		0	0	0	
2/4	Down	DISABLED		0	0	0	
3/1	Down	DISABLED		1676	1292	0	
3/2	Down	DISABLED		0	0	0	
3/3	Up	FORWARDING	100 FDX	0	1700512	0	
3/4	Up	BLOCKING	100 HDX	135413162	10530	4	
3/5	Down	DISABLED		0	0	0	
3/6	Down	DISABLED		0	0	0	
3/7	Down	DISABLED		0	0	0	
3/8	Down	DISABLED		0	0	0	

Auto refresh: Off, 5s, 15s, 30s, 60s

Refresh Clear All

Alarm	An alarm icon appears at the start of a line if there is a link alarm on a port.
Port	The port label.
Link	The status of the link. Up or down.
State	**FORWARDING** Unit forwards packets. Normal operation. **LEARNING** The port is preparing itself for entering FORWARDING state. **BLOCKING** Unit does not forward any packets. **DISABLED** Port does not participate in operation.
Speed / Duplex	The current speed and duplex negotiated or set on the port.
Total Bytes In	Total number of bytes received on the port.
Total Bytes Out	Total number of bytes sent out on the port.
FCS Errors	Total number of inbound packets with check sum error received on the port.
Details	Click this icon to view more detailed statistics for the port.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.
Clear All	Clear all statistics counters for all ports.

9.2.2 Detailed Statistics

Menu path: Status ⇒ Port ⇒ 

When clicking the *details*-icon in the overview page you will be presented to the detailed statistics page for the port.

Port Status and Statistics - Port 2/2

Link Status	Up

Traffic Counters		
	Inbound	Outbound
Total Bytes	37866325	38069876
Broadcast Packets	183551	183792
Multicast Packets	623	193
Unicast Packets	3456	3528
Dropped Packets	0	

Errors, Inbound		Traffic Size, Inbound	
Type	Packets	Octets	Packets
Fragments	0	64	42684
Oversize	0	65 -> 127	787
Undersize	0	128 -> 255	143520
Jabber	0	256 -> 511	43
Frame Checksum	0	512 -> 1023	596
		1024 -> Max	0

Errors, Outbound	
Type	Packets
Total Collisions	0
Single Collisions	0
Multiple Collisions	0
Excessive Collisions	0
Late Collisions	0
Other Collisions	0
Deferred	0
Filtered	0

Auto-Refresh: Off, 5s, 15s, 30s, 60s

<<Previous Next>> Refresh Clear Counters

Link Status	Status of link (Up/Down). If a link-alarm is associated with this port, an alarm icon is displayed if the link-alarm is active.
Total Bytes	Total number of bytes received (inbound) or transmitted (outbound) on this port.
Broadcast Packets	Total number of good broadcast packets received (inbound) or transmitted (outbound) on this port.
Multicast Packets	Total number of good multicast packets received (inbound) or transmitted (outbound) on this port.
Unicast Packets	Total number of good unicast packets received (inbound) or transmitted (outbound) on this port.
Dropped Packets	Total number of packets received that have been discarded.
Fragments	Total number of fragmented packets received on this port.
Oversize	Total number of oversized packets received on this port.
Undersize	Total number of undersized, but otherwise well formed, packets received on this port.
Jabber	Total number of packets received on this port larger than the network segment's maximum transfer unit (MTU).
Frame Checksum	Total number of packets received on this port with checksum error.
Traffic Size, Inbound	Number of octets received in different size categories.
Total Collisions	Total number of collisions detected on this port (sum of *single*, *multiple*, *excessive*, *late*, and *other* collision counters).
Single Collisions	The number of packets involved in a single collision, but then sent successfully.
Multiple Collisions	The number of packets involved in more than one collision, but finally sent successfully.
Excessive Collisions	The number of packets failing (i.e., dropped) due to excessive collisions (16 consecutive collisions).

Continued on next page

	Continued from previous page
Late Collisions	The number of collisions detected later than a *512-bits time* into the packet transmission.
Other collisions	Other collisions than *single*, *multiple*, *excessive* or *late* collisions discovered on a port.
Deferred	The number of packets experiencing a busy medium on its first transmission attempt, and which is later sent successfully, and without experiencing any collision.
Filtered	Outbound packets dropped outbound policy rules or because the port was in LEARNING, LISTENING or BLOCKING state.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
«Previous	Goto statistics for previous port.
Next»	Goto statistics for next port.
Refresh	Click on this button to reload with updated statistics.
Clear Port	Clear all statistics counters for the port shown.

9.3 Statistics via the CLI

The table below shows statistic features available via the CLI.

Command	Default	Section
rmon		Section 9.3.1
statistics [PORT]		Section 9.3.2
clear-stats [PORT]		Section 9.3.3
show rmon [PORT]		Section 9.3.4

9.3.1 Managing Ethernet Statistics

Syntax `rmon`

Context Admin Exec context

Usage Enter Ethernet statistics context (RMON Statistics context). WeOS starts gathering statistics when this command is issued, thus there is a 2 seconds delay before the RMON context is entered.

Default values Not applicable.

9.3.2 List Current Ethernet Statistics

Syntax `statistics [PORT]`

Context RMON Statistics context

Usage Show Ethernet statistics. If no PORT is given (**”statistics”**, a summary of statistics for all Ethernet ports is presented.

If a PORT is given as argument (e.g., **”statistics 1/1”**) detailed statistics for that port is presented.

For information about what the different statistics counters represent, see section 9.1.

Default values If no PORT argument is given, a summary of statistics for all Ethernet ports is presented.

9.3.3 Clear Ethernet Statistics

Syntax `clear-stats [PORT]`

Context RMON Statistics context

Usage Clear Ethernet statistic counters. If no PORT is given (**”clear-stats”**, counters for all Ethernet ports are cleared.

If a PORT is given as argument (e.g., **”clear-stats 1/1”**) the counters for that port are cleared.

Default values If no PORT argument is given, counters for all Ethernet ports are cleared.

9.3.4 Show Ethernet Statistics

Syntax `show rmon [PORT]`

Context Admin Exec context. Also available as **”show [PORT]”** command within the RMON Statistics context.

Usage Show Ethernet statistics. This command provides the same information as the **”statistics”** command (section 9.3.2). The only difference is that the **”show rmon [PORT]”** command is available from the Admin Exec context.

If no PORT is given (**”show rmon”**, a summary of statistics for all Ethernet ports is presented.

If a PORT is given as argument (e.g., **”show rmon 1/1”**) detailed statistics for that port is presented.

For information about what the different statistics counters represent, see section 9.1.

Default values If no PORT argument is given, a summary of statistics for all Ethernet ports is presented.

Chapter 10

SHDSL Port Management

Wolverine family switches (DDW225/DDW-226/DDW-142/DDW-142-485) are equipped with two SHDSL ports (Symmetric High-speed Digital Subscriber Line), enabling LAN networks to be extended over legacy copper cabling.

10.1 Overview of SHDSL Port Management

10.1.1 SHDSL overview

With SHDSL Ethernet LANs can be extended over legacy copper cabling. Switches can be connected in a simple point-to-point setup, but also in multi-drop and ring topologies, as shown in fig. 10.1.

In a SHDSL connection, the port on one unit shall be configured as *Central Office* (CO) and the port on the other unit as Customer Premises Equipment (CPE). SHDSL ports are named according to the name convention described in section 8.1.1). By default 1/1 or DSL 1 is configured as *CPE* while the 1/2 (or DSL 2) is configured as *CO*.

SHDSL support in WeOS is based on *Ethernet First Mile* (EFM) technology, and SHDSL can to a large extent be treated in the same way as Ethernet ports, e.g., you can add SHDSL ports to VLANs (chapter 13), you can run link-layer redundancy protocols such as FRNT (chapter 14) and RSTP (chapter 16) over them, etc. Settings specific to SHDSL ports are described in section 10.1.2 while port settings of more general nature is covered in section 10.1.3.

Feature	Web	CLI	General Description
CO/CPE mode selection	X	X	Section 10.1.1-10.1.2
DSL link rate	X	X	Section 10.1.1-10.1.2
DSL noise margin	X	X	Section 10.1.1-10.1.2
G.HS threshold	X	X	Section 10.1.2
PAF	X	X	Section 10.1.2
Low-Jitter	X	X	Section 10.1.2
EMF	X	X	Section 10.1.2
Settings in common with Ethernet ports			
Enable/disable port	X	X	Section 10.1.3
Port priority (level)	X	X	Section 10.1.3
Port priority mode	X	X	Section 10.1.3
Link alarm	X	X	Section 10.1.3
Inbound rate limit	X	X	Section 10.1.3
Outbound traffic shaping	X	X	Section 10.1.3
Fall-back default-VID		X	Section 10.1.3
View DSL port configuration	X	X	
View DSL port status	X	X	

10.1.2 Settings specific to SHDSL ports

- *Port role*: One unit shall be configured as *Central Office* (CO) and the other unit as Customer Premises Equipment (CPE). CO is the answering central unit. CPE (Customer Premises Equipment) is the unit that initiates the connection. In WeOS the SHDSL ports are named *1/1* and *1/2* in products with slot based numbering and DSL 1 and DSL 2 in products with simple port numbering: by default 1/1 (or DSL 1) is configured as *CPE* and 1/2 or DSL 2 configured as *CO*.

- *Data rate:* For a *regular* SHDSL connection, data rates can be achieved in the range from 192 kbit/s up to 5696 kbit/s depending on cable characteristics and communication distance. For products supporting *turbo-SHDSL*, data rates from 32 kbit/s up to 15304 kbit/s are possible. When using PAF in DDW-142 (and DDW-142-485), data rates up to 30608 kbit/s are possible.

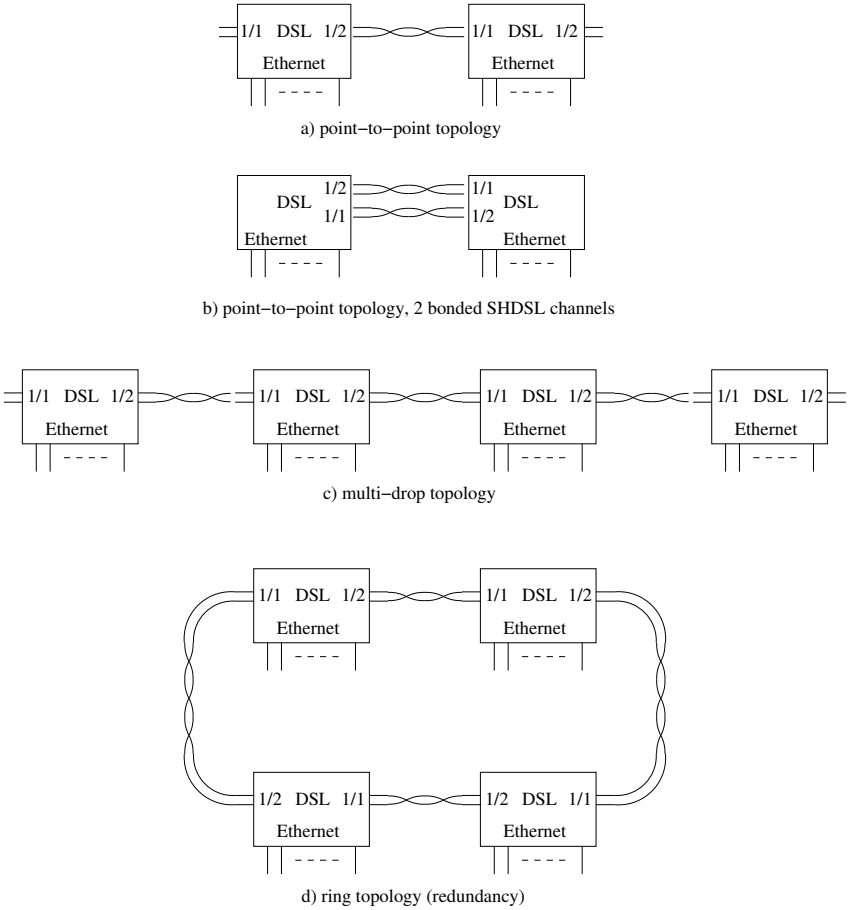


Figure 10.1: SHDSL topologies: Point-to-point (a), point-to-point, 2 bonded SHDSL channels (b), multi-drop (c) and ring (d).

Products with Turbo-speed support

Turbo-speed is supported on *all DDW-226* and *DDW-142/DDW-142-485 devices*, and on all but the earliest DDW-225 devices. To see if your *DDW-225* unit supports turbo-speed SHDSL, inspect its article number, either by reading its attached label, or remotely by viewing the **”Status ⇒ System”** Web page or by using the **”show system-information”** command in the CLI. If the article number says ”3642-0230” the product lacks turbo-speed support. If it says ”3642-0250”, the product supports turbo-speed.

Turbo-speed data rates can only be achieved if the SHDSL devices at both ends of the connection have turbo-speed support.

The operator can either specify a fixed data rate to be used, or let the CO and CPE discover the achievable data rate automatically.

Using *Auto* mode will optimise the data rate for the current SNR conditions.

- *Noise margin*: The noise margin is the difference between the required SNR for a certain bit rate, and the actual SNR.

  When the SHDSL connection data rate is set to auto-negotiation mode, the operator can configure an *administrative noise margin* (also referred to as *target noise margin* or *target SNR margin*). A large *administrative noise margin* gives robustness against SNR fluctuations. But as the *required SNR* increases with data rate, specifying a a large *administrative noise margin* may imply that a low data rate is negotiated.

  Thus, when configuring the *administrative noise margin* the operator can optimise the connection for *reliability* (noise margin 10dB), *high speed* (noise margin 3dB) or as a tradeoff thereof (*normal* mode, i.e., noise margin 6dB).

  To monitor the quality of the connection, WeOS enables the operator to read the *current noise margin*.

- *G.HS Threshold*: The G.HS threshold setting is only needed if the units are located in a noisy environment with SHDSL line cables of good quality, and where a connection can not even be established at SHDSL rate 192kbit/s. The setting configures a higher threshold of the G.HS idle parameter in order to detect idle. The SHDSL line length capability will be affected, since the G.HS idle threshold and the G.HS signals meet earlier when the G.HS Threshold is raised.

  When enabling GHS threshold, possible settings include ’low(750)’, ’medium (1500)’, ’high(3000)’ and a custom configured value.

  Corresponding values to the fixed value settings are [low-750; medium-1500; high-3000]. The custom configured value could be set in the range [0-32767] in steps of 1.

- *PAF - PME Aggregation Function:* PAF functionality is used to aggregate the 2 SHDSL ports on DDW-142 (and DDW-142-485) to achieve higher bandwidth. The 2 "bonded" ports can reach rates from 64 kbit/s to 30,6 Mbit/s.

- *Low Jitter function:* Low Jitter is a SHDSL port specific function that can be used in applications where high accuracy of the Ethernet packet jitter is needed. If enabled the jitter of the latency over the SHDSL link will be minimized.

This functionality is using a different SHDSL mode compared to default setting, thus the Low Jitter configuration must be set on both SHDSL ports sharing the physical cable.

- *EMF - Emergency Freeze function:* EMF enabled makes the unit detect exception situations on the SHDSL links. The detection will freeze the SHDSL transceiver parameters temporarily to keep the link up. With this function enabled the unit might avoid a complete SHDSL retrain that could take up to a minute. The unit may lose data even with EMF enabled, but only for a short period of time.

Note
Only the data rate and noise margin settings of the CO are used in the SHDSL connection. These parameters are passed to the CPE during the connection establishment phase.

10.1.3 General port settings

The following parameters can be configured for SHDSL ports in the same way as for Ethernet ports. The SHDSL uses Ethernet First Mile (EFM) encapsulation, thus many Ethernet settings apply to the SHDSL ports. More detailed information is found in chapter 8.

- *Port enable/disable:* Ports can be disabled and enabled administratively.
- *Port priority mode:* Define whether incoming packets should be prioritised based on VLAN tag, VLAN ID, port ID, IP ToS, etc. See also section 8.1.4.
- *Port priority (level):* The inbound priority associated with this port. See also section 8.1.4.
- *Link alarm:* Link status can be configured as an alarm source. See also section 8.1.5.
- *Inbound rate limit:* Setting the inbound rate limit is possible on DSL ports, but is likely of less interest than on Ethernet ports, since the DSL data rates are primarily limited by the rate of the DSL line. See also sections 8.1.6 and 10.1.2.
- *Outbound traffic shaping:* Setting the outbound rate limit (traffic shaping) is possible on DSL ports, but is likely of less interest than on Ethernet ports, since the DSL data rates are primarily limited by the rate of the DSL line.

Furthermore, outbound traffic shaping in *frames per second* mode is not availble on DSL ports. See also sections 8.1.7 and 10.1.2.

- *Fall-back default-VID*: The fall-back default VID setting is only of interest for the special case when *untagged* packets are received over a link only associated with *tagged* VLANs.

Ethernet settings for *port speed/duplex* mode, and *MDI/MDIX* mode do not apply to SHDSL ports, thus are not configurable.

Note
As of WeOS v4.17.1, enabling/disabling flow control (as described in section section 8.1.3) has no effect on SHDSL ports.

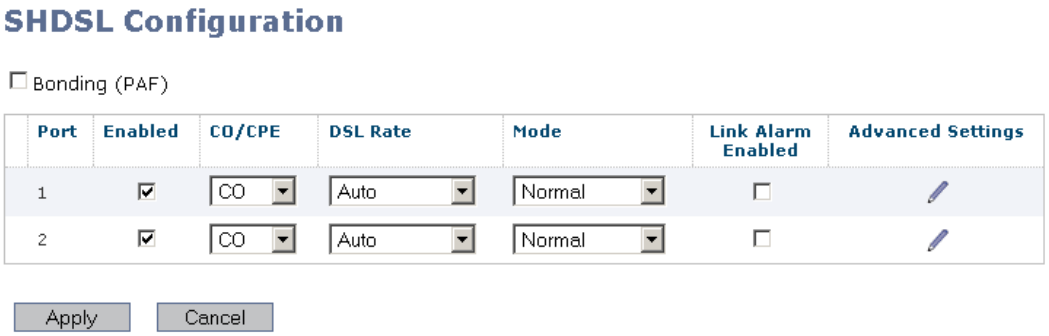
10.2 Managing SHDSL ports via the web interface

The Web interface provides configuration of SHDSL ports as well as listing of SHDSL port statistics.

The SHDSL statistics is provided in two views – an *overview* with a selection of statistics for all SHDSL ports, including some status information, and a *detailed* page with a larger set of statistics.

10.2.1 List and Edit SHDSL Port Settings

Menu path: Configuration ⇒ Port ⇒ SHDSL



On this page you can list and change the settings for the SHDSL ports.

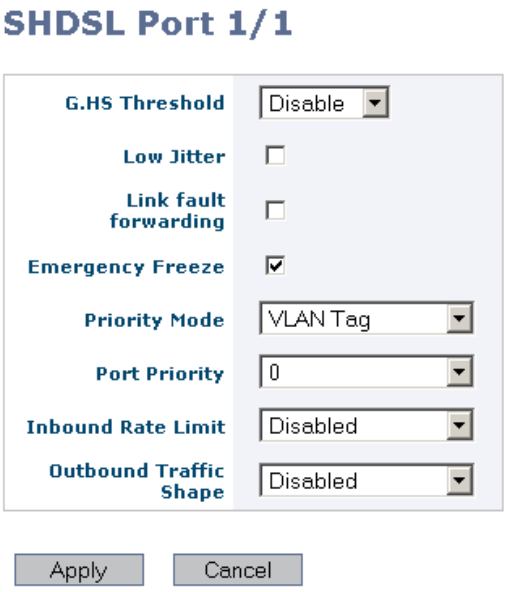
PAF	PAF aggregates the 2 SHDSL ports to achieve higher bandwidth. The functionality demands that the rate do not differ more the 4 times between port 1 and 2 to ensure good performance. Note: This functionality is only available on DDW-142 and DDW-142-485. Check to enable, un-check to disable. Default is **`Disabled`**.
Port	The SHDSL port label.
CO/CPE	To establish a connection between two DSL-ports, one has to be configured as Central Office (CO) and one has to be configured as Customer Premises Equipment (CPE). Default for port 1/1 is *CPE*, and default for port 1/2 is *CO*.

Continued on next page

	Continued from previous page
DSL Rate	Speed setting is only valid if the port is configured as CO (the CPE rate setting is not used, since the CPE speed automatically follows the CO to which it becomes connected). See section 10.1.2 for information on using SHDSL-turbo speed data rates. Default is **`Auto`**.
Mode	The *noise-margin mode*. The *noise-margin mode* setting is only valid if the port is configured as CO (the CPE setting is not used, since the CPE *noise-margin mode* automatically follows the CO to which it becomes connected). The CO can be configured to choose a faster less reliable speed (**`High Speed`**), a slower more reliable speed (**`Reliable`**), or a tradeoff between these two objectives (**`Normal`**). Default is **`Normal`**.
Link Alarm	When link alarm is enabled an alarm will be generated if port link is down. Alarms trigger an SNMP trap message to be sent and alarms to be shown on the administration web.
 **Edit**	Click this icon to edit a port's settings.

10.2.2 Edit Port Settings

Menu path: Configuration ⇒ Port ⇒ SHDSL ⇒ PortNo ⇒ 



On this page you can change the settings for the port.

G.HS Threshold	The G.HS Threshold setting is only needed if the unit are located in a noisy environment with SHDSL line cables of good quality and where a connection can not even be established at SHDSL rate 192kbit/s. The setting configures a higher threshold of the G.HS idle parameter in order to detect idle. The SHDSL line length capability will be affected, since the G.HS idle threshold and the G.HS signals meet earlier when the G.HS Threshold is raised. When enabling G.HS Threshold, possible settings include 'low', 'medium' and 'high'. Corresponding values to the fixed value settings are [low-750; medium-1500; high-3000] If a custom value is configured in CLI, it will be displayed in the drop-down list. Default is **Disabled**

Continued on next page

Continued from previous page

Low Jitter	The Low Jitter mode can be enabled to minimize the jitter of the latency over the SHDSL link in applications where high accuracy of the Ethernet packet jitter is needed. This functionality is using a different SHDSL mode compared to default setting, thus the Low Jitter configuration must be set on both SHDSL ports sharing the physical cable. Check to enable, un-check to disable. Notice: Make sure that you have both line partners configured enabled or disabled. Default is **`Disabled`**.
Link Fault Forward (LFF)	On devices with SHDSL ports, alarms can be triggered when the remote SHDSL switch indicates it has link down on its Ethernet port. That is, this feature can be used in topologies where an Ethernet is extended over an SHDSL link, and where the remote SHDSL switch (e.g. a DDW-120) is able to signal that the Ethernet link is down on its side. Check to enable, un-check to disable. Default is **`Disabled`**.
Emergency Freeze EMF	EMF enabled makes the unit detect exception situations on the SHDSL links. The detection will freeze the SHDSL transceiver parameters temporarily to keep the link up. With this function enabled the unit might avoid a complete SHDSL retrain that could take up to a minute. The unit may lose data even with this functionality enabled, but only for a short period of time. Check to enable, un-check to disable. Default is **`Enabled`**.
Priority Mode	Here you select on what information priority will be based: **Port Based** — Based on the port's priority. See the next item (**Priority**). **IP** — Based on the content of the IP ToS bits (IPv4) or the IP TC bits (IPv6). **VLAN Tag** — Based on the content of the (802.1p) priority field inside the received packet's VLAN tag.

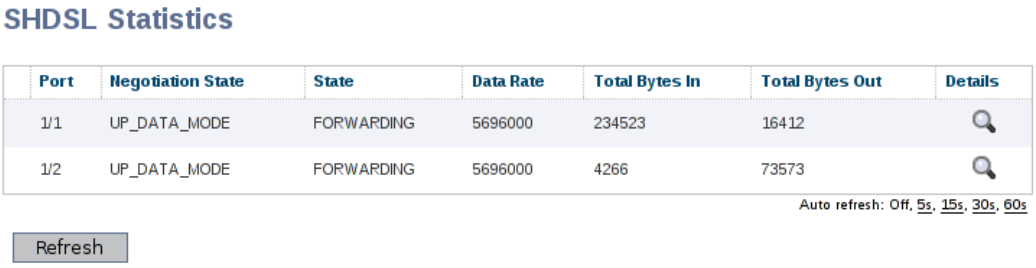
Continued on next page

	Continued from previous page
Priority	The port's priority level. Zero (0) is low priority and seven (7) high priority.
Inbound Rate Limit	Bandwidth limit for inbound traffic. *Disabled* means no limiting.
Outbound Traffic Shape	Bandwidth limit for outbound traffic. *Disabled* means no limiting.

10.2.3 SHDSL statistics Overview

Menu path: Status ⇒ Port ⇒ SHDSL

On the SHDSL port statistics overview page you will be presented to a selection of static data for each port. Additional statistic numbers are presented on the detailed view page.

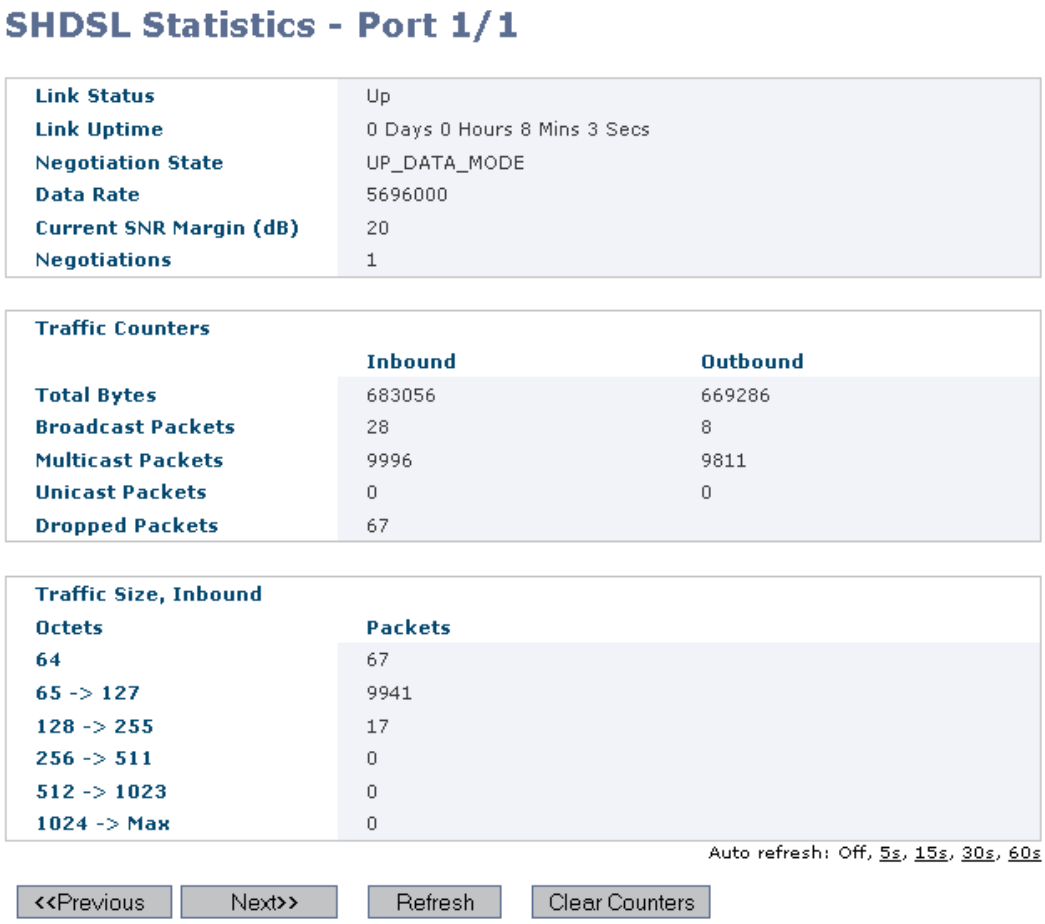


Alarm	An alarm icon appears at the start of a line if there is a link alarm on a port.
Port	The port label. If PAF is configured, the background color of the port identifier is pink
Negotiation State	Current state of the DSL-line negotiation. Possible values are UP_DATA_MODE, INITIALISING, DOWN_READY and DOWN_NOT_READY. Note: if no link is established the normal state for a CO-mode configured port is DOWN_NOT_READY, for a CPE-configured port the normal state is DOWN_READY.
State	**FORWARDING** Unit forwards packets. Normal operation. **LEARNING** The port is preparing itself for entering FORWARDING state. **BLOCKING** Unit does not forward any packets. **DISABLED** Port does not participate in operation.
Data Rate	Negotiated DSL data rate in bit/s.
Total Bytes In	Total number of bytes received on the port.
Total Bytes Out	Total number of bytes sent out on the port.
Details	Click this icon to view more detailed statistics for the port.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.

10.2.4 Detailed SHDSL Port Statistics

Menu path: Status ⇒ Port ⇒ SHDSL ⇒ 

When clicking the *details*-icon in the overview page you will be presented to the detailed statistics page for the SHDSL port.



Link Status	Status of link, (Up/Down). If a link-alarm is associated with this port, an alarm icon is displayed if the link-alarm is active.
Link Uptime	The time since link was established.

Continued on next page

	Continued from previous page
Negotiation State	Current state of the DSL-line negotiation. Possible values are UP_DATA_MODE, INITIALISING, DOWN_READY and DOWN_NOT_READY. Note: if no link is established the normal state for a CO-mode configured port is DOWN_NOT_READY, for a CPE-configured port the normal state is DOWN_READY.
Data Rate	Negotiated DSL data rate in bit/s.
Current SNR Margin	Signal to Noise Ratio in dB on this link.
Negotiations	Number of negotiations since unit startup.
Total Bytes	Total number of bytes received (inbound) or transmitted (outbound) on this port.
Broadcast Packets	Total number of good broadcast packets received (inbound) or transmitted (outbound) on this port.
Multicast Packets	Total number of good multicast packets received (inbound) or transmitted (outbound) on this port.
Unicast Packets	Total number of good unicast packets received (inbound) or transmitted (outbound) on this port.
Dropped Packets	Total number of packets received that have been discarded.
Traffic Size, Inbound	Number of octets received in different size categories.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
<<Previous	Goto statistics for previous port.
Next>>	Goto statistics for next port.
Refresh	Click on this button to reload with updated statistics.
Clear Port	Clear all statistics counters for the port shown.

10.3 Managing SHDSL ports via the CLI

The table below shows SHDSL port management features available via the CLI.

Command	Default	Section
Configure SHDSL port settings		
port [dsl|shdsl|...] <PORTLIST>		Section 10.3.1
[no] co		Section 10.3.2
[no] speed <auto|auto-5696k|0-15304k>	Auto	Section 10.3.3
[no] noise-margin	Normal	Section 10.3.4
[no] ghs-threshold <low|medium|high>	Disabled	Section 10.3.5
[no] paf	Disabled	Section 10.3.6
[no] low-jitter	Disabled	Section 10.3.7
[no] emf	Enabled	Section 10.3.8
Port settings in common with Ethernet ports (chapter 8)		
[no] enable	Enabled	Section 8.3.3
[no] priority <0-7>	0	Section 8.3.6
[no] priority-mode <tag|ip|port>	tag	Section 8.3.7
[no] link-alarm	Disabled	Section 8.3.8
[no] rate-limit <70-2560>	Disabled	Section 8.3.9
[no] traffic-shaping <70-2560>	Disabled	Section 8.3.10
[no] default-vid <VLAN_ID>	Disabled	Section 8.3.14
Show SHDSL related status and statistics		
show <dsl|shdsl>		Section 10.3.9
show ports		Section 8.3.15
show rmon		Section 9.3

10.3.1 Managing SHDSL port settings

Syntax `port [dsl|shdsl|...] <PORTLIST>`

Context Global Configuration context

Usage Enter the SHDSL Port Configuration context.

A **”PORTLIST”** is a comma separated list of ranges of SHDSL ports without intermediate spaces, e.g., **”1/1,1/2”** on a *slotted* product, or **”1-3,5”** on a *non-slotted* product.

The port qualifier keyword **”shdsl”** (or **”dsl”**) is not needed if the numbers in the **”PORTLIST”** are unique to DSL ports.

For a more general description of the **”port”** command, see section 8.3.1.

Use **”show port [dsl|shdsl] [<PORT|PORTLIST>]”** port configuration information of the given PORT or PORTLIST. Alternatively, the command **”show”** can be run within the SHDSL Port Configuration context, to show the configuration of a port (or list of ports).

Default values Not applicable.

10.3.2 Setting SHDSL port mode (CO/CPE)

Syntax `[no] co`

Context SHDSL Port Configuration context

Usage Set the SHDSL port to operate in *central office* (CO) or *customer premises equipment* (CPE) mode.

When connecting switches via SHDSL it is important that one side puts its SHDSL port in CO mode (**”co”**) while the other side puts its SHDSL port in CPE mode (**”no co”**).

Default values Factory default for DDW-225/226 is to have port 1/1 in *CPE* mode (**”no co”**), and port 1/2 in *CO* mode (**”co”**). Factory default for DDW-142 (and DDW-142-485) is to have port DSL 1 in *CPE* mode (**”no co”**), and port DSL2 in *CO* mode (**”co”**).

Use **”show co”** to show whether the SHDSL port is configured to operate as *Central Office* or *Customer Premises Equipment*.

10.3.3 Setting SHDSL port rate

Syntax `[no] speed <auto|auto-5696k|0-5696k|0-15304k>`

Context SHDSL Port Configuration context

Usage Set SHDSL port rate, either bye specifying that auto-negotiation should be used, or that a specific fixed rate should be used. Only the **”speed”** setting on the CO has affect on the established connection.

- *Auto-negotiate:* Use **”speed auto”**, **”speed 0”**, or **”no speed”** to let the rate be auto-negotiated between the SHDSL nodes in the extended SHDSL range 32-15288 kbps on Turbo HW; if not Turbo HW the range is 192-5696 kbps.

  Use **”speed auto-5696k”** to let the rate be auto-negotiated in the standard SHDSL range 192-5696 kbit/s.

- *Fixed rate:* Use **”speed RATE”**, where RATE is in range **”1k-15304k”** on products *with* Turbo-HW support, and in range **”1k-5096k”** on products *without* Turbo-HW support, to specify a fixed data rate in kbit/s. Alternatively, specify **”speed 1-15304000”** and **”speed 1-5696000”** respectively, to specify a fixed data rate in bit/s.

  The following fixed rates are supported on all SHDSL products: 192k, 384k, 512k, 768k, 1024k, 1280k, 2048k, 2304k, 2688k, 3072k, 3456k, 3840k, 4224k, 4608k, 4992k, 5376k, and 5696k.
  Products with Turbo-HW support the following additional fixed data rates: 32k, 64k, 128k, 6200k, 6712k, 7224k, 7736k, 8248k, 8760k, 9272k, 9784k, 10296k, 10808k, 11320k, 11832k, 12344k, 13112k, 13880k, 14648k and 15304k.

  If other rates are specified, WeOS will round the value upwards to the nearest supported rate.

  Use **”show speed”** to show the SHDSL port’s rate setting.

Default values **”speed auto”**

10.3.4 Setting SHDSL port noise-margin

Syntax `[no] noise-margin <reliable|normal|high-speed [nonstrict]>`

Context SHDSL Port Configuration context

Usage Set SHDSL port *noise-margin*. *Note:* The noise-margin setting is only relevant when the data rate is set to *auto-negotiate* (**”rate 0”**), see section 10.3.3).

Available noise-margin modes:

- *Reliable:* Select **”noise-margin reliable”** to let the rate auto-negotiation optimise for reliability (rather than high data rate).
- *High-Speed:* Select **”noise-margin high-speed”** to let the rate auto-negotiation optimise for high data rate (rather than reliability).
- *Normal:* **”noise-margin normal”** is the default setting for the noise-margin, which gives a tradeoff between reliability and high-speed. Alternatively, the command **”no noise-margin”** can be used.

  Using the parameter *nonstrict* after the selected noise-margin mode will configure the unit to a less strict algorithm during the connection phase. The resulting current SNR will not necessary match the configured noise-margin mode.

Use **”show noise-margin”** to show the SHDSL port’s noise-margin setting.

Default values **”noise-margin normal”**

Error messages None defined yet.

10.3.5 Setting SHDSL port G.HS Threshold

Syntax `[no] ghs-threshold <low|medium|high>`

Context SHDSL Port Configuration context

Usage Set SHDSL port to operate with new *G.HS Threshold* value. The G.HS Threshold setting is only needed if the unit are located in a noisy environment with SHDSL line cables of good quality and where a connection can not even be established at SHDSL rate 192kbit/s, see section 10.1.2. The setting configures a higher threshold of the G.HS idle parameter in order to detect idle. The SHDSL line length capability will be affected, since the G.HS idle threshold and the G.HS signals meet earlier when the G.HS Threshold is raised.

When enabling G.HS Threshold, possible settings include**”low”**, **”medium”**, **”high”** and a custom configured value. Corresponding values to the fixed value settings are [low-750; medium-1500; high-3000].

The custom configured value could be set in the range [0-32767] in steps of 1.

Use **”no ghs-threshold”** to disable the G.HS threshold.

Use **”show ghs-threshold”** to show the SHDSL port’s G.HS Threshold setting.

Default values Disabled (**”no ghs-threshold”**)

10.3.6 Setting SHDSL PAF mode

Syntax `[no] paf`

Context SHDSL Port Configuration context

Usage Set the SHDSL unit to operate in *paf* mode.
PAF aggregates the 2 SHDSL ports to achieve higher bandwidth. The functionality demands that the rate do not differ more than 4 times between port 1 and 2 to ensure good performance. Port 2 must be configured to the same role (CO/CPE) as port 1 to get the functionality working.

> **Note**
> This functionality is only available on DDW-142 and DDW-142-485.

Use **”show paf”** to show whether PAF is enabled or not.

Default values Disabled

10.3.7 Setting SHDSL low jitter mode

Syntax `[no] low-jitter`

Context SHDSL Port Configuration context

Usage Set the SHDSL unit to operate in *low jitter* mode.
Low Jitter can be enabled to minimize the jitter of the latency over the SHDSL link in applications where high accuracy of the Ethernet packet jitter is needed. This functionality is using a different SHDSL mode compared to default setting, thus the Low Jitter configuration must be set on both SHDSL ports sharing the physical cable.

Use **”show low-jitter”** to show the SHDSL port’s low-jitter setting.

Default values Disabled

10.3.8 Setting SHDSL emergency freeze mode

Syntax `[no] emf`

Context SHDSL Port Configuration context

Usage Set the SHDSL unit to operate in *emf* mode.
EMF enabled makes the unit detect exception situations on the SHDSL links. The detection will freeze the SHDSL transceiver parameters temporarily to keep the link up. With this function enabled the unit might avoid a complete SHDSL retrain that could take up to a minute. The unit may lose data even with this functionality enabled, but only for a short period of time.

Use **”`show emf`”** to show the SHDSL port’s emergency freeze setting.

Default values Enabled

10.3.9 Show SHDSL port status

Syntax `show shdsl`

Context Admin Exec context.

Usage Show the status of all SHDSL ports.

Default values Not applicable.

Chapter 11

ADSL/VDSL Port Management

The Falcon-206 is equipped with a xDSL port, i.e., a port capable of operating in either ADSL or VDSL mode. Thus, the Falcon-206 can be used as customer premises equipment (CPE), acting either as switch or router, when connecting to an ISP over an ADSL or VDSL line.

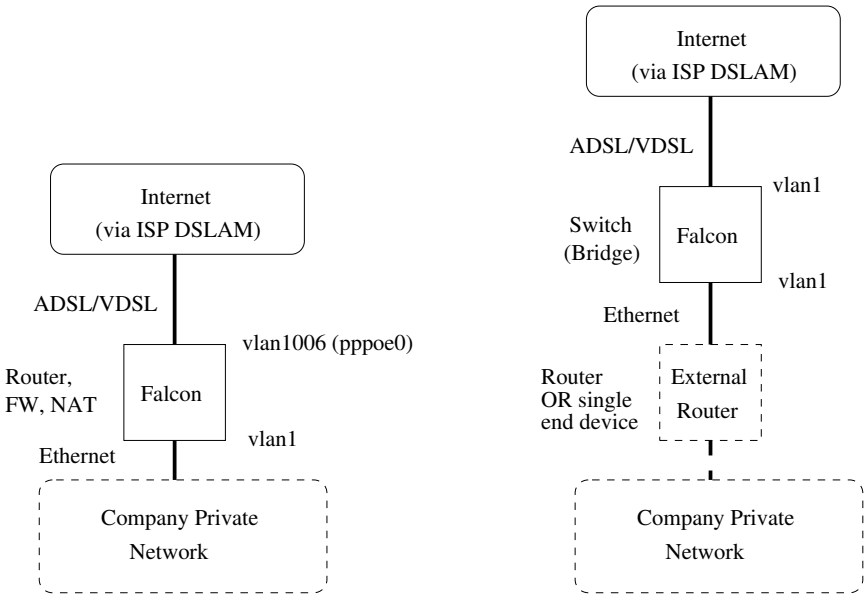
This chapter describes how to setup and manage your xDSL port, as well as the most common configuration steps to connect to your ISP.

11.1 Overview of ADSL/VDSL Port Management

Feature	Web	CLI	General Description
xDSL mode (ADSL/VDSL)	X	X	Section 11.1.1
xDSL carrier (POTS/ISDN)	X	X	Section 11.1.1
External splitter (filter)	X	X	Section 11.1.1
ADSL/ATM specific settings			
ATM VPI/VCI	X	X	Section 11.1.1-11.1.2
ATM Encapsulation	X	X	Section 11.1.1-11.1.2
Restart/retrain xDSL link	X	X	
View xDSL port configuration	X	X	
View xDSL port status/statistics	X	X	
xDSL settings in common with Ethernet ports	X	X	Section 11.1.3
ISP and network settings	X	X	Section 11.1.1, 11.1.4

11.1.1 ADSL/VDSL overview

A Falcon xDSL router is typically used as a *broadband router* (fig. 11.1a, when connecting a private company network to the Internet via xDSL. An alternative is to use Falcon as a *xDSL/Ethernet bridge* (fig. 11.1b), to connect a single PC or an external (non-"xDSL capable") router to the Internet.



a) Using Falcon as broadband router

b) Using Falcon as switch (bridge)

Figure 11.1: Common ADSL/VDSL topologies: a) Using Falcon as broadband router, or b) using Falcon as on xDSL/Ethernet switch (bridge).

When connection your Falcon xDSL unit to your ISP, you may have to configure settings related to the *xDSL port* as well as IP settings specific to your xDSL provider. To configure your Falcon router for the first time, it is recommended to use the Web based *Basic Setup Page*, see section 11.2.1.

More information on xDSL settings is found below and in sections 11.1.2-11.1.4.

- **xDSL settings:**
  - *ADSL or VDSL:* As the Falcon can be used both for ADSL and VDSL connections, you may have to configure the xDSL mode.

    Default: **ADSL**

– *POTS or ISDN carrier:* Depending on the kind of telecom network used to carry your xDSL connection, you should configure the *Annex* setting accordingly: **”Annex A, I, L, and/or M”** for POTS carrier networks, or **”Annex B or J”** for ISDN carrier networks.

  Further details on configuration of the *Annex* setting:

  * **Annex setting for ADSL over POTS:** For ADSL over POTS carrier, ”Annex A” is base annex implicitly available. Other annexes for POTS (I, L, M, N) are extensions to Annex A. You can specify to only use Annex A, or you can specify to use additional extensions:

    - **”Annex A”**: The WeOS unit announces capability of Annex A (ADSL over POTS). This is the default setting of ADSL.
    - **”Annex I”**: The WeOS unit announces capability of Annex A and I. Annex I allows for additional encoding techniques for ADSL over POTS.
    - **”Annex L”**: The WeOS unit announces capability of Annex A and L. Annex I allows for additional frequency bands (at lower POTS frequency), and longer reach.
    - **”Annex M”**: The WeOS unit announces capability of Annex A and M. Annex I allows for additional frequency bands (at higher POTS frequency).
    - **”Annex L-M”**: The WeOS unit announces capability of Annex A, L and M.

  * **Annex setting for ADSL over ISDN:** For ADSL over ISDN carrier, ”Annex B” is base annex implicitly available. Annex J is an extension to Annex B for ADSL over ISDN. You have the following configuration options:

    - **”Annex B”**: The WeOS unit announces capability of Annex B (ADSL over ISDN).
    - **”Annex J”**: The WeOS unit announces capability of Annex B and J. Annex I allows for additional encoding techniques for ADSL over ISDN.

  * **Annex setting for VDSL:** For VDSL it is possible to let the WeOS unit automatically probe what carrier network is used (by choosing **”Annex A-B”**); if this does not work to bring up the VDSL line, one

can manually try the individual settings **”Annex A”** and **”Annex B”** respectively.

Default: **ADSL Annex A** (ADSL over POTS)

– *Use of external Filter (Splitter) or not:* If you wish to use your xDSL connection for regular phone calls, the Falcon xDSL port should (1) be connected to a splitter which in turn connects to the (first) telephone jack, and (2) the Falcon xDSL **”filter”** setting should be *enabled*.

Otherwise, the Falcon xDSL port should (1) be connected directly to the (first) telephone jack, and (2) the Falcon xDSL **”filter”** setting should be *disabled*.

Default: **Filter enabled** (i.e., it is assumed the Falcon xDSL port is connected via a *splitter*)

- **ADSL specific settings:** When using Falcon for ADSL (as opposed to VDSL), a few settings related to ADSL/ATM encapsulation and VPI/VCI may have to be set, see section 11.1.2 below.

- **Use of PPPoE, DHCP or Static IP address assignment:** xDSL providers use different schemes to assign IP addresses to their customers. These methods to assign an IP address are not specific to xDSL connections, thus are explained in detail in other chapters: chapter 33 describes use of PPPoE, and chapter 19 covers use of DHCP as well as static IP address assignment.

  To simplify configuring IP settings appropriate for your xDSL subscription, the Falcon Web interface has a *Basic Setup Page*, see section 11.2.1.

  For those who wish to configure Falcon via the CLI, section 11.1.4 below provides useful information.

  Default: **DHCP** (i.e., acquire your IP address from your ISP via DHCP)

- **VLAN settings:** By factory default, the xDSL port will belong to VLAN 1006 (untagged), while all Ethernet ports will be belong to VLAN 1 (untagged).

  If the Falcon is configured to act as *xDSL/Ethernet Bridge* via the *Basic Setup Page* (see section 11.2.1), all ports (xDSL and Ethernet) will be mapped to VLAN 1.

11.1.2 ADSL specific settings

There are two types of ADSL specific xDSL settings, and both of them concern the use of ATM as ADSL carrier.

- *VPI and VCI:* In WeOS you need to define the identifier of your ATM permanent virtual circuit (PVC) to your ADSL provider. This identifier contains two parts, the virtual path identifier (VPI) and the virtual circuit identifier (VCI). What values to use depends on your ISP provider.

  Default: **VPI 8** and **VCI 35**

- *ATM Encapsulation:* Falcon units support two ATM encapsulation modes: ”bridged LLC” and ”bridged VC-MUX” ([9]). Which setting to use depends on your ISP provider.

  Default: **bridged LLC**

There is also an additional ADSL related setting to specify if the ADSL is carried over a POTS telecom network (Annex A, I, L, or M), or an ISDN telecom network (Annex B or J). However, as of WeOS v4.17.1 the **”annex”** setting also applies to VDSL, see section 11.1.1.

Default: **Annex A (POTS)**

11.1.3 xDSL settings in common with Ethernet ports

The following parameters can be configured for xDSL ports in the same way as for Ethernet ports. In WeOS, VDSL uses Ethernet First Mile (EFM) encapsulation, and ADSL uses ”bridged” LLC or VC-MUX encapsulation (see section 11.1.2), thus many Ethernet settings apply to xDSL ports. More detailed information is found in chapter 8.

- *Port enable/disable:* Ports can be disabled and enabled administratively.
- *Port priority mode:* Define whether incoming packets should be prioritised based on VLAN tag, VLAN ID, port ID, IP ToS, etc. See also section 8.1.4.
- *Port priority (level):* The inbound priority associated with this port. See also section 8.1.4.
- *Link alarm:* Link status can be configured as an alarm source. See also section 8.1.5.

- *Inbound rate limit:* Setting the inbound rate limit is possible on DSL ports, but is likely of less interest than on Ethernet ports, since the DSL data rates are primarily limited by the rate of the DSL line. See also sections 8.1.6 and 11.1.1.
- *Outbound traffic shaping:* Setting the outbound rate limit (traffic shaping) is possible on DSL ports, but is likely of less interest than on Ethernet ports, since the DSL data rates are primarily limited by the rate of the DSL line. Furthermore, outbound traffic shaping in *frames per second* mode is not available on DSL ports. See also sections 8.1.7 and 11.1.1.
- *Fall-back default-VID*: The fall-back default VID setting is only of interest for the special case when *untagged* packets are received over a link only associated with *tagged* VLANs.

Ethernet settings for *port speed/duplex* mode, and *MDI/MDIX* mode do not apply to xDSL ports, thus are not configurable.

Note
As of WeOS v4.17.1, enabling/disabling flow control (as described in section section 8.1.3) has no effect on xDSL ports.

11.1.4 Connecting to your ISP over an xDSL line

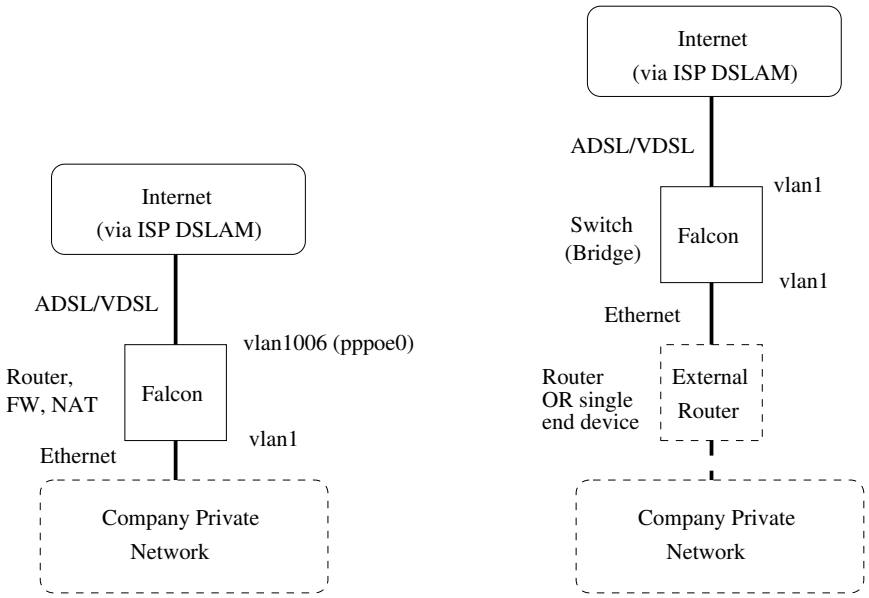
Recommendation: Use Basic Setup in Web
The simplest way to configure your Falcon unit to connect to your ISP is to use the *Basic Setup* web page, see section 11.2.1.
This section is intended (1) for those who wish to configure the Falcon via the CLI, and (2) for those looking for more background details on how to configure Falcon as an xDSL router or bridge.

This section describes the most common steps to configure your Falcon xDSL router to connect to your ISP. Although many configuration settings are affected, setting up your ISP should be straight-forward:

- The factory default configuration of xDSL are adapted to using the Falcon as an xDSL router.
- On the Falcon xDSL router, the web interface includes a *basic setup* page, for easy configuration of the most common use cases, see section 11.2.1.

A common setup is use Falcon as broadband router when connecting your company network towards the Internet, see fig. 11.2a.

An alternative is to use the Falcon as a xDSL/Ethernet bridge to connect a single end device (such as a PC), or to use a separate router to connect your local network, as shown in fig. 11.2b.



a) Using Falcon as broadband router

b) Using Falcon as switch (bridge)

Figure 11.2: Common ADSL/VDSL topologies: a) Using Falcon as broadband router, or b) using Falcon as an xDSL/Ethernet switch (bridge) with an external router (or single end-device such as a PC) behind.

Section 11.1.4.1 focus on using Falcon as a router, while section 11.1.4.1 covers on how to use Falcon as a switch (bridge). Both sections assume you have configured the xDSL port settings appropriately for your xDSL subscription (see also sections 11.1.1 and 11.1.2).

11.1.4.1 Using Falcon as a Router

By factory default, Falcon is configured as a router:

- **Port Segmentation:** The xDSL and Ethernet ports are mapped to two VLANs.

- *WAN port:* The xDSL port is mapped to VLAN 1006. VLAN 1006 had IGMP snooping disabled, thereby avoiding sending IGMP queries towards your ISP.

  Example

  ```
  vlan 1006
          untagged dsl 1
          no igmp
          ...
  ```

- *LAN ports:* The Ethernet ports are used as LAN ports, and are all mapped to the default VLAN, i.e., VLAN 1. VLAN 1 has the same factory default settings as other WeOS products.

  Example

  ```
  vlan 1
          untagged eth 1-4
          igmp
          ...
  ```

- **Network Interface Settings:**
- *WAN interface:* There are three methods to assign the IP address of the WAN interface, and which method to use depends on your xDSL Internet Service Provider (ISP): (1) acquire it via *DHCP*, (2) configure a *static* IP address, or (3) acquire the IP address via *PPPoE*. Each method is described below.

  By default, Falcon is configured to acquire the WAN interface address via DHCP.

  1. *Address via DHCP:* The WAN interface will by default use DHCP to get its IP address automatically from the ISP. In addition, interface *vlan1006* is assigned *admin distance* ”1” (section 19.2.6), in order to dynamically learn default gateway, DNS server and other global information via DHCP. Management services such as SSH, HTTP (Web), etc. are by default disabled to avoid unauthorised access from the public Internet.

     Example

     ```
     iface vlan1006 inet dhcp
             distance 1
             no management
             end
     ```

  2. *Static IP address:* The WAN interface can be configured to get its IP address assigned statically. If your ISP provides this option, the ISP

will inform you what address to use for your subscription. The example below uses address *192.168.5.4* and netmask *255.255.255.192* to illustrate the method.

Example

```
iface vlan1006 inet static
        distance 1
        no management
        address 192.168.5.4/26
        end
```

With static IP assignment you would also need to set the IP address of the default gateway and DNS server(s) (information provided by your ISP). In the example below the default gateway has address *192.168.5.1*, and a DNS server at *192.168.5.2*.

Example

```
ip
        route default 192.168.5.1
        name-server 192.168.5.2
        ...
        end
```

3. *Address via PPPoE:* Some ISPs use *PPPoE* for authorisation of, and IP address assignment to, their customers. To configure a WAN interface to use PPPoE, a PPPoE instance is created and mapped to the associated VLAN interface (here *vlan1006*). This will in turn create a PPPoE interface (here *pppoe0*), which now acts as our WAN interface. The example below shows the default setting for the PPPoE interface; the *admin distance* and *management* settings are automatically *copied* from the configuration of interface *vlan1006*.

Example

```
pppoe 0
        iface vlan1006
        ppp-advanced
                identity username@provider password sEcReT
                end
        end
...
iface pppoe0 inet dynamic
        mtu 1492
        tcp-mss 1412
        distance 1
        no management
        end
```

As interface *pppoe0* is typically used as upstreams interface, the NAT settings should be adapted, see *Routing, Firewall and NAT* below.

- *LAN interface:* The LAN interface *vlan1* is by default assigned IP address *192.168.2.200*. All management services are enabled on the LAN interface.

Example

```
iface vlan1 inet static
        distance 16
        management ssh http https ipconfig snmp
        address 192.168.2.200/24
        end
```

- **Routing, Firewall and NAT:** Falcon by default has IP forwarding (routing) and NAT enabled. Thereby Falcon can to route packets between a private network on its LAN interface (*vlan1*) and the public Internet on its WAN interface.

  The default firewall and NAT rules will block all incoming traffic on the WAN interface, except for packets belonging to established connections. (Such connections are in turn initiated from the private network, i.e., from the LAN side.) These settings are chosen to limit the risk for security attacks when connecting the Falcon to a public network such as the Internet.

  Special firewall deny rules are set up for TCP and UDP port 53 (DNS). These are to prevent the Falcon to become an open DNS relay on the WAN side.

  Open DNS relay is considered to be a security problem and can be used for remote attacks of the ISP's DNS server. DNS relay is enabled on all interfaces and should be filtered away on all interfaces facing public networks. Normal DNS traffic originating from the inside (from the LAN) will work as expected and is not affected by these rules.

Example

```
ip
        forwarding
        firewall
                policy input DROP
                policy forward DROP
                filter allow in vlan1 proto icmp
                filter deny in vlan1006 dport 53 proto udp
                filter deny in vlan1006 dport 53 proto tcp
                nat type napt out vlan1006 addfilter
                enable
        end
```

Adapting Firewall and NAT rules when using PPPoE
When PPPoE is used for WAN IP address assignment (see above), the firewall and NAT rules must be adapted accordingly, i.e., ”vlan1006” should be replaced by ”pppoe0” as shown in the example below.

Example

```
ip
        forwarding
        firewall
                policy input DROP
                policy forward DROP
                filter allow in vlan1 proto icmp
                filter deny in  pppoe0 dport 53 proto udp
                filter deny in  pppoe0 dport 53 proto tcp
                nat type napt out  pppoe0 addfilter
                enable
        end
```

- **Other Configurations:** The items above cover the most important configuration settings when connecting a Falcon to your ISP. Notes on a few more settings are given below:
  - *RSTP:* Westermo switches running WeOS typically have RSTP enabled on all Ethernet and DSL ports. However, the xDSL port on Falcon have RSTP disabled by default. For more information on RSTP, see chapter 16.
  - *VPN:* Its possible to use the Falcon as a VPN gateway. For more information on configuring VPNs in WeOS, see part IV.
  - *DHCP Server:* For information on how to make your Falcon act as DHCP server on your local network (*vlan1*), see chapter 22.

11.1.4.2 Using Falcon as a Switch (Bridge)

As shown in fig. 11.2b, it is possible to use the Falcon as a xDSL/Ethernet bridge. That is, the xDSL port does not have to be used as a dedicated *router* port; instead the Falcon could *switch* packets between Ethernet and xDSL ports, given that they are mapped to the same VLAN (see chapter 13).

Although it is possible to make the Falcon work as a *regular* WeOS switch, there are some differences:

- *Falcon is a router by default:* All WeOS devices can be configured to act as *router* or *switch*. The difference is that Falcon is configured as router in its

factory default setting (able to route between the WAN interface *vlan1006* and the LAN interface *vlan1*, while other WeOS devices act as switches by default (all ports on VLAN 1).

- *Layer-2 Redundancy (RSTP/FRNT):* As the xDSL port is used to connect to a xDSL provider (ISP), the remote end is managed by an external organisation. Thus, layer-2 redundancy protocols such as RSTP and FRNT should not be used on the xDSL port; for FRNT this is prohibited, and for RSTP it is disabled by default.

The simplest way to configure your Falcon to act as a switch is by using the *Basic Setup Page* in the Web interface (section 11.2.1). This way, all ports (Ethernet and xDSL) will be mapped to VLAN 1. The Falcon will then be accessible via the default IP address (IP address 192.168.2.200, netmask 255.255.255.0) unless you have changed the IP settings of interface *vlan1*. As an alternative to using the *Basic Setup Page*, you could achieve the corresponding result by removing VLAN 1006, either via the Web interface (section 13.3) or via the CLI (section 13.4) as shown below.

Example

```
falcon:/#> show vlan

VID  Name             Oper Untagged/Tagged
---- ---------------- ---- ------------------------------------------------------

   1 vlan1            DOWN U:eth 1-4
                           T:

1006 vlan1006         DOWN U:dsl 1
                           T:

------------------------------------------------------------------------------
falcon:/#>
```

Example

```
falcon:/#> configure
falcon:/config/#> no vlan 1006
falcon:/config/#> end
Port dsl 1  did not belong to any VLAN, setting as untagged in VLAN 1.
vlans: Problem activating settings.
There was some problem activating your configuration changes!

This could result in a non-functional system, continue anyway (y/N)? y
OK, accepting configuration anyway -- please review the running configuration.
Stopping DHCP Clients ..................................... [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
falcon:/#> show vlan
Press Ctrl-C or Q(uit) to quit viewer, Space for next page, <CR> for next line.

VID  Name             Oper Untagged/Tagged
---- ---------------- ---- ------------------------------------------------------
   1 vlan1            UP   U:ALL
                           T:

------------------------------------------------------------------------------
falcon:/#> cp running-config startup-config
falcon:/#>
```

There are additional setting you may consider changing when running the Falcon as a switch:

- *Limit remote management:* When the xDSL port is mapped to VLAN 1, the Falcon will be open for remote management via the xDSL port just as it is via the Ethernet ports on VLAN 1.

  This is usually no problem, as the Falcon by default is assigned the default IP address (192.168.2.200) on interface *vlan1*, and that address is not routable via the ISP. However, if limiting remote management is still a concern, you could, e.g., remove the IP address of interface *vlan1*. The Falcon can then be managed only via the console port (CLI) instead.

  Example

  ```
  falcon:/#> configure
  falcon:/config/#> iface vlan1
  falcon:/config/iface-vlan1/#> inet static
  falcon:/config/iface-vlan1/#> show address
  192.168.2.200/24
  falcon:/config/iface-vlan1/#> no address
  falcon:/config/iface-vlan1/#> leave
  Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
  falcon:/#>
  ```

- *As switch there is no firewall:* Acting as a switch, the Falcon no longer serves as a firewall towards the Internet. You should therefore ensure that you

protect your local network, typically by running firewall in an external router (or in a directly attached PC), see fig. 11.2b.

- *Disable IGMP Snooping:* VLAN 1 has IGMP snooping enabled by default. This should be fine even when the xDSL port is on VLAN 1, however, if you have concerns about running IGMP snooping on a port towards your ISP you can disable IGMP snooping on VLAN 1.

Example

```
falcon:/#> configure
falcon:/config/#> vlan 1
falcon:/config/vlan-1/#> no igmp
falcon:/config/vlan-1/#> leave
Stopping IGMP Snooping daemon ............................. [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
falcon:/#>
```

- *Disable IP Forwarding:* As long as all ports are mapped to VLAN 1, the Falcon will act as a switch, even though the *IP forwarding* configuration option is enabled. However, if you have concerns about having IP forwarding enabled, you can disable it.

  If you use the *Basic Setup Page* in the Web interface (section 11.2.1) to configure the Falcon as switch (bridge), IP forwarding will be disabled automatically.

Example

```
falcon:/#> configure
falcon:/config/#> ip
falcon:/config/ip/#> no forwarding
falcon:/config/ip/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
falcon:/#>
```

11.2 Managing ADSL/VDSL ports via the web interface

The Web interface provides configuration of xDSL ports (sections 11.2.1-11.2.3) as well as listing of xDSL port statistics.

The xDSL statistics is provided in two views – an *overview* with a selection of statistics for all xDSL ports, including some status information (section 11.2.4), and a *detailed* page with a larger set of statistics (section 11.2.5).

11.2.1 Basic Setup for Falcon DSL router

Menu path: Basic Setup

This feature requires a JavaScript enabled web browser. To simplify the setup of the Falcon unit for remote access, a basic setup page is provided with the most basic settings compiled into one view. In many cases this page may be sufficient for setting up the Falcon for remote access.

Note
When you enter the basic setup page and make changes to the configuration and press the apply button, some settings will be reset. See section 11.2.1.1 below for more information.

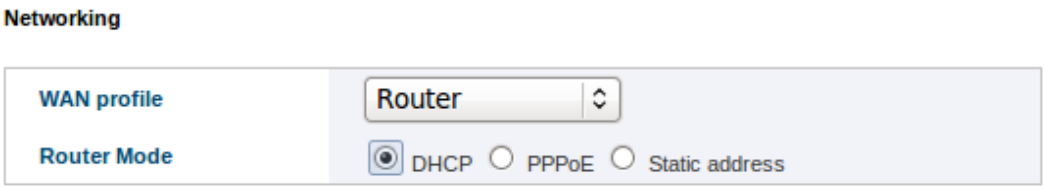


Figure 11.3: Basic Setup Profile and Mode

To set up the switch using the Basic Setup, two fundamental settings have to be set first. These two settings control the other options displayed on the page.

WAN Profile	**Router**	The unit will be set up as a router with a firewall protecting the LAN side from the WAN side.
	Bridged	The unit will act as a plain switch.

Continued on next page

		Continued from previous page
Router Mode	**DHCP**	The WAN side will expect a DHCP-server to provide the switch with an IP address.
	PPPoE	The WAN side will set up a PPPoE connection with the ISP to provide the internet connection.
	Static	The IP address, netmask and gateway will be manually entered.

The *DHCP* router mode (shown above) does not need any additional settings. The *Static IP* and *PPPoE* router modes require additional settings as described below.

Irrespective of the selected *router mode*, you may also need to fill out ADSL/VDSL port settings, as shown at the end of this section.

Static IP Settings

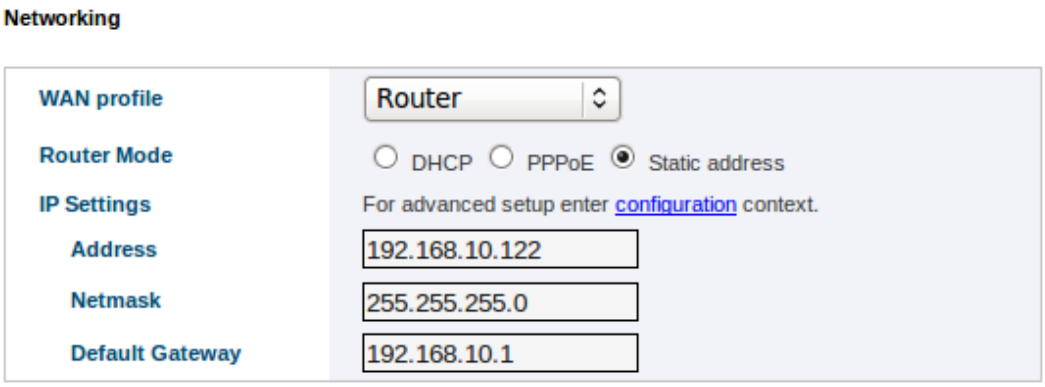


Figure 11.4: Basic Setup Static IP

If the static IP mode is selected you are asked to fill in the following entries.

Address	The IPv4 address to assign to the interface.
Netmask	The netmask for the IPv4 address. Identifies which IP addresses are located on the same subnet.
	Continued on next page

	Continued from previous page
Default Gateway	Statically configured default gateway of the unit. This is the IP address of the gateway to send packages to when no more specific route can be found in the routing table. This value overrides any value retrieved dynamically (e.g. using DHCP). Leave empty to enable dynamically retrieved gateway address or if no default gateway should be available.

PPPoE Settings

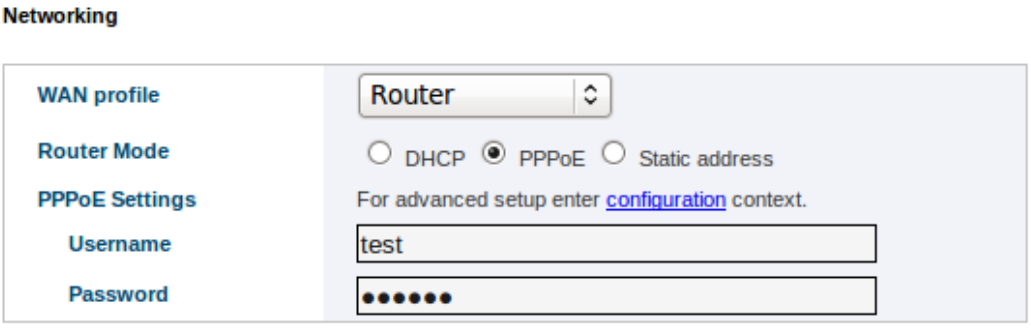


Figure 11.5: Basic Setup PPPoE

If the PPPoE mode is selected you are asked to fill in the following entries.

Username	The username provided by the PPPoE provider.
Password	The password provided by the PPPoE provider.

DSL Settings

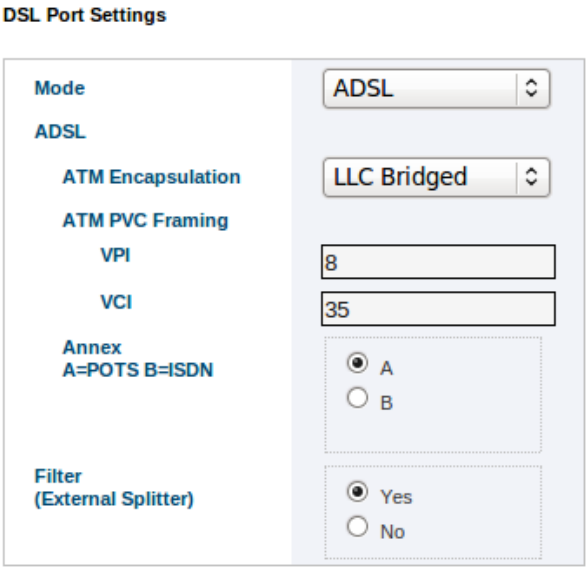


Figure 11.6: Basic Setup DSL settings

In addition you may have to change the DSL settings if they do not satisfy the requirements from your ISP, see fig. 11.6.

Mode	Specify whether the xDSL port should operate ADSL port or VDSL port. Default: ADSL
ATM Encapsulation	ATM encapsulation. Default: LLC
ATM PVC Framing	Set the appropriate VPI and VCI for the ATM PVC. Default: VPI 8, VCI 35
Annex	Annex A or B can be set for either ADSL or VDSL mode. Annex L, M, L-M, I and J can only be set for ADSL. The annex I and J options are extensions of ADSL annex A and B. The annex L and M options are extensions of ADSL annex A. The annex A-B option is only available for VDSL mode. Default: Annex A (POTS)
Filter	External splitter or not. POTS/ISDN filter. Default: Enabled

11.2.1.1 Basic Setup Behavior

As noted above, some settings will be reset when applying the basic setup page. This is what will happen:

- **When applying bridged profile:**
  All but one VLAN is removed and its interface settings are reset.
  Details:
  - All VLANS are removed and VLAN 1 re-created. As a result of this, all advanced settings on VLAN 1 and it's associated interface will be lost.
  - All ports are associated untagged to VLAN 1.
  - The firewall is removed.
  - IP-forwarding (routing) is turned off.
- **When applying a router profile:**
  All settings for LAN and WAN and its associated interfaces are reset. Firewall rules are reset. All existing PPPoE configurations are removed.
  Details:
  - All VLANS are removed and VLAN 1 (LAN) and VLAN 1006 (WAN) re-created. As a result of this, all advanced settings on VLAN 1 and VLAN 1006 and their associated interfaces will be lost.
  - The DSL port is associated untagged to VLAN 1006 (WAN).
  - All remaining ports are associated untagged to VLAN 1 (LAN).
  - The firewall is removed and then re-created. This will result in loss of all current NAT, Port forwarding and Access rules.
  - IP-forwarding (routing) is turned on.

  In addition for the DHCP and Static modes:
  - A NAT-rule for external interface VLAN 1006 (WAN) and internal VLAN 1 (LAN) is added.
  - Firewall filtering rules denying inbound UDP and TCP port 53 (DNS) are added for the external interface VLAN 1006 (WAN).

  In addition for the PPPoE mode:
  - A PPPoE configuration is added.

- A NAT-rule for the PPPoE interface (WAN) and internal VLAN 1 (LAN) is added.
- Firewall filtering rules denying inbound UDP and TCP port 53 (DNS) are added for the PPPoE interface (WAN).

Note

Firewall filtering of inbound UDP and TCP port 53 is added to prevent the unit to become an open DNS relay on the WAN side.
Open DNS relay is considered to be a security problem and can be used for remote attacks of the ISP's DNS server. DNS relay is enabled on all interfaces and should be filtered away on all interfaces facing public networks. Normal DNS traffic originating from the inside (from the LAN) will work as expected and is not affected by these rules.

11.2.2 List and Edit ADSL/VDSL Port Settings

Menu path: Configuration ⇒ Port ⇒ DSL

When entering the DSL configuration page you will be presented to a list of all DSL ports available on your switch, see fig. 11.7.

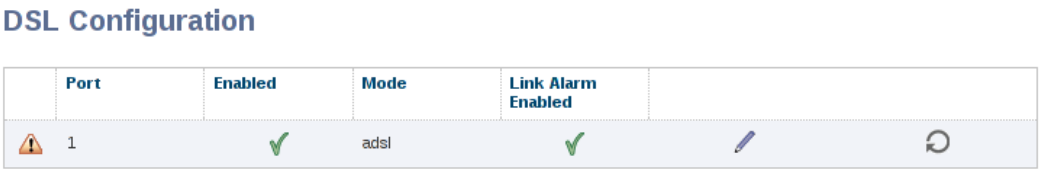


Figure 11.7: DSL Port configuration settings overview

Alarm	There is an active link alarm associated with the port. Only shown if link alarm is enabled and the link is down.
Port	The port label.
Enabled	A green check-mark means the xDSL port is enabled, and a dash means it is disabled.
Type	ADSL or VDSL
Link Alarm Enabled	When link alarm is enabled an alarm will be generated if port link is down. Alarms trigger an SNMP trap message to be sent and alarms to be shown on the administration web. In the ports overview table a green check-mark means enabled, and a dash means disabled.
Edit	Click this icon to edit a port's settings.
Restart	Click this icon to retrain the DSL ports.

To change the settings for a specific xDSL port you will have to click the edit icon which will take you to the DSL port setting edit page see section 11.2.3.

11.2.3 Edit xDSL Port Settings

Menu path: Configuration ⇒ Port ⇒ DSL ⇒ 

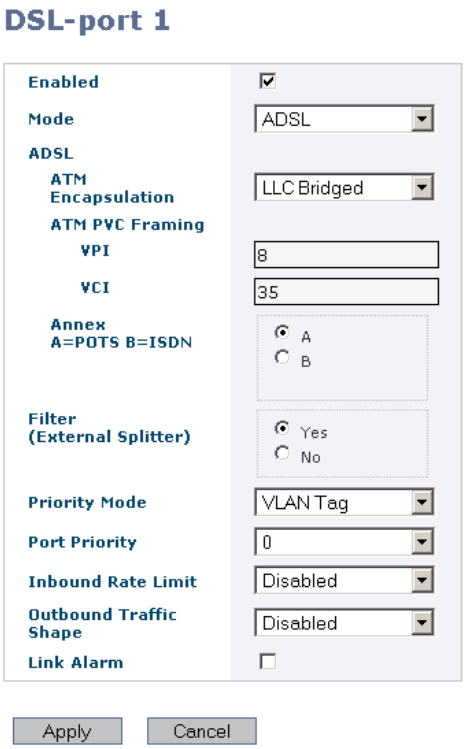


Figure 11.8: DSL port configuration settings edit page

On this page you can change the settings for the xDSL port.

Enabled	Enable or Disable the port
Mode	Specify whether the xDSL port should operate ADSL port or VDSL port. Default: ADSL
ATM Encapsulation	ATM encapsulation. Default: LLC
ATM PVC Framing	Set the appropriate VPI and VCI for the ATM PVC. Default: VPI 8, VCI 35

Continued on next page

Continued from previous page

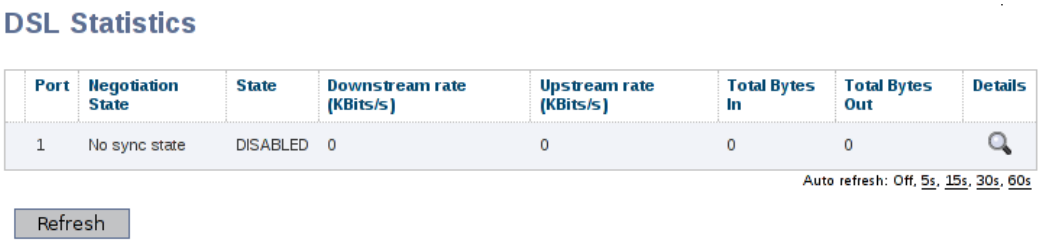
Annex	Annex A or B can be set for either ADSL or VDSL mode. Annex L, M, L-M, I and J can only be set for ADSL. The annex I and J options are extensions of ADSL annex A and B. The annex L and M options are extensions of ADSL annex A. The annex A-B option is only available for VDSL mode. Default: Annex A (POTS)
Filter	External splitter or not. POTS/ISDN filter. Default: Enabled
Priority Mode	Here you select on what information priority will be based: **Port Based** Based on the port's priority. See the next item (**Priority**). **IP** Based on the content of the IP ToS bits (IPv4) or the IP TC bits (IPv6). **VLAN Tag** Based on the content of the (802.1p) priority field inside the received packet's VLAN tag.
Port Priority	The port's priority level.
Inbound Rate Limit	Bandwidth limit for inbound traffic
Outbound Traffic Shape	Bandwidth limit for outbound traffic
Link Alarm	When link alarm is enabled an alarm will be generated if port link is down. Alarms trigger an SNMP trap message to be sent and alarms to be shown on the administration web.

11.2.4 ADSL/VDSL statistics Overview

Menu path: Status ⇒ Port ⇒ DSL

On the DSL port statistics overview page you will be presented to a selection of static data for each port. Additional statistic numbers are presented on the detailed view page.

Note: If only one DSL port is present in the unit, you will be redirected to the detailed statistics and status page.

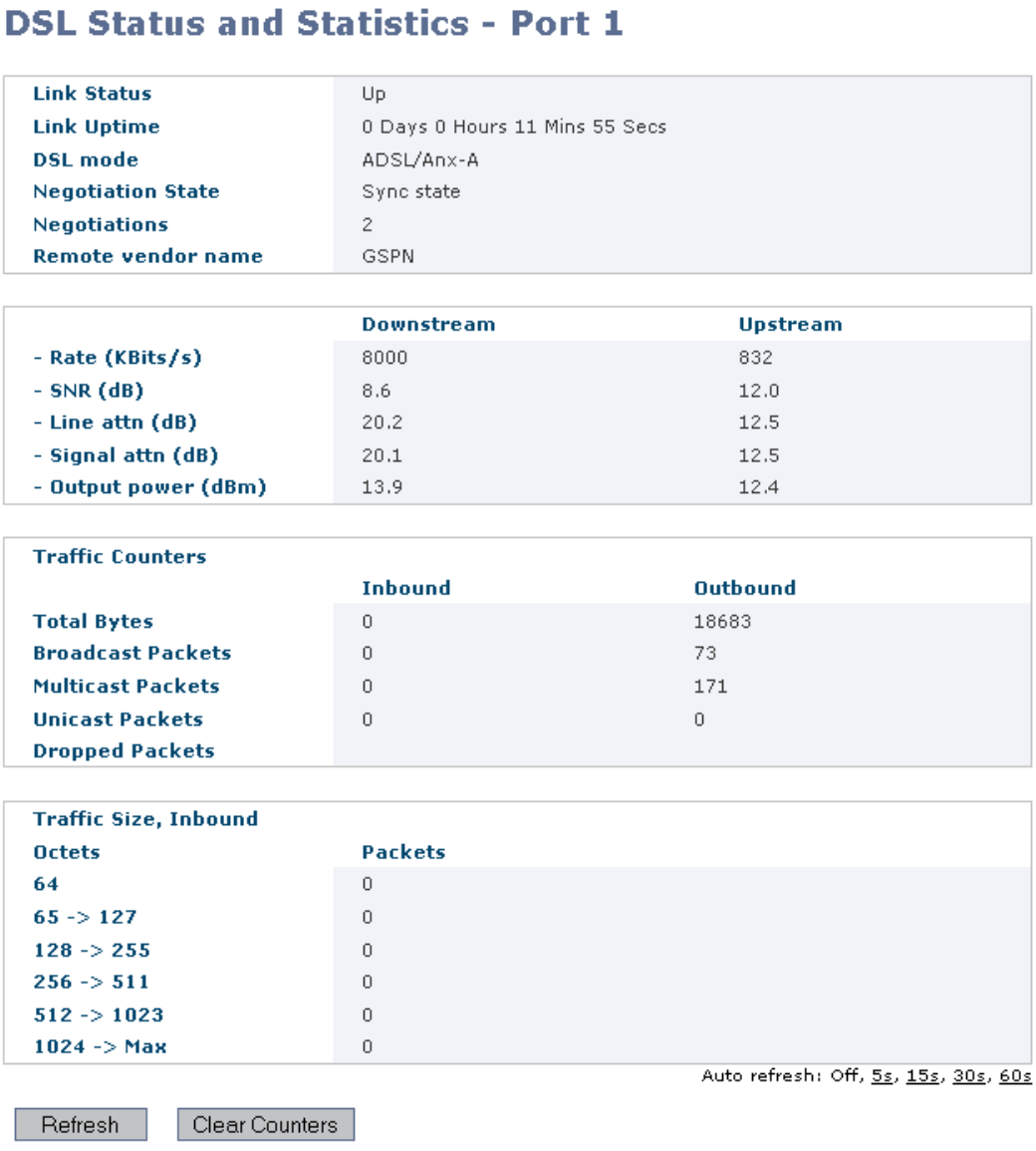


Alarm	An alarm icon appears at the start of a line if there is a link alarm on a port.
Port	The port label.
Negotiation State	Current state of the DSL-line negotiation.
State	Link state
Downstream Rate	Negotiated DSL downstream rate in bit/s.
Upstream Rate	Negotiated DSL upstream rate in bit/s.
Total Bytes In	Total number of bytes received on the port.
Total Bytes Out	Total number of bytes sent out on the port.
Details	Click this icon to view more detailed statistics for the port.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.

11.2.5 Detailed ADSL/VDSL Port Statistics

Menu path: Status ⇒ Port ⇒ DSL ⇒ 

If only one DSL port is present in the unit, or when clicking the *details*-icon in the overview page you will be presented to the detailed statistics page for the DSL port.



Link Status	Status of link, (Up/Down). If a link-alarm is associated with this port, an alarm icon is displayed if the link-alarm is active.
Link Uptime	The time since link was established.
DSL mode	ADSL or VDSL
Negotiation State	Current state of the DSL-line negotiation.
Negotiations	Number of negotiations since unit startup.
Remote Vendor Name	Identifier string of DSLAM vendor.
Rate	Negotiated DSL downstream and upstream rate in bit/s.
SNR (dB)	Upstream and Downstream Signal to Noise Ratio (SNR) in dB on this link.
Line attn (dB)	Line attenuation is the loss of signal over distance, in dB, downstream and upstream.
Signal attn (dB)	Signal attenuation in dB, downstream and upstream.
Output power (dBm)	Output power in dBm, downstream and upstream.
Traffic Counters	See section 9.2.2 for details.
Traffic Size, Inbound	See section 9.2.2 for details.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
<<Previous	Go to statistics for previous port. Only shown if more than one DSL port available.
Next>>	Go to statistics for next port. Only shown if more than one DSL port available.
Refresh	Click on this button to reload with updated statistics.
Clear Port	Clear all statistics counters for the port shown.

11.3 Managing ADSL/VDSL ports via the CLI

The table below shows xDSL port management features available via the CLI.

Command	Default	Section
Configure ADSL and VDSL port settings		
port [dsl|xdsl|. . .] <PORTLIST>		Section 11.3.1
[no] mode <adsl [annex <a|b|i|j|l|m|l-m>] | vdsl [annex <a|b|a-b>]>	adsl annex a	Section 11.3.2
[no] filter	Enabled	Section 11.3.3
ADSL specific port settings		
mode adsl		
[no] encap <llc|vcmux>	llc	Section 11.3.4
[no] pvc <VPI/VCI>	8/35	Section 11.3.5
Port settings in common with Ethernet ports (chapter 8)		
[no] enable	Enabled	Section 8.3.3
[no] priority <0-7>	0	Section 8.3.6
[no] priority-mode <tag|ip|port>	tag	Section 8.3.7
[no] link-alarm	Disabled	Section 8.3.8
[no] rate-limit <70-2560>	Disabled	Section 8.3.9
[no] traffic-shaping <70-2560>	Disabled	Section 8.3.10
[no] default-vid <VLAN_ID>	Disabled	Section 8.3.14
Show ADSL/VDSL related status and statistics		
show dsl		Section 11.3.6
show ports		Section 8.3.15
show rmon		Section 9.3

11.3.1 Managing xDSL port settings

Syntax `port [dsl|xdsl|...] <PORTLIST>`

Context Global Configuration context

Usage Enter the xDSL Port Configuration context.

A **”PORTLIST”** is a comma separated list of ranges of xDSL ports without intermediate spaces, e.g., **”1/1,1/2”** on a *slotted* product, or **”1-3,5”** on a *non-slotted* product.

The port qualifier keyword **”xdsl”** (or **”dsl”**) is not needed if the numbers in the **”PORTLIST”** are unique to DSL ports.

For a more general description of the **”port”** command, see section 8.3.1.

Use **”show port dsl <PORT|PORTLIST>”** or **”show port xdsl <PORT|PORTLIST>”** to list port configuration information for the given xDSL port(s). Also available as **”show”** command within the xDSL Port Configuration context.

Default values Not applicable.

Entering the xDSL configuration context on a Falcon:

Example

```
falcon:/#> configure
falcon:/config/#> port dsl 1
falcon:/config/port-dsl1/#>
```

Listing configuration information on the xDSL port on a Falcon:

Example

```
falcon:/config/#> show port dsl 1
xDSL ------------------------------------------ Priority ---- Limit - Default
Port    Ena Mode Filter Encap PVC   Annex Alarm  Mode Level    In | Out     Vid
================================================================================
DSL 1   YES adsl    YES   llc 8/35      A    NO  tag      0   None None    Auto
================================================================================
falcon:/config/#>
```

11.3.2 Setting xDSL port mode (ADSL or VDSL) and carrier type

Syntax `[no] mode <adsl [annex <a|b|i|j|l|m|l-m> | vdsl [annex <a|b|a-b>>`

Context xDSL Port Configuration context

Usage Specify whether the xDSL port should operate as ADSL port or VDSL port, and

- *ADSL:*

  - Use **”mode adsl annex <a|i|l|m|l-m>”** to specify ADSL mode over a POTS carrier network.
  - Use **”mode adsl annex <b|j>”** to specify ADSL mode over an ISDN carrier network.
  - For further information on ADSL Annex settings, see section 11.1.1.

  When selecting ADSL mode, the ADSL specific settings **”encap”** (section 11.3.4) and **”pvc”** (section 11.3.5) are enabled.

- *VDSL:*
  - Use **”mode vdsl annex a”** to specify VDSL mode over a POTS carrier network.
  - Use **”mode vdsl annex b”** to specify VDSL mode over an ISDN carrier network.
  - Use **”mode vdsl annex a-b”** to auto-detect whether your VDSL connection is over a POTS or ISDN carrier. This alternative is usually preferable for VDSL connections due to its simplicity, but the auto-detection mechanism may experience problems on long copper cables. If this is the case, please try **”mode vdsl annex a”** or **”mode vdsl annex b”** depending on the carrier type of your VDSL connection.

Use **”no mode”** to reset the mode setting to the default value.

Use **”show mode”** to show whether the xDSL port is set to operate as ADSL or VDSL port, and the type of carrier network used, Annex A, Annex I, Annex L or Annex M (POTS) or Annex B or Annex J (ISDN). Annex I and J not supported in VDSL mode.

Default values ADSL over POTS (**”mode adsl annex a”**)

11.3.3 Specify whether external splitter is used or not

Syntax `[no] filter`

Context xDSL Port Configuration context

Usage Specify whether a (external) splitter is used or not, i.e., is the Falcon unit connected directly to the telephone jack or via a splitter.

Use command **”filter”** if a splitter is used, and **”no filter”** if no splitter is used.

Use **”show filter”** to show the xDSL port’s filter setting.

Default values **”filter”** (i.e., an external splitter is assumed by default)

11.3.4 Configure ADSL/ATM encapsulation type

Syntax `[no] encap <llc|vcmux>`

Context xDSL Port Configuration context (only available when ADSL mode is used, see section 11.3.2)

Usage Specify whether *bridged LLC* or *bridged VC-MUX* ATM encapsulation is used. What encapsulation option to use depends on your ADSL provider.

Use command **”llc”** to use *bridged LLC* and **”vcmux”** to use *bridged VC-MUX* encapsulation.

Use **”no encap”** to reset the encapsulation mode to the default setting.

Use **”show encap”** to show the xDSL port’s ADSL/ATM encapsulation setting.

Default values **”llc”**

11.3.5 Configure ADSL/ATM VPI and VCI

Syntax `[no] pvc <VPI/VCI>`

Context xDSL Port Configuration context (only available when ADSL mode is used, see section 11.3.2)

Usage Specify the VCI and VPI used for the ATM PVC by your ADSL provider.

Some examples: **”pvc 0/38”** is common in U.K., **”1/32”** is common in Germany, while **”pvc 8/35”** is common for many other ADSL providers inside and outside Europe.

Use **”no pvc”** to reset the PVC to use default VPI/VCI. (In future versions of WeOS the use of **”no pvc”**, as well as the default PVC setting, may change.)

Use **”show pvc”** to show the ATM PVC setting, i.e., which VPI and VCI are configured.

Default values ”pvc 8/35”

11.3.6 Show xDSL port status

Syntax `show dsl`

Context Admin Exec context.

Usage Show the status of all xDSL ports.

Default values Not applicable.

Example

```
falcon:/#> show dsl

Port, DSL mode          : DSL 1, ADSL/Anx-A
Channel, role           : channel 0, role CPE
Link state, uptime      : UP, 0 Days 0 Hours 7 Mins 15 Secs
Negotiation state       : Sync state, 4 changes since boot
Remote vendor name      : GSPN
Downstream ------------------------
           Rate         : 8000 kbps
           SNR          : 12.5 dB
           Line attn    : 8.3 dB
           Signal attn  : 8.2 dB
           Output power: N/A
Upstream --------------------------
           Rate         : 832 kbps
           SNR          : 12.0 dB
           Line attn    : 7.0 dB
           Signal attn  : 7.0 dB
           Output power: 12.4 dB
falcon:/#>
```

Chapter 12

Power Over Ethernet (PoE)

Some WeOS Viper products[52] have Ethernet ports with support for Power Over Ethernet (PoE[16] and PoE+[17]).

This chapter gives an overview of PoE support in WeOS products (section 12.1). Sections 12.2 and 12.3 concern PoE management support via the Web interface and CLI. PoE related SNMP support is covered in chapter 6, while management of PoE alarms/events is documented in chapter 24.

As of WeOS v4.17.1, PoE management via LLDP[17] is not supported.

12.1 Overview of Power over Ethernet (PoE)

Feature	Web	CLI	General Description
Per-Port PoE Configuration			
Enable/Disable	X	X	
Allocation Priority	X	X	Section 12.1.2
Power Limit	X	X	-”-
PoE Status			
Consumed power	X	X	
Allocated Power	X	X	Sections 12.1.1-12.1.2
Detected PoE Units	X	X	Section 12.1.1

12.1.1 PoE Power Classes

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

Table 12.1 lists the maximum power consumption for units 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) ($P_{class,unit}$)	Allocated Power (W) ($P_{class,alloc}$)
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

Table 12.1: Power allocated to and consumed by units of different PoE classes.

It is also possible to configure a maximum power limit on each individual PoE port, see section 12.1.2. The power allocated on the port then becomes the minimum of the (a) configured power limit, and (b) the power allocated for attached unit's class (as listed in table 12.1).

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

- *Good:* Ok. A PoE unit 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* curcuit.)
- *Low:* The connected unit is detected as a PoE unit and served, although its the resistance is too low to meet the PoE specification (and too high to be determined as short circuit (non-PoE unit connected)).
- *High:* The connected unit is detected as a PoE unit and served, although its

the resistance is too high to meet the PoE specification (and too low to be determined as unconnected (open curcuit)).

12.1.2 Allocation of PoE Power

There is maximum value for the amount of power a PoE switch can deliver ($P_{switch,max}$), see the User Guide of your PoE product[52] for more information on max output power. When more power is requested than available, the switch will stop/refuse[1] delivering power on the port(s) with lowest *priority*.

12.1.2.1 Calculating available power, and per-port power limitation

As of WeOS v4.17.1, PoE power is always allocated to handle *max* consumption by all admitted PoE units. For each port, the *max* consumption ($P_{port,max}$) is calculated as the minimum value of:

- ($P_{class,alloc}$): The power allocated to units of the attached class (see right column of table 12.1).
- ($P_{port-limit}$): The power limit configured for the port (if any).

The available power is calculated as max output power of the switch[1], minus the sum the max power for all (admitted) ports.

$$P_{available} = P_{switch,max} - \sum_{admitted} P_{port,max}$$

If a new PoE unit is attached, its $P_{class,alloc}$ will be compared to $P_{available}$:

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

[1] To compensate for limited accuracy in measured power consumption, your WeOS PoE unit may allow the measured and allocated power to raise somewhat above the stated $P_{switch,max}$ of the product, before power delivery is stopped/refused on some port. The customer should still ensure that PoE equipment attached to the WeOS PoE switch do not use more than $P_{switch,max}$ in total.

Preference Order	Port Name	Configured Priority	Tie-break Priority
0 (lowest)	X7	low	0
1	X9	low	2
2	X10	low	3
3	X5	high	6
4	X6	high	7
5	X3	critical	1
6	X2	critical	4
7 (highest)	X1	critical	5

Table 12.2: Example of allocation preference order for a given PoE priority configuration on a Viper-212-PoE unit.

12.1.2.2 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 PoE units, preference will be given to ports with higher priority.

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. For Viper-212-PoE[52] the ”tie-break” priority is as follows (starting with the lowest tie-break priority): X7(0), X3(1), X9(2), X10(3), X2(4), X1(5), X5(6), X6(7). The tie-break priority order may become configurable in future WeOS releases.

Table 12.2 illustrates the power allocation preference order in a specific configuration example on a Viper-212-PoE, where ports X1-X3 have been configured with priority *critical*, X5-X6 with priority *high*, and X7-X10 have priority *low*.

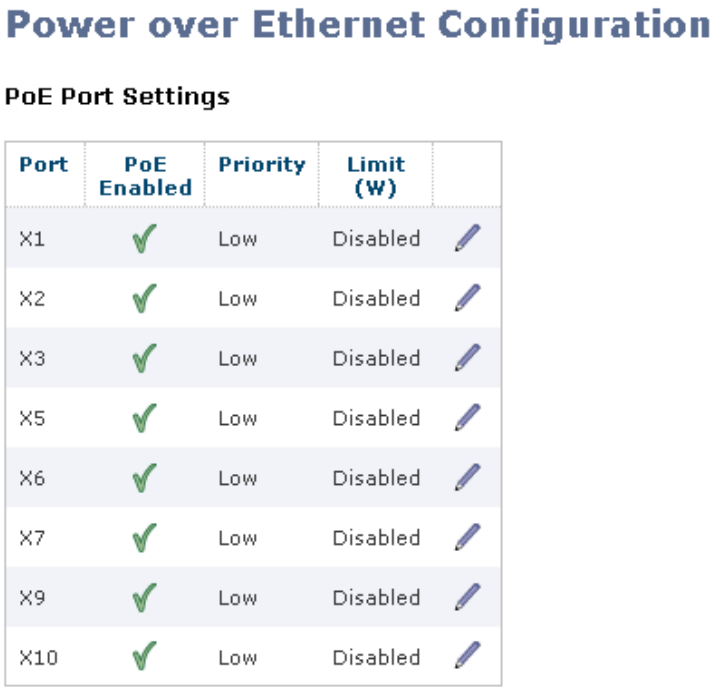
12.2 Managing PoE via the web interface

The Web interface provides configuration of PoE ports as well as listing of global and port specific PoE status.

12.2.1 List PoE Settings

Menu path: Configuration ⇒ PoE

When entering the PoE configuration page you will be presented to a list of all PoE ports available on your switch, and their settings.



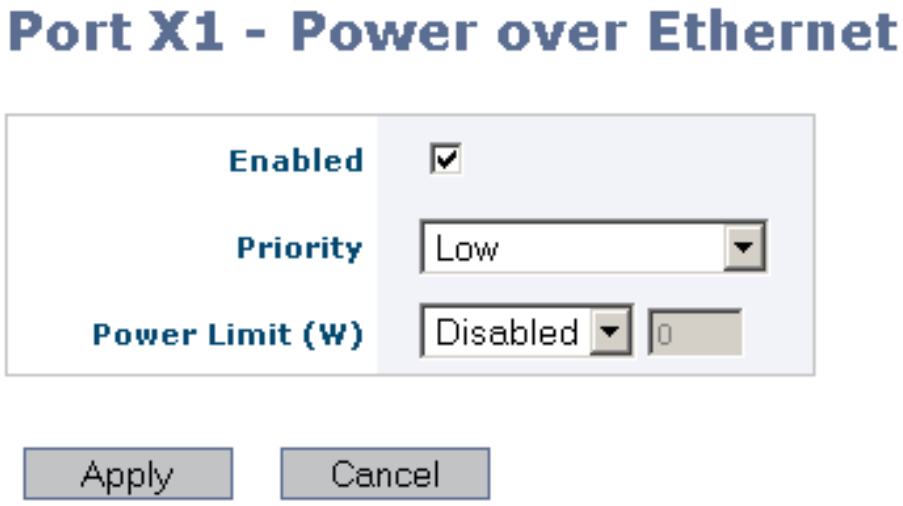
Port	The port label. (Only PoE capable Ethernet ports are listed.)
PoE Enabled	Shows if PoE is enabled or disabled on the port.
Priority	Shows the configured PoE priority (Low, High or Critical) for the port.

Continued on next page

	Continued from previous page
Limit	Shows the configured Power Limit for the port (in Watts), or **`Disabled`** if no port specific limit has been set.
Edit	Click this icon to edit a port's PoE settings.

12.2.2 Edit PoE Port Settings

Menu path: Configuration ⇒ PoE ⇒



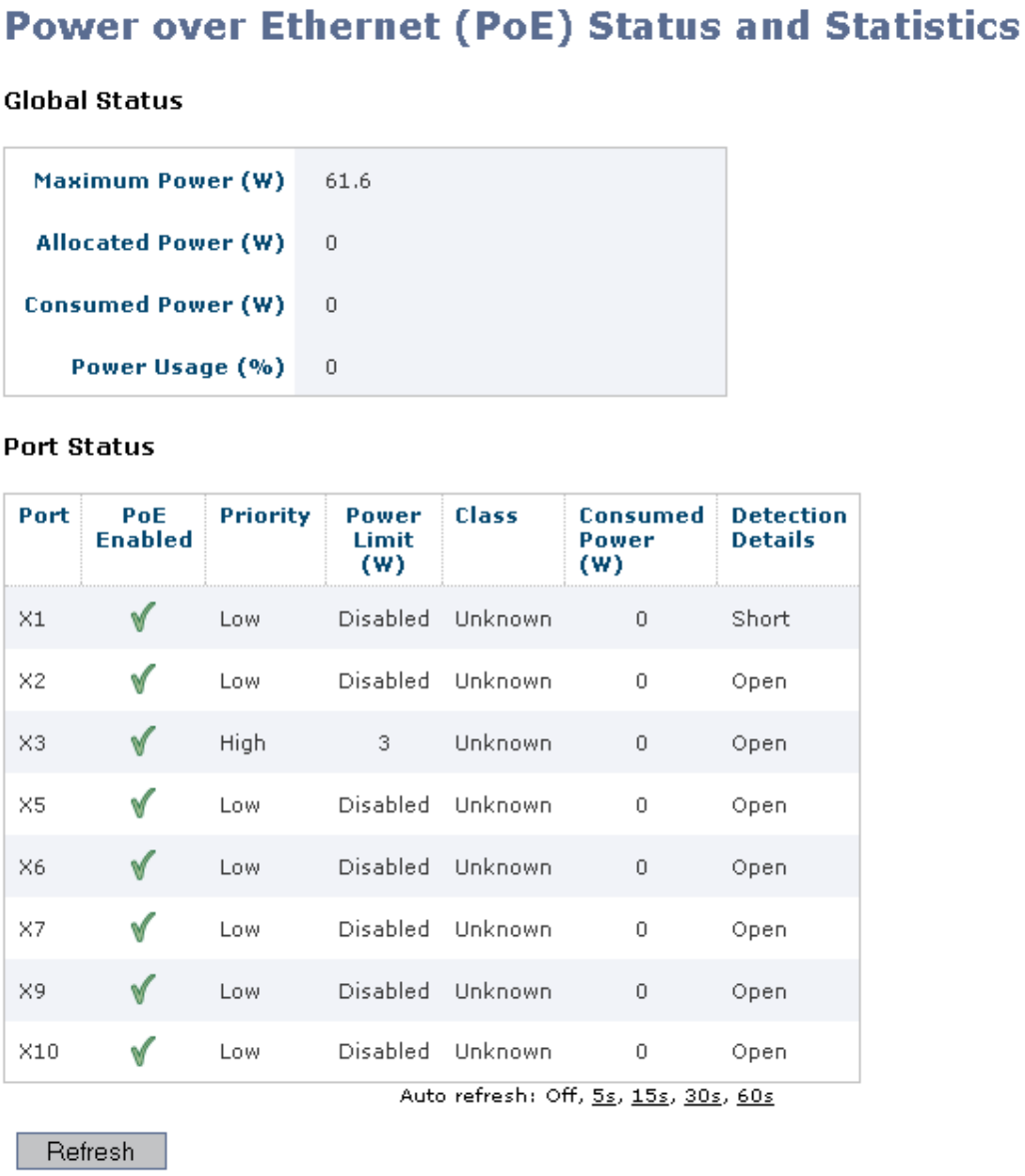
On this page you can change the PoE settings for the port.

Enabled	Enable/disable PoE on the port
Priority	PoE power allocation priority. When more power is requested than available, power will be dropped on the ports with lowest priority. Possible values: • **Low** (Shut down first) • **High** • **Critical** (Shut down last) See section 12.1.2 for more information.
Power Limit	Set port specific power limit. Allowed values are 1-30 (Watts), or **`Disabled`** (i.e., no port specific power limit).

12.2.3 PoE Status

Menu path: Status ⇒ PoE

On the PoE port status page you will be presented to global and port specific PoE status data.



Power over Ethernet (PoE) Status and Statistics

Global Status

Maximum Power (W)	61.6
Allocated Power (W)	0
Consumed Power (W)	0
Power Usage (%)	0

Port Status

Port	PoE Enabled	Priority	Power Limit (W)	Class	Consumed Power (W)	Detection Details
X1	✓	Low	Disabled	Unknown	0	Short
X2	✓	Low	Disabled	Unknown	0	Open
X3	✓	High	3	Unknown	0	Open
X5	✓	Low	Disabled	Unknown	0	Open
X6	✓	Low	Disabled	Unknown	0	Open
X7	✓	Low	Disabled	Unknown	0	Open
X9	✓	Low	Disabled	Unknown	0	Open
X10	✓	Low	Disabled	Unknown	0	Open

Auto refresh: Off, 5s, 15s, 30s, 60s

Refresh

Global Status	
Maximum Power	The maximum power (in Watts) the switch is able to deliver.
Allocated Power	Allocated power (in Watts). See section 12.1.2 for information on allocation and classes.
Consumed Power	The total power consumed on all PoE ports.
Power Usage	Percentage of available power currently consumed (i.e., ”Consumed Power”/”Maximum Power”).

Port Status	
Port	The PoE port label.
PoE Enabled	Shows if PoE is enabled or disabled on the port.
Priority	Shows the configured PoE priority (Low, High or Critical) for the port.
Power Limit	Shows the configured power Limit for the port (in Watts), or **`Disabled`** if no port specific limit has been set.
Class	Shows the PoE class (0-4) of the connected PoE unit, or **`Unknown`** if the class cannot be determined.
Consumed Power	Currently consumed power (in Watts) by the connected PoE unit.
Detection Details	Additional details on the unit connected to the PoE port (see also the **Class** column): • **Unknown:** Unit Resistance Unknown (e.g. PoE disabled on port) • **Short:** Non-PoE unit connected • **Low:** PoE Unit connected. Resistance OK, but low • **Good:** PoE unit connected. Resistance Good. • **High:** PoE Unit connected. Resistance OK, but high • **Open:** Nothing Connected
Auto Refresh	Click on a value to make the page reload with updated status automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
Refresh	Click on this button to reload with updated status.

12.3 Managing PoE via the CLI interface

Command	Default	Section
Configure PoE settings		
poe		Section 12.3.1
[no] port <PORTLIST|all>		Section 12.3.2
[no] enable	Enabled	Section 12.3.3
[no] priority <low|high|critical>	Low	Section 12.3.4
[no] limit <1-30>	Disabled	Section 12.3.5
Show PoE settings		
show poe [port <PORTLIST|all>]		Section 12.3.6
poe		
port <PORTLIST>|all>		
show enable		Section 12.3.7
show priority		Section 12.3.8
show limit		Section 12.3.9
Show PoE status		
show poe [full] [port <PORTLIST>]		Section 12.3.10

12.3.1 Manage PoE Settings

Syntax poe

Context *Global Configuration* context.

Usage Enter *PoE* configuration context.

Default values Not applicable.

Error messages None defined yet.

12.3.2 Manage per-port PoE settings

Syntax `[no] port <PORTLIST|all>`

Context *PoE* configuration context.

Usage Enter PoE port configuration context.

Use **”port <PORTLIST>”** to configure per-port settings for the PoE ports in the given list, e.g., **”port X2”**, or **”port X1-X5,X10”**. Use **”port all”** to configure per-port settings for all PoE ports.

Use **”no port <PORTLIST>”** to reset PoE port settings to their default values for the given port range.

Default values Not applicable

Error messages None defined yet.

12.3.3 Enable/Disable PoE on a PoE port

Syntax `[no] enable`

Context *PoE Port* configuration context.

Usage Enable/disable PoE on this port.

Default values Enabled (**”enable”**)

Error messages None defined yet.

12.3.4 Set PoE allocation priority

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

Context *PoE Port* configuration context.

Usage Configure PoE allocation priority setting (**”priority low”** is the lowest priority, while **”priority critical”** is the highest .

”no priority” will reset priority to default (**”priority low”**).

See section 12.1.2 for information on how to select between ports of the same configured priority.

Default values Low (**”priority low”**)

Error messages None defined yet.

12.3.5 Set PoE Power Limit

Syntax `[no] limit <1-30>`

Context *PoE Port* configuration context.

Usage Configure specific PoE Power limit (in Watts) on this port, e.g., **"limit 20"** to limit the delivered power to 20 Watts. The max power delivered is the minimum value of the limit configured for this port, and the power allocated for class of PoE unit attached (see $P_{class,alloc}$ in table 12.1).

Use **"no limit"** to remove port specific power limits (max power based solely on $P_{class,alloc}$).

Default values Disabled (**"no limit"**)

Error messages None defined yet.

12.3.6 Show PoE Settings

Syntax `show poe [port <PORTLIST|all>]`

Context *Global Configuration context.* configuration context. Also available as

- **"show [port <PORTLIST|all>]"** command in *PoE configuration* context, and as
- **"show"** command in *PoE Port configuration* context

Usage Show global PoE settings and per-port PoE settings. If the **"port <PORTLIST|all>"** parameter is given (or if run in *PoE Port configuration* context), only port specific settings for the given port(s) are listed.

Default values List PoE settings for all ports.

Error messages None defined yet.

12.3.7 Show Enable/Disable PoE Setting

Syntax `show enable`

Context *PoE Port* configuration context.

Usage Show whether PoE is enabled (or disabled) on this port.

Default values Not applicable

Error messages None defined yet.

12.3.8 Show PoE Allocation Priority Setting

Syntax `show priority`

Context *PoE Port* configuration context.

Usage Show PoE allocation priority setting on this port.

Default values Not applicable

Error messages None defined yet.

12.3.9 Show PoE Power Limit Setting

Syntax `show limit`

Context *PoE Port* configuration context.

Usage Show PoE power limit setting on this port.

Default values Not applicable

Error messages None defined yet.

12.3.10 Show PoE Settings

Syntax `show poe [full] [port <PORTLIST>]`

Context *Admin Exec* context.

Usage Show PoE global and per port status.

Use **”show poe”** (or **”show poe port <PORTLIST>”**) to list global PoE status information, and a status summary for all PoE ports (or a given subset).

Use **”show poe full”** (or **”show poe full port <PORTLIST>”**) to list global PoE status information, and detailed status information for all PoE ports (or a given subset).

Default values Not applicable

Error messages None defined yet.

Chapter 13

Virtual LAN

WeOS supports static port based VLANs and VLAN tagging according to IEEE 802.1Q[14]. In addition, WeOS supports WeOS Adaptive VLAN Trunking (AVT) to simplify VLAN configuration in larger WeOS networks.

Section 13.1 provides general information about the VLAN properties and VLAN management features in WeOS. This section also covers features available to manage and inspect the MAC forwarding database on WeOS devices.

Section 13.3 covers VLAN settings via the Web interface, and section 13.4 covers VLAN and MAC forwarding database settings via the CLI.

13.1 Overview of VLAN Properties and Management Features

Table 13.1 summarises VLAN management features in WeOS. Section 13.1.1 provides general VLAN information and sections 13.1.2-13.1.6 contain further information on specific VLAN features.

13.1.1 Introduction to VLANs

Virtual LAN (VLAN) technology is used to create a set of separate LANs over a single physical LAN infrastructure. Each VLAN constitutes a broadcast domain, and traffic on one VLAN is (logically) isolated from traffic on another VLAN. WeOS

Feature	Web	CLI	General Description
General VLAN functionality			
Enable/disable dynamic VLAN	X	X	Sec. 13.1.7
Per VLAN functionality			
Add/modify/delete VLAN	X	X	Secs. 13.1.1-13.1.3
Enable/disable VLAN	X	X	
VLAN name		X	
Untagged/Tagged ports	X	X	Sec. 13.1.1
VLAN priority	X	X	Sec. 13.1.4
IGMP Snooping	X	X	Sec. 13.1.5
VLAN CPU Channel		X	Sec. 13.1.6
Forbid ports	X	X	Sec. 13.1.7
Port-based access control	X	X	Sec. 13.2
View VLAN settings	X	X	
View VLAN status	X	X	
MAC forwarding database functionality			
Set MAC aging timeout		X	Sec. 13.1.8
Set static MAC filters		X	Sec. 13.1.8
View forwarding database settings		X	
View forwarding database status		X	

Table 13.1: Summary of VLAN management features.

supports creation of static port based VLANs and VLAN tagging as described further in this section. We start with two examples to explain the terms *untagged* and *tagged*.

Fig. 13.1 shows a situation where three networks, the *ADMIN* VLAN, the *OFFICE* VLAN, and the *MARKETING* VLAN share a single switch.

- Each VLAN is assigned a VLAN identifier, a VLAN ID (VID); in this example VIDs 1 (ADMIN), 2 (OFFICE) and 3 (MARKETING).
- Each VLAN is assigned a set of ports. In this example ports 1/1-1/2 are associated with the ADMIN VLAN, Ports 2/1-2/4 with the OFFICE VLAN, and ports 2/5-2/8 with the MARKETING VLAN.

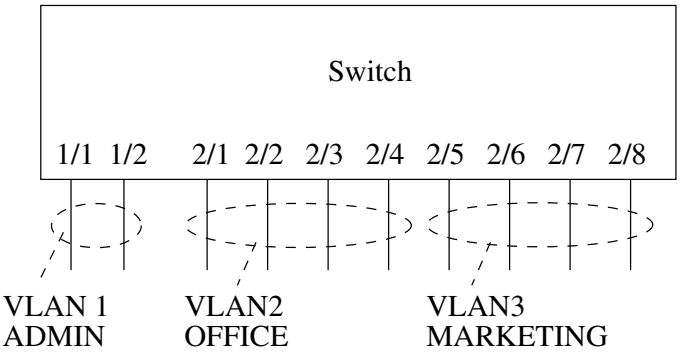


Figure 13.1: VLANs sharing a single switch.

In this example we have assumed that only regular hosts (PCs, servers, etc.; not other switches) attach to the ports of the switch. Traffic sent and received on each switch port are regular Ethernet packets (without VLAN headers), and here we refer to this by saying that the switch ports are associated with their respective VLAN *untagged*.

Note
A port associated *untagged* on a VLAN, will send and receive regular Ethernet packets (i.e., without VLAN header) on that port.

Consider the case where a PC attached to port 2/1 of the switch in fig. 13.1 transmits a *broadcast* packet. That packet will be forwarded onto all other ports of VLAN 2 (OFFICE), i.e., ports 2/2-2/4, but not to any of the other ports.

Fig. 13.2 shows a situation where three networks, the *ADMIN* VLAN, the *OFFICE* VLAN, and the *MARKETING* VLAN share two switches as well as the connection between them.

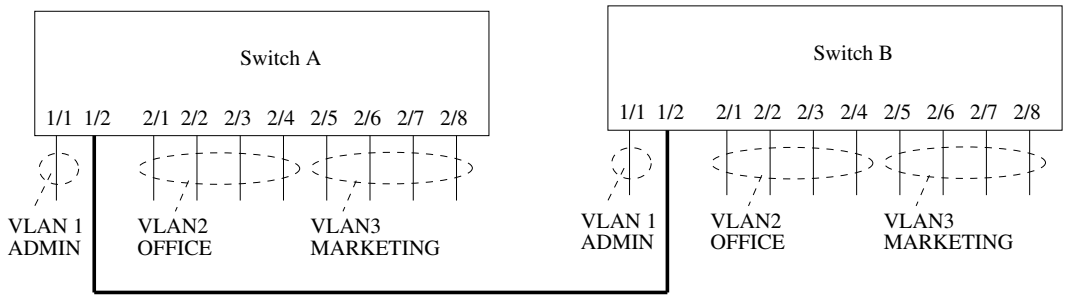


Figure 13.2: VLANs sharing two switches and the connection between them.

- As in the previous example, each VLAN is assigned a VID; in this example VIDs 1 (ADMIN), 2 (OFFICE) and 3 (MARKETING).
- Each VLAN is assigned a set of ports. (For simplicity of this example, we have chosen to use the same port assignment on both switches.) Port 1/1 is associated (untagged) with the ADMIN VLAN, Ports 2/1-2/4 are associated (untagged) with the OFFICE VLAN, and ports 2/5-2/8 are associated (untagged) with the MARKETING VLAN.

In addition, port 1/2, where the cable between the two switches is connected, is associated with all three VLANs. In order for the switches to distinguish which VLAN a packet belongs to when transmitted over a shared connection, the switch will insert a VLAN header (VLAN tag) into the packet, which includes information about the VLAN ID (here 1, 2 or 3). Thus, in this example port 1/2 would be associated with VLAN 1, 2 and 3 *tagged*[1].

Note
A port associated *tagged* on a VLAN, will send and receive *tagged* Ethernet packets (i.e., Ethernet packets including a VLAN header) on that port.

Consider the case where a PC attached to port 2/1 of *switch A* in fig. 13.2 transmits a *broadcast* packet. That packet will be forwarded onto ports 2/2-2/4 of switch A *untagged*, and onto port 1/2 of switch A *tagged* with VID 2. When the *tagged* packet is received on port 1/2 on switch B, that switch can determine that the packet belongs to VLAN 2, and will forward it onto ports 2/1-2/4 *untagged*.

Note
A port cannot be associated with more than one VLAN *untagged*. A port cannot be associated both *untagged* and *tagged* with the same VLAN.

We refer to the VLAN with VID 1 as the *switch default VLAN*. Ports not associated with any VLAN (*untagged* or *tagged*) will automatically be associated with the default VLAN. Section 13.1.3 provides more information on the *default VLAN*.

For each VLAN on a switch, an associated network interface will be created. The name of a VLAN network interface is *vlan<VID>*, e.g., *vlan1* for VLAN 1, and *vlan100* for VLAN 100. The network interface can be assigned an IP address (IPv4), and the switch can then be managed remotely via that VLAN. It is also

[1] It is recommended that a port, which is shared between several VLANs, is associated *tagged* with all those VLANs, however, it is possible to configure the port *untagged* on one VLAN and *tagged* on all other VLANs without risk for ambiguity.

possible to *route* IP traffic between network interfaces. For more information on network interfaces and routing, see chapter 19.

Internally, a WeOS switch can have one or more *channels* to the CPU, where each channel has a capacity of 100 Mbit/s or 1000 Mbit/s. Section 13.1.6 describes how VLANs can be mapped to different CPU channels to achieve increased routing performance.

Layer-2 priority was described in a previous chapter, see section 8.1.4. In addition to different per port priority settings, it is possible to assign specific layer-2 priority per VLAN, see section 13.1.4.

The switch supports efficient distribution of IP multicast packets by use of *IGMP snooping*. See section 13.1.5 for more information on per VLAN IGMP snooping features.

The switch provides support for dynamic VLANs by WeOS Adaptive VLAN Trunking (AVT). AVT can be used to simplify VLAN configuration in larger WeOS LAN infrastructures. AVT is described further in section 13.1.7.

13.1.2 Supported number of VLANs and VLAN integrity

Every VLAN needs to be associated with a unique VLAN ID (VID).

- Switches *support* configuration of up to 64 simultaneous VLANs[2].
- Valid VIDs for configuration are in range 1-4094.
- Some VLAN IDs are reserved for specific use - currently this concerns a set of VIDs in use by the FRNT protocol, see section 14.1.3.

Switches only accept packets for VLANs to which the inbound port is associated. Additional rules for accepting a packet is described below:

- When an untagged packet is received on a port, that packet will be mapped to the port's default VID. If the port is associated with that VLAN (tagged or untagged), the packet will be accepted, otherwise dropped.
- The port's default VID will be the VID of the VLAN to which the port is associated *untagged*. If the port is not associated *untagged* to any VLAN, the default VID is set to the *fall-back default-VID* (see also section 8.1.10) if configured, otherwise to VID 1.

[2] Special restriction on DDW-142/DDW-142-485: On these products the limit is 60 VLANs when FRNT is configured on the unit, and 64 VLANs when FRNT is not configured.

- *Priority tagged* packets, i.e., packets with VID 0, will be associated with the port's default VID.
- Typically *tagged* packets (VID in range 1-4094) or priority tagged packets (VID 0) are only accepted on ports where there is at least on VLAN associated *tagged*. In addition, the packet will only be accepted if the inbound port is associated (*untagged* or *tagged*) the VLAN of the packet.

A common MAC address database is used for all VLANs (shared VLAN learning).

13.1.3 Switch default VLAN

In WeOS the VLAN with VID 1 (VLAN 1) is denoted as the *switch default VLAN*. Ports not associated with any VLAN (neither *untagged* nor *tagged*) will automatically be configured *untagged* to the switch default VLAN. This could happen when a port is removed from a VLAN, or when a whole VLAN is removed.

Note
The main purpose of the switch default VLAN is to avoid loss of remote manageability of a switch due to a change in the VLAN configuration. *Without* a default VLAN, you risk to lose remote access to the switch if the ports used to connect to the switch are removed from all VLANs (unintentionally or deliberately).
With the default VLAN feature, the switch is still manageable via those ports, given that proper IP and firewall settings are configured for the network interface associated with the switch default VLAN.

The switch default VLAN cannot be removed. However, it is possible to remove all ports from the default VLAN by assigning them to other VLANs.

13.1.4 VLAN Priority

It is possible to assign an IEEE 802.1p priority to a VLAN. This feature can be useful when an operator likes to assign a higher priority to traffic on a certain VLAN, e.g., a VLAN dedicated for IP telephony.

When a *VLAN priority* is configured, all packets associated with that VLAN will be treated according to the given VLAN priority, rather than basing the packet's priority on VLAN tag priority, IP ToS/DiffServ or inbound port identifier. For more information on layer-2 priority, see section 8.1.4.

13.1.5 IGMP Snooping and VLANs

Switches use IGMP snooping for efficient distribution of IP(v4) multicast over the LAN. With IGMP snooping *enabled* on a VLAN, IP multicast packets will only be forwarded onto ports leading to a receiver of that IP multicast address, or to ports assumed to lead to an IP multicast router.

With IGMP snooping *disabled* on a VLAN, multicast traffic will be forwarded on all ports of that VLAN, i.e., it is treated similar to broadcast traffic.

By default IGMP snooping is enabled on each newly created VLAN. More information on IGMP Snooping and IGMP Snooping settings is found in chapter 18.

13.1.6 Mapping VLANs to a CPU channel

A switch can have one or more channels to the switch CPU, each with a capacity of 100 Mbit/s or 1000 Mbit/s[3]. By default every new VLAN (with a network interface) is mapped to CPU channel ”0” (zero).

On devices with multiple CPU channels increased routing performance may be achieved by assigning different VLANs to different CPU channels. Assume VLANs 1 and 2 are mapped to the same CPU channel of 100 Mbit/s capacity. Then the maximum theoretical routing throughput between the two VLAN interfaces is 50 Mbit/s full duplex, while the maximum theoretical routing throughput would be 100 Mbit/s full duplex if these VLANs were mapped to different CPU channels.

Note
Routing performance may also be limited by CPU performance, packet size and enabled services.

A VLAN can only be mapped to a single CPU channel.

13.1.7 Dynamic VLANs

WeOS provides dynamic VLAN support via the WeOS Adaptive VLAN Trunking (AVT) protocol. With AVT enabled, VLAN configuration on *inter-switch links* is

[3] WeOS products with ”Corazon” platform (see section 1.5) have 1000 Mbit/s channels to CPU, while others have 100 Mbit/s channels.

simplified - once a switch detects that it is connected to another switch, all VLANs defined on the local switch will automatically be added to that port, see fig. 13.3.

Future versions of WeOS may include dynamic VLAN support via the standard IEEE GVRP[14] protocol in addition to AVT.

13.1.7.1 Determining Inter-Switch Ports

To determine if a port on a switch is connected to another switch, AVT will utilise information from the FRNT and RSTP protocols:

- *FRNT:* If FRNT is enabled on the switch, any port configured as an FRNT port will be classified as an inter-switch port by AVT. If FRNT is disabled, or if the FRNT port configuration is changed, AVT will adapt its inter-switch port classification accordingly. For more information on FRNT, see chapter 14.
- *RSTP:* If RSTP is enabled on a port, AVT will consider the reception of an RSTP or STP message as a sign that it is connected to another switch on the receiving port. The port will continue to be classified as an inter-switch port until the link goes down or until RSTP is disabled on that port. For more information on RSTP, see chapter 16.

13.1.7.2 Dynamic addition/deletion of VLANs to Inter-Switch Ports

Once a port has been defined as an inter-switch port, that port will dynamically be associated (tagged) with all VLANs *configured on the switch*. The exception is when that port has been configured in association mode *forbid* on some VLAN(s) - the port will *not* be associated with those VLANs.

Further details of the mechanism to associate VLANs dynamically to an inter-switch port are given below:

- *Association mode of dynamically added VLANs:* All VLANs configured on the switch will be associated *tagged* by AVT. This applies even to those VLANs configured *untagged* on that port. Fig. 13.3 shows an example.

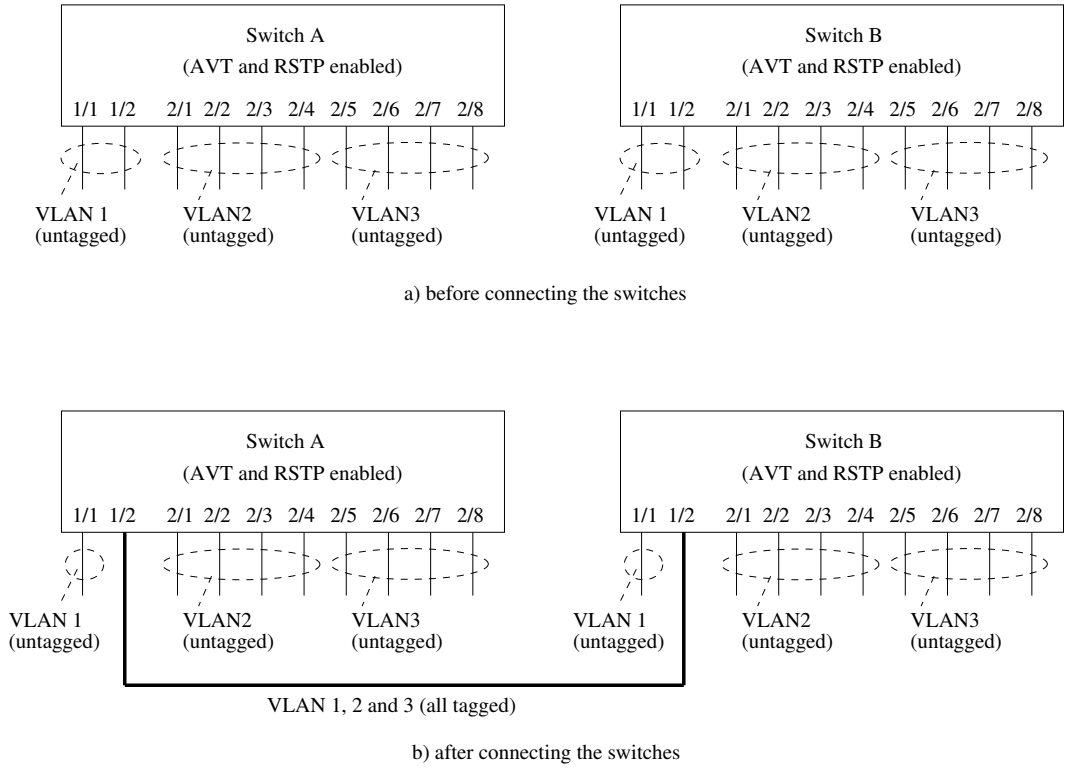


Figure 13.3: Using Adaptive VLAN trunking (AVT) to dynamically add VLANs to inter-switch ports.

> **Note**
> As AVT only considers the VLANs configured on the (local) switch when adding VLANs to an inter-switch port, the operator of the LAN infrastructure should ensure that all switches have the same set of VLANs defined. Otherwise the VLANs forwarded by different switches will be inconsistent, resulting in lack of full connectivity on some VLAN(s).

- *Removing dynamically added VLANs:* When a port loses its status as inter-switch port, all VLANs dynamically added to that port will be removed. The port will then only be associated with the VLANs it has been configured with, and with association mode (tagged or untagged) according to the configuration.
- *Prohibiting that a VLAN is added to a port:* It is possible to prohibit that some VLAN(s) is dynamically added to a port even when AVT is enabled.

This feature is useful when the unit acts as a routing switch, where traffic between some ports should be *routed* rather than *switched*.

To prohibit that a VLAN is dynamically added to a port, that port should be configured with association mode *forbid* on that VLAN.

As of WeOS version v4.17.1 the *forbid* association mode only hinders a port to be added to a VLAN dynamically via AVT. Ports not configured untagged/tagged with any VLAN will still be mapped to the switch default VLAN (VLAN 1), irrespective if that port is configured as *forbid* on VLAN 1. For more information about the switch default VLAN, see section 13.1.3.

13.1.7.3 Prohibit disabling of Inter-Switch Ports

A port determined as inter-switch port by AVT will not be possible to disable by management (Web, CLI, SNMP, etc.). This feature is added in order to avoid unintentional loss of connectivity to the switch.

13.1.8 MAC forwarding database

WeOS switches maintain a MAC forwarding database holding information about where to forward packets for each known MAC address. As of WeOS v4.17.1 a single MAC forwarding database is used for all VLANs, referred to as *shared VLAN learning* in [14].

13.1.8.1 Managing Unicast MAC addresses

When the switch comes up, it will not know which stations are attached to its ports. The switch inspects the destination MAC address of each incoming packet without finding a match in the forwarding database - unknown unicast MAC addresses will be broadcasted on all ports of the associated VLAN.

The switch will automatically learn the location of stations in the LAN, by inspecting the source MAC address of each incoming packet. Once it knows on which port a certain MAC address resides, all future packets to that station will be forwarded only onto that port.

Note
Switches ”learn” the location of (unicast) MAC address by inspecting the ”source” MAC address, while they ”forward” packets based on the ”destination” MAC address.

Unicast MAC addresses learnt automatically will stay in the MAC forwarding database until they are aged out – the aging timeout defaults to 300 seconds. The aging timeout is configurable, and aging can be disabled.

13.1.8.2 Managing Broadcast and Multicast MAC addresses

Packets transmitted to the *broadcast MAC address* (”ff:ff:ff:ff:ff:ff”) will be forwarded onto all ports in the associated VLAN. Other group MAC addresses (here referred to as multicast MAC addresses) are handled differently if *IGMP Snooping* is enabled or not (see chapter 18 for detailed information on IGMP Snooping):

- *IGMP Snooping Disabled:* With IGMP Snooping disabled on a VLAN, packets sent to multicast MAC addresses will be handled in the same way as broadcast, i.e., such packets will be forwarded onto all ports in the associated VLAN.
- *IGMP Snooping Enabled:* With IGMP Snooping enabled on a VLAN, packets sent to multicast MAC addresses will be blocked on all ports by default, and only forwarded onto ports (1) where the switch has learnt that there is a host interested in receiving traffic to that multicast MAC address, or (2) which the switch believes lead to a multicast router.

WeOS also allows an operator to manually specify where to forward multicast MAC addresses, i.e., the operator can add *static multicast MAC filters*. This feature is useful for several reasons:

- *IGMP snooping and non-IP multicast:* With IGMP snooping enabled, all MAC multicast will be blocked, except those learnt via IGMP snooping. As IGMP snooping only learns MAC multicast based on IP multicast, all other types of MAC multicast will be blocked.

  Adding static MAC filters enables the use of non-IP multicast on VLANs where IGMP snooping is enabled.
- *IGMP Snooping and IP multicast in the 224.0.0.X range:* IP multicast in the *224.0.0.X* range should be forwarded onto all ports in the VLAN irrespective if any host has indicated interest in that multicast address via IGMP or not.

In WeOS the operator has the flexibility to select which addresses in the 224.0.0.X range to forward on a LAN, by adding filters for the corresponding multicast MAC address. The factory default configuration includes MAC filters for some of the most common multicast addresses in the 224.0.0.X range, which are then forwarded onto all ports even if IGMP snooping is enabled.

When specifying the destination port list in a MAC filter, one can specify both regular Ethernet (and DSL) ports, as well as the internal CPU port(s) of the switch. The latter is used if the multicast packet should be processed by the switch itself.

13.2 Port-based network access control

WeOS supports port-based network access control (PNAC). This security feature is used to stop unauthorised PCs or other equipment to access the network. Authentication is required to gain access. WeOS provides two authentication methods: *IEEE 802.1X* and *MAC based authentication*.

Ports with access control enabled (i.e., *controlled ports*) will by default be ”blocked” for incoming traffic. Only when a connected device has successfully authenticated itself will it be allowed/authorised 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.

Incoming broadcast and multicast packets from unauthorised devices will also be blocked. *Outgoing* broadcast and multicast packets will, however, **not** be blocked and are sent out as usual on *controlled* ports. IGMP joining of multicast groups will not work for unauthorised clients, as incoming IGMP join messages are dropped until the client is granted access.

In WeOS, port-based network access control is managed per VLAN. Enabling access control on a VLAN implies that all *untagged* ports on that VLAN are subject to access control by default. Often some or a few ports need to be excluded from access control, e.g., ports connected to a server, uplink ports (towards Internet), and VLAN trunk ports. These ports can be excluded by a special configuration option in the CLI **”except-auth”** (see section 13.4.17) or in the web GUI (see section 13.3.5).

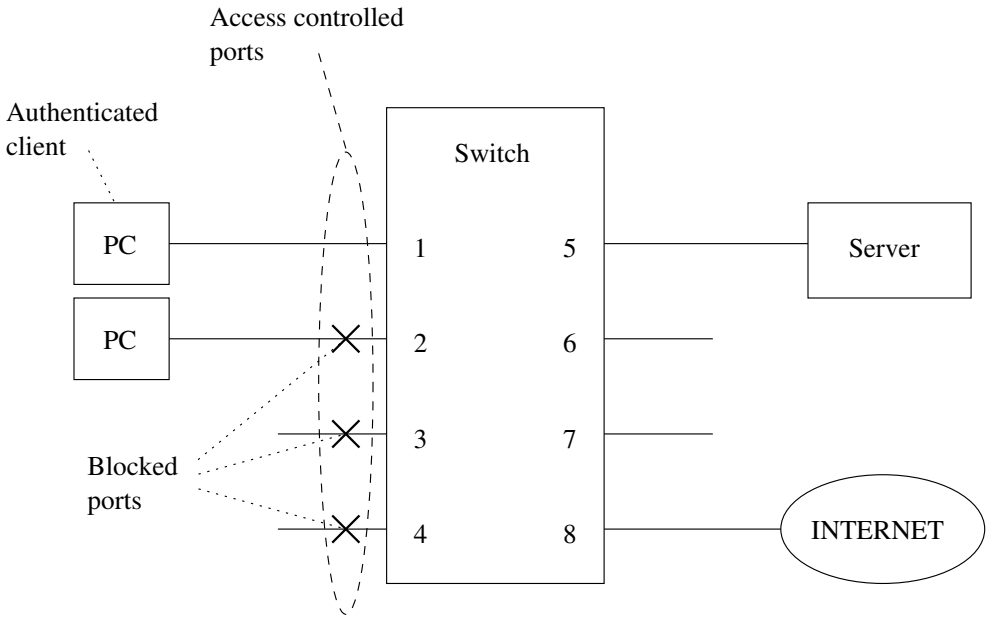


Figure 13.4: Port-based network access control

Port-based access control and VLAN trunk ports

As of WeOS v4.17.1, port-based access control is only working as expected for access ports, i.e., ports only associated with a single VLAN (untagged). VLAN trunk ports (ports associated tagged to one or more VLANs) should be excluded from access control. Although it is possible to have access control enabled on such ports, the behaviour is neither defined nor supported, and may change in future WeOS releases.

In order to acquire access, the connected device needs to authenticate itself to the switch. See fig. 13.4 for a scenario. The PC on port 1 has authenticated itself, whereas the one on port 2 has not. The first PC is able to access the server or the Internet connection on ports 6 and 8. The second PC or anything connected to ports 3 or 4 will be blocked by the switch until they have authenticated themselves.

The two authentication mechanisms available in WeOS for port-based network access control are described further below: *IEEE 802.1X* in section 13.2.1 and *MAC based authentication* in section 13.2.2.

13.2.1 Authentication using IEEE 802.1X

WeOS units are able to act as IEEE 802.1X [15] *authenticators*. WeOS uses the RADIUS[34] protocol with extensions for Extensible Authentication Protocol (EAP[33]) to communicate to a backend *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 of WeOS v4.17.1, WeOS does not support *Authenticator initiation* as defined by §8.4.2.1 in the IEEE 802.1X standard[15]. The 802.1X client (*supplicant*) must initiate the authentication procedure to gain access[4].

Fig. 13.5 illustrates 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 backend 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.

When configuring the 802.1X authenticator in WeOS, the RADIUS server (or group of RADIUS servers) must be specified. The procedure is as follows:

1. *RADIUS server settings (AAA):* Enter the appropriate settings for your RADIUS server(s): IP address, password, etc. See chapter 21 on Authentication, Authorisation and Accounting (AAA) for more information.

2. *Define RADIUS server group (AAA):* (Optional) The RADIUS servers can be grouped together, simplifying configuration in some cases. See chapter 21 on AAA for more information.

3. *Define AAA instance(s) for 802.1X (AAA):* To allow individual RADIUS servers or server groups to be used as 802.1X authentication backends, they need to be listed in an 802.1X AAA instance. See chapter 21 on AAA for more information.

4. *Enable 802.1X per VLAN:* When 802.1X is enabled on a VLAN, the relevant AAA instance is defined, thereby defining which RADIUS server(s) to relay 802.1X messages to from this VLAN. See sections 13.3.4 (Web) and 13.4.15 (CLI) for further details.

[4]The 802.1X supplicants included with Microsoft Windows, Ubuntu Linux and most other equipment supports supplicant initiation.

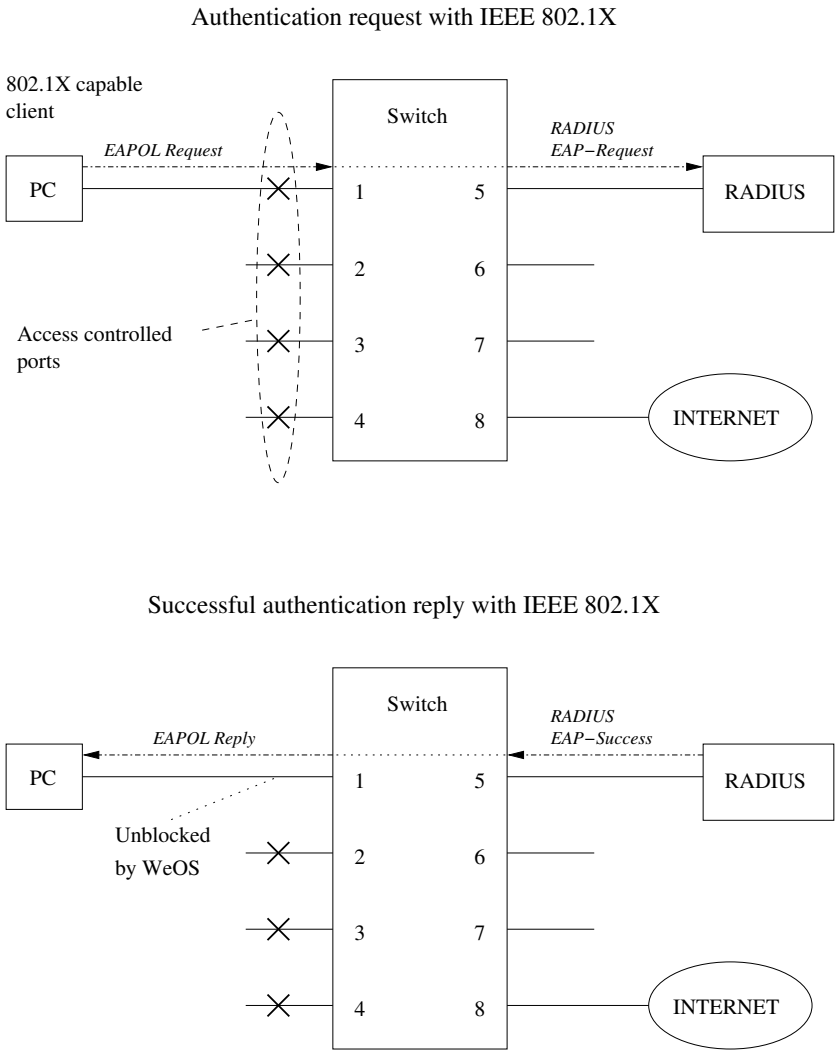


Figure 13.5: Principles of authentication with IEEE 802.1X and RADIUS

13.2.2 Authentication based on MAC addresses

Authentication can be based on the client's MAC address. This is often combined with IEEE 802.1X authentication to grant access to 802.1X capable devices and legacy equipment lacking 802.1X support. When combined, MAC authentication will have precedence over 802.1X 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 set up using lists 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 enabling MAC authentication on a VLAN in WeOS, the associated MAC list (white-list) must be specified. The procedure is as follows:

1. *Create MAC Authentication List (AAA):* Create a MAC list, and add MAC patterns to that list. A MAC pattern by default applies to all ports on the VLAN the MAC list will be mapped to, however, the MAC pattern may apply to a specific port. See chapter 21 on Authentication, Authorisation and Accounting (AAA) for more information, in particular sections 21.3.20-21.3.23 (CLI), and 21.2.16 (Web).
2. *Enable MAC authentication per VLAN:* When MAC authentication is enabled on a VLAN, the relevant MAC list is specified, thereby defining which MAC addresses to grant access. Access is granted on all ports, except for MAC patterns limited to a specific port. See sections 13.3.4 (Web) and 13.4.15 (CLI) for further details.

The switch will listen on the controlled ports for Ethernet packets originating from currently unknown MAC addresses. When such a packet arrives, it will use the packet's source MAC and search through the specified MAC list for a matching entry. If one is found, the port will be opened for the specific MAC address. Packets that do not match will be discarded (alternatively, such packets can be authentication via 802.1X).

A port will remain open for an authorised MAC as long as traffic flows. If no packets is received through the port from an authorised MAC address for 5 minutes[5], the port will be closed again for this address, and the authentication procedure will be re-done when new packets arrive.

As of WeOS v4.17.1 does **not** support MAC based authentication with a backend authentication server (e.g, RADIUS).

[5] MAC aging time is by default 5 minutes, see sections 13.1.8.1 and 13.4.2 for more information.

13.3 Managing VLAN settings via the web interface

Menu path: Configuration ⇒ VLAN ⇒ VLANs

When entering the VLAN configuration page you will be presented to a list of all VLANs configured on your switch, see below. Here you get an overview of the settings for all VLANs and you can create or delete VLANs. The default VLAN (VID 1) cannot be removed (see section 13.4.6). To change the settings for a specific VLAN, click the edit icon which will take you to the VLAN settings edit page.

VID	The VLAN's unique identifier.
Name	The name of the VLAN. Automatically generated from VLAN identifier when the VLAN is created using the web tool.
Enabled	Used to enable or disable a VLAN. Ports on a disabled VLAN are temporarily moved to the system default VLAN. A green check-mark means the VLAN is enabled, and a dash means it is disabled.
Status	Current operational status of the VLAN, **Up** or **Down**.
Prio	VLAN priority setting. Values between 0-7 or disabled. See also section 13.1.4. Disabled is shown using a dash.
IGMP	In the VLAN overview table a green check-mark means that IGMP snooping is enabled, and a dash means it is disabled, on a specific VLAN. See section 13.1.5 for more information.
Interface	A list of associated interfaces.

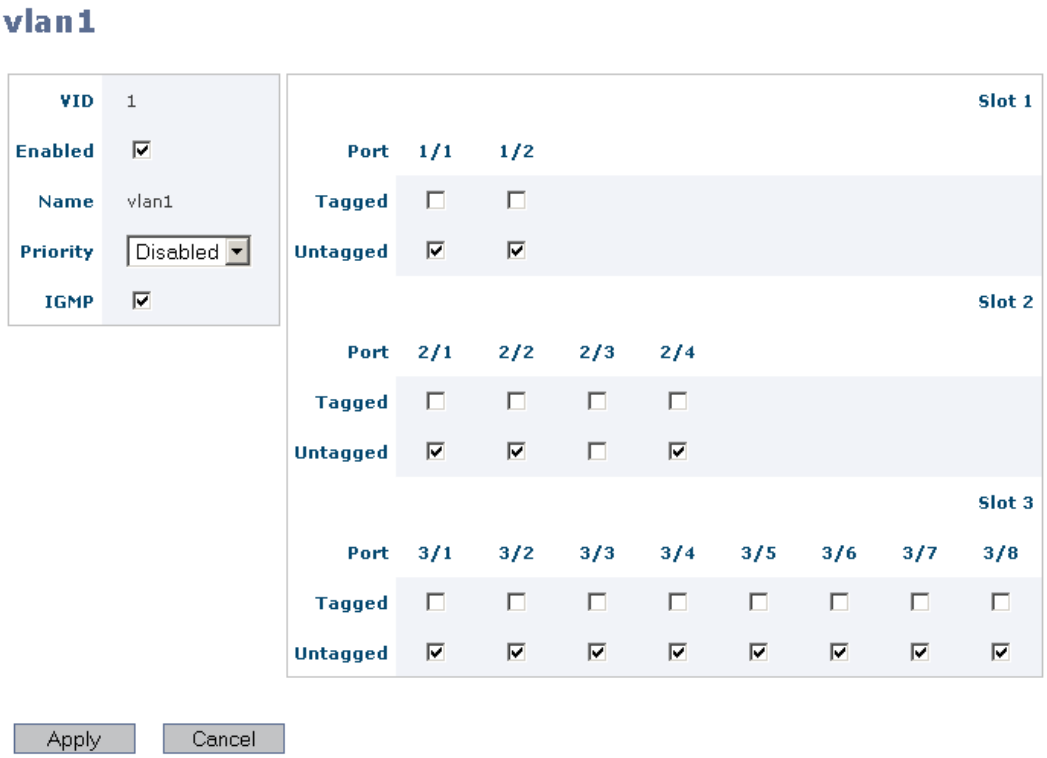
Continued on next page

	Continued from previous page
Port(s)	List of ports assigned to each VLAN. Grouped as **tagged** and **untagged** for ports configured statically to this VLAN, or as **dynamic** for ports dynamically added to this VLAN by WeOS Adaptive VLAN Trunking (AVT). (See section 13.1.7 for more information on AVT). 1/1-1/3 means port 1/1, 1/2 and 1/3, the first and last port, and all ports in-between.
New VLAN	Click this button to create a new VLAN. You will be presented to a form where you can configure the new VLAN.
 **Edit**	Click this icon to edit a VLAN.
 **Delete**	Click this icon to remove a VLAN. You will be asked to acknowledge the removal before it is actually executed.

13.3.1 Edit VLAN settings using the web interface

Menu path: Configuration ⇒ VLAN ⇒ VLANs ⇒ 

When clicking the *Edit* icon for a VLAN you will be presented to the VLAN edit page.



On **`VLAN Edit page`** you can change the settings for the VLAN as described below:

VID	The VLAN's unique identifier. You cannot change the VID of an already created VLAN.
Enabled	Used to enable or disable a VLAN. Ports on a disabled VLAN are temporarily moved to the system default VLAN. To enable the VLAN - check the box, to disable un-check the box.
Name	The name of the VLAN. You cannot change the VLAN name using the web tool.

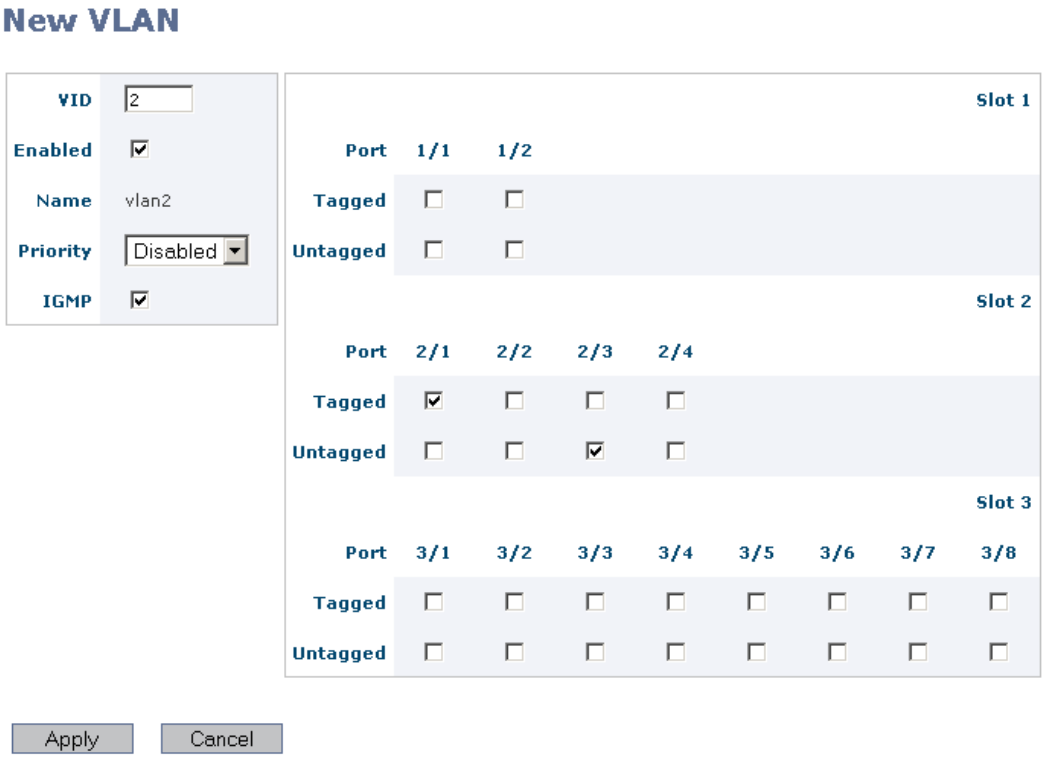
Continued on next page

	Continued from previous page
Prio	VLAN priority setting. Values between 0-7 or disabled. See also section 13.1.4. Select the desired VLAN priority in the drop down list, or select disable to disable VLAN priority.
IGMP	To enable IGMP snooping on this VLAN - check the box, to disable IGMP un-check the box. See section 13.1.5 for more information.
Port	The ports on your switch is grouped as on the actual hardware, in slots. To assign a port to the VLAN, check the **Tagged** or **Untagged** check-box located underneath the port label. In the picture above you see all ports but 2/3 associated *untagged* to VLAN 1. A port may not be associated tagged and untagged to the same VLAN at the same time. It may not be associated untagged to more than one VLAN at a time. If you associate a port untagged to a VLAN any existing untagged association to another VLAN on that port will automatically be removed. You will be notified if this happens. For more information on the *tagged* and *untagged* association modes, see section 13.1.1. The **Forbidden** check-box is used to specify that this port can not be dynamically assigned to this VLAN (see section 13.1.7 for more information on dynamic VLANs).

13.3.2 Create a new VLAN using the web interface

Menu path: Configuration ⇒ VLAN ⇒ VLANs ⇒ **New VLAN**

When clicking the **New VLAN** button you will be presented to the **new VLAN** page.



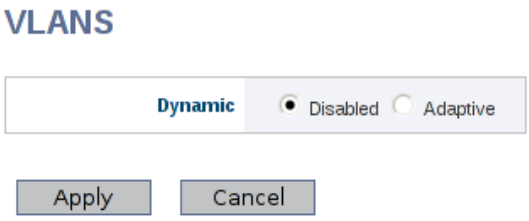
The **New VLAN** and the **Edit VLAN** pages differ only by the possibility to change the VID (VLAN ID). See section 13.3.1 for additional attribute descriptions.

VID	The VLAN's unique identifier.
Name	The VLAN name will be automatically generated when using the web management tool. The name is shown directly when you change and leave the VID field if your browser is JavaScript enabled, otherwise it will be generated when you click the **Apply** button.

13.3.3 Managing Dynamic VLAN using the web interface

This enables WeOS Adaptive Dynamic Trunking (AVT) on the switch. For more information on AVT in section 13.1.7.

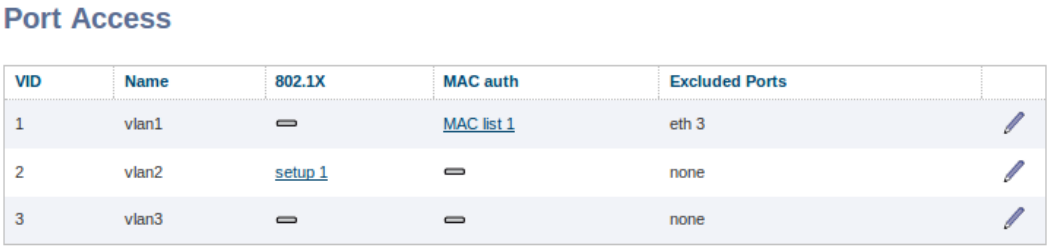
Menu path: Configuration ⇒ VLAN ⇒ Dynamic



13.3.4 Managing port-based access control using the web interface

Menu path: Configuration ⇒ VLAN ⇒ Port Access

The VLAN Port Access page shows an overview of the currently configured VLANs with the port-based access control settings.

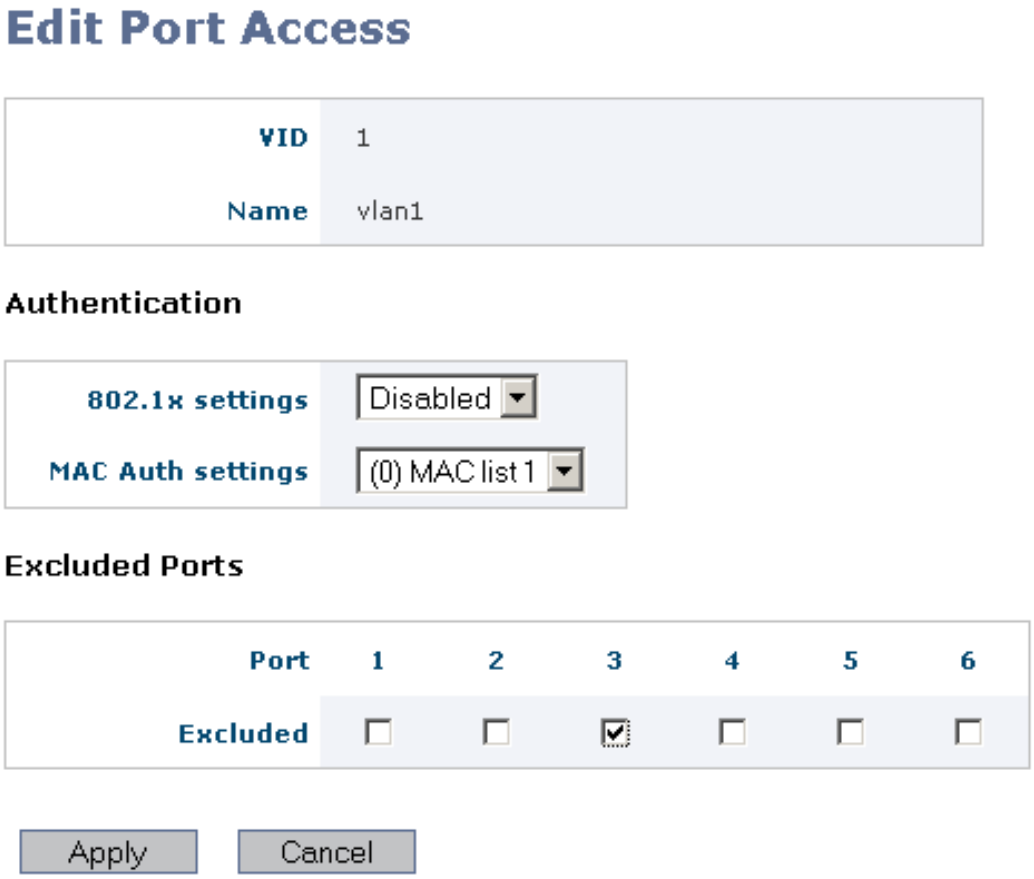


VID	The VLAN's unique identifier.
Name	The name of the VLAN.
802.1X	The description of the referenced 802.1X configuration, a dash means it is disabled. See section 21.2.13 for configuration of 802.1X.
MAC auth	The description of the referenced MAC authentication configuration, a dash means it is disabled. See section 21.2.16 for configuration of MAC authentication
Edit	Click this icon to edit the port access configuration for this VLAN.

13.3.5 Edit port-based access control settings

Menu path: Configuration ⇒ VLAN ⇒ Port Access ⇒ 

When clicking the *Edit* icon for a VLAN you will be presented to the VLAN Port Access edit page.



VID	The VLAN's unique identifier.
Name	The name of the VLAN.
802.1X settings	Enable IEEE 802.1X authentication for ports on this VLAN by selecting a 802.1X configuration. See section 21.2.13 for how to create and edit the 802.1X configurations.

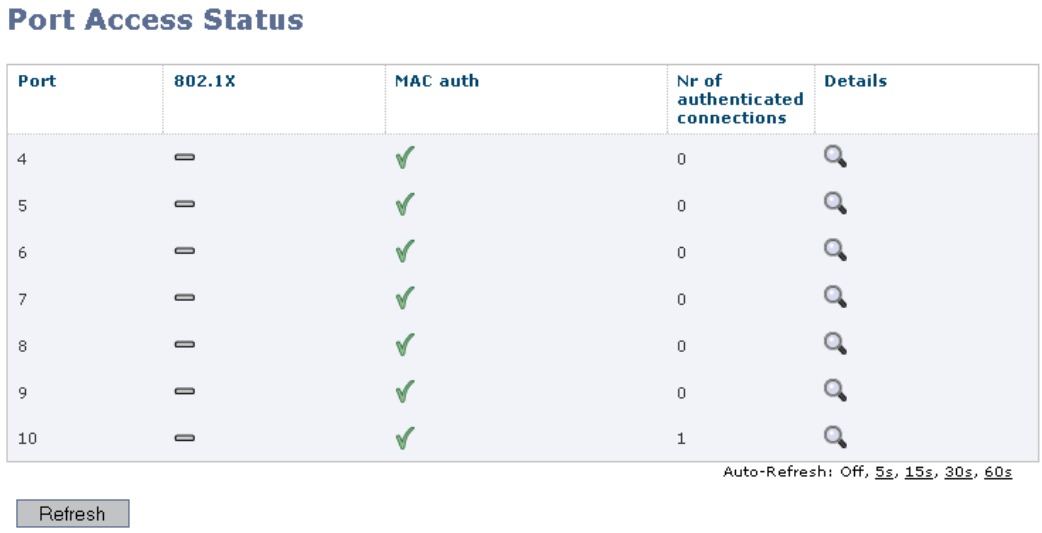
Continued on next page

	Continued from previous page
MAC Auth settings	Enable MAC based authentication by selecting a configuration. See section 21.2.16 for managing MAC authentication configurations.
Excluded Ports	The ports on your switch is grouped as on the actual hardware, in slots. Check the box underneath the port label to exclude that port from access control. An excluded port will be open and does not require authentication. This is suited for uplink ports, trunk ports and for connecting servers. The default for ports is unchecked, thus enabling port access control/authentication. Check-boxes can be shown as disabled, like port 1 and 2 in the above picture. This means that the current VLAN does not have this port as a member and is therefore not relevant for exclusion. See section 13.3.1 for managing the relations between ports and VLANs.

13.3.6 Port-based access control statistics

Menu path: Status ⇒ Port Access

Here you can see an overview over port access status on a per-port basis. The 802.1X column shows if IEEE 802.1X is enabled for a port or not. The MAC auth column shows if MAC based authentication is enabled.
You can also see the current number of authenticated hosts. This value is only showing hosts that have authenticated recently. There may be more hosts on the network that can be authenticated via MAC based authentication but are inactive on the network for the moment. See section 13.2.2 for information about inactivity and MAC based authentication.



A detailed view of the authenticated hosts is shown if you click on the magnifier icon for a port. This view shows all authenticated host by their MAC address. This list shows hosts that are authenticated with both IEEE 802.1X and MAC based authenticated together.

13.4 Managing VLAN settings via the CLI

Command	Default	Section
MAC Forwarding Database Configuration		
fdb		Section 13.4.1
[no] aging-timeout <0|1-3825>	300	Section 13.4.2
[no] mac <MACADDR> port <PORTLIST>		Section 13.4.3
General VLAN Configuration		
[no] vlans		Section 13.4.4
[no] dynamic <adaptive>	Disabled	Section 13.4.5
Per VLAN Configuration		
[no] vlan <VID>		Section 13.4.6
[no] enable	Enabled	Section 13.4.7
name <VLANNAME>	vlan<VID>	Section 13.4.8
[no] untagged <PORTLIST>		Section 13.4.9
[no] tagged <PORTLIST>		Section 13.4.10
[no] forbid <PORTLIST>		Section 13.4.11
[no] priority <0-7>	Disabled	Section 13.4.12
[no] igmp	Enabled	Section 13.4.13
channel <CHANNELID>	0	Section 13.4.14
[no] dot1x-auth <ID>	Disabled	Section 13.4.15
[no] mac-auth <ID>	Disabled	Section 13.4.16
[no] except-auth <PORTLIST>	Disabled	Section 13.4.17
Show VLAN Status and MAC Forwarding Database Status		
show vlans		Section 13.4.18
show fdb		Section 13.4.19
Show Port-Based Access Control Status		
show dot1x-auth		Section 13.4.20
show mac-auth		Section 13.4.21

13.4.1 Managing MAC Forwarding Database Settings

Syntax `fdb`

Context Global Configuration context

Usage Use the **"fdb"** command to enter the MAC Forwarding Databas context (*fdb*).

Use **"show fdb"** to show current FDB settings (list of configured MAC address filters, and the configured aging timeout). Also available as **"show"** command within the MAC Forwarding Databas.

Default values Not applicable.

13.4.2 Configure MAC Address Aging Timeout

Syntax `[no] aging-timeout <0|1-3825>`

Context MAC Forwarding Databas context (*fdb)*

Usage Set the aging timeout (in seconds) for unicast MAC addresses learnt dynamically. The configured aging timeout will only be an approximation of the actual aging timeout. The value is first rounded upwards in steps of 15 seconds. The MAC entries will be purged from the forwarding database within 1/7th of the resulting aging timeout.

Use **"no aging-timeout"** or **"aging-timeout 0"** to disable aging entirely.

Use **"show aging-timeout"** to view the current setting.

Default values 300 (seconds)

13.4.3 Configure Static MAC Filter Entries

Syntax `[no] mac <MACADDRESS> port <[PORTS] [ALL] [CPU] | [NONE]>`

Context MAC Forwarding Databas context (*fdb)*

Usage Add or delete a static MAC address filter. The **"MACADDRESS"** is written as a colon separated hexadecimal value, e.g., **"01:23:45:56:89:AB"**.

The **"PORTLIST"** states the port(s) where packets with the given (destination) MAC address are to be forwarded. As of WeOS v4.17.1, the static MAC

filters are only intended to be used for multicast MAC addresses (not unicast MAC or the broadcast MAC addresses).

The **”PORTLIST”** can include both visual ports (e.g., **”eth 2/1-2/4”** on a slotted WeOS unit) as well as the internal CPU port(s):

- PORT(S): Port, set of or range of ports, e.g. eth 1,3-5
- ALL: All visible ports, excluding internal CPU port(s)
- NONE: No ports, filter this MAC address
- CPU: The internal CPU port(s)

Use **”no MAC <MACADDRESS>”** to remove a specific static MAC filter, or **”no MAC”** to remove all static MAC filters.

Use **”show mac”** to list configured MAC address entries.

Default values (The factory default configuration includes a set of static MAC filters.)

13.4.4 Managing general VLAN settings

Syntax `[no] vlans`

Context Global Configuration context

Usage Enter the General VLAN Configuration context (*vlans*). The General VLAN Configuration context can be used to configure VLAN settings applicable to all VLANs.

Use **”no vlans”** to remove all VLANs except the switch default VLAN (VLAN 1). All ports will be configured *untagged* on VLAN 1.

Use **”show vlans”** to list all configured VLANs and general VLAN settings.

Default values Not applicable.

13.4.5 Enable dynamic VLAN

Syntax `[no] dynamic <adaptive>`

Context General VLAN Configuration context (*vlans)*

Usage Use the **”dynamic adaptive”** command to enable WeOS Adaptive Dynamic Trunking (AVT) on the switch. For more information on AVT in section 13.1.7.

Future versions of WeOS may include support for dynamic VLAN via GVRP in addition to AVT, but currently only AVT is supported.

Use **”no dynamic”** to disable dynamic VLAN support.

Use **”show dynamic”** to see the dynamic VLAN setting.

Default values Disabled

13.4.6 Managing individual VLANs

Syntax `[no] vlan <VID>`

Context Global Configuration context

Usage Enter VLAN Configuration context of the given VID. If this is a new VLAN, the VLAN will be created first upon leaving the VLAN context with *end* or *leave*.

Use **”no vlan <VID>”** to remove an existing VLAN. The default VLAN (VLAN 1) cannot be removed. Removal of a VLAN may imply that some ports will no longer be associated with any VLAN - such ports will be configured to the default VLAN (VLAN 1) untagged.

Use **”show vlan”** (or **”show vlans”**) to list all configured VLANs and general VLAN settings. Use **”show vlan VID”** to list detailed configuration information for a specific VLAN (also available as **”show”** command within the VLAN Configuration context of the given VID.

Default values Not applicable.

Example

```
example:/config/#> show vlan 1
 VLAN ID          : 1
 Status           : Enabled
 Name             : vlan1
 Channel          : 0
 Priority         : Disabled
 Untagged         : U:eth 1-4
 Tagged           : T:
 Forbid           : F:
 IGMP             : Enabled
 Learning         : Enabled
 802.1Q VLAN      : Enabled
 802.1X Auth      : Disabled
 MAC Auth         : Disabled
 Except Port Auth :
example:/config/#>
```

13.4.7 Enable/disable a VLAN

Syntax `[no] enable`

Context VLAN Configuration context

Usage Enable or disable a VLAN. A disabled VLAN is similar to a deleted VLAN, except that its configuration is stored, and will be activated when the VLAN is *enabled*. That is, when a VLAN is disabled, its ports may be moved onto the default VLAN (unless they are associated with another VLAN), and any network interface associated with the VLAN will be disabled.

Use **”`show enable`”** to view the current configuration.

Default values *enable*

13.4.8 VLAN name

Syntax `name <ID>`

Context VLAN Configuration context

Usage Specify VLAN name, i.e., VLAN description. Max 15 characters, only alpha-numerical characters ([a-z,A-Z,0-9]) allowed.

Use **”`show name`”** to view the VLAN name setting.

Default values If no VLAN **”name”** command is given, the VLAN name defaults to *vlanVID*, e.g., *vlan100* for VID 100.

13.4.9 Manage untagged ports

Syntax `[no] untagged <PORT|PORTLIST>`

Context VLAN Configuration context

Usage Associate port(s) with this VLAN VID in *untagged* mode. Only a single VLAN VID can be associated *untagged* with each port. Ports associated with a VLAN VID *untagged* will have that VID as *default VID* - this will have precedence over any (fall-back) default VID configuration set in *port* context.

Use **”no untagged <PORTLIST>”** to remove *untagged* ports from a VLAN. If removal of an *untagged* port implies that the port is no longer associated with any VLAN, that port will be configured to VLAN 1 *untagged*.

Use **”show untagged”** to view ports associated untagged with this VLAN.

Default values Factory default lets all ports be associated with the default VLAN (VLAN 1) *untagged*. For new VLANs, ports must explicitly be added.

Error messages

- A notification message is given in case the addition of port as *untagged* on one VLAN implies that the same port will be removed as *untagged* on another VLAN.
- A notification message is given in case the addition of port as *untagged* on one VLAN implies that the same port will be removed as *tagged* on the same VLAN (a port cannot be associated both *tagged* and *untagged* with the same VLAN).

A **”PORTLIST”** is a comma separated list of port ranges without intermediate spaces, e.g., **”1/1-1/3,2/3”**.

13.4.10 Manage tagged ports

Syntax `[no] tagged <PORT|PORTLIST>`

Context VLAN Configuration context

Usage Associate port(s) with this VLAN VID in *tagged* mode.

Use **”no tagged <PORTLIST>”** to remove *tagged* ports from a VLAN. If removal of a *tagged* port implies that the port is no longer associated with any VLAN, that port will be configured to VLAN 1 *untagged*.

Use **”show tagged”** to view ports associated tagged with this VLAN.

Default values Not applicable.

Error messages A notification message is given in case the addition of port as tagged on one VLAN implies that the same port will be removed as *untagged* on the same VLAN (a port cannot be associated both *tagged* and *untagged* with the same VLAN).

A **”PORTLIST”** is a comma separated list of port ranges without intermediate spaces, e.g., **”1/1-1/3,2/3”**.

13.4.11 Manage forbidden ports

Syntax `[no] forbid <PORT|PORTLIST>`

Context VLAN Configuration context

Usage Prohibit that ports are dynamically added (AVT) to this VLAN ID, see also sections 13.1.7 and 13.4.5.
Use **”no forbid <PORTLIST>”** to remove ports from the list of ports forbidden to be associated with this VLAN.

Use **”show forbidden”** to view ports associated forbidden with this VLAN.

Default values Not applicable.

A **”PORTLIST”** is a comma separated list of port ranges without intermediate spaces, e.g., **”1/1-1/3,2/3”**.

13.4.12 VLAN priority setting

Syntax `[no] priority <0-7>`

Context VLAN Configuration context.

Usage Set the (IEEE 802.1p) priority associated with this VLAN. Incoming packets associated with this VLAN will receive this priority.

”no priority” will disable VLAN priority for this VLAN. Priority for packets associated with this VLAN will then be based on port priority settings.

Use **”show priority”** to view the priority setting for this VLAN.

Default values Disabled (**”no priority”**).

13.4.13 VLAN IGMP Snooping

Syntax `[no] igmp`

Context VLAN Configuration context.

Usage Enable, or disable IGMP Snooping for this VLAN.

Use **”show igmp”** to view the IGMP snooping setting for this VLAN.

Default values IGMP snooping enabled.

13.4.14 CPU channel mapping

Syntax `channel <CHANNELID>`

Context VLAN Configuration context.

Usage Specify CPU channel to use for this VLAN. The channel identifier can take values in the range <0-CHANNELIDMAX>. The purpose of this command is to improve routing performance by mapping VLANs to different CPU channels, see section 13.1.6.

> **Hint**
>
> Use the **”show system-information”** command (see section 7.3.2) to find out the number of channels.
> - Look for the line ”Channel interfaces” in the information of the CPU card to see the number of channels.
> - CHANNELIDMAX equals ”number of channels”-1.

Use **”show channel”** to view the CPU channel setting for this VLAN.

Default values 0 (zero), i.e., by default all VLANs will use channel 0.

13.4.15 IEEE 802.1X authentication

Syntax `[no] dot1x-auth <ID>`

Context VLAN Configuration context.

Usage Specify the IEEE 802.1X configuration to be used for this VLAN. Setting this enables port-based access control for all ports untagged in this VLAN, except for the ports defined with **"except-auth"** (see section 13.4.17). The ID value references the 802.1X configuration. This configuration is managed in the AAA subsystem, see chapter 21. Use **"no dot1x-auth"** to disable IEEE 802.1X authentication for this VLAN.

Use **"show dot1x-auth"** to view the IEEE 802.1X authentication setting for this VLAN.

Default values Disabled, i.e. IEEE 802.1X is not used.

13.4.16 MAC based authentication

Syntax `[no] mac-auth <ID>`

Context VLAN Configuration context.

Usage Specify the MAC authentication configuration to be used for this VLAN. Setting this enables port-based access control for all ports untagged in this VLAN, except for the ports defined with **"except-auth"** (see section 13.4.17). The ID value references the MAC authentication configuration. This configuration is managed in the AAA subsystem, see chapter 21. Use **"no mac-auth"** to disable MAC based authentication for this VLAN.

Use **"show mac-auth"** to view the MAC based authentication setting for this VLAN.

Default values Disabled, i.e. MAC based authentication is not used.

13.4.17 Except ports from authentication

Syntax `[no] except-auth <PORT|PORTLIST>`

Context VLAN Configuration context.

Usage Disables port-based access controls for specific ports. This is used together with **”dot1x-auth”** and **”mac-auth”** to exclude specific ports from needing authentication. This is suitable for uplinks, trunks and ports with servers connected. Use **”no except-auth”** to remove all port exceptions, thus *enabling* access control on all untagged ports in this VLAN.

Use **”show mac-auth”** to view ports configured to be excluded from port-based access control for this VLAN.

Default values Disabled, no ports excluded.

13.4.18 Show VLAN status (all VLANs)

Syntax `show vlans`

Context Admin Exec context

Usage Show VLAN status information for all VLANs.

Default values Not applicable.

13.4.19 Show Current MAC Forwarding Database

Syntax `show fdb`

Context Admin Exec context

Usage Show the current state of the MAC forwarding database. This includes the list of MAC addresses known to the switch, and the port(s) to forward packets to each MAC address. The ageing timeout for automatically learnt unicast MAC addresses is also shown.

Default values Not applicable.

Example

```
example:/#> show fdb
MAC                VLAN  State  Portvec     Port(s)
================================================================================
00:07:7c:81:de:1a   ANY   0x0f  0x0         CPU
00:07:7c:81:de:1d   ANY   0x01  0x0         CPU
00:0d:88:cd:3a:9c   ANY   0x01  0x1         ETH 1/1
01:00:5e:00:00:01   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:02   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:04   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:05   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:06   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:09   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:0a   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:0d   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:0e   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:12   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:18   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:66   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:6b   ANY   0x07  0x3fff      ALL
01:00:5e:00:00:fb   ANY   0x07  0x3fff      ALL
01:80:c2:00:00:0e   ANY   0x07  0x3f        ETH 1/1-ETH 2/4
FDB Aging time: 300 sec.
example:/#>
```

13.4.20 Show IEEE 802.1X authentication status

Syntax `show dot1x-auth`

Context Admin Exec context

Usage Show hosts that are currently authenticated with IEEE 802.1X.

Default values Not applicable.

13.4.21 Show MAC based authentication status

Syntax `show mac-auth`

Context Admin Exec context

Usage Show hosts that are currently authenticated with MAC based access control.

Note
There may be hosts on the network that matches the MAC authentication filters, but are inactive for the moment. Inactive hosts are flushed out of this list and will be re-authenticated again on resumed activity. See section 13.2.2 for details.

Default values Not applicable.

Chapter 14

FRNT

The Fast Reconfiguration of Network Topology (FRNT) protocol handles fast reconfiguration in switched 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.

In addition to FRNT, WeOS supports the standard RSTP protocol. Management of RSTP is described in chapter 16.

14.1 Overview of the FRNT protocol and its features

The table below summarises FRNT features available via the Web and CLI interfaces. A general description of the FRNT protocol and its features are presented in sections 14.1.1 and 14.2. If you are only interested in knowing how to manage the FRNT features via the Web or CLI, please visit sections 14.3 or 14.4 directly.

Feature	Web	CLI	General Description
Enable FRNT	X	X	Section 14.1.1
Set FRNT mode (focal-point or member switch)	X	X	-”-
Set FRNT ring ports	X	X	-”-
View FRNT Status	X	X	-”-

14.1.1 FRNT introduction

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 (*port "M"*) in *blocking* mode, see fig. 14.1.

Note
In an FRNT ring, only one of the switches can be configured as focal point. The other switches should be configured as member switches (i.e., non-"focal-point").

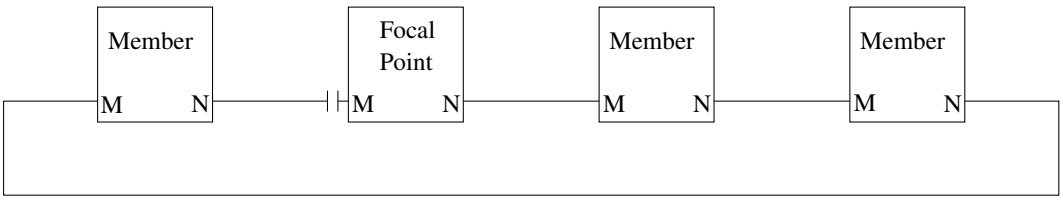


Figure 14.1: FRNT network operating in *ring mode*. Port "M" on the Focal Point is in BLOCKING state.

Once a link failure is detected somewhere along the ring, the focal point will put its blocked port (*port "M"*) in *forwarding* mode to establish full connectivity between the switches (see fig. 14.2). FRNT is *event based*: switches detecting a *link down* event will immediately send a *link down* FRNT message towards the focal point. Intermediate switches will forward the FRNT messages with highest priority, and the focal point will open its BLOCKED port (port "M") upon receiving the *link down* message.

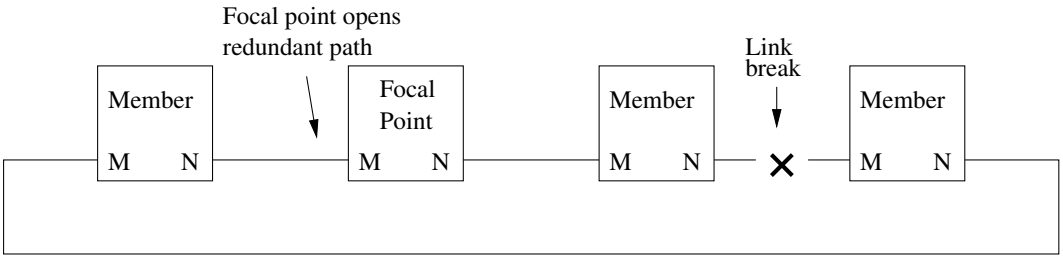


Figure 14.2: FRNT network operating in *bus mode* due to broken link.

Similarly, when a broken link comes back up again and the ring is fully connected,

the focal point will react and put its *port ”M”* back to blocking state.

14.1.2 Guidelines when selecting FRNT ports

When enabling FRNT on a switch, you need to select two ports to use as FRNT ports – FRNT *port ”M”* and FRNT *port ”N”*[1]. Below are some recommendations and rules when selecting and configuring the FRNT ports.

- *Fixed speed, full duplex:* When using Ethernet ports as FRNT ports, fixed speed (and full duplex) is recommended over *auto-negotiation* of speed and duplex mode on the FRNT ports. Avoid using 10 Mbit/s speed.
- *Avoid using copper SFPs as FRNT ports:* When using Ethernet ports as FRNT ports, choose fixed Ethernet ports or fiber SFPs. Copper SFPs may be used as FRNT ports, but will generally imply non-negligible degradation of fail-over performance.
- *SHDSL ports as FRNT ports:* It is possible to use SHDSL ports as FRNT ports, but failover performance is degraded as compared to (fixed) Ethernet ports. FRNT will not work correctly on SHDSL links with speed below 64 kbit/s.

14.1.3 VLANs used by FRNT

FRNT uses VLAN IDs 4020-4022 and 4032-4033 for its signalling. Thus, when FRNT is enabled on a switch, these VLANs are implicitly reserved and cannot be configured by the user.

Warning

Note on using intermediate active equipment For FRNT to operate properly, there should **not** be any ”non-FRNT- enabled” switches (or other active equipment) in the FRNT ring. However, if two FRNT nodes are interconnected via a non-FRNT switch for **testing** purposes, that intermediate switch must be configured to let VLANs 4020-4022 and 4032-4033 through.

[1] In earlier WeOS versions, port ”M” and ”N” have been denoted port ”1” and ”2” respectively.

14.2 FRNT and RSTP coexistence

With WeOS it is possible to run FRNT and RSTP on the same switch, be it with some topology restrictions. Fig. 14.3 shows an example of such a configuration, where two of the switches in the FRNT ring (thick lines) are running RSTP on the ”non-FRNT” ports.

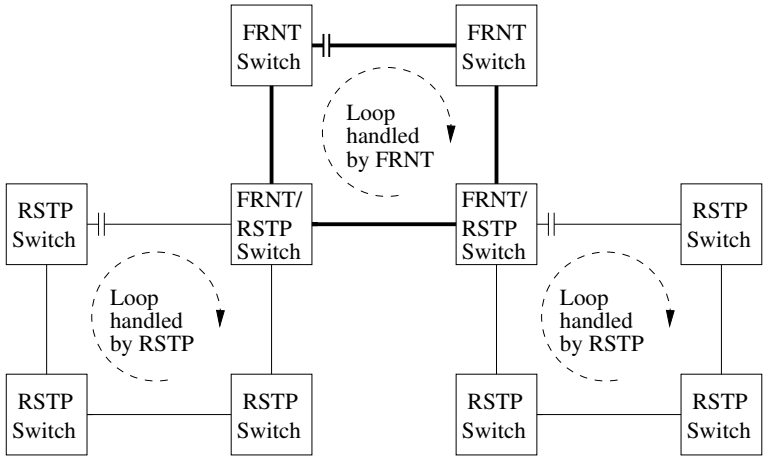


Figure 14.3: Example of coexistence of FRNT and RSTP.

As both RSTP and FRNT want to control a port’s state (FORWARDING/BLOCKING), only one of the protocols may be activated on each port to avoid protocol conflicts. Therefore, if both FRNT and RSTP are configured to operate on a certain port, FRNT will have precedence to control the port’s state.

Warning

FRNT and RSTP are each able to handle loops within their respective domains, however, if a physical loop is created including some links controlled by RSTP and others by FRNT, a broadcast storm is likely to occur, since neither RSTP nor FRNT is able to discover the loop, see fig. 14.4. Thus, if RSTP and FRNT is mixed in the same layer-2 network, the operator must ensure that loops across RSTP and FRNT links never occur.

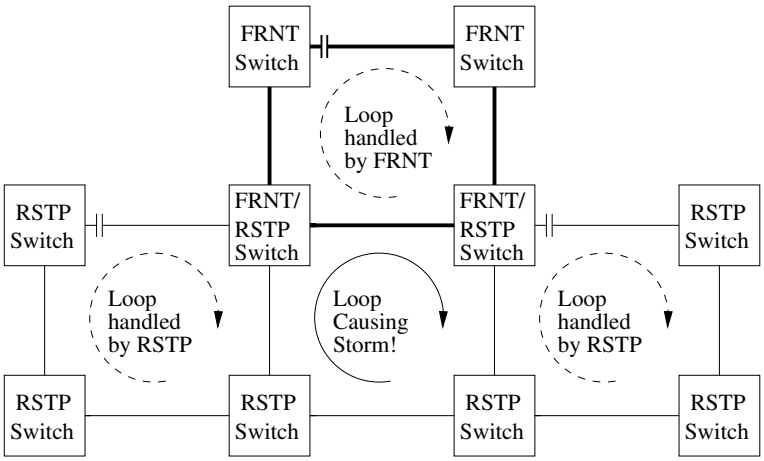


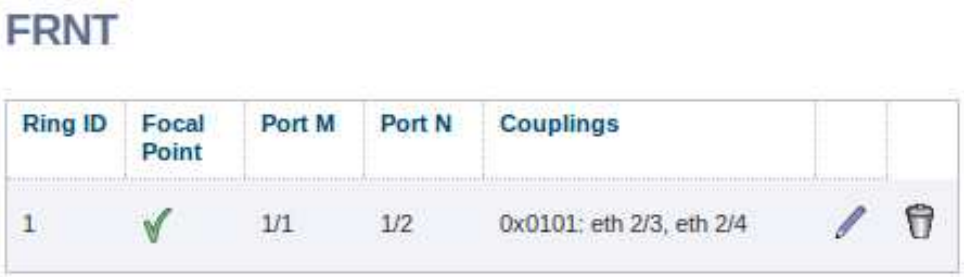
Figure 14.4: Example of loop spanning FRNT and RSTP links - a broadcast storm is likely to occur.

14.3 Managing FRNT settings via the web interface

14.3.1 Managing FRNT settings

Menu path: Configuration ⇒ L2 Redundancy ⇒ FRNT

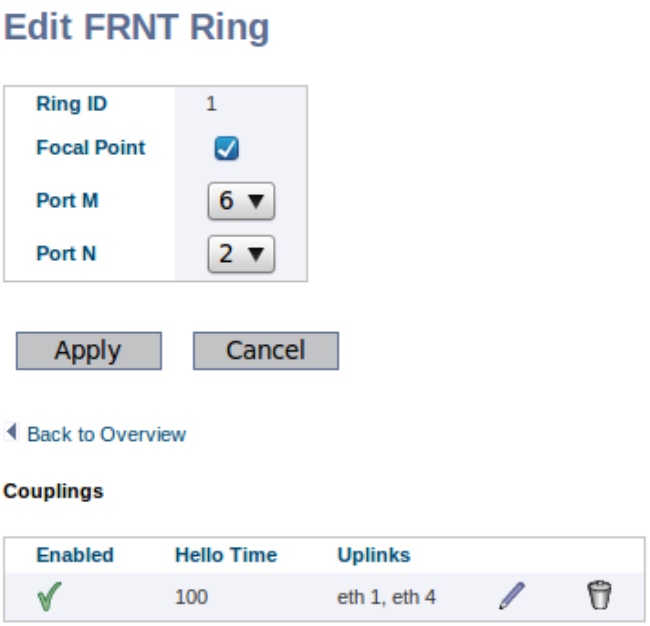
On the FRNT configuration page you will be presented to the current settings for FRNT on your switch, see below.



Ring ID	A unique identifier for the FRNT-ring. Currently only one ring is available.
Focal Point	The focal point is the unit in the ring which is responsible for making decisions on topology change. A green check-mark indicates this unit will take the role as focal point in the FRNT ring. A dash indicates the unit will act as a *member* unit.
Port M/Port N	FRNT requires two ports to be assigned FRNT-ports. These are connected to peer units participating in the FRNT ring. The two ports connected to other units in the FRNT ring. **Note:** Ports with copper SFPs should not be used as FRNT ports, due to slow link down indication on copper SFPs. See section 14.1.2 for further guidelines on FRNT port selection.
Couplings	Lists the currently configured FRNT Ring-Couplings associated with this FRNT-ring, and the coupling uplink ports.
Edit	Click this icon to edit an FRNT instance.
Delete	Click this icon to remove an FRNT instance.

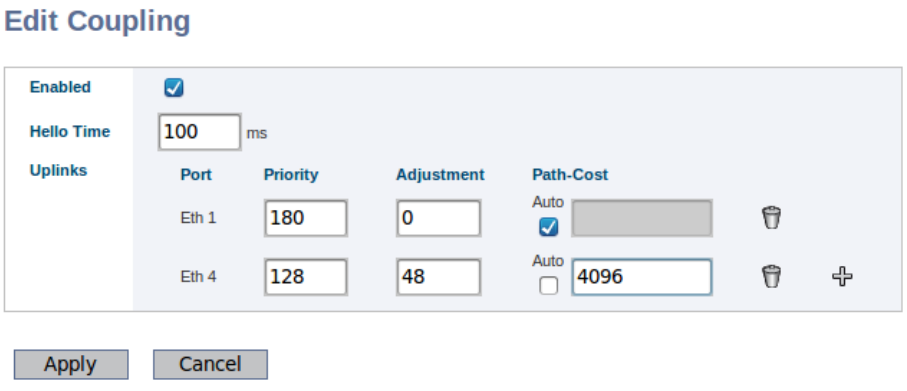
If no FRNT instance is configured you may create one by clicking the **N**ew button.

When editing a new or existing instance the page below is displayed.



The FRNT settings are described in the table above. The lower part contains a section, **Couplings**, where the FRNT Ring-Couplings, associated with this FRNT instance, is listed. This section will appear after clicking the **Apply** when a new FRNT instance is created.

To create a new Coupling instance, click the **New Coupling** button (visible until MAX_RING_COUPLING_INSTANCES (section 15.4) has been reached). New and existing Ring-Couplings are edited on the page below:



Continued on next page

	Continued from previous page
Enabled	A green checkbox if the coupling instance is enabled, a minus sign if not. On edit page, check/uncheck box to enable/disable coupling instance.
Hello Time	The interval between two hello messages in mili-seconds.
Uplinks	
Port	The uplink port.
Priority	The uplinks priority. Used for calculating active uplink.
Adjustment	Priority adjustment delta for this uplink. Makes the uplink sticky by adjusting the effective priority with this value when uplink becomes active.
Path Cost	The uplinks path cost. Used for calculating active uplink. Auto (check-box checked) indicates path-cost is automatically calculated (based on link speed).
 **Delete**	Click this icon to remove a coupling instance.
 **Add**	Click this icon to add a new coupling instance.

14.3.2 FRNT Status and Statistics

Menu path: Status ⇒ L2 Redundancy ⇒ FRNT

On this page FRNT status and statistics are presented.

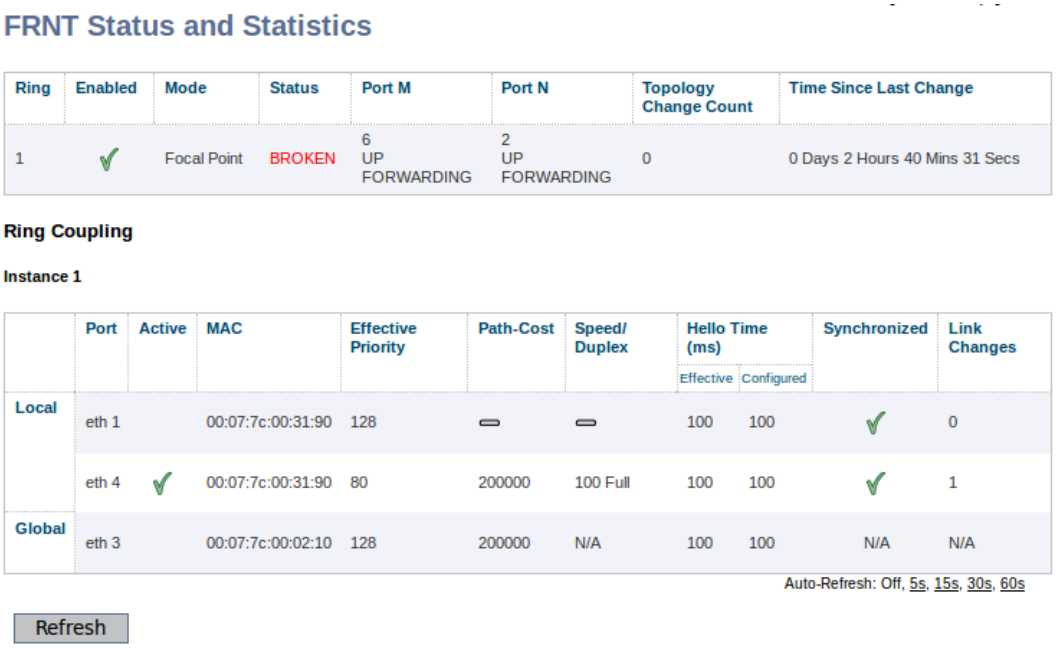


Figure 14.5: FRNT status and statistics in web

FRNT Status and Statistics	
Ring	Instance ID for the FRNT ring.
Enabled	Indication if the ring is enabled or not.
Mode	Focal point or member.
Status	Ring status, OK or BROKEN.
Port M	Status of port operating[1] as FRNT port M.
Port N	Status of port operating[1] as FRNT port N.
Topology Change Count	Number of FRNT topology changes.
Time Since Last Change	Time since last FRNT topology change.

Continued on next page

	Continued from previous page
Ring Coupling	
Local/Global	Local - uplinks located on this switch. Global - uplinks reported from other switches in the FRNT ring.
Port	The uplink port name, if any available on the distributing unit. Otherwise an information message stating that no uplinks are available.
Active	A green check-mark indicates this is the active uplink for the ring coupling instance.
MAC	The MAC address of the unit distributing this piece of uplink information.
Effective Priority	The actual priority value used in uplink selection. When configuring an adjustment delta this may differ from the configured priority for an active uplink. Used to minimize uplink changes when an active uplink goes down and up again.
Path-Cost	The current path-cost. If auto configuration selected, this value is calculated based on port speed.
Speed/Duplex	Speed duplex on the uplink port. Only applicable for local uplinks.
Hello Time	The configured and effective (negotiated) hello-time on each unit.
Synchronized	A green check-box indicates this uplink has been synchronized with its neighbor. Only applicable for local uplinks.
Link Changes	Number of link changes. Only applicable for local uplinks.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.

[1] If the port referred to as FRNT port ”M” and FRNT port ”N” in the FRNT statistics page (operational FRNT ”M” and ”N”) does not match the administratively configured FRNT ”M” and ”N” ports (see the FRNT configuration page in section 14.3.1), the ports are *logically* swapped/aligned with the ”M” and ”N” ports of the focal-point.

14.4 Managing FRNT settings via the CLI

Command	Default	Section
Configure FRNT settings		
[no] frnt [ID]		Section 14.4.1
[no] focal-point	focal-point	Section 14.4.2
ring-ports <PORT-M, PORT-N>		Section 14.4.3
Show FRNT status		
show rings		Section 14.4.4
show ipconfig		Section 7.3.34

14.4.1 Managing FRNT

Syntax `[no] frnt [<ID>]`

Context Global Configuration context

Usage Enter FRNT context of the given FRNT instance ID. Currently only a single FRNT instance is supported, thus the value of the FRNT ID is ignored.

The FRNT instance is only activated upon the selection of valid FRNT ring ports, see section 14.4.3.

Use **”no frnt [ID]”** to remove an existing FRNT instance.

Use **”show frnt”** to list configured FRNT settings (also available as **”show”** command within the FRNT Configuration context).

Default values Default ID is 1

14.4.2 FRNT focal point and member switch

Syntax `[no] focal-point`

Context FRNT Configuration context

Usage Configure device to act as FRNT focal point for this FRNT instance. Use **”focal-point”** to configure the device to act as an FRNT *focal-point*, and **”[no] focal-point”** to configure the device as an FRNT *member* switch.

Use **”show focal-point”** to show whether the unit is configured as focal-point or member switch

Default values focal-point

14.4.3 FRNT Ring Ports

Syntax `ring-ports <PORT-M,PORT-N>`

Context FRNT Configuration context

Usage Set the physical ports (Ethernet ports or SHDSL ports) to use as FRNT ports ”M” and ”N”.

For each FRNT instance, there are two FRNT ports named Port ”M” and Port ”N”. On a member switch Port ”M” and ”N” have similar roles, however, on a focal point their roles differ - when the ring is fully connected the focal point will put its Port ”M” in BLOCKING state.

> **Note**
> For restrictions on how to select FRNT ports, see section 14.1.2.

Use **”show ring-ports”** to show configured FRNT ring ports.

Default values Not applicable

14.4.4 Show FRNT ring status

Syntax `show rings`

Context Admin Exec context.

Usage Show status of configured FRNT rings. This will provide information

- whether the ring is up (ring mode) or if the ring is broken (bus mode). Note: A *focal point* switch will detect ring failures located anywhere in the ring, while a *member* switch can only detect local failures (local FRNT port is down, or if a neighbour is down).
- If the FRNT ports on this switch are connected in-line with the M/N ports of the focal-point, or if they are *logically* swapped (i.e., if the FRNT ports’

administrative M/N state equals the *operational* M/N state, or if ports are swapped).

- The status of the local FRNT ports (UP/DOWN, FORWARDING/BLOCKING).

Default values Not applicable.

Chapter 15

FRNT Ring Coupling and Multi-Link Dual Homing

This chapter describes WeOS *FRNT Ring Coupling* and *Multi-Link Dual Homing*, two similar layer-2 (switching) fail-over functions.

FRNT Ring Coupling enables bridging of two or more FRNT rings via multiple layer-2 *uplinks*. Only one uplink is *active* at a time, while others are hot stand-by *backups*, providing redundancy and loop-free connectivity.

Multi-Link Dual-Homing (or simply ”Dual-Homing”) lets you connect a WeOS switch to a layer-2 topology via multiple *uplinks*. Optimal fail-over performance is achieved when connecting the dual-homing switch uplinks to a single FRNT ring, or to two adjacent FRNT rings (connected with Ring Coupling), but Dual-Homing can also be used in other layer-2 topologies.

Both FRNT Ring Coupling and Multi-Link Dual-Homing provide fine-grained control of which *uplink* is to be preferred as *active*. By default, the link with highest speed/duplex mode is elected. To avoid shifting between active uplinks when a new uplink becomes available, a feature referred to as *sticky uplink* is provided. Enabling *sticky* uplink gives ”zero” fail-over time on link-up and mitigates possible problems with flapping links.

Section 15.1 presents further information on FRNT Ring Coupling and the Multi-Link Dual-Homing functionality. Web and CLI support for these features are covered in sections 15.2 and 15.3 respectively.

15.1 Overview

Feature	Web	CLI	General Description
FRNT Ring Coupling	X	X	Section 15.1.1
Enable	X	X	
Hello Interval	X	X	Section 15.1.1.2
Define Uplink(s)	X	X	Sections 15.1.1 and 15.1.3
Uplink Path-Cost	X	X	Section 15.1.3
Uplink Priority	X	X	-”-
Ring Coupling Status	X	X	
Multi-Link Dual-Homing	X	X	Section 15.1.2
Enable	X	X	
Synchronized		X	Section 15.1.2.1
Multiple Instances		X	Section 15.1.2.2
Define Uplink(s)	X	X	Sections 15.1.2 and 15.1.3
Uplink Path-Cost	X	X	Section 15.1.3
Uplink Priority	X	X	-”-
Dual-Homing Status	X	X	

15.1.1 FRNT Ring Coupling

FRNT Ring Coupling (RiCo) enables redundant bridging between two or more FRNT rings. Fig. 15.1a shows a simple example where two RiCo nodes in an FRNT *sub-ring* are connected with one uplink each to the FRNT *super-ring*. A *super-ring* is a ring *without* RiCo nodes. It is possible to use more than two RiCo nodes in the sub-ring, and each RiCo node can have more than one uplink, as shown in fig. 15.1b.

Only one of the uplinks is forwarding data – the *active uplink*, ”solid” in fig. 15.1, while the other uplink(s) are hot-standby backups, ”dashed” in fig. 15.1. To prevent traffic to flow over backup uplinks the RiCo nodes put all backup uplinks in *BLOCKING* state.

In the CLI example below the leftmost Ring Coupling node in fig. 15.1a is a WeOS unit configured as an FRNT member switch (see chapter 14) with ring-ports ’1’

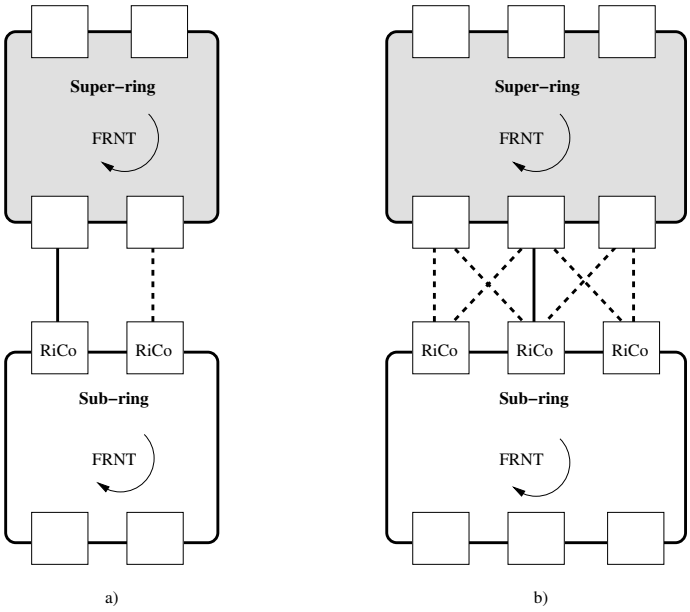


Figure 15.1: Ring Coupling with two FRNT rings: (a) single uplinks, and (b) multiple uplinks per Ring Coupling node.

and ’2’, and port ’3’ as uplink.

Example

```
example:/#> configure
example:/config/#> frnt
Activating FRNT0 with default settings, remember to change the ring ports!
Invalid settings: No ring ports defined
example:/config/frnt-1/#> ring-ports 1,2
example:/config/frnt-1/#> no focal-point
example:/config/frnt-1/#> coupling
Creating new instance 1
example:/config/frnt-1/coupling-1/#> uplink 3
example:/config/frnt-1/coupling-1/uplink-eth3/#> priority 100
example:/config/frnt-1/coupling-1/uplink-eth3/#> leave
Starting Fast Redundant Network Topology v0 daemon ......... [ OK ]
Starting Ring bridging/dual-homing daemon ................... [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

Here, the *uplink priority* was given the value **”100”** to make it the preferred active uplink, the default is 128, for further details see section 15.1.3.

It is of course possible to connect multiple sub-rings to one super-ring. The uplinks from the sub-rings can be connected to *individual* nodes in the super-ring

(see fig. 15.2a) or the nodes in the super-ring can be *shared*, see fig. 15.2b.

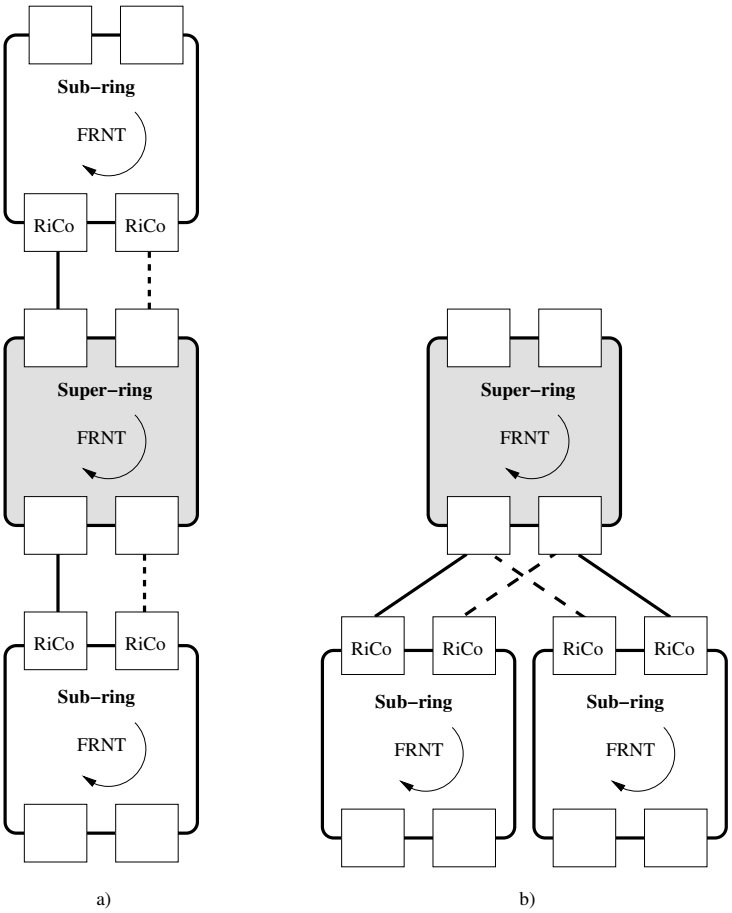


Figure 15.2: Two sub-rings connecting to (a) individual nodes in the super-ring, or (b) shared nodes in the super-ring.

The topology can be extended even further by connecting sub-rings to sub-rings in a tree structure with a super-ring as *root*. Fig. 15.3 shows two examples, a ladder topology (a) and a tree topology (b).

15.1.1.1 Ring Coupling and Routing

FRNT Ring Coupling is a function to connect FRNT rings at layer-2 (switching). This improves capacity and failover performance compared to layer-3 (routing) mechanisms such as OSPF (chapter 27). Nevertheless, routing techniques have

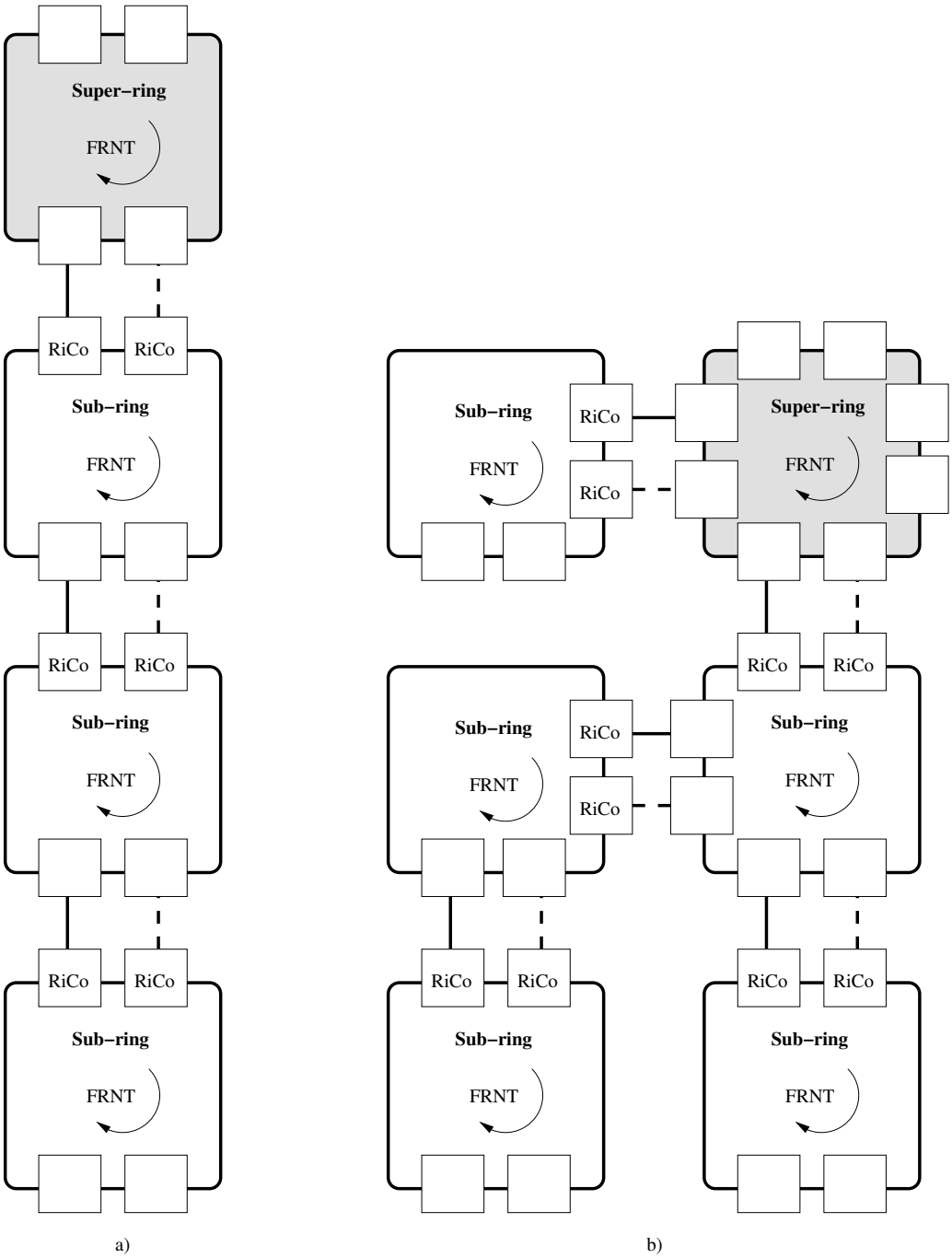


Figure 15.3: Examples of tree topologies: (a) shows a ”ladder” (a tree without branches), and (b) a more generic tree of FRNT rings.

good scalability characteristics as the network is segmented into different broadcast domains.

Although technically feasible, it is *strongly recommend* to separate ring coupling and routing, localizing the distinct functions in dedicated WeOS units, i.e., do not use RiCo nodes also as routers.

> **Note**
> In some cases using RiCo nodes as routers makes sense. To ensure correct operation of the RiCo node, the CPU bandwidth is reduced by default, i.e., when **”cpu-bandwidth-limit”** is set to **”auto”** (section 20.2.6) on a WeOS unit configured for FRNT Ring Coupling or Multi-link Dual-Homing. This in turn reduces routing performance.
> This automatic reduction of CPU bandwidth can be overridden by changing the CPU bandwidth limit setting (section 20.2.6).

15.1.1.2 Ring Coupling Hello Interval

RiCo nodes in the same FRNT ring exchange *Hello* messages as part of the active uplink election process, see also section 15.1.3. These Hello messages are transmitted every 100 ms by default. The *hello interval* can be fine tuned – a lower value gives faster failover, but may have an adverse effect on the CPU usage. When the CPU usage increases RiCo nodes may not be able to send *Hello* messages and will time out. This can lead to unpredictable performance and loss of connectivity.

It is recommended that all RiCo nodes within an FRNT ring are configured with the same Hello interval. If there are RiCo nodes with different Hello interval in an FRNT ring, the protocol will default to the highest interval announced by any RiCo node, i.e., a RiCo node’s *effective hello interval* may differ from its *configured hello interval*. E.g., if you wish to transition from using **”hello-time 100”** to **”hello-time 80”**, all RiCo nodes will use interval 100 ms until all RiCo node’s in the FRNT ring has been configured with interval 80 ms.

15.1.2 Multi-Link Dual-Homing

Multi-Link Dual-Homing makes is possible for a WeOS dual-homing node to have redundant connections to an FRNT ring. Fig. 15.4 shows an example where two dual-homing nodes are connected to the FRNT ring with two connections each.

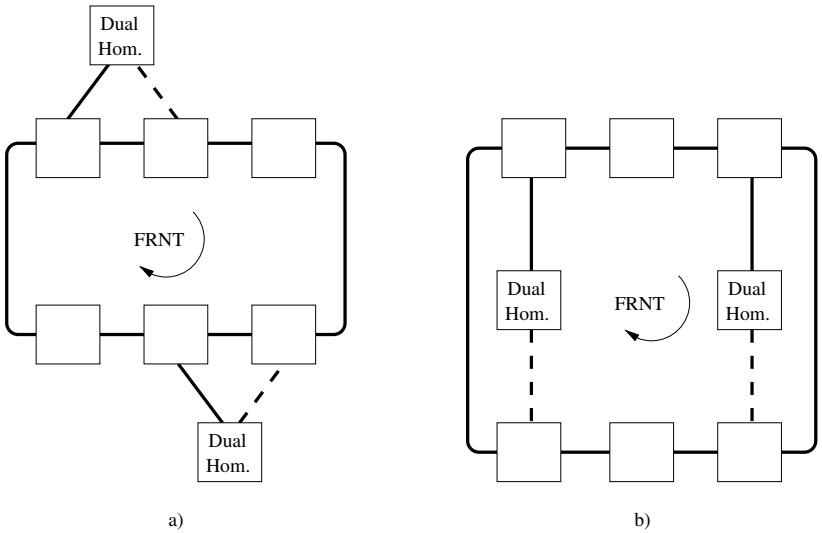


Figure 15.4: Dual-Homing with single FRNT ring (super-ring). (a) and (b) show the same topology in different ways.

Consider one of the dual-homing nodes in fig. 15.4a, assuming it is a WeOS switch with Ethernet ports ’1’ and ’2’ as uplinks, and where port ’1’ is to be active by default. A possible configuration is given below:

Example

```
example:/#> configure
example:/config/#> dual-homing
Creating new instance 1
example:/config/dual-homing-1/#> uplink 1
example:/config/dual-homing-1/uplink-eth1/#> priority 100
example:/config/dual-homing-1/uplink-eth1/#> end
example:/config/dual-homing-1/#> uplink 2
example:/config/dual-homing-1/uplink-eth2/#> leave
Starting Ring bridging/dual-homing daemon ................. [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

Note
It is possible to connect uplinks of a WeOS dual-homing switch to any layer-2 topology, but failover performance is optimised for FRNT rings.
When connecting uplinks to LAN topologies other than FRNT the *synchronised* option in dual-homing must be disabled, see section 15.1.2.1, otherwise the uplinks will not come up.

Section 15.1.2.1 describes the *synchronised dual-homing* function, illustrates how you can combine Multi-Link Dual-Homing and FRNT Ring Coupling, supporting topologies where you can connect the dual-homing uplinks to two adjacent FRNT rings.

Section 15.1.2.2 presents the possibility of using multiple instances of dual-homing.

15.1.2.1 Synchronized Dual-Homing

WeOS Multi-Link Dual-Homing provides a mechanism referred to as *synchronised dual-homing*. Synchronised dual-homing has two purposes:

- *Integrity of the uplink:* A dual-homing switch monitors uplink connectivity by exchanging specific *echo packets* with the remote switch. This ensures that a link break is detected even in cases where intermediate transceivers do not propagate link down.

- *Define preferred uplink when connecting two adjacent FRNT rings:* It is possible to connect the uplinks of a dual-homing node to two adjacent FRNT rings, which in turn are connected by ring coupling. The synchronised dual-homing feature will give preference to the uplink connected to a ring coupling *sub-ring*, i.e., the ring containing the RiCo nodes. More details later in this section.

If you wish to connect a dual-homing switch to topologies other than FRNT you need to disable the synchronised dual-homing feature in the dual-homing node. An example is given below where ports 1 and 2 are configured as uplinks to non-FRNT nodes.

Example

```
example:/#> configure
example:/config/#> dual-homing
Creating new instance 1
example:/config/dual-homing-1/#> uplink 1
example:/config/dual-homing-1/uplink-eth1/#> priority 100
example:/config/dual-homing-1/uplink-eth1/#> end
example:/config/dual-homing-1/#> uplink 2
example:/config/dual-homing-1/uplink-eth2/#> end
example:/config/dual-homing-1/#> no synchronized
example:/config/dual-homing-1/#> leave
Starting Ring Coupling/Dual-Homing daemon .................. [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

Additional remarks if you intend to use dual-homing to connect the uplinks to other than FRNT nodes.

- *Fail-over performance:* Fail-over performance is optimised when connecting the dual-homing node to a single FRNT ring, or to two FRNT rings. (Which in turn may be connected by FRNT ring coupling). In other topologies the switches may temporarily have stale MAC entries in their learning caches for a short period of time (unicast traffic). Furthermore, if IGMP snooping is used multicast traffic will also be disrupted until switches receive new IGMP Reports via the new uplink.
- *Integrity of the uplink:* With synchronised dual-homing disabled, the uplink status is determined based on its physical status (up/down). If you wish additional control in this case, you could consider running LACP on the uplink, i.e., you could create a link-aggregate with the uplink as the only member link. See chapter 17 for further information on LACP and link aggregation.

It is possible to combine the use of Multi-Link Dual-Homing and FRNT Ring Coupling. Fig. 15.5 shows how a dual-homing node can be connected to two adjacent FRNT rings, and fig. 15.6 illustrates an example with several dual homing nodes.

The *synchronised dual-homing* feature will give preference to the uplink leading to the FRNT ring containing RiCo nodes (the ring coupling ”sub-ring”). This means that the ’left’ dual-homing uplink (port ’1’) in fig. 15.5 will be active as long as a RiCo node in that ring is reachable, and in turn has an active uplink.

To ensure that the dual-homing node fail-over to the other uplink (the ’right’ uplink (port ’2’) in fig. 15.5) if no RiCo node is reachable via the sub-ring, port ’2’ should be configured with better uplink *priority*, see also section 15.1.3. A configuration example is given below.

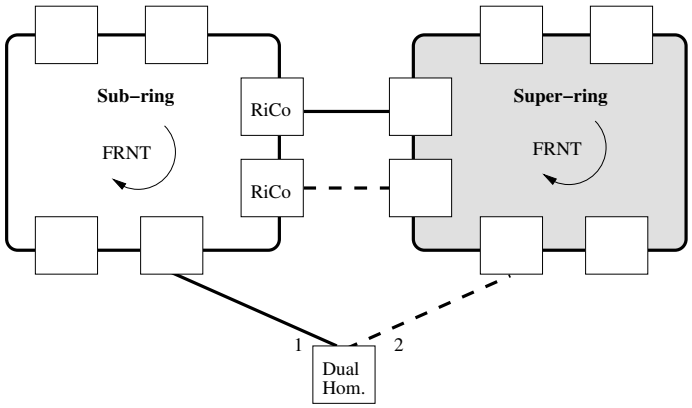


Figure 15.5: Dual-Homing used in an FRNT Ring Coupling Topology.

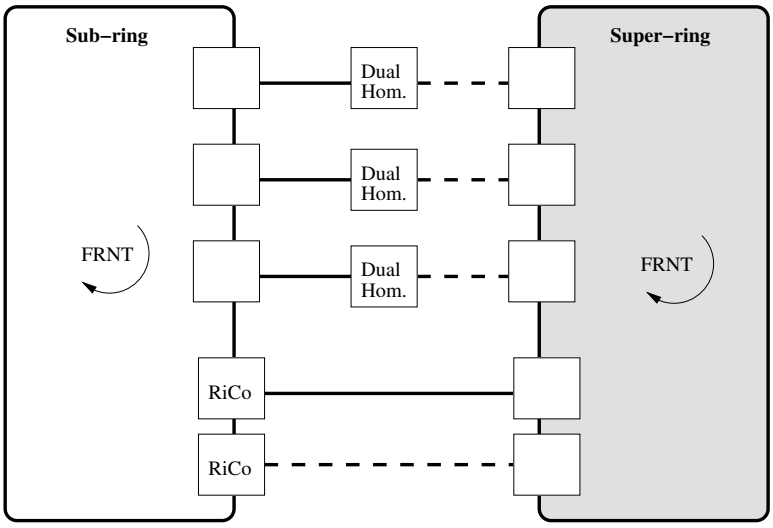


Figure 15.6: Multiple Dual-Homing nodes in an FRNT Ring Coupling Topology.

Example

```
example:/#> configure
example:/config/#> dual-homing
Creating new instance 1
example:/config/dual-homing-1/#> uplink 1
example:/config/dual-homing-1/uplink-eth1/#> end
example:/config/dual-homing-1/#> uplink 2
example:/config/dual-homing-1/uplink-eth2/#> priority 100
example:/config/dual-homing-1/uplink-eth2/#> leave
Starting Ring Coupling/Dual-Homing daemon .................. [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#>
```

15.1.2.2 Multiple instances of Dual-Homing

It is possible to create multiple dual-homing instances on a WeOS switch. Each instance has its own set of uplinks, referred to as an *uplink domain* – one of the uplinks in the domain will be active, while the others are backups. A sample topology is shown in fig. 15.7.

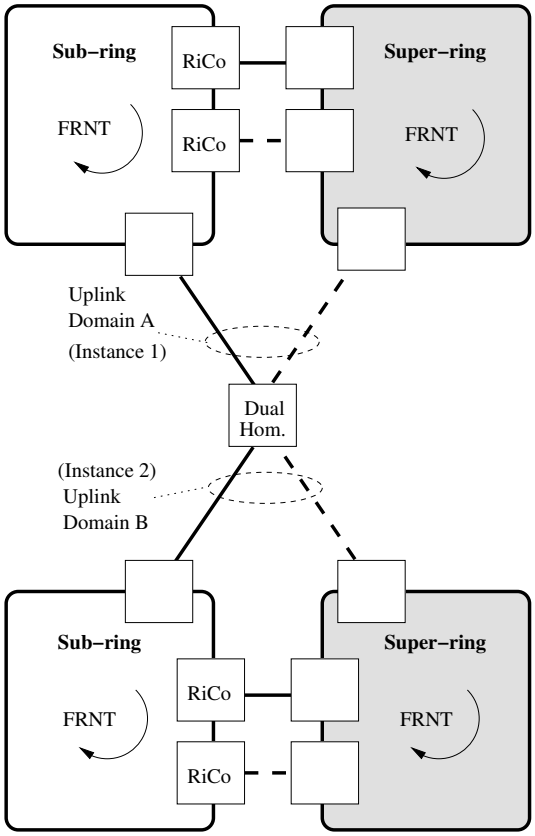


Figure 15.7: Possibility to setup multiple uplink domains for dual-homing.

Warning

The upper and lower LANs in fig. 15.7 **must not** have additional interconnections. Otherwise a layer-2 loop would be created via the dual-homing node, unless the dual-homing node has VLAN barriers between uplinks of the different instances.

15.1.3 Active uplink election

Both FRNT Ring Coupling and Multi-Link Dual Homing makes use of an uplink election mechanism to determine which of the available uplink that should become *active* and which should be *backups*. For Dual-Homing, the election is handled within the dual-homing node itself, but for FRNT ring coupling the election process the RiCo nodes in the FRNT ring negotiate which uplink should be active.

To determine which uplink is preferred, an *cost vector* is formed for every uplink and compared. The uplink with the *lowest cost vector* is elected as active uplink[1]. The cost vector consists of the following fields.

- *Link speed/duplex Cost:* The most significant component of the cost vector depends on the link's speed and duplex setting. This link speed/duplex component is calculated as shown in table 15.2, similar to link cost calculation in RSTP (see section 16.1.3). It is also possible to configure the link speed/duplex cost manually. Default: **Auto** (see table 15.2)
- *Priority:* The next component of the cost vector is the uplink *priority*, which is used when two or more uplinks have the same link speed/duplex cost. Then the uplink with the *lowest priority value* is elected as active uplink. Default: **128**
- *Base MAC address:* (Only for Ring Coupling) If *link speed/duplex cost* and *uplink priority* are equal for two RiCo nodes, the node with the lowest base-MAC address will win.
- *Link/port identity:* Finally, if all other components are equal, the port with the lowest port number is elected as *active*.

Bandwidth	Full Duplex	Half Duplex	Two Aggregated Links
10 Mbps	2,000,000	4,000,000	1,000,000
100 Mbps	200,000	400,000	100,000
1 Gbps	20,000	40,000	10,000

Table 15.2: Link speed & duplex to link cost component translation table. For aggregated links (see chapter 17) the link speed/duplex cost is half the cost of a single link for the given link speed and duplex mode. This is shown in the right-most column.

[1] The exception is dual-homing with synchronised dual-homing enabled. Then uplinks to FRNT rings with reachable Ring Coupling nodes have precedence over other uplinks, see also section 15.1.2.1.

To mitigate issues with flapping uplinks, e.g., caused by bad cables, dual-homing nodes and ring coupling nodes can be configured to use a *sticky uplink*, as opposed to the *deterministic uplink* election described above. With sticky uplink enabled, the *priority* component of an uplink's cost vector is reduced with a given value (the *adjustment value*) once that link is elected as *active*. That is, with sticky uplink configured, the *effective priority* of an uplink can differ from the *configured priority*.

Example

Consider three uplinks with same speed and duplex. *Link A* has **"priority 100"**, *link B* has **"priority 110"** with **"adjustment 20"**, and *link C* has **"priority 120"** with **"adjustment 40"**. All nodes keep information about each-others announced link cost (*100*, *110* and *120*). If *Link A* goes down, *link B* will take over as it has lower (i.e., better) priority than *link C* (110<120), and *link B* will decrease its effective priority to *90* in its announcements.
If *link A* comes up again, *link B* will continue to be active as "90<100". The mechanism works in the same way for dual-homing, even though priority is never "announced" to any other node.

15.1.4 Handling Multicast

To provide fast fail-over of multicast traffic, FRNT Ring Coupling and Multi-Link Dual-Homing uplinks are added to the list of *multicast router ports*, see section 18.1.1. This is both done at the Ring Coupling nodes and Dual-Homing nodes, as well as on switches on the remote side of the uplink[2]. This means that all layer-2 multicast traffic is always sent over the uplinks, even if IGMP snooping is enabled.

[2] An exception is when connecting a Dual-Homing uplink to a non-FRNT switch, the fail-over of multicast traffic will instead occur on the next reception of an IGMP Report (if IGMP snooping is enabled). See also section 15.1.2.1.

15.2 Managing via the Web

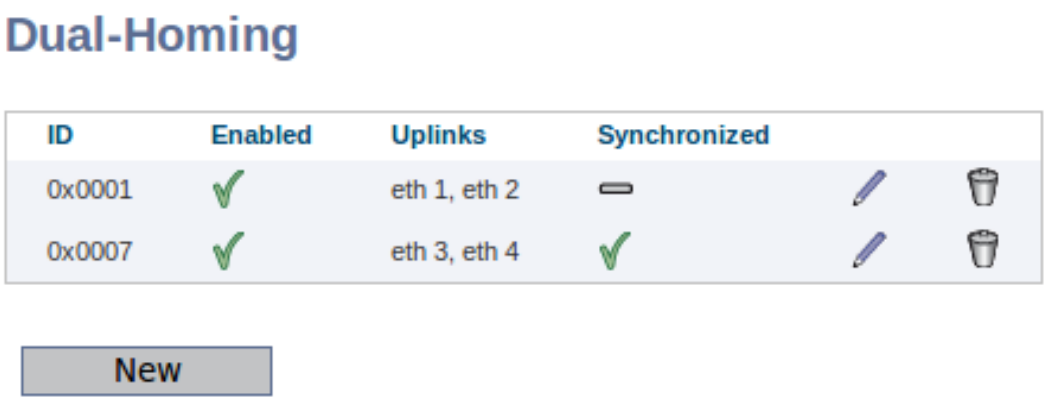
15.2.1 Managing FRNT Ring Coupling Settings

FRNT ring couplings are set up in the FRNT context, see section 14.3.1 for further information.

15.2.2 Managing Dual-Homing Settings

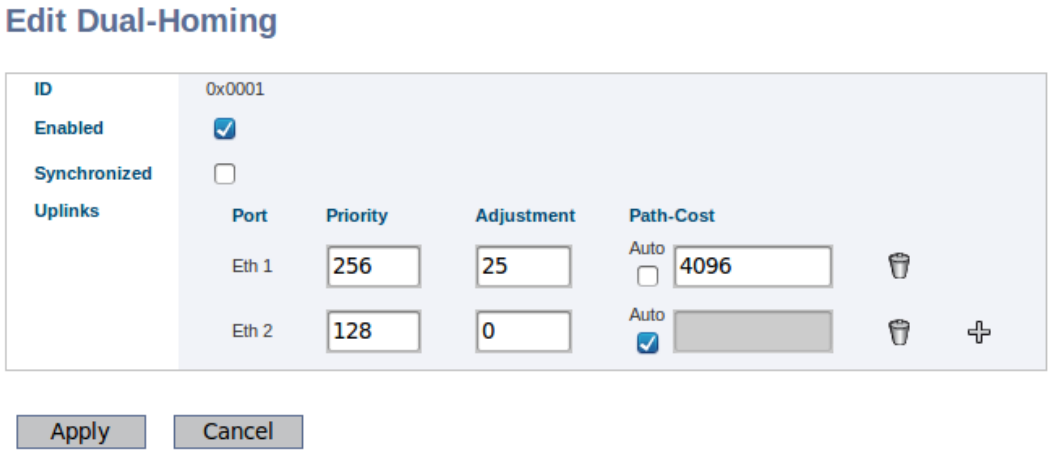
Menu path: Configuration ⇒ L2 Redundancy ⇒ Dual-Homing

Here the list of currently configured Dual-Homing instances is found.



ID	A unique identifier for the dual-homing instance.
Enabled	A green checkbox if the dual-homing instance is enabled, a minus sign if not.
Uplinks	A list of the uplinks configured for this dual-homing instance.
Synchronized	A green check-box indicates this uplink has been synchronized with its neighbor.
Edit	Click this icon to edit a dual-homing instance.
Delete	Click this icon to remove a dual-homing instance.

Use the **New** button to create a new Dual-Homing instance.
Up to MAX_DUAL_HOMING_INSTANCES (section 15.4) can be created.



Enabled	Check/uncheck box to enable/disable dual-homing instance.
Synchronized	Check/uncheck box to enable/disable *Synchronized* mode which requires synchronization with its neighbour.
Uplinks	
Port	The uplink port.
Priority	The uplinks priority. Used for calculating active uplink.
Adjustment	Priority adjustment delta for this uplink. Makes the uplink sticky by adjusting the effective priority with this value when uplink becomes active.
Path Cost	The uplinks path cost. Used for calculating active uplink. Auto (check-box checked) indicates path-cost is automatically calculated (based on link speed).
Delete	Click this icon to remove a dual-homing instance.
Add	Click this icon to add a new dual-homing instance.

15.2.3 Dual-Homing Status and Statistics

Menu path: Status ⇒ L2 Redundancy ⇒ Dual-Homing

On this page dual-homing status and statistics is presented.

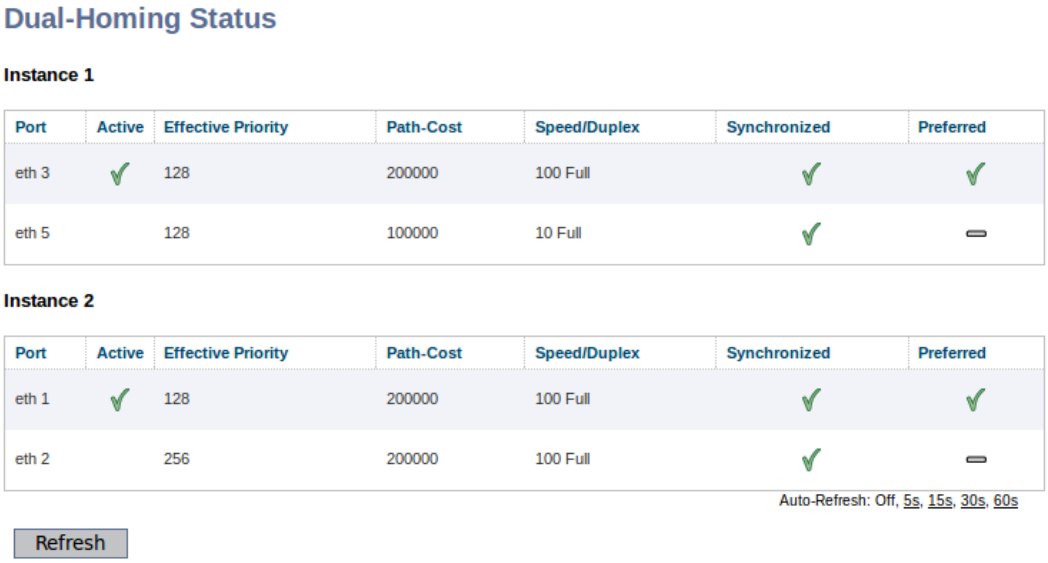


Figure 15.8: Dual-homing status and statistics in web

Port	The uplink port name.
Active	A green check-mark indicates this is the active uplink for the dual-homing instance.
Effective Priority	The actual priority value used in uplink selection. When configuring an adjustment delta this may differ from the configured priority for an active uplink. Used to minimize uplink changes when an active uplink goes down and up again.
Path-Cost	The current path-cost. If auto configuration selected, this value is calculated based on port speed.
Speed/Duplex	Speed duplex on the uplink port.
Synchronized	A green check-box indicates this uplink has been synchronized with its neighbor. Only applicable for local uplinks.
Preferred	A green check-box indicates this uplink is preferred.

Continued on next page

Continued from previous page

Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.

15.3 Managing via CLI

Command	Default	Section
Configure Ring Coupling Settings		
frnt		Sec. 14.4.1
[no] coupling [ID]	1	Sec. 15.3.1
[no] enable	Enabled	Sec. 15.3.2
[no] hello-interval <50..10000>	100 (msec)	Sec. 15.3.3
[no] uplink <PORT>		Sec. 15.3.4
[no] path-cost <auto|<COST>>	Auto	Sec. 15.3.5
[no] priority <1..65535> [adjust <DELTA>]	128	Sec. 15.3.6
Configure Multi-Link Dual-Homing Settings		
[no] dual-homing [ID]	1	Sec. 15.3.7
[no] enable	Enabled	Sec. 15.3.8
[no] synchronized	Enabled	Sec. 15.3.9
[no] uplink <PORT>		Sec. 15.3.10
[no] path-cost <auto|<COST>>	Auto	Sec. 15.3.11
[no] priority <1..65535> [adjust <DELTA>]	128	Sec. 15.3.12
FRNT Ring Coupling and Multi-Link Dual-Homing Status		
show coupling		Sec. 15.3.13
show dual-homing [ID]		Sec. 15.3.14

15.3.1 Managing FRNT Ring Coupling

Syntax `[no] coupling [ID]`

Context FRNT Configuration context

Usage Use **”coupling ID”** to enter FRNT Ring Coupling Configuration context of the given Ring Coupling instance ID. Currently only a single Ring Coupling instance is supported, thus the value of the coupling ID is ignored. **”coupling ID”** creates an FRNT Ring Coupling instance unless it already exists.

Use **”no coupling”** to remove the ring coupling instance.

Use **”show coupling”** to show configuration information for the ring coupling instance. (Also available as **”show”** command within the FRNT Ring Coupling Configuration context.)

Default values Default ID is 1

15.3.2 Enable/Disable FRNT Ring Coupling

Syntax `[no] enable`

Context FRNT Ring Coupling Configuration context

Usage Enable or disable an FRNT Ring Coupling instance. Use **”enable”** to enable the coupling instance, and **”no enable”** to disable the coupling instance (without losing configuration settings for this instance).

Use **”show enable”** to show whether the coupling instance is enabled or disabled.

Default values Enabled

15.3.3 Set FRNT Ring Coupling Hello Interval

Syntax `[no] hello-interval <50..10000>`

Context FRNT Ring Coupling Configuration context

Usage Use **”hello-interval VALUE”** to set the hello interval (in milliseconds) to be announced by his ring coupling node.

> **Note**
> The *effective hello-interval* used will be the highest interval announced by any ring coupling node in the FRNT ring.

”no hello-interval” resets the configured hello interval to the default setting (100 milliseconds).

Use **”show hello-interval”** to show the configured hello interval.

Default values 100 (msec)

15.3.4 Managing FRNT Ring Coupling Uplink Ports

Syntax `[no] uplink [PORT]`

Context FRNT Ring Coupling Configuration context

Usage Use **"uplink PORT"** to define the given port as uplink for this ring coupling node, and enter the *Ring Coupling Uplink Configuration* context for the port. A port can be an Ethernet port (chapter 8), a DSL port (chapter 10 and chapter 11), or a link aggregate (chapter 17).

Up to MAX_RING_COUPLING_UPLINKS (section 15.4) can be created.

Use **"no uplink PORT"** to remove the give port as uplink for this ring coupling node, or use **"no uplink"** to remove all uplinks for the node.

Use **"show uplink"** to list configuration information for all uplinks, and **"show uplink PORT"** to list uplink configuration settings for the given port (also available as **"show"** command within the *Ring Coupling Uplink Configuration* context for the port.)

Default values Not applicable

15.3.5 Set Ring Coupling Uplink Path-Cost

Syntax `[no] path-cost <auto|COST>`

Context Ring Coupling Uplink Configuration context

Usage Configure uplink path-cost. By default, the path-cost depends on the link speed and duplex mode (higher speed gives lower cost). It is also possible to set a cost manually in range 1..2^{32}-1 (1..4294967295).

The *path-cost* is used when electing the active uplink – the link with the lowest cost will be the active uplink. If the costs of two uplinks are equal, their *uplink priority* (section 15.3.6) is considered. For more details, see section 15.1.3.

Use **"path-cost auto"** to have the uplink's path-cost depend on its link speed and duplex mode. Use **"path-cost COST"** to set a static path-cost for the uplink. **"no path-cost"** will reset the path cost to the default setting (auto).

"show path-cost" will show the configured uplink path-cost.

Default values Auto

15.3.6 Set Ring Coupling Uplink Priority

Syntax `[no] priority <1..65535> [adjust <DELTA>]`

Context Ring Coupling Uplink Configuration context

Usage Configure uplink priority, and optionally enable *sticky* uplink election by setting adjust value.

- Use **”priority VALUE”** to set priority value. A lower value increases the chance for this uplink to be elected as active uplink (lower is better). With equal path-cost (section 15.3.5), an uplink with **”priority 100”** is preferred as uplink over an uplink with **”priority 110”**.
- Use the optional **”adjust DELTA”** setting to *improve* its priority (i.e., *lower* its priority with the specified **”DELTA”**) once the uplink is elected as active uplink. This gives a *sticky* uplink behaviour where shifting active uplink will be less common.

  Consider the following example with *uplink A* (**”priority 100”**), *uplink B* (**”priority 110 adjust 20”**), and *uplink C* (**”priority 120 adjust 40”**), and where *uplink A* came up first and is the active uplink. If *uplink A* goes down, *uplink B* takes over as it has lower priority than *uplink C* (110 < 120). *Uplink B* will then apply its adjustment and announce priority to 90 (110 − 20) in its *hello* messages. Uplink B will stay as active uplink even if uplink A comes up again (90 < 100).

”show priority” will show the configured uplink priority.

Default values `priority` 128 (no adjustment)

15.3.7 Managing Multi-Link Dual-Homing

Syntax `[no] dual-homing [ID]`

Context Global Configuration context

Usage Use **”dual-homing ID”** to enter *Dual-Homing Configuration* context of the given Dual-Homing instance ID. Default instance ID is ”1”, thus command **”dual-homing”** will enter the context of dual-homing instance 1.

”dual-homing ID” creates a dual-homing instance with given ID, unless it already exists.

Up to MAX_DUAL_HOMING_INSTANCES (section 15.4) can be created.

Use **”no dual-homing ID”** to remove a specific dual-homing instance, or **”no dual-homing”** to remove all dual-homing instances.

Use **”show dual-homing”** to list configuration information on all dual-homing instances, and **”show dual-homing ID”** for configuration information on a specific dual-homing instance (also available as **”show”** command within the Dual-Homing Configuration context).

Default values Default ID is 1

15.3.8 Enable/Disable Multi-Link Dual-Homing

Syntax `[no] enable`

Context Dual-Homing Configuration context

Usage Enable or disable a dual-homing instance. Use **”enable”** to enable the dual-homing instance, and **”no enable”** to disable the dual-homing instance (without losing configuration settings for this instance).

Use **”show enable”** to show whether the dual-homing instance is enabled or disabled.

Default values Enabled

15.3.9 Synchronized Multi-Link Dual-Homing

Syntax `[no] synchronized`

Context Dual-Homing Configuration context

Usage Enable or disable the dual-homing *synchronization* feature. When enabled, preference when selecting active uplink will be given to uplinks where the uplink peer announces that it has connectivity to ring-coupling node with active uplink. See section 15.1.2.1 for more information.

Use **”synchronized”** to enable and **”no synchronized”** to disable synchronized dual-homing.

Use **"show synchronized"** to show whether synchronized dual-homing is enabled or disabled.

Default values Enabled

15.3.10 Managing Multi-Link Dual-Homing Uplink Ports

Syntax `[no] uplink [PORT]`

Context Dual-Homing Configuration context

Usage Use **"uplink PORT"** to define the given port as uplink for this dual-homing node, and enter the *Dual-Homing Uplink Configuration* context for the port. A port can be an Ethernet port (chapter 8), a DSL port (chapter 10 and chapter 11), or a link aggregate (chapter 17).

Up to MAX_DUAL_HOMING_UPLINKS (section 15.4) can be created.

Use **"no uplink PORT"** to remove the give port as uplink for this dual-homing node, or use **"no uplink"** to remove all uplinks for the node.

Use **"show uplink"** to list configuration information for all uplinks, and **"show uplink PORT"** to list uplink configuration settings for the given port (also available as **"show"** command within the *Dual-Homing Uplink Configuration* context for the port.)

Default values Not applicable

15.3.11 Set Multi-Link Dual-Homing Uplink Path-Cost

Syntax `[no] path-cost <auto|COST>`

Context Dual-Homing Uplink Configuration context

Usage Configure uplink path-cost. By default, the path-cost depends on the link speed and duplex mode (higher speed gives lower cost). It is also possible to set a cost manually in range $1..2^{32}$-1 (1..4294967295).

The *path-cost* is used when electing the active uplink – the link with the lowest cost will be the active uplink. If the costs of two uplinks are equal, their *uplink priority* (section 15.3.12) is considered. For more details, see section 15.1.3.

Use **”path-cost auto”** to have the uplink’s path-cost depend on its link speed and duplex mode. Use **”path-cost COST”** to set a static path-cost for the uplink. **”no path-cost”** will reset the path cost to the default setting (auto).

”show path-cost” will show the configured uplink path-cost.

Default values Auto

15.3.12 Set Multi-Link Dual-Homing Uplink Priority

Syntax `[no] priority <1..65535> [adjust <DELTA>]`

Context Dual-Homing Uplink Configuration context

Usage Configure uplink priority, and optionally enable *sticky* uplink election by setting adjust value.

- Use **”priority VALUE”** to set priority value. A lower value increases the chance for this uplink to be elected as active uplink (lower is better). With equal path-cost (section 15.3.11), an uplink with **”priority 100”** is preferred as uplink over an uplink with **”priority 110”**.
- Use the optional **”adjust DELTA”** setting to *improve* its priority (i.e., *lower* its priority with the specified **”DELTA”**) once the uplink is elected as active uplink. This gives a *sticky* uplink behaviour where shifting active uplink will be less common.

  Consider the following example with *uplink A* (**”priority 100”**), *uplink B* (**”priority 110 adjust 20”**), and *uplink C* (**”priority 120 adjust 40”**), and where *uplink A* came up first and is the active uplink. If *uplink A* goes down, *uplink B* takes over as it has lower priority than *uplink C* (110 < 120). *Uplink B* will then apply its adjustment and announce priority to 90 (110 − 20) in its *hello* messages. Uplink B will stay as active uplink even if uplink A comes up again (90 < 100).

”show priority” will show the configured uplink priority.

Default values `priority` 128 (no adjustment)

15.3.13 Show FRNT Ring Coupling Status

Syntax `show coupling`

Context Admin Exec context

Usage Use **"show coupling"** to show status of FRNT Ring Coupling.

Default values Not applicable

Example

```
example:/#> show coupling
================================================================== ID 0x0101
Local uplink(s) ----------------------------------------------------------
  ID  MAC               Uplink    Prio/Delta  Cost       Speed       Hello
-------------------------------------------------------------------------------
>> 7  00:07:7c:84:90:44  eth 7      128/0    200000     100-Full   100(100)ms

Global uplink(s) ---------------------------------------------------------
      MAC               Uplink    Prio        Cost                   Hello
-------------------------------------------------------------------------------
      00:07:7c:87:85:62  eth 7      128      200000                100(100)ms

example:/#>
```

The active uplink is marked with **>>**. In this case, lowest MAC address was used as tie-breaker to elect active uplink.

15.3.14 Show Multi-Link Dual-Homing Status

Syntax `show dual-homing`

Context Admin Exec context

Usage Use **"show dual-homing"** to show status of Multi-Link Dual-Homing instances.

Default values Not applicable

Example

```
example:/#> show dual-homing
================================================================================
Instance ID: 0x0001
  Synchronized mode    : Enabled
Local uplink(s) ------------------------------------------------------------
  ID  MAC               Uplink    Prio/Delta  Cost      Speed    Sync   Pref
--------------------------------------------------------------------------------
   4  00:07:7c:10:df:00  eth 4      100/0      -------  --------  No     No
>> 3  00:07:7c:10:df:00  eth 3      128/0      200000   100-Full  No     No

================================================================================
Instance ID: 0x0002
  Synchronized mode    : Enabled
Local uplink(s) ------------------------------------------------------------
  ID  MAC               Uplink    Prio/Delta  Cost      Speed    Sync   Pref
--------------------------------------------------------------------------------
   6  00:07:7c:10:df:00  eth 6      128/0      -------  --------  No     No
>> 5  00:07:7c:10:df:00  eth 5      128/0      200000   100-Full  No     No

example:/#>
```

The active uplink is marked with **>>**. In this case, only one uplink was up in each of the dual-homing instances.

15.4 Feature Parameters

MAX_RING_COUPLING_INSTANCES	1
MAX_RING_COUPLING_UPLINKS	4
MAX_DUAL_HOMING_INSTANCES	8
MAX_DUAL_HOMING_UPLINKS	4

Chapter 16

Spanning Tree Protocol - RSTP and STP

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. WeOS supports RSTP with fall-back to STP when connecting the switch to another device only capable of STP.

STP/RSTP does not provide the same convergence performance as FRNT, however, STP/RSTP can handle arbitrary switched topologies, while FRNT operates in a *ring* structure. For information on FRNT, and coexistence between FRNT and RSTP, see chapter 14 .

RSTP is disabled at factory default.

16.1 Overview of RSTP/STP features

Table 16.1 provides a summary of available RSTP/STP features in WeOS. Further descriptions of the spanning tree protocol and the available features are provided in sections 16.1.1-16.1.3.

16.1.1 Spanning Tree Introduction

Loops in switched networks are dangerous, since packets can loop around forever and jam the network - as opposed to IP and routed networks, Ethernet frames do

Feature	Web	CLI	General Description
Enable STP	X	X	
Bridge priority	X	X	Section 16.1.2
Max age	X	X	Section 16.1.1
Hello time	X	X	Section 16.1.1
Forward delay	X	X	Section 16.1.1
View general RSTP/STP settings	X	X	
Per Port settings			
Enable STP	X	X	
Admin Edge	X	X	Section 16.1.1
Path Cost		X	Section 16.1.3
View per port RSTP/STP settings	X	X	
View RSTP/STP status	X	X	

Table 16.1: Summary of RSTP/STP features.

not include a *hop count* by which the switches could decide to drop a packet circulating around. Since a switched network may contain multiple loops, broadcast packets (or other packets flooded by the switches), leads to packet proliferation; this situation is generally referred to as a *broadcast storm*. On the other hand, loops in switched networks are desirable from a redundancy perspective.

Note
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.

Since loops in switched networks are so dangerous, layer-2 redundancy protocols such as STP and RSTP are very restrictive before putting a link in *forwarding* state. The main difference between STP and RSTP is that RSTP is able to react quicker to topology changes, thus can open an alternative path if a link in the active tree is broken, i.e., RSTP has shorter *convergence time* than STP. (FRNT has even faster convergence, see chapter 14.)

In RSTP/STP terminology, a switch is referred to as a *bridge*. Spanning tree is a

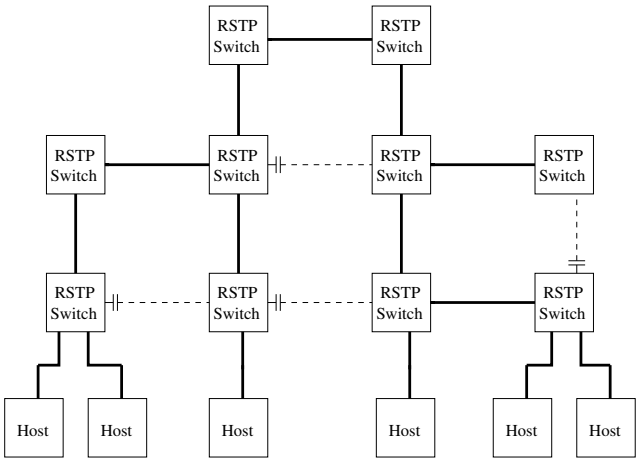


Figure 16.1: Example of RSTP creating a spanning tree. Dashed links have logically been ”cut off” from the active topology by RSTP, eliminating the loops.

plug-and-play protocol - bridges can use RSTP/STP to form a tree without need for any configuration. However, the protocol provides a set of parameters which the operator can use to fine-tune the network setup. Below is a list of those parameters of specific interest for the WeOS RSTP/STP implementation:

- *Bridge priority*: Used for *root bridge* and *designated bridge* election. See section 16.1.2.
- *Port/Path cost*: 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. See section 16.1.3.
- *Max age/Hello time*: Used to detect that a STP/RSTP neighbour is down. The *max age* also puts a protocol limit to the *size* of the network[1].
- *Forward Delay*: 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).
- *Admin Edge:* 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

[1] In RSTP the *Message Age* field in the *Hello Messages* effectively acts as a hop count, counting the distance from the Root. If the *Message Age* exceeds the *Max Age* the packet is dropped. Thus, the setting of the *Max Age* parameter restricts the size of the RSTP LAN.

can be configured as an *admin edge* port.

Access ports and inter-switch ports
It is recommended that all ”inter-switch ports” (ports connecting switches) are configured as ”non-edge ports” (admin edge disabled), and that all ”access ports” (ports where hosts connect) are configured as ”edge ports” (admin edge enabled).
For robustness purposes, all ports are set to **”no admin-edge”** when *spanning tree* is enabled. To improve performance on ”access ports” (leading to hosts), these ports should be configured as **”admin-edge”**.

When configured as *admin edge* the port will:

- be put in *FORWARDING* state quickly after system boot, and
- be kept in *FORWARDING* state during periods when the spanning tree topology is changing.

An *admin edge* assumes the port leads to a host or a router (i.e., not another bridge), and the port is therefore put in *FORWARDING* state without first verifying that the LAN is still loop free. The bridge will still send *Hello Messages* on *admin edge* ports, and will react on any incoming *Hello Messages* as it would on regular (non-”admin edge”) ports. Thus, even if loops may occur via an *admin edge* port, the bridge will generally be able to receive the high-priority RSTP messages, and cut the loop by putting the appropriate port in *BLOCKING*.

The IEEE std 802.1D-2004 specifies restrictions on the *Max age* parameter with respect to the *Hello time* and the *Forward delay* as shown below. This affects how these parameters can be configured.

- $Max\ age \geq 2 * (Hello\ time + 1)$
- $Max\ age \leq 2 * (Forward\ Delay - 1)$

Note
Some of the RSTP/STP parameters (Max age, Hello time, and Forward Delay) need to be set consistently throughout all bridges with the LAN infrastructure. Therefore, bridges inherit these parameter values from the current *root bridge*, irrespective of the corresponding parameter setting in the bridge itself.

16.1.2 Bridge Identity

Each bridge is assigned an 8 byte bridge identifier (bridge ID) as shown in fig. 16.2.

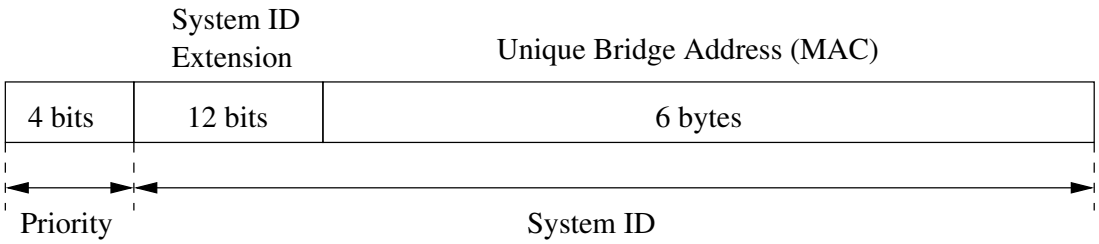


Figure 16.2: Structure of bridge ID.

The bridge ID is divided into a *priority* part (4 bits) and a *system ID* (60 bits). The bridge with the lowest bridge ID within the LAN will become the root bridge, i.e., lower *priority* means greater chance to become root bridge. The bridge ID is also used to select a *designated bridge* on a link, when multiple bridges on the link have the same ”least cost path” to the root bridge.

The format of the bridge ID follows IEEE std. 802.1D-2004 (RSTP). It differs from the structure specified in IEEE std. 802.1D-1998 (STP), where the *priority* field was 2 bytes and the *system ID* field was 6 bytes. The change in structure was made with respect to the multiple spanning tree protocol (MSTP) defined in IEEE std. 802.1Q-2005 (WeOS currently does not support MSTP).

- *Priority (4 bits)*: Can take values in range 0-15, where 8 is default. 0 (zero) means highest priority and 15 lowest priority. Compared to the ”old” 2 byte priority field of STP, this is rather a *priority factor* field, which can be multiplied by *4096* to get the ”old” STP priority.
- *System ID Extension (12 bits)*: Set to all zeros in WeOS.
- *Unique Bridge Address*: Tie-breaker ensuring the bridge ID will be unique. WeOS uses the *base MAC address* assigned to the switch for this field.

16.1.3 Path Cost

Each port is associated with a cost referred to as a *path cost*. Low-speed links are generally given a high cost, which increases the probability of the port ending up in *blocking* state (and vice versa), in case spanning tree discovers a loop.

By default, the path cost of a port is assigned dynamically with values related to the port speed (in-line with the recommendations of IEEE std 802.1D-2004). The same path costs are used irrespective if the port is operating in RSTP or STP mode.

Port Speed (Mbit/s)	RSTP path cost
10	2000000
100	200000
1000	20000

It is also possible to configure the path cost manually. That may be useful to get more fine grain control of which port in the LAN should be put in *blocking* state. Setting path costs manually may be desirable when operating a LAN including a mix of RSTP and STP capable, since STP uses a different set of default path costs.

16.1.4 RSTP and STP coexistence

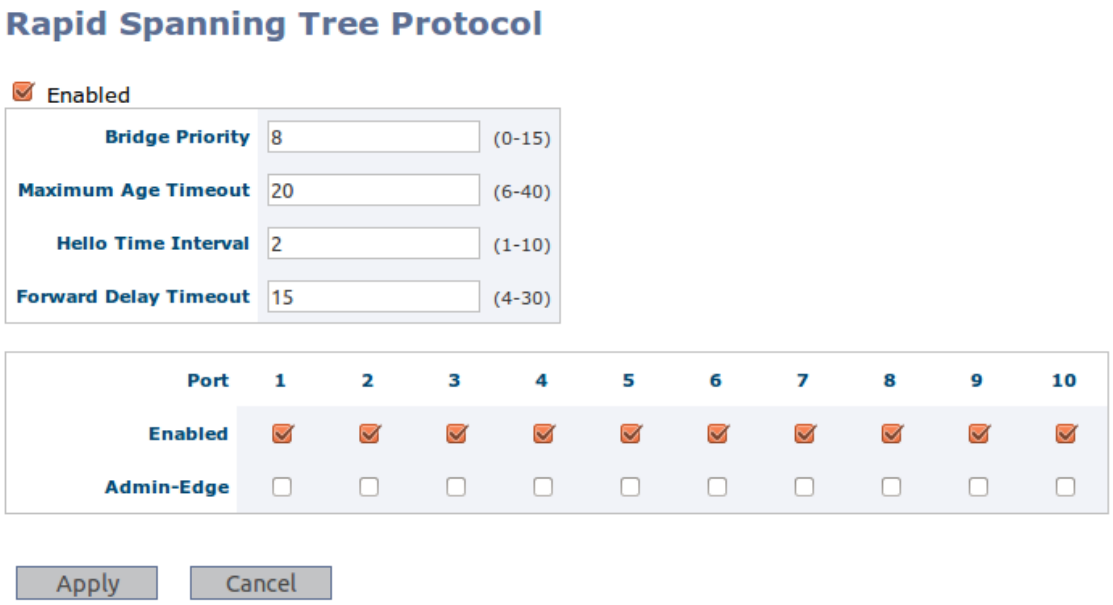
WeOS supports both RSTP and STP, but WeOS always attempts to run RSTP on every spanning-tree enabled port. WeOS automatically shifts to STP mode on a port, if it detects a bridge running STP on that port. Other ports continue operating in RSTP mode. When operating a network including a mix of RSTP and STP bridges, it may be necessary to configure path costs manually to get the intended spanning tree behaviour, see also section 16.1.3.

16.2 Managing RSTP via the web interface

16.2.1 Managing RSTP Settings

Menu path: Configuration ⇒ L2 Redundancy ⇒ RSTP

On the RSTP configuration page you will be presented to the current settings for RSTP on your switch, see below. You may change the settings by editing the page.



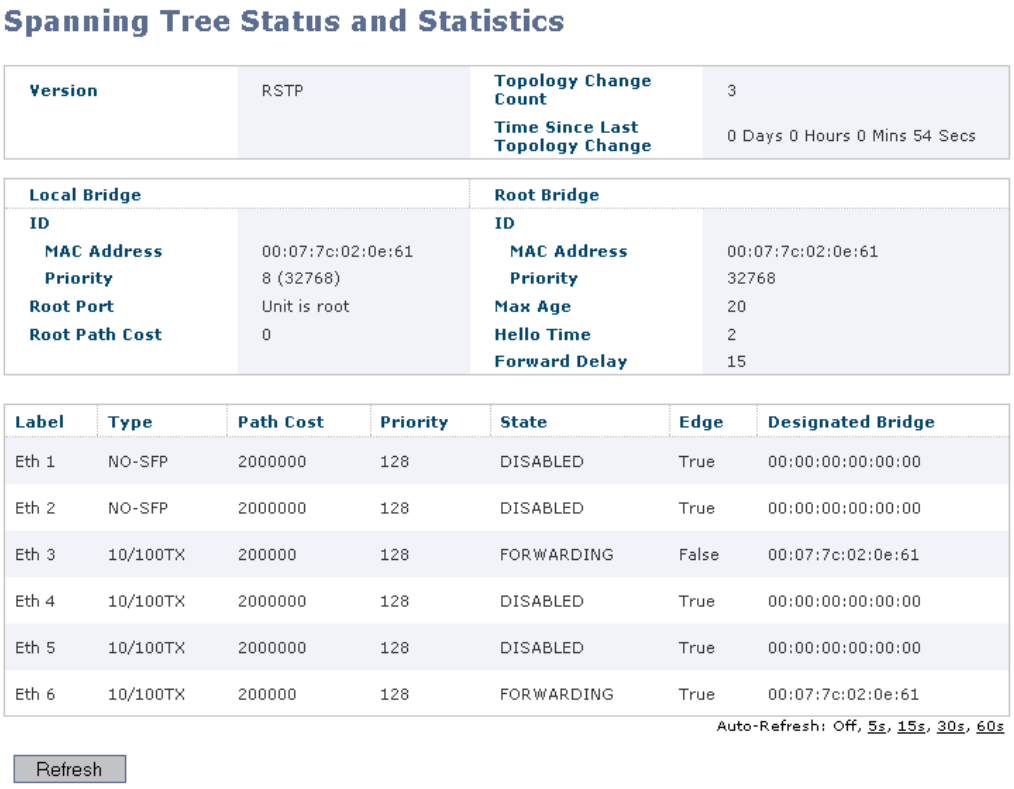
Enabled	Check the box to enable RSTP. If you have a JavaScript enabled browser the other settings will not be displayed unless you check this box.
Bridge Priority	A priority level used in root bridge selection. A lower value increases the probability for this switch to be elected as root bridge.
Maximum Age Timeout	The time the unit will wait before considering a neighbour designated bridge is down after the last Hello message was heard from the neighbour.

Continued on next page

	Continued from previous page
Hello Time Interval	The time between two consecutive transmissions of hello messages.
Forward Delay Timeout	The time an interface takes to change from blocking to forwarding state. Only used when operating in STP mode.
Edge Port	Ports connected to end hosts and routers (i.e., not to another switch) can be set as admin-edge ports. This avoids unnecessary BLOCKING of such ports at system startup or when a topology change occurs. It is *recommended* that this box is checked for every port where it is certain that only end hosts and routers connect. Ports which (may) connect to another switch should un-check this box.

16.2.2 RSTP Status and Statistics

Menu path: Status ⇒ L2 Redundancy ⇒ RSTP



Version	Always RSTP, with fallback to STP.
Topology Change Count	Number of RSTP topology changes since switch start-up.
Time Since Last Topology Change	Time since last topology change.
ID	The local and elected root bridge ID, used for root bridge and designated bridge election; consists of two parts: **MAC Address** The local MAC-address that is used for bridge ID. If local and root values are equal, this switch is root. **Priority** Priority value configured on the unit.

Continued on next page

	Continued from previous page
Root Port	The port with the open path to the root switch. If this switch is root, the text **`Unit is root`** will be displayed.
Root Path Cost	Calculated cost to designated root switch.
Max Age	Used to detect that a STP/RSTP neighbor is down. Current value learnt from BPDUs.
Hello Time	The time between two consecutive transmissions of hello messages. Current value learnt from BPDUs.
Forward Delay	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. Current value learnt from BPDUs.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click **`Off`** to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.

Port Status

Label	Port label, identifying the port.
Type	Type of port, e.g. Eth for ethernet.
Path Cost	Path cost associated with the port.
State	**FORWARDING** Unit forwards packets. Normal operation. **LEARNING** The port is preparing itself for entering FORWARDING state. **BLOCKING** Unit does not forward any packets. **DISABLED** Port does not participate in operation.
Edge	If TRUE the port is in admin edge mode and assumes the port leads to a host or a router (i.e., not another bridge), and the port is therefore put in FORWARDING state without first verifying that the LAN is loop free. If FRNT, the port is controlled by FRNT protocol.
Designated Bridge	The designated bridge MAC-address.

16.3 Managing RSTP via the CLI

Command	Default	Section
Spanning Tree Configuration		
[no] spanning-tree	Disabled	Section 16.3.1
priority <0-15|0-65536>	8 (32768)	Section 16.3.2
max-age-time <6-40>	20	Section 16.3.3
hello-time <1-10>	2	Section 16.3.4
forward-delay <4-30>	15	Section 16.3.5
stp-port <PORTLIST|all>		Section 16.3.6
[no] enable	Enabled	Section 16.3.7
[no] admin-edge	Disabled	Section 16.3.8
[no] path-cost <0-20000000>	0 (Auto)	Section 16.3.9
Spanning Tree Status		
show spanning-tree		Section 16.3.10

16.3.1 Manage RSTP

Syntax `[no] spanning-tree`

Context Global Configuration context

Usage Enter Spanning Tree Configuration context, and activate spanning-tree (if not already activated). Use **"`no spanning-tree`"** to disable spanning-tree and to remove spanning-tree configurations.

Use **"`show spanning-tree`"** to view general spanning-tree settings, given that spanning-tree is enabled (also available as **"`show`"** command within the Spanning Tree Configuration context.

Default values Disabled

16.3.2 Bridge Priority Setting

Syntax `priority <0-15|0-65535>`

Context Spanning Tree Configuration context

Usage Set bridge priority, where a low value means high priority, which increase the probability of being elected as *root bridge*. Values can be entered in two ways, either in range 0-15, which corresponds to the 4-bit priority field specified in IEEE std 802.1D-2004, or in range 16-65535 which corresponds to the traditional 2 byte priority field defined in IEEE 802.1D-1998. In the latter case, the value is divided by 4096, and stored as a value 0-15. See section 16.1.2 for more information.

"no priority" resets the bridge priority to the default setting.

Use **"show priority"** to view the current bridge priority setting.

Default values 8 (32768)

16.3.3 Max Age Setting

Syntax `max-age-time <6-40>`

Context Spanning Tree Configuration context

Usage 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.

"no max-age-time" resets the max age timeout to the default setting.

Use **"show max-age-timeout"** to view the current max age timeout setting.

Default values 20

Error messages An error message is given if the **"max-age-time"** is not given a valid value with respect to **"hello-time"** or **"forward-delay"**, see section 16.1.1.

16.3.4 Hello Interval

Syntax `hello-time <1-10>`

Context Spanning Tree Configuration context

Usage Set spanning-tree hello time interval. Since bridges use the hello time configured at the root bridge, this parameter setting only matters if this bridge becomes the root bridge.

”no hello-time” resets the hello time to the default setting.

Use **”show hello-time”** to view the current hello time setting.

Default values 2 (seconds)

Error messages An error message is given if the **”hello-time”** is not given a valid value with respect to **”max-age-time”**, see section 16.1.1.

16.3.5 Forward Delay

Syntax `forward-delay <4-30>`

Context Spanning Tree Configuration context

Usage Set spanning-tree forward delay. Since bridges use the forward delay configured at the root bridge, this parameter setting only matters if this bridge becomes the root bridge.

”no forward-delay” resets the forward delay to the default setting.

Use **”show forward-delay”** to view the current forward delay setting.

Default values 15 (seconds)

Error messages An error message is given if the **”forward-delay”** is not given a valid value with respect to **”max-age-time”**, see section 16.1.1.

16.3.6 Manage RSTP Ports

Syntax `[no] stp-port <PORTLIST|all>`

Context Spanning Tree Configuration context

Usage Enter Spanning Tree Port Configuration context to manage per port spanning-tree settings for one or more ports.

”no stp-port <PORTLIST|all>” (e.g., **”no stp-port all”**) will disable spanning tree for the specified ports.

Use **”show stp-port <PORTLIST|all>”** to view the spanning tree settings for the specified port(s).

Default values Not applicable.

Example

```
example:/config/spanning-tree/#> show stp-port all
Port     Enabled  Admin-Edge  Path-cost
================================================================================
Eth 1    YES       NO              AUTO
Eth 2    YES       NO              AUTO
Eth 3    YES      YES              AUTO
Eth 4    YES      YES              AUTO
example:/config/spanning-tree/#>
```

16.3.7 Enable Spanning Tree on a Port

Syntax `[no] enable`

Context Spanning Tree Port Configuration context

Usage Enable the spanning tree protocol on a port. Use **”no enable”** to disable spanning tree protocol on a port.

Default values Enabled

16.3.8 Admin Edge Setting

Syntax `[no] admin-edge`

Context Spanning Tree Port Configuration context

Usage Configure the port as an *access* port (**”no admin-edge”**), or as an *inter-switch port* (**”admin-edge”**).

Note

It is *recommended* that every port where it is certain that only end hosts and routers connect (but **not** switches/bridges) are configured as **”admin-edge”**. Port which (may) connect to **another switch/bridge** should be configured as **”no admin-edge”**.

Use **”show admin-edge”** to view the admin edge setting for this port.

Default values Disabled (**”no admin-edge”**)

16.3.9 Path Cost Setting

Syntax `[no] path-cost <0-20000000>`

Context Spanning Tree Port Configuration context

Usage Configure the spanning tree path cost for a port. A low speed link should get a higher cost, a high speed link a lower cost. Use **”path-cost 0”** (or **”no path-cost”**) to have the path-cost assigned automatically depending on the port speed (see section 16.1.3).

Values in range 1-20000000 means a statically configured path cost of the given value.

Use **”show path-cost”** to view the path cost setting for this port.

Default values Automatic (**”path-cost 0”**)

16.3.10 Show RSTP Status

Syntax `show spanning-tree`

Context Admin Exec context.

Usage Show spanning-tree status information, including current port states, root bridge ID, etc..

Default values Not applicable.

Chapter 17

Link Aggregation

This chapter describes WeOS support for link aggregation (IEEE 802.3ad/802.1AX[13]). With link aggregation, two or more Ethernet links can be bundled and treated as a single MAC entity by the upper layer protocols. The primary use is to achieve *redundancy* in layer-2 bus topologies. A coarse form of load balancing is also provided, but only if different traffic flows are mapped to different aggregate member links.

WeOS supports the standard Link Aggregation Control Protocol (LACP[13]) for aggregation control, but also *static* aggregation control, where the active set of member links is solely determined based on their link up/down state.

17.1 Link Aggregation Support in WeOS

Feature	Web	CLI	General Description
Enable/Disable Aggregate	X	X	Section 17.1.1
Define Member Ports	X	X	-”-
Static Aggregation Control	X	X	Section 17.1.2
LACP Aggregation Control	X	X	Section 17.1.3
Timeout (Short/Long)	X	X	-”-
Active/Passive	X	X	-”-
Show Link Aggregate Status			

17.1.1 Introduction to Link Aggregation

Link aggregation enables physical links to be bundled together to form a single logical link, an *aggregated link*, see fig. 17.1. Upper layer protocols will treat the aggregate as a single MAC entity, i.e., as one Ethernet port with its own label, a MAC address assigned, and so on. In WeOS, aggregates are named **”a0”**, **”a1”**, etc., and inherit their MAC address from one of their member ports.

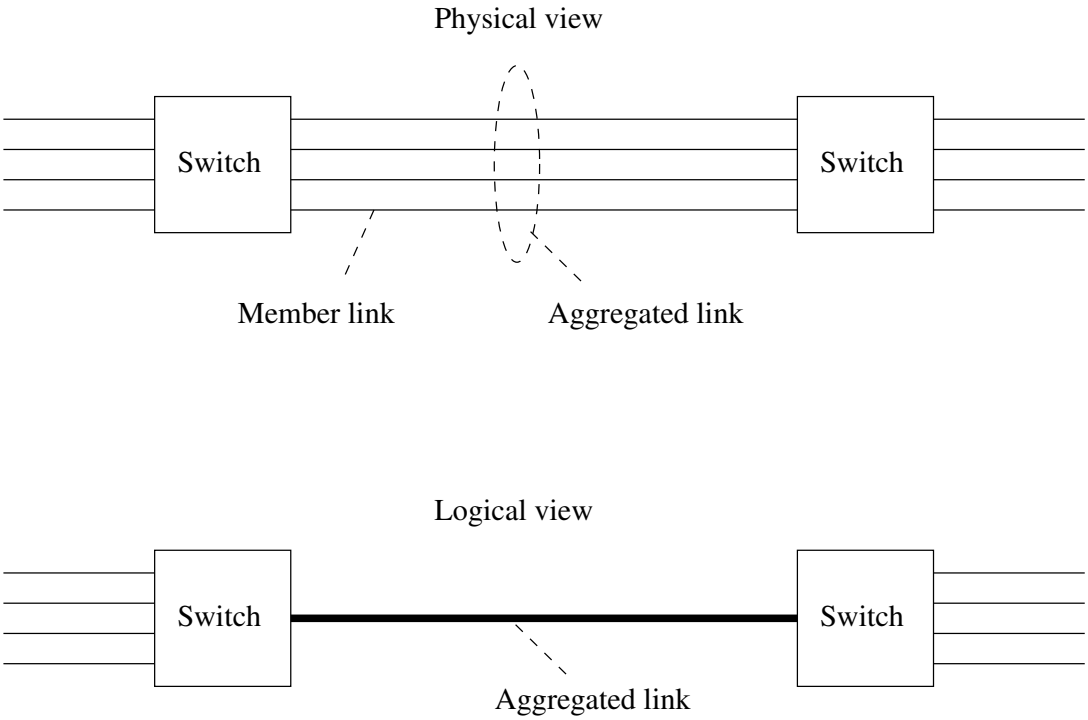


Figure 17.1: Example of link aggregation with four member links

All member ports in an aggregate are able to forward data. However, the IEEE802.1AX standard[13] mandates the aggregate to deliver packets *in order* per *data flow* to avoid problems for upper layer protocols. This means the switch will send all traffic of an individual *data flow* through the same member link. Other *flows* may be sent through other member links. The effectiveness of this *load balancing* depends on several factors:

- The granularity by which the switch can distinguish between different traffic flows: WeOS units determine packet flow based on the combination of the

source and destination MAC address of the packet[1] (done in hardware).

- *The distribution of traffic flows:*. If there are *many* flows (and if they are of *equal load*) the ability to load balance improves. This depends on the traffic patterns in your network. Avoiding patterns where all traffic end up with the same source and destination MAC over the aggregate improves the ability to load balance[2].
- *The mapping of traffic flows to different member links:* WeOS units map traffic flows to different (active) member links in a *static* way. This mapping aims to equalise the number of flows mapped to each member link, but its effectiveness is limited when the number of flows are low.

Note
To summarise, link aggregation should generally be used as a means to achieve **redundancy** in bus topologies. It may be used to increase data capacity, however, the ability to load balance between the member links is limited and depends on the use case.

When an aggregate is configured in WeOS, the following restrictions apply:

- *Ethernet as member ports:* Only aggregation of Ethernet ports is supported.
- *Member ports explicitly associated with aggregate:* For a port to be part of an aggregate, it must explicitly be associated with that aggregate.
- *Maximum 8 aggregates:* At most 8 aggregates can be configured on a WeOS unit.
- *Maximum 8 member ports per aggregate:* Each aggregate can have at most 8 member ports.
- *Member ports in same slot:* In slot based WeOS products (see section 8.1.1) all member ports must reside in the same slot as of WeOS v4.17.1. Similar restrictions apply to WeOS Viper, RedFox Rail (RFR) and RedFox Industrial Rack (RFIR) products.

A aggregate has state *Down* when all its member ports have state *Down*, and the aggregate is *Up* when at least one of its member ports has state *Up*.

[1] The algorithm to determine flow uses a hash function applied to the packet's source and destination MAC address.

[2] *Switching* traffic over the link aggregate may improve load balancing as opposed to *routing* (routers typically use the same source and destination MAC for all unicast traffic). Multicast flows commonly utilise different destination MACs irrespective if the WeOS units are switching or routing, thus has good load balancing properties.

The next subsections provide additional information on WeOS support for link aggregates: sections 17.1.2 (static) and 17.1.3 (LACP) contain information on the methods to control link aggregates in WeOS, while section 17.1.4 include more details on using link aggregates in various low-layer features in WeOS.

17.1.2 Static Link Aggregates

For static link aggregates the including member ports are the only settings that have to be specified in the configuration. The members in an aggregate do not need to have the same speed settings, although that is the preferred setting (otherwise the capacity of the aggregate will be unbalanced).

Ports that are included in an aggregate and have link up will be qualified as active ports, and the network traffic will be sent on those links. If a link goes down or up in the aggregate the network traffic will be distributed over the new set of active links. Because an active link in an aggregate is qualified on the link status **no media converters** are allowed between statically aggregated ports. Below is a CLI configuration example where the static link aggregate a1 is configured with member ports 3 and 7 on a WeOS switch.

Example

```
example:/#> configure
example:/config/#> aggregate a1
example:/config/aggregate-a1/#> ports 3,7
example:/config/aggregate-a1/#> type static
example:/config/aggregate-a1/#> show
Name               : a1
 Status            : Enabled
 Type              : static
 Ports             : 3,7
example:/config/aggregate-a1/#> end
example:/config/#>
```

17.1.3 LACP Controlled Link Aggregates

The Link Aggregation Control Protocol (IEEE 802.3ad/802.1AX [13]) is a standard method for aggregating member links that have the same speed and duplex mode. The primary advantage over static link aggregation is the ability to confirm that the remote partner can handle aggregation. It is also possible to handle failover when media converters are present.

LACP relies upon periodic transmission of information and state between the switches. The protocol messages (LACP-PDUs) are sent by the first party (the Actor) to the second party (the Actor's protocol Partner) with information about what the Actor knows, both about its own state and that of the Partner.

Switches can be configured to *active* or *passive* participation in LACP. Passive LACP indicates the preference for not transmitting LACP-PDUs unless its Partner is Active LACP, i.e. it does not generate any LACP traffic by its own. Active LACP indicates the preference to participate in the protocol regardless of the Partner setting, i.e. it always generates LACP traffic.

LACP-PDUs are transmitted periodically when either the Actor or the Partner is configured with Active LACP. These transmissions will occur at either a *fast* or *slow* transmission rate depending upon the timeout setting (*short* or *long* timeout) of the Partner system.

The LACP state is determined by the contents of the LACP-PDUs and can be in any of the following states:

Detached The port is being detached from the aggregator.

Waiting The port is being attached to the aggregator.

Attached The port is attached to the selected aggregator.

Collecting Indicates that the receive function of this link is enabled.

Distributing Indicates that the transmit function of this link is enabled.

The switch will set a member port in forwarding state when LACP state is Distributing. For all other LACP states the port state will be blocking[3]. The aggregate is in forwarding state as long as at least one member port is in forwarding state. Also, the aggregate will be up as long as at least one member port is up.

WeOS assumes that the configured aggregate connects two switches. If the aggregate member ports on one switch is connected to several other switches LACP will only include member ports to one of the neighbours in the active port set:

- Ports to the neighbour with the highest total bandwidth will be selected.
- If several aggregates share the same bandwidth, then the aggregate is selected based on LACP *system priority*, *system identifier*, *port priority*, and *operational key*.

[3] If RSTP or FRNT are run over the aggregate, those protocols may also decide to set the ports in blocking state.

In WeOS v4.17.1, the LACP system priority is set to *0x8000* (hex), system identifier is set to the *MAC address of the first member port* of the aggregate, the port priority is set to *0x8000* (hex), and the operational key is set to the configured *aggregate identifier* (see sections 17.2 and 17.3). More information about aggregate selection can be found in IEEE 802.3ad/802.1AX [13].

17.1.4 Link Aggregates and Low layer protocols

17.1.4.1 Link Aggregation and VLAN

Ethernet and DSL ports on WeOS units are associated (*untagged* or *tagged*) with one or more VLANs as described in chapter 13. Link aggregates can **not** be mapped directly to VLANs. Instead the user must add each of the aggregate member ports to the intended VLAN(s).

For the setup in fig. 17.2, the physical ports 1-4 are mapped tagged (**"tagged 1-4"**) to VLANs 1&2 rather than the aggregates (i.e., **"tagged a1,a2"** is not possible as of WeOS v4.17.1). An extract of the configuration file is shown below.

Example

```
vlan 1
        name vlan1
        untagged 5-7
        tagged 1-4
        end

vlan 2
        name vlan2
        untagged 8-10
        tagged 1-4
        end
```

17.1.4.2 Link Aggregation and Link Alarms

As described in section 24.1 the operational state (Up/Down) of Ethernet and DSL ports can be used as alarm triggers, i.e., *link alarms*. When a port is a member of a link aggregate, it is still possible to define link alarms for the individual member ports. It is also possible to create link alarms for the aggregates.

Below is a CLI configuration example where a link alarm is configured for aggregate *a1*. The aggregate has state *Down* when all its member ports has state

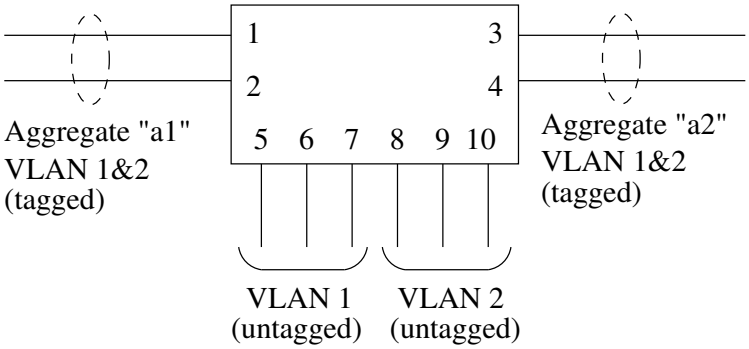


Figure 17.2: The physical ports 1-4 rather than the logical aggregates (*a1* and *a2*) are associated with the VLANs (VLAN 1 and 2).

Down, and the aggregate is *Up* when at least one of its member ports has state *Up*.

Example

```
example:/#> configure
example:/config/#> alarm
example:/config/alarm/#> trigger link-alarm
example:/config/alarm/trigger-2/#> port a1
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#>
```

17.1.4.3 Link Aggregation and unicast/multicast MAC learning

The MAC forwarding database (FDB, see section 13.1.8) holds information on where to forward *known* MAC addresses. Unicast addresses are learnt dynamically by looking at the source MAC of incoming packets, while multicast addresses are typically learnt dynamically via IGMP snooping (chapter 18), or entered manually[4] by the operator.

When a (unicast/multicast) MAC address is learnt dynamically on a member port of a link aggregate, all ports of the aggregate are added to the MAC address' FDB entry, since the link aggregation flow distribution mechanism can map traffic to the MAC address on any member port.

In the example below, aggregate *a1* consists of member ports *5* and *6*, and IGMP snooping is enabled on the VLAN the ports are associated with. An IGMP report

[4] See section 13.4.3 for CLI command to enter MAC forwarding database entries manually.

has been received for IP multicast address *225.1.2.3* (MAC *01:00:5e:01:02:03*) on one of the member ports and both ports are added to the *forwarding database* for that MAC address.

Example

```
example:/#> sh ip igmp
 VID  Querier IP       Querier MAC        Port  Interval  Timeout
-------------------------------------------------------------------------------
   1  192.168.2.200    LOCAL

 VID  Multicast Group  Filtered MAC Addr  Active ports
-------------------------------------------------------------------------------
   1  225.1.2.3        01:00:5E:01:02:03  a1

-------------------------------------------------------------------------------
Total: 1 filters, max 1200, in 1 VLAN.

example:/#> sh fdb
MAC                VLAN  State       Port(s)
===============================================================================
...
01:00:5e:01:02:03   ANY  IGMP        5-6
...
===============================================================================
FDB Aging time: 300 sec.
example:/#>
```

Similarly, traffic from unicast address *00:07:7c:00:02:61* has come in on one member port, thus both member ports are automatically added to the MAC's FDB entry.

Example

```
example:/#> sh fdb
MAC                VLAN  State       Port(s)
===============================================================================
...
00:07:7c:00:02:61   ANY  294 s       5-6
...
===============================================================================
FDB Aging time: 300 sec.
example:/#>
```

When adding (multicast) MAC addresses statically to the MAC FDB, each of the individual member ports needs to be specified. Thus, in the example below, with ports 5 and 6 belonging to aggregate *a1*, the command **"`mac 01:00:5e:00:11:22 port 5,6`"** is used (while **"`mac 01:00:5e:00:11:22 port a1`"** would not work as of WeOS v4.17.1).

Example

```
example:/#>
example:/#> configure
example:/config/#> fdb
example:/config/fdb/#> mac 01:00:5e:00:11:22 port 5,6
example:/config/fdb/#> end
```

17.1.4.4 Running FRNT or RSTP over Link Aggregates

It is possible to run FRNT (chapter 14) or RSTP (chapter 16) over a link aggregate. Fig. 17.3 shows an example of using FRNT together with link aggregation.

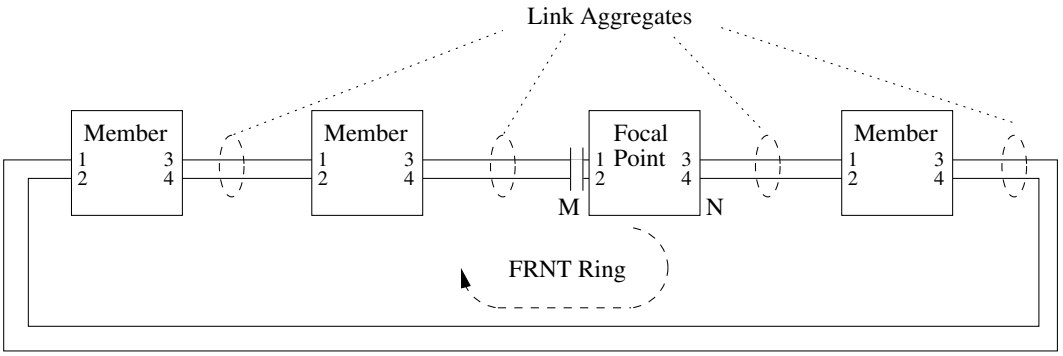


Figure 17.3: FRNT can run over aggregated links

Additional information on running RSTP over a link aggregate:

- *Failover performance:* RSTP failover performance may be degraded when running RSTP over a link aggregate as opposed to using regular links.
- *Forwarding/Blocking state:* An aggregate is forwarding data packets only if *both* RSTP and the link aggregate itself determine that it should be in forwarding state.
- *RSTP link cost:* The RSTP link cost can be configured manually. If ”auto” is used for cost calculation, WeOS determines the aggregate link cost based the aggregated bandwidth of the member ports (higher aggregated capacity gives lower RSTP cost).
- *Link Up/Down:* An aggregate is up if at least one of its member ports are considered up. An aggregate is down if all its member ports are down.

Additional information on running FRNT over a link aggregate:

- *Failover performance:* FRNT failover performance may be degraded when running FRNT over a link aggregate as opposed to using regular links.
- *Forwarding/Blocking state:* An aggregate is forwarding data packets only if *both* FRNT and the link aggregate itself determine that it should be in forwarding state.
- *Link Up/Down:* An aggregate is up if at least one of its member ports are considered up. An aggregate is down if all its member ports are down.
- *Mixing aggregated and regular links:* The topology in fig. 17.3 uses link aggregation throughout the whole FRNT ring. It is possible to run link aggregation on a subset of the links in the FRNT ring.

17.1.4.5 Link Aggregation and other Low-level WeOS features

Use of link aggregation with other low-level features, e.g., *port monitoring* (section 7.1.10), port access control (section 13.2 and chapter 21), etc. is not supported as of WeOS v4.17.1. To use those features together with link aggregation it may be possible to specify the individual member ports in the configuration, however, the behaviour is undefined and its use is unsupported.

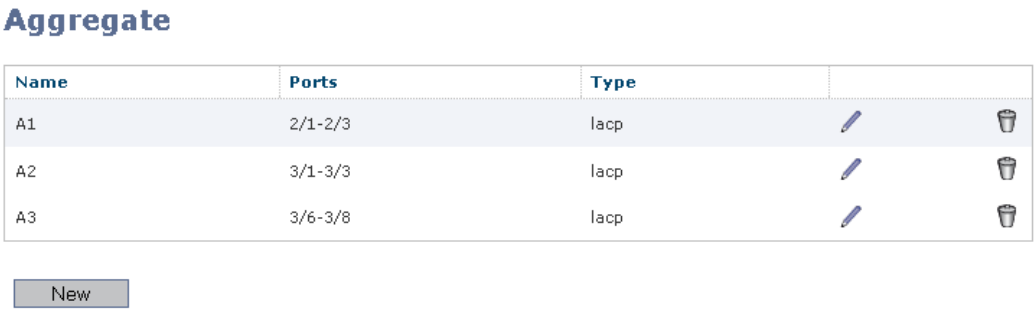
17.2 Link Aggregation Settings and Status via the Web Interface

17.2.1 Configuring Link Aggregation Settings via the Web Interface

Menu path: Configuration ⇒ Port ⇒ Aggregate

On the Link Aggregate overview page all configured link aggregates will be presented in a list, see below.

When first accessing this page link aggregates can be created by pressing the **New** button.

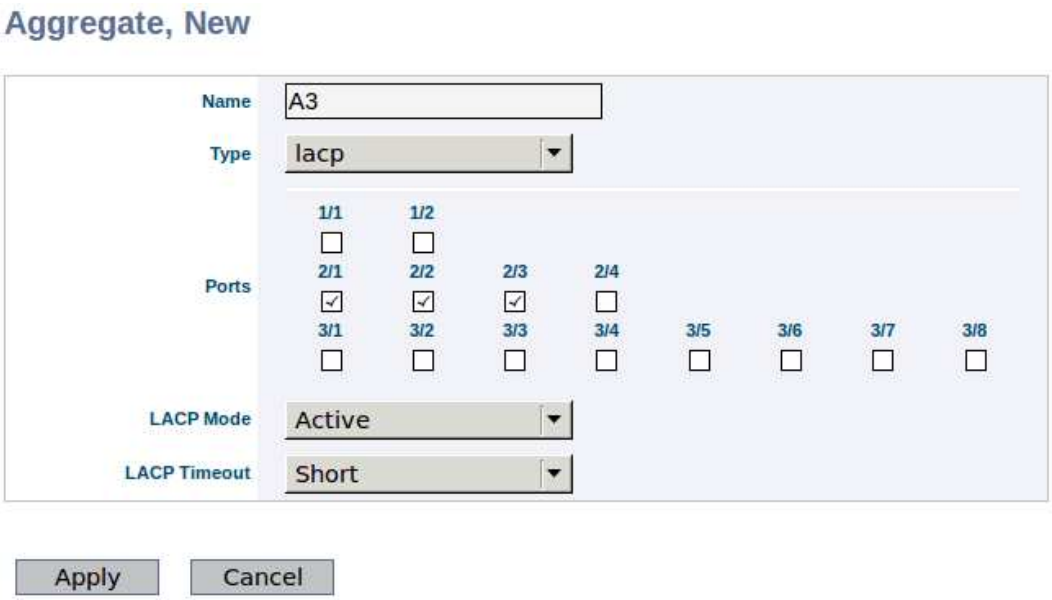


Name	The link aggregate name.
Ports	The set of ports defined for this aggregate.
Type	The type of the aggregate, **Static** or **LACP**.
Edit	Click this icon to edit an existing aggregate.
Delete	Click this icon to remove an aggregate. You will be asked to acknowledge the removal before it is actually executed.
New	Click the **New** button to create a new link aggregate.

17.2.2 Create new link aggregate using the web interface

Menu path: Configuration ⇒ Port ⇒ Aggregate ⇒ **New**

When clicking the **New** button, you will be presented to the aggregate *new* page.



Name	The link aggregate name. Valid values are A{n} or a{n}, where n is an integer.
Ports	The set of ports to be included in this aggregate. Only ports in the same slot may be aggregated together.
Type	The type of the aggregate, **Static or LACP.**
LACP Mode	Only available for type LACP. Modes: **Active** Always send frames (LACP-PDUs) along the configured links. **Passive** Only send frames (LACP-PDUs) along the configured links if any LACP-PDU frames have been received.
LACP Timeout	Only available for type LACP. The type of the aggregate: **Short** 3 seconds **Long** 90 seconds

For more information, see section 17.1.

17.2.3 Edit link aggregate settings using the web interface

Menu path: Configuration ⇒ Port ⇒ Aggregate ⇒ 

When clicking the *Edit* icon for an aggregate you will be presented to the aggregate edit page, which is identical to the *new* page. See section 17.2.2 for description of fields.

17.2.4 Link Aggregation Status via the Web Interface

Menu path: Status ⇒ Port ⇒ Aggregate

This page display status information for the currently configured link aggregates.

Aggregate Status

Name	Link	Mac	Type	Port
A1	UP	00:07:7c:82:1f:c9	lacp	Label / Link / Active / Link State / LACP State Eth 2/1 / UP / No / FORWARDING / DISTRIBUTING Eth 2/2 / Down / No / BLOCKING / ATTACHED Eth 2/3 / Down / No / BLOCKING / ATTACHED
A2	Down	00:07:7c:82:1f:cd	lacp	Label / Link / Active / Link State / LACP State Eth 3/1 / Down / No / BLOCKING / ATTACHED Eth 3/2 / Down / No / BLOCKING / ATTACHED Eth 3/3 / Down / No / BLOCKING / ATTACHED
A3	UP	00:07:7c:82:1f:d2	lacp	Label / Link / Active / Link State / LACP State Eth 3/6 / UP / Yes / FORWARDING / DISTRIBUTING Eth 3/7 / Down / No / BLOCKING / ATTACHED Eth 3/8 / UP / Yes / FORWARDING / DISTRIBUTING

Auto refresh: Off, 5s, 15s, 30s, 60s

Refresh

Name	The link aggregate name.
Link	The aggregate link status. Up/Down.
MAC	The aggregate MAC address.
Type	The type of the aggregate, **Static or LACP.**
Port Label	The port label for the ports included in the aggregate.
Port Link	Up/Down.

Continued on next page

Continued from previous page

Port Active	Indicates if this port is an active member of this aggregate.
Port Link State	The port state for this port. **FORWARDING** Unit forwards packets. Normal operation. **LEARNING** The port is preparing itself for entering FORWARDING state. (Only applicable if RSTP/STP is used on the aggregate.) **BLOCKING** Unit does not forward any packets. The port is put in blocking state by LACP, or by STP/RSTP or FRNT if used on the aggregate. **DISABLED** Port does not participate in operation.
Port LACP State	The LACP negotiation state for this port: **DETACHED**, **WAITING**, **ATTACHED**, **COLLECTING**, or **DISTRIBUTING**. In the **DISTRIBUTING** state, the port is ready to send and receive data as part of the aggregate. See section 17.1.3 or [13] for more information.

17.3 Managing Link Aggregation via CLI

Command	Default	Section
Configure Link Aggregate		
[no] aggregate <AGGREGATE_ID>	N/A	Section 17.3.1
[no] enable	Enabled	Section 17.3.2
[no] ports <PORTLIST>	N/A	Section 17.3.3
[no] type <static|flhp|lacp>	lacp	Section 17.3.4
LACP Specific Settings		
[no] active	active	Section 17.3.5
[no] timeout <short|long>	short	Section 17.3.6
Aggregate Status		
show aggregate		Section 17.3.7

17.3.1 Manage a Link Aggregate

Syntax `[no] aggregate <AGGREGATE_ID>`

Context Global Configuration context

Usage Create, modify or remove a link aggregate.
Enter the *Link Aggregate Configuration* context of the given aggregate identifier (a0-a*N*), where N is a number (up to 8 aggregates can be created). If this is a new link aggregate, the aggregate is created.

Use **"no aggregate <AGGREGATE_ID>"** to remove an existing link aggregate, or **"no aggregate"** to remove all link aggregates.

Use **"show aggregate"** to list configured aggregates. To list details of a configured aggregate, enter its configuration context and run **"show"** from there.

Default values When using the **"no aggregate"** form (without providing a specific aggregate ID), all link aggregates are removed.

Example Listing configured aggregates, and listing details for a LACP aggregate.

Example

```
example:/config/#> show aggregate
a1                 static  1-2
a2                   lacp  5-6
example:/config/#> aggregate a2
example:/config/aggregate-a2/#> show
Name               : a2
 Status            : Enabled
 Type              : lacp
 Ports             : 5-6
 LACP mode         : active
 LACP timeout      : short
example:/config/aggregate-a2/#>
```

17.3.2 Enable/disable a Link Aggregate

Syntax `[no] enable`

Context Link Aggregate Configuration context

Usage Enable/disable this aggregate instance. Use **"enable"** to enable and **"no enable"** to disable this aggregate. When disabled, the configured member ports will not be part of this aggregate, i.e., they will operate as regular (non-aggregate) ports.

Use **"show enable"** to view the currently configured setting.

Default values Enabled (**"enable"**)

17.3.3 Configure Link Aggregation Member Ports

Syntax `[no] ports <PORTLIST>`

Context Link Aggregate Configuration context

Usage Add/remove a list of ports to/from the port member set of this link aggregate. Use **"no ports"** (without providing a port list) to remove all ports from the member set.

Use **"show ports"** to view the currently configured list of ports.

Default values When using the **"no ports"** form (without providing a specific PORTLIST), all ports are removed.

”PORTLIST” is a comma separated list of port ranges without intermediate spaces, e.g., **”X1-X2,X4”**.

17.3.4 Configure Link Aggregate Control Mode

Syntax `[no] type <static|flhp|lacp>`

Context Link Aggregate Configuration context

Usage Set mode/operation for this aggregate. Use **”no type”** (without providing a mode) to reset to default value.

> **Warning**
> As of WeOS version v4.17.1, the use of FLHP for link aggregation control is provided as a technology preview feature. All use of the FLHP link aggregation control feature except for testing is discouraged.

Use **”show type”** to view the currently configured mode.

Default values lacp (**”no type”**)

17.3.5 Configure LACP Active/Passive Mode

Syntax `[no] active`

Context Link Aggregate Configuration context (only available when aggregate control mode is lacp)

Usage Select LACP mode, i.e. active or passive participation in LACP (see section 17.1.3). Use **”active”** to select active mode and **”no active”** to select passive mode.

Use **”show active”** to view the currently configured setting.

Default values Active (**”active”**)

17.3.6 Configure LACP Timeout

Syntax `[no] timeout <short|long>`

Context Link Aggregate Configuration context (only available when aggregate control mode is lacp)

Usage Select LACP timeout, i.e. the number of seconds before invalidating received LACP information (see section 17.1.3). Use **”timeout short”** to set the timeout to 3 seconds and **”timeout long”** to set the timeout to 90 seconds.

Use **”show timeout”** to view the currently configured setting.

Default values Short, i.e. 3 seconds (**”no timeout”**)

17.3.7 Show Status of Link Aggregates

Syntax `show aggregates`

Context Admin Exec context

Usage Display status information for all configured aggregates. The header line displays the aggregate information including the name, its MAC address, and the aggregate control mode.

Each member link is listed with link status, whether or not the link is currently an active member of the aggregate, and the link state.

Aggregates using LACP also displays the LACP state (see section 17.1.3) and partner information. Partner ID is the system id of the peer, port is the remote port, and key is the operational key. In WeOS, the operational key is equal to the aggregate id.

Default values Not applicable

Example In this example an aggregate (a1) is configured. Both member ports are up, but port ’Eth 5’ is unused, since no LACP partner has been discovered on that link.

Example

```
example:/#> show aggregates
Aggregate a1  MAC: 00:07:7c:00:30:b5  Type: lacp
-------------------------------------------------------------------------------
Port     Link  Active   Link State  LACP State    Partner ID        Port  Key
-------------------------------------------------------------------------------
Eth 5      UP  No       Blocking    ATTACHED      00:00:00:00:00:00    0    0
Eth 6      UP  Yes      Forwarding  DISTRIBUTING  00:07:7c:00:02:61    2    1
example:/#>
```

Example In this example a static aggregate (a2) is configured. Two member ports are up and 'Eth 9' is down.

Example

```
example:/#> show aggregates
Aggregate a2  MAC: 00:07:7c:84:91:6b  Type: static
-------------------------------------------------------------------------------
Port     Link  Active   Link State
-------------------------------------------------------------------------------
Eth 7      UP  Yes      Forwarding
Eth 8      UP  Yes      Forwarding
Eth 9    DOWN  No       N/A
example:/#>
```

Chapter 18

Multicast in Switched Networks

This chapter gives a brief overview of multicast, with a focus on IP multicast, and how it can be controlled in a WeOS device using IGMP snooping. The chapter also covers non-IGMP capable devices and how they can be integrated into a network with IGMP enabled.

18.1 Overview

Feature	Web	CLI	General Description
IGMP Querier Mode	X	X	Section 18.1.1
IGMP Query Interval	X	X	-”-
IGMP Fast Leave	X	X	Section 18.1.3
Low Bandwidth Networks	X	X	Section 18.1.4
Multicast Router Ports	X	X	Section 18.1.2
Multicast Router Timeout		X	Section 18.1.1
View IGMP Snooping Settings	X	X	

Multicast, as opposed to unicast, is a very efficient means of communicating information to more than one receiver. The main difference between multicast and broadcast is that multicast can be controlled. When disabling its control mechanisms, like IGMP, multicast behaves like broadcast.

Thus, when distributing IP multicast data in a switched network, switches within the LAN can:

- treat multicast traffic as broadcast, i.e., forward it on all ports (in the same VLAN), or
- limit forwarding of multicast only to subscribers

The latter method requires switches to inspect Internet Group Management Protocol (IGMP) control messages exchanged by hosts and routers to learn which ports lead to subscribers – this mechanism is referred to as *IGMP snooping*[4]. With IGMP Snooping enabled, WeOS switches dynamically keep track of up to 2048 multicast addresses[1].

As part of the IGMP snooping support, WeOS also enables a *switch* to act as *IGMP querier* – a role which is usually handled by a *multicast router*. Having switches with IGMP querier capabilities enables efficient distribution of IP multicast in networks without multicast routers.

Warning

WeOS devices can only limit the broadcast effects of multicast on a Layer-2 basis, it is therefore important to design IPv4 multicast networks so that groups do not overlap. For example, 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. This means that both groups will be forwarded by the device and potentially overloading the intended receiver. See RFC 1112, http://tools.ietf.org/html/rfc1112, for details on how IP multicast groups map to MAC multicast addresses.

18.1.1 IGMP Snooping

The switch is capable of efficiently distributing IP(v4) multicast traffic on LAN interfaces by means of IGMP snooping. IGMP Snooping is enabled by default per VLAN, see section 13.1.5.

- With IGMP 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 snooping *disabled* on a VLAN, multicast traffic is forwarded on all ports in that VLAN, i.e., like broadcast traffic

[1] Special restriction for DDW-142 and DDW-142-485: On these products the MAC address database can hold at most 1000 addresses in total (unicast and multicast MAC). Thus, the upper limit for multicast addresses possible to keep track of is roughly 1000.

- Ports shared between multiple VLANs may have different IGMP snooping settings on different VLANs, i.e., one VLAN may have IGMP snooping *enabled* and another may have it *disabled*. The *disabled* mode takes precedence on such ports, i.e., multicast will be flooded on ports where at least one VLAN has IGMP Snooping *disabled*

As part of the IGMP snooping functionality, the switch can also act as an IGMP Querier, and settings for *querier mode*, and *query interval* are provided.

Querier mode: By default the switch has *auto mode* enabled. It relies on the standard IGMP protocol to elect a designated IGMP querier on each LAN[2]. With auto mode unknown multicast is flooded to the elected querier, which acts as a distribution point to the rest of the network. Keep this in mind when designing your network, for many use-cases it is valuable information.

The *forced querier mode* is a non-standard setting specific to WeOS that in some cases can be more fault tolerant. When all switches on a LAN are set to this mode they will all discard the election mechanism of the protocol and always send queries. This not only incurs a notable penalty on the network due to the overhead of IGMP messages flooding the LAN, but it also causes all unknown multicast to be flooded to all switches. This makes the network less vulnerable to the loss of one querier and all multicast is always available to end devices. It is however *not recommended* due to the broadcast like effects it causes.

In *proxy mode*, the switch normally only acts as a silent forwarder of IGMP queries (and reports) between the IGMP querier and end devices. However, to prevent loss of multicast traffic in the case when there exist no elected IGMP querier on a LAN, the switch will initiate queries with the source IP address 0.0.0.0[3]. This feature of proxy mode can be used to optimise low-bandwidth setups, see section 18.1.4 for more information.

On VLANs where the network interface is not assigned an IP address, the switch will automatically fall back to *proxy mode*, regardless of the querier mode setting.

Query interval: The switch can be configured to send out queries on intervals 12, 30, 70 and 150 seconds, default 12 sec. This interval is also used when timing out multicast to end devices that for some reason stop answering the

[2] The querier with the lowest IP address on each LAN is elected. Usually the gateway or multicast router.

[3] Address 0.0.0.0 is a special case and is never part of the IGMP querier election process, as clearly stated in the standard.

queries.

Multicast router timeout: When a multicast router, or a switch acting as IGMP querier, goes down, the lack of IGMP Query messages will cause a reelection to establish a new IGMP querier. This timeout can be configured via the CLI **”multicast-router-timeout”** setting. Default: 300 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 snooping leave latency (the time until the switch stops forwarding the associated multicast data) is within 2-3 times the configured *Query Interval*.

18.1.2 Multicast Router Ports

When IGMP snooping is enabled, the switch will learn on which ports there are interested receivers of a certain multicast group. It accomplishes this by listening to IGMP Report messages sent by all subscribers. Thus, the switch only forwards multicast on ports leading to members of each specific multicast group.

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 Queries are received, usually queries are sent by multicast routers, but also by IGMP snooping aware switches like WeOS
- FRNT Ring Coupling ports and Multi-link Dual-Homing ports: To provide fast fail-over of multicast traffic, FRNT Ring Coupling and Multi-link Dual-Homing uplinks (see chapter 15) are added to the list of multicast router ports. This is both done at the Ring Coupling nodes and Dual-Homing nodes, as well as on switches on the remote side of the uplink[4].

FRNT ring ports are no longer considered multicast router ports. The *Fast Reconnect* feature of FRNT is instead handled per multicast group: if a multicast receiver is located on a ring port, the other ring port is automatically added to the ATU MAC filter[5]. In case of ring breakage this practice ensures an extremely low reconfiguration time for multicast over FRNT.

[4] An exception is when connecting a Dual-Homing uplink to a non-FRNT switch, the fail-over of multicast traffic will instead occur on the next reception of an IGMP Report (if IGMP snooping is enabled). See also section 15.1.2.1.

[5] This can be seen using the CLI command **”show fdb”**

18.1.3 IGMP Fast Leave

WeOS IGMP snooping supports IGMP Leave by default and Fast Leave can be enabled on a per-port basis. The CLI **"igmp-fast-leave-ports"** setting allows using the keyword **"all"**, but Fast Leave is recommended only for access ports.

Example

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> no igmp-fast-leave-ports
example:/config/ip/#> igmp-fast-leave-ports eth 3,6
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
--------------------------------------------------------------------------------
 Static router ports        :  ---
 Dual homing/Coupling ports :  ---
 FRNT ports                 :  ---

 VID  Querier IP       Querier MAC        Port  Interval  Timeout
--------------------------------------------------------------------------------
   1  0.0.0.0          LOCAL

 VID  Multicast Group  Filtered MAC Addr  Active ports
--------------------------------------------------------------------------------
   1  239.255.255.250  01:00:5E:7F:FF:FA  6
   1  224.0.0.251      01:00:5E:00:00:FB  3, 6
   1  225.1.2.3        01:00:5E:01:02:03  6

--------------------------------------------------------------------------------
Total: 3 filters, max 2048, in 1 VLAN.

example:/#>
```

When an IGMP Leave is received on a port configured with Fast Leave it will issue a group specific query for the group being left and then immediately cut the multicast stream for that (multicast MAC) group. With Fast Leave disabled WeOS honors a grace period of, at most, two query intervals for the benefit of 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.

18.1.4 Low Bandwidth Networks

In low-bandwidth topologies, like FRNT over an SHDSL ring, you typically cannot afford wasting bandwidth on unwanted traffic. With the IGMP *Proxy Mode* and *Fast Leave* settings for IGMP snooping this can be avoided.

In the standard auto mode of IGMP all unknown multicast must be forwarded to the elected querier. But if there is no elected querier, or if all switches instead have proxy mode enabled, *unknown* multicast will be stopped before entering the low-bandwidth ring.

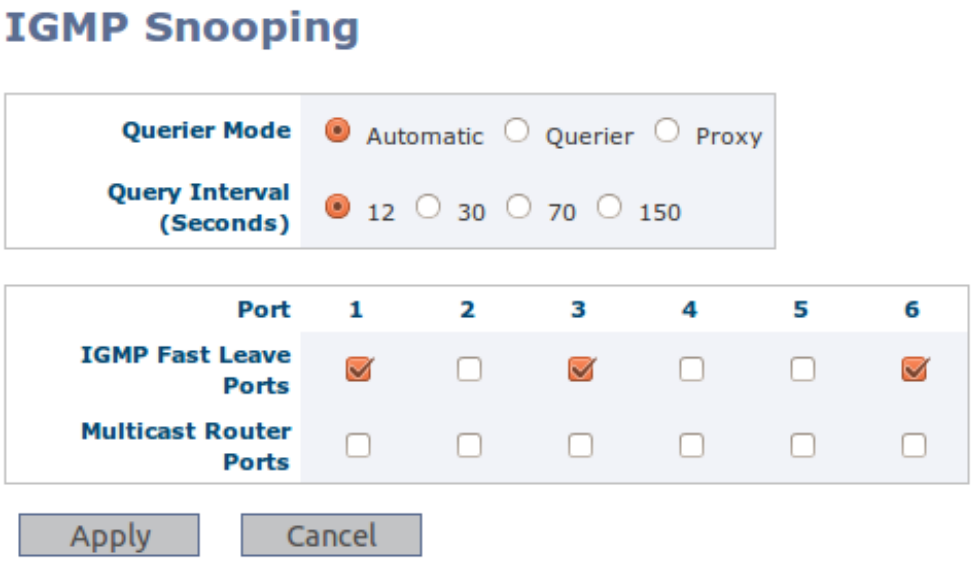
Only when a subscriber appears will the traffic be classified as *known* and forwarded on the ring to the receiver. By also enabling Fast Leave, on the access port towards the receiver, the multicast overhead can be kept to a near minimum.

18.2 Managing IGMP in the Web Interface

Global Configuration

Menu path: Configuration ⇒ IGMP

When entering the IGMP configuration page you will be presented with the global settings for IGMP snooping. Enabling or disabling IGMP is done per VLAN, see Section 13.



Querier Mode	The IGMP querier mode should have the same setting across all devices in the same LAN: **Automatic:** Automatic querier election. Recommended default **Querier:** Forced Querier mode, the device always sends IGMP queries, every Query Interval seconds **Proxy:** Fallback mode in which the switch normally does not initiate queries by itself, only forwards queries and reports. For more information on the modes, see Section 18.1.1
Query Interval	Number of seconds between each query.

Fast Leave Ports	Ports where multicast should not linger when receiving IGMP Leave.
Multicast Router Ports	Ports on which to forward all multicast. Useful if the switch fails to automatically detect a multicast router, or when you have a non-IGMP aware end devices.

Click **Apply** to save and apply the changes.

IGMP Status

Menu path: Configuration ⇒ IGMP Status

18.3 Managing IGMP in the CLI

The available general IP settings and monitoring commands are shown below.

Command	Default	Section
Configure General IGMP Snooping settings		
ip		Section 19.7.1
[no] igmp-mode <auto|querier|proxy>	auto	Section 18.3.1
[no] igmp-interval <12|30|70|150>	12 sec	Section 18.3.2
[no] igmp-fast-leave-ports <PORTLIST>	Disabled	Section 18.3.3
[no] multicast-router-ports <PORTLIST>	Disabled	Section 18.3.4
[no] multicast-router-timeout <1-2147483647>	300	Section 18.3.5
Per VLAN IGMP Snooping settings		
vlan <VID>		Section 13.4.6
[no] igmp	Enabled	Section 13.4.13
Show IGMP Snooping Status		
show ip igmp		Section 18.3.6

18.3.1 IGMP Querier Mode

Syntax `[no] igmp-mode <auto|querier|proxy>`

Context IP Configuration context

Usage Set IGMP Querier mode. In *auto mode* the device will participate in the querier election process (querier with lowest IP becomes querier). In *forced querier mode* the device will send IGMP queries even if there are other querier present with lower IP address. In *proxy mode* the device will act as an IGMP proxy, only initiating queries when no other eligible querier is available.

Note: if there is no IP address configured for an interface, the device will fall back to proxy mode regardless of the mode setting.

"no igmp-mode" resets the IGMP Querier mode to the default setting (**"auto"**).

Use **"show igmp-mode"** to view configured IGMP Querier mode (**"auto"**, **"querier"** or **"proxy"**).

Default `auto`

18.3.2 IGMP Querier Interval

Syntax `[no] igmp-interval <12|30|70|150>`

Context IP Configuration context

Usage Set IGMP Querier interval (seconds). The same interval is used for all interfaces.

”no igmp-interval” resets the IGMP Querier interval to the default setting, **”12”** sec.

Use **”show igmp-interval”** to view configured IGMP Querier interval.

Default 12 (sec)

18.3.3 IGMP Fast Leave

Syntax `[no] igmp-fast-leave-ports <PORTLIST>`

Context IP Configuration context

Usage Add or remove IGMP Fast Leave ports. For details, see section 18.1.3

”no igmp-fast-leave-ports <PORTLIST>” removes the specified port(s) and **”no igmp-fast-leave-ports”** all ports from the list of IGMP Fast Leave ports.

Use **”show igmp-fast-leave-ports”** to view configured multicast router ports.

Default Disabled

A **”PORTLIST”** is a comma separated list of port ranges without intermediate spaces, e.g., **”1/1-1/3,2/3”**.

18.3.4 Static Multicast Router Port Settings

Syntax `[no] multicast-router-ports <PORTLIST>`

Context IP Configuration context

Usage Add or remove multicast router ports. All (layer-2) multicast traffic will be forwarded on multicast router ports, see section 18.1.1.

”no multicast-router-port <PORTLIST>” removes the specified port(s) and **”no multicast-router-port”** all ports from the list of multicast router ports.

Use **”show multicast-router-port”** to view configured multicast router ports.

Default Disabled

A **”PORTLIST”** is a comma separated list of port ranges without intermediate spaces, e.g., **”1/1-1/3,2/3”**.

18.3.5 Multicast Router Timeout

Syntax `[no] multicast-router-timeout <1-2147483647>`

Context IP Configuration context

Usage Set the ”other IGMP Querier present” timeout (sec). The same interval is used for all interfaces.

Timeout for learned multicast router ports. With IGMP, and IGMP Snooping for switches, the elected querier is a critical component for successful operation. If it is lost, or suddenly gets a new IP address, another device must take over. This timeout adjusts the timeout before this device can take over.

”no multicast-router-timeout” resets the ”other IGMP Querier present” timeout to the default setting (**”300”**).

Use **”show multicast-router-timeout”** to view configured ”other IGMP Querier present” timeout.

The timeout should never be set lower than the IGMP Query Interval!

Default 300 (sec)

18.3.6 Show IGMP Snooping Status Information

Syntax `show ip igmp`

Context Admin Exec context

Usage Show IGMP snooping status information.

Default N/A

Chapter 19

General Interface and Network Settings

This chapter presents WeOS network interface settings, such as the interface IP address and common IP network settings, e.g., default gateway, DNS server and NTP server. Topics specific to various routing protocols and services, e.g., RIP, OSPF, VRRP, etc. are left to chapters 26-31.

Section 19.1 presents the general concepts of network interfaces in WeOS. It also covers the notion of *interface admin distance* and *management interface*, as well as IP related settings for DNS, NTP, etc. Section 19.4 and section 19.5 cover management of interfaces and general network settings via the Web interface. The corresponding CLI settings are divided into section 19.6, interface settings, and section 19.7, general network settings.

19.1 Overview

The table below summarises general interface and network features. Sections 19.2-19.3 contain further information on specific interface and network features.

Feature	Web	CLI	Description
Interface settings			
Enable/disable interface	X	X	Section 19.2.1
MAC address		X	Section 19.2.4

Continued on next page

Continued from previous page

Feature	Web	CLI	Description
Primary IP address	X	X	Section 19.2.5
Secondary IP addresses	X	X	Section 19.2.5
Netmask (Prefix Length)	X	X	Section 19.2.5
MTU	X	X	
Interface admin distance	X	X	Section 19.2.6
Management interface	X	X	Section 19.2.7
ICMP Redirect (sending)		X	Section 19.2.8
View interface configuration	X	X	
View interface status	X	X	
General network settings			
Default gateway	X	X	Section 19.3.1
Enable/disable unicast routing	X	X	”
DNS client support			
Set DNS server	X	X	Section 19.3.3
Dynamic DNS	X	X	”
DNS search path		X	”
DNS proxy server support		X	Section 19.3.4
NTP (NTP client)	X	X	Section 19.3.2
View general network config.	X	X	
View general network status	X	X	

19.2 Network interfaces

WeOS supports several kinds of network interfaces:

- *LAN/VLAN network interfaces:* A network interface is created for every VLAN configured on the switch (chapter 13).
- *PPP network interfaces:* (only for WeOS Extended) A network interface is created for every PPP instance configured on the switch (chapter 33). As of WeOS v4.17.1, PPP support is available over Ethernet/DSL ports using PPP over Ethernet (PPPoE), and over serial ports with or without external modem.

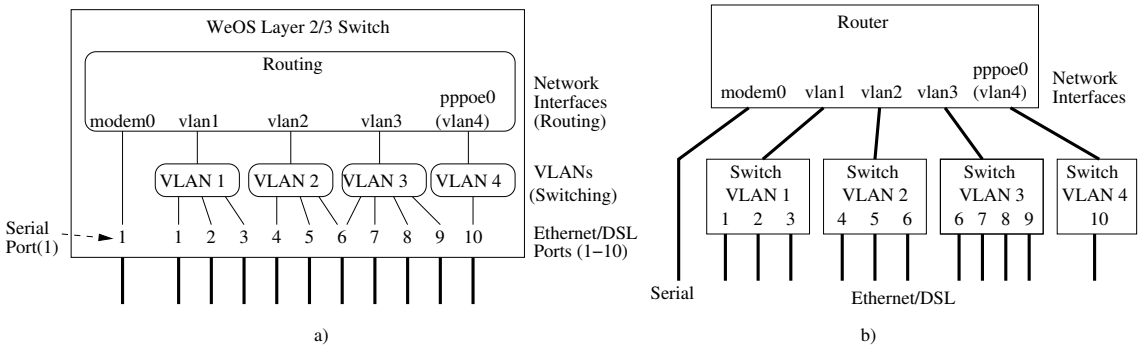


Figure 19.1: A network interface is associated with each VLAN, and VLANs are in turn associated with Ethernet (or DSL) ports as shown in figure a). Furthermore, when using PPPoE, a PPP network interface will be created and mapped on top of an associated VLAN interface, see *pppoe0* and *vlan4*. The routing switch can conceptually be seen as a router connecting a set of switches, as shown in figure b). In this sample setup, port 6 is shared by VLANs 2 and 3 (by use of VLAN tagging).

- *Loopback network interface:* The *loopback* interface *lo* is a logical network interface, which is always present. Its primary IP address cannot be changed, but it is possible to add *secondary* IP addresses, which can be useful in some situations,e.g., for OSPF (chapter 27).
- *GRE interfaces:* (only for WeOS Extended) For every configured GRE tunnel (chapter 34), an associated *GRE network interface* is created.
- *SSL interfaces:* (only for WeOS Extended) For every configured SSL VPN tunnel (chapter 36), an associated *SSL network interface* is created.
- *Blackhole interface:* WeOS has a hidden *blackhole* interface (**"null0"**), which can be used to avoid routing loops in case of incomplete subnetting, or to avoid that VPN traffic is forwarded towards the default gateway when the VPN tunnel is down. See section 26.1.4.3.

Fig. 19.1 shows how VLAN interfaces (*vlan1-vlan4*) are mapped to VLANs and ports, i.e., Ethernet and DSL ports. When using PPPoE, a PPP interface is created on top of a VLAN interface (see *pppoe0* and *vlan4* in fig. 19.1). *modem0* represents the network interface when running PPP over a serial port. The GRE and loopback interfaces are logical interfaces not directly associated with any physical port.

Every network interface can be assigned an IP(v4) address and netmask. By assigning an IP address to an interface, the operator is able to remotely manage the switch via that interface. Furthermore, if routing (IP forwarding) is enabled, the switch is able to *forward* packets *between* network interfaces. Section 19.3 gives a brief overview of WeOS routing features. chapter 26 gives a more detailed introduction to WeOS routing support, while chapters 27 and 28 covers dynamic routing with OSPF and RIP.

Note
IP forwarding is **not** available for products running software level WeOS Standard. However, it is possible to configure static (unicast) routes in WeOS Standard products as described in sections 26.2.1 (Web) and 19.7.3 (CLI).

19.2.1 Interface Operational Status (up/down)

For a network interface to get *operational* status *up*, it must be *enabled* in the configuration. But for some types of interfaces there may be additional critera to reach interface (operational) status *up*, as shown in the list below:

- *Loopback network interface:* The loopback interface *lo* is always *up*.
- *LAN/VLAN network interfaces:* For a VLAN interface to get status *up*, the interface must be *enabled* and its associated VLAN must also be up. In turn, the associated VLAN is up when that VLAN is *enabled*, and any of its associated ports have *link up* status. See chapter 13 for more information on VLANs.

  Note
  It is possible to circumvent the link status propagation property by configuring a LAN/VLAN network interface as always up (**”enable always”**, see section 19.6.2). 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.

- *PPP network interfaces:* (only for WeOS Extended) For a PPP interface to get status *up*, the PPP interface (and the associated PPP instance) must be enabled and successfully have carried out the PPP handshaking, including PPP authentication and IP address negotiation. For PPPoE, this implies that

the underlying VLAN interface must also be up. See chapter 33 for more information on PPP.

- *GRE interfaces:* (only for WeOS Extended) For a GRE interface to get status *up*, the GRE interface (and the associated GRE tunnel instance) must be enabled.

19.2.2 Interface Settings at Factory Default

WeOS products typically have all Ethernet and DSL ports mapped to VLAN 1 by factory default, and the network interface associated with VLAN 1 is named *vlan1*. The exception is Falcon, as described later in this section.hus by factory default, a WeOS unit has network interfaces *vlan1* and *lo* (logical ”loopback” interface).

The factory default settings for interfaces *vlan1* and *lo* are presented below. Most of the loopback settings are permanent (non-configurable).

Interface parameters	**Factory Default Setting (General)**	
	vlan1	**lo**
Administrative Mode	**Enabled**	**Enabled**
IP address	**Dynamic (DHCP)**	**Static 127.0.0.1**
Netmask	**(Dynamic)**	**255.0.0.0**
Secondary IP addresses	**192.168.2.200**	**Disabled**
Secondary Netmask	**255.255.255.0**	**N/A**
MAC address	**Auto**	**N/A**
MTU	**Auto (1500)**	**16436**
TCP-MSS	**Disabled**	**Disabled**
Admin Distance	**1**	**16**
Management Interface	**Enabled**[1]	**Disabled**

The interface *administrative distance* and *management interface* concepts are described in sections 19.2.6 and 19.2.7.

As stated earlier, Falcon has a different factory default settings than other WeOS products. The Ethernet ports are all mapped to VLAN 1 and interface *vlan1* as usual, but the Falcon xDSL port resides on a separate VLAN (VLAN 1006) and interface (*vlan1006*). The factory default settings for the associated interfaces are shown below. Most of the loopback interface (lo) settings are permanent (non-configurable).

Interface parameters	Factory Default Setting (Falcon)		
	vlan1	**vlan1006**	**lo**
Administrative Mode	**Enabled**	**Enabled**	**Enabled**
IP address	**Static 192.168.2.200**	**Dynamic (DHCP)**	**Static 127.0.0.1**
Netmask	**255.255.255.0**	**N/A**	**255.0.0.0**
Secondary IP addresses	**Disabled**	**Disabled**	**Disabled**
MAC address	**Auto**	**Auto**	**N/A**
MTU	**Auto (1500)**	**Auto (1500)**	**16436**
TCP-MSS	**Disabled**	**Disabled**	**Disabled**
Admin Distance	**16**	**1**	**16**
Management Interface	**Enabled**[1]	**Disabled**	**Disabled**

Note
On Falcon, the xDSL port associated with VLAN 1006 is intended to be used as the upstream ”WAN” port for Internet access. Interface *vlan1006* inherits its *admin distance* from the base interface, which by default is 1. For security reasons, management services are filtered out on *vlan1006* by default.

19.2.3 Creating Additional Network Interfaces

As shown in fig. 19.1 the switch will have one network interface for every VLAN defined on the switch. Thus, additional VLAN network interfaces can be created by creating new VLANs (see chapter 13). Similarly, a PPP network interface is created for every configured PPP instance, a GRE network interface is created for every configured GRE instance, etc.

The default settings for new VLAN and PPP (PPPoE and PPP over serial/modem) interfaces are shown in the table below, followed by a table presenting default settings for GRE and SSL VPN interfaces (PPP, GRE and SSL VPN interfaces are available for products running software level WeOS Extended).

It is not possible to create additional loopback interfaces. To have additional loopback IP addresses you can instead configure *secondary* IP addresses on the *lo* interface.

[1] At factory default, all management services **except Telnet** are *allowed* on interface *vlan1*.

Interface Parameters	Default Setting		
	vlan<VID>	**pppoe<ID>**	**modem<ID>**[2]
Administrative Mode	**Enabled**	**Enabled**	**Enabled**
IP address	**Static**[1] **Disabled**	**Dynamic**[3] **(IPCP)**	**Dynamic**[3] **(IPCP)**
Netmask	**Disabled**	**N/A**	**N/A**
MAC address	**Auto**	**N/A**	**N/A**
MTU	**Auto (1500)**	**1492**[4]	**Auto (1500)**
Admin Distance	**16**	**”Inherited”**	**16**
TCP-MSS	**Disabled**	**1412**	**Disabled**
Management Interface	**Enabled**[5]	**”Inherited”**	**Enabled**[4]

Interface Parameters	Default Setting	
	gre<ID>	**ssl<ID>**
Administrative Mode	**Enabled**	**Enabled**
IP address	**Static Disabled**	**Static Disabled**
Netmask	**Disabled**	**Disabled**
MAC address	**N/A**	**Auto**[6]
MTU	**1476**	**Auto (1500)**
Admin Distance	**16**	**16**
TCP-MSS	**Disabled**	**Disabled**
Management Interface	**Enabled**[5]	**Enabled**[5]

The *interface admin distance* and *management interface* concepts are described in sections 19.2.6 and 19.2.7.

VLAN network interfaces will be named according to the associated VLAN ID, e.g., the interface of VLAN 100 will be named *vlan100*. PPP, GRE and SSL interfaces will

[1] The exception is interface *vlan1* (VID 1). If *vlan1* does not exist, or if it is created without an address method defined, *vlan1* will default to acquire its address dynamically via DHCP.

[2] Interfaces for PPP over serial port (modem<ID>) are only available for products equipped a serial port.

[3] For PPP interfaces, the IP address assignment is handled by the PPP configuration, see section 33.1.7.

[4] When using PPPoE the default PPP interface MTU is 8 bytes less than the associated VLAN interface MTU, which is typically 1500 bytes.

[5] On new interfaces, all management services **except Telnet** are *allowed* by default.

[6] Only layer-2 SSL interfaces have MAC addresses. As of WeOS v4.17.1 the *auto* mode picks a random MAC address, however, this may change in the future WeOS releases.

be named according to their associated instance ID, e.g., *pppoe0* is the interface of PPPoE instance ”0”, *modem0* is the interface of serial/modem instance ”0”, and so on.

To communicate with the switch via a newly created interface, an IP address must be assigned to the interface, see section 19.2.5.

When creating a PPP instance of type PPPoE, the admin distance and management interface properties of the associated VLAN network interface are *inherited* by the PPP interface. This inheritance does not work in the reverse direction though, i.e., if the PPP instance is removed, the management and admin distance properties of the PPP interface are not passed back to the associated VLAN interface.

Note
With PPPoE, one must specify which VLAN interface to run PPPoE over, e.g., see interface *vlan4* in fig. 19.1. The resulting PPP interface will be said to ”own” the associated VLAN interface. As of WeOS v4.17.1, it is not possible to access a switch via ”owned” VLAN interfaces — access is only possible via the PPP interface.

19.2.4 VLAN Interface MAC address

Each VLAN network interface will be assigned a MAC address (also known as the Ethernet address, the link address, the hardware address, or the IEEE EUI-48 address).

In WeOS products, each *Ethernet port* (or DSL port) is assigned a MAC address, and a *VLAN interface* will by default inherit its MAC address from one of its member ports. It is also possible to manually configure a MAC address for a VLAN interface.

The algorithm to assign VLAN interface MAC address uses the following preference order:

1. If the interface has been configured with a custom MAC address, use that address as the interface MAC address.
2. If the VLAN has one or more ports assigned *untagged*, use the MAC address of the ”lowest” untagged port as the interface MAC address.
3. If the port has one or more ports assigned *tagged*, use the MAC address of the ”lowest” tagged port as the interface MAC address.

4. Use the MAC address of the *channel* (section 13.1.6) associated with the VLAN.

Consider the sample configuration in fig. 19.1. When all interfaces get their MAC address automatically, interface *vlan1* inherits the MAC address of port 1, *vlan2* inherits its MAC from port 4, *vlan3* from port 7 (assuming port 6 is tagged on VLAN 3), and interface *vlan4* from port 10.

Note
For the automatic MAC assignment methods (steps 2-4 above), the MAC address may change when the set of ports associated with the VLAN changes. When this happens, the WeOS device will submit a *gratuitous ARP* to update stale ARP caches in neighbour nodes.

For VLANs created dynamically (section 13.1.7), no associated network interface is created. Thus, for such VLANs no interface MAC address is needed.

19.2.5 IP address settings

Each network interface can be assigned a *primary* IP address and up to 8 *secondary* IP addresses, this is sometimes referred to as multinetting, but can also be another address on the same subnet as the primary address. The primary IP address can either be statically or dynamically assigned, depending on the address method configured for the interface (**”inet static”** or **”inet dynamic”**).

The secondary IP addresses can only be statically configured, but can be used with both static and dynamic primary address.

Options for configuring the primary address for different interface types:

- *VLAN interfaces:* The primary IP address of a VLAN interface can be configured statically, or configured to acquire its address dynamically (DHCP). It is also possible to have a VLAN interface without any IP address.
- *PPP interfaces:* For PPP interfaces the address setting is set to *dynamic*, but the actual IP address assignment is handled by the PPP configuration (IPCP), see section 33.1.7.
- *GRE interfaces:* For GRE interfaces, the primary IP address can only be configured statically.
- *Loopback interface (lo):* The primary IP address of the loopback interface (lo) is permanently set to 127.0.0.1.

In the example below, interface vlan2 is assigned a static primary IP address (**”192.168.11.1”**) and an additional secondary IP address (**”192.168.12.1”**), i.e., multinetting is used. Here the IP address *netmask* (255.255.255.0) for both addresses has been written in *prefix length* format (’/24’).

Example

```
example:/config/#> interface vlan2
example:/config/iface-vlan2/#> inet static
example:/config/iface-vlan2/#> address 192.168.11.1/24
example:/config/iface-vlan2/#> address 192.168.12.1/24 secondary
example:/config/iface-vlan2/#> end
example:/config/#>
```

Interfaces with dynamic address assignment use DHCP to acquire their IP address from a DHCP server, or IPCP for PPP interfaces. If no DHCP server is present, the interface will generally end up without any IP address. The exception is the interface with best *admin distance*, which will always acquire a *link-local* IP address. The interface *admin distance* and *link-local address* concepts are further described in section 19.2.6.

19.2.6 Dynamic Address Assignment and Admin Distance

An interface can be configured to retrieve its IP settings dynamically via DHCP (VLAN interfaces) or IPCP (PPP interfaces). In addition to interface settings such as IP address and netmask, the switch can also acquire general network settings such as default gateway and DNS server(s) from the DHCP server, or via PPP. More information on general network settings is given in section 19.3.

Multiple network interfaces can acquire their IP settings dynamically, but only one default route, one set of DNS servers, one domain search path and one set of NTP servers can be active at one time in the system. WeOS handles this using a set of precedence rules. When setting up a device with automatic fail-over between multiple upstream connections these rules are important to be aware of.

Prior to WeOS 4.14.0 the precedence was handled by something called the *primary interface*. However, this has been replaced with the concept of *administrative distance* for both static routes and interfaces. Administrative distance is also available to dynamic routing protocols such as OSPF and RIP, see chapters 27 and 28, respectively.

The admin distance is a priority value ranging from 1–255, where 255 is treated

as infinite distance. E.g., a static route installed with distance 255 is guaranteed to never be activated. WeOS makes use of this in fail-over scenarios with multiple upstream interfaces and *ping triggers*.

The following list summarises the rules for dynamically retrieved settings and how they are applied to the system.

- Dynamic IP address and netmask are always set on the interface, without affecting any secondary IP address configured statically.
- Default route, domain search path, and DNS servers are always saved, but not necessarily installed.
- Default routes are installed with the configured interface *admin distance* and the 'best' route is set as the active default route in the system.
- The interface with the best (lowest) distance wins. If that interface goes down, the default route of the next best interface distance is activated.
- If there are multiple interfaces with lowest distance, the system will select one of those interface as 'best'. A user wishing to have full control of what interface is 'best' should assign a unique admin distance per interface.

  Hint

  Assign unique admin distance values to your interfaces.

- A ping trigger can be associated with the interface distance setting. When it signals loss of connectivity, the distance of the associated default route is raised to infinity (255).
- When the best upstream interface has been established, *domain search path*, *domain name servers* (DNS) and *network time protocol servers* (NTP) are set from that source, unless there exist statically configured settings.
- Statically configured DNS, domain and NTP always win, regardless of any distance.
- *NTP server* may be acquired from a DHCP server when no *NTP server* has been configured statically (see section 19.3.2).
- The 'primary' setting in WeOS prior to 4.14.0 is converted to a distance value: the primary interface gets distance 1 and all other interfaces get the default distance, 16.

- Static configuration of routes, including the default route, competes with routes learned on DHCP client interfaces *as well as* routes from dynamic routing protocols. An obvious benefit of this is to have a statically configured fallback default route that is activated automatically when no better route is available. This is often referred to as a *floating static route*.

- The default gateway setting **"ip default-gateway <IPADDR>"** is deprecated. Setting up a default gateway in the CLI will install a static default route with distance 1. Use the route command instead (see section 19.7.3, notice the new keyword 'default' for '0.0.0.0/0'

Example

```
example:/config/#>  ip route default 192.168.11.1 10
```

In the example below interface *vlan3* is configured to acquire its IP address via DHCP with distance 1. The system default interface, *vlan1* is moved to distance 200 and a *floating static route* to a gateway reachable via *vlan1* is setup with a distance of 200 as well. The default route acquired by DHCP on *vlan3* will be installed with distance 1 and will be made the active route.

Example

```
example:/config/#>  interface vlan1
example:/config/iface-vlan1/#>  address 192.168.11.2/24
example:/config/iface-vlan1/#>  distance 200
example:/config/iface-vlan1/#>  end
example:/config/#>  interface vlan3
example:/config/iface-vlan3/#>  inet dhcp
example:/config/iface-vlan3/#>  distance 1
example:/config/iface-vlan3/#>  end
example:/config/#>  ip default 192.168.11.1 200
example:/config/#>
```

If no DHCP server is present, an interface configured to use DHCP client for address assignment will end up without any IP address. The exception is the DHCP client interface with the best distance, which will always acquire a *link-local* IP address in the range 169.254.0.0/16 in addition to any address assigned via DHCP. The link-local address is taken from the 169.254.0.0/16 range such that address collisions are avoided and that an interface is likely to get the same address every time it comes up.

19.2.7 Management Interface

The operator can manage the switch remotely in several ways: Web (HTTP/HTTPS), SSH, Telnet, SNMP and WeConfig (using the IPConfig service). As described in chapter 7 it is possible to completely disable individual management services, however, there are situations when an operator may wish to limit management access to a certain network interface or VLAN. WeOS provides a powerful mechanism for controlling access to management services on a *per interface* basis. An interface where one or more management services are enabled is referred to as a *management interface*.

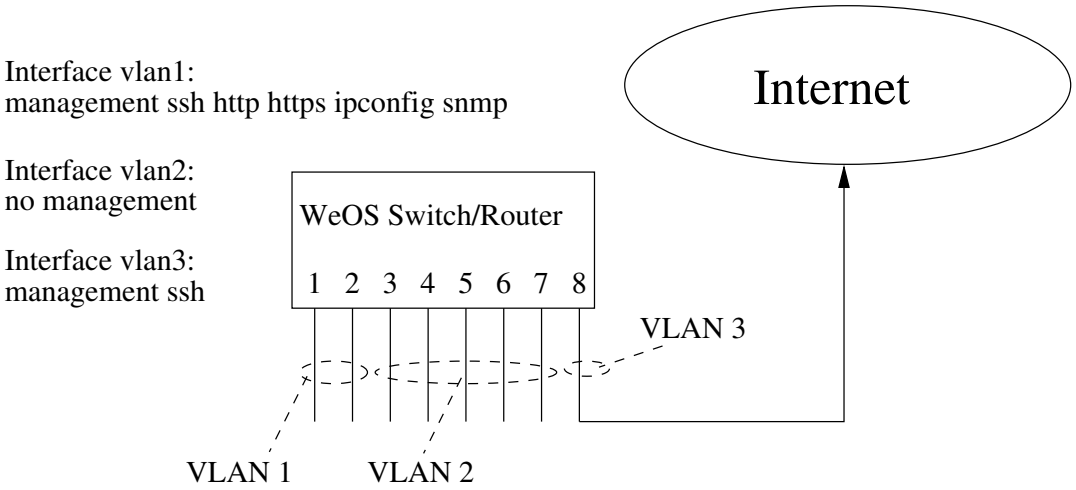


Figure 19.2: Management service filtering per interface.

Fig. 19.2 gives an example on the flexibility by the *management interface* feature in WeOS. The switch has three network interfaces – one for each VLAN. VLAN 1 is the administrator's local LAN with full management capabilities. VLAN 2 is another local LAN for regular *in-house* users, from which no management is allowed. VLAN 3 is used as the upstream connection; in this example SSH is allowed on this network interface, while other services are disabled.

Note

WeOS use the term ”management interface” rather than ”management VLAN”. This is because management is not be limited only to VLAN network interfaces. For example, the operator may wish to manage a switch remotely through a modem connection (i.e., a PPP interface on a switch equipped with a serial port).
The equivalent of a management VLAN can be setup by filtering out management services on all interfaces but the network interfaces associated with that VLAN.

The default behaviour aims to avoid unintentional loss of management access to the switch. Sections 19.2.2 and 19.2.3 describe the default settings for network interfaces, settings at factory default as well as settings for newly created interfaces[1].

Warning

Access to management services on all interfaces is convenient, but may pose a security risk if connected to an untrusted network. By default the device is (typically) manageable via all network interfaces, it is therefore strongly recommended that the operator use the interface management filter to only allow a select set of services, or none, on untrusted networks.
E.g., for an interface connected to the public Internet one should consider disallowing all management services, or perhaps only allow management via secure protocols such as SSH and HTTPS.
Also crucial to cyber security is the password policy and setting up adequately secure passwords when providing management access via an interface connected to an untrusted/public network.

A word of caution is in order, it is entirely possible to get locked out of a device when setting up the management service filter. For devices with a console port this may not be a problem, for others this is the time to be reminded about the ”crossed–cables factory reset” (section 7.1.3.3).

However, WeOS actually does implement some safeguards to prevent against locking yourself out. If all management is disabled on all interfaces, the system falls back to enabling secure shell, SSH, access on interface *vlan1*. Furthermore, if *Web* (for instance) is the only management service allowed on any interface,

[1] As mentioned in section 19.2.2 factory default on Falcon switches include a separate VLAN for the xDSL port, and the associated interface (*vlan1006*) has management services disallowed for security purposes.

but the Web server has been disabled, the same fall-back solution is triggered.

Hint

From security standpoint it is recommended to separate the management interface from the upstream WAN interfaces, but also from interface *vlan1* since it is also the fallback interface in WeOS.
E.g., use interface *vlan1* as a LAN interface, with high interface distance, and interface *vlan2* as the upstream WAN interface, with distance 1.
If you, e.g., remove the unrelated VLAN 3 without assigning its ports to any other VLAN, then WeOS will automatically place them as untagged in VLAN 1, the default/fallback VLAN. In most cases you do not want those ports ending up on the upstream side . . .

19.2.8 Control Sending of ICMP Redirect

A WeOS router is able to send ICMP Redirect messages when it receives IP packets which could have been routed more optimal. The topology shown in fig. 19.3 can be used to illustrate a situation where ICMP Redirect is useful.

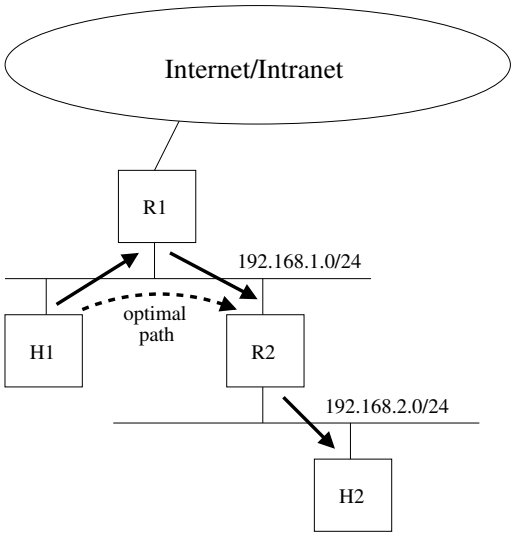


Figure 19.3: Example where ICMP Redirect is useful.

Assume that *Host 1* (H1) wishes to communicate with *Host 2*, and that H1 (only) knows about its local subnet (192.168.1.0/24) and its default route pointing to

Router 1 (R1). In this case all packets from H1 to H2 will go to R1, which in turn sends them back on the same LAN to R2. The packets will be sent twice over the LAN, resulting in waste of network capacity and increased delay. By enabling sending of *ICMP Redirect* on R1, the router will send ICMP Redirect messages to H1, informing the host that it can route packets directly to R2. If the host accepts *ICMP Redirect* messages, it will update its routing table and forward future packets to H2 directly via R2.

In WeOS, the sending of ICMP Redirect messages can be enabled/disabled per network interface. By default sending ICMP Redirect messages is enabled.

Note
A WeOS unit does not accept *incoming* ICMP redirect messages.

19.3 General IP settings

The general IP settings provided fall into three categories:

- Routing: Configuration of default gateway, static IP routes, and ability to enable/disable IP forwarding (IP forwarding is available for products running software level WeOS Extended).
- IGMP: Configuration of IGMP snooping parameters such as *querier mode*, *query interval* and static multicast router ports. (IGMP snooping is covered in chapter 18.)
- Services: Examples of include settings for DNS and DDNS servers, domain search path, and NTP client settings.

19.3.1 Routing

To manage the WeOS unit remotely, it should generally be configured with a default gateway. It is also possible to configure additional, static IP routes.

WeOS units running software level WeOS Extended are capable of *IP forwarding*, i.e., it can *route* incoming IP packets to other interfaces and IP subnets. For unicast, both static routing and dynamic routing (RIP and OSPF) are supported. Units running WeOS Extended act as routers by default, i.e., IP forwarding is *enabled* in the factory default setting.

WeOS units are also able to route IP multicast (static multicast routing). In addition, WeOS devices can efficiently distribute IP multicast packets in a switched LAN by use of IGMP snooping.

This chapter only covers rudimentary routing features, such as enabling/disabling IP forwarding and configuring a default gateway. WeOS routing support is described further in chapters 26-30. IGMP snooping support is covered in chapter 18.

19.3.2 Time synchronisation via NTP Server

The switch can synchronise its clock with an external time server via the NTP protocol. Up to 8 NTP servers can be configured, but it is also possible to acquire NTP server(s) via DHCP when no static NTP server is configured (see section 19.2.6).

19.3.3 DNS client - setting DNS server and dynamic DNS

Most users find it is easier to refer to Internet hosts using *domain names*, e.g., http://www.example.com, than using IP addresses, e.g., http://93.184.216.119. To facilitate the use of the Domain Name System (DNS), WeOS supports configuration of up to two DNS server entries. It is also possible to configure a *domain search path*. These settings can also be acquired dynamically via DHCP or PPP (see section 19.2.6).

Use of domain names on a switch can be convenient, e.g., when setting up ping triggers, VPN peers or when troubleshooting with tools such as *ping* or *traceroute*, see section 7.1.10.

It is also convenient to communicate *with* the switch using domain names. When the switch acquires its IP address dynamically (via DHCP or PPP), maintaining the DNS server entry is cumbersome. To manage this situation, WeOS includes support for dynamic DNS (DDNS). With DDNS enabled, the switch will update its DNS server entry automatically when acquiring a new IP address.

Examples of supported DDNS providers are:

- **dyndns:** http://www.dyndns.org,
- **freedns:** http://freedns.afraid.org
- **no-ip:** http://www.no-ip.com

See the CLI or Web online help for a more up-to-date list.

19.3.4 Proxy DNS server

WeOS units are able to act as DNS proxy servers (enabled by default). When enabled, the unit will act as a DNS server and respond to DNS queries for *known hosts*:

- either statically added by the **”host”** (section 19.7.8), see also the **”show ip host”** (section 19.7.28) command, or
- hosts for which this unit acts as DHCP server (chapter 22), see also the **”show dhcp-clients”** (section 22.3.20) command .

As DNS proxy, the WeOS the unit will also act as a caching DNS forwarder; DNS queries of unknown hosts are forwarded to the unit’s own DNS server (see the

”show ip name-server” command described in section 19.7.26), and the answer is cached for fast response of subsequent requests for the same host.

When proxy DNS server is enabled on a WeOS unit, it will accept incoming DNS packets on all its interfaces.

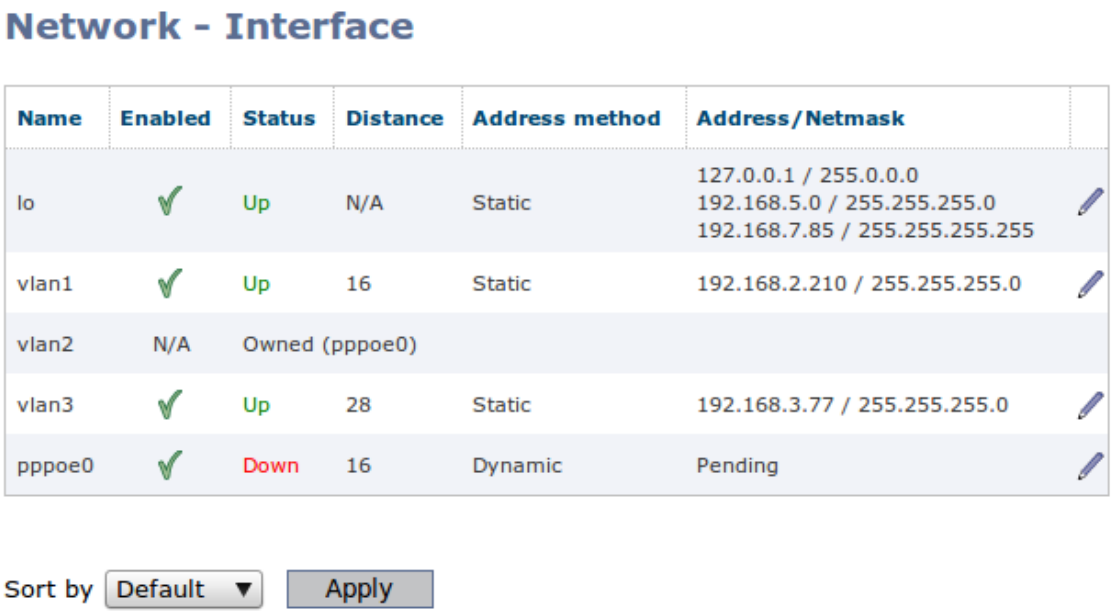
Hint

For security purposes you may wish to avoid accepting DNS packets on some interfaces, e.g., your upstream interface towards the Internet. To block such request you are recommended to configure appropriate *deny* filter rules, e.g., **”filter deny in vlan1 dport 53 proto udp”** and **”filter deny in vlan1 dport 53 proto tcp”** to block incoming DNS request on interface *vlan1*. For more details on the WeOS firewall, see chapter 31.
Alternatively, disable the DNS proxy service.
For WeOS products running software level WeOS Standard attached directly to the Internet, it is recommended to disable the DNS proxy service.

19.4 Managing network interfaces via the web interface

This section covers network interface settings of the unit. Settings related to IGMP snooping is described in section 18.2.

Menu path: Configuration ⇒ Network (IP) ⇒ Interface

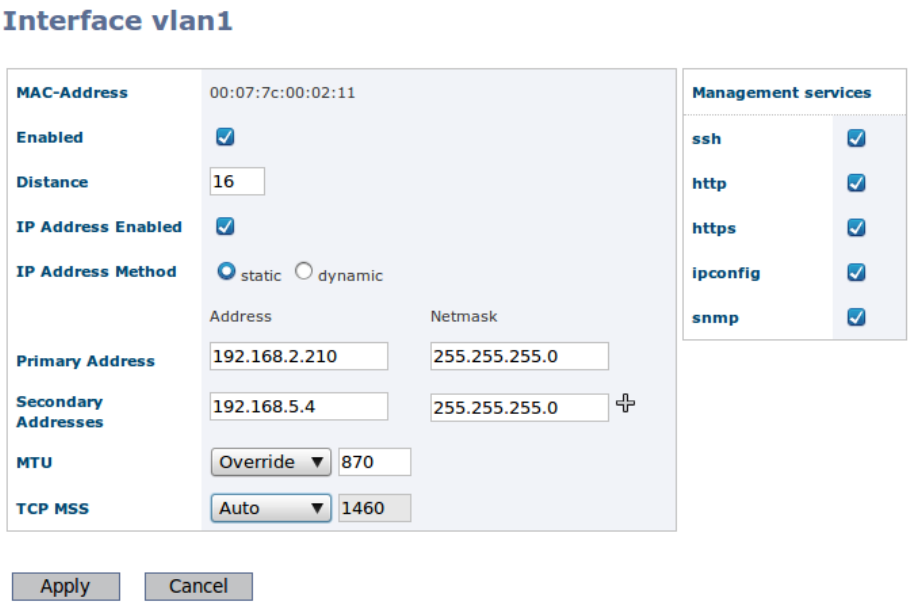


Name	A unique identifier for the interface. Automatically generated from VLAN/PPP/GRE/SSH identifier when the VLAN/PPP/GRE/SSH/ instance is created. *lo* is the *loopback* interface. (PPP, GRE and SSH interfaces are available for WeOS Extended.)
Enabled	Shows whether the interface is enabled or disabled. A green checkmark means the interface is enabled, and a dash means it is disabled.
Status	The status of the interface, *Up* or *Down*.
Distance	The administrative distance value used for routes acquired on this interface. Route selection is based on this number. A lower value indicates a more preferred route.

Continued on next page

	Continued from previous page
Address method	The IPv4 address assignment method used for the interface: *Static* means the IPv4 address is configured manually, *Dynamic* means the address is acquired automatically via DHCP (for VLAN interfaces) or is part of the PPP configuration (for PPP interfaces), and *Disabled* means IPv4 address assignment is disabled on the interface.
Address/ Netmask	The IPv4 address, and its associated netmask, assigned to the interface. The netmask identifies what IP addresses are located on the same subnet. Displays configured IP address, when address method *Static* is used. Displays the dynamically assigned address, or *Pending* if *Dynamic* address method is set. Text *Disabled* is shown if IP address assignment is disabled. Text *Owned* is shown when there is a PPPoE interface associated with that VLAN interface. Secondary addresses assigned to the interface are also listed.
Edit	Click this icon to edit the interface.
Sort by	The list of interfaces may be sorted either in a default sort order, or by the distance value. Select desired sort order and press apply button.

When clicking the *Edit* icon for an interface you will be presented to its associated edit page.



Note: The user support to only display relevant input fields is only available when using a JavaScript enabled browser.

MAC Address	(Only applicable for VLAN interfaces.) The media access control (MAC) address is used for controlling the communication on OSI layer 2. Shows the MAC-address associated to this interface.
Enabled	The interface may be activated or deactivated by the Enabled setting. Click the check-box to activate/deactivate the interface.
Distance	The administrative distance value used for routes acquired on this interface. Route selection is based on this number. A lower value indicates a more preferred route.
IP Address Enabled	(Only applicable for VLAN interfaces.) When disabling the IP address, traffic may not be sent to the switch from units connected to the VLAN associated with this interface. The address may be disabled to e.g. prevent administration access from specific VLANs. The IP address mode field, and for static address mode the IP address and netmask fields, will not be visible unless this box has been checked.
IP Address Mode	Choose *Static* to manually configure IP address and netmask or *Dynamic* to let the unit query a DHCP server for address information.(PPP interfaces can only be specified for dynamic IP address, but the actual IP address assignment is handled by the PPP configuration, see section 33.2.)
Primary Address	The IPv4 address, and its associated netmask, assigned to the interface. The netmask identifies what IP addresses are located on the same subnet. Not applicable for PPP and loopback interfaces. These fields will only be visible if static IP Address Mode has been selected.
Secondary Addresses	Address and netmask for the secondary IPv4-addresses associated to this interface. These fields will only be visible if *IP Address Enable* has been checked. Up to eight secondary IPv4-addresses may be associated to the interface. Click the **`plus`** sign to add new lines. Click the  to delete a row.

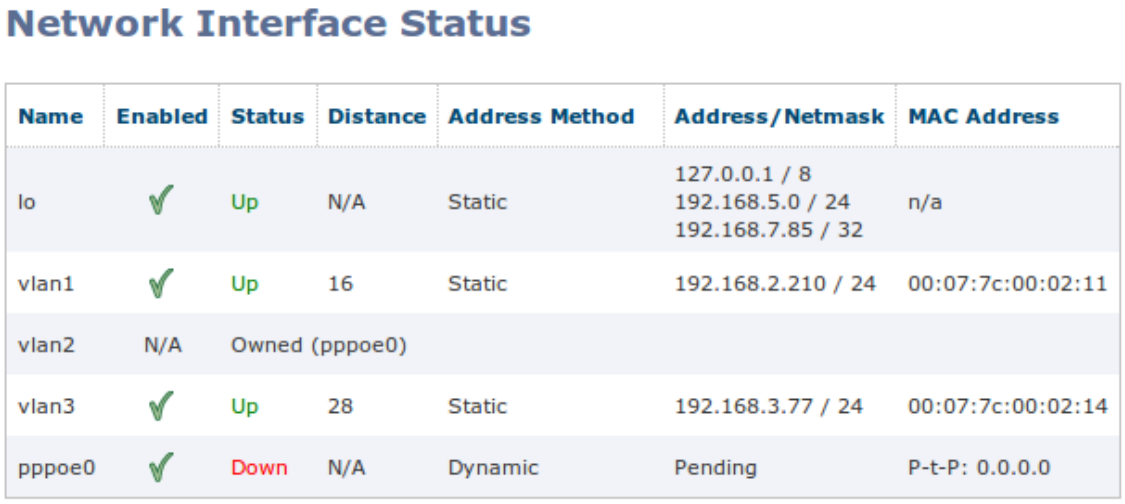
Continued on next page

	Continued from previous page
MTU	This option is not available for all interface types. **Override** Set a non-default MTU size by entering an override value. **Auto** The interface will let its MTU be the default MTU of the associated link type.
TCP MSS	This option is not available for all interface types. **Override** Limit TCP-MSS to the given number of bytes. **Auto** Lets the TCP-MSS depend on the MTU of the interface This will work fine for typical TCP connections, but is not likely to work over IPsec tunnels or when additional IP header options are in use. **Disabled** Disables TCP-MSS clamping.
Management Services	Check the boxes for the services that should be accessible from this interface.

Click the **Apply** button to save and apply the changes.

19.4.1 Interface Status

Menu path: Status ⇒ Interface



Name	A unique identifier for the interface. Automatically generated from VLAN/PPP/GRE/SSH identifier when the VLAN/PPP/GRE/SSH/ instance is created. *lo* is the *loopback* interface. (PPP, GRE and SSH interfaces are available for WeOS Extended.)
Enabled	Shows whether the interface is enabled or disabled. A green check-mark means the interface is enabled, and a dash means it is disabled.
Status	The status of the interface, *Up* or *Down*.Text *Owned* is shown when there is a PPPoE interface associated with the VLAN interface. The *owner* is also displayed within parenthesis.
Distance	The administrative distance value used for routes acquired on this interface. Route selection is based on this number. A lower value indicates a more preferred route.
Address method	The IPv4 address assignment method used for the interface: *Static* means the IPv4 address is configured manually, *Dynamic* means the address is acquired automatically via DHCP (for VLAN interfaces) or is part of the PPP configuration (for PPP interfaces), and *Disabled* means IPv4 address assignment is disabled on the interface.

Continued on next page

	Continued from previous page
Address/ Netmask	The IPv4 address, and its associated netmask, assigned to the interface. The netmask identifies what IP addresses are located on the same subnet. Displays configured IP address, when address method *Static* is used. Displays the dynamically assigned address, or *Pending* if *Dynamic* address method is set. Text *Disabled* is shown if IP address assignment is disabled. Secondary addresses assigned to the interface are also listed.
Sort by	The list of interfaces may be sorted either in a default sort order, or by the distance value. Select desired sort order and press apply button.

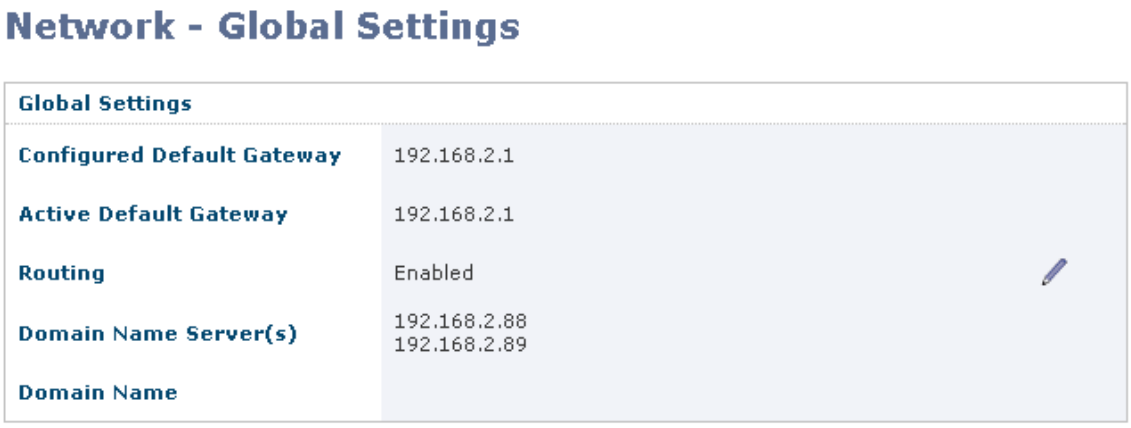
19.5 Managing general IP settings via the web interface

This section covers general IP related settings of the unit. Settings related to IGMP snooping are described in section 18.2.

19.5.1 Global Network Settings Overview

Menu path: Configuration ⇒ Network(IP) ⇒ Global settings

When entering the Network(IP) configuration page you will be presented to a list of common network settings.



Global Settings (Default Gateway, Routing and DNS servers)

Configured Default Gateway	Statically configured default gateway of the unit. This is the IP address of the gateway to send packages to when no more specific route can be found in the routing table. *Empty field* indicates that no default gateway address has been statically configured.
Active Default Gateway	The currently active default gateway in use. *N/A* indicates that no default gateway is in active use. A default gateway cannot be active if no route to the default gateway is available.

Continued on next page

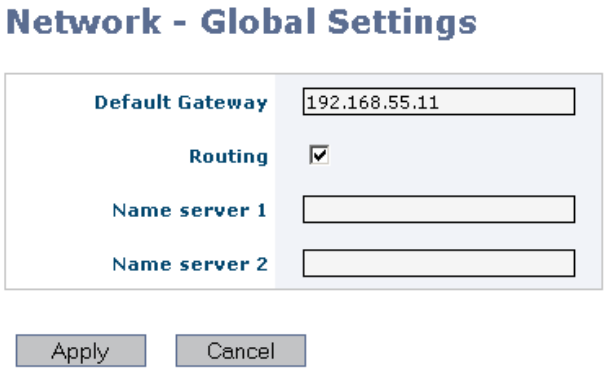
	Continued from previous page
Routing	(only for WeOS Extended) Routing, also known as IP-forwarding, allows traffic to flow between VLANs. Use the firewall to protect VLANs from unwanted traffic. Texts *Enabled* and *Disabled* shows routing status.
Domain Name Server(s)	List manually configured DNS servers. An empty field indicates that no DNS server has been manually configured.
 **Edit**	Click this icon to edit ”this part” of the global settings.

These settings are described further in section 19.5.2.

To change the settings for a specific Interface click the associated edit icon which will take you to the interface settings edit page. Interface settings are described further in section 19.4.

19.5.2 Edit Common Network Settings

Menu path: Configuration ⇒ Network (IP) ⇒ Global settings ⇒ 



Default Gateway	Statically configured default gateway of the unit. This is the IP address of the gateway to send packages to when no more specific route can be found in the routing table. Leave empty if no default gateway is desired.
Routing	(only for WeOS Extended) Routing, also known as IP-forwarding, allows traffic to flow between VLANs. Use the firewall to protect VLANs from unwanted traffic. Check this box to enable routing, uncheck to disable.
Name server 1	IP address of (primary) DNS server.
Name server 2	IP address of (secondary) DNS server.

Click the **`Apply`** button to save and apply the changes.

19.5.3 NTP client

Menu path: Configuration ⇒ System ⇒ Date & Time

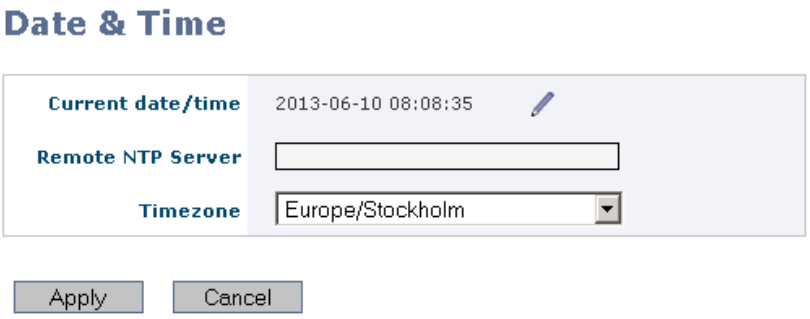


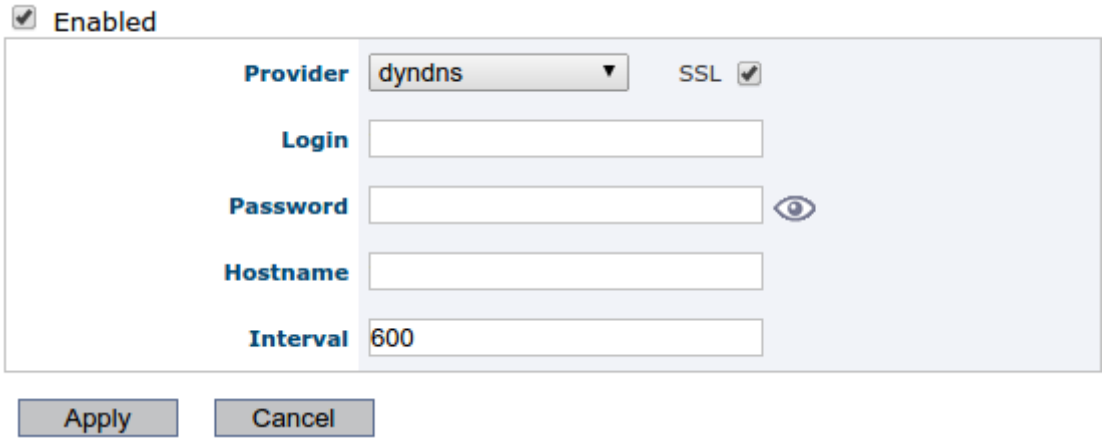
Figure 19.4: Switch date and time settings, NTP client

Current Date/Time	Shows current date and time. Click the icon to manually set date/time .
Remote NTP Server	The IP address of a time server to be used to keep the units calendar time synchronised. Leave empty if you do not want to use a time server, or if NTP server should be acquired via DHCP or PPP.
Timezone	Select a timezone region to get adjusted local time.

19.5.4 DDNS settings

Menu path: Configuration ⇒ Network (IP) ⇒ DDNS

Dynamic DNS (DDNS) provider settings



Enabled	Check this box to enable Dynamic DNS, uncheck to disable.
Provider	Select DDNS provider. Example of supported providers: **dyndns** http://www.dyndns.org, **freedns** http://freedns.afraid.org, and **no-ip** http://www.no-ip.com See the online help for more.
SSL	Check this box if your DDNS provider supports HTTPS updates.
Login	Set login *username* for the account at your DDNS provider
Password	Set login *password* for the account at your DDNS provider
Hostname	Set the DNS hostname, i.e., registered domain name which should map to the IP address of this your switch. When selecting `freedns`, the domain name must be followed by a hash value (**"HOSTNAME,HASH"**); the *hash* is provided by FreeDNS).
Interval	Set the interval by which DDNS verifies that the IP address mapping at your DDNS provider matches the IP address of your switch. Maximum 10 days (864000 seconds).

Click the **Apply** button to save and apply the changes.

19.6 Managing network interfaces via the CLI

The available interface settings and monitoring commands are shown in the table below:

Command	Default	Section
iface <IFNAME> inet <static|dynamic>	*Differs*[1]	Sec. 19.6.1
[no] enable [always]	Enabled	Sec. 19.6.2
[no] address <ADDRESS/LEN| ADDRESS NETMASK> [secondary]	Disabled	Sec. 19.6.3
[no] primary	DEPRECATED	Sec. 19.6.4
[no] distance <1-255>	16	Sec. 19.6.5
[no] management <[ssh] [telnet] [http] [https] [ipconfig] [snmp] | all>	Enabled[2]	Sec. 19.6.6
[no] mtu <68-1500>	*Differs*[1]	Sec. 19.6.8
[no] tcp-mss <40-1460|auto>	*Differs*[1]	Sec. 19.6.9
[no] redirect	Enabled	Sec. 19.6.10
Only for VLAN interfaces		
[no] mac <X:X:X:X:X:X>	Auto	Sec. 19.6.7
Show interface status		
show iface [IFNAME]		Sec. 19.6.11

19.6.1 Manage Network Interfaces

Syntax `iface <IFNAME> inet <static|dynamic>`

Context Global Configuration context

Usage Enter Interface Configuration context, and specify IP address assignment method.

- **”`static`”** means static IP address assignment. The IP address is configured via the **”`[no] address <ADDRESS/LEN|ADDRESS NETMASK>`”** com-

[1] Some interface ”native” default settings depend on the interface type, see section 19.2.3. Section 19.2.2 provides information on ”factory” default settings.

[2] By default, all management services **except Telnet** are *allowed* on newly created VLAN and PPP interfaces.

mand, see section 19.6.3.

- If **”dynamic”** is selected, the switch attempts to acquire its address via DHCP (VLAN interfaces) or IPCP (PPP interfaces). If no DHCP server is available, the interface will generally end up without an IP address. The exception is the interface with best *admin distance*, which will get a *link-local* IPv4 address if it fails to get an address via DHCP.

Use **”show iface”** to show network interface configuration information of all interfaces. Use **”show iface [IFNAME]”** to show configuration information for a specific interface (also available as **”show”** command within the Interface Configuration of that specific interface).

Default values **”static”** for VLAN and GRE interfaces, and **”dynamic”** for PPP interfaces. For VLAN interfaces there is one exception – If *vlan1* does not exist, or if it is created without an address method defined, *vlan1* will default to acquire its address dynamically via DHCP.

19.6.2 Interface Administrative Mode (Enabled or Not Enabled)

Syntax `[no] enable [always]`

Context Interface Configuration context

Usage Bring interface up/down. Note, even if an interface is configured administratively *up*, its operational status may still be *down* if the associated VLAN (or PPP instance) is not up.

Use command **”enable”** to configure an interface as *up*, and **”no enable”** to configure the interface as down.

On LAN/VLAN interfaces, it is possible to circumvent the link status propagation property by configuring an interface as always up (**”enable always”**). However, disabling link status propagation may significally 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.

> **Note**
> An interface configured as *always up* will in SNMP report *ifOperStatus* ”testing(3)”.

Use **”show enable”** to show whether this interface is configured as administratively enabled (up) or disabled (down).

Default values Enabled (**”enable”**)

19.6.3 IP Addresses (primary and secondary)

Syntax `[no] address <ADDRESS/LEN|ADDRESS NETMASK> [secondary]`

Context Interface Configuration context

Usage Set static IP address and netmask for an interface.

When *static address assignment* is chosen (**”inet static”**, see section 19.6.1), the **”address”** command can be used to the *primary* IP address of the interface, as well as *secondary* IP addresses of the interface (using the **”secondary”**) keyword.

When *dynamic address assignment* is chosen (**”inet dynamic”**, see section 19.6.1), the **”address”** command is limited to assign *secondary* IP addresses.

Up to 8 secondary addresses can be configured for an interface.

It is possible to specify the boundary between the *network part* and the *host specific part* of the IP address either as a prefix length (e.g. **”address 192.168.0.1/24”**) or as a regular netmask (e.g., **”address 192.168.0.1 255.255.255.0”**).

Use **”show address”** to show the IP address setting for this interface.

Default values Disabled (`no address`). That is, newly created interfaces have no IP address configured, see also section 19.2.3.

19.6.4 Primary Interface

Syntax `[no] primary`

Context Interface Configuration context

Usage This command is deprecated and only kept for backwards compatibility when upgrading. It is recommend to instead use the interface admin distance setting (section 19.6.5).

An old configuration file with this setting is converted to set the selected interface as distance 1 and keep other interfaces at their default distance of 16.

For more information, see section 19.2.6.

19.6.5 Interface Administrative Distance

Syntax `[no] distance <1-255> [trigger ID]`

Context Interface Configuration context

Usage 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 trigger ID may be set, e.g., for monitoring an upstream network with a ping trigger, and dynamically adjusting the default route to infinite distance. Effectively switching to another upstream interface not only on link loss.

For more information, see section 19.2.6.

Default values 16 (`no distance`)

Notes:

- A PPP interface created via PPPoE will ”inherit” the *admin distance* setting from its associated VLAN interface.
- The old *primary* setting on an interface is converted to distance 1 and all other interfaces are shifted downwards in priority.
- This setting does not apply to protocols such as RIP and OSPF.

19.6.6 Management Service Filtering

Syntax `[no] management <[ssh][telnet][http][https][ipconfig][snmp]|all>`

Context Interface Configuration context

Usage Filter management services on this interface.

The setting controls what services are allowed to use on this network interface. E.g., **”management ssh https”** adds SSH and HTTPS to the set of services accessible for traffic entering via this interface, and **”no management http”** disallows management via unencrypted HTTP on this interface.

Use **”no management”** to filter out access to all management services on this interface.

Use **”management all”** to allow all management services on this interface.

Use **”show management”** to show the list of currently allowed services via this interface.

Default values All services except **”telnet”** are allowed.

Note: PPP interfaces created via PPPoE will inherit the management settings from its associated VLAN interface.

19.6.7 VLAN Interface MAC address

Syntax `[no] mac <X:X:X:X:X:X>`

Context Interface Configuration context

Usage Configure a specific MAC address for this (VLAN) interface. The address is given as a colon-separated hexadecimal string of numbers, e.g., **”mac 00:1a:4b:7b:77:24”**. Leading zeros can be ignored. Uppercase or lowercase letters can be used.

Use **”no mac”** specify that the interface should get its MAC address automatically.

Use **”show mac”** to show the interface MAC setting for this (VLAN) interface.

For more information, see section 19.2.4.

Default values Auto (`no mac`)

19.6.8 Interface MTU Size

Syntax `[no] mtu <68-1500>`

Context Interface Configuration context

Usage Configure a non-default maximum transmission unit (MTU) size (in bytes) for this interface. The MTU size is the packet size a network interface will pass to the link layer for transmission, i.e., the maximum payload of the link layer protocol.

The default is to let the MTU depend on the type of link layer (*auto* mode). For interfaces associated with Ethernet and DSL links this implies a default MTU of 1500 bytes.

For PPP interfaces (PPPoE), the MTU is set to 8 bytes less than the MTU of the associated VLAN interface, which typically implies a PPP interface MTU of 1492 bytes (1500 − 8). This value is set at the time of PPP interface creation; if the VLAN interface MTU is changed afterwards, the PPP interface MTU is **not** updated automatically. Note: The operational MTU can change based on the PPP connection negotiation, see section 33.3.19.

The MTU of GRE interfaces defaults to 1476 bytes.

Use **”mtu <68-1500>”** to set a non-default MTU size. Use **”no mtu”** to specify that the interface should let its MTU be the default MTU of the associated link type.

Use **”show mtu”** to how the interface maximum transfer unit (MTU) size setting.

Default values

- *VLAN interfaces*: Auto (**”no mtu”**) For Ethernet and DSL links, this implies MTU 1500 bytes.
- *GRE interfaces*: 1476 bytes (**”mtu 1476”**)
- *PPP interfaces (PPPoE)*: Typically 1492 bytes (**”mtu 1492”**, i.e., 8 bytes less than the associated VLAN interface)

19.6.9 Interface TCP MSS Size

Syntax `[no] tcp-mss <40-1460|auto>`

Context Interface Configuration context

Usage Enable/disable TCP-MSS clamping on this interface.

TCP-MSS clamping is used to limit the packet size (or more precisely, limit the ”maximum TCP segment size”) of TCP connections over the given interface, and is useful in situations where path MTU discovery of some reason does not work.

Enabling TCP-MSS clamping implies additional packet processing, thus it degrades routing performance somewhat. It is disabled by default on most interface types (exception is PPP interface of type PPPoE).

Use **”tcp-mss <BYTES>”** to limit TCP-MSS to the given number of bytes.

Use **”tcp-mss auto”** to let the TCP-MSS depend on the MTU of the interface (”MTU-40”, i.e., interface MTU minus typical size of IP and TCP headers). This will work fine for typical TCP connections, but is not likely to work over IPsec tunnels or when additional IP header options are in use.

Use **”no tcp-mss”** to disable TCP-MSS clamping.

Use **”show tcp-mss”** to show the interface maximum TCP segment size (MSS).

Default values Disabled (`no tcp-mss`) (Exception: **”tcp-mss 1412”** for PPPoE PPP interfaces.)

19.6.10 Sending ICMP Redirect messages

Syntax `[no] redirect`

Context Interface Configuration context

Usage Enable/disable sending of ICMP Redirect messages. When enabled on a WeOS router, the router will send ICMP Redirect messages when detecting that packets coming in on this interface have a more optimal route towards the destination.

Use **”redirect”** to enable sending of ICMP Redirect, and **”no redirect”** to disable it.

Use **”show redirect”** to show if sending of ICMP Redirect is enabled or disabled.

Default values Enabled

19.6.11 Show Network Interface Status

Syntax `show iface [IFNAME]`

Context Admin Exec context.

Usage Show status information for this interface (or all interfaces). If dynamic address assignment is configured on an interface, this command will display the IP address acquired.

Default values Unless a specific interface is specified, status for all interfaces will be shown.

19.7 Managing general IP settings via the CLI

The available general IP settings and monitoring commands are shown below.

Command	Default	Section
Configure general IP settings		
ip		Section 19.7.1
[no] default-gateway <IPADDR>	(DEPRECATED)	Section 19.7.2
[no] route <NETWORK/LEN>		Section 19.7.3
<GATEWAY|IFNAME> [DISTANCE]	Distance 1	
[no] forwarding	Enabled	Section 19.7.4
[no] name-server <IPADDR>	Disabled	Section 19.7.5
[no] domain <DOMAIN>	Disabled	Section 19.7.6
[no] domain-proxy	Enabled	Section 19.7.7
[no] host <FQDN | HOSTNAME> <IPADDR>		Section 19.7.8
[no] ddns	Disabled	Section 19.7.9
[no] provider <dyndns|freedns|no-ip>	dyndns	Section 19.7.10
[no] ssl	Disabled	Section 19.7.11
[no] login <USERNAME> <PASSWORD>	Disabled	Section 19.7.12
[no] hostname <HOSTNAME>[,HASH]	Disabled	Section 19.7.13
[no] interval <SECONDS>	600	Section 19.7.14
icmp		Section 19.7.15
[no] broadcast-ping	Enabled	Section 19.7.16
[no] ntp	Disabled	Section 19.7.17
[no] enable	Enabled	Section 19.7.18
[no] server <FQDN|IPADDR>	N/A	Section 19.7.19
[no] enable	Enabled	Section 19.7.20
[no] poll-interval <SECONDS>	600 sec	Section 19.7.21
[no] sntp	(DEPRECATED)	Section 19.7.22
[no] server <FQDN|IPADDR>	Disabled	Section 19.7.23
[no] poll-interval <SECONDS>	600 sec	Section 19.7.24
Show general IP status		
show ip route		Section 19.7.25
show ip name-server		Section 19.7.26
show ip domain		Section 19.7.27
show ip host		Section 19.7.28
show ntp		Section 19.7.29

19.7.1 Manage Global IP Settings

Syntax `ip`

Context Global Configuration context

Usage Enter IP Configuration context

Use **"show ip"** to show general IP configuration settings.

Default values Not applicable.

19.7.2 Configure IP Default Gateway

Syntax `[no] default-gateway <ADDRESS>`

Context IP Configuration context

Usage This command is deprecated and only kept for backwards compatibility when upgrading. It is recommend to instead use the route command since it also has the distance attribute.

A default route configured using this command will always get a distance of 1. With multiple upstream WAN connections using PPPoE or DHCP it is recommended to use the route command instead.

Use **"show gateway"** to show configured default gateway.

Default values Disabled (**"no default-gateway"**)

19.7.3 Configure Static IP Routes

Syntax `[no] route <NET MASK | NET/LEN> <GATEWAY | IFNAME> [DISTANCE]`

Context IP Configuration context

Usage Add or remove a static IP route, including default routes.

The network boundary of the destination subnet can be given as a netmask (e.g., **"route 192.168.3.0 255.255.255.0 192.168.0.1"**) or as a prefix length (e.g., **"route 192.168.3.0/24 192.168.0.1"**).

System default routes are setup using the subnet 0.0.0.0 with prefix length 0, but the key keyword 'default' is much easier to use **"route default**

192.168.0.1”. The optional distance is useful when setting up backup routes in multiple upstream scenarios where interfaces acquire default routes using PPPoE or DHCP.

The destination network is however typically located *remotely* (specify the next hop gateway, e.g., **”route 192.168.3.0/24 192.168.0.1”**), but it is also possible to use the static route command to specify additional *directly connected* subnets (specify the local interface, e.g., **”route 192.168.3.0/24 vlan1”**).

Use the ”no”-form to remove a static route, e.g., **”no route 192.168.3.0/24 192.168.0.1”**.

Use **”show route”** to list configured static routes.

Default values Using **”no route”** (without a subnet address, etc.) removes all configured static routes.

19.7.4 Manage IP Forwarding

Syntax `[no] forwarding`

Context IP Configuration context

Usage (only for WeOS Extended) Enable/disable IPv4 routing.

Use **”show forwarding”** to show whether IP forwarding (routing) is enabled or disabled.

Default values Enabled (**”forwarding”**)

19.7.5 Name Server (DNS)

Syntax `[no] name-server <ADDRESS>`

Context IP Configuration context

Usage Add/remove name-server (DNS). Two name-servers can be configured - call the same **”name-server”** command twice.

Run **”no name-server <ADDRESS>”** to remove a specific name server, or **”no name-server”** to remove all configured name servers.

If a name server is not configured using the **”name-server”** command, name server(s) (and domain search path) can be acquired dynamically from an interface with DHCP address assignment.

Use **”show name-server”** to show configured name servers.

Default values Disabled (**”no name-server”**) Running **”no name-server”** (without specifying any name removes all configured name servers.

19.7.6 Domain Search Path

Syntax `[no] domain <DOMAIN>`

Context IP Configuration context

Usage Add/remove domain search path. A single search path can be added.

Run **”no domain”** to remove the domain search path.

If a name server is not configured using the **”name-server”** command, domain(s) can be acquired dynamically from an interface with DHCP address assignment.

Use **”show domain”** to show configured domain search path.

Default values Disabled (**”no domain”**)

19.7.7 Enable/Disable DNS proxy service

Syntax `[no] domain-proxy`

Context IP Configuration context

Usage Enable or disable DNS proxy support. When enabled, the unit will act as a DNS server and respond to DNS queries for *known hosts*:

- either statically added by the **”host”** (section 19.7.8), see also the **”show ip host”** (section 19.7.28) command, or
- hosts for which this unit acts as DHCP server (chapter 22), see also the **”show dhcp-clients”** (section 22.3.20) command .

Furthermore, the unit will act as a caching DNS forwarder; DNS queries of unknown hosts are forwarded to the unit’s own DNS server (see the **”show**

ip name-server” command described in section 19.7.26), and the answer is cached for fast response of subsequent requests for the same host.

Use command **”domain-proxy”** to enable the DNS proxy service, and **”no domain-proxy”** to disable it.

Use **”show domain-proxy”** to view the current setting.

Default values Enabled (**”domain-proxy”**)

Example

```
example:/#> show ip host
127.0.0.1 localhost
127.0.1.1 example.local example

192.168.3.11    mypc
example:/#> ping mypc
Press Ctrl-C to abort PING mypc (192.168.3.11): 56 data bytes
64 bytes from 192.168.3.11: seq=0 ttl=64 time=1.049 ms
64 bytes from 192.168.3.11: seq=1 ttl=64 time=0.627 ms
^C
--- mypc ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.627/0.838/1.049 ms
example:/#> show dhcp-clients
Lease Time  MAC Address        IP Address       Hostname         Client ID
================================================================================
120         00:07:7c:03:ec:02  192.168.5.106    alice            01:00:07:7c:03:ec:02
example:/#> ping alice
Press Ctrl-C to abort PING alice (192.168.5.106): 56 data bytes
64 bytes from 192.168.5.106: seq=0 ttl=64 time=1.182 ms
64 bytes from 192.168.5.106: seq=1 ttl=64 time=0.754 ms
^C
--- alice ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.754/0.968/1.182 ms
example:/#>
```

19.7.8 Add static hostname lookup entry

Syntax `[no] host <FQDN | HOSTNAME> <IPADDR>`

Context IP Configuration context

Usage Add or delete entries in the static hostname resolution table (host table). The table is both used when resolving hostnames of DNS requests originating from the unit itself (e.g., when running **”ping www.example.com”** from the CLI command line), and when responding to DNS queries from hosts (assuming this unit is configured as DNS proxy, see section 19.7.7).

- Hostnames containing a dot (".") are interpreted as fully qualified domain names (FQDN).
- Hostnames without a dot are interpreted as simple hostnames. The system will both be able to resolve DNS queries for the *hostname*, as well as *hostname* concatenated with the unit's *domain search path*. Use **"show ip domain"** (section 19.7.27) to view the unit's search path domain.

Example

```
example:/#> configure
example:/config/#> ip
example:/config/ip/#> domain example.org
example:/config/ip/#> host mypc 192.168.10.1
example:/config/ip/#> host www.anotherexample.org 10.0.0.1
example:/config/ip/#> leave
example:/#> show ip hosts
127.0.0.1 localhost
127.0.1.1 example.local example

192.168.10.1    mypc    mypc.example.org
10.0.0.1        www.anotherexample.org
example:/#> ping mypc
Press Ctrl-C to abort PING mypc (192.168.10.1): 56 data bytes
64 bytes from 192.168.10.1: seq=0 ttl=64 time=8.291 ms
64 bytes from 192.168.10.1: seq=1 ttl=64 time=0.650 ms
^C
--- mypc ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.650/4.470/8.291 ms
example:/#>
```

Use **"no host <HOSTNAME>"** to remove a specific entry in the host table, and **"no host"** to remove all configured entries in the host table.

Use **"show host"** to view the currently configured static host entries.

Default values Not applicable (no static host entries configured)

19.7.9 Manage DDNS Settings

Syntax `[no] ddns`

Context IP Configuration context

Usage Enter DDNS Configuration context. Upon entering the context, the DDNS service will be enabled. However, it will not be activated until valid DDNS parameters (login, etc.) are configured. Use **"no ddns"** to disable the DDNS service.

Use **”show ddns”** to show configured DDNS settings (also available as **”show”** command within the DDNS Configuration context).

Default values Disabled (**”no ddns”**)

19.7.10 Set DDNS Provider

Syntax `[no] provider <dyndns|freedns|no-ip>`

Context DDNS Configuration context

Usage Set DDNS provider. Example of supported providers:

dyndns `http://www.dyndns.org`,

freedns `http://freedns.afraid.org`, and

no-ip `http://www.no-ip.com`

For a complete list of supported DDNS providers, type **”help provider”**. Use **”no provider”** to return to the default provider setting.

Default values `dyndns`

19.7.11 Enable HTTPS Updates

Syntax `[no] ssl`

Context DDNS Configuration context

Usage Enable/disable HTTPS updates, if the provider supports it.

Use **”show ssl”** to show whether HTTPS updates is enabled or disabled.

Default values Disabled (HTTP)

19.7.12 Set DDNS Login and Password

Syntax `[no] login <USERNAME> <PASSWORD>`

Context DDNS Configuration context

Usage Set login *username* and *password* for your account at your DDNS provider (see section 19.7.10). Use **”no login”** to remove a configured DDNS login setting.

Default values Disabled

19.7.13 Set DDNS Hostname

Syntax `[no] hostname <HOSTNAME>[,HASH]`

Context DDNS Configuration context

Usage Set the DNS hostname, i.e., registered domain name which should map to the IP address of this your switch.

When selecting **”provider freedns”**, the domain name must be followed by a hash value (**”hostname HOSTNAME,HASH”**); the *hash* is provided by FreeDNS).

Default values Disabled

19.7.14 Set DDNS interval

Syntax `[no] interval <SECONDS>`

Context DDNS Configuration context

Usage Set the interval by which DDNS verifies that the IP address mapping at your DDNS provider matches the IP address of your switch. Maximum 10 days (864000 seconds).

Use **”no interval”** to return to the default provider setting.

Default values `600` (seconds)

19.7.15 Manage ICMP Settings

Syntax `icmp`

Context IP Configuration context

Usage Enter ICMP Configuration context.

Use **”show icmp”** to show ICMP settings (also available as **”show”** command within the ICMP Configuration context).

Default values Not applicable.

19.7.16 Enable/disable Broadcast Ping

Syntax `[no] broadcast-ping`

Context ICMP Configuration context

Usage Define whether the switch should respond to broadcast ”ping” (ICMP Echo Request) messages or not. Responding to broadcast ping is convenient when troubleshooting the network, but can in some situations be considered a security risk.

Use **”no broadcast-ping”** to disable responding to broadcast ping messages.

Use **”show broadcast-ping”** to show whether the switch is configured to respond to broadcast ping messages or not.

Default values Enabled (**”broadcast-ping”**)

19.7.17 Manage NTP Settings

Syntax `[no] ntp`

Context Global Configuration context

Usage Enter NTP Configuration context by using the **”ntp”** command.

Use **”no ntp”** to remove all configured NTP settings.

Use **”show ntp”** to show NTP settings (also available as **”show”** command within the NTP Configuration context).

Default values Not applicable.

19.7.18 Enable/Disable NTP Settings

Syntax `[no] enable`

Context NTP Configuration context

Usage Enable or disable configured NTP settings.

Use **”enable”** to enable/activate configured NTP settings. Use **”no enable”** to disable/deactivate configured NTP settings (the settings are not removed, only deactivated).

Use **”show enable”** to show whether NTP settings are enabled or disabled.

Default values Enabled

19.7.19 Manage (remote) NTP Server(s)

Syntax `[no] server <FQDN|IPADDR>`

Context NTP Configuration context

Usage Add, delete, or manage a *remote* NTP server with specified IP Address or domain name, to set the time on this unit. Up to 8 NTP servers can be configured. With the **”server <FQDN|IPADDR>”** you enter the NTP Remote Server Configuration context for that specific NTP server.

If no (remote) NTP server is configured, the unit can acquire NTP server(s) dynamically from an interface with DHCP address assignment.

Use **”no server <FQDN|IPADDR>”** to remove a specific NTP server, or **”no server”** to remove all configured NTP servers.

Use **”show server”** to show settings for all configured NTP servers, or **”show server <FQDN|IPADDR>”** to show NTP settings for a specific NTP server (also available as **”show”** command within the NTP Remote Server Configuration context).

Default values Not applicable

19.7.20 Enable/Disable (remote) NTP Server

Syntax `[no] enable`

Context NTP Remote Server Configuration context

Usage Enable or disable configured settings for this NTP server.

Use **”enable”** to enable/activate configured NTP server settings. Use **”no enable”** to disable/deactivate configured NTP server settings (the settings are not removed, only deactivated).

Use **”show enable”** to show whether NTP server settings are enabled or disabled.

Default values Enabled

19.7.21 Set NTP Server Poll Interval

Syntax `[no] poll-interval <30-720>`

Context NTP Remote Server Configuration context

Usage Set NTP server poll interval (in seconds) for this NTP server. **”no poll-interval”** will reset the poll interval to its default (600 seconds).

Use **”show poll-interval”** to show configured poll interval.

Default values 600 (seconds)

19.7.22 Manage NTP Client Settings (Deprecated)

Syntax `[no] sntp`

Context Global Configuration context

Usage Enable NTP client and enter NTP Client Configuration context by using the **”sntp”** command.

Use **”no sntp”** to disable the NTP client service.

Use **”show sntp”** to show NTP client settings (also available as **”show”** command within the NTP Client Configuration context).

> **Note**
> The NTP Client Configuration context is deprecated and kept for backwards compatibility. NTP client settings is instead handled as part of other NTP settings in the NTP Configuration, see section 19.7.17.

Default values Not applicable.

19.7.23 Set NTP Client (Remote) Server Address (deprecated)

Syntax `[no] server <FQDN|IPADDR>`

Context NTP Client Configuration context

Usage Set IP Address, or domain name, of NTP Server used by this client.

A single NTP server IP address, or a fully qualified domain name, FQDN, can be configured. If disabled, NTP server can be acquired dynamically from an interface with DHCP address assignment.

”no server” to remove a configured NTP server address.

Use **”show server”** to show NTP client (remote) server setting.

> **Note**
> The **”server”** command within NTP Client Configuration context is deprecated and kept for backwards compatibility. NTP client settings for remote server is instead handled as part of other NTP settings in the NTP Remote Server Configuration, see section 19.7.19.

Default values Disabled

19.7.24 Set NTP Poll Interval (deprecated)

Syntax `[no] poll-interval <30-720>`

Context NTP Client Configuration context

Usage Set NTP server poll interval (in seconds). **”no poll-interval”** will reset the poll interval to its default (600 seconds).

Use **”show poll-interval”** to show configured poll interval.

> **Note**
> The **”poll-interval”** command within NTP Client Configuration context is deprecated and kept for backwards compatibility. NTP client settings for remote server is instead handled as part of other NTP settings in the NTP Remote Server Configuration, see section 19.7.21.

Default values 600 (seconds)

19.7.25 Show IP Forwarding Table

Syntax `show ip route`

Context Admin Exec context

Usage Show IP Forwarding table (summary of configured routes and routes acquired dynamically).

Default values Not applicable.

19.7.26 Show Name Server and Domain Search Path Status Information

Syntax `show ip name-server`

Context Admin Exec context

Usage Show name-server and domain search path status information (statically configured or acquired dynamically)

19.7.27 Show Domain Search Path Status Information

Syntax `show ip domain`

Context Admin Exec context

Usage Show domain search path status information (statically configured or acquired dynamically)

Example

```
example:/#> show ip domain
example.org
example:/#>
```

19.7.28 Show local host table

Syntax `show ip hosts`

Context Admin Exec context

Usage Show the local hostname resolution table. Static hostname resolution entries configured with the **"host"** command (section 19.7.8) are listed, as well as entries for the unit itself (localhost and entries for the unit's own hostname, see section 20.2.2).

Example

```
example:/#> show ip hosts
127.0.0.1 localhost
127.0.1.1 example.local example

192.168.10.1    mypc    mypc.example.org
10.0.0.1        www.anotherexample.org
example:/#>
```

19.7.29 Show NTP Status Information

Syntax `show ntp [verbose]`

Context Admin Exec context

Usage Show NTP status information. An asterisk '*' shows which NTP server is used to synchronize the time. For more information, use **"show ntp verbose"**.

Example

```
example:/#> show ntp
NTP Client/Server running as PID: 805
210 Number of sources = 2
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* ntp-anycast.kth.se            2   9    37   370   +222us[ -916ms] +/-   22ms
^- cecar.ddg.lth.se              2   9    37   370  -8317us[-8317us] +/-   81ms
```

Chapter 20

General System Settings

WeOS provides management of a set of features related to *system identity* and other general system settings. The table below gives a summary of the features available via the web and CLI management interfaces.

System *hostname*, *location* and *contact* correspond to the associated system objects of the original MIB-2 standard MIB (RFC 1213). For more information on WeOS SNMP support, see chapter 6.

Feature	Web	CLI
System Hostname	X	X
System Location	X	X
System Contact	X	X
System Time Zone	X[1]	X
System Date/Time	X	X
CPU bandwidth limitation		X

Section 20.1 covers management of general system settings via the Web interface, and section 20.2 describes the corresponding features in the CLI.

[1] Web configuration of System Time Zone is done as part of the Network settings, see section 19.5.

20.1 Managing switch identity information via the web interface

20.1.1 Manage System Identity Information

Menu path: Configuration ⇒ System ⇒ Identity

Fig 20.1 shows the page where you can set hostname, location and contact information for your switch.

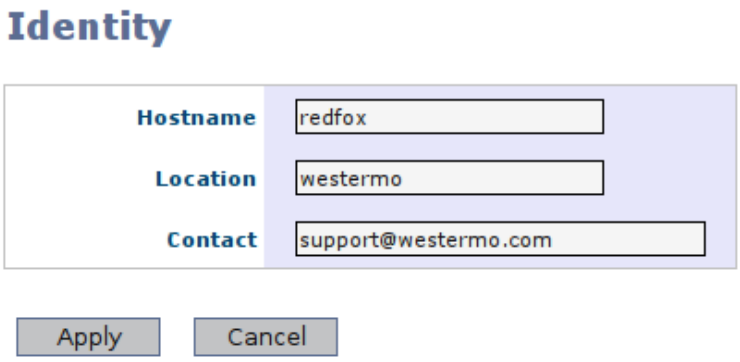


Figure 20.1: Switch identity settings example

Hostname	A name to identify this unit. Max 64 characters. Valid characters are A-Z, a-z, 0-9, and hyphen (-). The first character should be alphabetic (A-Z, a-z). Hyphen is not valid as first or last character.
Location	A description to identify where the unit is located. Max 64 characters. Valid characters are ASCII 32-126 except '#' (ASCII 35). "Space" (ASCII 32) is not valid as first or last character.
Contact	A description identifying whom to contact regarding management of the unit. Max 64 characters. Valid characters are ASCII 32-126 except '#' (ASCII 35). "Space" (ASCII 32) is not valid as first or last character.

Change the values to appropriate values for your switch and click the **Apply** button.

20.1.2 Set System Date and Time

Menu path: Configuration ⇒ System ⇒ Date & Time

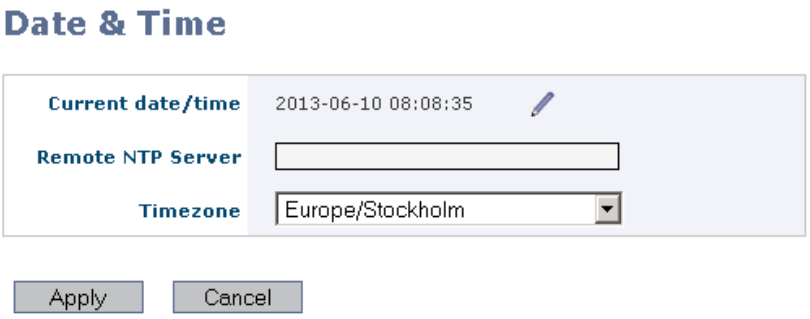


Figure 20.2: Switch date and time settings, NTP server

Current Date/Time	Shows current date and time. Click the icon to maually set date/time .
Remote NTP Server	The IP address of a time server to be used to keep the units calendar time synchronised. Leave empty if you do not want to use a time server.
Timezone	Select a timezone region to get adjusted local time.

20.2 Managing switch identity information via CLI

Command	Default	Section
Configure General System Settings		
system		Section 20.2.1
[no] hostname <ID>	*"family"*[1]	Section 20.2.2
[no] location <ID>	(empty)	Section 20.2.3
[no] contact <ID>	(empty)	Section 20.2.4
[no] timezone <TIMEZONE>	Etc/UTC	Section 20.2.5
[no] cpu-bandwidth-limit <auto|<64-100000>| <7700-1488000> fps>	Auto	Section 20.2.6
Set date and time		
date		Section 20.2.7
View available time zones		
system		
show timezone [QUERY|SUBSTRING]		Section 20.2.8

20.2.1 Manage System Identity Information

Syntax `system`

Context Global Configuration context

Usage Enter General System Configuration configuration context.

> Use **"`show system`"** to show all general system configuration settings (also available as **"`show`"** command within the General System Configuration).

Default values Not applicable

20.2.2 System Hostname

Syntax `hostname <STRING>`

[1] The default hostname depends on the product family, e.g., Lynx products have default hostname *lynx*.

Context General System Configuration context

Usage Set system hostname string.

Max 64 characters. Valid characters are A-Z, a-z, 0-9, and hyphen (-). The first character should be alphabetic (A-Z, a-z). Hyphen is not valid as first or last character.

”no hostname” resets the hostname to its default value.

Use **”show hostname”** to view the configured hostname setting.

Default values *”family”* (The default hostname depends on the product family, e.g., Lynx products have default hostname *lynx*.)

20.2.3 System Location

Syntax `location <STRING>`

Context General System Configuration context

Usage Set system location string.

Max 64 characters. Valid characters are ASCII 32-126 except ’#’ (ASCII 35). ”Space” (ASCII 32) is not valid as first or last character.

”no location” resets the location string to its default value (empty).

Use **”show location”** to view the configured location setting.

Default values (empty)

20.2.4 System Contact

Syntax `contact <STRING>`

Context General System Configuration context

Usage Set system contact string.

Max 64 characters. Valid characters are ASCII 32-126 except ’#’ (ASCII 35). ”Space” (ASCII 32) is not valid as first or last character.

”no contact” resets the contact string to its default value (empty).

Use **”show contact”** to view the configured contact setting.

Default values (empty)

20.2.5 Set System Time Zone

Syntax `[no] timezone <TIMEZONE>`

Context General System Configuration context.

Usage Set system time zone string.

> **Note**
> For information of available time zone settings, see section 20.2.8.

"no timezone" resets the timezone to its default value (Etc/UTC).

Use **"show timezone"** to view the configured timezone setting.

Default values Etc/UTC

20.2.6 CPU bandwidth limitation

Syntax `[no] cpu-bandwidth-limit <auto|<64-1000000>|<7700-1488000> fps>`

Context General System Configuration context

Usage Limit the traffic sent to the CPU in kbit/s or frames per second (traffic *from* the CPU is not affected). It is also possible use ISO modifiers k/M/G, e.g., 256k or 10M as specifiers for kbps and Mbps.

> **Note**
> CPU bandwidth limit in *frames per second* mode is available on all WeOS products, with exceptions for some RedFox models. On RedFox, the *frames per second* mode is available for products based on the "Corazon" platform, including RedFox Industrial Rack, newer RedFox Rail (RFR-212-FB)[50] and newer RedFox Industrial[48]. But the *frames per second* mode is *not* available on RedFox products based on the (older) "Atlas" platform, i.e, RedFox Industrial listed in [47] or older RedFox Rail (RFR-12-FB).

Set values are rounded off to the nearest possible HW setting.

Note
Default is **"auto"**, which means that system will automatically reduce CPU bandwidth when critical services are enabled. As of WeOS v4.17.1, FRNT Ring Bridging or Multi-link Dual-Homing (see chapter 15) are considered critical, but the set of critical services may change in future WeOS releases.
A user can override the default with **"no cpu-bandwidth-limit"** or any more specific setting (e.g., **"cpu-bandwidth 4M"**). However, for critical services it is recommended leave the default **"auto"**.

On units with multiple CPU channels (see section 13.1.6), the setting will apply for each of the channels..

Use **"no cpu-bandwidth-limit"** to disable CPU bandwidth limitation.

Use **"show cpu-bandwidth-limit"** to view the configured CPU bandwidth limit setting.

Default values Auto (**"cpu-bandwidth-limit auto"**)

20.2.7 Set or show System Date and Time

Syntax `date [[YYYY-MM-DD ]hh:mm[:ss]]`

Context Admin Exec context.

Usage Set system date and time, or only time.

Use **"show date"** to view the current date and time.

Default values If no date or time is given, the current date and time will be displayed.

Example

```
example:/#> date 2013-05-31 10:18
Fri May 31 10:18:00 UTC 2013
example:/#> show date
Fri May 31 10:18:09 UTC 2013
example:/#>
```

20.2.8 Show System Time Zone

Syntax `show timezone [QUERY|SUBSTRING]`

Context General System Configuration context.

Usage Show system time zone setting/list available time zones.

When given without any argument (**”show timezone”**), the configured time zone setting is presented.

When providing an argument, the available time zone settings matching that argument is listed, e.g., issuing the command **”show timezone asia”** will list all possible time zone configuration settings for Asia (or more precisely, all available time zones containing the substring ’asia’.) See section 20.2.5 for information of how to set the system time zone.

Default values Not applicable.

Chapter 21

AAA - Authentication, Authorisation and Accounting

This chapter describes WeOS AAA support - Authentication, Authorisation and Accounting. The AAA configuration gathers authentication methods and policies in one place and is referenced from other subsystems in WeOS. Three uses of AAA are currently supported:

- *WeOS unit login:* The login password to the unit is part of AAA.
- *Port Based Access Control (PNAC):* WeOS supports port access control with IEEE 802.1X and MAC based authentication. This is configured in two different places, in AAA and as settings related to VLAN. The configuration in AAA specifies RADIUS backends and MAC filtering lists, and the configuration in VLAN which ports are enabled for port access control. See section 13.2 for an introduction and guidance to configure port based access control with WeOS.
- *PPP Peer Authentication:* You can create and use local user database lists to authenticate and authorise your PPP peers. This is typically used for PPP connections in dial-in/server mode (see section 33), but you can also use this to authenticate and authorise your peer in other PPP modes.

21.1 Overview over AAA

Feature	Web	CLI	General Description
Login account management	X	X	Section 21.1.1
Local user DB	X	X	Section 21.1.2
RADIUS			
RADIUS servers	X	X	
RADIUS server groups	X	X	
Port Based Access Control			
IEEE 802.1X Access Control Instances	X	X	
MAC authentication lists	X	X	

21.1.1 Login Account Management

Currently WeOS only supports a single login user account, the **admin** user account. The same account is used when managing the switch via the Web or via the CLI. Factory default settings for the user account is:

- Login: **admin**
- Password: **westermo**

The *admin* password can be changed, both via the Web and the CLI interfaces. Account passwords can be at most 64 characters long (longer passwords are truncated). Printable ASCII[1] characters except ”space” (ASCII 33-126) are allowed in the password.

Section 7.1.3 provides information on how to proceed in case you forget the **admin** password.

21.1.2 User Authentication Lists - Local Databases

Local user databases are useful for storing authentication credentials with no need for any external infrastructure. The lists consist of user name and password pairs which are stored in plain text. In the future it will also be possible to store hashed passwords.

[1] American Standard Code for Information Interchange (ASCII), see e.g. http://en.wikipedia.org/wiki/ASCII (accessed May 2009).

Currently local databases can only be used to authenticate PPP peers. A PPP peer can be a user connecting via an external modem or over PPPoE (and in future releases of WeOS, via an L2TP or PPTP VPN tunnel). The most common configuration is to require the peer to authenticate itself when the WeOS device has a server role, but it is also possible to require authentication in a client configuration.

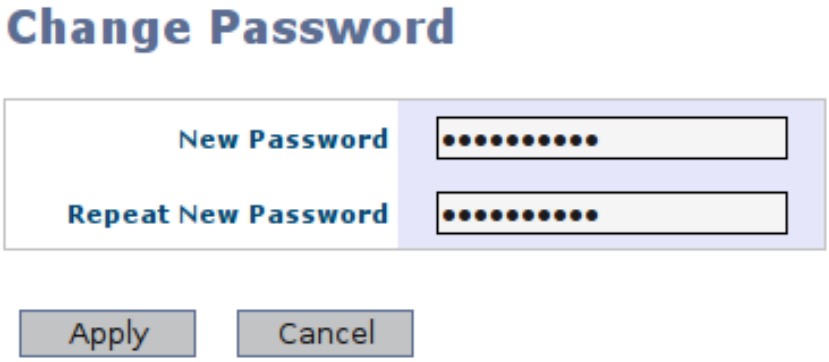
When a local database is created, a numeric ID is associated with it. This ID will be used to reference the database from other contexts within WeOS. Additionally, a description string may also be associated with the instance to make it easier to remember their purpose, i.e. “maintainers”.

21.2 Managing AAA via the web interface

21.2.1 Login Account Management via the Web Interface

Menu path: Maintenance ⇒ Password

The only account management feature in the web management tool at the moment is change of the *admin* password.



New Password	Enter the new password for the *admin* account.
Repeat New Password	To minimise risk of typing error, enter the new password for the *admin* account once again.

21.2.2 Local User Databases

Menu path: Configuration ⇒ AAA ⇒ Local User DB

The main page for local user databases shows an overview of created databases.

ID	A unique identifier for the local user database.
Description	The users description of this database.
Edit	Click this icon to edit the user database. See section 21.2.4 for details.
Delete	Click this icon to remove the user database. You will be asked to acknowledge the removal before it is actually executed.
New	Click this button to add a new user database. See section 21.2.3 for details. You can create at maximum 4 databases.

21.2.3 New Local User Database

Menu path: Configuration ⇒ AAA ⇒ Local Users DB ⇒ **New**

ID	The local user database identifier. This is generated automatically in the web interface and can not be changed.
Description	Optional. A user defined description for this database that will be visible in listings.

After pressing the **Apply** button, users can be added to the database. See section 21.2.5.

21.2.4 Edit a local user database

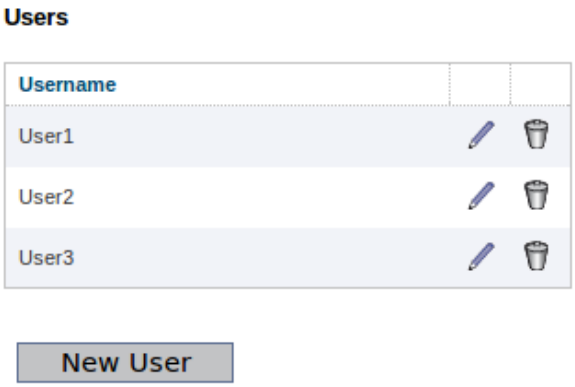
Menu path: Configuration ⇒ AAA ⇒ Local Users DB ⇒ 

See section 21.2.3 for descriptions of the fields on this page.

21.2.5 Users

Menu path: Configuration ⇒ AAA ⇒ Local Users DB ⇒ 

The users list is displayed on the edit page for the local user database.



Username	A username unique in this database.
New User	Press this button to create a new user in this database. See section 21.2.6

21.2.6 New User

Menu path: Configuration ⇒ AAA ⇒ Local Users DB ⇒ ✎ ⇒ **`New User`**

Username	A username unique in this database.
Password	The password for this user.

21.2.7 Edit User

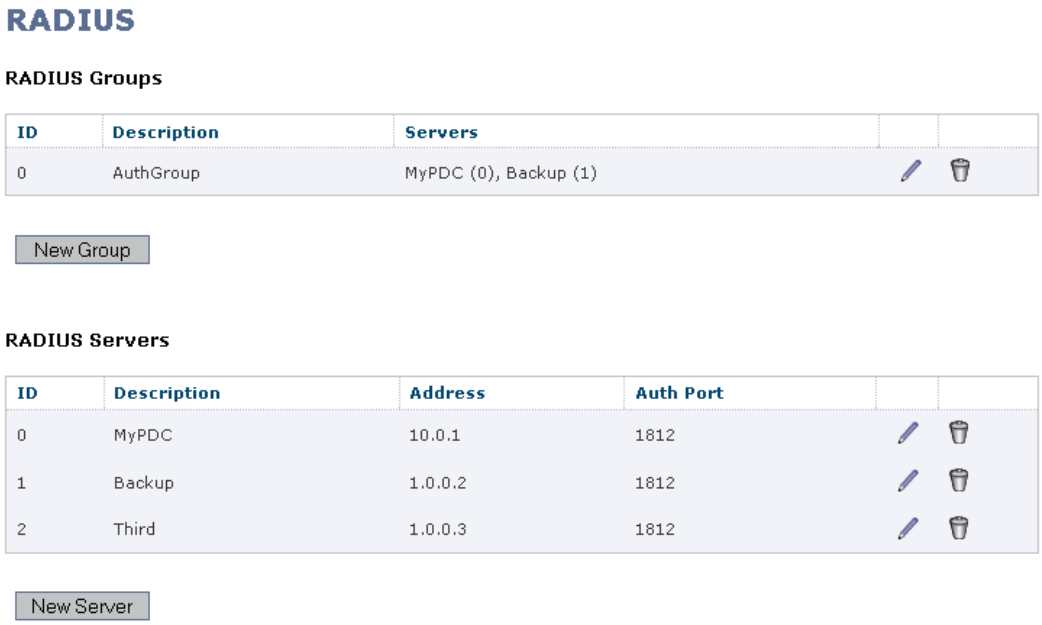
Menu path: Configuration ⇒ AAA ⇒ Local Users DB ⇒ ✎ ⇒ ✎ (Users table)

See section 21.2.6 for descriptions of the fields on this page.

21.2.8 RADIUS overview

Menu path: Configuration ⇒ AAA ⇒ RADIUS

The main page for RADIUS shows an overview of configured RADIUS groups and the remote RADIUS server configurations.



21.2.8.1 RADIUS groups in the overview

ID	The RADIUS group identifier
Description	The user defined name of this group
Servers	List of RADIUS servers included in this group. Each server is presented by its description name and the server identifier inside parentheses
Edit	Click this icon to edit the RADIUS group. See section 21.2.9 for details.

Continued on next page

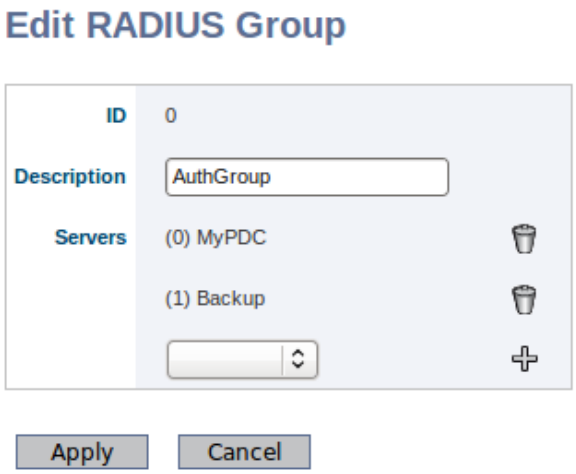
	Continued from previous page
Delete	Click this icon to remove the RADIUS group. You will be asked to acknowledge the removal before it is actually executed. Removing a group will not remove the config of the included servers.
New Group	Click this button to add a new RADIUS group. See section 21.2.10 for details. You can create at maximum 2 RADIUS groups.

21.2.8.2 RADIUS servers in the overview

ID	The remote RADIUS server identifier
Description	The user defined name on this server
Address	IP or FQDN of the RADIUS server
Auth Port	The UDP port used for authentication
Edit	Click this icon to edit the remote RADIUS server setting. See section 21.2.11 for details.
Delete	Click this icon to remove the remote RADIUS server setting. You will be asked to acknowledge the removal before it is actually executed.
New Server	Click this button to add a new remote RADIUS server configuration. See section 21.2.12 for details. You can define at maximum 6 remote RADIUS configurations.

21.2.9 Edit a RADIUS group

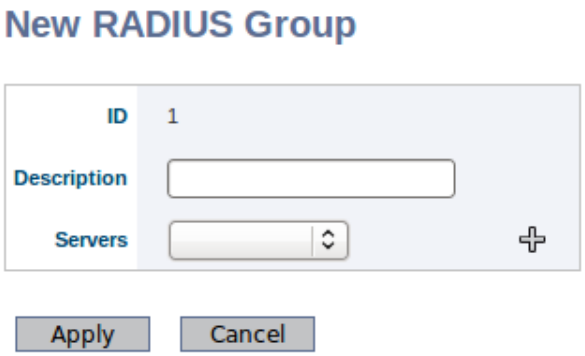
Menu path: Configuration ⇒ AAA ⇒ RADIUS ⇒ (RADIUS group)



ID	The RADIUS group identifier. This is generated automatically in the web interface and can not be changed.
Description	Optional. A user defined name for this group that will be visible in listings.
Servers	Remote RADIUS servers that are included in this group. The order of this list is important as it defines the order that servers are queried. Select a server in the drop-down list and add it by clicking the **plus** icon. Use the icon to remove a server from the group. You are limited to max 3 servers per group.

21.2.10 Add a RADIUS group

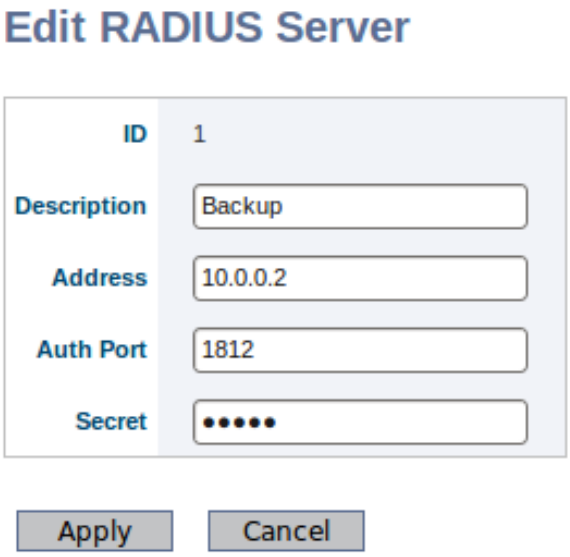
Menu path: Configuration ⇒ AAA ⇒ RADIUS ⇒ New Group



See section 21.2.9 for descriptions of the fields on this page. You can have at maximum 2 RADIUS server groups.

21.2.11 Edit a remote RADIUS server

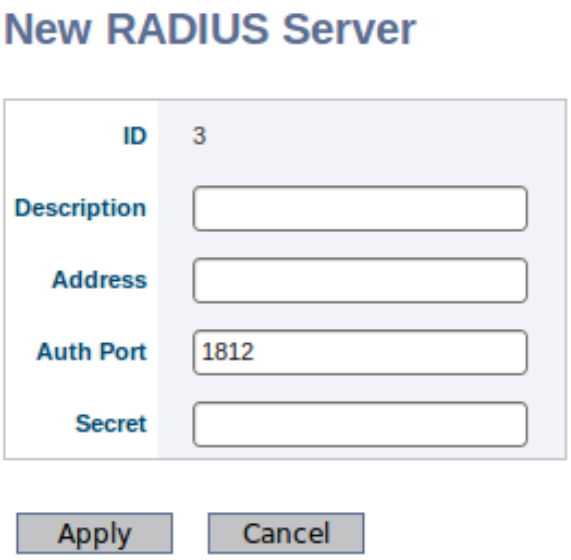
Menu path: Configuration ⇒ AAA ⇒ RADIUS ⇒ (RADIUS server)



ID	The remote RADIUS server identifier. This is generated automatically in the web interface and can not be changed.
Description	Optional. A user defined name for this server configuration that will be visible in listings.
Address	Mandatory. The IP number or Fully Qualified Domain Name (FQDN) to the remote RADIUS server
Auth Port	Mandatory. The UDP port number for RADIUS authentication requests. The default and standardised port number for this is 1812 but can be changed here if needed.
Secret	Optional. A shared secret (password) that should be used to encrypt the communication with this RADIUS server.

21.2.12 Add a remote RADIUS server

Menu path: Configuration ⇒ AAA ⇒ RADIUS ⇒ New Server

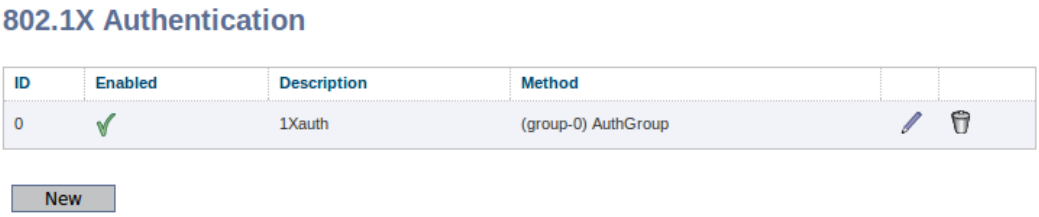


See section 21.2.11 for descriptions of the fields on this page. You can have at maximum 6 remote RADIUS server configurations.

21.2.13 IEEE 802.1X authentication

Menu path: Configuration ⇒ AAA ⇒ 802.1X

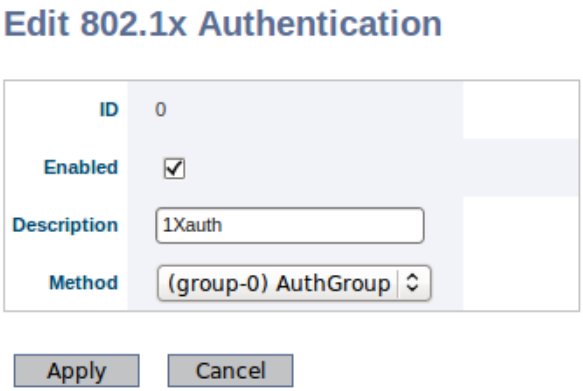
Here you see a listing of currently configured 802.1X instances.



ID	The IEEE 802.1X instance identifier
Enabled	If this instance is active, A green check-mark means yes and a dash means no
Description	The user defined name on this IEEE 802.1X instance
Method	The RADIUS server or group used for this instance
Edit	Click this icon to edit the instance See section 21.2.14 for details.
Delete	Click this icon to remove the instance. You will be asked to acknowledge the removal before it is actually executed. Removing an IEEE 802.1X instance will not remove the referenced RADIUS group or server.
New	Click this button to add a new IEEE 802.1X instance. See section 21.2.15 for details. *You can currently only create one instance*.

21.2.14 Edit an IEEE 802.1X instance

Menu path: Configuration ⇒ AAA ⇒ 802.1X ⇒ 

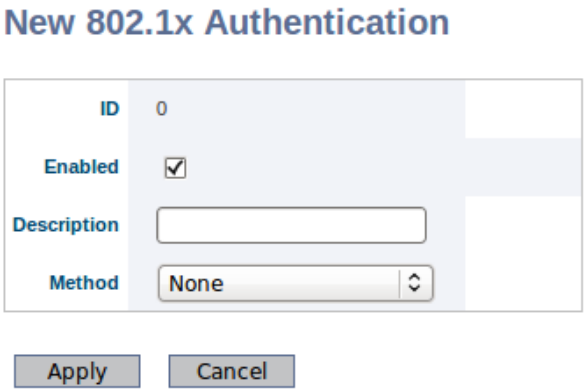


ID	The IEEE 802.1X instance identifier. This is generated automatically in the web interface and can not be changed.
Enabled	Check to enable this instance.
Description	Optional. A user defined name for this instance.
Method	Mandatory. Use this drop-down menu to select a RADIUS group or a remote RADIUS server. RADIUS groups and remote servers are created separately. See section 21.2.10 and section 21.2.12.

IMPORTANT: Creating an IEEE 802.1X instance does *not* in itself activate authentication. Port access is managed in the VLAN configuration. See sections 13.2 and 13.3.4. The instance here must be referenced from the port access configuration for it to be used!

21.2.15 Add an IEEE 802.1X instance

Menu path: Configuration ⇒ AAA ⇒ 802.1X ⇒ New

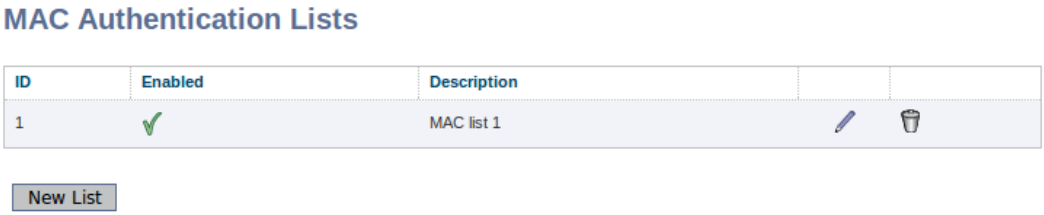


See section 21.2.14 for descriptions of the fields on this page. *You can currently only configure one IEEE 802.1X instance*.

21.2.16 MAC based authentication

Menu path: Configuration ⇒ AAA ⇒ MAC Auth

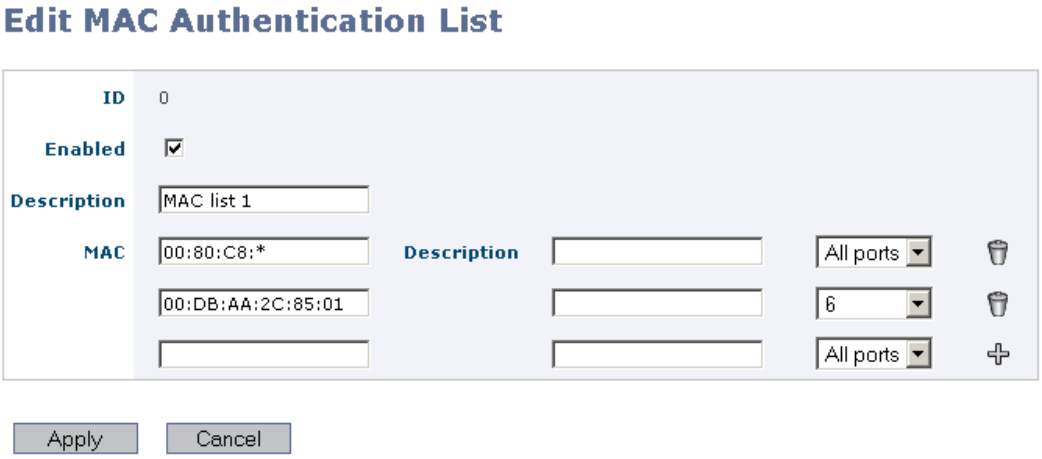
Here you see a listing of currently configured MAC authentication lists.



ID	The MAC authentication list identifier
Enabled	If this list is active, A green check-mark means yes and a dash means no
Description	The user defined name on this MAC authentication list
Edit	Click this icon to edit the list See section 21.2.17 for details.
Delete	Click this icon to remove the list. You will be asked to acknowledge the removal before it is actually executed.
New List	Click this button to add a new MAC authentication list. See section 21.2.18 for details. You can create up to 8 MAC authentication lists.

21.2.17 Edit a MAC authentication list

Menu path: Configuration ⇒ AAA ⇒ MAC Auth ⇒ 

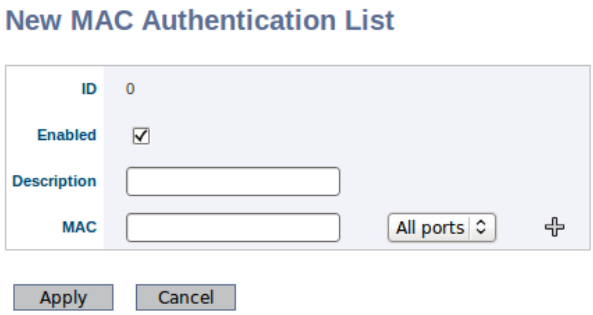


ID	The MAC authentication list identifier. This is generated automatically in the web interface and can not be changed.
Enabled	Check to enable this list.
Description	Optional. A user defined name for this list.
MAC	Optional. A list of MAC addresses and MAC address patterns. Single MAC addresses are specified in the format: *HH:HH:HH:HH:HH:HH*. A wildcard * can be used at the end of the pattern to match a block of addresses. Examples: *00:80:C8:**, *00:D8:AA:2C:85:01*. Use the drop-down list to select a port if you want the pattern to only be valid for requests coming in through a specific port. The description field is optional. Add a pattern by clicking on the **plus** icon. Use the  icon to remove a pattern. A list is limited to max 44 addresses/patterns.

IMPORTANT: Creating a MAC authentication list does *not* in itself activate filtering of addresses. Port access is managed in the VLAN configuration. See sections 13.2 and 13.3.4. The created MAC authentication list must be referenced from the port access configuration for it to be used!

21.2.18 Add a new MAC authentication list

Menu path: Configuration ⇒ AAA ⇒ MAC Auth ⇒ New List



See section 21.2.17 for descriptions of the fields on this page.

21.3 Managing AAA via the CLI

The table below shows AAA management features available via the CLI.

Command	Default	Section
Account management (Login)		
aaa		Section 21.3.1
username <USERNAME> [hash] <PASSWORD>		Section 21.3.2
Local User Database Lists (PPP, . . .)		
aaa		
local-db <ID> [plain]		Section 21.3.3
[no] username <USERNAME><PASSWORD>		Section 21.3.4
[no] description <STRING>		Section 21.3.5
Configure Remote (RADIUS) Server Connectors		
aaa		
[no] remote-server <ID> [type <TYPE>]		Section 21.3.6
type <TYPE>	radius	Section 21.3.7
[no] description <STRING>		Section 21.3.8
[no] address <IP | FQDN>		Section 21.3.9
[no] password <PASSWORD>		Section 21.3.10
[no] auth-port <PORT>	1812	Section 21.3.11
Configure (RADIUS) Server Groups		
aaa		
[no] server-group <GID> [type <TYPE>]		Section 21.3.12
type <TYPE>	radius	Section 21.3.13
[no] description <STRING>		Section 21.3.14
[no] server <ID|ID,ID|ID,ID,ID>		Section 21.3.15
Configure IEEE 802.1X Authentication		
aaa		
[no] dot1x-auth <ID>		Section 21.3.16
[no] enable	Enabled	Section 21.3.17

Continued on next page

Continued from previous page

Command	Default	Section
[no] description <STRING>		Section 21.3.18
[no] method <group <GID>|server <ID>>		Section 21.3.19
Configure MAC Authentication Lists		
aaa		
[no] mac-auth <ID>		Section 21.3.20
[no] enable	Enabled	Section 21.3.21
[no] description <STRING>		Section 21.3.22
[no] mac match <MAC-PATTERN> [limit <PORT>] [description <STRING>]		Section 21.3.23

21.3.1 Manage AAA Settings

Syntax aaa

Context Global Configuration

Usage Enter AAA Configuration context (Authentication, Authorisation and Accounting). The AAA context is used for managing user account settings, etc.

Use **"show aaa"** to show all configured AAA settings: list the local users and any configured remote servers, server groups, IEEE 802.1X authentications and MAC authentications.

Default values Not applicable.

21.3.2 Changing Account Password

Syntax `username <USERNAME> [hash] <PASSWORD>`

Context *AAA* context

Usage Change password of a certain user account, e.g., the **"admin"** account. By default, the password is entered as clear-text, and saved as a hash.

The **”hash”** keyword is not intended to be used by regular users - instead it is used by the switch itself when reading a configuration file including a hashed password. By adding the **”hash”** keyword, the system expects that a hashed password is entered (as opposed to a clear-text password).

Use **”show username <USERNAME>”** to show the hashed password for the specified user.

Default values Password is entered in clear-text.

Example Setting the **”admin”** password to ”foobar”.

Example

```
example:/config/aaa/#> username admin foobar
example:/config/aaa/#>
```

21.3.3 Manage Local User Database Lists

Syntax `[no] local-db <ID> [<TYPE>]`

Context *AAA* context

Usage Enter Local User Database Configuration context to create, modify or remove a local user database.

Use **”local-db <ID>”** to create a local database, or to enter the configuration context of an existing database. **”ID”** must be a number greater or equal to 0 and is referenced from other commands. As of WeOS v4.17.1, you can specify up to 4 local databases.

An optional **”TYPE”** parameter is used to specify how passwords within the database are stored. The only supported type in the current version of WeOS is **”plain”**, which means that all passwords are stored as plain text.

Use **”no local-db <ID>”** to remove a specific database, or **”no local-db”** to remove all configured databases.

To list all configured databases, use **”show local-db”**.

Default values The **”TYPE”** parameter is **”plain”** by default.

21.3.4 Add/Delete User in Local Database List

Syntax `[no] username <USERNAME> <SECRET>`

Context Local User Database Configuration context

Usage Add or remove users to or from the database.
Use **"username <USERNAME> <SECRET>"** to add a new user called **"USERNAME"**, whose password is **"SECRET"**.

Use **"no username <USERNAME>"** to remove a specific user from the database.

To list all the users in the database, use **"show username"**. To show the credentials of a particular user, use **"show username <USERNAME>"**.

Default values Not Applicable.

Examples

Example

```
example:/config/aaa/local-db-0/#> username alpha foobar
example:/config/aaa/local-db-0/#>
```

21.3.5 Local Database List Description Setting

Syntax `[no] description <STRING>`

Context Local User Database Configuration context

Usage Set or remove the local user database description string.
Use **"description <STRING>"** to set a description for this database.

Use **"no description"** to remove the current description.
Use citation marks around the string if you want to have a description containing space characters.

To view the current description, use **"show description"**.

Default values Empty.

Examples

Example

```
example:/config/aaa/local-db-0/#>  description PPPUsers
 or ...
example:/config/aaa/local-db-0/#>  description ''PPP Users''
```

21.3.6 Manage Remote (RADIUS) Server Connectors

Syntax `[no] remote-server <ID> [type <TYPE>]`

Context AAA Configuration context

Usage Enter Remote Server Configuration context to create, modify or remove a RADIUS server connector.

Use **"remote-server <ID>"** to create a new connector, or to enter the configuration context of an existing connector. **"ID"** must be a number greater or equal to 0 and is referenced from other commands. As of WeOS v4.17.1, you can specify up to 6 server connectors.

An optional **"type"** parameter is used to specify the type of server. The only supported type in the current version of WeOS is **"radius"**.

Use **"no remote-server <ID>"** to remove a specific server, or **"no remote-server"** to remove all configured servers.

Use **"show remote-server"** to list all configured connectors, or **"show remote-server <ID>"** to show information on a specific connector.

Default values The **"type"** parameter is **"radius"** by default.

21.3.7 Set Remote Server Type

Syntax `type <TYPE>`

Context Remote Server Configuration context

Usage Set the remote server type.

Use this command to specify the type of a remote server connector. As of WeOS v4.17.1, the only supported type is **"radius"**.

Use **"show type"** to show the configured remote server type.

Default values **"radius"**

21.3.8 Configure Remote Server Description

Syntax `[no] description <STRING>`

Context Remote Server Configuration context

Usage Set or remove the remote server description string.

Use **"description <STRING>"** to set a description for this server or **"no description"** to remove the current description.
Use citation marks around the string if you want to have a description containing space characters.

Use **"show description"** to show the configured remote server description.

Default values Empty.

Examples

Example

```
example:/config/aaa/remote-server-0/#> description MyRadius
 or ...
example:/config/aaa/remote-server-0/#> description ''Backup server''
```

21.3.9 Configure Remote Server Address

Syntax `[no] address <IP | FQDN>`

Context Remote Server Configuration context

Usage Set or remove the remote server address.

Use this command to point out the (RADIUS) server address. You can use an IP address or a name. Names will be looked up via DNS.

Use **"show address"** to show the configured remote server address.

Default values Empty. This will reject authentication for the services using this server.

Examples

Example

```
example:/config/aaa/remote-server-0/#> address 1.2.3.4
 or ...
example:/config/aaa/remote-server-0/#> address myserver.mydomain.se
```

21.3.10 Configure Remote Server Password

Syntax `[no] password <PASSWORD>`

Context Remote Server Configuration context

Usage Set or remove the remote server password.

Use this command to set the shared secret password to use with this server. This is used in RADIUS to hash passwords that are sent in the protocol exchange. The hashing is using the MD5 algorithm and that is no longer considered to be secure to attacks.

It is also only used for exchanged passwords and not for other data. Consider setting up a VPN tunnel if you need a secure way to communicate to the remote server.

Use **"show password"** to show the configured remote server password setting.

Default values Empty. No hashing will be used for passwords.

21.3.11 Configure Remote Server Authentication Port

Syntax `[no] auth-port <PORT>`

Context Remote Server Configuration context

Usage Set the UDP port number used when communicating with the remote server.

The default value for RADIUS authentication requests is to use the UDP port 1812, but you can override it here. **"no port"** will reset the value to the standard port number 1812.

Use **"show auth-port"** to show the configured UDP port used for authentication requests to the server.

Default values 1812

21.3.12 Manage (RADIUS) Server Groups

Syntax `[no] server-group <GID> [type <TYPE>]`

Context AAA Configuration context

Usage Enter Server Group Configuration context to create, modify or remove a RADIUS server group.

Use **”server-group <GID>”** to create a new group, or to enter the configuration context of an existing group. **”GID”** must be a number greater or equal to 0 and is referenced from other commands.

An optional **”type”** parameter is used to specify the type of server. The only supported type in the current version of WeOS is **”radius”**. You can specify up to 2 server groups in this version of WeOS.

Use **”no server-group <GID>”** to remove a specific group, or **”no server-group”** to remove all configured groups.

Use **”show server-group”** to list all configured server groups, or **”show server-group <GID>”** to show information on a specific server group (also available as **”show”** command within the Server Group Configuration context).

Default values The **”type”** parameter is **”radius”** by default.

21.3.13 Set Server Group Type

Syntax `type <TYPE>`

Context Server Group Configuration context

Usage Set the server group type.

Use this command to specify the type of the servers included in the group. The only supported type in the current version of WeOS is **”radius”**.

Use **”show type”** to show the configured server group type.

Default values **”radius”**

21.3.14 Configure Server Group Description

Syntax `[no] description <STRING>`

Context Server Group Configuration context

Usage Set or remove the server group description string.

Use **"description <STRING>"** to set a description for this group or **"no description"** to remove the current description. Use citation marks around the string if you want to have a description containing space characters.

Use **"show description"** to show the configured server group description.

Default values Empty.

Examples

Example

```
example:/config/aaa/server-group-0/#> description MyGroup
 or ...
example:/config/aaa/server-group-0/#> description ''Backup servers''
```

21.3.15 Configure Server Group Members

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

Context Server Group Configuration context

Usage Set the server(s) that are included in the server group.

Use this command to specify which servers belong to this server group. You can specify up to three servers comma separated by their remote server ID. Each server must be configured separately before the group is set up. See section 21.3.6.

Note

The order of the servers IS important and is used as fall-back order. The first (leftmost) defined server in the group is queried first. If the first server returns an error or does not reply the second is queried and so on.

Use **”show server”** to show the configured members of the server group (listed order is fall-back order).

Default values Empty. This will reject authentication for the services using this group.

21.3.16 Manage IEEE 802.1X authentication instances

Syntax `[no] dot1x-auth <ID>`

Context AAA Configuration context

Usage Enter 802.1X Configuration context to create, modify or remove an IEEE 802.1X authentication instance.

Use **”dot1x-auth <ID>”** to create a new 802.1X authentication instance, or to enter the configuration context of an existing instance. (As of WeOS v4.17.1 you can only create one 802.1X authentication instance.) **”ID”** must be a number greater or equal to 0 and is referenced from other commands.

> **Important**
> Creating an IEEE 802.1X authentication instance does *not* in itself activate authentication. Port access is managed in the VLAN configuration. See section 13.2. The created 802.1X instance must be referenced from the port access configuration for it to be used!

Use **”no dot1x-auth <ID>”** to remove a specific instance, or **”no dot1x-auth”** to remove all 802.1X instances.

Use **”show dot1x-auth”** to list all 802.1X authentication instances, or **”show dot1x-auth <ID>”** to show information on a specific instance (also available as **”show”** command within the 802.1X Configuration context).

Default values Not applicable.

21.3.17 Enable/Disable an IEEE 802.1X authentication instance

Syntax `[no] enable`

Context 802.1X Configuration context

Usage Enable or disable an 802.1X authentication instance.

Use **”no enable”** to disable.

Use **”show enable”** to show whether this instance is enabled or disabled.

Default values Enabled.

21.3.18 Set IEEE 802.1X authentication instance description

Syntax `[no] description <STRING>`

Context 802.1X Configuration context

Usage Set or remove the description string for this 802.1X authentication instance.

Use **”description <STRING>”** to set a description or **”no description”** to remove the current description. Use citation marks around the string if you want to have a description containing space characters.

Use **”show description”** to show the configured instance description setting.

Default values Empty.

Examples

Example

```
example:/config/aaa/dot1x-auth-0/#> description My_1X_net
 or ...
example:/config/aaa/dot1x-auth-0/#> description ''Employees only''
```

21.3.19 Configure IEEE 802.1X authentication back-end servers

Syntax `[no] method <group <GID>|server <ID>>`

Context 802.1X Configuration context

Usage Set or remove the back-end method for the 802.1X authentication instance.

IEEE 802.1X commonly use the RADIUS protocol as back-end. A RADIUS server connection or a server group must be configured separately before you can use the method command. See sections 21.3.6 and 21.3.12.

Use the syntax **”method group <GID>”** to select a specific RADIUS server group as back-end.

Use the syntax **”method server <ID>”** to select a specific RADIUS server as back-end.

Use **”no method”** to remove the back-end selection setting.

Use **”show method”** to show the ID/GID of the configured back-end server or back-end server group.

Default values No backend. 802.1X authentication attempts will fail.

21.3.20 Manage MAC authentication lists

Syntax `[no] mac-auth <ID>`

Context AAA Configuration context

Usage Create, modify or remove a MAC authentication list.

Use **”mac-auth <ID>”** to create a new list, or to enter the configuration context of an existing list. **”ID”** must be a number greater or equal to 0 and is referenced from other commands. As of WeOS v4.17.1, you can create up to 8 MAC authentication lists.

> **Important**
> Creating a MAC authentication list does *not* in itself activate filtering of addresses. Port access is managed in the VLAN configuration. See section 13.2. The created MAC authentication list must be referenced from the port access configuration for it to be used!

Use **”no mac-auth <ID>”** to remove a specific list, or **”no mac-auth”** to remove all configured MAC authentication lists.

Use **”show mac-auth”** to list all MAC authentication lists, or **”show mac-auth <ID>”** to show information on a specific instance (also available as **”show”** command within the MAC Authentication List Configuration context).

Default values Not applicable.

21.3.21 Enable/Disable a MAC authentication list

Syntax `[no] enable`

Context MAC Authentication List Configuration context

Usage Enable or disable a MAC authentication list.

Use **”no enable”** to disable.

Use **”show enable”** to show whether this list is enabled or disabled.

Default values Enabled.

21.3.22 Set MAC authentication list description

Syntax `[no] description <STRING>`

Context MAC Authentication List Configuration context

Usage Set or remove the description string for this list.

Use **”description <STRING>”** to set a description or **”no description”** to remove the current description. Use citation marks around the string if you want to have a description containing space characters.

Use **”show description”** to show the configured list description setting.

Default values Empty.

Examples

Example

```
example:/config/aaa/mac-auth-0/#> description MyMACList
 or ...
example:/config/aaa/mac-auth-0/#> description ''Trusted MAC addresses''
```

21.3.23 Configure MAC authentication list filters

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

Context MAC Authentication List Configuration context

Usage Add or remove MAC filter patterns.

A MAC Authentication List is built up by MAC filter patterns. Use the syntax **”mac match <MAC-PATTERN>”** to create a new filter pattern. To match a single MAC address specify the hardware Ethernet MAC in the format *HH:HH:HH:HH:HH:HH* as <MAC-PATTERN>. You can also specify whole blocks of addresses by using a wild-card * at the end of the pattern. You can also optionally filter on the port by using the **”limit”** argument to this command. A comment may also be added with the optional **”description”** argument.

Use **”no mac match <MAC-PATTERN>”** to remove a specific filter, or **”no mac”** to remove all filters in this list.

As of WeOS v4.17.1, you can create up to 44 MAC filter patterns per MAC authentication list.

Use **”show mac”** to show the defined MAC filter rules for this authentication list.

Default values Empty, no filters.

Examples

Example

```
mac-auth-0/#>  mac match 00:D8:AA:2C:85:01
 or with wildcard...
mac-auth-0/#>  mac match 00:80:C8:*
 or with wildcard, limit filter, and description ...
mac-auth-0/#>  mac match 00:D8:BB:C5:* limit 1/2 description ''Laser printers on 1/2''
```

Chapter 22

DHCP Server

The WeOS DHCP server is capable of handing out IP settings to hosts (DHCP clients) on *local* and *remote* IP subnets. For each defined IP subnet, the DHCP server can assign IP addresses dynamically from a *pool* of addresses, but also statically based on

- the *port* the (DHCP) client is connected to (”one IP per port”, DHCP option 82),
- the DHCP *client identifier* provided by the connecting client, or
- the *MAC address* of the connecting client

To serve clients on remote IP subnets, DHCP relay agents would be used to forward the DHCP messages between the clients and the DHCP server. In WeOS you can even configure a DHCP relay agent on the same unit as the DHCP server – this is useful if you wish to hand out addresses per port (DHCP option 82) on the DHCP server unit itself. For more information on configuring DHCP relay agents, see chapter 23.

The WeOS DHCP server is also able to act as a (proxy) DNS server for the DHCP clients it serves (see section 19.3.4).

Being part of an embedded system, the WeOS DHCP server does *not* store the current set of leases in persistent storage. In most use cases this is fine, however if it necessary that the current lease table survives a reboot you are recommended to use a dedicated DHCP server instead.

22.1 Overview of DHCP Server Support in WeOS

Table 22.1 presents a summary of DHCP server functionality in WeOS.

Feature	Web	CLI	General Description
General DHCP Server Functionality			
Enable/disable DHCP Server	X	X	
Define subnets to serve	X	X	Sections 22.1.1-22.1.2
Caching DNS server		X	Section 22.1.1
Enable/Disable Ping check	X	X	Section 22.1.3
Server Listening UDP Port		X	-”-
Server Source UDP Port		X	-”-
Per Subnet Functionality			
Client IP settings			
Address pool	X	X	Section 22.1.2
Per port (Option 82)	X	X	-”-
Per client-ID	X	X	-”-
Per MAC	X	X	-”-
Additional client configuration parameters			
Default Gateway	X	X	-”-
DNS Server	X	X	-”-
Domain search path	X	X	-”-
NTP Server	X	X	-”-
Other features			
Define lease time	X	X	-”-
Deny client (per MAC)	X	X	-”-
DHCP Server Status			
List current clients		X	

Table 22.1: DHCP server features

22.1.1 Introduction to WeOS DHCP server support

DHCP servers are typically used to dynamically assign IP settings (IP address, netmask, default gateway, etc.) to hosts on the local subnet, see fig. 22.1a. The

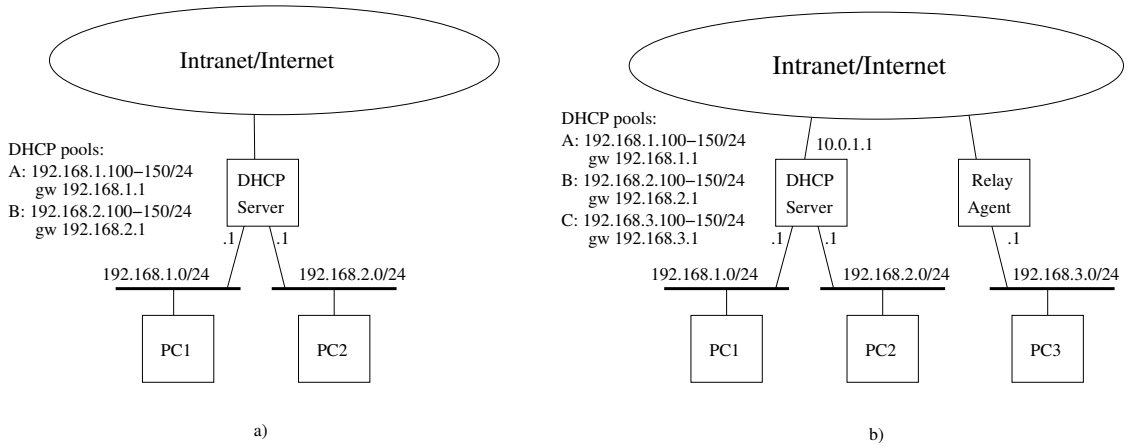


Figure 22.1: Sample DHCP use cases: (a) DHCP server serving local subnets, and (b) serving local and remote subnets.

server maintains an *address pool* for each served subnet, from which it assigns addresses to DHCP clients currently present on that LAN. Addresses in the pool are maintained dynamically - they are assigned to clients for a configurable time (DHCP *lease time*), and if a client goes away, that address can be reused and assigned to another client.

The DHCP server also hands out configuration settings for *default gateway* and *DNS server(s)*. For local clients as in fig. 22.1a, the DHCP server unit will commonly act as default gateway and DNS server[1] too.

To provide DHCP service on multiple subnets throughout your infrastructure, you could either deploy a DHCP server on each subnet, or you could use DHCP *relay agents* to forward DHCP packets between the remote subnet and a central DHCP server, as shown in fig. 22.1b.

When configuring the server, there is no major difference if the subnet is local or remote – you will simply define which subnets to serve. When the server receives a DHCP message, it will automatically detect which subnet the request originated from and thereby be able to hand out an address from the pool it has defined for that subnet.

In addition to handing out addresses dynamically from a pool, it is also possible to assign addresses more *specifically* based on the client's *MAC address*, the *client identifier* (client-ID) included in the DHCP messages from the client, or the

[1] A WeOS unit acts as (proxy) DNS server by default, see section 19.3.4

physical port where the client is connected. More information on this is given in sections 22.1.2 and 22.1.4.

The DHCP server unit will by default accept incoming DHCP packets on any of its interfaces, including the loopback interface **”lo”**. (The exception is those interface where a DHCP relay agent has been configured on the local unit (see section 22.1.4) – there DHCP packets will be handled by the relay agent.)

Hint

For security purposes you may wish to avoid accepting DHCP packets on some interfaces, e.g., your upstream interface towards the Internet. To block such request you are recommended to configure appropriate *deny* filter rules, e.g., **”filter deny in vlan1 dport 67 proto udp”** to block incoming DHCP request on interface *vlan1*. For more details on the WeOS firewall, see chapter 31.

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.

22.1.2 Per-subnet DHCP Server Settings

Most DHCP server settings are configured per subnet, where the IP subnet is defined by an IP address (e.g., 10.10.2.0) and subnet mask, which defaults to 255.255.255.0 (/24). For each subnet you can define what IP address to assign to clients, as well as other relevant IP settings.

22.1.2.1 Defining IP Address assignment

The addresses can either be assigned *dynamically* from an *address pool*, or be assigned statically depending on the client’s MAC, its DHCP client identifier, or the port to which it is connected.

- *Address pool:* For each subnet served it is possible to define a pool of addresses for dynamic assignment. The default range is ”100-199”, e.g., 10.10.2.100-10.10.2.199 on the 10.10.2.0/24 subnet.

  It is possible to disable dynamic address allocation using the **”no pool”** syntax in the CLI. This is mostly useful in combination with fixed assignment.

- *Fixed assignments:* Instead of handing out addresses from a dynamic pool, the WeOS DHCP server enables you to assign addresses with more fine grain control:
  - *Client MAC:* You can reserve a specific address to a client with a certain MAC address.
  - *Client identifier (option 61):* You can reserve a specific address to a client including a certain *client-identifier* in its DHCP messages (DHCP option 61[1]). In the DHCP server, you can specify the client-id as a hexadecimal sequence (e.g., ”01485b392f34bc”) or as a text string such as ”foobar”.

    Note
    If the client-id is specified as a text string, it would match a DHCP option 61 holding a hexadecimal sequence of the corresponding ASCII numbers[a], e.g., ”foobar” would match an option 61 holding value ”666f6f626172” (hex).

    [a] American Standard Code for Information Interchange (ASCII), see e.g. http://en.wikipedia.org/wiki/ASCII (accessed May 2009).
  - *Connected Port (option 82):* The server can be configured to assign a specific address to the client connected to a certain switch port (”one IP per port”). This is useful when you wish to replace a client unit, such as a CCTV camera, and ensure that the new unit gets the same IP as the replaced unit.

    As described in chapter 23, DHCP relay agents can add information to identify the client’s port in a relay information option (DHCP option 82[28]). The DHCP server can then extract relevant information (*circuit-id* and *remote-id*) and use that when assigning the IP address.

    WeOS DHCP server allows for flexible specification of *circuit-id* and *remote-id* (both as hexadecimal sequences and text strings), enabling it to work with relay agents of various vendors. E.g., to make the DHCP server hand out a specific IP address to a client unit attached to WeOS Relay Agent with default settings, the DHCP server can be configured as follows:

    * *Circuit-id:* If the client is supposed to connect to Ethernet port 2, then specify **”Eth2”** (string) for the circuit-id. If a slotted WeOS product is used, then specify e.g., **”Eth3/5”** for Ethernet port 5 on slot 3.

  * *Remote-id:* The remote-id is optional, but needed to distinguish between relay agents on the same subnet. A WeOS relay agent defaults to using its *base MAC*[2] address as remote-id.. E.g., specify **"00077c8209d0"** (hex) for a WeOS relay agent with base MAC *00:07:7c:82:09:d0*.

    Note: to assign IP addresses per (local) ports on the DHCP server itself in WeOS v4.17.1, you will need to setup a Relay Agent on the same unit (see section 22.1.4).

- *Deny statements:* The fixed assignment methods (MAC, Client-id, Option 82) can also be used to *deny* clients an IP address. To specify this feature, use the keyword **"deny"** instead of an IP address in the assignment command.

A note on preference order

A client request associated with a subnet served by the DHCP server will be checked for matching IP assignment entries in the following preference order[a]: *Client-Id* (first), *MAC address*, *DHCP Option 82*, and finally *Address pool* (last).

[a] This preference order is used as of WeOS v4.17.1, but may be changed in future releases.

22.1.2.2 Configuration Options other than IP address

In addition to IP address, the WeOS DHCP server allows you configure the following configuration options:

- *Lease time:* The lease time can be configured in range 120-5256000 seconds. It defaults to 864000 seconds (10 days).
- *Netmask:* The IP netmask is only configured implicitly, i.e., it is taken from the subnet definition. IP netmask is passed to the client in DHCP option 1. By default, the netmask is set to *255.255.255.0*.
- *Router IP address:* The DHCP server will pass information about what router (default gateway) the DHCP client should use. If you leave this blank, the will automatically fill out a value likely to work for the client.

[2] To find the base MAC of your WeOS unit, see sections 4.4.2 (Web) or 7.3.2 (CLI).

  - *Local clients:* For DHCP requests originating on the local subnets, the DHCP server will put its own IP address on that subnet as *gateway* IP address.
  - *Remote clients:* For DHCP requests originating on remote subnets, the DHCP server will put the IP address of the relay agent as *gateway* IP address.

  The router/gateway IP is passed in DHCP option 3. By default, the gateway setting is empty, i.e., the ”auto” behaviour described above is used.

  As of WeOS v4.17.1 there is no way to hinder the DHCP server to send the router/gateway IP address (option 3). This may change in future WeOS releases.

- *DNS Server(s):* It is possible to specify up to two DNS servers to be passed to the DHCP client (option 6). If no DNS server is specified, the DHCP server will fill in its own IP address as DNS server (the DHCP server unit will act as DNS forwarder and forward any (non-cached) incoming DNS requests to the name-server(s) configured on the unit, see chapter 19).

  As of WeOS v4.17.1 there is no way to hinder the DHCP server to send the Domain Name Server option (option 6) to the client. This may change in future WeOS releases.

- *Domain search path:* The DHCP server can be configured to pass a *domain search path* to the DHCP client (option 15). (Leaving the setting empty implies that no domain search path is sent to the client.)

- *NTP Server(s):* The DHCP server can be configured to pass up to two *NTP Servers* to the DHCP client (option 42). (Leaving the setting empty implies that no NTP server is sent to the client.)

22.1.3 General DHCP Server settings

WeOS allows you to configure a set of general DHCP server settings. These are advanced settings, and are primarily of interest to users with special demands. The default values are sufficient in almost all use cases.

- *DHCP Server Listening UDP port:* The DHCP server listens to UDP port 67 by default (in-line with RFC2131[7]). It is possible to set the server port to a different value. That may be of interest in some specific DHCP relay setups,

to avoid that the server receives packets directly from clients (in addition to *relayed* packets).

> **Note**
> It is possible to configure the WeOS relay agent to forward DHCP messages to non-standard UDP ports on the server, see chapter 23.

- *DHCP Server Source UDP port (client port):* The DHCP server will send packets with source UDP port 68 by default. It is possible to set the source UDP port to a non-default value.
- *Enable/disable Duplicate Address Detection (ICMP Ping Check)*: Before a DHCP server offers a client an address it will check that no-one is already using that address. The server conducts this *duplicate address detection* mechanism by attempting to ”ping” the IP address a couple of time to verify that it gets no response. Disabling ”ping check” can speed up address assignment.

  The ”ping check” mechanism is recommended for robustness and is enabled by default. Only consider disabling ”ping-check” if you are using static leases.

  > **Warning**
  > The WeOS DHCP server does **not** store the lease table in persistent storage. Disabling ”ping check” can therefore lead to situations where a server reboot causes a host to be assigned an address, which was already assigned to (and in use by) another host.

22.1.4 Running a DHCP server and relay agent on the same unit

There are situations when you wish to run a DHCP relay agent (chapter 23) on the same WeOS unit as your DHCP server.

- **IP per port on DHCP server unit:** Section 22.1.4.1 describes how to use a DHCP server and a relay agent to assign IP addresses per port on the DHCP server unit itself.
- **Non-”DHCP snooping” relay agents in switched topologies:** Section 22.1.4.2 explains how to handle non-”DHCP snooping” relay agents in *switched* (as opposed to *routed*) topologies. (An alternative approach is to let the DHCP server listen to a non-default UDP port, see section 22.1.3.)

22.1.4.1 IP per port on local DHCP server ports

With DHCP option 82, a relay agent can inform the DHCP server which port (circuit-id) the client is connected to, thereby enabling the server to assign IP addresses per port. In WeOS, the same approach is used when you wish to hand out IP addresses per port on the DHCP server's local ports.

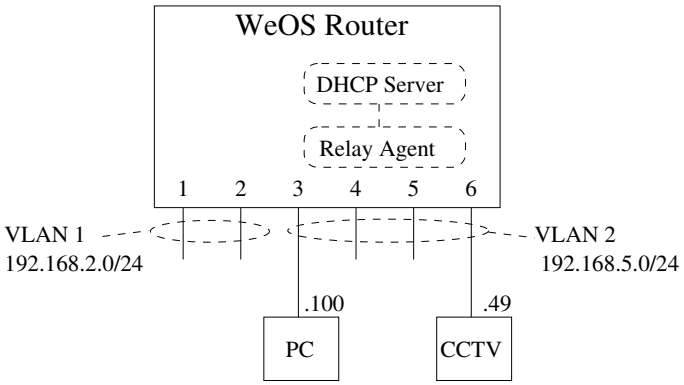


Figure 22.2: Running both a DHCP Server and a DHCP Relay Agent on the same unit enables you to assign IP address per port on the DHCP server unit.

Fig. 22.2 illustrates an example where the WeOS unit is configured to hand out addresses on interface **"vlan2"** (*subnet 192.168.5.0/24*). Regular hosts, such as the PC, will be assigned their IP addresses from an address pool, but the unit attached to port 6 should always be assigned IP address *192.168.5.49*. This can be achieved by configuring a DHCP relay agent on interface **"vlan2"**, and to instruct the relay agent to forward DHCP request to the local DHCP server (address **"127.0.0.1"**). Relevant parts of the WeOS configuration is listed in fig. 22.3.

The WeOS DHCP relay will by default pass its *base-MAC address*[3] as *remote-id* (**"00:07:7c:00:30:b0"** in the configuration example in fig. 22.3.). As the base-MAC is unit specific, this setting will not work if you wish to replace the unit, but keep the same configuration file. In such situations, using **"system-name"** or **"ip"** as remote-id is recommended, see sections 23.2.1 (Web) and 23.3.9 (CLI) for more information. An example using the system name as remote-id is given in fig. 22.4.

[3] To find the base MAC of your WeOS unit, see sections 4.4.2 (Web) or 7.3.2 (CLI).

Example

```
dhcp-server
        subnet 192.168.5.0/24
                pool 192.168.5.100 192.168.5.199
                lease-time 864000
                netmask 255.255.255.0
                no gateway
                no domain
                end
        host 1
                match option82 circuit-id string "Eth6" remote-id hex 00:07:7c:00:30:b0
                address 192.168.5.49
                end
        end

dhcp-relay
        iface vlan2
        server 127.0.0.1
        option82 discard
        end
```

Figure 22.3: Configuration example with DHCP relay and server on same unit, here with base-MAC address as Option82 circuit-ID.

Example

```
system
        hostname foobar
        end

dhcp-server
        subnet 192.168.5.0/24
                pool 192.168.5.100 192.168.5.199
                lease-time 864000
                netmask 255.255.255.0
                no gateway
                no domain
                end
        host 1
                match option82 circuit-id string "Eth6" remote-id string "foobar"
                address 192.168.2.49
                end
        end

dhcp-relay
        iface vlan2
        server 127.0.0.1
        option82 discard
        remoteid-type system-name
        end
```

Figure 22.4: Configuration example with DHCP relay and server on same unit, here with system hostname as Option82 circuit-ID.

22.1.4.2 Handling non-snooping relay agents in switched topologies

As described in section 23.1.4, use of relay agents to add option 82 information in *switched topologies* is challenging if the relay agents do not support DHCP snooping. A (broadcast) DHCP message from a client will then result in two messages being forwarded towards the DHCP server - one *relayed* message including option 82 information, and one *regular* message being *switched* and lacking option 82.

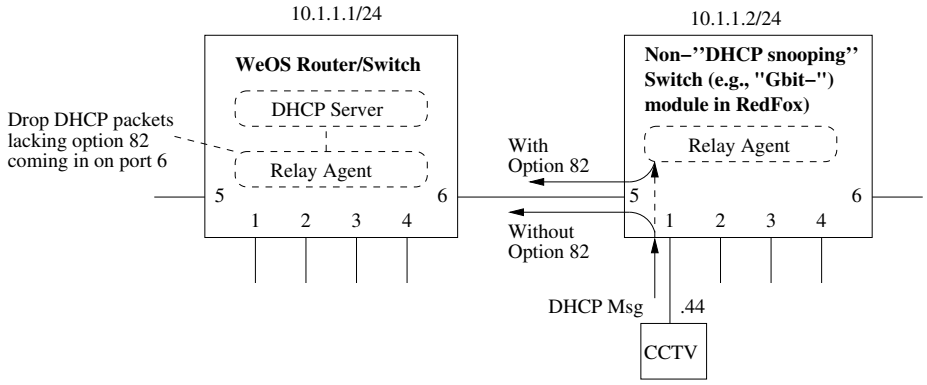


Figure 22.5: A non-"DHCP snooping" relay agent (right unit) will likely result in multiple "copies" of the DHCP messages. This can be handled by running a DHCP Relay Agent also the DHCP server unit (left unit).

Fig. 22.5 illustrates the situation. All ports are assumed to be on the same VLAN (e.g., VLAN 1)

1. A broadcast DHCP message is sent by the PC on port 1 of the non-snooping switch. That packet is forwarded onto all ports on the same VLAN including port 5 towards the DHCP server.
2. The packet is also processed by the relay agent process, which adds option 82 information and relays the message (unicast) towards the DHCP server.
3. If both DHCP requests would reach the DHCP server, it is likely that the PC will be handed an address from the pool rather than an address dedicated for that specific port. Or possibly the PC will get multiple responses to its request.

   In WeOS you can handle this by running a DHCP relay agent on the DHCP server unit. The relay agent can be configured to drop DHCP packets not including option 82, thus only the relayed packet will be forwarded to the

DHCP server process.

Below (page 498) sample configurations for the DHCP server and DHCP relay agent units are shown. The CCTV connected to port 1[4] of the (non-snooping) relay agent should be assigned IP address 10.1.1.44/24.

Hint

An alternative approach is to let the DHCP server listen to a non-default UDP port, see section 22.1.3. Then the DHCP relay agent must be configured to send to this UDP port when relaying packets to the server.

Example

```
-- DHCP Server Unit (IP 10.1.1.1/24)

dhcp-server
        subnet 10.1.1.0/24
                pool 10.1.1.100 10.1.1.199
                lease-time 864000
                netmask 255.255.255.0
                gateway 10.1.1.1
                no domain
                end
        host 1
                match option82 circuit-id string "Eth1" remote-id string "10.1.1.2"
                address 10.1.1.44
                end
        end

dhcp-relay
        iface vlan1
        server 127.0.0.1
        option82 discard
        remoteid-type ip
        port 6
                option82 require
                end
        end

-- DHCP Relay Agent Unit (IP 10.1.1.2/24)

dhcp-relay
        iface vlan1
        server 10.1.1.1
        option82 discard
        remoteid-type ip
        end
```

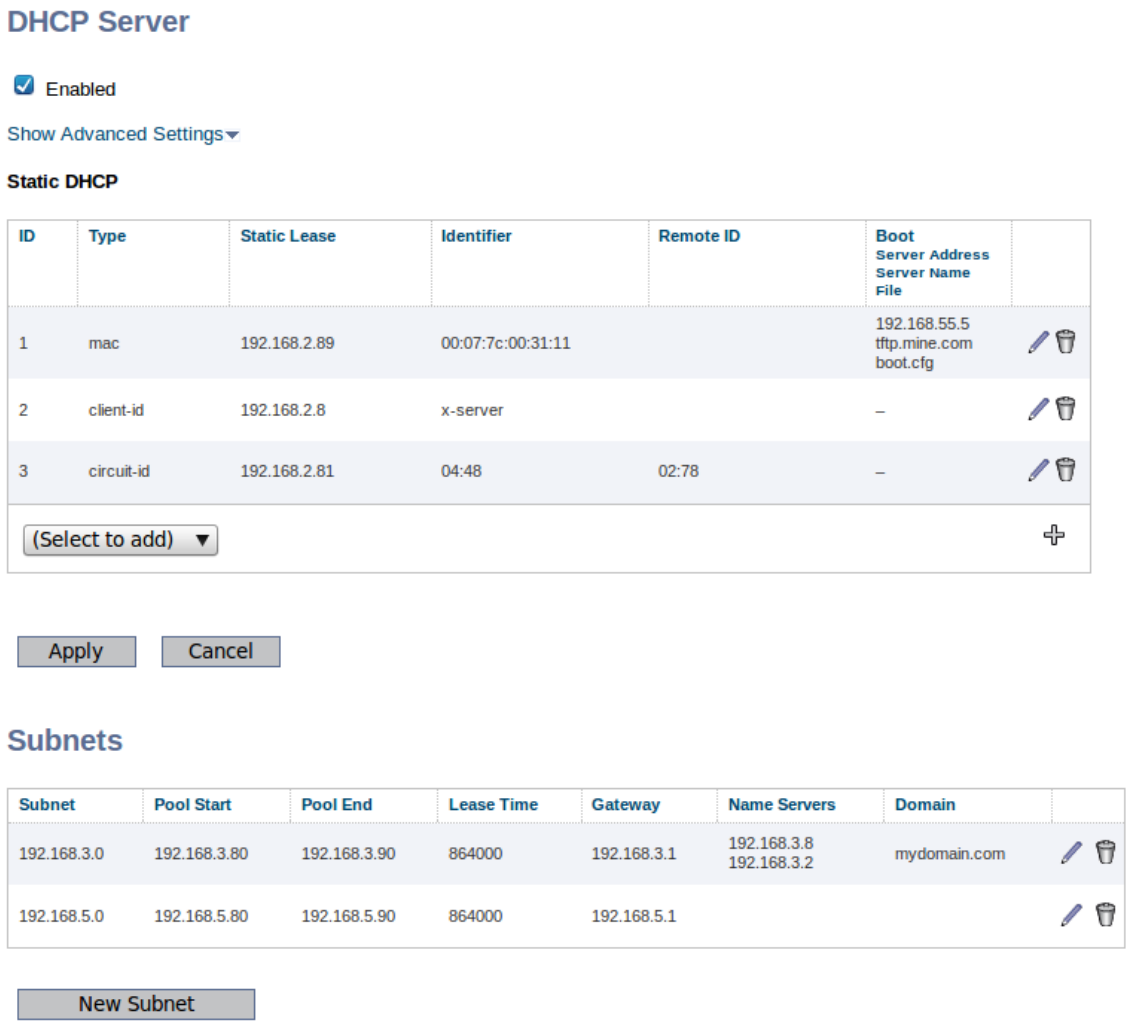
[4] If the relay agent unit is a RedFox Industrial, the port labels would be written in slot/id form (1/1, 1/2, etc.). The server configuration would then reflect this, e.g., **”match option82 circuit-id string "Eth1/2" remote-id string "10.1.1.2"”** if the CCTV is connected to port 1/2.

22.2 Configuring DHCP Server Settings via the Web

The Web interface provides management of DHCP Server.

22.2.1 DHCP Server settings

Menu path: Configuration ⇒ Network (IP) ⇒ DHCP-Server

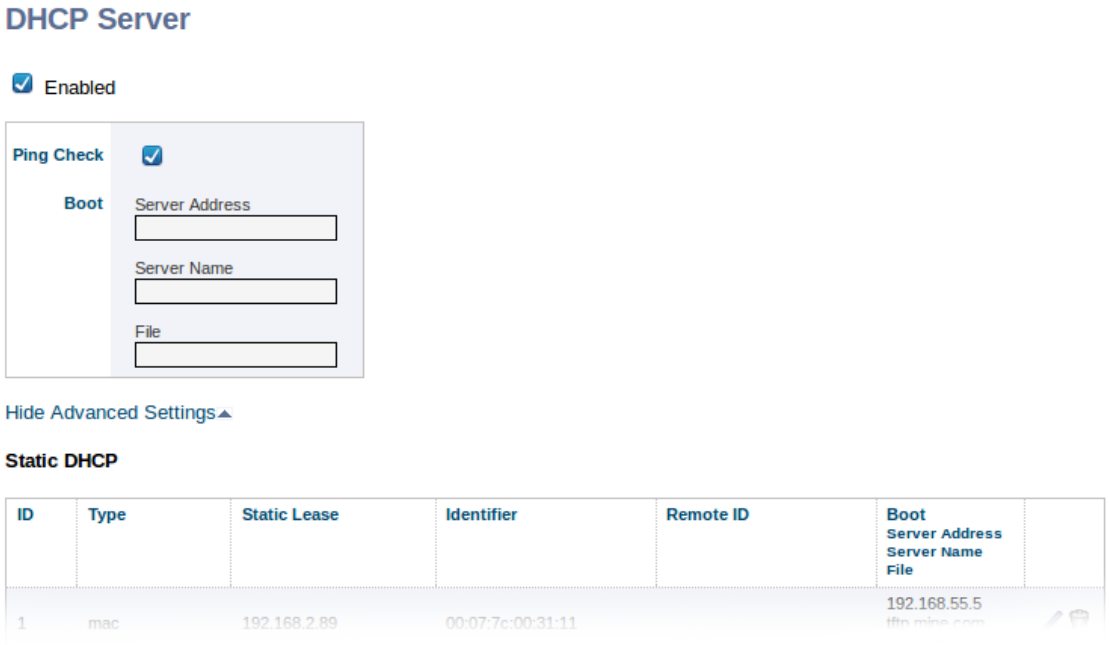


Continued on next page

	Continued from previous page
Enabled	Check the box to enable the DHCP server. If you have a JavaScript enabled browser the other settings will not be displayed unless you check this box.
Static DHCP	The static leases for this subnet. To add a lease select type (MAC, Client-id or Option82) and fill in the values. To add additional static leases, click on the Add icon (). Click on the Edit icon ()to edit the lease. The boot server-address/server-name and boot file options may be set on a static lease, overriding any common setting in the advanced server settings section.
Subnets	Lists the configured DHCP subnets To add a Subnet click on the "New subnet" button bellow the table. Click on the Edit icon ()to edit the settings for a specific Subnet.

22.2.1.1 Advanced DHCP Server settings

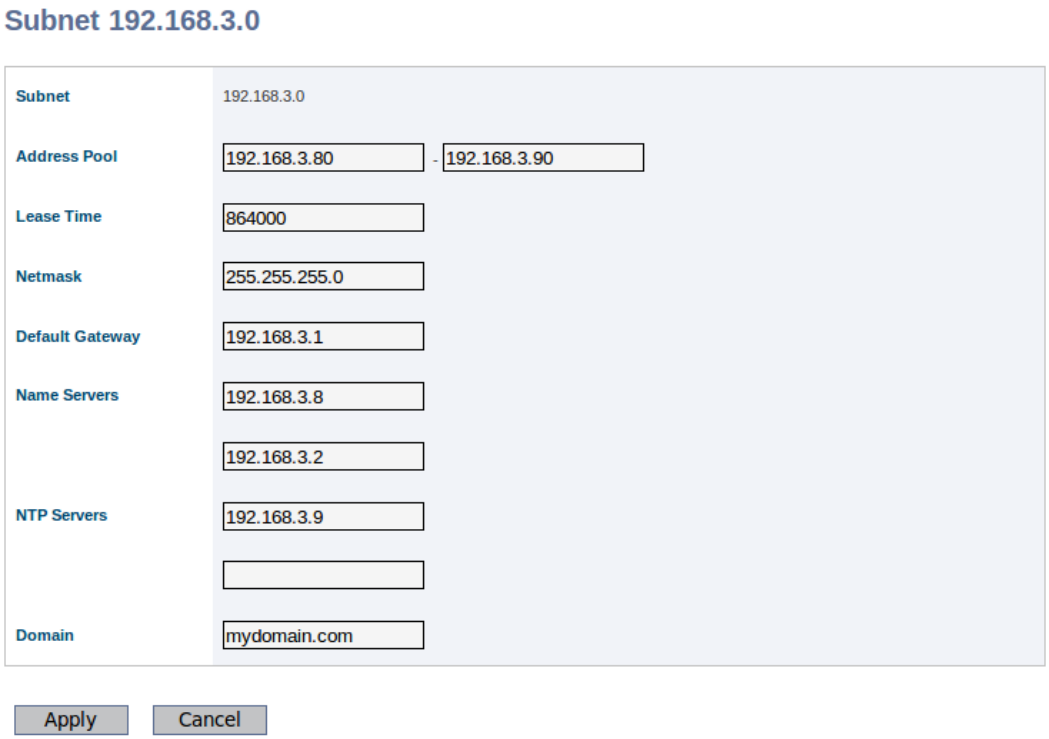
Menu path: Configuration ⇒ Network (IP) ⇒ DHCP-Server ⇒ Advanced Settings



Ping Check	Enables/disables the 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. Default enabled
Boot Server Address	IP address for server from which the client should retrieve the boot file.
Boot Server Name	DNS name for server from which the client should retrieve the boot file.
Boot File	Name of the boot file to retrieve from boot server.

22.2.2 Edit DHCP Subnet Settings

Menu path: Configuration ⇒ Network (IP) ⇒ DHCP-Server ⇒ 



On this page you can change the settings for the Subnet.

Address Pool	IP address pool from which the DHCP server will hand out leases
Lease Time	DHCP address lease time (seconds) for addresses handed out to DHCP clients
Netmask	The netmask option handed to DHCP clients.
Default Gateway	The IP default gateway (default router) option handed to DHCP clients.
Name Servers	The (DNS) name server option handed to DHCP clients.
NTP Servers	The time server (NTP) option handed to DHCP clients.
Domain	Domain name search path option handed to DHCP clients

22.3 Configuring DHCP Server Settings via the CLI

Command	Default	Section
Configure DHCP Server		
[no] dhcp-server	Disabled	Section 22.3.1
[no] enable	Enabled	Section 22.3.2
[no] ping-check	Enabled	Section 22.3.3
[no] server-port <UDPPORT>	67	Section 22.3.4
[no] client-port <UDPPORT>	68	Section 22.3.5
[no] server-address <IPADDR>	Disabled	Section 22.3.6
[no] server-name <DOMAINNAME>	Disabled	Section 22.3.7
[no] file <FILENAME>	Disabled	Section 22.3.8
[no] host [INDEX]	1	Section 22.3.9
[no] match <mac <MACADDR> | clientid <hex|string> <CLIENTID> | option82 [remote-id <hex|string> <REMOTEID>] <circuit-id <hex|string> <CIRCUITID>>		Section 22.3.10
[no] address <IPADDR|deny>	Disabled	Section 22.3.11
[no] server-address <IPADDR>	Disabled	Section 22.3.6
[no] server-name <DOMAINNAME>	Disabled	Section 22.3.7
[no] file <FILENAME>	Disabled	Section 22.3.8
[no] subnet <IPADDR[/LEN] | IPADDR [MASK]>	/24	Section 22.3.12
[no] netmask <NETMASK>		Section 22.3.13
[no] pool <IPADDR_START> <NUM|IPADDR_END>	Auto[1]	Section 22.3.14
[no] lease-time <120-5256000>	864000	Section 22.3.15
[no] gateway <IPADDR>	Empty[2]	Section 22.3.16
[no] name-server <IPADDR>[,<IPADDR>]	Empty[2]	Section 22.3.17
[no] domain <DOMAINNAME>	Disabled	Section 22.3.18
[no] ntp-server <IPADDR>	Disabled	Section 22.3.19
View DHCP Server Status		
show dhcp-clients		Section 22.3.20

[1] A pool may be created automatically. See Section 22.3.14.
[2] Empty values have special meaning here. See Section 22.3.16 and Section 22.3.17.

22.3.1 Manage DHCP Server

Syntax `[no] dhcp-server`

Context Global Configuration context

Usage Create, modify or remove a DHCP Server.

Enter DHCP server context. If this is a new DHCP server, the DHCP server is created. As a side-effect, a *caching* (DNS) name server is started, which forwards incoming DNS requests to the DNS server configured for the switch (see chapter 19).

Use **"no dhcp-server"** to remove an existing DHCP server.

Use **"show dhcp-server"** to list all settings of a DHCP server. Alternatively, you can run the **"show"** command from within the *DHCP server* context.

Default values Disabled (No DHCP server configured)

22.3.2 Disable DHCP Server

Syntax `[no] enable`

Context DHCP Server Configuration context

Usage Enable/disable the DHCP server. Useful to disable a fully setup DHCP server before deployment, the configuration will remain dormant while disabled.

Use **"no enable"** to disable the DHCP server (without losing the DHCP server configuration) and **"enable"** to enable the DHCP server.

Use **"show enable"** to show whether the DHCP server is configured enabled or disabled.

Default values Enabled

22.3.3 Disable ICMP "ping" Check

Syntax `[no] ping-check`

Context DHCP Server Configuration context

Usage Enable/disable the 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.

Use **”no ping-check”** to disable the ping-check mechanism, and **”ping-check”** to enable it.

Run **”show ping-check”** to show whether the ping-check mechanism is configured enabled or disabled.

Default values Enabled

22.3.4 DHCP Server Listening UDP port

Syntax `[no] server-port <UDPPORT>`

Context DHCP Server Configuration context

Usage Set DHCP Server listening (UDP) port in range 1..65535. By default the server listens to UDP port 67. Use **”server-port UDPPORT”** to set a non-default UDP port to listen on. See also section 23.3.5 for the corresponding DHCP relay agent setting.

Use **”no server-port”** to reset to default value (port 67).

Use **”show server-port”** to show current server-port settings.

Default values 67

22.3.5 DHCP Server Source/Client UDP port

Syntax `[no] client-port <UDPPORT>`

Context DHCP Server Configuration context

Usage Set DHCP Server source (UDP) port in range 1..65535. By default the server sends DHCP messages with source UDP port 68. Use **”client-port UDPPORT”** to set a non-default UDP port to send from.

Use **”no client-port”** to reset to default value (port 68).

Use **”show client-port”** to show current client-port settings.

Default values 68

22.3.6 Next Server Address – BOOTP ”siaddr”

Syntax `[no] tftp-server <IPADDR>`

Context DHCP Server Configuration or DHCP Server Host Configuration context

Usage Set the next-server address *siaddr* in DHCP and BOOTP messages from the DHCP server, i.e., the IP address of a TFTP server (or other type of file transfer server) used by a BOOTP/DHCP client to retrieve a boot file.

> **Note**
> Using the **”tftp-server”** command in DHCP Server Configuration will apply to all DHCP messages from the server. Using the **”tftp-server”** command in DHCP Server Host Configuration will apply to a specific host entry.

Use **”no tftp-server”** to remove a configured next-server address.

Use **”show tftp-server”** to show the next-server setting.

Default values Disabled

22.3.7 Next Server Name – BOOTP ”sname”, DHCP Option 66

Syntax `[no] tftp-server-name <DOMAINNAME>`

Context DHCP Server Configuration or DHCP Server Host Configuration context

Usage Set the next-server domain name. It can be used to inform BOOTP/DHCP clients about their next-server to download a boot file, as an alternative to the next-server address (see section 22.3.6).

The server name 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.

> **Note**
> Using the **”tftp-server-name”** command in DHCP Server Configuration will apply to all DHCP messages from the server. Using the **”tftp-server-name”** command in DHCP Server Host Configuration will apply to a specific host entry.

Use **”no tftp-server-name”** to remove a configured next-server name.

Use **”show tftp-server-name”** to show the next-server name setting.

Default values Disabled

22.3.8 Bootfile Name – BOOTP ”file”, DHCP Option 67

Syntax `[no] bootfile <FILENAME>`

Context DHCP Server Configuration or DHCP Server Host Configuration context

Usage Set the boot filename (as stored at 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.

> **Note**
> Using the **”bootfile”** command in DHCP Server Configuration will apply to all DHCP messages from the server. Using the **”bootfile”** command in DHCP Server Host Configuration will apply to a specific host entry.

Use **”no bootfile”** to remove a configured bootfile name.

Use **”show bootfile”** to show the next-server name setting.

Default values Disabled

22.3.9 Configure Host Entry

Syntax `[no] host [INDEX]`

Context DHCP Server Host Configuration context

Usage Enter the DHCP Server Host Configuration to specify host specific DHCP Server 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 64 can be configured. Each entry is given an index (default 1), e.g., **”host 3”** will enter the DHCP Server Host Configuration for entry number 3; the entry will be created if it does not yet exist.

Use **”no host”** to remove all configured host entries, and use **”no host <INDEX>”** to remove a specific host entry (e.g. **”no host 3”**).

Use **”show host”** to show a list configured host entries, and use **”show host <INDEX>”** to show information on a specific host entry. Alternatively, you can run the **”show”** command within the DHCP Server Host Configuration context of that specific subnet.

Default values Default index is 1.

22.3.10 Configure Host Entry Match Setting

Syntax `[no] match <mac <MACADDR> | clientid <hex|string> CLIENTID> | option82 [remote-id <hex|string> <REMOTEID>] <hex|string> circuit-id <CIRCUITID>>`

Context DHCP Server Host Configuration context

Usage Specify the match type (mac, clientid or option82) to identify the host for this entry, e.g., **”match mac 12:34:56:78:9a:bc:de”**.

Use **”no match”** to remove all match settings, and **”no match <mac|clientid|option82>”** to remove a match setting of a specific type.

Use **”show match”** to show the current match setting for this host entry.

Default values Not applicable. section 22.3.12.

22.3.11 Configure Host IP Address or Deny Service

Syntax `[no] address <IPADDRESS|deny>`

Context DHCP Server Host Configuration context

Usage Specify the IP address to assign to this host, e.g., **”address 192.168.1.51”**.

> **Note**
> To hand out the specified address (e.g., **”192.168.1.51”**) the DHCP server must also be configured to serve the associated IP subnet, see section 22.3.12 for information on the **”subnet”** command. Other IP settings (netmask, default gateway, etc.) will be inherited from settings of the associated subnet.

Use **”address deny”** to prohibit the host to be served by this DHCP server.

A host must either be assigned an IP address or explicitly be denied an address. **”no address”** is not a valid setting, i.e., then the host entry will not be activated.

Use **”show address”** to show the current address setting.

Default values None

22.3.12 Configure DHCP Server Subnet

Syntax `[no] subnet <IPADDR[/LEN] | IPADDR [NETMASK]>`

Context DHCP Server Configuration context

Usage Specify a subnet for which the DHCP server will hand out IP addresses, and enter the DHCP Server Subnet Configuration for that subnet. Optionally, the subnet netmask can be specified as a prefix length or as a netmask, with **”/24”** (**”255.255.255.0”**) as default. It can later be changed with the **”netmask”** command, see section 22.3.14.

Use **”no subnet”** to remove all configured subnets, and use **”no subnet IPADDR”** to remove a specific subnet.

Use **”show subnet”** to show a list configured subnets for the DHCP server, and use **”show subnet IPADDR”** to show information on a specific subnet (alternatively, you can run the **”show”** command within the DHCP Server Configuration context of that specific subnet).

The DHCP server can handle up to 1024 subnets.

Default values Default prefix length is 24 (i.e., netmask 255.255.255.0).

22.3.13 Configure DHCP Subnet Netmask

Syntax `[no] netmask <NETMASK>`

Context DHCP Server Subnet Configuration context

Usage Specify/modify the netmask for the subnet to serve, e.g., **”netmask 255.255.128.0”**.

Use **”no netmask”** to reset the netmask to its default value.

Use **”show netmask”** to show the current netmask setting.

Default values The netmask defaults to **”255.255.255.0”**, however, a different netmask can be specified in the **”subnet”** command, see section 22.3.12.

22.3.14 Configure DHCP Server Address Pool

Syntax `[no] pool <IPADDRESS_START> <NUM|IPADDRESS_END>`

Context DHCP Server Subnet Configuration context

Usage Specify the IP address pool from which the DHCP server will hand out leases. The *end* of the address range can be specified as an IP address (**”IPADDRESS_END”**), or as a number (**”NUM”**). **”NUM”** specifies the number of addresses in the pool, thus ”IPADDRESS_END” is computed as *”IPADDRESS_START* + *NUM* − 1”.

Use **”no pool”** to disable dynamic address assignment. When disabled, only static host entries are allowed in the range defined by the subnet itself and the netmask option.

Use **”show pool”** to see the IP addresses in the pool.

Default values A pool based on the configured subnet is automatically setup when creating a new DHCP subnet.

22.3.15 Configure DHCP Server Lease Time

Syntax `[no] lease-time <120-5256000>`

Context DHCP Server Subnet Configuration context

Usage Specify the DHCP address lease time (seconds) for addresses handed out to DHCP clients.

Use **”no lease-time”** to reset the lease time setting to its default value.

Use **”show lease-time”** to show the current lease-time setting.

Default values 864000 seconds (i.e., 10 days)

22.3.16 Configure DHCP Server Default Gateway Option

Syntax `[no] gateway <IPADDRESS>`

Context DHCP Server Subnet Configuration context

Usage Specify the IP default gateway (default router) option for leases handed to DHCP clients. A single default gateway can be specified. If no default gateway is specified, the switch IP address on this interface will be provided in the default gateway option for *local DHCP clients* (that is, the switch will act as default gateway for hosts on local subnets), or the DHCP relay agent IP address for DHCP requests on remote subnets.

> **Note**
> When acting as router for local DHCP clients, please remember to enable routing on this unit (chapter 19) and enable appropriate NAT and firewall rules if necessary (chapter 31).

Use **"no gateway"** to remove any statically configured default gateway option.

Use **"show gateway"** to list the gateway option settings.

Default values Empty, this means that the switch IP address on this interface will be provided in the default gateway option.

22.3.17 Configure DHCP Server Name Server Option

Syntax `[no] name-server <IPADDRESS>[,<IPADDRESS>]`

Context DHCP Server Subnet Configuration context

Usage Specify name server (DNS) options for leases handed to DHCP clients. Up to two DNS name servers can be specified, either as comma separated IP addresses on the command line, or by repeating the command for each address.

Use **"no name-server"** to remove all configured name server DHCP options.

If no name server is specified, the switch IP address on this interface will be provided in the name server option (that is, the switch will act as DNS name server for hosts on this interface. In this case, the switch will act as

a caching name server and forward any (non-cached) incoming requests to the name-server(s) configured on the switch, see chapter 19).

Use **”show name-server”** to list DNS name server option settings.

Default values Empty, this means that the switch IP address on this interface will be provided in the name server option.

22.3.18 Configure Domain Name Option

Syntax `[no] domain <DOMAIN>`

Context DHCP Server Subnet Configuration context

Usage Specify the domain name search path option for leases handed to DHCP clients. A single domain name option can be specified.

Use **”no domain”** to disable this option.

Use **”show domain”** to list domain name option settings.

Default values Disabled, the domain name option will not be used.

22.3.19 Configure NTP Server Option (DHCP Option 42)

Syntax `[no] ntp-server <IPADDR>`

Context DHCP Server Subnet Configuration context

Usage Specify the NTP-server option (DHCP option 42) for leases handed to DHCP clients, e.g., **”ntp-server 192.168.1.3”**. Up to two NTP servers can be specified.

Use **”no ntp-server <IPADDR>”** to remove a specific NTP server, or **”no ntp-server”** to disable all NTP server options.

Use **”show ntp-server”** to list NTP-server option settings.

Default values Disabled, the NTP-server option will not be used.

22.3.20 Show list of current DHCP clients

Syntax `show dhcp-clients`

Context Admin Exec context

Usage Show list of current DHCP clients.

Default values Not applicable

Example

```
example:/#> show dhcp-clients
Lease Time  MAC Address        IP Address       Hostname         Client ID
================================================================================
864000      00:07:7c:8a:e2:41  192.168.2.109    *                01:00:07:7c:8a:e2:41
example:/#>
```

Chapter 23

DHCP Relay Agent

This chapter describes WeOS *DHCP Relay Agent* support. For information on WeOS *DHCP Server* support, see chapter 22.

DHCP Relay Agents relay DHCP messages between DHCP clients on a local LAN to a central DHCP Server, usually located on a remote network. The two most common reasons for using 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.
- *Assigning IP address per port (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).

For redundancy purposes, the WeOS DHCP Relay Agent enables you to specify up to two DHCP servers, to which the Relay Agent forwards incoming DHCP requests.

In case you wish to hand out addresses per port on the *DHCP server* unit (as opposed to the *DHCP relay agent*), WeOS allows you to achieve this by running a relay agent on the DHCP server unit, see the chapter on DHCP server (section 22.1.4).

23.1 Overview of DHCP Relay Agent Support

The table below lists the features available in the WeOS DHCP Relay Agent.

Feature	Web	CLI	General Description
General DHCP Relay settings			
Enable/disable Relay Agent	X	X	Section 23.1.1
Define interfaces to serve	X	X	-”-
DHCP server IP address	X	X	-”-
DHCP server UDP port		X	-”-
DHCP Option 82			Section 23.1.2
Enable/Disable DHCP Option 82	X	X	-”-
Default Policy	X	X	-”-
Default Circuit-ID type	X	X	-”-
Remote-ID	X	X	-”-
DHCP Proxy Mode			Section 23.1.3
Force DHCP Option 54		X	-”-
Per-Port DHCP Relay settings			
Enable/Disable DHCP Relay	X	X	Section 23.1.4
DHCP Option 82			
Policy	X	X	Section 23.1.2
Circuit-ID type	X	X	-”-

23.1.1 Introduction to DHCP Relay Agents

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 present on the LAN can forward DHCP messages between local DHCP clients, and a central DHCP server.

Fig. 23.1 can be used to illustrate the use of DHCP relays and a central DHCP server.

- *(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 (Relay Agent 1 & 3) or *multiple* LANs (Relay Agent 2). In WeOS the LANs to serve is selected by configuring which (VLAN) network interfaces the relay agent should *listen* on.

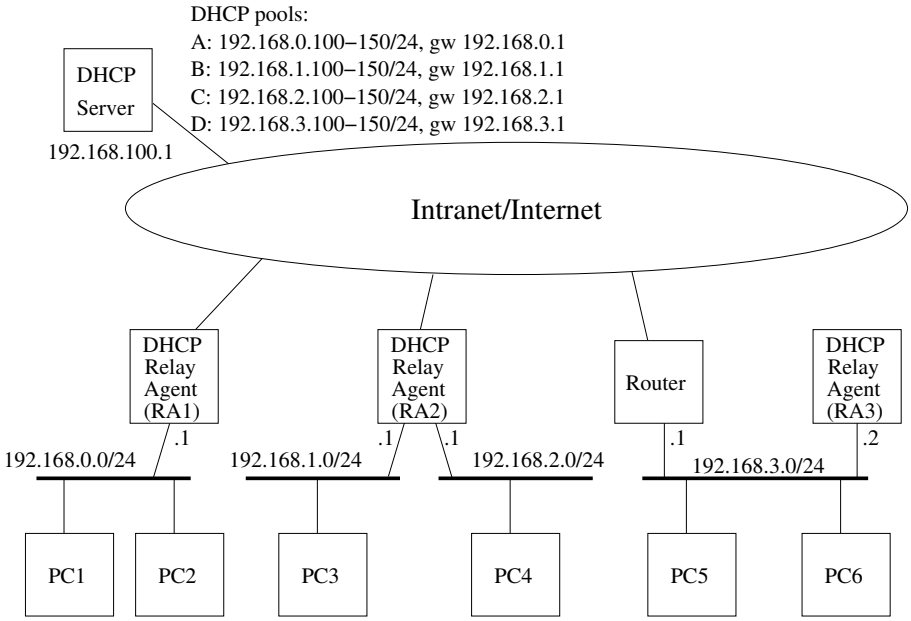


Figure 23.1: Sample topology where DHCP relay agents serve local DHCP clients, and forwards DHCP requests to/from a central DHCP server.

- *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*). As of WeOS v4.17.1, the relay agent can be configured with up to two DHCP servers. When configuring two DHCP servers, the DHCP relay will forward the DHCP requests to both servers, thereby providing redundancy.

  DHCP servers listen to UDP port 67 by default. It is possible configure the WeOS relay agent to forward packets to a different port on the server, see also sections 23.1.5 and 22.1.3.

- *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”).

The DHCP server should also be configured with other relevant settings, e.g., default gateway, lease times, etc. (see chapter 22).

- *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 fig. 23.1, 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 23.1.2 for information on DHCP Option 82). In such use cases, you may also wish to run several relay agents within the same LAN – section 23.1.4 provides more information on running relay agents in switched networks.

As of WeOS v4.17.1, it is only possible to run a single relay agent *instance* per WeOS unit. This is no major limitation, but implies, e.g., that a relay agent serving multiple LANs (RA2 in fig. 23.1) can not be configured to forward the DHCP requests from different LANs to different sets of DHCP servers.

23.1.2 DHCP Option 82

The *relay agent information option* (DHCP option 82, see RFC3046[28]) enables a relay agent to pass information to the DHCP server regarding which port the DHCP request came in on. Thus, an *option 82 aware* DHCP server would be able to assign IP settings (IP address, etc.) to a PC based on the port the PC connects to.

The DHCP option 82 contains two sub-options, *Circuit ID* and *Remote ID*:

- *Circuit ID:* The *circuit ID* identifies the port on the relay agent, where the DHCP request was received. Since the circuit ID can only be considered unique within the reporting relay agent, the DHCP server generally needs to consider both the *circuit ID* and an identifier of the specific relay agent (e.g.,*giaddr* or option-82 *remote ID*, see below) when processing the DHCP request.

  In WeOS the circuit ID can be set according to the following methods:

  - *Disabled:* When circuit ID is *disabled*, no circuit ID sub-option is passed as part of the Relay Agent Information option (DHCP option 82).

  - *Port Name:* Selecting the *port name* method implies that the circuit ID will be represented as *Type* appended by the *port identifier*, e.g., Eth1 and DSL1 on a single slot product, or Eth1/1 and DSL1/1 on a multi-slot product (see section 8.1.1 for more information on WeOS port naming conventions).
  - *Port Description:* By selecting the *port description* method, the circuit ID will be represented by the *port description* setting of the associated port. However, as of WeOS v4.17.1 the port description (chapter 8) can not yet be configured. Until configuration of port description is supported, the circuit ID will fall-back to using the *port name*, see above.
  - *Manual:* You can configure the Circuit-ID manually per port. The Circuit ID will be sent as a byte sequence (max 9 bytes), and you can choose to enter your manual circuit ID setting either as an ASCII string (max 9 characters) or as hexadecimal number (max 18 hex characters).

- *Remote ID:* According to RFC3046[28], the purpose of the remote ID should is to enable the DHCP relay agent to supply a trusted unique identifier of the DHCP client. In practice, it is commonly used as an identifier of the relay agent itself – the option 82 aware DHCP server can then base the IP address assignment on the combination of *circuit ID* and *remote ID*. In WeOS the remote ID can be set according to the following methods:
  - *Disabled:* When remote ID is *disabled*, no remote ID sub-option is passed as part of the Relay Agent Information option (DHCP option 82).
  - *MAC:* By selecting the *MAC* method, the unit's *base MAC address* (6 bytes, hexadecimal) will be used as remote ID. See sections 4.4.2 (Web) and 7.3.2 (CLI) for information on how to read the unit's base MAC address.
  - *IP:* By selecting the *IP* method, the relay agent will use the IP address of the interface where the DHCP request came in as remote ID (i.e., the *giaddr*). E.g., if RA2 in fig. 23.1 receives a DHCP request from PC4, it would use *192.168.2.1* as remote ID.
  - *System Name:* By selecting the *System Name* method, the unit's configured *hostname/system name* will be used as remote ID. See sections 20.1.1 (Web) and 20.2.2 (CLI) for information on how to configure the unit's hostname/system name.

When configuring a DHCP relay agent in WeOS, use of the relay agent information option is by default disabled. When enabling DHCP option 82, the relay agent will

add its relay information option to incoming DHCP requests, *unless* the request already contains a relay agent information option (added by some ”downstream” relay agent)[1].

Below the possible policy settings are listed how the relay agent should handle incoming DHCP requests already containing a relay agent information option. The policy can both be specified globally (i.e., per relay agent), as well as on per port basis.

- *Discard:* Drop requests already containing a relay agent information option.
- *Forward:* If the request already contains a relay agent information option, keep that entry when forwarding the request towards your DHCP server(s).
- *Replace:* If the request already contains a relay agent information option, replace that with your own DHCP option 82 field when forwarding the request towards your DHCP server(s).
- *Append:* If the request already contains a relay agent information option, append your own relay agent information option field when forwarding the request towards your DHCP server(s).
- *Require:* Discard requests lacking a relay agent information option. If the request already contains a relay agent information option, keep that entry when forwarding the request towards your DHCP server(s). This option may be useful in topologies including a mix of relay agents supporting and not supporting *DHCP snooping* (see sections 23.1.4, and 22.1.4.2).

When handling DHCP requests already containing a relay agent information option, the following mechanisms apply to all policies:

- *Dropping requests lacking a giaddr:* As of WeOS v4.17.1, incoming requests containing a relay agent information option, but lacking a *giaddr*, will be discarded.
- *Keeping existing giaddr:* When forward a request which already contains a relay agent information option, the *giaddr* field will be unchanged.

As of WeOS v4.17.1 no validation is performed by the relay agent on relay agent information option field(s) included in DHCP messages returned from the DHCP Server. The relay agent information is always removed[2] before passing it back to the DHCP client (PC), or to a relay agent closer to the PC. This behaviour *may* give

[1] The exception is when policy ”Require” is configured - then the packet will be discarded if it does not contain a relay agent information option.

[2] If more than one relay information option is included, the last option is removed.

problems at downstream relay agents when using the *Forward*, *Append*, *Replace*, and *Require* policies. WeOS handling of packets on the return path from the DHCP server may be modified in upcoming WeOS releases.

23.1.3 DHCP Proxy Mode

According to the RFC2131[7] a DHCP relay agent would only be involved in the initial DHCP message, while subsequent DHCP lease renew messages would be sent directly between client and server, as shown in fig. 23.2.

For many use-cases however, this behaviour is not desirable. In particular with Option 82 (see section 23.1.2) all DHCP messages from the client to the server need to have this extra piece of information appended so that the server can properly identify the client. This is called DHCP Proxy Mode, or DHCP Server Identifier Override, defined in RFC5107[18].

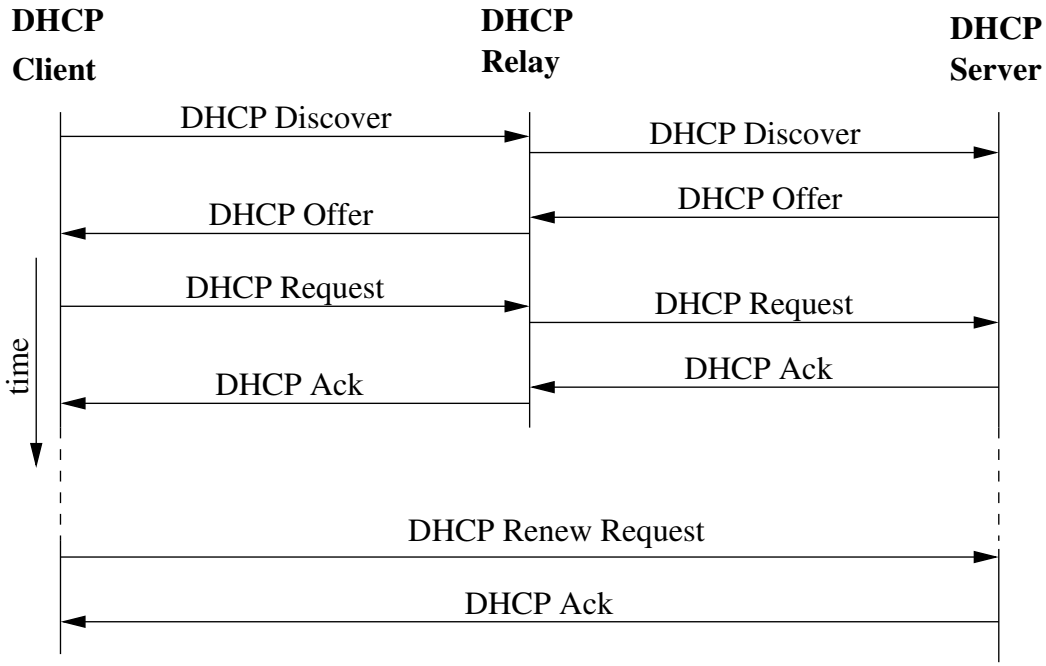


Figure 23.2: Typically only the initial DHCP exchange is done via the relay agent, while *lease renew messages* are sent directly (unicast) between client and server.

Most modern DHCP servers support RFC5107[18], which is a sub-option to Option 82. But some older DHCP servers do not and for this particular case the WeOS

relay agent can be configured to forcibly override Option 54, the Server Identity field. In effect, making sure that the client will send all DHCP messages via the relay agent, see fig. 23.3.

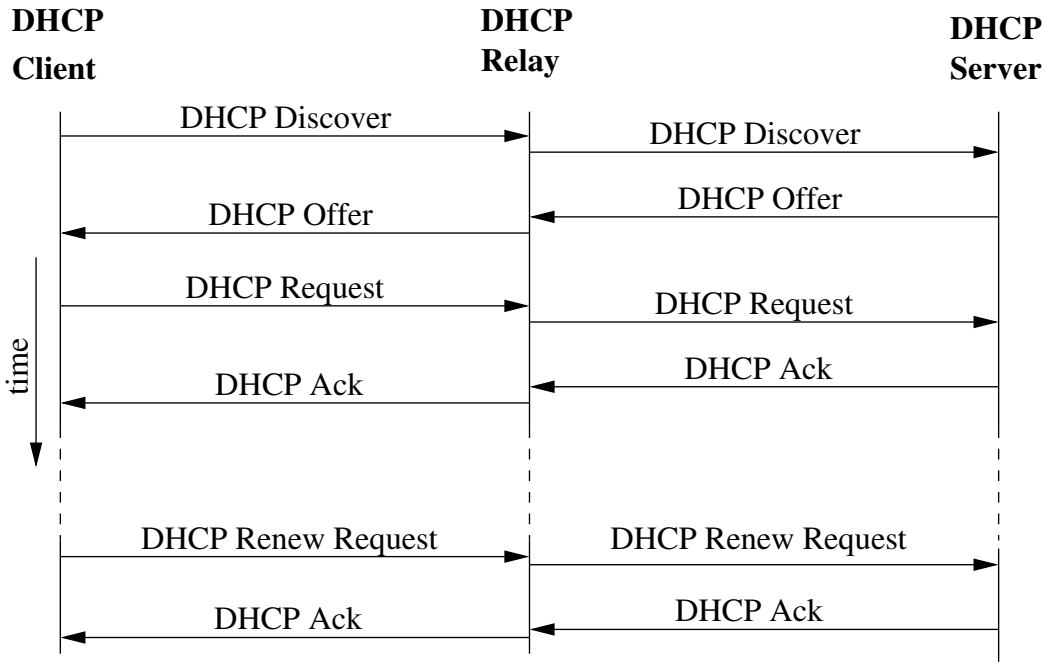


Figure 23.3: DHCP Proxy Mode, all messages goes via the relay agent.

Hence, there are two levels of DHCP Proxy Mode support in WeOS.

- *Hint to server:* The WeOS relay agent adds sub-option 11 to option 82 in all DHCP messages forwarded to the server, to *hint* the server to fill in the IP address of the relay agent in the DHCP server identity field in the server responses. If the server supports RFC5107[18], the relay agent will act as a server proxy towards the client (fig. 23.3).

  Note
  The WeOS DHCP server (chapter 22) supports RFC5107[18].

- *Force identity override:* The WeOS relay agent can force server identity override by updating the packets sent towards the DHCP client. This feature can be useful in situations where the DHCP server does not support RFC5107[18]. Forcing DHCP server identity override is disabled by default.

23.1.4 Relay Agents in Switched Networks

The DHCP protocol uses layer-2 broadcast (Destination MAC: ff:ff:ff:ff:ff:ff) for some of its protocol messages. Therefore, a (broadcast) DHCP packet coming in to a switch, will typically be flooded on all ports of the same LAN. This is illustrated in fig. 23.4a):

- A broadcast DHCP message comes in on port ”A” of the switch (step ”1a”).
- The message is broadcasted *unmodified* on all other ports within the LAN (here ports ”B”-”F”), see step ”1b”.
- In this case, the switch is also running a DHCP relay service on the LAN. The relay agent will process the incoming DHCP packet, and forwards it to the configured DHCP server, which here happens to reside in the direction of port ”E” (step ”2”). The packet in step ”2” is modified as compared to the initial broadcast packet: It is sent as unicast to the DHCP server, and it contains the relay agents IP address as *giaddr*. If the relay agent has DHCP option 82 enabled, such information is also added.

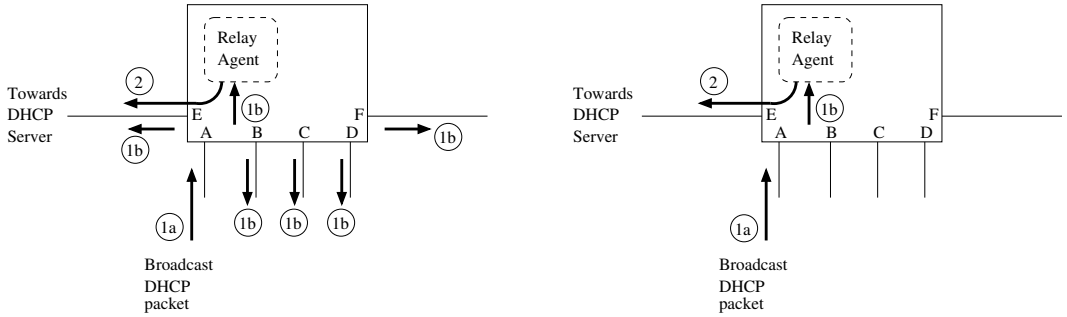


Figure 23.4: Propagation of DHCP broadcast packets in switches running DHCP relay agents. All ports are on the same (V)LAN. The switch in figure a) does not support DHCP snooping, while the switch in figure b) supports DHCP snooping.

As seen in fig. 23.4a), using (layer-2) switches as DHCP Relay Agents can result in multiple versions of a DHCP message to be sent towards the DHCP server: the original request being switched/broadcasted, and the one being relayed by the relay agent process. This will not cause any problems if the DHCP server is located on some remote network; then only the relayed packet will reach the server. However, if the DHCP server is located within the same LAN, adequate support is needed at the DHCP server to know which request to serve and which

to ignore (see section 22.1.4.2 in the DHCP server chapter for more information). The number of ”copies/versions” of a DHCP request can increase further if a LAN consists of several switches with DHCP relay agents (discussed later on, see fig. 23.5).

To mitigate multiplication of broadcast DHCP messages, some switches support *DHCP snooping* (see also section 23.1.5 for an alternative approach). With DHCP snooping enabled on an Ethernet/DSL port, *all* DHCP packets will pass through the DHCP relay agent – this includes broadcast and unicast DHCP packets, both DHCP requests (to server) or DHCP responses (from server) coming in on that port. Fig. 23.4b) shows the result when a broadcast DHCP packet comes in on a port with DHCP snooping enabled.

When configuring a WeOS relay agent on a VLAN interface, all ports on that VLAN will have DHCP snooping enabled

- the exception is products lacking hardware support for DHCP snooping[3]. More fine-grain control to enable/disable DHCP snooping per port may be supported in later WeOS versions.

DHCP relay service can be disabled on a per port basis. If DHCP relaying is disabled on an Ethernet/DSL port, incoming DHCP packets will be switched as other layer-2 packets (no DHCP snooping), and the DHCP relay agent on the switch will ignore DHCP requests entering the switch on that port.

Fig. 23.5 presents an example where multiple relays are located within the same VLAN – port 1-6 on all RA units are in the same VLAN, while port 7 on RA1 and RA2 are associated with another VLAN used and used as upstreams interface. The topology in fig. 23.5 utilise several WeOS features to achieve a robust network: FRNT (chapter 14) is used to handle single link failures within the local network. VRRP (chapter 30) is used to handle router redundancy (RA1 and RA2). A second DHCP server to protect against DHCP server failure[4].

The relay agents (RA1-RA5) server DHCP clients connecting to the local access ports (ports 1-4), and will relay each request (unicast) to the configured DHCP server(s). Below a sample DHCP relay configuration is shown, which would be suitable for all relay agents in fig. 23.5.

[3] In WeOS products, DHCP Snooping is supported on all Ethernet/DSL ports, **except** for ports of switchcore(s): MV88E6095, MV88E6185 and MV88E6046. Please see *Detailed System Overview* page in the Web (section 4.4.2) or use the **”show system-information”** in the CLI (section 7.3.2) to find information about what switchcore(s) is used in your product.

[4] As of WeOS v4.17.1, the WeOS DHCP server (chapter 22) does not provide dedicated DHCP server failover support, but you can achieve redundancy using ”static” address assignment (no address pools) with the same configuration at both DHCP servers.

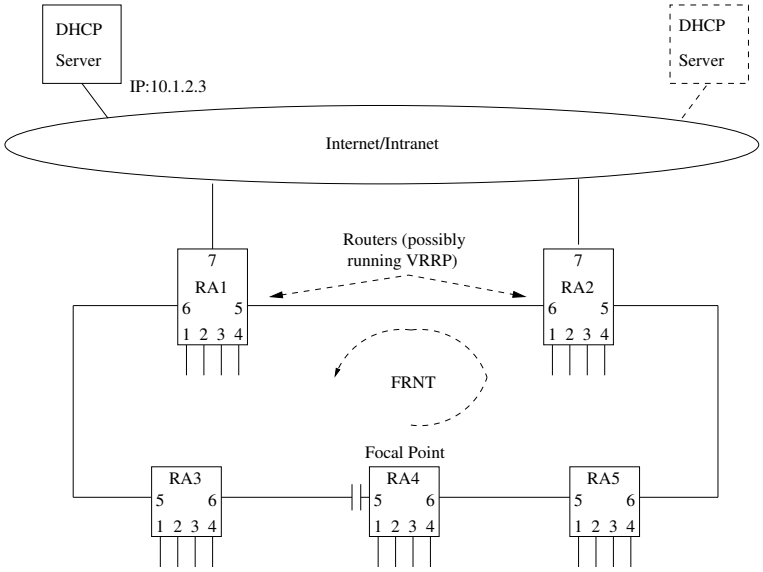


Figure 23.5: Example with multiple DHCP Relay Agents within the same VLAN (port 1-6 on all RAs are assumed to be on the same VLAN, e.g., VLAN 1).

Example

```
dhcp-relay
        iface vlan1
        server 10.1.2.3
        option82 discard
        port 5-6
                no enable
                end
        end
```

- DHCP relay has been enabled on interface vlan1 (this assumes that ports 1-6 are all associated with VLAN 1).
- A single DHCP server has been configured (here 10.1.2.3). As of WeOS v4.17.1, up to two DHCP servers can be configured.
- Option 82 is enabled, with policy discard. Option 82 information will be added to all incoming requests. Packets which already include option 82 information will be discarded. Default settings for *circuit-id* (port name) and *remote-id* (base-MAC) will be used.
- DHCP requests coming in on port 5 or 6 will be ignored by the relay agent. No DHCP snooping will be done on those ports, thus a DHCP request being

relayed by RA4 to the DHCP server, will be forwarded through RA5 like any other packet.

23.1.5 Mitigating duplication of DHCP messages by using a different server port

An alternative to address the issue with multiple DHCP requests in switched topologies with non-snooping relay agents is to let the DHCP server listen on a non-standard UDP port (section 22.1.3). The DHCP relay agent can be configured to forward its packets to this server port (section 23.1.1), thus all *relayed* packets will reach the server. Packets coming directly from the client will be dropped by server, since they are sent to the regular DHCP server UDP port (67).

Example

```
-- DHCP Server configuration of non-standard listen port

example-server:/#> configure
example-server:/config/#>
example-server:/config/#> dhcp-server
example-server:/config/dhcp-server/#> server-port 6767
example-server:/config/dhcp-server/#> leave
Stopping DHCP/DNS Server ..................................... [ OK ]
Starting DHCP/DNS Server ..................................... [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example-server:/#>

-- DHCP Relay Agent configuration of non-standard server port

example-relay:/#> configure
example-relay:/config/#> dhcp-relay
example-relay:/config/dhcp-relay/#> server-port 6767
example-relay:/config/dhcp-relay/#> leave
Stopping DHCP/DNS Server ..................................... [ OK ]
Starting DHCP/DNS Server ..................................... [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
Starting DHCP Relay Agent .................................... [ OK ]
example-relay:/#>
```

23.2 Configuring DHCP Relay Agent via the Web

The Web interface provides management of the DHCP Relay Agent.

23.2.1 DHCP Relay Agent settings

Menu path: Configuration ⇒ Network (IP) ⇒ DHCP-Relay

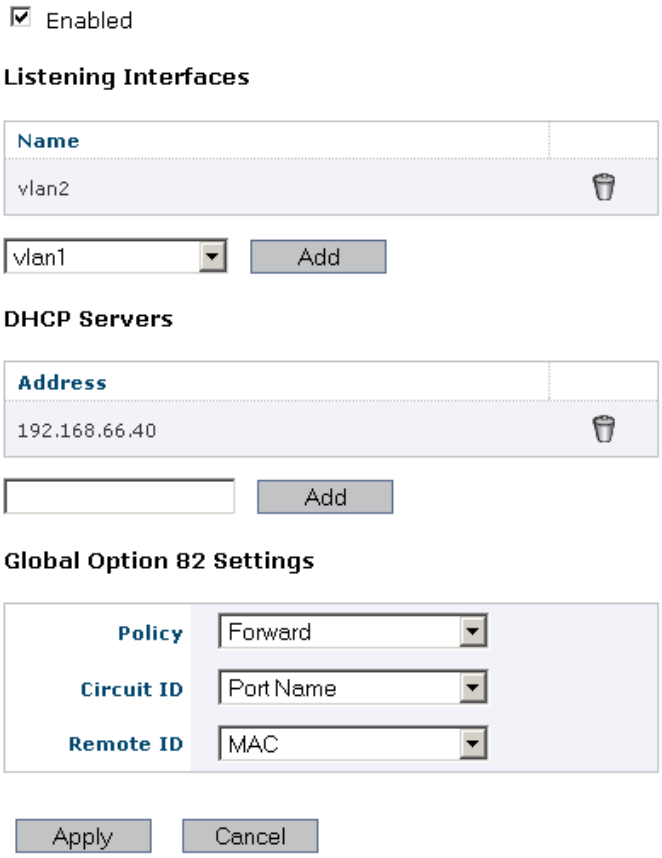


Figure 23.6: DHCP Relay Agent settings

Listening Interfaces	The **Listening Interface** specifies on which interface(s) the relay agent will listen for client requests. DHCP server responses may come in through any interface.
DHCP Servers	The **DHCP Servers** settings determine to which DHCP servers each DHCP client request will be sent. At most two servers may be configured.
Global Option 82 Settings	The **Global Option 82 Settings** determine how the DHCP Relay Agent Information option, also known as Option 82, will be handled. The **policy** specify how to treat incoming client requests that already contain an Agent Information option. • **Disable:** Do not add option 82 field. Any existing option 82 will be retained. • **Forward:** Adds a new option 82 or forwards any existing option 82. • **Append:** Appends a new option 82 in addition to any existing option 82. • **Discard:** Drops the whole packet if it contains an option 82. • **Replace:** Removes any existing option 82 and adds a new option 82. • **Require:** Requires that the incoming packet contains an option 82 otherwise it will be dropped.

The **Circuit ID** setting determines how the Circuit-Id field of option 82 will be filled. It can be one of **None**, **Port Name** and **Port Description**. **None** will leave this field with zero length, **Port Name** will fill this field with the port type and name of the port as seen on front foil, stripped of any whitespace. E.g. Eth6 for Ethernet port 6. Lastly **Port Description** will use the description given to the port in the port settings.

In a similar fashion the **Remote ID** tells how the remote id field of option 82 will be set. **None** set its length to zero, **IP** sets it to the IP address of the inbound interface. **MAC** uses the base MAC address of the unit. Lastly, **System Name** uses the hostname of the system.

23.2.2 DHCP Relay Agent Per-Port Settings

Menu path: Configuration ⇒ Network (IP) ⇒ DHCP-Relay Agent ⇒ Port Specific Settings Show

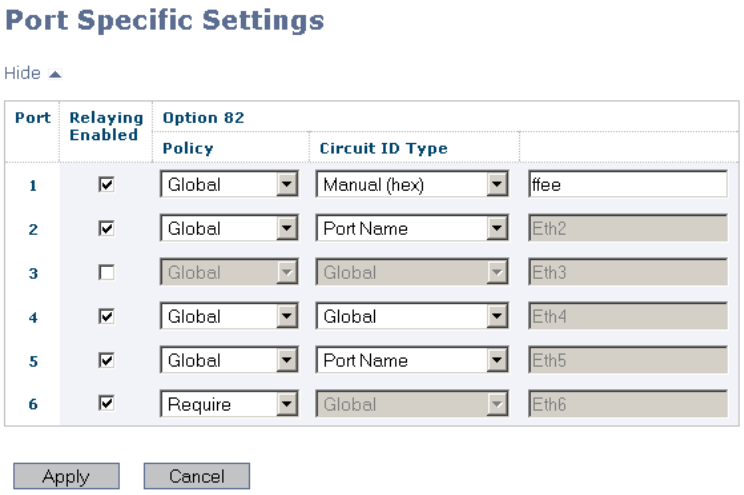


Figure 23.7: DHCP-Relay Agent Per-Port Settings page

Enabled	The **Enabled** checkbox tells whether to enable the relay agent on this port, i.e. whether to listen for client requests on this port or not. If enabled, you can override the global settings.
Option 82 Policy	See section 23.2.1 for an explanation of the different policy options. In the port specific section, the **Policy** setting has an additional option **`Global`**, indicates that the global policy setting (see fig. 23.6) will be used for this port.
Option 82 Circuit ID	See section 23.2.1 for an explanation of the different circuit ID types. In the port specific section, the **Circuit ID** setting has additional options for the Circuit ID type. • **Global:** Indicates that the global circuit ID setting (see fig. 23.6) will be used for this port. • **Manual (hex)** and **Manual (string):** A user specified *hex* or *string* value will be used as circuit ID. Value is entered in the **Manual Circuit ID** field.

23.3 Configuring DHCP Relay Agent via the CLI

Command	Default	Section
Configure DHCP Relay Agent		
[no] dhcp-relay		Section 23.3.1
[no] enable	Enabled	Section 23.3.2
[no] iface <IFACE>	Disabled	Section 23.3.3
[no] server <IPADDR>	Disabled	Section 23.3.4
[no] server-port <PORT>	Disabled	Section 23.3.5
[no] force-server-identity	Disabled	Section 23.3.6
[no] option82 <forward|discard|append|replace|require>	Disabled	Section 23.3.7
[no] circuitid-type <portname|portdescription>	”portname”	Section 23.3.8
[no] remoteid-type <mac|ip|system-name>	”mac”	Section 23.3.9
port <PORTLIST|all>		Section 23.3.10
[no] enable	Enabled	Section 23.3.11
[no] option82 <auto|forward|discard|append|replace|require>	”auto”	Section 23.3.12
[no] circuitid-type <auto|portname|portdescription|manual <hex|string> <ID>>	”auto”	Section 23.3.13
View DHCP Relay Agent Settings		
show dhcp-relay		Section 23.3.14
dhcp-relay		
show port [PORTLIST]	”all”	Section 23.3.15

23.3.1 Manage DHCP Relay Agent

Syntax `[no] dhcp-relay`

Context Global Configuration context

Usage Create, modify or remove the DHCP Relay Agent.

Enter DHCP relay agent context.

Use **”no dhcp-relay”** to remove an existing DHCP relay configuration.

Default values Not applicable.

23.3.2 Enable DHCP Relay Agent

Syntax `[no] enable`

Context DHCP Relay Configuration context

Usage Enable the DHCP Relay Agent.

Default values Enabled.

23.3.3 Listening Interfaces

Syntax `[no] iface <IFACE>`

Context DHCP Relay Configuration context

Usage Specify the interfaces that the relay agent will listen to.

Default values Not applicable.

23.3.4 DHCP Servers (IP addresses)

Syntax `[no] server <ADDRESS>`

Context DHCP Relay Configuration context

Usage Specify the DHCP server that the relay agent will forward requests to.

Default values Not applicable.

23.3.5 DHCP Server UDP port

Syntax `[no] server-port <UDPPORT>`

Context DHCP Relay Configuration context

Usage Specify the DHCP server UDP port that the relay agent will forward requests to. See also section 22.3.4 for the corresponding DHCP relay agent setting.

Use **”no server-port”** to reset to default value (port 67).

Use **”show server-port”** to show current server-port settings.

Default values 67

23.3.6 Force DHCP Server Identity Override

Syntax `[no] force-server-identity`

Context DHCP Relay Configuration context

Usage By enabling the *force DHCP server override* setting, the DHCP relay agent can work-around older DHCP servers that do not support RFC5107[18] (a hint/extension to Option 82) by overriding Option 54 in the server response to the client with the relay agents IP address.

It is recommended to leave this setting disabled and instead either use the WeOS DHCP server, or upgrade to another RFC compliant DHCP server.

Use **”force-server-identity”** to enable *force DHCP server override* and use **”no force-server-identity”** to disable it.

Use **”show force-server-identity”** to show the current setting.

Default values Disabled

23.3.7 Option 82

Syntax `[no] option82 <forward|discard|append|replace|require>`

Context DHCP Relay Configuration context

Usage Enable or disable the addition of option 82, a.k.a. relay agent information, to DHCP requests. The policy for how to handle any existing option 82 can optionally be specified as follows.

Forward
Adds a new option 82 or forwards any existing option 82.

Append
Appends a new option 82 in addition to any existing option 82.

Discard
Drops the whole packet if it contains an option 82.

Replace
Removes any existing option 82 and adds a new option 82.

Require
Requires that the incoming packet contains an option 82 otherwise it will be dropped.

Default values Option 82 is disabled by default, if enabled and policy is omitted it defaults to **forward**.

23.3.8 Circuit ID Type

Syntax `[no] circuitid-type <portname | portdescription>`

Context DHCP Relay Configuration context

Usage Specify how the circuit id in option 82 will be set. **portname** will use the name of the port as it is printed on the front foil plus the port type. For Ethernet ports it will be Eth, so e.g. requests coming in on port 6 will have the Circuit ID set to “Eth6”. **portdescription** is currently the same as **portname** but will use the port description set in the port configuration, as soon as that feature is released.

Default values **portname**.

23.3.9 Remote ID Type

Syntax `[no] remoteid-type <mac | ip | system-name>`

Context DHCP Relay Configuration context

Usage Specify how the remote id in option 82 will be set. **mac** will use the base MAC address of the unit. **ip** will use the IP address of the inbound interface. **system-name** will use the hostname.

Default values **mac**

23.3.10 Manage DHCP Relay Agent Per-Port Settings

Syntax `port <PORT|PORTS>`

Context DHCP Relay Configuration context

Usage Modify DHCP Relay Agent configuration for one or several ports.

Default values Not applicable.

23.3.11 Enable/disable DHCP Relay Agent per port

Syntax `[no] enable`

Context DHCP Relay Configuration context

Usage Enable or disable the DHCP Relay Agent on a port.

Default values Enabled.

23.3.12 Option 82 policy per port

Syntax `[no] option82 <auto|forward|discard|append|replace|require>`

Context DHCP Relay Port Configuration context

Usage Enable or disable the addition of option 82 on one ore more ports. The **auto** policy uses the same a policy as specified in the *DHCP Relay* context.

Default values **auto**.

23.3.13 Option 82 Circuit ID per port

Syntax `[no] circuitid-type <auto|portname|portdescription>`

Context DHCP Relay Port Configuration context

Usage Specify how the circuit id in option 82 will be set for this port. In addition to the keywords defined in section 23.3.8 **auto** can be used, meaning the configured circuit ID type in *DHCP relay* context.

Default values **auto**.

23.3.14 Show DHCP Relay Agent Settings

Syntax `show dhcp-relay` Also available as **"show"** command within the DHCP Relay Configuration context context.

Context Global Configuration context

Usage Show DHCP relay agent settings.

Default values

23.3.15 Show DHCP Relay Agent Per-port Settings

Syntax `show port [PORTLIST]` Also available as **"show"** command within the DHCP Relay Port Configuration context.

Context DHCP Relay Configuration context

Usage Show DHCP relay agent per port settings. Furthermore, not only the circuit ID *type* settings are listed, but also the resulting *circuit ID*.

Default values If no PORTLIST is given, settings are listed for all ports associated with the given (VLAN) interfaces (see also section 23.3.3).

Example

```
example:/config/dhcp-relay/#> show port
Port     Enabled  Policy  Circuit-ID type       (Circuit ID)
=================================================================================
Eth 1     NO      auto    auto                  (Eth1)
Eth 2     NO      auto    auto                  (Eth2)
Eth 3    YES      auto    auto                  (Eth3)
Eth 4    YES      auto    auto                  (Eth4)
Eth 5    YES      auto    auto                  (Eth5)
Eth 6    YES      auto    auto                  (Eth6)
example:/config/dhcp-relay/#>
```

Chapter 24

Alarm handling, Front panel LEDs and Digital I/O

This chapter describes WeOS features for alarm and event handling (sections 24.1-24.3). The chapter also covers general information on functionality related to *Digital I/O* and *front panel LEDs* (sections 24.4 and 24.5).

24.1 Alarm handling features

The table below summarises the WeOS alarm handling features.

Feature	Web	CLI	General Description
Configure alarm triggers	X	X	Sections 24.1.1-24.1.3
Configure alarm actions	X	X	Sections 24.1.1 and 24.1.4
Configure alarm targets	X	X	Sections 24.1.1 and 24.1.5
View alarm status[1]	X	X	Section 24.1.5

24.1.1 Introduction to the WeOS alarm handling support

The WeOS alarm handling support makes use of the following terminology:

[1] In addition to monitoring alarm status via Web and CLI, there are other ways in which an operator can get notified when an alarm is triggered.

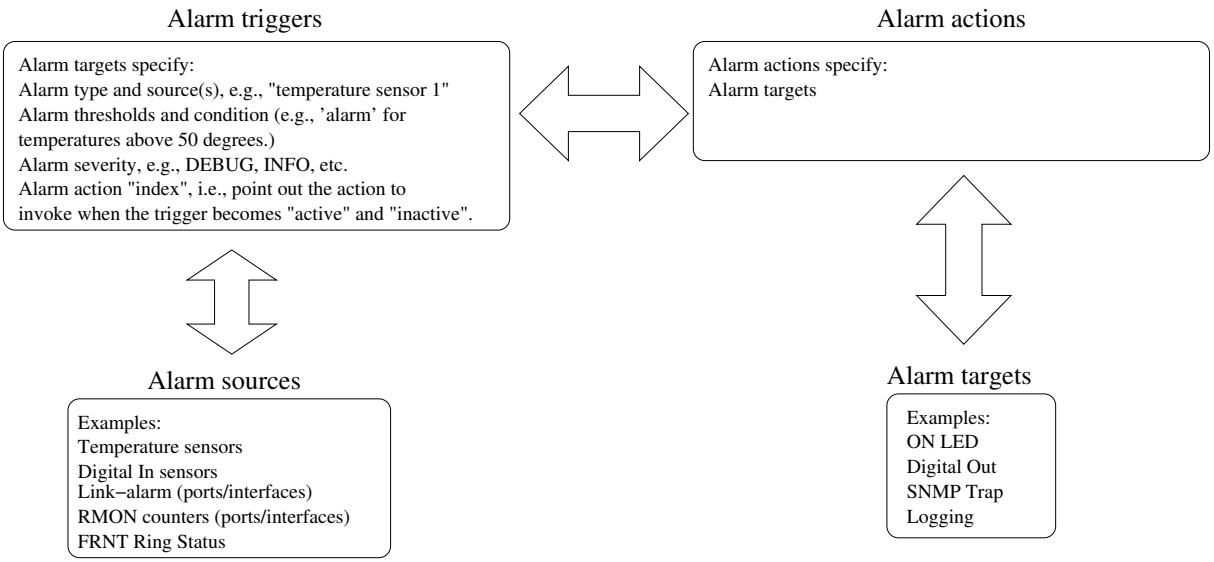


Figure 24.1: Overview of WeOS alarm entities: Alarm triggers monitor the state of alarm source, and define conditions and thresholds when to invoke an associated alarm action. The invoked alarm action specifies what alarm target(s) to use to notify the operator.

- *Alarm sources:* An *alarm source* is an object being monitored by an *alarm trigger*, e.g., the link status (up/down) of an Ethernet port, the input byte counter of a network interface, or the temperature value of a temperature sensor. Alarm sources are described further in section 24.1.2.
- *Alarm trigger:* An *alarm trigger* monitors alarm sources, and defines the conditions when alarm events occur, i.e., when the trigger becomes *active* (alarm situation) or *inactive* (normal situation).

  In addition, the alarm trigger specifies the *alarm action* to be invoked once an alarm event occurs. Alarm triggers are described further in section 24.1.3.
- *Alarm actions and alarm targets:* When an alarm event occurs, the operator can be notified via SNMP traps, logging, digital-out, and front panel status LED. These notification mechanisms are referred to as *alarm targets*.

  Instead of mapping triggers directly to targets, a trigger is mapped to an *alarm action (profile)*. The alarm action defines what specific targets to use when an alarm event occurs. For example, a link alarm trigger for ports 1-3 can be mapped to a specific alarm action, which in turn specifies *logging* and *SNMP traps* as targets. Alarm actions and targets are described further

in sections 24.1.4 and 24.1.5 respectively.

24.1.2 Alarm sources

As of WeOS v4.17.1 the following alarm sources are supported:

- *Power failure:* If the unit is equipped with redundant power feed (or redundant power supply), an alarm can be triggered if one of the feeds lack input power.

  Note: if all power is lacking on all feeds, the unit is powerless and cannot trigger alarms via SNMP traps or remote logging. To detect such a situation remotely, the operator could *poll* the unit (e.g., by *pinging* the unit on a regular interval). The drawback is that it is difficult to distinguish problems in the intermediate network from problems in the monitored device.

  An alternative is to use out-of-band signalling, e.g., via GPRS equipment connected to *digital-out* to get an alarm notification instantly if a device goes down.

- *Link alarm:* It is possible to configure link alarm triggers to react when a link goes down (and up).

- *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.

- *Temperature sensor alarms:* Temperature alarm triggers can be configured to react when the temperature rises above (or falls below) some defined threshold.

- *FRNT status:* The FRNT ring status trigger will react when an FRNT ring is broken (bus mode) or healed (ring mode)[1].

- *Hardware failure:* Hardware alarms triggers notifies that the unit has detected a hardware failure (typically if an unsupported SFP is inserted).

- *SHDSL/xDSL SNR Margin:* On devices with SHDSL/xDSL ports, alarms can be triggered when the SNR margin falls below some configured threshold.

- *Link Fault Forward (LFF):* On devices with SHDSL ports, alarms can be triggered when the remote SHDSL switch indicates it has link down on its Ethernet port. That is, this feature can be used in topologies where an Ethernet

[1] Only an FRNT focal point can determine the ring status with certainty.

is extended over an SHDSL link, and where the remote SHDSL switch (e.g., a DDW-120) is able to signal that the Ethernet link is down on its side.

- *Network Connectivity (Ping):* It is possible to have a trigger to monitor network connectivity by using the *ping* command to a specific host. The remote node is considered unreachable if a configurable number of pings are lost, and considered reachable if the same number of pings are successfully received.

  Note
  Make sure the remote host responds to ICMP ping. A typical behaviour of many hosts is that ICMP ping is blocked in the host's firewall.

- *PoE Power Usage:* On units supporting Power Over Ethernet (PoE), alarms can be triggered when the total power usage raises above (or falls below) some configured threshold.
- *Microlok Session Status* On units running a Microlok Gateway (chapter 41, an alarm can be triggered if any of the established sessions go down.

24.1.3 Alarm triggers

An alarm trigger defines the rules for when alarm events should be generated for a monitored alarm source. Alarm triggers also define which *alarm action* to invoke when an alarm event occurs.

Currently supported alarm trigger types:

- Power failure
- Link alarm
- Digital-In
- Temperature
- FRNT ring status
- Hardware failure (The *hardware failure* alarm trigger is *implicit*, and cannot be removed or modified.)
- SNR margin (SHDSL and xDSL ports)
- LFF (SHDSL ports)

- Timer
- Ping
- PoE power usage
- Microlok Session Status

As the WeOS alarm handling support is designed to include triggers for additional alarm sources, the following description is of more general nature, thus contains more options than needed for the trigger types currently supported.

Note
As of WeOS v4.17.1 there is no support for making an alarm trigger *persistent*. When an alarm condition is no longer fulfilled, the trigger status will become *inactive*. As alarms are not persistent, it is not possible for an operator to clear (i.e., acknowledge) an alarm.

24.1.3.1 Specifying what alarm source(s) a trigger should monitor

Different types of alarm triggers operate on different types of alarm sources:

- Power failure: A power failure 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 failure trigger is used to monitor both power feed sensors.
- Digital-In: A digital-in trigger can monitor one or more digital-in sensors. WeOS products typically have a single digital-in sensor.
- Link alarm: Link alarm triggers monitor the operational status (up/down) of Ethernet or DSL ports. Thus when configuring a link alarm trigger the port (or ports) to monitor should be specified.

  Note
  It is possible to define multiple link alarm triggers, where each trigger can monitor different ports and be mapped to different alarm actions.

  In the future, link alarm triggers can be extended to monitor the operational status of *network interfaces* and *VLANs* in addition to physical ports (Ethernet, SHDSL, etc.).

- RMON statistics (not yet supported): The alarm source for an RMON trigger is specified by two parameters: (1) the name of the statistics counter (e.g., *etherStatsPkts*), and (2) the port (or list of ports) for which this counter should be monitored.

  Note
  In WeOS the term RMON is used to refer to data traffic statistics in general; not only to the Ethernet statistics defined in the RMON MIB. Thus, if a counter from the IF-MIB (such as *ifHCInUcastPkts* is specified, the alarm source could refer to *network interfaces* or *VLANs* as well as a physical ports (Ethernet, SHDSL, etc.).

- Temperature: Temperature triggers can apply to one or more temperature sensors.

- FRNT: FRNT triggers can apply to one or more FRNT rings (as of WeOS v4.17.1 only a single FRNT ring is supported).

- Timer: Timer triggers are configured to go off at given *time interval*. As of WeOS v4.17.1, only daily timers are supported, e.g., **"timeout daily 02:30"**, and only apply to **"log"** and **"reboot"** action targets.

- SNR Margin: An SNR Margin trigger applies to one or more SHDSL/xDSL ports.

- LFF (Link Fault Forward): An LFF trigger applies to one or more SHDSL ports.

- Ping: A connectivity checker, sends an ICMP ping in a configurable interval.

- PoE Power Usage: The WeOS PoE enabled units have a *single* PoE power module, and its current usage level is used as trigger source (i.e., no need to select a trigger source).

Typically there would be no more than one trigger monitoring the status of a specific alarm source. However, in some cases it would make sense to have multiple triggers monitoring a single alarm source. For example, one could define two temperature triggers for a single temperature sensor, where one trigger reacts if the temperature rises above a *warning threshold* (say 60°C), and the other if the temperature gets *critically high* (say 75°C).

24.1.3.2 Alarm thresholds and trigger output

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, etc.) have thresholds defined *implicitly*.

For sources which can take values in a wider range (temperature, SNR Margin, received packets within a given time interval, etc.) the alarm thresholds should be *configured*. Fig. 24.2a) illustrates use of alarm thresholds for a temperature trigger.

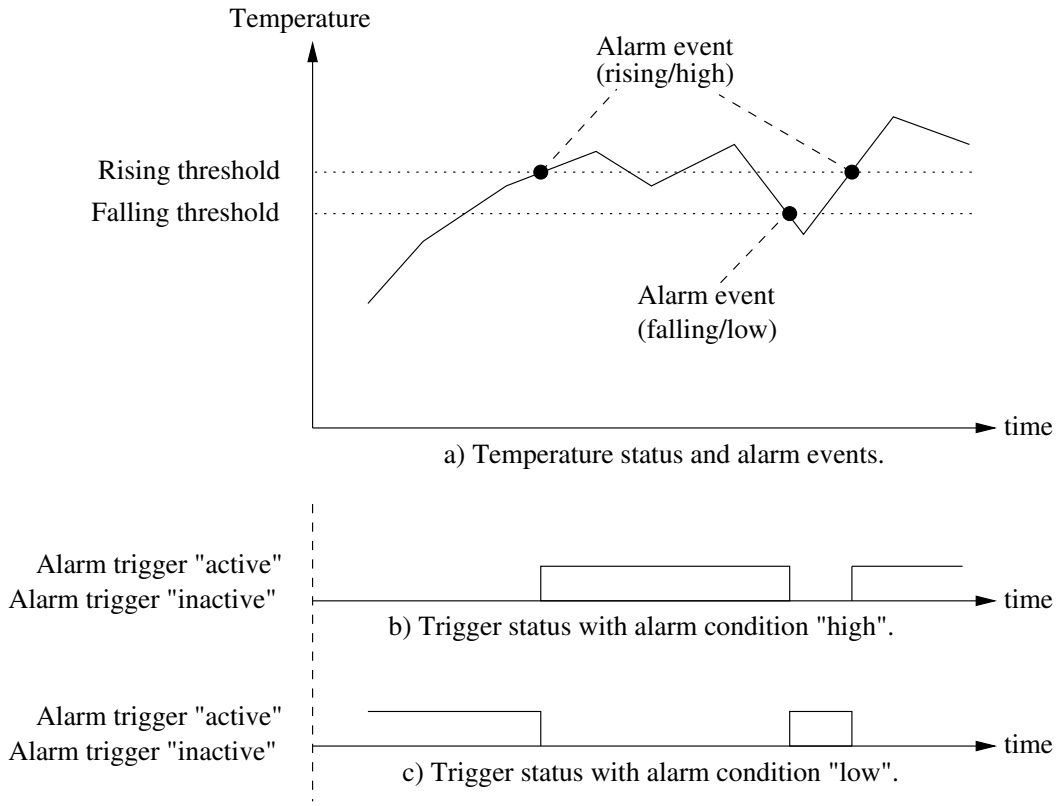


Figure 24.2: Example use of rising and falling thresholds for a temperature alarm trigger (a), and alarm condition setting to affect active and inactive trigger status (b and c).

As can be seen in fig. 24.2a), two thresholds are used – a *rising* threshold and a *falling* threshold. Alarm events will be generated when reaching the rising thresh-

old 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.

Alarm targets such as *Digital-Out* and the *ON LED* provide a summary alarm function (see section 24.1.5.1), and these targets assume that every alarm trigger define the condition when the alarm is *active* (”alarm” situation) and *inactive* (”normal” situation). To define this the alarm *condition* configuration option is used. To warn the operator for high temperatures, the alarm condition should be set to ”high”, see fig. 24.2b). If we instead wish to warn the operator for low temperatures, the alarm condition should be set to ”low”, see fig. 24.2c). A corresponding example for a *Digital-In* trigger is shown in fig. 24.3.

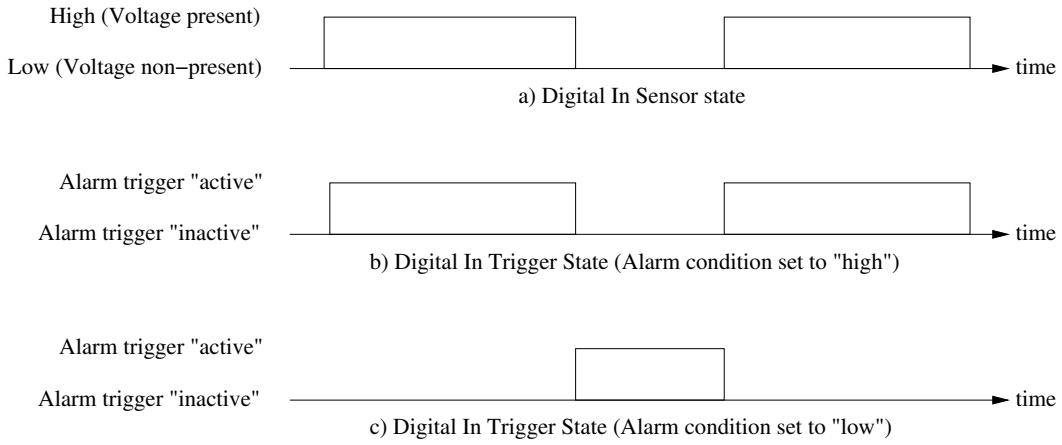


Figure 24.3: Alarm condition example: The alarm trigger for digital-in can be configured to become active when the signal is high (b) or when it is low (c).

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*).

24.1.3.3 Sample types and interval

Two sample types are possible: *absolute* and *delta* sampling. With absolute sampling, the value is compared directly to the alarm thresholds. With delta sampling it is the difference between the current sample and the previous sample which is compared to the alarm thresholds.

Alarm sources of *counter* type, such as RMON data traffic statistics, are well suited for delta sampling. As the delta is computed over a given time interval (sample interval), the alarm thresholds should be configured with respect to the configured sample interval.

Note
As of WeOS v4.17.1 only absolute sampling is supported, and the sampling interval is not configurable for any trigger type.

24.1.3.4 Alarm severity

For each trigger it is possible to define the severity level of the associated alarm events. The levels defined by Unix Syslog are used:

- EMERG: System is unusable
- ALERT: Action must be taken immediately
- CRIT: Critical conditions
- ERR: Error conditions
- WARNING: Warning conditions
- NOTICE: Normal, but significant, condition
- INFO: Informational message
- DEBUG: Debug-level message

It is also possible to configure severity 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 digital-out and ON LED targets.

Note
Severity levels can be configured independently for the events when an alarm trigger becomes ”active” and ”inactive”. Default severity level are WARNING for ”active” alarm events and NOTICE for ”inactive” alarm events.

24.1.3.5 Mapping triggers to actions

Triggers can be mapped to alarm actions (profiles) that are invoked when an alarm event occurs, for more information see section 24.1.4. However, it is also possible to leave a trigger unmapped, e.g., when defining a *ping trigger* to adjust VRRP priority dynamically (see section 30.1.1).

24.1.4 Alarm actions - mapping triggers to targets

Instead of mapping triggers directly to alarm targets, each trigger is mapped to an alarm action (alarm action profile). The alarm action specifies which targets to use (SNMP traps, Logging, ON LED, and Digital-Out) when an alarm event occurs.

It is possible to configure several actions (action profiles). Each trigger can be mapped to an individual action, but it is also possible for multiple triggers to share the same action. This can be particularly useful when managing several triggers of similar type, such as different types of RMON triggers.

By default a trigger is mapped to the *default alarm action* (index 1). The default alarm action cannot be removed.

24.1.5 Alarm presentation (alarm targets)

When an alarm situation occurs, such as an FRNT ring failure, WeOS enables the operator to be notified in numerous ways:

- *SNMP trap:* Alarms can be configured to generate SNMP traps[2]. See chapter 6 for general information on SNMP.

[2] As of WeOS v4.17.1 there is no support for SNMP traps for *timer* or *hardware* alarms.

- *Log files and remote logging:* Alarms can be logged locally or passed to a remote logging server. See chapter 25 for general information on event and alarm logging.
- *Digital-Out:* On units equipped with a *Digital I/O* contact, the *Digital-Out* pins can be used as an *alarm target*. Similar to the 'ON' LED, digital-out provides a *summary alarm* function, where the 'gate' is *closed* when the switch is operating 'OK', and *open* when any of the associated alarm triggers becomes active (or when the unit has no power).

  See section 24.4 for general information on Digital I/O.
- *'ON' LED:* There 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.

  See section 24.5 for general information on front panel LEDs.
- *Reboot:* (USE WITH CARE) The *reboot* target is used to make the unit to reboot upon a specified alarm event. The purpose is to provide a way to reboot the unit on a regular basis (i.e., by mapping a timer trigger to an action profile with *target reboot*, see section 24.3.2.8).

In addition, an operator can view the alarm status via the Web and CLI interfaces.

24.1.5.1 Summary alarm

The *summary alarm* in use by the *digital-out* and *ON LED* targets assumes that every alarm trigger define the condition when the alarm is *active* ("alarm" situation) and *inactive* ("normal" situation).

- For many triggers this definition is implicit, e.g., a link alarm is active when the port (or interface) is *down* and inactive it is *up*.
- Other triggers, such as temperature or digital-in sensor triggers allow for the operator to define if the alarm is active: high or low temperature, voltage signal present or not present, etc. See section 24.1.3.2, and in particular figs. 24.2 and 24.3, for further information on the *active* and *inactive* trigger states.

Working as a summary alarm, digital-out as well as the ON LED will indicate 'alarm' as soon as any of the associated alarm triggers become active. For the

ON LED alarm is indicated with a ’red’ light, as shown in fig. 24.4. For Digital-Out, alarm is indicated by having the gate in ’open’ state. See sections 24.4 and 24.5 for general information on Digital I/O and front panel LEDs.

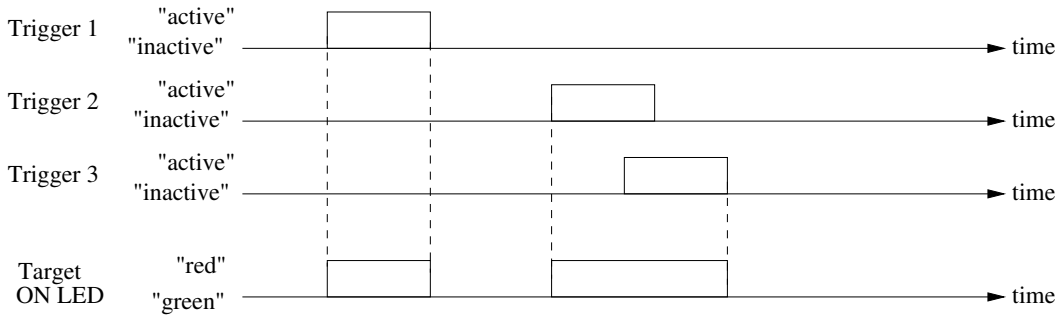


Figure 24.4: Summary alarm example with three alarm triggers mapped to the ON LED alarm target. The ON LED indicates ’alarm’ (red) when any of the associated triggers are active.

24.1.5.2 Target Severity thresholds

As of WeOS v4.17.1 setting target severity thresholds is not yet supported.

For *logging* and *SNMP trap* targets it is possible to filter alarm events depending on *severity*. E.g., if the SNMP trap target configures its severity threshold to *WARNING*, only events of severity level *WARNING* or higher will cause SNMP traps to be sent.

By default, both logging and SNMP trap targets have severity threshold set to level INFO. See section 24.1.3.4 for information on how to classify the severity for alarm triggers.

24.2 Managing Alarms via the Web

24.2.1 Show alarm status

Alarm status is presented in the *System Overview* and the *Detailed System Overview* web pages, which are described in sections 4.4 and 4.4.2.

Fig. 24.5 shows the *System Overview* page when a *Link Alarm* is activated.

Figure 24.5: The basic system overview page with a link alarm activated.

24.2.2 Trigger configuration overview page

Menu path: Configuration ⇒ Alarm ⇒ Triggers

When entering the Alarm configuration page you will be presented to a list of all alarm triggers configured on your unit, see below.

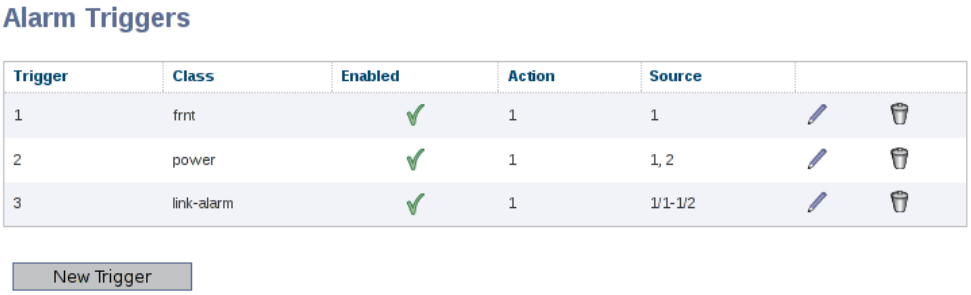


Figure 24.6: The alarm trigger configuration overview page.

Trigger	The index number of this trigger.
Type	The trigger type.
Enabled	A green check-mark means the trigger is enabled, and a dash means it is disabled.
Action	The index of the action profile associated with this trigger. The action profile controls what targets (LED, Digital Out, SNMP traps and/or Logging) to invoke for this alarm trigger.
Source	A list of alarm sources associated with this trigger. For link alarms, this is a list of port numbers, for a power alarm it is the identifiers for the associated power sensors, etc.
Edit	Click this icon to edit a trigger.
Delete	Click this icon to remove a trigger.
New Trigger	Click this button to create a new alarm trigger. You will be presented to a form where you can configure the new trigger.

24.2.3 Create a new alarm trigger using the web interface

Menu path: Configuration ⇒ Alarm ⇒ Triggers ⇒ **New Trigger**

When clicking the **New Trigger** button you will be presented to list of trigger types. Select the trigger type and click next to continue.

Figure 24.7: The trigger type selection page.

When clicking the **Next** button you will be presented to the **New trigger** page.

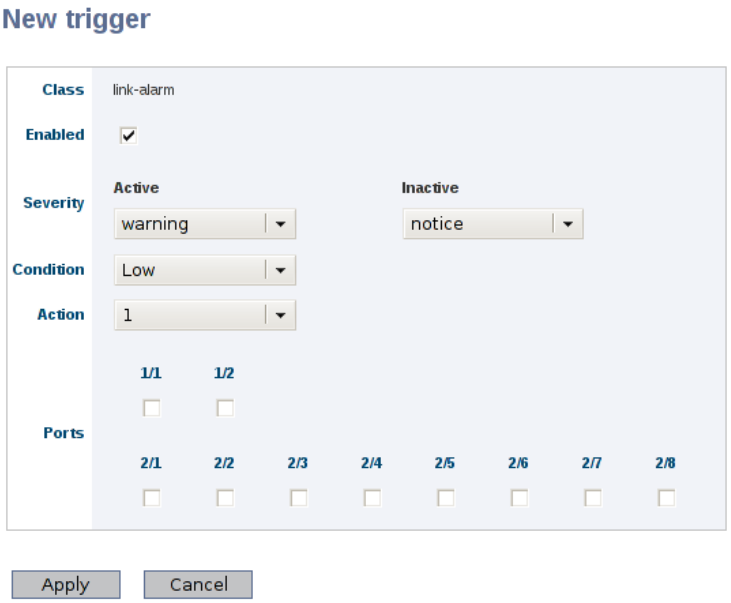


Figure 24.8: The alarm trigger creation page.

Type	The type of alarm trigger.

Continued on next page

	Continued from previous page
Enabled	To enable the trigger - check the box, to disable uncheck the box.
Severity Active	Severity level when active
Severity Inactive	Severity level when inactive
Condition	Controls the condition for triggering (High/low)
Sensors	The sensor source for this trigger
Threshold Rising	The Rising threshold is the higher threshold value for the sensor. When the current sample value is higher than this value, and the last sample was lower than this value, an action is triggered. Valid for none binary sensors such as temperature and SNR.
Threshold Falling	The falling threshold is the lower threshold value for the sensor. When the current sample value is less than this value, and the last sample was greater than this value, an action is triggered. Valid for none binary sensors such as temperature and SNR.
Action	Selects the action for the trigger
Port	The ports on your switch is grouped as on the actual hardware, in slots. To get alarms for a a specific port, check the check-box located underneath the port label. In the picture above you see ports 1/1, 1/2 and 2/1 are marked as alarm sources for this link alarm trigger.

24.2.4 Create a new alarm trigger with sensor value

Triggers controlled by an analogue sensor, must be configure with threshold value. E.g. if you want to create a trigger that alarms if the temperature gets above a given temperature, you must set the rising threshold value to the alarm temperature. The falling thresholds may be set to the same value, but by using different thresholds (rising higher than falling) one can avoid receiving multiple events when the temperature fluctuates around the alarm threshold.

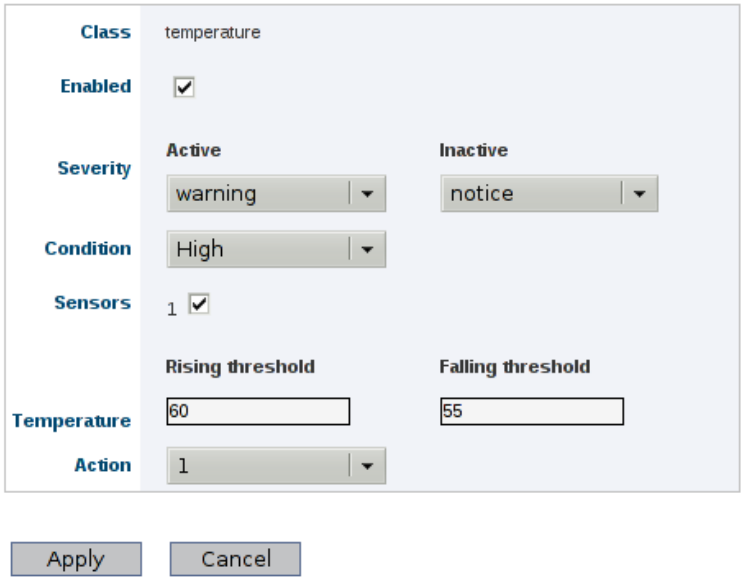


Figure 24.9: Example of a temperature trigger.

24.2.5 Action configuration overview page

Menu path: Configuration ⇒ Alarm ⇒ Actions

When entering the Alarm action configuration page you will be presented to a list of all alarm actions configured on your unit, see below.

Figure 24.10: The alarm action configuration overview page.

Action	The index number of this action.
Targets	The targets for this action.
Edit	Click this icon to edit an action.
Delete	Click this icon to remove an action.
New action	Click this button to add a new alarm action. You will be presented to a form where you can configure the new action.

24.3 Managing Alarms via the CLI

The table below shows alarm management features available via the CLI.

Command	Default	Section
Configure Alarm Configuration Settings		
alarm		Section 24.3.1
[no] trigger <<INDEX> | <TYPE>>		Section 24.3.2
[no] enable	Enabled	Section 24.3.3
[no] <port <PORTLIST> | sensor <SENSORIDLIST> | ring <FRNTINSTANCE> timeout <TIMESPEC> peer <FQDN|IPADDR>		Section 24.3.4
[no] severity <<LEVEL> | [active <LEVEL>] | [inactive <LEVEL>]>		Section 24.3.5
condition <high|low>		Section 24.3.6
threshold <NUM | [rising <NUM>] | [falling <NUM>]>	rising 0 falling 0	Section 24.3.7
[no] interval <SECONDS>	3	Section 24.3.8
[no] number <NUM>	3	Section 24.3.9
[no] outbound <IFNAME>	Disabled	Section 24.3.10
[no] action <INDEX>	1	Section 24.3.11
show types		Section 24.3.12
[no] action <INDEX>		Section 24.3.13
[no] target <[log] [snmp] [led] > [digout] [reboot] [custom]>	log	Section 24.3.14
[no] custom <COMMAND>	Disabled	Section 24.3.15
[no] summary-trap		Section 24.3.16
Alarm Status		
alarm		Section 24.3.17
show		Section 24.3.18

24.3.1 Managing Alarm Settings

Syntax `alarm`

Context Global Configuration context

Usage Enter the Alarm Configuration context.

Use command **"show alarm"** to list an overview global alarm settings as well as configured alarm triggers and actions.

Default values Not applicable.

24.3.2 Manage Alarm Triggers

Syntax `[no] trigger <<INDEX> | <TYPE>>`

Context Alarm Configuration context

Usage Enter the Alarm Trigger Configuration to create, remove or update an alarm trigger.

- Use **"trigger <TYPE>"** to create a new trigger and enter the Trigger context, e.g., **"trigger link-alarm"** to create a new link-alarm trigger.

  Use **"show types"** (section 24.3.12) to list supported trigger types.

  An index will be assigned to each created index. This index can be used to update or remove the trigger, see items below.

- Use **"trigger <INDEX>"** to manage an existing trigger.

- Use **"no trigger <INDEX>"** to remove an existing trigger.

Use command **"show trigger"** to list configured alarm triggers. This is useful to find the index of a trigger, which is needed to edit or remove an existing trigger, see above.

Default values Not applicable.

Some examples of alarm trigger configurations are given in sections 24.3.2.1-24.3.2.11. Details of individual alarm trigger configuration settings are given in sections 24.3.3-24.3.11.

24.3.2.1 Link Alarm Trigger Configuration Example

Syntax `trigger link-alarm`

Context Alarm Configuration context

Usage Create a link-alarm trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for link-alarm triggers are listed below. The only mandatory setting is the list of ports - no link-alarm alarm events will occur until ports are defined.

- Port(s) (mandatory): Define the port or ports this link-alarm trigger is associated with.
- Enable/Disable: By default, the trigger is enabled.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

Example

```
example:/#> configure
example:/config/#> alarm
example:/config/alarm/#> trigger link-alarm
Created trigger 2
example:/config/alarm/trigger-2/#> port 1-2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
=========================================
      1  power       YES           1  1 2
      2  link-alarm  YES           1  1 2

Action  Targets
================================================================================
      1  snmp log led digout

================================================================================
Summary alarm traps: Disabled

example:/config/alarm/#>
```

24.3.2.2 Digital-In Trigger Configuration Example

Syntax `trigger digin`

Context Alarm Configuration context

Usage Create a digital-in trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for digital-in triggers are listed below.

- Sensor: By default, digital-in sensor with ID 1 is used. Use **"show env"** (in Admin Exec context) to list available sensors, see section 7.3.50.
- Condition: By default, the alarm condition is set to *low*. That is, *high* is considered normal and *low* is considered an alarm situation.
- Enable/Disable: By default, the trigger is enabled.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

Example

```
example:/#> configure
example:/config/#> alarm
example:/config/alarm/#> trigger digin
Created trigger 2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
=========================================
      1  power       YES           1  1 2
      2  digin       YES           1  1

Action  Targets
================================================================================
      1  snmp log led digout

================================================================================
Summary alarm traps: Disabled

example:/config/alarm/#>
```

24.3.2.3 Power Trigger Configuration Example

Syntax `trigger power`

Context Alarm Configuration context

Usage Create a power trigger, and enter the Alarm Trigger Configurationcontext for this trigger.

Additional settings for power triggers are listed below. The only mandatory setting is the list of power sensors - no power alarm events will occur until power sensors are defined.

- Sensor: WeOS units typically have two power sensors; sensor 1 for DC1 and sensor 2 for DC2. Use **"show env"** (in Admin Exec context) to list available sensors, see section 7.3.50.
- Enable/Disable: By default, the trigger is enabled.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

Example

```
example:/#> configure
example:/config/#> alarm
example:/config/alarm/#> trigger power
Created trigger 1
example:/config/alarm/trigger-1/#> sensor 1,2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
=========================================
      1  power       YES           1  1 2

Action  Targets
=================================================================================
      1  snmp log led digout

=================================================================================
Summary alarm traps: Disabled

example:/config/alarm/#>
```

24.3.2.4 SNR-Margin Trigger Configuration Example

Note, this setting only applies to units equipped with DSL ports.

Syntax `trigger snr-margin`

Context Alarm Configuration context

Usage Create a SNR-margin trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for SNR-margin triggers are listed below. The only mandatory setting is the list of (DSL) ports - no snr-margin alarm events will occur until (DSL) ports are defined.

- Port(s) (mandatory): Define the port or ports this SNR-margin trigger is associated with.

  Note: SNR-margin alarms can only be generated for ports where a connection has been established.

- Alarm threshold: As of WeOS v4.17.1 the SNR-margin falling threshold is set to 3 (dB) by default, and the rising threshold to 6 (dB) by default.
- Enable/Disable: By default, the trigger is enabled.
- Condition: By default, the alarm condition is set to *low*. That is, *high* is considered normal and *low* is considered an alarm situation.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

In this example an SNR-margin trigger is created for DSL ports 1/1 and 1/2, with falling threshold 4 dB and rising threshold 6 dB.

Example

```
wolverine:/#> configure
wolverine:/config/#> alarm
wolverine:/config/alarm/#> trigger snr-margin
Created trigger 2
wolverine:/config/alarm/trigger-2/#> port 1/1-1/2
wolverine:/config/alarm/trigger-2/#> threshold falling 4 rising 6
wolverine:/config/alarm/trigger-2/#> end
wolverine:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
================================================
      1  power       YES           1  1 2
      2  snr-margin  YES           1  1/1 1/2

Action  Targets
================================================================================
      1  snmp log led digout

================================================================================
Summary alarm traps: Disabled

wolverine:/config/alarm/#>
```

24.3.2.5 Temperature Trigger Configuration Example

Syntax `trigger temperature`

Context Alarm Configuration context

Usage Create a temperature trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for temperature triggers are listed below. The only mandatory setting is the temperature sensor (or list of sensors) - no temperature alarm events will occur until a sensor is defined.

- Sensor(s): Define the temperature sensor(s) this temperature trigger is associated with (default is temperature sensor is ”1”). Use **”show env”** (in Admin Exec context) to list available sensors, see section 7.3.50.
- Alarm threshold: As of WeOS v4.17.1 the temperature falling threshold and rising threshold are both set to 0°C by default.
- Enable/Disable: By default, the trigger is enabled.
- Condition: By default, the alarm condition is set to *high*. That is, temperatures below the falling threshold are considered normal, and temperatures above the rising threshold is considered an alarm situation.

- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

In this example two temperature triggers are created, one to give alarm if the temperature drops below 10°C, and a second trigger to create an alarm if the temperature rises above 60°C.

Example

```
example:/config/alarm/#> trigger temperature
example:/config/alarm/trigger-2/#> sensor 1
example:/config/alarm/trigger-2/#> threshold falling -10 rising -5
example:/config/alarm/trigger-2/#> condition low
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> trigger temperature
example:/config/alarm/trigger-3/#> sensor 1
example:/config/alarm/trigger-3/#> threshold falling 55 rising 60
example:/config/alarm/trigger-3/#> condition high
example:/config/alarm/trigger-3/#> end
example:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
================================================================================
      1  frnt        YES           1  1
      2  temperature YES           1  1
      3  temperature YES           1  1

Action  Targets
================================================================================
      1  snmp log led digout
================================================================================
Summary alarm traps: Disabled

example:/config/alarm/#>
```

24.3.2.6 FRNT Trigger Configuration Example

An FRNT trigger exists in the factory default configuration. Thus, when FRNT is enabled, FRNT alarms will be presented on the default alarm targets without requiring the user to create a trigger.

Syntax `trigger frnt`

Context Alarm Configuration context

Usage Create an FRNT trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for digital-in triggers are listed below.

- Ring: By default, FRNT ring ID 1 is used (as of WeOS v4.17.1 only a single FRNT ring is supported, thus other values are invalid.) Use **"show env"** (in Admin Exec context) to list available sensors, see section 7.3.50.
- Condition: By default, the alarm condition is set to *down* (or *low*). That is, ring status *up* (*high*) is considered normal and ring *down* (*low*) is considered an alarm situation.
- Enable/Disable: By default, the trigger is enabled.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

Example

```
example:/#> configure
example:/config/#> alarm
example:/config/alarm/#> trigger frnt
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> show
Trigger  Type             Enabled    Action  Source
================================================================================
      1  power            YES             1  1 2
      2  frnt             YES             1  Instance 1

Action  Targets
================================================================================
      1  snmp log led digout
================================================================================
Summary alarm traps: Disabled

example:/config/alarm/#>
```

24.3.2.7 LFF Trigger Configuration Example

Note, this setting only applies to units equipped with SHDSL ports.

Syntax `trigger lff`

Context Alarm Configuration context

Usage Create a Link Fault Forward (LFF) trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for LFF triggers are listed below. The only mandatory setting is the list of (SHDSL) ports - no LFF alarm events will occur until (SHDSL) ports are defined.

- Port(s) (mandatory): Define the port or ports this LFF trigger is associated with.

  Note: LFF alarms are generated both when detecting that the remote SHDSL switch indicated LFF, or when the SHDSL link is down.

- Enable/Disable: By default, the trigger is enabled.
- Condition: By default, the alarm condition is set to *low*. That is, *high* (remote link ”up”) is considered normal and *low* (remote link ”down”) is considered an alarm situation.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

In this example an LFF trigger is created to monitor incoming LFF indications on SHDSL port 1/1.

Example

```
wolverine:/config/alarm/#> trigger lff
wolverine:/config/alarm/trigger-2/#> port 1/1
wolverine:/config/alarm/trigger-2/#> end
wolverine:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
================================================================================
      1  frnt        YES           1  1
      2  lff         YES           1  dsl 1/1

Action  Targets
================================================================================
      1  snmp log led digout
================================================================================
Summary alarm traps: Disabled

wolverine:/config/alarm/#>
```

24.3.2.8 Timer Trigger Configuration Example

Syntax `trigger timer`

Context Alarm Configuration context

Usage Create a timer trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for timer triggers are listed below.

- Timeout time: As of WeOS v4.17.1, only *daily* timeouts can be specified, e.g., **”timeout daily 02:30”**
- Enable/Disable: By default, the trigger is enabled.
- Condition: The condition setting has no meaning for a timer trigger, since as of WeOS v4.17.1 the timer trigger should not affect the *ON LED* or the *digital out* action targets.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

In this example a timer trigger is created to force a switch reboot daily at 02:30 in the morning.

Example

```
example:/config/alarm/#> trigger timer
example:/config/alarm/trigger-2/#> timeout daily 02:30
example:/config/alarm/trigger-2/#> action 2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> action 2
example:/config/alarm/action-2/#> target log reboot
example:/config/alarm/action-2/#> end
example:/config/alarm/#> show
Trigger Class       Enabled  Action  Source
================================================================================
      1 frnt        YES           1  Instance 1
      2 timer       YES           2  daily 02:30

Action  Targets
================================================================================
      1  snmp log led digout
      2  log reboot

================================================================================
Summary alarm traps: Disabled
```

24.3.2.9 Ping Trigger Configuration Example

Syntax `trigger ping`

Context Alarm Configuration context

Usage Create a ping trigger, and enter the Alarm Trigger Configuration context for this trigger. The ping trigger monitors the network connectivity (i.e., network reachability) to a given host, using the *ping* command.

Associated with the ping trigger are the following settings:

- *peer:* The host to test the connectivity against.
- *interval:* the ping *interval* can be configured (see section 24.3.8)
- *number:* a *robustness threshold*, i.e., number of failed (or successful, depending on the *condition*) pings required to consider the remote host to be unreachable (or reachable), see section 24.3.9)
- *outbound:* to force ping to use a specific interface. Useful with dynamic VRRP priority (see section 30.1.1), where you do not want to rely on the system default gateway.

In this example a ping trigger is created and mapped to the default action profile, to indicate alarm when the peer become unreachable after 3 retries.

Example

```
example:/config/alarm/#> trigger ping
Trigger 2: Peer is mandatory
example:/config/alarm/trigger-2/#> peer bbc.co.uk
example:/config/alarm/trigger-2/#> number 3
example:/config/alarm/trigger-2/#> interval 3
example:/config/alarm/trigger-2/#> action 2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
================================================================================
      1  frnt        YES           1  Instance 1
      2  ping        YES           1  peer bbc.co.uk

Action  Targets
================================================================================
      1  snmp log led digout

================================================================================
Summary alarm traps: Disabled

example:/config/alarm/#>
```

In this example a ping trigger is created to trigger digital out when the peer become reachable, to do this change the condition argument (default: low).

Example

```
example:/config/alarm/#> trigger ping
Trigger 2: Peer is mandatory
example:/config/alarm/trigger-2/#> peer bbc.co.uk
example:/config/alarm/trigger-2/#> number 3
example:/config/alarm/trigger-2/#> interval 3
example:/config/alarm/trigger-2/#> condition high
example:/config/alarm/trigger-2/#> action 2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> action 2
example:/config/alarm/action-2/#> target digout
example:/config/alarm/action-2/#> end
example:/config/alarm/#> show
Trigger  Type        Enabled  Action  Source
================================================================================
      1  frnt        YES           1  Instance 1
      2  ping        YES           2  peer bbc.co.uk

Action  Targets
================================================================================
      1  snmp log led digout
      2  log digout

================================================================================
Summary alarm traps: Disabled
```

24.3.2.10 PoE Power Usage Trigger Configuration Example

Syntax `trigger poe`

Context Alarm Configuration context

Usage Create a PoE power usage trigger, and enter the Alarm Trigger Configuration context for this trigger. The power usage is defined as the percentage of consumed/maximum power.

Additional settings for temperature triggers are listed below. The only mandatory setting is the temperature sensor (or list of sensors) - no temperature alarm events will occur until a sensor is defined.

- Alarm threshold: Set the threshold to usage level (1-99 (%)) when an alarm is desired. By default, the rising threshold is set to 95(%) and the falling threshold to 90(%).

- Enable/Disable: By default, the trigger is enabled.
- Condition: By default, the alarm condition is set to *high*. That is, usage levels below the falling threshold are considered normal, and temperatures above the rising threshold is considered an alarm situation.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

In this example a PoE trigger is created to give an alarm if the usage rises above 80%.

Example

```
viper:/config/alarm/#> trigger poe
viper:/config/alarm/trigger-2/#> threshold rising 80 falling 75
viper:/config/alarm/trigger-2/#> condition high
viper:/config/alarm/trigger-2/#> end
viper:/config/alarm/#> show
Trigger  Type         Enabled    Action  Source
================================================================================
      1  frnt         YES             1  Instance 1
      2  poe          YES             1  1

Action  Targets
================================================================================
      1  snmp log led

================================================================================
Summary alarm traps: Disabled

viper:/config/alarm/#>
```

24.3.2.11 Microlok Trigger Configuration Example

Syntax `trigger microlok`

Context Alarm Configuration context

Usage Create a Microlok session summary alarm trigger, and enter the Alarm Trigger Configuration context for this trigger.

Additional settings for link-alarm triggers are listed below. As of WeOS v4.17.1 there can only be one Microlok Gateway instance, thus the gateway instance (i.e., instance 1) is implicit.

- Enable/Disable: By default, the trigger is enabled.
- Severity: By default, active severity is *WARNING* and inactive severity is *NOTIFY*.
- Action: By default, the trigger is mapped to the default action profile (action 1).

Example

```
example:/config/microlok-1/#> map station 1a serial 1 session-timeout 2000
example:/config/microlok-1/#> map station 1b serial 1 session-timeout 2000
example:/config/microlok-1/#> map station 2a remote 192.168.2.1 session-
timeout 2000
example:/config/microlok-1/#> map station 2b remote 192.168.2.1 session-
timeout 2000
example:/config/microlok-1/#> end
example:/config/#> alarm
example:/config/alarm/#> trigger microlok
example:/config/alarm/trigger-2/#> action 2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> action 2
example:/config/alarm/action-2/#> target log digout
example:/config/alarm/action-2/#> end
example:/config/alarm/#> show
Trigger  Type         Enabled    Action  Source
================================================================================
      1  frnt         YES             1  Instance 1
      2  microlok     YES             2  1

Action  Targets
================================================================================
      1  snmp log led digout
      2  log digout

================================================================================
Summary alarm traps: Disabled

example:/config/alarm/#>
```

24.3.3 Enable/disable a Trigger

Syntax `[no] enable`

Context Alarm Trigger Configuration context

Usage Enable or disable an alarm trigger. A disabled trigger will keep its configuration settings, but will not affect any alarm targets.

Use **”enable”** to enable and **”no enable”** to disable a trigger.

Use **”show enable”** to show whether this trigger is *enabled* or *disabled*.

Default values Enabled

24.3.4 Manage alarm sources

Syntax `[no] <port <PORTLIST> | sensor <SENSORIDLIST> | ring <FRNTINSTANCE> timeout <daily <HH:MM>>>`

Context Alarm Trigger Configuration context

Usage Specify which alarm sources the trigger should monitor. The command syntax differs depending on the trigger type:

- Use **”[no] port <PORTLIST>”** to specify which port(s) a *link-alarm* trigger should apply to, e.g., use **”port 1/1,2/2-2/4”** to add ports 1/1, and 2/2-2/4 to the list of ports monitored by this link-alarm trigger.
- Use **”[no] ring <FRNTINSTANCE>”** to specify which FRNT ring an FRNT alarm trigger should apply to.
- Use **”[no] sensor <SENSORIDLIST>”** to specify which sensors a *digital in*, *power* or *temperature* trigger should apply to, e.g., use **”sensor 1,2”** to add power sensors 1 and 2 to the list of power sensors monitored by this power trigger.

  Use command `show env` (section 7.3.50) to list available sensors and their index values.
- Use **”[no] timeout <daily <HH:MM>>”** to specify how often and when an timer trigger should go off, e.g., use **”timeout daily 02:30”** to make the timer trigger to go off every day at 02:30 in the morning.
- Use **”[no] peer <FQDN|IPADDR>”** to specify the peer (domain name or IP address) to test the connectivity to.

  ”no peer” will delete the configured peer, however, having a *ping* trigger without a configured *peer* is not a valid setting.

Use **”no port <PORTLIST>”** remove a specific set of ports, or **”no port”** to remove all ports from a trigger (the same goes for other source types).

If no sources are defined when exiting the trigger context, the trigger will automatically be configured as *disabled* (see section 24.3.3).

Use command **"show "** to show the alarm sources associated with this trigger. The type of alarm source differs depending on the trigger type. See section 24.3.4 for more information.

Default values

24.3.5 Alarm Event Severity

Syntax `[no] severity <<LEVEL>|[active <LEVEL>]|[inactive <LEVEL>]>`

Context Alarm Trigger Configuration context

Usage Specify the severity level of *active* and *inactive* alarm events detected by this trigger. See section 24.1.3.4 for information on available severity levels.

Active and inactive severity levels can be configured together or independently.

"no severity" to will 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 digital-out and ON LED targets.

Use **"show severity"** to show the severity setting (*active* and *inactive* severity) for this trigger.

Default values `active warning` and `inactive notice`

The examples below show how to set severity level for active and inactive alarm events together and how to set it individually. The final example shows how to set severity 'NONE' for both active and inactive events.

Example

```
example:/config/alarm/trigger-2/#> severity err
example:/config/alarm/trigger-2/#> show severity
active err, inactive err
example:/config/alarm/trigger-2/#> severity inactive debug
example:/config/alarm/trigger-2/#> show severity
active err, inactive debug
example:/config/alarm/trigger-2/#>
example:/config/alarm/trigger-2/#> no severity
example:/config/alarm/trigger-2/#> show severity
active none, inactive none
example:/config/alarm/trigger-2/#>
```

24.3.6 Configure Alarm Condition Setting

Syntax `condition <high|low>`

Alternate keywords are possible:

- *rising* and *up* are equivalents to *high*.
- *falling* and *down* are equivalents to *low*.

Context Alarm Trigger Configuration context

Usage Define whether the *high* or *low* trigger state should be considered the *alarm state*, while the other is considered the *normal state*.

Some triggers, such as *link-alarm* and *power* triggers have a static (predefined) alarm condition setting. (Both link-alarm and power triggers have *condition* set to *low*). For other triggers, the alarm condition setting is configurable.

See section 24.1.3.2 for more information.

Use **"`show condition`"** to show the alarm condition setting for this trigger.

Default values Differs for different trigger types

24.3.7 Configure Rising and Falling Thresholds

Syntax `threshold <NUM|[rising <NUM>]|[falling <NUM>]>`

Context Alarm Trigger Configuration context

Usage Set falling and rising thresholds. The thresholds may be set to the same value, but by using different thresholds (rising higher than falling) one can avoid receiving multiple events when the alarm source fluctuates around the alarm threshold.

Triggers which are *binary* to their nature, such as *link-alarm*, *power*, and digital-in triggers have implicit thresholds, which cannot be configured.

See section 24.1.3.2 for more information.

Use command **"`show threshold`"** to show the trigger threshold setting (both *rising* and *falling* thresholds) for this trigger.

Default values `rising 0` and `falling 0` (except for *binary* alarm sources)

24.3.8 Configure Ping Interval

Syntax `[no] interval <SEC>`

Context Alarm Trigger Configuration context

Usage Specify the interval between ICMP Ping.

Use command **"show interval"** to show the ping trigger *pinging interval* setting, i.e., interval of which ping messages are sent to probe the reachability to the peer.

24.3.9 Configure Ping Robustness Number

Syntax `[no] number <NUM>`

Context Alarm Trigger Configuration context (ping trigger)

Usage Specify the number of ICMP ping that should be lost (or received) to determine if a host is unreachable (or reachable).

Use command **"show number"** to show the ping trigger robustness number setting, i.e., the number of pings required to be lost before the peer is considered unreachable, or the number of pings required to succeed before the peer is considered reachable.

24.3.10 Configure Ping Outbound Interface

Syntax `[no] outbound <IFNAME>`

Context Alarm Trigger Configuration context (ping trigger)

Usage Force pings to use a specific outbound interface. This is very useful when tracking upstreams connectivity in a VRRP dynamic priority scenario (see section 30.1.1). Because then you want to make sure the default gateway, or any other route, is avoided.

Use **"no outbound"** to disable the setting. This makes ping rely on network routes and fall back to use the *default gateway*.

Use command **"show outbound"** to show the configured *outbound interface* for this ping trigger. When unset, **"Default Gateway"** is shown and the

system will use the system default route, or a matching network route, for ping packets.

Default values Disabled (default gateway)

24.3.11 Configure Trigger Action

Syntax `[no] action <INDEX>`

Context Alarm Trigger Configuration context

Usage Specify the action (profile) to be invoked when this trigger detects an alarm event.

Use **"no action"** to disable the mapping to an alarm action. E.g., when in use by another subsystem (e.g., VRRP with dynamic priority, see section 30.1.1), or if you simply want to temporarily disable or debug your alarms.

Use command **"show action"** to show the action profile mapped to this trigger.

Default values 1 (default action)

24.3.12 Show Supported Trigger Types

Syntax `show types`

Context Alarm Configuration context

Usage List supported trigger types. These are the types to be used with the **"trigger <TYPE>"** command (see section 24.3.2).

Default values Not applicable

24.3.13 Manage Alarm Actions

Syntax `[no] action <INDEX>`

Context Alarm Configuration context

Usage Create, remove or update an alarm action (profile). Use **"action <INDEX>"** to enter the Alarm Action Configuration context and create a new or update an existing action.

Use **"no action <INDEX>"** remove an existing action. The default action (index 1) cannot be removed, but you can disable all targets.

Use command **"show action"** to list all configured alarm action profiles, or **"show action <ID>"** to show detailed configuration information on a specific action profile (also available as **"show"** command within the Alarm Action Configuration of that profile).

Default values Not applicable.

24.3.14 Manage Action Targets

Syntax `[no] target <[log] [snmp] [led] [digout] [reboot] [custom]>`

Context Alarm Action Configuration context

Usage Add or remove alarm target to an alarm action (profile).

- led: Set ON/Status LED
- log: Log status change to syslog
- snmp: Generate an SNMP trap
- digout: Control digital out relay
- reboot: Reboot the unit. USE WITH CAUTION!
- custom: Run any admin-exec level command. DEPRECATED!

> **Warning**
> The **"custom"** target is for experimental purposes only! A .conf file containing **"target custom"** and **"custom reboot"** (see section 24.3.15) will be translated to **"target reboot"** automatically. That is to be backwards compatible. Other **"custom"** commands are not guaranteed to be supported in future releases.

Use command **"show target"** to show the alarm target(s) configured for this action profile.

Default values `target log` (New action profiles has **"target log"** as default.

24.3.15 Set Custom Action Target

Syntax `[no] custom <COMMAND>`

Context Alarm Action Configuration context

Usage Set custom action command. The custom target allows the user to connect, e.g., a timer trigger to a CLI Admin Exec level command, such as **"reboot"**, see section 7.3.27.

> **Warning**
> This is a deprecated feature not guaranteed to be supported in future releases. For experimental purposes only!

Use **"no custom"** to remove a custom command.

Use command **"show custom"** to show the configured custom action command configured for this action profile.

Default values Disabled

Examples See section 24.3.2.8.

24.3.16 Enable/disable Summary Alarm Traps

Syntax `[no] summary-trap`

Context Alarm Configuration context

Usage Enable or disable summary alarm traps. When enabled, a trap will be sent whenever the summary alarm status changes (from *OK* to *Warning* or vice versa). The summary alarm status follows the status of the *ON LED*. See also section 6.1.3 for more information summary alarm status and its associated SNMP trap, and see sections 24.1.5.1 and 24.3.14 for more information on the *ON LED* alarm target.

Use **"summary-trap"** to enable and **"no summary-trap"** to disable a SNMP traps for the summary alarm status.

Use **"show summary-trap"** to show whether summary alarm traps are *enabled* or *disabled*.

Default values Disabled

Example

```
example:#> configure
example:/config/#> alarm
example:/config/alarm/#> summary-trap
example:/config/alarm/#> show summary-trap
Enabled
example:/config/alarm/#> end
example:/config/#>
```

24.3.17 Handling Alarm Status

Syntax `alarm`

Context Admin Exec context

Usage Enter the Alarm Status context.

Default values Not applicable.

24.3.18 Show overall alarm status

Syntax `show`

Context Alarm Status context

Usage Show status of all alarms.

Default values Not applicable.

24.4 Digital I/O

WeOS products are typically equipped with a *Digital I/O* connector as the one shown in fig. 24.11. The location of the connector differs between products; on RedFox Industrial it is located on the CPU card as shown in fig. 24.12).

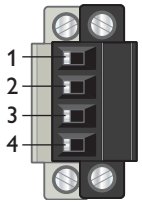


Figure 24.11: Digital I/O connector.

The Pin-Out of the Digital I/O connector is *typically* as follows:

Position	Description
1	Digital-Out + (Relay Output +)
2	Digital-Out - (Relay Output -)
3	Digital-In +
4	Digital-In -

Note
For a detailed specification on the Digital I/O connector (definite pin-out mapping, voltage levels, etc.), see the *User Guide* of your specific WeOS product (section 1.5).

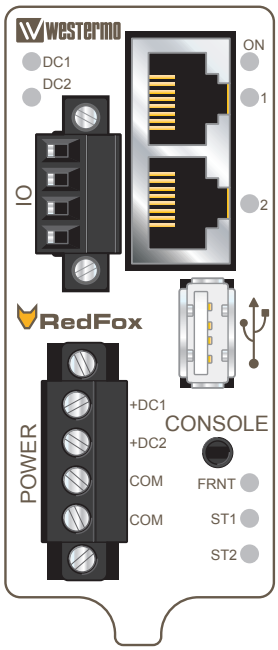
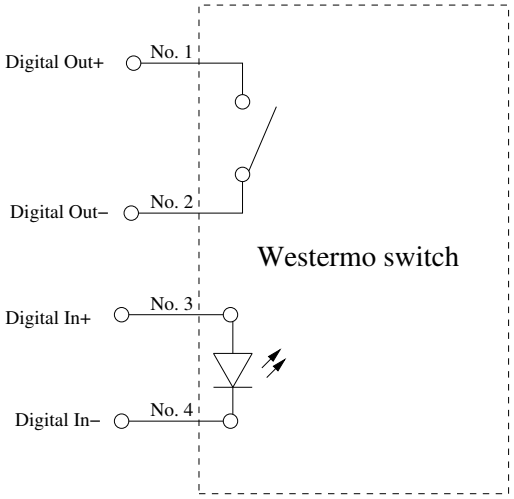


Figure 24.12: The Power and CPU module of a RedFox Industrial unit



As described in section 24.1, *Digital-In* can be used as an alarm source, while *Digital-Out* is utilised as an alarm target (*summary alarm*).

- The Digital-In alarm is triggered when there is *lack of* voltage on the Digital-In pins. For information on appropriate voltage/current levels to trigger

alarms via Digital-In, see the *User Guide* of your specific product (section 1.5).

- The Digital-Out pins are internally connected to a *gate*. The gate is *open* when the switch has no power, or when any *alarm sources* are active. When the switch is operating normally (the switch has booted up, and no alarm source is active), the gate is *closed*.

24.5 LEDs

The LED functionality when running WeOS is described in the *User Guide* of your product (section 1.5). Here the information on LED functionality of *all* WeOS products is summarised. Note that your product may not have all LED types listed here.

LED	Status	Description
ON	OFF	Unit has no power
	GREEN	All OK, no alarm condition.
	RED	Alarm condition, or until unit has started up. (Alarm conditions are configurable, see sections 24.1-24.3.)
	GREEN BLINK	Location indicator (”Here I am!”). Activated when connected to WeConfig Tool, or upon request from Web, or when entering the CLI configuration context. Duration of blinking: 10 seconds.
	RED BLINK	Location indicator (see previous item) **or** indication of pending cable factory reset, see section 7.1.3.3.
DC1[1]	OFF	Unit has no power.
	GREEN	Power OK on DC1.
	RED	Power failure on DC1.
DC2[1]	OFF	Unit has no power.
	GREEN	Power OK on DC2.
	RED	Power failure on DC2.
AC1	OFF	Unit has no power.
	GREEN	Power OK on AC1.
FRNT	OFF	FRNT disabled
	GREEN	FRNT OK. (See also the *FRNT Error* item below.)
	RED	FRNT Error. A focal point can detect and indicate local FRNT errors (FRNT link down) as well as FRNT errors elsewhere in the FRNT ring. A member switch only detects and indicates local FRNT errors (FRNT link down).
	BLINK	Unit configured as focal point.
RSTP	OFF	RSTP disabled.

Continued on next page

Continued from previous page

LED	Status	Description
(formerly ST1)	GREEN	RSTP enabled.
	BLINK	Unit elected as RSTP/STP root switch.
USR1/VPN[2] (formerly ST2)	OFF	VPN disabled[3].
	GREEN	At least one VPN tunnel up and OK[3].
	RED	All VPN tunnels down[3].
Ethernet ports	OFF	No link.
	GREEN	Link established.
	GREEN FLASH	Data traffic indication.
	YELLOW	Port alarm and no link. Or if FRNT, RSTP or Link Aggregation mode, port is blocked.
SHDSL ports	OFF	No SHDSL link.
	GREEN	SHDSL link established.
	GREEN BLINK	SHDSL link negotiation.
	GREEN FLASH	Data traffic indication.
	YELLOW	Port alarm and no link. Or if FRNT or RSTP mode, port is blocked.
	YELLOW BLINK	Only during unit startup. Firmware downloading to SHDSL chip.
SHDSL Link Quality Indicator	All OFF	No SHDSL link.
	3 RED	SNR below 3 dB. Unstable SHDSL link.
	1 GREEN	SNR 3-5 dB. Marginal SHDSL link.
	2 GREEN	SNR 6-9 dB. Normal SHDSL link.
	3 GREEN	SNR 10 dB or higher. Strong SHDSL link.
ADSL/ VDSL ports	OFF	No xDSL link.
	GREEN	xDSL link established.
	GREEN BLINK	xDSL link negotiation.
TD	OFF	No serial data received.
	GREEN FLASH	Serial data received.
	YELLOW FLASH[4]	Error on RS-422/485 bus.
RD	OFF	No serial data transmitted.
	GREEN FLASH	Serial data transmitted.

Additional explanations:

- BLINK means that the LED is blinking with a frequency about 1 Hz.

- FLASH means that the LED is blinking with a higher frequency.
- SHDSL LEDs only apply to products with SHDSL ports.
- xDSL (ADSL/VDSL) LEDs only apply to products with xDSL ports.
- TD and RD LEDs only apply to products with serial port(s). As the WeOS serial ports operate in DCE mode, TD denotes receiving, and RD denotes transmitting serial data.

[1] Viper has a common power indicator LED named ”DC”. When the ”DC” LED is ”GREEN” power is OK on DC1 and DC2. When the ”DC” LED is ”RED” there is a power failure on DC1 or DC2.

[2] The ”USR1” LED is referred to as ”VPN” on some WeOS products and ”ST2” on older RFI products.

[3] Only for products with software level WeOS Extended. As of WeOS v4.17.1, the USR1/VPN LED presents VPN status as described above. Alternative (configurable) use is intended but not yet supported.

[4] Only applicable for the DDW-142-485[38] product.

Chapter 25

Logging Support

This chapter describes WeOS support for alarm and generic event logging.

In WeOS general events detected by the system (such as user login attempts), as well as alarm events defined by configured alarm triggers (see chapter 24) can be logged for further analysis. Three logging methods are available:

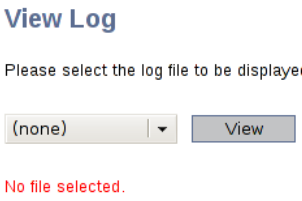
- *Logging to file:* General events and alarm events are always logged to a local log file.
- *Logging to console:* It is possible to direct logging messages to the console port. Messages of severity level *DEBUG* or higher are shown on the console port.
- *Logging to a remote syslog server:* Logging messages can be sent to a remote syslog server for further processing. Messages of severity level *NOTICE* or higher are forwarded to the remote syslog server(s).

As of WeOS v4.17.1 logging support is only available via the CLI. The severity thresholds for console and remote syslog logging are not configurable, however, such support is planned.

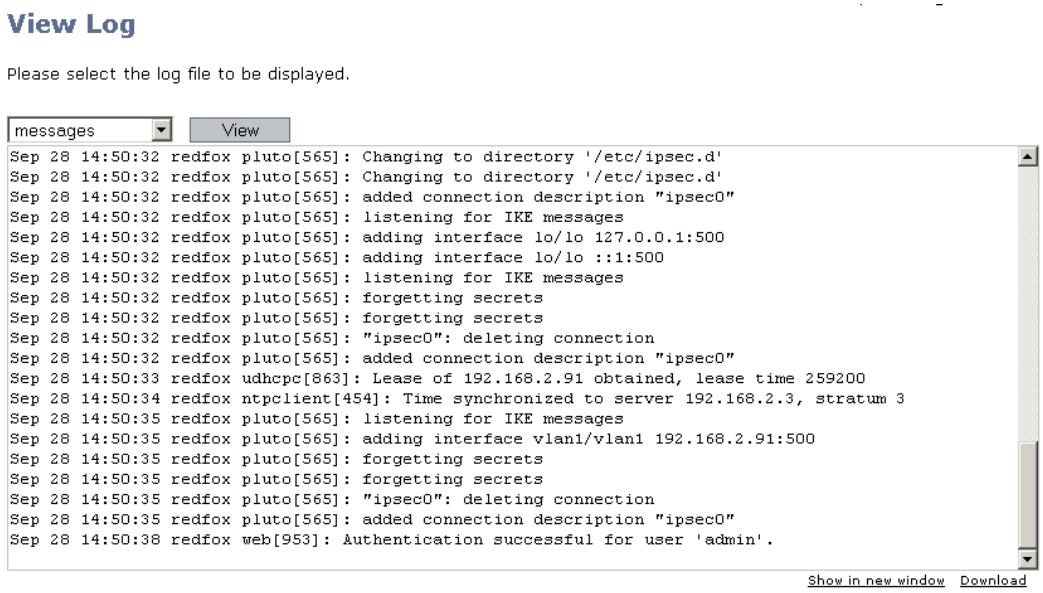
25.1 Logging Support in the web interface

Select the log file in the drop down list and press **View** to the display desired log file.

Menu path: Maintenance ⇒View Log



Select the log file in the drop down list and press **View** to the display desired log file.



25.2 Managing Logging Support via the CLI

Command	Default	Section
Configuring Logging Settings		
[no] logging	Disabled	Section 25.2.1
[no] console		Section 25.2.2
[no] server <ADDRESS1 [ADDRESS2]>	Disabled	Section 25.2.3
Managing Log Files		
dir <cfg:///|log:///|usb://>		Section 7.3.21
copy <FROM_FILE> <TO_FILE>		Section 7.3.22
erase <file>		Section 7.3.23
show <running-config | startup-config | factory-config | [<filesys>://]FILENAME>		Section 7.3.24

25.2.1 Managing Logging Settings

Syntax `[no] logging`

Context Global Configuration context

Usage Enter Logging Configuration context.

Use **”no logging”** to disable all logging (to console and remote syslog server).

Use **”show logging”** to show logging configuration settings. Also available as **”show”** command within the Logging Configuration context.

Default values Disabled

25.2.2 Logging to console port

Syntax `[no] console`

Context Logging Configuration context

Usage Enable or disable console logging.

Use **”console”** to enable console logging, and **”no console”** to disable console logging.

When enabled, general events detected by the system, as well as alarm events associated with configured alarm triggers, will be presented on the console port.

Use **”show console”** to show whether console port logging is enabled or disabled.

Default values *Disabled*

25.2.3 Logging to remote syslog server

Syntax `[no] server <ADDRESS1 [ADDRESS2]>`

Context Logging Configuration context

Usage Set remote syslog server(s) (IPv4 addresses). A maximum of two remote syslog servers are supported. The syntax allows typing them in one line or two separate lines.

Use **”no server <ADDRESS>”** to remove a single server. Use **”no server”** to remove all servers.

When enabled, general events detected by the system, as well as alarm

events associated with configured alarm triggers, will be forwarded to the configured syslog server via UDP to port 514. If two servers are configured, messages are sent to both of them.

Use **”show server”** to show whether remote syslog logging is enabled or disabled. If enabled, the IP address(es) of the configured server(s) are presented.

Default values *Disabled*

Part III

Router/Gateway Services

Chapter 26

IP Routing in WeOS

In addition to *switching* (layer-2), WeOS devices (with proper WeOS level) are able to *route* data packets (layer-3), i.e., they are *routing switches*. The WeOS routing support includes static routing and dynamic unicast routing via OSPF and RIP, static multicast routing, as well as other useful router features such as firewall, NAT, and VRRP.

This chapter introduces the IP routing capabilities in WeOS in general. More information on dynamic routing is found in chapters 27 (OSPF) and 28 (RIP), while static multicast routing support is described in chapter 29. Supplementary router services are covered in the chapters to follow: VRRP in chapter 30, and firewall and NAT in chapter 31.

Support for VPN and tunneling techniques are presented separately, see part IV.

26.1 Summary of WeOS Routing and Router Features

Table 26.1 presents the routing and router features available in WeOS.

26.1.1 Introduction to WeOS Routing and Router Features

IP routing enables us to connect our networks together, and to let (TCP/IP) devices communicate across networks of different type and topology, and possibly over multiple network ”hops” and long distances. A router looks at the destination IP address carried within each IP packet, consults its *routing table* to make a

Feature	Web	CLI	General Description
Enable/disable routing	X	X	Section 26.1.1
Default gateway	X	X	Section 26.1.1
Static unicast routing	X	X	Section 26.1.4
Blackhole routes		X	Section 26.1.4.3
Dynamic unicast routing			
- OSPF	X	X	Section 26.1.1, Chapter 27
- RIP (v1/v2)	X	X	Section 26.1.1, Chapter 28
Static multicast routing	X	X	Section 26.1.1, Chapter 29
View routing table	X	X	
Router redundancy (VRRP)	X	X	Section 26.1.1, Chapter 30
Firewall and NAT	X	X	Section 26.1.1, Chapter 31

Table 26.1: Summary of router and routing features.

routing decision, and forwards the packet onto the next router in the path to the destination.

The routing table can either be managed manually via *static IP routing*, or automatically by using dynamic routing protocols, or a combination of both. Static IP routing is usually fine for small IP networks, or networks with no redundant paths. To manage routing in larger networks, it is preferred to use *dynamic IP routing*. With dynamic routing, the routers will exchange routing information and build up their routing tables dynamically. Furthermore, dynamic routing utilises network redundancy; if a link goes down, routers will inform each other and packets will automatically be routed along another path. Thus, dynamic routing protocols perform a similar service in routed networks as FRNT (chapter 14) and RSTP (chapter 16) perform in switched networks. The time to react on a topology change is referred to as the *convergence time*. WeOS supports two dynamic routing protocols: Open Shortest Path First (OSPF) and Routing Information Protocol (RIP). OSPF is the recommended over RIP, due to its superior *convergence* characteristics.

OSPF and RIP are examples of unicast Interior Gateway Protocols (IGPs), which means they can be used to handle routing *within* a routing domain, such as an corporate network. This is also referred to as *intra-domain* routing, as opposed *inter-domain* routing, which is commonly handled using the Border Gateway Protocol (BGP)[1]. OSPF and RIP are covered in chapters 27 and 28 respectively.

[1]As of WeOS v4.17.1, dynamic routing is limited to intra-domain (unicast) routing with RIP and OSPF. WeOS does *not* support dynamic inter-domain routing via BGP (Border Gateway Protocol), or

IP multicast routing enables efficient distribution of multicast data in a routed network. A *source*, such as an IP camera, will send its data to a specific multicast IP address (also referred to as a multicast group), and *receivers* (the group members) will listen in to this address by joining the group. WeOS supports static multicast routing, which enables the network manager to manually set the multicast routing entries in the routers. Dynamic multicast routing protocols, such as DVMRP or PIM-SM, are not yet supported. See chapter 29 for more details on IP multicast routing.

While dynamic routing protocols such as RIP and OSPF enable routers to find redundant paths in case a link or router goes down, they do not enable end devices (hosts) to use a second router if their *regular* router goes down. To support redundancy between hosts and routers the Virtual Router Redundancy Protocol (VRRP) is used. With VRRP, a backup router will take over if a router fails, and communication from connected hosts can continue automatically. VRRP support is covered in chapter 30.

When a router is used as a company gateway to a public network, such as the Internet, there is an obvious need to protect the local company network against network intrusion and other attacks. It is also common that the hosts and routers within the company network use *private* IP addresses. To protect the company network and to enable the use of private IP addresses, WeOS includes *firewall* and *network address translation* (NAT) support. Chapter 31 describes the NAT and firewall features in WeOS.

Another need which occurs when connecting company networks to the Internet is to ensure communication privacy. WeOS supports IPsec VPN and SSL VPN (OpenVPN) to establish secure communication over public networks. With VPNs, a company can secure communication between a head office and different branch offices by installing a WeOS device as VPN gateway at each site. WeOS VPN support is covered in part IV.

26.1.2 Using a WeOS device as a switch or as a router

WeOS devices are both able to route and to switch packets, i.e., they are *routing switches*. Switching is performed between ports in the same VLAN, while routing is performed between IP subnets or network interfaces (please see fig. 19.1 in section 19.2 for information on the distinction between ports, VLANs and network

dynamic multicast routing.

interfaces in WeOS). Routing can be disabled, and the WeOS device will then act as a VLAN capable *switch*.

26.1.3 Learning routing information from different sources

A WeOS device will learn about routing information by manual configuration (connected interfaces or static routes), dynamic address assignment (e.g., DHCP), or via dynamic routing protocols (OSPF and RIP). As described in chapters 27 and 28, a router is able to redistribute external routing information into an OSPF or RIP routing domain.

In some situations a router will learn the route to the same destination through different mechanisms. In this case, the route to use will depend on the *administrative distance* (or simply ”admin distance”) associated with the involved routing mechanisms. A route with a lower admin distance will be prioritised over a router with higher admin distance.

Connected routes are always preferred (they have admin distance ’0’ (zero)). In WeOS the admin distance of static routes, and routes learnt dynamically via RIP and OSPF can be configured, but defaults to the values shown in the table below. Routes learnt dynamically via DHCP or PPP will have admin distance according to the distance assigned to the associated interface, see section 19.2.6.

	Administrative Distance
Connected	0
Static	1
OSPF	110
RIP	120

Configuring static routes with higher administrative distance than set for OSPF or RIP is also referred to *floating static routes*, see section 26.1.4.2 for further details.

26.1.4 Static routing

WeOS supports static IP routing. With static routing a WeOS devices can specify the next hop router to use to reach a given IP subnet, or add additional (directly attached) subnets to a local interface.

26.1.4.1 Using Static Route with Next-Hop or Interface as target

When defining a static route, the target is typically an IP address, e.g., **”route 192.168.5.0/24 192.168.1.1”** where **”192.168.1.1”** would be the IP address of the next-hop router towards the destination.

In other situations you could define the target as a network interface of your unit, e.g., **”route 192.168.5.0/24 ssl0”** where all traffic towards **”192.168.5.0/24”** would be sent via your SSL VPN interface (chapter 36).

Note
Using an interface as target of a static route is almost only used on point-to-point interfaces, e.g., SSL or GRE interfaces. In rare cases it can be used on LAN interfaces when you have multiple subnets on a VLAN, but in those cases it is often simpler to use a secondary IP address on that LAN interface, see section 19.2.5.

26.1.4.2 Floating Static Routes - Administrative Distance for Static Routes

Floating static routes are static routes with higher *administrative distance* (see section 26.1.3) than routes learnt dynamically, e.g., via routing protocols such as OSPF and RIP, or via dynamic configuration protocols such as DHCP or IPCP (PPP).

An example where a default route acquired via DHCP is given precedence over a floating static (default) route is given in section 19.2.6. To complement this, an example where routes learnt via OSPF is given precedence over a floating static route is illustrated in fig. 26.1.

In this example, the user could have used OSPF over the low-speed backup link, but has instead chosen to use a floating static route. Relevant parts of the configuration at routers 1, 2 and 3 are shown below.

Router 1 injects a default route into the OSPF area, a defines a floating static route towards *192.168.35.0/24* via Router2.

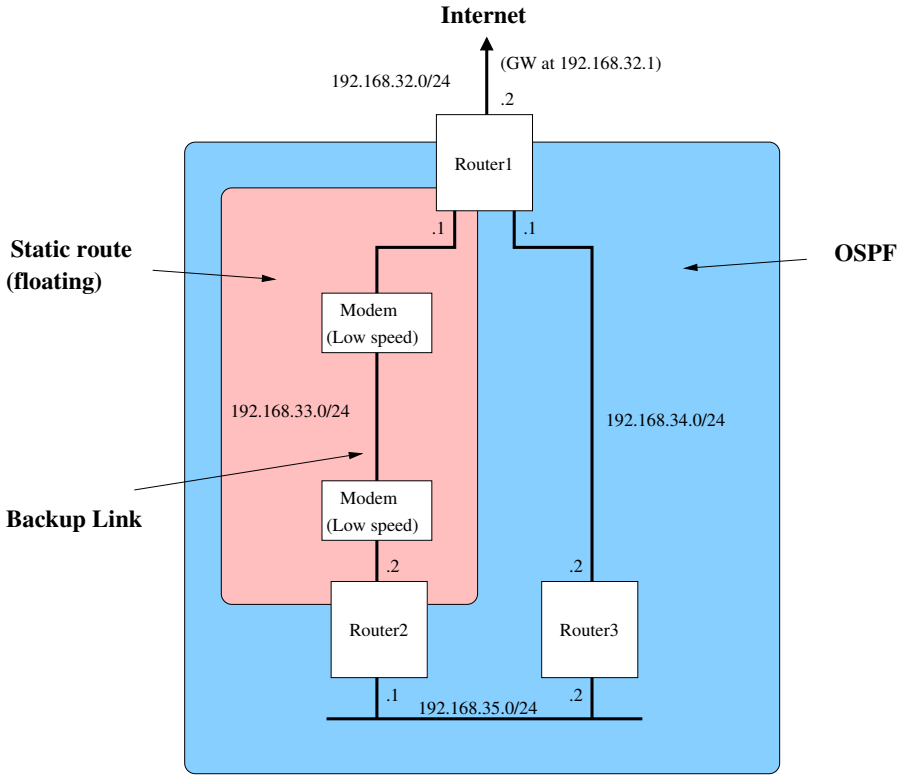


Figure 26.1: Use of floating static route for on low-speed backup link.

Example

```
#Router1
ip
        route 0.0.0.0/0 192.168.32.1
        route 192.168.35.0/24 192.168.33.2 200
        end

router
        ospf
                network 192.168.34.0/24 area 0.0.0.0
                distribute-default always
                end
        end
```

Router 2 defines a floating static default route towards via Router1, and injects a default route into the OSPF area *given* that its floating default route is active (no **"always"** attribute; compare with Router1 configuration).

Example

```
#Router2
ip
        route 0.0.0.0/0 192.168.33.1 200
        end

router
        ospf
                network 192.168.35.0/24 area 0.0.0.0
                distribute-default
                end
        end
```

Router 3 has no static routes, i.e., it only uses OSPF.

Example

```
#Router3
router
        ospf
                network 192.168.34.0/24 area 0.0.0.0
                network 192.168.35.0/24 area 0.0.0.0
                end
        end
```

26.1.4.3 Blackhole routes

WeOS has a *blackhole* interface referred to as **”null0”**. This interface is hidden in the sense that it cannot be configured (no IP address, management settings, etc.). The blackhole interface is useful to avoid routing loops in networks with incomplete subnetting.

An example is shown in fig. 26.2. R1 has set a static route for the ”192.168.0.0/22” range towards R2. R2 only has routes to a part of this range, i.e., the directly connected subnets ”192.168.0.0/24”, ”192.168.1.0/24” and ”192.168.2.0/24”, while ”192.168.3.0/24” is currently unused. As R2 has defined R1 as its default route, a packet sent towards e.g., ”192.168.3.11” would bounce back and forth between R1 and R2, unless R2 defines a blackhole route.

Note

In this example, the static blackhole route for ”192.168.0.0/22” has a shorter prefix than the directly connected routes. Therefore only traffic in range ”192.168.3.0/24” will be sent to ”null0” as long as the interfaces to the directly connected subnets are up.

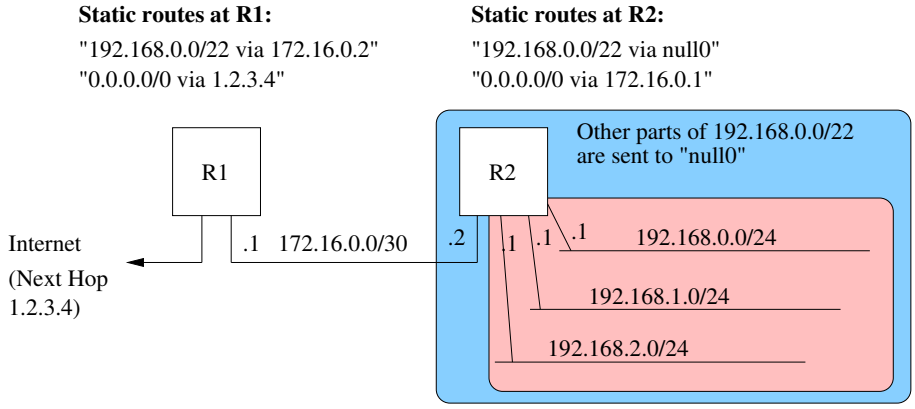


Figure 26.2: Use of blackhole route at router R2 to avoid a routing loop for addresses within range 192.168.2.0-192.168.255.255.

Use of blackhole routes is also useful when setting up SSL VPNs or IPsec VPNs. By use of blackhole routes, you can avoid that private traffic to the peer side is routed (unencrypted) towards the Internet when the VPN tunnel is down. See section 36.1.6 for an example of using blackhole routes with SSL VPNs.

26.1.5 Limitations When Using RSTP and Routing

As of WeOS v4.17.1 a single RSTP instance per WeOS unit is supported. This works fine in a switched environment where all VLANs on a switch can be added to inter-switch ports, see also chapters 13 (VLAN) and 16 (RSTP/STP).

However, when using RSTP in a routed environment it is often needed to run a separate instance of RSTP per VLAN. Otherwise there is a risk that RSTP incorrectly detects a loop (at layer-2) and blocks some port, even though there is a ”routing barrier”, which already handles the loop. The result of RSTP blocking ports may be loss of connectivity at layer-3.

RSTP is typically enabled on all ports by default. When using the WeOS device as a router, it is therefore recommended either to

- disable RSTP as a whole, or
- disable RSTP on all ports but one VLAN, or a group of VLANs with a shared layer-2 backbone (such as a ring).

Support for multiple RSTP/STP instances is planned but not yet implemented.

26.2 View Unicast Routing Table and Manage Static Unicast Routes via Web Interface

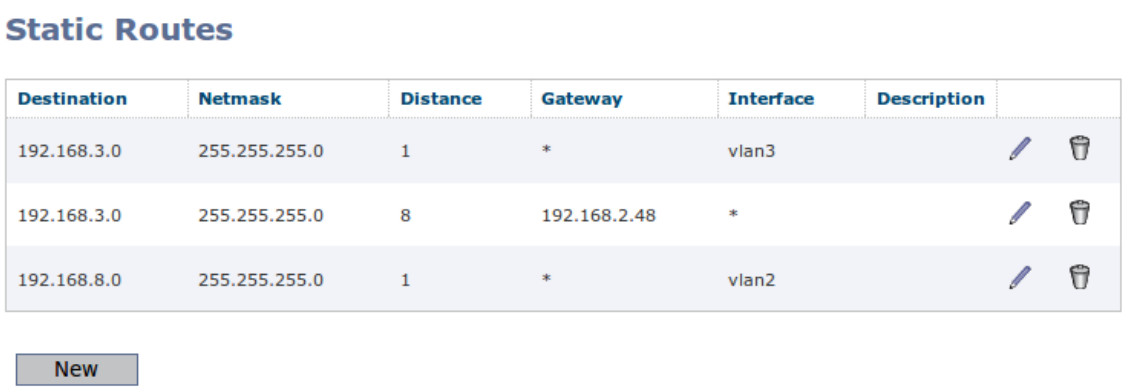
Web configuration of static *unicast* routes is presented in section 26.2.1, and examination of the current (unicast) routing table via Web is covered in section 26.2.2.

Web configuration of static *multicast* routes and examination of the *multicast* routing table is instead handled in chapter 29.

26.2.1 Managing Static Unicasts Routing via Web Interface

Menu path: Configuration ⇒ Routing ⇒ Static Route

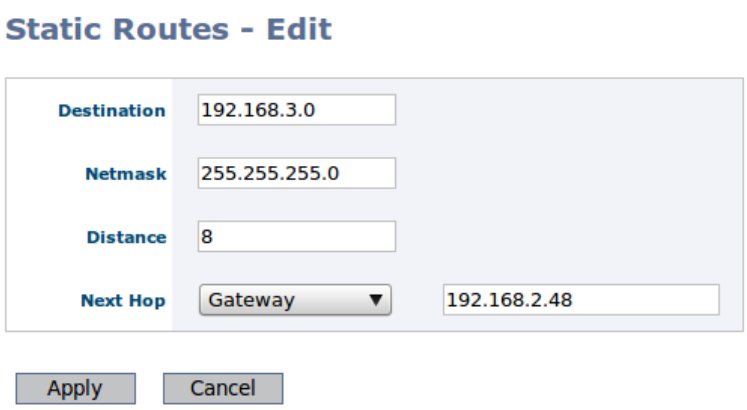
The main static routing configuration page lists the currently configured static routes.



Destination	The subnet to route
Netmask	The netmask defining the subnet
Distance	The administrative distance used when selecting between multiple routes to the same destination (floating static route).
Gateway	The destination gateway
Interface	The destination interface
Edit	Click this icon to edit a route.
Delete	Click this icon to remove a route. You will be asked to acknowledge the removal before it is actually executed.

Menu path: Configuration ⇒ Routing ⇒ Static Route ⇒ **Edit**

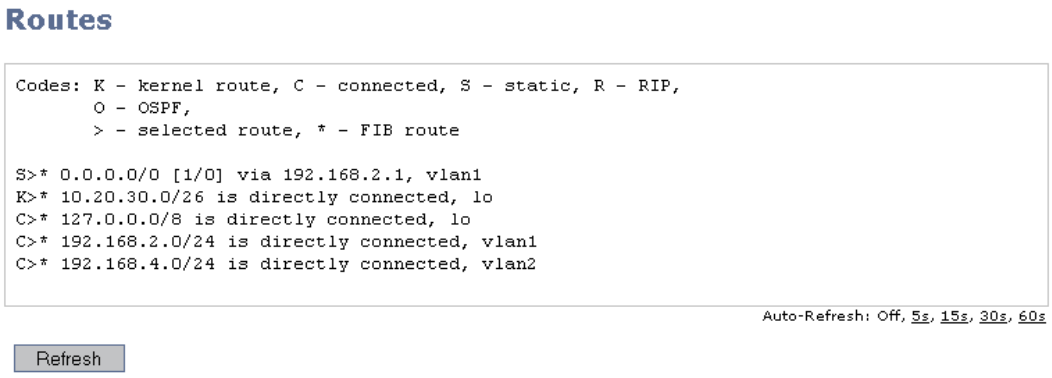
The edit page, see table above for descriptions.



26.2.2 Examine Routing Table via the Web Interface

Menu path: Status ⇒ Routing ⇒ Routes

On this page the current IP routes are listed.



One or more codes describe which source the route has, and if it is selected.

C	Connected - A network is known by a direct connection to the switch.
K	Kernel route
S	Static - A statically configured route.
R	RIP - The route is known through the RIP protocol.
O	OSPF - The route is known through the OSPF protocol.
>	Selected route
*	FIB route

26.3 Enabling Routing, Managing Static Routing, etc., via CLI

The table below shows WeOS CLI commands relevant for handling static routing. The detailed description of these commands is found in other chapters as listed in the table.

Dynamic routing (RIP and OSPF) and other router related protocols (VRRP) share a common *router* configuration context which is also listed in the table.

Command	Default	Section
Configure general routing settings		
ip		Section 19.7.1
[no] default-gateway <IPADDR>	DEPRECATED	Section 19.7.2
[no] route <NETWORK NETMASK| NETWORK/LEN> <GATEWAY|IFNAME> [DISTANCE]	DISTANCE 1	Section 19.7.3
[no] forwarding	Enabled	Section 19.7.4
router		Section 26.3.1
[no] ospf		Section 27.3
[no] rip		Section 28.3
[no] vrrp <ID>		Section 30.3
Show general routing status		
show ip route		Section 19.7.25

26.3.1 Manage Router Protocols

Syntax `router`

Context Global Configuration context

Usage Enter the Router Protocol Configuration context. From here you can configure dynamic routing protocols such as OSPF (section 27.3) and RIP (section 28.3) and, as well as other router related protocols such as VRRP (section 30.3).

Use **"show router"** to list general router protocol settings (also available **"show"** command within the Router Protocol Configuration context.

Default values N/A

Example

Example

```
example:/config/#> router
example:/config/router/#> show
OSPF/RIP not enabled.
VRRP Instances ================================================================
ID    Interface   Router-ID   Priority   Address
================================================================================
  1       vlan1           1        100   192.168.2.1

example:/config/router/#>
```

Chapter 27

Dynamic Routing with OSPF

This chapter describes WeOS support for the OSPF dynamic routing protocol.

27.1 Overview of OSPF features

Feature	Web	CLI	General Description
General OSPF settings			
Router-id	X	X	Section 27.1.1.1
OSPF Networks	X	X	Section 27.1.1.1
Area type (regular, stub, NSSA)	X	X	Sections 27.1.1.2, and 27.1.1.4-27.1.1.5
Redistribution (static, connected, RIP)	X	X	Section 27.1.1.3
Distribute default route	X	X	Section 27.1.1.3
Inter-area summarisation	X	X	Section 27.1.1.6
Inter-area filtering	X	X	Section 27.1.1.6
Passive interface default	X	X	Section 27.1.1.7
Per interface OSPF settings			
Link cost	X	X	Section 27.1.1
Passive interface	X	X	Section 27.1.1.7
Authentication (MD5, plain)	X	X	Section 27.1.1.8
Hello/Dead intervals	X	X	Section 27.1.1.9
Designated Router priority	X	X	Section 27.1.1.10

Note
As of WeOS v4.17.1 there is no support for ”load balancing” in case there are multiple paths with equal cost to reach a destination.
When an OSPF configuration change is done in WeOS, OSPF will be restarted on that router. Until the OSPF routing protocol has converged, this may cause a temporary loss of connectivity in parts of your network.

27.1.1 OSPF introduction

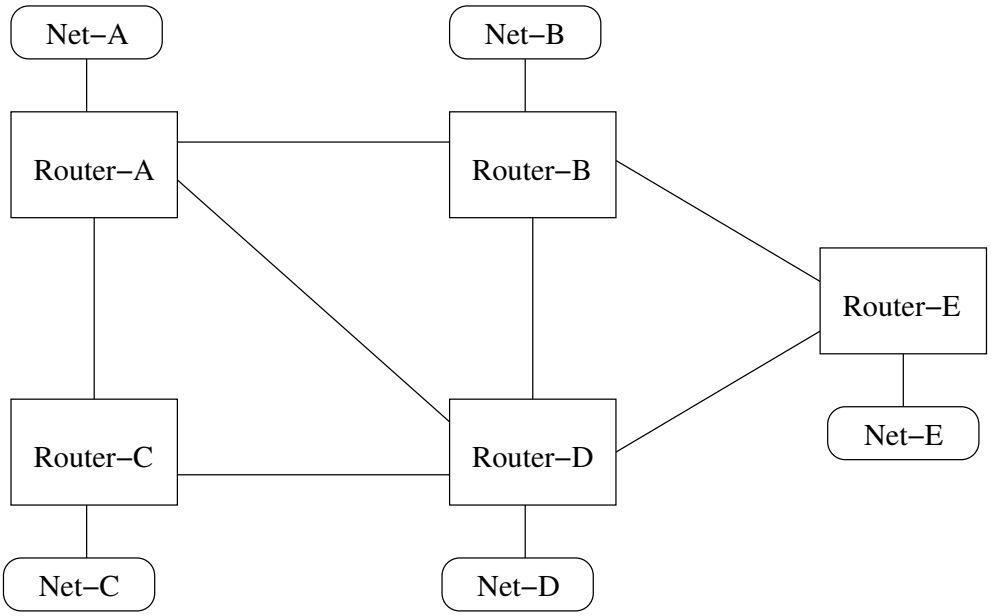


Figure 27.1: Simple network topology with interconnected routers and networks.

Dynamic routing protocols such as OSPF and RIP (chapter 28) simplifies router configuration, and improves network robustness.

- *Simplified configuration:* Manual configuration of static routes is not needed, and thereby a time consuming and error-prone procedure is avoided. In the network shown in fig. 27.1, each router would only have to be configured with information about its own identity and the IP subnets it is attached to. Routers will then exchange this information, and be able to establish the appropriate routing table by themselves.

- *Improved robustness:* If the topology changes, perhaps because a link failed, routers will automatically detect this and inform each other. The data traffic will be forwarded other ways, given that a redundant path to the destination exists.

OSPF is an example of a *link-state* routing protocol. In a link-state routing protocol, each router announces information about its own identity (*router-id*), its directly connected networks, and its neighbour routers. This information is *flooded* throughout the OSPF domain, and each router will store the information in a local OSPF database. Each router will gain complete knowledge about every router and link in the whole topology, and is therefore able to compute the best path (the least cost path) to reach every destination[1].

For example, Router-A in fig. 27.1 would send out OSPF messages informing other routers about its *router-id*, its connected networks, i.e., Net-A and the links towards routers A, B, and C, the identity of (and link to) to its neighbour routers (A, B and C).

A major advantage of link-state routing protocols, such as OSPF, over distance vector routing protocols, such as RIP, is the *fast convergence* after a topology change. If a link goes down, information about this can be flooded rapidly to all routers within the routing domain, and each router can then update their routing table accordingly.

27.1.1.1 OSPF Router-ID and OSPF Networks

We use the example below to explain some essential OSPF parameter settings (the example is for *Router-A* in fig. 27.2).

Example

```
router
        ospf
                router-id 10.0.11.1
                network 10.0.1.0/24 area 0.0.0.0
                network 10.0.2.0/24 area 0.0.0.0
                network 10.0.3.0/24 area 0.0.0.0
                network 10.0.11.0/24 area 0.0.0.0
                end
end
```

[1] In OSPF, a cost is associated with every link. As of WeOS v4.17.1, the default cost per link is ”10”. The link cost can be configured per interface, see section 27.3.16 for details.

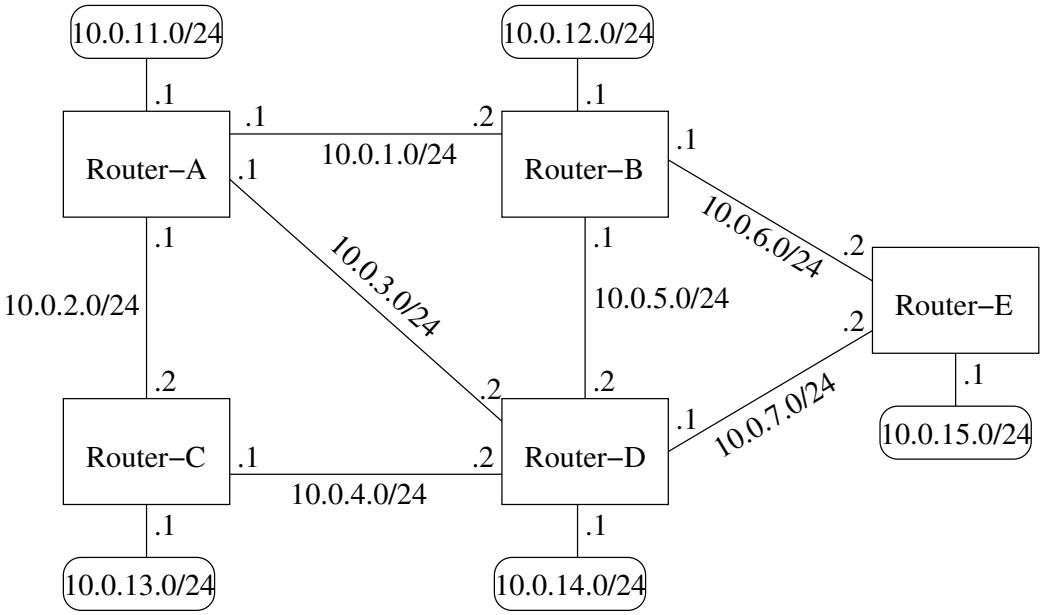


Figure 27.2: Example OSPF network with IP addresses and subnets.

The **”router-id”** line states the identity of this OSPF router, and must be unique within this OSPF routing domain.

- The router-id is 32-bit value, and can be specified either as a regular integer value, or in *dotted-decimal* form, just like an IP address.
- It is *common practise* to set the *router-id* to one of the IP addresses assigned to the router.
- If no router-id is configured, WeOS will pick one of the router’s configured IP addresses, and use that as router-id.

As mentioned in section 27.1.1, the router should inform the other routers about its attached links and networks. However, a router will announce its networks and links first when they are declared to be within the OSPF routing domain – this is done via the **”network”** command. Furthermore, a **”network”** declaration implies that OSPF messages will be exchanged through the corresponding network interface. (In some network setups one likes to include a subnet within the OSPF domain, without activating OSPF on the corresponding interface. This can be achieved by configured that interface as *passive*, see section 27.1.1.7.)

In the example above, Router-A has been configured to include and announce all

its subnets in the OSPF domain (10.0.1.0/24, 10.0.2.0/24, etc.). From the example we can also see that the **"network"** declaration contains an *area* parameter. OSPF areas are further explained in section 27.1.1.2.

27.1.1.2 OSPF hierarchy and areas

Being a link state protocol, OSPF requires routers to keep a lot of routing information in their database:

- Each OSPF router will typically keep a database with information of every router and link in the whole OSPF domain.
- OSPF routers will also redistribute and keep routing information learnt from external sources (static routes, routes learnt via other routing protocols, etc.).

To reduce the burden of keeping keeping state information about the whole OSPF domain, the domain can be split into OSPF *areas*. (For information on how to avoid the need to keep information on external routing information, see section 27.1.1.4.)

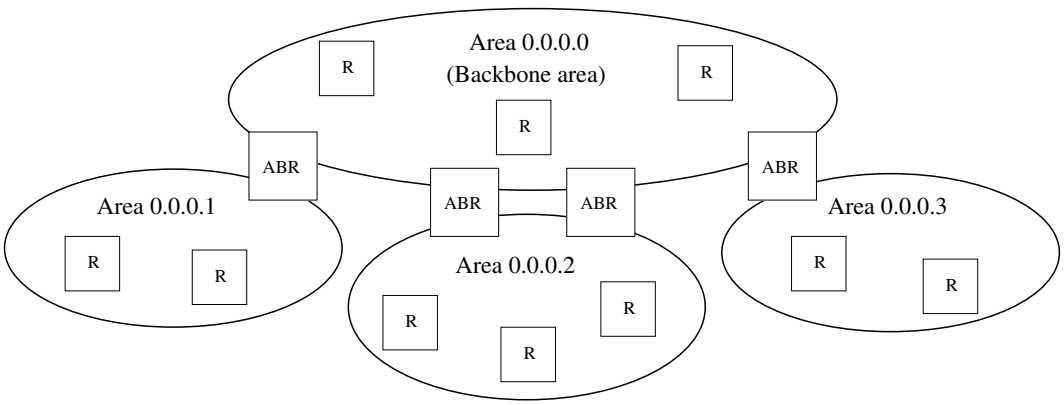


Figure 27.3: Sample OSPF hierarchy with a backbone area and three other areas.

The routers in fig. 27.3 have been divided into four areas. When splitting the network into multiple areas, each router will only have full knowledge of the topology within their respective area. Routers will also keep *summary* information about destinations outside their own area, but routers will not have knowledge about the actual topology inside other areas.

Each IP subnet can only be part of one OSPF area, and when configuring OSPF networks you should also define which area it belongs to. The area identifier is a 32 bit value, which can be stated as a decimal value, but is commonly written in *dotted decimal form*. E.g., **”network 10.0.1.0/24 area 0.0.0.0”** is equivalent to writing **”network 10.0.1.0/24 area 0”**.

A router which have networks in different areas is called an *area border router* (ABR). An example is given below.

Example

```
router
        ospf
                router-id 192.168.5.11
                network 192.168.5.0/24 area 0.0.0.0
                network 192.168.11.0/24 area 0.0.0.1
        end
```

In OSPF, areas are organised in a two-level hierarchy. At the top we have *area 0*, which is referred to as the *backbone area*. As the hierarchy is limited to two levels, every ABR must be connected to the backbone area. Direct connections between areas at lower level is prohibited; all inter-area traffic should go via the backbone area[2].

To allow for a more flexible area hierarchy, OSPF provides a feature referred to as *virtual links*, however, OSPF virtual links are not supported in WeOS v4.17.1.

27.1.1.3 Route redistribution and default route

Route information learnt from other routing protocols (RIP, BGP[3], etc.) *can* be redistributed (i.e., imported) into the OSPF domain. The same goes for static routes, and directly connected networks.

To let a router redistribute routing information into the OSPF domain, the **”redistribute”** command is used, e.g., **”redistribute rip”** to import routes learnt via RIP. An OSPF router performing route distribution into the OSPF domain is referred to as an administrative system border router (ASBR).

Routers can inject a default route (0.0.0.0/0) into the OSPF domain. This is done using the **”distribute-default [always]”** command. Without the **”always”** keyword, the router will only inject the default route if it itself has a default route.

[2] The reason for introducing these topology limitations is to avoid the ”counting to infinity” seen in *distance vector* protocols (see chapter 28) problem to occur for OSPF inter-area routing.)

[3] As of WeOS v4.17.1 BGP is not supported.

External routes can be added at two levels, *type 1* and *type 2* external routes:

- *Type 1:* Type 1 external routes are typically used when importing routes, that are locally managed, e.g., a static routes inside your domain, or from a local RIP domain.

  The ASBR located in area 0.0.0.2 in fig. 27.4 would preferably redistribute the routes learnt via RIP as *type 1* external routes.

- *Type 2:* Type 2 external routes are typically used when importing routes managed by another operator, e.g., routes learnt via BGP.

  The ASBRs located in area 0.0.0.0 in fig. 27.4 would preferably redistribute the routes learnt via BGP as *type 2* external routes.

27.1.1.4 Stub areas and totally stubby areas

In some situations one wish to limit the routing information going into an area to be limited even further, perhaps due to limited resources on the router. For this situation, OSPF provides a special area type referred to as a *stub area*.

As with other OSPF routers, routers inside a stub area will have full routing information for networks and routers within their own area and summary routes to destinations in other areas, *but* need not keep routing information learnt from *external* sources (static routes, or routes learnt via other routing protocols such as RIP, BGP, etc.). In a stub area, routing to networks outside the OSPF domain is instead based on *default routing* towards the ABR(s); i.e., the ABR will filter out all external routing information and instead inject a default route (pointing to itself) area.

To create a *stub* area, **all routers** in the area (ABRs as well as internal routers) must declare the area as stub. An example is given below.

Example

```
router
        ospf
                router-id 192.168.5.11
                network 192.168.5.0/24 area 0.0.0.0
                network 192.168.11.0/24 area 0.0.0.1
                area 0.0.0.1
                        stub
                end

        end
```

To reduce the routing information going into a stub area even further, it is possible to prohibit *summary* routes from other areas to go into a stub area. This is done by adding the *no-summary* parameter to the stub command (**”stub no-summary”**); this is only needed on the ABR(s) of the stub area.

Such areas are referred to as *totally stubby* areas.

The cost of the default route being injected into the stub area is by default set to ”1”. The cost value can be configured via the **”default-cost”** command within the area context.

The backbone area cannot be configured as a stub area.

27.1.1.5 Not so stubby areas (NSSAs)

In a stub area, no router can redistribute routing information learnt from external sources (static routes, BGP, etc.). That is, a stub area cannot contain an *autonomous system border router* (ASBR).

If you wish to have an ASBR in an area, but limit the amount of routing information to keep track of as in a stub area, OSPF provides an area type known as *not so stubby area* (NSSA).

Fig. 27.4 demonstrates a case where NSSAs can be a useful choice. Here we assume that area 0.0.0.1 and area 0.0.0.2 are preferably defined as *stub areas* to avoid that BGP routes (redistributed by the ASBRs in the backbone area) are propagated into those areas. But area 0.0.0.2 includes a router connected to a local RIP network. By defining area 0.0.0.2 as a NSSA, the RIP routes can be redistributed into the OSPF network.

NSSA are created in the same way as a *stub* area (see section 27.1.1.4). **All routers** in the area must declare the area as NSSA. An example is given below.

Example

```
router
ospf
router-id 192.168.5.12
network 192.168.5.0/24 area 0.0.0.0
network 192.168.16.0/24 area 0.0.0.2
                area 0.0.0.2
                        nssa
                end
end
```

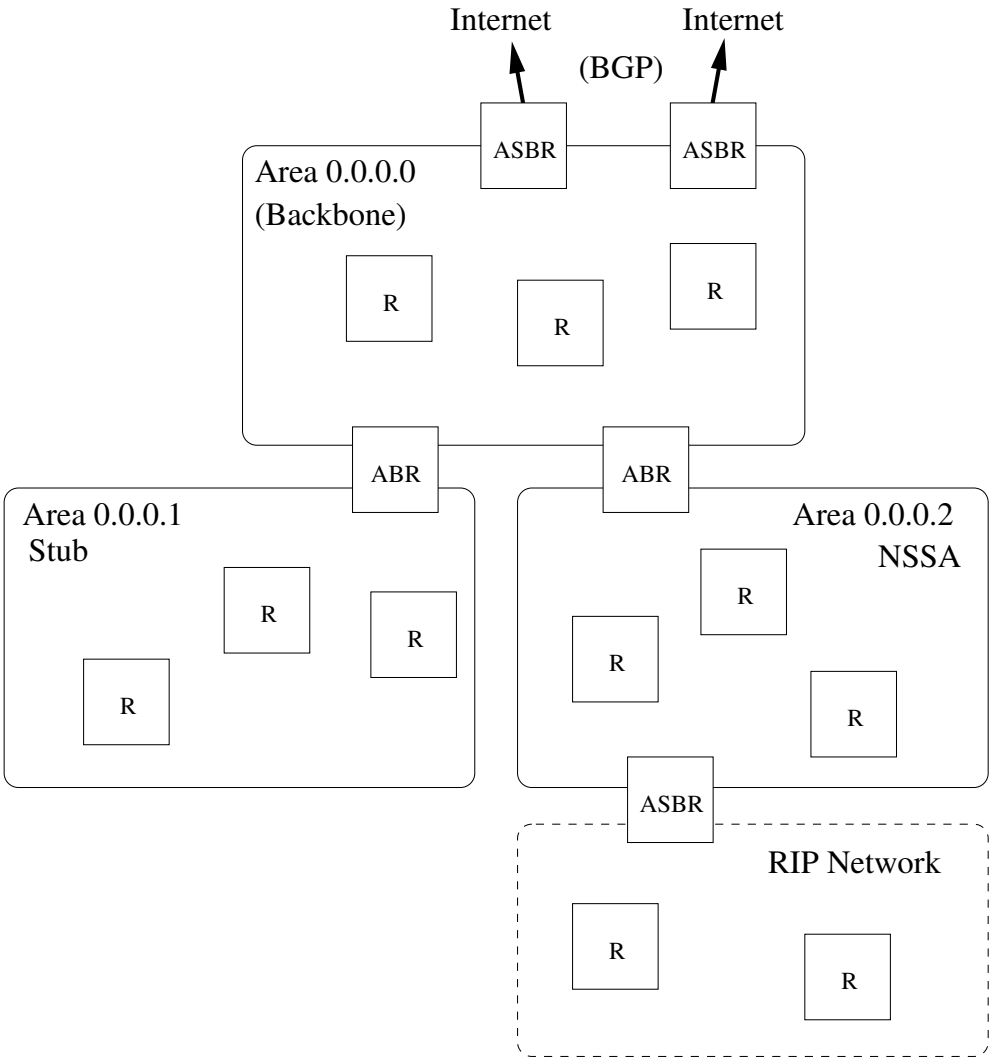


Figure 27.4: Topology where NSSA areas are useful.

As with stub areas, NSSAs are able to prohibit inter-area routing information to be distributed inside the area (use **”nssa no-summary”** on the ABRs of the area). Such areas are called *NSSA totally stub areas*.

The backbone area cannot be configured as a NSSA area.

27.1.1.6 Additional Area Specific Settings

ABRs are able to filter and to aggregate routing information before distributing it into another area. This is managed using the **"range <NETWORK/LEN> [not-advertise]"** command.

- *Route filtering:* With the **"not-advertise"** keyword, any route matching the given range will be filtered out when distributing routing information outside a certain area.
- *Route summarisation:* Without the **"not-advertise"** keyword, all routes matching the given range will be summarised (aggregated) as a single destination (of given network and prefix length) outside of a certain area.

Below is and example where an ABR will filter out routes in *192.168.16.0/20* when distributing routes from *area 0.0.0.2*. Similarly, all routes inside *area 0.0.0.2* matching *172.16.0.0/16* will be summarised to single route, when distributing routes from *area 0.0.0.2*.

Example

```
router
        ospf
                router-id 192.168.5.12
                network 192.168.5.0/24 area 0.0.0.0
                network 192.168.16.0/24 area 0.0.0.2
                network 192.168.19.0/24 area 0.0.0.2
                area 0.0.0.2
                        range 192.168.16.0/20 not-advertise
                        range 172.16.0.0/16
                end
end
```

27.1.1.7 Passive Interfaces

In some situations you may wish to include a router's subnets as part of the OSPF routing domain without running OSPF on the associated network interface. To accomplish this the *network* should be defined in the *router ospf* context (as usual), and the related interface should be declared as *passive* in the *interface ospf* context. Below is an example where network *192.168.33.0/24* should be included in the OSPF domain, but where the associated interface (*vlan100*) is declared as passive.

Example

```
[frame=single]
iface vlan100 inet static
        ...
        ... Skipping lines
        ...
        address 192.168.33.1/24
        ospf
                passive
                end
        end

router
        ospf
                router-id 192.168.15.1
                network 192.168.15.0/24 area 0.0.0.0
                network 192.168.33.0/24 area 0.0.0.0
                end
        end
```

By default, OSPF will run on all interfaces which have an associated network declared as an OSPF network. If OSPF should *not* run on such an interface, that interface should be declared as passive, as described above. However, WeOS is able to support use cases where the interfaces should be passive by default. The parameters controlling the behaviour are the **”passive-interface”** setting in *router ospf* context, and the **”passive”** setting in the *interface ospf* context.

- *passive-interface:* Use the **”[no] passive-interface”** setting in *router ospf* context to control whether interfaces should be passive in OSPF by default or not. Default setting: Active (**”no passive-interface”**)

- *passive:* Use the **”[no] passive [auto]”** setting in *interface ospf* context to control whether a specific interface should be passive (**”passive”**), active (**”no passive”**), or to automatically follow (**”passive auto”**) the global OSPF setting declared by the **”[no] passive-interface”** setting in *router ospf* context. Default: Auto (**”passive auto”**)

Below is an example, with the same result as above, where interfaces are passive in OSPF by default.

Example

```
iface vlan110 inet static
        ...
        ... Skipping lines
        ...
        address 192.168.15.1/24
        ospf
                no passive
                end
        end

router
        ospf
                router-id 192.168.15.1
                passive-interface
                network 192.168.15.0/24 area 0.0.0.0
                network 192.168.33.0/24 area 0.0.0.0
                end
        end
```

27.1.1.8 OSPF security

If an ”external” OSPF router happens to connect to your network (maliciously or by mistake) the routing inside your domain can be affected severely. E.g., if that router injects a default route into the OSPF domain, all traffic supposed to go to your Internet gateway may instead be routed towards this ”foreign” router.

To avoid that this happens, it is good practise to enable authentication of all OSPF messages inside your network. WeOS provides to forms of authentication of OSPF messages:

- *Plain:* Plain text authentication will protect against the situation when careless users attach an OSPF router to your network *by mistake*. However, since the password is sent in plain text inside the OSPF messages, it does not prohibit a deliberate attacker to inject routing information into your network. Plain text secrets are text strings of 4-8 characters.
- *MD5:* With MD5 authentication each OSPF message will include a cryptographic checksum, i.e., message authentication code (MAC), based on a secret only known by the system administrator. MD5 secrets are text strings of 4-16 characters.

Authentication of OSPF messages is configured per network interface, and is disabled by default.

Use of MD5 authentication is recommended. When using MD5 authentication,

an associated *key identifier* must be specified. The purpose of the *key identifier* is to enable use of multiple MD5 keys in parallel when performing *key roll-over*. However, as of WeOS version v4.17.1 only a single OSPF secret per interface is supported.

Warning

Configuring OSPF authentication remotely in an operational network can be dangerous, since the communication towards that router can be broken if the neighbour routers do not yet have the corresponding authentication configuration. In this case it is good practice to always have a redundant routing path to the router you are configuring.
If the you end up in the situation where you can no longer reach a router due to a change in OSPF authentication configuration, you may be able to solve the situation by first logging into a ”neighbour” of the ”unreachable router”, and from that router use SSH (see section 7.3.32) to login to the ”unreachable router”, and then update the configuration appropriately.

27.1.1.9 Finding OSPF Neighbours

OSPF routers will periodically transmit OSPF *Hello* messages, and routers can thereby discover new neighbour routers, and also detect if a neighbour router is down. There two parameter settings related to the OSPF hello messages. These settings are configured per interface.

- *Hello-interval:* The interval (in seconds) at which this router is transmitting Hello messages. Default: 10 seconds
- *Dead-interval:* The interval (in seconds) after which a neighbour router is considered down if no Hello message from that router is received[4]. Default: 40 seconds

Note

All routers attached to a link must have identical ”hello-interval” and ”dead-interval” settings. That is, an OSPF router will only accept incoming Hello messages with identical hello and dead interval values as the router itself is using on that interface.

[4] If the interface towards that neighbour goes down (e.g., if (all) the Ethernet port(s) associated with that interface goes down), the router will react immediately instead of waiting for the *dead-interval* to expire.

27.1.1.10 Designated OSPF router

In shared networks, such as Ethernets, there may be several routers attached to the same LAN. Representing a LAN as a full mesh of links between the attached routers may grow the OSPF database substantially if the number routers is large. Instead, link state protocols, such as OSPF, treats a shared link as a logical star, with a *virtual node* in the middle representing the shared network, see 27.5. The router which takes the role of network is referred to as the *designated router*.

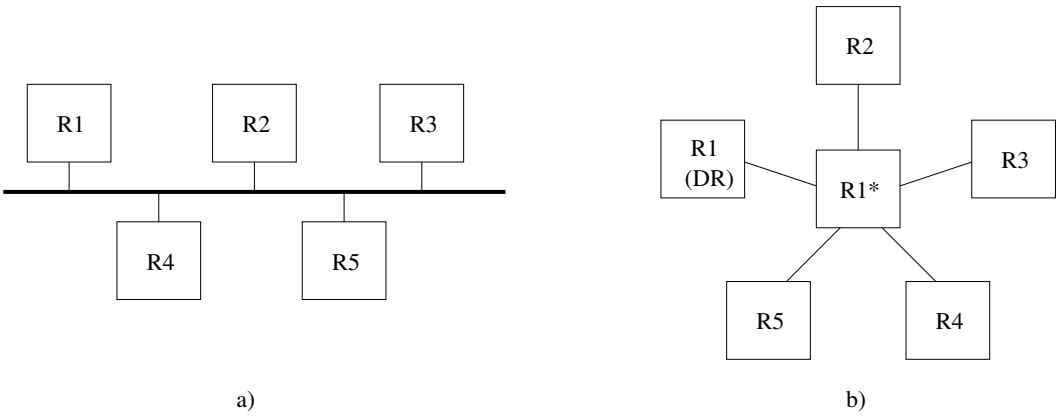


Figure 27.5: Link state protocols such as OSPF logically represent a shared link (a) as a star (b). One of the attached routers (here R1), will take the role as *designated router* and represent the ”network” in the middle.

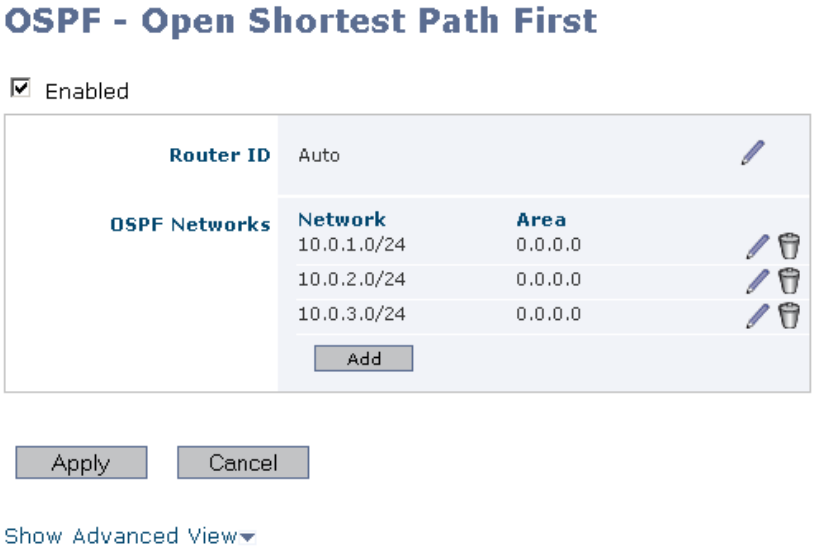
The designated router (DR), as well as a backup designated router (BDR), are elected automatically. If no node has been elected as DR or BDR, the router with the highest configured DR election *priority* becomes the DR, using the *router-id* as tie-breaker when more than one router has highest priority.

OSPF implements a *sticky* DR election scheme. Once a router has become DR, it will keep that role even when a router with higher DR priority comes up. However, a DR will give up its role if it discovers another router, which also consider itself to be DR, *and* if that router has higher priority (with router-id as tie). Such a situation could occur if a segmented LAN becomes connected.

27.2 OSPF Web

The Web interface provides configuration of OSPF.

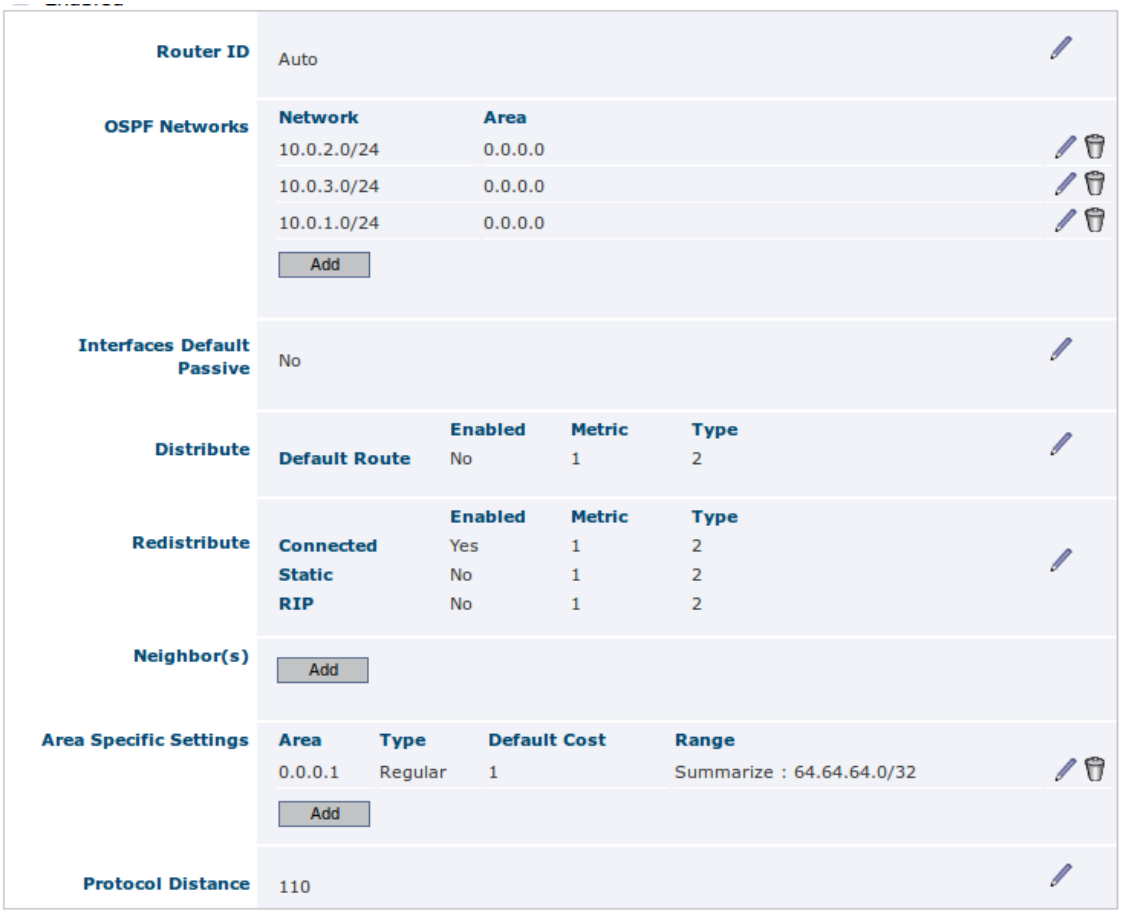
Menu path: Configuration ⇒ Routing ⇒ OSPF



When entering the OSPF configuration page the basic settings are presented.

Router ID	Click on the icon to set the OSPF router identifier. The router ID is given in a dotted decimal form <a.b.c.d> or as an integer
OSPF Networks	Enable OSPF on the router interface with the specified IP subnet (NETWORK/LEN). Click on the to edit settings or the icon to delete an entry. Press the Add button to add an entry.

To view all settings, click on **`Show Advanced View`** (see next page).



Interface Settings

Interface	Passive	Cost	Hello Interval	Dead Interval	Priority	Authentication
vlan1	Auto	10	10	40	1	None
vlan2	Auto	10	10	40	1	None

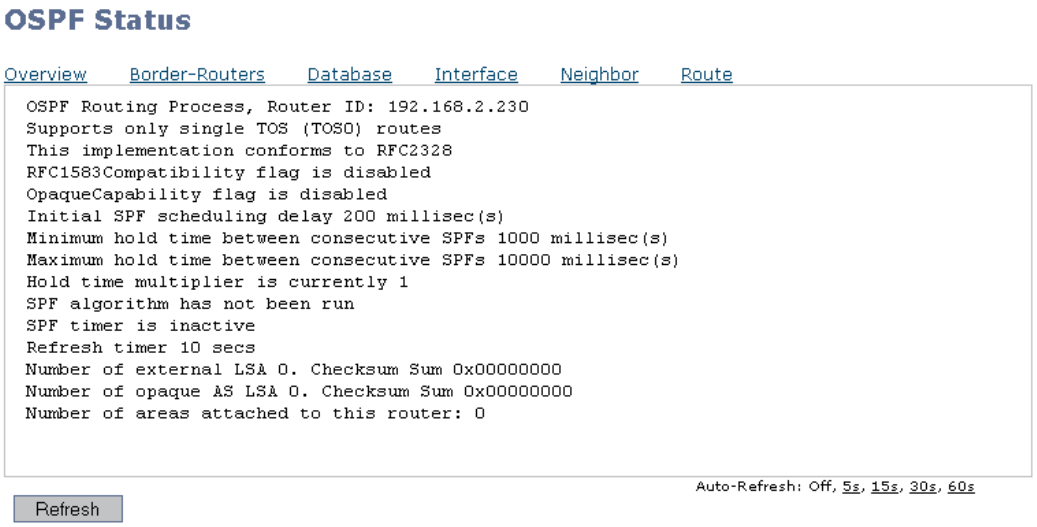
Router ID	Click on the icon to set the OSPF router identifier. The router ID is given in a dotted decimal form <a.b.c.d> or as an integer

Continued on next page

	Continued from previous page
OSPF Networks	Enable OSPF on the router interface with the specified IP subnet (NETWORK/LEN). Click on the  to edit settings or the  icon to delete an entry. Press the Add button to add an entry.
Interfaces Default Passive	Define whether OSPF should be run on the interfaces defined (implicitly) via the OSPF network settings.
Distribute Default Route	Enable/disabled injection of a default route into the OSPF domain
Redistribute	Enable/disabled import of external routing information into the OSPF domain
Neighbor(s)	Setup OSPF neighbor routers explicitly
Area Specific Settings	Add specific settings to an area. Click on the  to edit settings or the  icon to delete an entry. Press the Add button to add an entry.
Protocol Distance	The administrative distance used when selecting between multiple routes to the same destination.

27.2.1 OSPF Status Page

Menu path: Status ⇒ Routing ⇒ OSPF



Show the status of OSPF.

27.3 Managing OSPF via the CLI

The table below shows OSPF management features available via the CLI.

Command	Default	Section
Configure General OSPF Settings		
router		Sec. 26.3.1
[no] ospf	Disabled	Sec. 27.3.1
[no] router-id <ROUTERID>	Auto	Sec. 27.3.2
[no] network <NETWORK/LEN> [area <AREAID>]	area 0	Sec. 27.3.3
[no] neighbor <ADDRESSLIST>	Disabled	Sec. 27.3.4
[no] passive-interface	Active	Sec. 27.3.5
[no] distribute-default [always] [metric-type <1|2>] [metric <0-16777214>]	Disabled	Sec. 27.3.6
[no] redistribute connected [metric-type <1|2>] [metric <0-16777214>]	Disabled	Sec. 27.3.7
[no] redistribute static [metric-type <1|2>] [metric <0-16777214>]	Disabled	Sec. 27.3.7
[no] redistribute rip [metric-type <1|2>] [metric <0-16777214>]	Disabled	Sec. 27.3.7
[no] distance <1-255>	110	Sec. 27.3.8
[no] area <AREAID>		Sec. 27.3.9
[no] stub [no-summary]	Disabled	Sec. 27.3.10
[no] nssa [no-summary]	Disabled	Sec. 27.3.11
[no] default-cost <0-16777215>	0	Sec. 27.3.12
[no] range <NETWORK/LEN> [<advertise|not-advertise>]	advertise	Sec. 27.3.13
Configure Interface Specific OSPF Settings		
interface <IFACE>		
[no] ospf		Sec. 27.3.14
[no] passive [auto]	Auto	Sec. 27.3.15
[no] cost <1-65535>	10	Sec. 27.3.16

Continued on next page

Continued from previous page

Command	Default	Section
[no] hello-interval <1-65535>	10	Sec. 27.3.17
[no] dead-interval <1-65535>	40	Sec. 27.3.18
[no] auth <md5 [KEYID] | plain> <SECRET>	Disabled	Sec. 27.3.19
[no] priority <0-255>	1	Sec. 27.3.20
View OSPF Status		
show ip ospf		Sec. 27.3.21
show ip ospf route		Sec. 27.3.22
show ip ospf neighbor [<IFACE | detail>]		Sec. 27.3.23
show ip ospf database [asbr-summary|external| network|router|summary>		Sec. 27.3.24
show ip ospf database max-age		Sec. 27.3.24
show ip ospf database self-originate		Sec. 27.3.24

27.3.1 Activate OSPF and Manage General OSPF Settings

Syntax `[no] ospf`

Context Router Protocol Configuration context

Usage Enter the Router OSPF Configuration context, and *activate* OSPF with default settings if OSPF is not activated already. Instead of running **"ospf"** from the Router Protocol Configuration context, you can use **"router ospf"** directly from the Global Configuration

Use **"no ospf"** to disable OSPF and delete all existing OSPF configuration.

Use **"show ospf"** to show a summary of all general OSPF settings. Also available as **"show"** command within the Router OSPF Configuration context.

Default values Disabled (`no ospf`)

27.3.2 Configure OSPF Router-ID

Syntax `[no] router-id <ROUTER-ID>`

Context Router OSPF Configuration context

Usage Set the OSPF router identifier, which must be unique within your OSPF domain. The router ID is a 32-bit value, and is given in a dotted 1decimal form <a.b.c.d> (where a-d are numbers in the range 0-255), or as an integer ($0..2^{32}-1$). Commonly the router ID is set equal to one of the router's IP addresses.

In *Auto* mode, the router ID is *automatically* set to the IP address of one of the router's interface (the highest IP address), and stick to that value until the OSPF process is restarted.

Use **"show router-id"** to show the router-ID setting.

Default values Auto (`no router-id`)

27.3.3 Enable OSPF on an Interface

Syntax `[no] network <NETWORK/LEN> [area <AREAID>]`

Context Router OSPF Configuration context

Usage Enable OSPF on the router interface with the specified IP subnet (NETWORK/LEN), include that IP subnet in the OSPF routing domain, and determine the associated OSPF area.

The area ID is a 32-bit number, and is entered in dotted decimal form, or as an integer ($0..2^{32}-1$). By default, the backbone area (0.0.0.0) is assumed.

Use **"no network <NETWORK/LEN> [area <AREAID>]"** to delete a configured **"network"** entry.

Use **"show network"** to show the OSPF network settings.

Default values Disabled, i.e., no **"network"** entries exist when first activating OSPF (see section 27.3.2). The backbone area (0.0.0.0) is used as default area.

27.3.4 Configure Static Neighbour Router

Syntax `[no] neighbor <ADDRESSLIST>`

Context Router OSPF Configuration context

Usage Manually configure 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.

Use **"neighbor <IPADDR>"** to manually add one (or more) OSPF neighbour router(s). Use **"no neighbor"** to remove all manually configured neighbours, or **"no neighbor <IPADDDR>"** to remove a specific neighbour.

Use **"show neighbor"** to show manually configured OSPF neighbours.

27.3.5 Configure Interface Default Active/Passive Setting

Syntax `[no] passive-interface`

Context Router OSPF Configuration context

Usage Define whether OSPF should be run on the interfaces defined (implicitly) via the OSPF **"network"** command (see section 27.3.3).

If the setting is **"no passive-interface"**, the interfaces associated with the **"network"** command will automatically run OSPF, unless OSPF is explicitly disabled on the interface (see the **"passive"** command in section 27.3.15).

Similarly, if the setting is **"passive-interface"**, the interfaces associated with the **"network"** command will not run OSPF, unless OSPF is explicitly enabled on the interface (see the **"no passive"** command in section 27.3.15).

Use **"show passive-interface"** to show the default behaviour of OSPF interfaces (passive or active).

Default values Active (**"no passive-interface"**)

27.3.6 Configure Distribution of Default Route into OSPF Domain

Syntax `[no] distribute-default [always] [metric-type <1|2>] [metric <0-16777214>]`

Context Router OSPF Configuration context

Usage Inject a default route into the OSPF domain, i.e., announce that this router can reach *network 0.0.0.0/0*.

Use the **”always”** keyword to make the router always advertise the default route, regardless if it has one or not. Without the "always" keyword, it will only advertise if it has one.

Use **”show distribute-default”** to show the whether this router is configured to inject a default route into the OSPF domain.

Default values Disabled (**”no distribute-default”**)

27.3.7 Configure Redistribution of External Route Information into OSPF Domain

Syntax `[no] redistribute <connected|static|rip> [metric-type <1|2>] [metric <0-16777214>]`

Context Router OSPF Configuration context

Usage Import external routing information into the OSPF domain. Redistribution of connected routes, static routes, and routes learnt via RIP is handled independently, e.g., use **”redistribute rip”** to import routes learnt via RIP.

Use **”no redistribute”** to remove all redistribution, and **”no redistribute rip”** to remove redistribution of routes learnt via RIP, etc.

Use **”show how redistribute [<connected|static|rip>]”** to show the OSPF redistribution settings. Use **”show redistribute”** to show all redistribution settings, or **”show redistribute connected”**, etc., to show redistribute settings for specific types of redistribution.

Default values Disabled (**”no redistribute”**)

27.3.8 Configure Admin Distance for OSPF

Syntax `[no] distance <1-255>`

Context Router RIP Configuration context

Usage Configure admin distance for all routes learnt via OSPF. If the same route is learnt via different routing protocols (or as connected or static route), the route associated with the lowest admin distance will be used. For OSPF the admin distance defaults to 110. See also sections 19.2.6 and 26.1.3.

Use **”no distance”** to reset the OSPF admin distance to its default value.

Use **”show distance”** to show the configured OSPF admin distance value.

Default values 110

27.3.9 Manage area specific settings

Syntax `[no] area <AREAID>`

Context Router OSPF Configuration context

Usage Enter the OSPF Area Configuration context of the specified *AREAID* to configure area specific settings, such as area type (regular, stub, nssa), inter-area route summarisation, etc.

Use **”no area <AREAID>”** to remove specific for a single area, and **”no area”** to remove specific settings for all areas.

Use **”show area [<AREAID>]>]”** to show a summary of area specific settings. Use **”show area”** to show settings for all areas, and **”show area <AREAID>”** to show settings for a specific area. (Also available as **”show”** command within the OSPF Area Configuration context.)

Default values Disabled (**”no area”**)

27.3.10 Configure an Area as Stub

Syntax `[no] stub [no-summary]`

Context OSPF Area Configuration context

Usage 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.

To configure the area as a *totally stubby area*, all ABRs in the area should add the *no-summary* parameter to the stub command (**”stub no-summary”**).

Use **”no stub”** to let a stub (or nssa) area become a *regular* area.

Use **”show stub”** to show whether this area is configured as *stub* or not. If this is a stub area, it will show whether the **”no-summary”** keyword is set or not, i.e., if it is a *totally stubby* area or just a *stub* area.

Default values Disabled (i.e., areas are ”regular” OSPF areas by default)

27.3.11 Configure an Area as NSSA

Syntax `[no] nssa [no-summary]`

Context OSPF Area Configuration context

Usage Configure an area as a *nssa* area. To create a *nssa* area, **all routers** in the area (ABRs as well as internal routers) must declare the area as nssa.

To configure the area as a *NSSA totally stub area*, all ABRs in the area should add the *no-summary* parameter to the nssa command (**"nssa no-summary"**).

Use **"no nssa"** to let a nssa (or stub) area become a *regular* area.

Use **"show nssa"** to show whether this area is configured as *NSSA* or not. If this is a NSSA area, it will show whether the **"no-summary"** keyword is set or not, i.e., if it is a *NSSA totally stub* area or just a *NSSA* area.

Default values Disabled (i.e., areas are "regular" OSPF areas by default)

27.3.12 Configure default route cost in stub and NSSA areas

Syntax `[no] default-cost`

Context OSPF Area Configuration context

Usage Configure the cost of the default route injected into a stub area. This setting only applies to the ABRs of a stub or NSSA area.

Use **"no default-cost"** to use the *default* value for the *default cost* setting.

Use **"show default-cost"** to show the setting of the default-cost, i.e., the cost of the default route injected by ABRs into a stub or NSSA area.

Default values **"default-cost 0"**

27.3.13 Configure inter-area route summarisation and filtering

Syntax `[no] range <NETWORK/LEN> [<advertise|not-advertise]`

Context OSPF Area Configuration context

Usage Configure inter-area route *summarisation* or route *filtering*.

Use the **”range <NETWORK/LEN>”** (**”range <NETWORK/LEN> advertise”** is equivalent) to aggregate routes (within this area) matching the specified <NETWORK/LEN> range, before distributing the routes outside this area. That is, all routes within this range are *summarised* as a single route, when advertised outside this area.

Use the **”range <NETWORK/LEN> not-advertise”** to prohibit routes (within this area) matching the specified <NETWORK/LEN> range, to be distributed outside this area. That is, routes within this range are *filtered*.

Use **”no range <NETWORK/LEN>”** to remove a specific summary/filter setting, or **”no range”** to remove all summary/filter settings for this area.

Use **”show default-cost”** to show configured route summarisation and route filtering settings for this area.

Default values Disabled

27.3.14 Manage Interface Specific OSPF Settings

Syntax `[no] ospf`

Context Interface Configuration context

Usage Enter the Interface OSPF Configuration context, i.e., the context where Interface specific OSPF settings are configured.

Use **”no ospf”** to remove any specific OSPF settings for this interface.

Use **”show ospf”** to show a summary of OSPF settings for this interface. (Also available as **”show”** command within the Interface OSPF Configuration context.)

Default values Disabled (i.e., no interface specific OSPF settings)

27.3.15 Configure Interface OSPF Passive Settings

Syntax `[no] passive [auto]`

Context Interface OSPF Configuration context

Usage Control whether a specific interface should be passive (**”passive”**), active (**”no passive”**), or to automatically follow (**”passive auto”**) the global

OSPF setting declared by the **”[no] passive-interface”** setting in *router ospf* context (see section 27.3.5).

Use **”show passive”** to show the OSPF passive interface setting (passive, active or ”auto”) for this interface.

Default values Auto (**”passive auto”**)

27.3.16 Configure Interface OSPF Cost Settings

Syntax `[no] cost <1-65535>`

Context Interface OSPF Configuration context

Usage Configure interface OSPF cost.

Use **”no cost”** to return to the default setting.

> **Note**
> As of WeOS v4.17.1 only static configuration of the interface OSPF cost setting is available. Support to let the cost automatically depend on the interface data rate is planned, but not yet implemented.

Use **”show cost”** to show the OSPF cost setting for this interface.

Default values 10 (this may be subject to change in later versions of WeOS.

27.3.17 Configure Interface OSPF Hello Interval Settings

Syntax `[no] hello-interval <1-65535>`

Context Interface OSPF Configuration context

Usage Configure OSPF hello interval (in seconds) for this interface.

Use **”no hello-interval”** to return to the default setting.

> **Note**
> The hello interval setting must be the same on neighbour routers.

Use **”show hello-interval”** to show the OSPF hello interval setting for this interface.

Default values 10 (seconds)

27.3.18 Configure Interface OSPF Dead Interval Settings

Syntax `[no] dead-interval <1-65535>`

Context Interface OSPF Configuration context

Usage Configure OSPF dead interval (in seconds) for this interface.

Use **"no dead-interval"** to return to the default setting.

> **Note**
> The dead interval setting must be the same on neighbour routers.

Use **"show dead-interval"** to sow the OSPF dead interval setting for this interface.

Default values 40 (seconds)

27.3.19 Configure Authentication of OSPF Messages

Syntax `[no] auth <md5 [KEYID] | plain> <SECRET>`

Context Interface OSPF Configuration context

Usage Configure authentication of OSPF messages *on this interface*. Two authentication methods are available:

- *MD5:* Use **"auth md5 <KEYID> <SECRET>"** to use a MD5 cryptographic authentication. MD5 secrets are text strings of 8-16 characters. A key identifier (0-255) is associated with MD5 keys. (Both the secret and the key identifier must be the same on neighbour routers.)
- *Plain:* Use **"auth plain <SECRET>"** to use a clear-text password as authentication. Plain text secrets are text strings of 4-8 characters. (The secret must be the same on neighbour routers.)

Use **"no auth"** to disable authentication of OSPF messages on this interface.

Use **"show auth"** to show the OSPF authentication setting for this interface.

Default values Disabled

27.3.20 Configure OSPF Designated Router Priority

Syntax `[no] priority <0-255>`

Context Interface OSPF Configuration context

Usage Configure the OSPF designated router priority, which affects the chance to become designated router on a broadcast network. A higher value increases the chance to become designated router.

Use **”priority 0”** to state that this router is not eligible as designated router on this interface/”IP subnet”.

Use **”no priority”** to return to the default setting.

Use **”show priority”** to show the OSPF designated router election priority setting for this interface.

Default values 1 (**”priority 1”**)

27.3.21 Show General OSPF Status

Syntax `show ip ospf`

Context Admin Exec context.

Usage Show general OSPF status information.

Default values Not applicable

27.3.22 Show OSPF Routes

Syntax `show ip ospf route`

Context Admin Exec context.

Usage Show the current least-cost routes learnt via OSPF. See also the command **”show ip route”** (section 19.7.25), which displays the full forwarding/routing table.

Default values Not applicable

27.3.23 Show OSPF Neighbours

Syntax `show ip ospf neighbor [<IFACE | detail>]`

Context Admin Exec context.

Usage Show current list of OSPF neighbours. Use **"show ip ospf neighbor IFACE"** to list OSPF neighbours for a specific interface, or the keyword **"detail"** to receive a more detailed listing.

Default values By default, neighbours on all interfaces are listed.

27.3.24 Show OSPF Database

Syntax

```
show ip ospf database [asbr-summary|external|network|router|summary>],
show ip ospf database max-age,
show ip ospf database self-originate
```

Context Admin Exec context.

Usage Use **"show ip ospf database"** to list the current OSPF database. Various keywords can be added to view specific parts of the database.

Default values By default, the full database is listed.

Chapter 28

Dynamic Routing with RIP

This chapter describes WeOS support for the Routing Information Protocol (RIP.)

WeOS supports dynamic routing via RIP version 1 (RIPv1) and version 2 (RIPv2). RIP is relatively simple to setup, but does not handle topology changes as rapidly as the OSPF dynamic routing protocol (support for OSPF is described in chapter 27). Therefore, OSPF is generally preferred over RIP when it is possible to select dynamic routing protocol.

28.1 Overview of RIP Features

Table 28.1 summarises RIP support in WeOS.

28.1.1 Introduction to RIP

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 quite simple to configure; commonly you only have to enable RIP and define which interfaces to run RIP on. The router will automatically discover its neighbours and start to exchange routing information.

To enable RIP on all interfaces on R1 in fig. 28.1, configuration shown below would suffice.

Feature	Web	CLI	General Description
General RIP settings			
RIP version	X	X	Section 28.1.1
RIP Timers	X	X	
Passive Interface Default	X	X	Section 28.1.4
RIP Networks/Interfaces	X	X	Section 28.1.1
RIP Neighbour	X	X	-”-
Redistribution (static, connected, OSPF)	X	X	Section 28.1.2
Distribute Default Route	X	X	-”-
RIP Admin Distance	X	X	
Authentication (MD5, plain)	X	X	Section 28.1.3
Passive interface	X	X	Section 28.1.4
Split Horizon	X	X	
Send RIP version	X	X	
Receive RIP version	X	X	

Table 28.1: Summary of RIP features.

Example

```
router
        rip
                network 10.0.1.0/24
                network 10.0.2.0/24
                network 10.0.3.0/24
                end
        end
```

The command **”network 10.0.1.0/24”** will enable RIP on all interfaces included within the given range; in this example it states that RIP should be activated on the ”upper interface” (i.e., the interface with address 10.0.1.3/24). It is also possible to specify the interfaces explicitly; assuming the three interfaces of R1 are called *vlan1*, *vlan2*, and *vlan3*, the following configuration would give the same result:

Example

```
router
        rip
                network vlan1
                network vlan2
                network vlan3
                end
        end
```

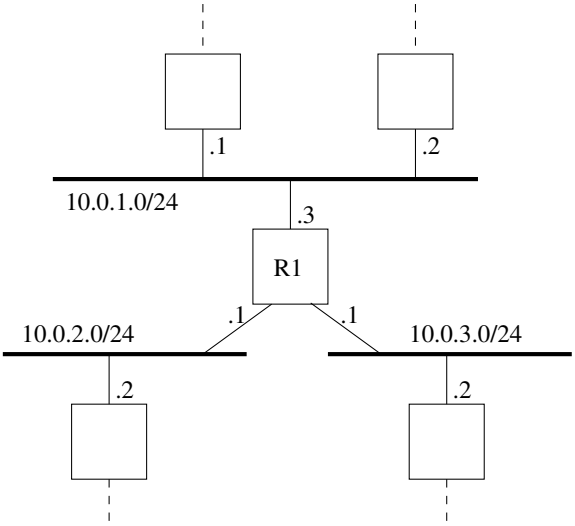


Figure 28.1: A router (R1) connected to other routers via three interfaces.

Both RIPv1[10] and RIPv2[21] are supported, and RIPv2 is used by default when RIP is enabled. The major difference between RIPv1 and RIPv2 is that RIPv2 supports flexible subnet masks (CIDR - classless inter-domain routing), while RIPv1 assumes that IP subnet masks follow the (deprecated) classful addressing scheme (class A, B and C). In addition, RIPv2 supports message authentication (section 28.1.3), and can therefore offer protection in situations when ”foreign RIP routers” are connected (by mistake or as a deliberate attack) to a network and inject RIP routing messages. Thus, use of RIPv2 is preferred over RIPv1, except for cases where legacy equipment require the use of RIPv1.

RIPv2 routers exchange routing information using IP multicast (IP address 224.0.0.9)[1]. In case a neighbour router is unable to handle IP multicast, the **”neighbor”** command enables the exchange of RIP messages using regular IP unicast.

28.1.2 Redistribution and Injection of Default Route

It is possible to redistribute routing information learnt externally (OSPF, connected routes or static routes) inside the RIP routing domain, using the **”redistribute”** command.

[1]While RIPv2 use IP multicast, RIPv1 exchange routing information using broadcast.

You can also let a RIP router inject a default route (0.0.0.0/0) into your RIP domain, using the **”distribute-default”**.

28.1.3 Authentication

To avoid that false routing information is injected into your network (deliberately or by mistake) it is possible to authenticate RIPv2 messages. Two authentication alternatives are available:

- *Plain:* Plain text authentication will protect against the situation when careless users attach a RIP router to your network *by mistake*. However, since the password is sent in plain text inside the RIP messages, it does not prohibit a deliberate attacker to inject routing information into your network. Plain text secrets are text strings of 4-16 characters.
- *MD5:* With MD5 authentication each RIP message will include a cryptographic checksum, i.e., message authentication code (MAC), based on a secret only known by the system administrator. MD5 secrets are text strings of 4-32 characters.

Authentication of RIP messages is configured per network interface, and is disabled by default.

Use of MD5 authentication is recommended. When using MD5 authentication, an associated *key identifier* must be specified. The purpose of the *key identifier* is to enable use of multiple MD5 keys in parallel when performing *key roll-over*. However, as of WeOS version v4.17.1 only a single RIP secret per interface is supported.

28.1.4 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. To accomplish this the *network* should be defined in the *router rip* context (as usual), and the related interface should be declared as *passive* in the *interface rip* context. Below is an example where network *10.0.3.0/24* should be included in the RIP domain, but where the associated interface (*vlan3*) is declared as passive.

Example

```
iface vlan3 inet static
        ...
        ... Skipping lines
        ...
        address 10.0.3.1/24
        rip
                passive
                end
        end

router
        rip
                network 10.0.1.0/24
                network 10.0.2.0/24
                network 10.0.3.0/24
                end
        end
```

By default, RIP will run on all interfaces which have an associated network declared as a RIP network. If RIP should *not* run on such an interface, that interface should be declared as passive, as described above. However, WeOS is able to support use cases where the interfaces should be passive by default. The parameters controlling the behaviour are the **”passive-interface”** setting in *router rip* context, and the **”passive”** setting in the *interface rip* context.

- *passive-interface:* Use the **”[no] passive-interface”** setting in *router rip* context to control whether interfaces should be passive in RIP by default or not. Default setting: Active (**”no passive-interface”**)

- *passive:* Use the **”[no] passive [auto]”** setting in *interface rip* context to control whether a specific interface should be passive (**”passive”**), active (**”no passive”**), or to automatically follow (**”passive auto”**) the global RIP setting declared by the **”[no] passive-interface”** setting in *router rip* context. Default: Auto (**”passive auto”**)

Below is an example, with the same result as above, where interfaces are passive in RIP by default.

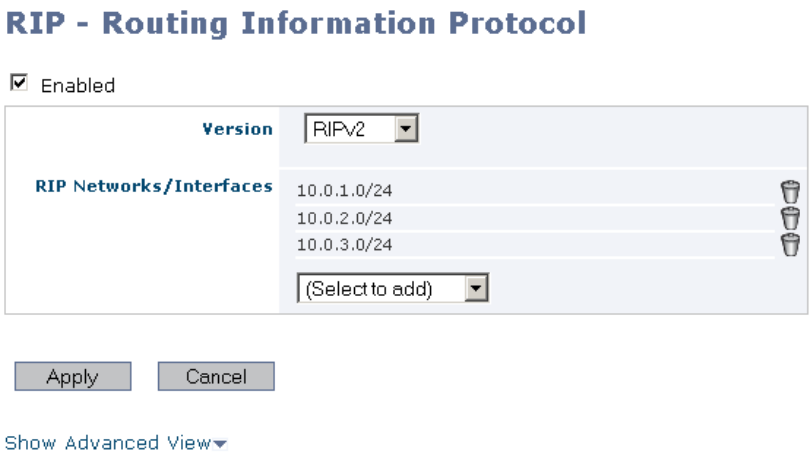
Example

```
iface vlan1 inet static
        ...
        ... Skipping lines
        ...
        address 10.0.1.3/24
        rip
                no passive
                end
        end

iface vlan2 inet static
        ...
        ... Skipping lines
        ...
        address 10.0.2.1/24
        rip
                no passive
                end
        end

router
        rip
                passive-interface
                network 10.0.1.0/24
                network 10.0.2.0/24
                network 10.0.3.0/24
                end
        end
```

28.2 RIP Web

The Web interface provides configuration of RIP.

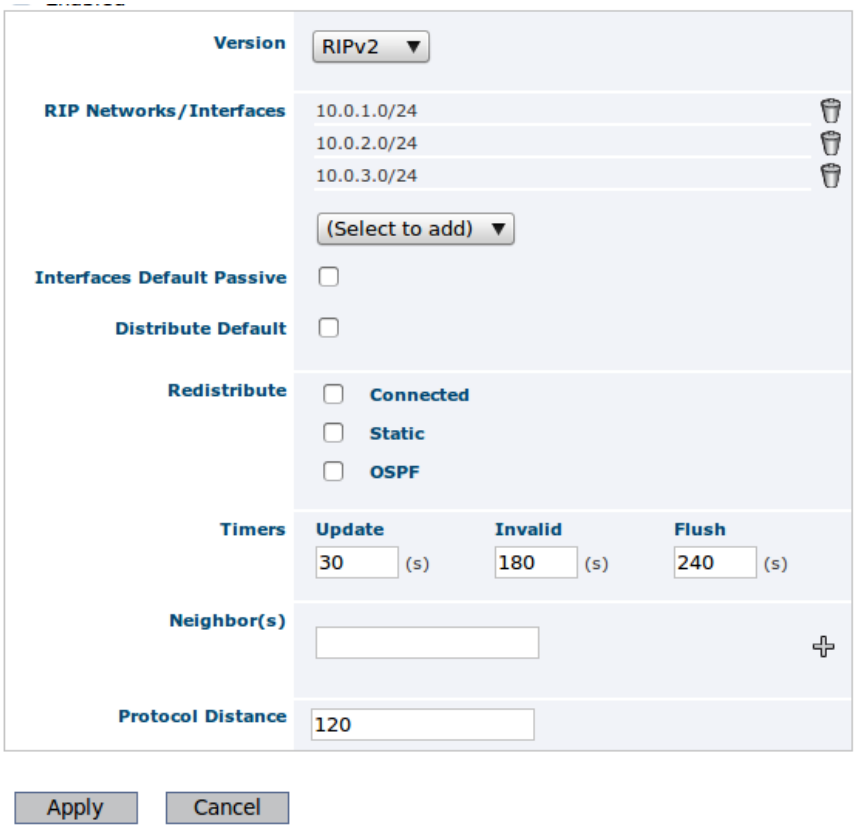
Menu path: Configuration ⇒ Routing ⇒ RIP



When entering the RIP configuration page the basic settings are presented.

Version	Select what RIP version (1 or 2) to use by default
RIP Networks/Interfaces	Enable RIP on the specified router Network/Interface
	Click this icon to delete a RIP Network or RIP Interface.

To view all settings, click on **`Show Advanced View`** (see next page).



Interface Settings

Interface	Passive	Split Horizon	Send Version	Receive Version	Authentication	
vlan1	Auto	Enabled	Auto	Auto	None	
vlan2	Auto	Enabled	Auto	Auto	None	

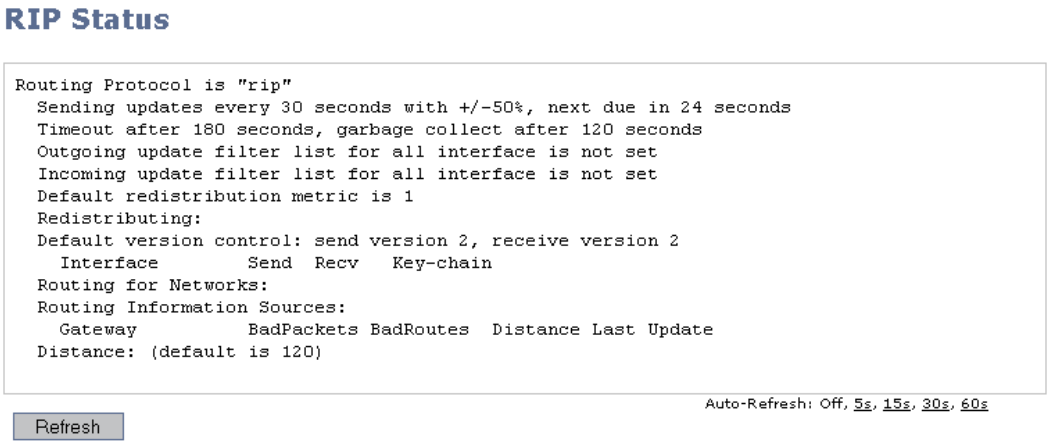
Version	Select what RIP version (1 or 2) to use by default
RIP Networks/Interfaces	Enable RIP on the specified router Network/Interface
Interfaces Default Passive	Define whether RIP should be run on the interfaces defined (implicitly) via the RIP

Continued on next page

	Continued from previous page
Distribute Default	Enable/disabled injection of a default route into the RIP domain
Redistribute	Enable/disabled import of external routing information into the RIP domain
Timers	Setup timers of the RIP protocol
Neighbor(s)	Setup RIP neighbor routers explicitly
	Click this icon to delete a RIP Network or RIP Interface.
Protocol Distance	The administrative distance used when selecting between multiple routes to the same destination.

28.2.1 Rip Status Page

Menu path: Status ⇒ Routing ⇒ RIP



Show the status of RIP.

28.3 Managing RIP via the CLI

The table below shows RIP management features available via the CLI.

Command	Default	Section
Configure General RIP Settings		
router		Sec. 26.3.1
[no] rip	Disabled	Sec. 28.3.1
[no] version <1|2>	version 2	Sec. 28.3.2
[no] timers [update <SEC>]	update 30	Sec. 28.3.3
[invalid <SEC>]	invalid 180	
[flush <SEC>]	flush 240	
[no] network <NETWORK | IFACE>		Sec. 28.3.4
[no] neighbor <ADDRESSLIST>		Sec. 28.3.5
[no] passive-interface	Active	Sec. 28.3.6
[no] distribute-default	Disabled	Sec. 28.3.7
[no] redistribute connected	Disabled	Sec. 28.3.8
[no] redistribute static	Disabled	Sec. 28.3.8
[no] redistribute ospf	Disabled	Sec. 28.3.8
[no] distance <1-255>	120	Sec. 28.3.9
Configure Interface Specific RIP Settings		
interface <IFACE>		Sec. 19.6.1
[no] rip		Sec. 28.3.10
[no] passive [auto]	Auto	Sec. 28.3.11
[no] split-horizon [poisoned-reverse]	Enabled	Sec. 28.3.12
[no] send-version <1,2>	Auto	Sec. 28.3.13
[no] receive-version <1,2>	Auto	Sec. 28.3.14
[no] auth <md5 [keyid] | plain> <SECRET>	Disabled	Sec. 28.3.15
View RIP Status		
show ip rip		Sec. 28.3.16

28.3.1 Activate RIP and Manage General RIP Settings

Syntax `[no] rip`

Context Router Protocol Configuration context

Usage Enter the Router RIP Configuration context, and *activate* RIP with default settings if RIP is not activated already. Instead of running **"rip"** from the *Router* context, you can use **"router rip"** directly from the Global Configuration

Use **"no rip"** to disable RIP and delete all existing RIP configuration.

Use **"show rip"** to show a summary of all general RIP settings. Also available as **"show"** command within the Router RIP Configurationcontext.

Default values Disabled (`no rip`)

28.3.2 Configure Default RIP Version

Syntax `[no] version <1|2>`

Context Router RIP Configuration context

Usage Select what RIP version (1 or 2) to use by default, both with respect to sending and receiving of RIP messages. The setting can be overridden per interface using the **"receive-version"** (section 28.3.14) and **"send-version"** (section 28.3.14) respectively.

Use **"no version"** to return to the default setting.

Use **"show version"** to show the default RIP version setting.

Default values RIPv2 (`version 2`)

28.3.3 Configure RIP Protocol Timers

Syntax `[no] timers [update <SEC>] [invalid <SEC>] [flush <SEC>]`

Context Router RIP Configuration context

Usage Several timers of the RIP protocol can be changed using the *timers* command. All timers take a value between <5-2147483647> seconds.

- The *update* timer controls the interval between sending unsolicited *Response Messages* to all neighboring routers.
- The *invalid* timer controls the time before a route is expired and removed from the kernel routing table. It is kept for $flush - invalid$ seconds in the internal RIP routing table to notify neighbors that a route has been dropped.
- The *flush* timer should be longer than the *invalid* timer. It controls the time when a route is finally cleared from the routing table.

Important
All routers should have the same timings setup.

Use **"show timers"** to show the configured RIP protocol timers.

Default values Use **"no timers"** to return to the default timers:

update 30 sec

invalid 180 sec

flush 240 sec

Example

```
timers update 5 invalid 15 flush 30
```

This sends out updates every five seconds, invalidates a route if a router is not heard from in 15 seconds and flushes the route after an additional 15 seconds.

28.3.4 Enable RIP on an Interface

Syntax `[no] network <NETWORK/LEN | IFACE>`

Context Router RIP Configuration context

Usage Enable RIP on the specified router interface. The interface can be specified either explicitly (**"network <IFACE>"**) or implicitly giving the IP subnet associated with the interface (**"network <NETWORK/LEN>"**).

Use **"no network <IFACE>"** and **"no network <NETWORK/LEN>"** to remove an existing **"network"** entry.

Use **”show network”** to show the RIP network settings, i.e., which interfaces/subnets that are included in the RIP routing domain.

Default values Disabled, i.e., when first activating RIP (section 28.3.1), RIP will not be enabled on any interface.

28.3.5 Configure Unicast Neighbor

Syntax `[no] neighbor <ADDRESSLIST>`

Context Router RIP Configuration context

Usage Configure one or more RIP neighbor routers explicitly. This is useful in case the neighbor router is unable to handle IP multicast. An **”ADDRESSLIST”** is a comma-separated list of IPv4 address, e.g, **”neighbor 192.168.1.1,192.168.3.2”**. Calling the **”neighbor”** command twice (with arguments ”192.168.1.1” and ”192.168.3.2” respectively) would be equivalent.

Use **”no neighbor”** to remove all configured neighbours, and **”no neighbor <ADDRESSLIST>”** to remove a specific neighbour settings.

Use **”show neighbor”** to show the configured RIP Unicast Neighbours.

Default values Disabled (No neighbours defined)

28.3.6 Configure Interface Default Active/Passive Setting

Syntax `[no] passive-interface`

Context Router RIP Configuration context

Usage Define whether RIP should be run on the interfaces defined (implicitly) via the RIP **”network”** command (see section 28.3.4).

If the setting is **”no passive-interface”**, the interfaces associated with the **”network”** command will automatically run RIP, unless RIP is explicitly disabled on the interface (see the **”passive”** command in section 28.3.11).

Similarly, if the setting is **”passive-interface”**, the interfaces associated with the **”network”** command will not run RIP, unless RIP is explicitly enabled on the interface (see the **”no passive”** command in section 28.3.11).

Use **”show passive-interface”** to show the default behaviour of RIP interfaces (passive or active).

Default values Active (**”no passive-interface”**)

28.3.7 Configure Distribution of Default Route into RIP Domain

Syntax `[no] distribute-default`

Context Router RIP Configuration context

Usage Inject a default route into the RIP domain, i.e., announce that this router can reach *network 0.0.0.0/0*.

Use **”[no distribute-default”** to stop this router from injecting a default route into the RIP domain.

Use **”show distribute-default”** to show the RIP redistribution settings. Use **”show redistribute”** to show all redistribution settings, or **”show redistribute connected”**, etc., to show redistribute settings for specific types of redistribution.

Default values Disabled (**”no distribute-default”**)

28.3.8 Configure Redistribution of External Route Information into RIP Domain

Syntax `[no] redistribute <connected|static|ospf>`

Context Router RIP Configuration context

Usage Import external routing information into the RIP domain. Redistribution of connected routes, static routes, and routes learnt via OSPF is handled independently, e.g., use **”redistribute ospf”** to import routes learnt via OSPF.

Use **”no redistribute”** to remove all redistribution, and **”no redistribute ospf”** to remove redistribution of routes learnt via OSPF, etc.

Use **”show redistribute [<connected|static|rip>]”** to show the RIP redistribution settings. Use **”show redistribute”** to show all redistribution settings, or **”show redistribute connected”**, etc., to show redistribute settings for specific types of redistribution.

Default values Disabled (**”no redistribute”**)

28.3.9 Configure Admin Distance for RIP

Syntax `[no] distance <1-255>`

Context Router RIP Configuration context

Usage Configure admin distance for all routes learnt via RIP. If the same route is learnt via different routing protocols (or as connected or static route), the route associated with the lowest admin distance will be used. For RIP the admin distance defaults to 120. See also sections 19.2.6 and 26.1.3.

Use **”no distance”** to reset the RIP admin distance to its default value.

Use **”show distance”** to show the configured RIP admin distance value.

Default values 120

28.3.10 Manage Interface Specific RIP Settings

Syntax `[no] rip`

Context Interface Configuration context

Usage Enter the Interface RIP configuration context, i.e., the context where Interface specific RIP settings are configured.

Use **”no rip”** to remove any specific RIP settings for this interface.

Use **”show rip”** to show a summary of RIP settings for this interface.

Default values Disabled (i.e., no interface specific RIP settings)

28.3.11 Configure Interface RIP Passive Settings

Syntax `[no] passive [auto]`

Context Interface RIP Configuration context

Usage Control whether a specific interface should be passive (**”passive”**), active (**”no passive”**), or to automatically follow (**”passive auto”**) the global

RIP setting declared by the **”[no] passive-interface”** setting in *router rip* context (see section 28.3.6).

Use **”show passive”** to show the RIP passive interface setting (passive, active or ”auto”) for this interface.

Default values Auto (**”passive auto”**)

28.3.12 Configure Split Horizon Setting

Syntax `[no] split-horizon [poisoned-reverse]`

Context Interface RIP Configuration context

Usage Enable or disable *split horizon* on this interface, with optional *poison reverse*. Split horizon is a RIP mechanism to mitigate the *counting to infinity* issue appearing in distance vector protocols such as RIP. Poisoned reverse is a variant where the router actively advertises routes as unreachable over the interface which they were learned. The effect of such an announcement is to immediately remove most looping routes before they can propagate through the network.

Use **”show split-horizon”** to show whether *split horizon* is enabled on this interface or not. If the optional *poisoned reverse* setting is enabled, that is also stated.

Default values Enabled (**”split-horizon”**), with poison reverse disabled.

28.3.13 Configure RIP Version for Sending on this Interface

Syntax `[no] send-version <1,2>`

Context Interface RIP Configuration context

Usage Control whether this interface should use the global RIP version setting (section 28.3.2) when sending RIP messages on this interface (**”no send-version”**), or to override the global setting by sending RIPv1 (**”send-version 1”**), RIPv2 (**”send-version 2”**), or both RIPv1 and RIPv2 (**”send-version 1,2”**).

Use **”no send-version”** to remove override settings and return to *auto* setting. (Override can also be removed for individual versions, e.g., **”no send-version 1”** to remove version 1 as override setting.)

Use **”show send-version”** to show RIP version override settings when accepting incoming RIP messages on this interface.

Default values Auto (**”no send-version”**)

28.3.14 Configure RIP Version for Receiving on this Interface

Syntax `[no] receive-version <1,2>`

Context Interface RIP Configuration context

Usage Control whether this interface should use the global RIP version setting (section 28.3.2) when accepting incoming RIP messages on this interface (**”no receive-version”**), or to override the global setting by accepting RIPv1 (**”receive-version 1”**), RIPv2 (**”receive-version 2”**), or both RIPv1 and RIPv2 (**”receive-version 1,2”**).

Use **”no receive-version”** to remove override settings and return to *auto* setting. (Override can also be removed for individual versions, e.g., **”no receive-version 1”** to remove version 1 as override setting.)

Use **”show receive-version”** to show RIP version override settings when accepting incoming RIP messages on this interface.

Default values Auto (**”no receive-version”**)

28.3.15 Configure Authentication of RIP Messages

Syntax `[no] auth <md5 [KEYID] | plain> <SECRET>`

Context Interface RIP Configuration context

Usage Configure authentication of RIP messages *on this interface*. Two authentication methods are available:

- *MD5:* Use **”auth md5 <KEYID> <SECRET>”** to use a MD5 cryptographic authentication. MD5 secrets are text strings of 4-32 characters. A key identifier (0-255) is associated with MD5 keys. (Both the secret and the key identifier must be the same on neighbour routers.)
- *Plain:* Use **”auth plain <SECRET>”** to use a clear-text password as authentication. Plain text secrets are text strings of 4-16 characters. (The secret must be the same on neighbour routers.)

Use **”no auth”** to disable authentication of RIP messages on this interface.

Use **”show auth”** to show the RIP authentication setting for this interface.

Default values Disabled

28.3.16 Show RIP Status Information

Syntax `show ip rip` (or simply **”show rip”**)

Context Admin Exec context.

Usage Show RIP status information, e.g., active interfaces, discovered RIP neighbours, etc.

Default values Not applicable

Chapter 29

IP Multicast Routing

This chapter describes the mechanisms involved in IP multicast routing and how to setup and debug static multicast routing in WeOS.

29.1 Summary of WeOS Multicast Routing Features

Feature	Web	CLI	General Description
Enable IP Forwarding	X	X	Section 29.1.1
Enable IP Multicast Forwarding	X	X	-”-
Configure Static Multicast Routes	X	X	-”-
Multicast Routing Statistics	X	X	-”-
Related Settings			
Layer-2 multicast forwarding			
IGMP Snooping	X	X	Section 29.1.3
Static Multicast Router Ports	X	X	-”-
Static MAC FDB entries		X	-”-
Block local ping responses	X	X	Section 29.1.4
VRRP control of IP Multicast	X	X	Section 30.1.6

29.1.1 Overview of IP multicast

Multicast is an efficient data distribution mechanism for purposes of reaching more than one receiver. IP multicast applications, such as a camera, need only send one packet to reach a group of receivers. The network infrastructure, switches and routers, send a copy of the packet to each subscriber of the group.

A multicast group is an IP address. In IPv4 the entire 224.0.0.0/4 block is reserved, i.e., 224.0.0.0 – 239.255.255.255. However, not all address are available to the end-user and some use-cases may not provide the most optimal distribution in switched (layer-2) networks.

The 224.0.0.0/24 subnet (224.0.0.*) is 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, see RFC1112[6] for details.

The lack of reserved multicast MAC addresses may be a problem in switched networks where the switch fabric often only supports IGMP Snooping (Sec. 18.1), i.e., filtering, per MAC address. E.g., subscribers of group 224.1.2.3 will also receive all traffic sent to group 225.1.2.3.

This is due to the mapping to MAC addresses, in our case

- 224.1.2.3 maps to 01:00:5e:01:02:03
- 225.1.2.3 maps to 01:00:5e:01:02:03
- etc.

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 currently only supports the latter.

29.1.2 Static multicast routing

Contrary to static unicast, multicast has 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
- Setup a multicast route
- Multicast data with 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 static multicast route is made up of a *group*, an *inbound interface*, an optional *sender address* and one or more *outbound interfaces*. There can be at most 128 multicast routes with at most eight (8) outbound interfaces per route.

The source, or *sender address*, is optional in WeOS but the underlying Linux kernel still needs a source address to be able to route the traffic. The multicast routing daemon in WeOS 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 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.

29.1.3 IP multicast and IGMP Snooping

In LAN networks IGMP Snooping is often employed in switches to limit the distribution of IP multicast. Without subscribers to a certain multicast group, distribution of a camera's multicast stream is halted at the first switch. When IGMP Snooping is disabled, the camera's multicast stream is instead broadcast to all ports on the switch, or all ports in the VLAN. For details, see Sec. 18.1 and Sec. 13.1.5.

In currently available network equipment, as well as modern operating systems, IGMP is a well established protocol that works well. There may however still exist older networking equipment, e.g., Programmable Logic Controllers (PLCs), that does not know how to join a multicast group using IGMP. For such devices to receive multicast it is possible in WeOS to either disable IGMP Snooping per VLAN, add a specific FDB MAC entry for the multicast group to open up additional ports in the switch, or use the multicast router port feature to forward all multicast on a given port.

29.1.4 Blocking Local Ping Responses

To ensure that the multicast stream actually is received for routing by the CPU, the WeOS router sends an IGMP join for the multicast group to be routed on the given inbound interface. This has the odd side-effect that the router now also responds to local pings to that group. To disable this, see Sec. 19.7.16.

29.2 Managing Multicast Routing via Web Interface

Menu path: Configuration ⇒ Routing ⇒ Common

The WeOS web interface has full support for managing, configuring and debugging, static IP multicast routing.

To be able to route multicast both the Unicast and Multicast forwarding tick boxes must be checked. The Unicast tick box is actually the big switch that controls all IP routing.

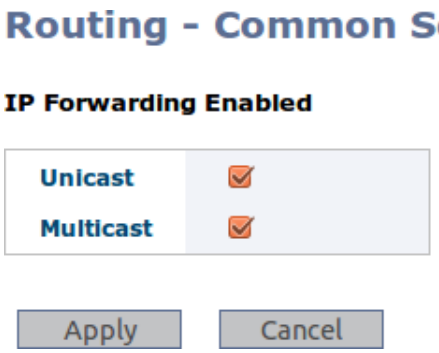


Figure 29.1: Enable IP multicast forwarding.

29.2.1 Adding a Static Multicast Route

Menu path: Configuration ⇒ Routing ⇒ Static Multicast

By default no static multicast routes are setup. Click on New to create a new static multicast route.

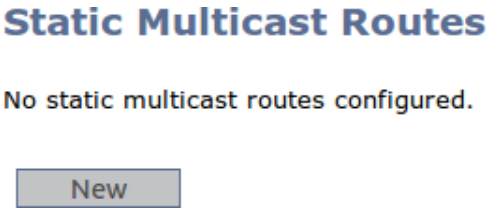


Figure 29.2: No multicast routes enabled by default.

Enter the IPv4 multicast group address, the inbound interface and the source of the sender.

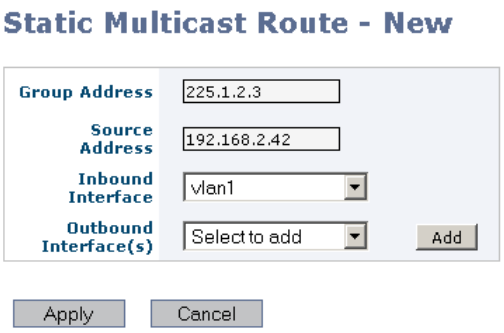


Figure 29.3: Declare multicast group, inbound interface and source of sender.

Add outbound interfaces to your multicast route by selecting them in the drop down and clicking Add for each one.

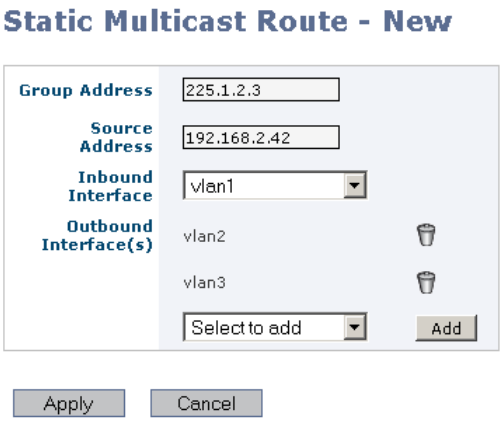


Figure 29.4: Select an outbound interface and press Add for each one.

29.2.2 Adding a Sourceless Static Multicast Route

Menu path: Configuration ⇒ Routing ⇒ Static Multicast

WeOS supports ”source-less” static multicast routes as well, simply leave the *Source Address* field empty.

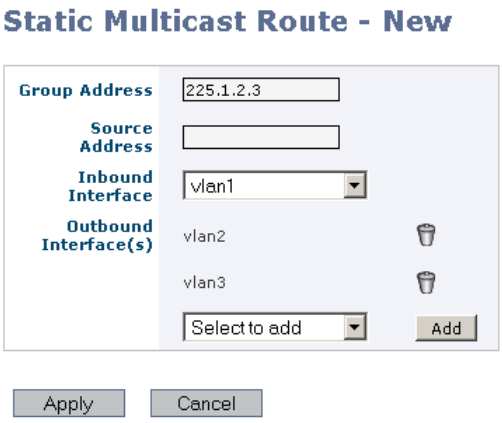


Figure 29.5: Source-less: declare only multicast group, inbound and outbound interfaces.

29.2.3 Overview of Configured Multicast Routes

Menu path: Configuration ⇒ Routing ⇒ Static Multicast

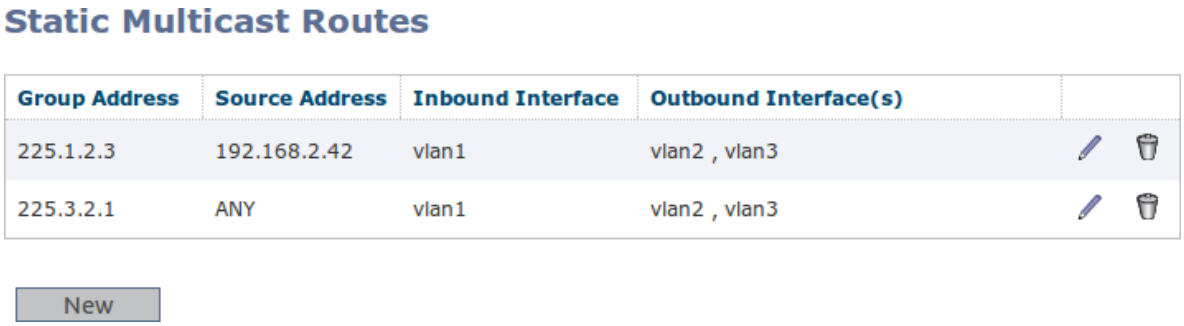


Figure 29.6: Overview of configured static multicast routes.

29.2.4 Deleting a Static Multicast Route

Menu path: Configuration ⇒ Routing ⇒ Static Multicast

In the overview, click the trashcan icon for the static multicast routing rule to delete.

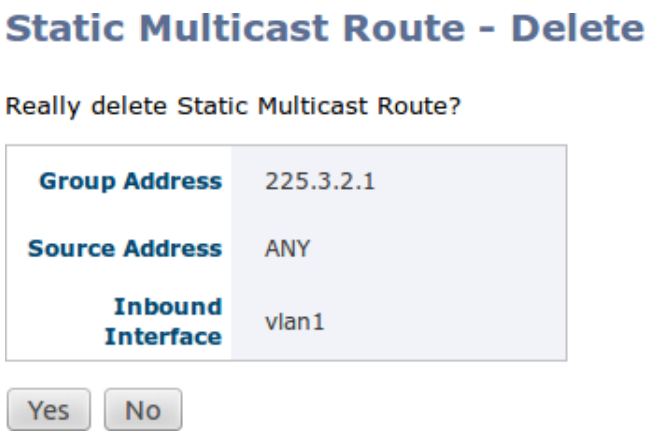


Figure 29.7: Confirm deleting a static multicast route by clicking Yes.

29.2.5 Show Kernel Multicast Routing Table

Menu path: Status ⇒ Routing ⇒ Multicast Routes

The actual kernel multicast routing table is very useful to inspect for debugging, e.g., seeing the amount of packets routed or any on-demand added ”source-less” multicast routes.

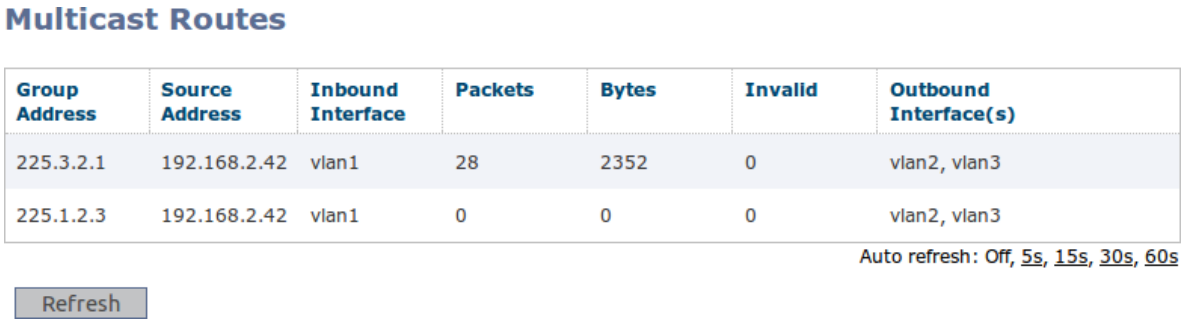


Figure 29.8: Kernel multicast routing table, active multicast routes.

29.3 Managing Multicast Routing via CLI

The following table shows CLI commands relevant for managing, debugging and querying static multicast routes in WeOS.

Command	Default	Section
Configure IP multicast routing		
ip		
[no] multicast-forwarding	Disabled	Section 29.3.1
[no] mroute group <MCADDR> in <IFNAME> [src <IPADDR>] out <IFNAME-LIST>		Section 29.3.2
Show IP multicast routing status		
show ip mroute		Section 29.3.3

There are some additional CLI settings which may be of interest when configuring IP multicast on your unit. The table below lists the most relevant settings.

Command	Default	Section
Related settings (IGMP, MAC FDB, VRRP, etc.)		
fdb		
[no] mac <MACADDR> port <PORTLIST>		Section 13.4.3
vlan <VID>		
[no] igmp	Enabled	Section 13.4.13
ip		
[no] mcast-router-ports <PORTLIST>	Disabled	Section 18.3.4
[no] forwarding	Enabled	Section 19.7.4
icmp		
[no] broadcast-ping	Enabled	Section 19.7.16
firewall		
[no] filter [ARGS . . .]		Section 31.3.3
[no] nat [ARGS . . .]		Section 31.3.5

Continued on next page

Continued from previous page

Command	Default	Section
iface <IFNAME>		
vrrp <INSTANCE>		
[no] mroute-ctrl	Disabled	Section 30.3.12
Related status commands (MAC FDB, IGMP, etc.)		
show fdb		Section 13.4.19
show ip igmp		Section 18.3.6
show firewall		Section 31.3.13

29.3.1 Enable/disable IP multicast forwarding

Syntax `[no] multicast-forwarding`

Context IP Configuration context

Usage Enable/disable IP multicast forwarding (multicast routing). Use command **”multicast-forwarding”** to enable IP multicast forwarding, given that IP forwarding (routing) is enabled (**”forwarding”**, see section 19.7.4).

”no multicast-forwarding” disables IP multicast forwarding.

Use **”show multicast-forwarding”** to show whether IP multicast forwarding is enabled or disabled.

Default values Disabled (**”no multicast-forwarding”**)

29.3.2 Configure static multicast routes

Syntax
```
[no] mroute group <MCADDR> in <IFNAME>
                  [src <IPADDR>] out <IFNAME-LIST>
```

group <MCADDR> IPv4 multicast group to route

in <IFNAME> Inbound interface for multicast stream

src <IPADDR> Optional IPv4 sender address of multicast stream

out <IFNAME-LIST> Comma separated list of destination/outbound interfaces for multicast stream. MAX:8

Context IP Configuration context

Usage Add/remove a static multicast route.

If the *src* field is omitted from an mroute rule, any multicast stream matching the given group and inbound interface will be added on-demand to the kernel multicast routing table. Use the *Admin Exec* command `show ip mroute` to inspect.

Use the ”no”-form of the command to remove rules. The *src* and *out* arguments are not needed, e.g., **”no mroute group 225.1.2.3 in vlan1”**. Without any arguments **”no route”** will remove all configured static multicast routes.

Use **”show mroute”** to list configured static IP multicast routes.

29.3.3 Show IP multicast status and statistics

Syntax `show ip mroute`

Context Admin Exec context

Usage Show IP Multicast Forwarding table and statistics.

This command is useful to inspect the actual routes setup in the kernel multicast routing table. In particular this command is useful when having setup ”source-less” mroute rules.

Default values Not applicable.

Example Assume you have configured the following mroute rules:

Example

```
example:/config/ip/#> mroute group 225.1.2.3 src 192.168.2.42 in vlan1 out vlan2,vlan3
example:/config/ip/#> mroute group 225.3.2.1 in vlan1 out vlan2,vlan3
```

Then the resulting kernel multicast routing table may end up looking like this:

Example

```
example:/#> show ip mroute
Group           Source          Inbound   Packets  Bytes    Invalid  Outbound
================================================================================
225.1.2.3       192.168.2.42    vlan1     0        0        0        vlan2, vlan3
225.3.2.1       192.168.2.20    vlan1     0        0        0        vlan2, vlan3
225.3.2.1       192.168.2.21    vlan1     0        0        0        vlan2, vlan3
================================================================================
```

The latter two entries have been added on-demand, this happens as soon as initial multicast data frames from unknown sources are received on interface *vlan1* destined for group 225.3.2.1.

The columns *Packets*, *Bytes* and *Invalid* denote the total number of packets, bytes and number of invalid packets per rule. Please note that when reconfiguring static multicast rules, or when related interfaces go up/down the statistics are reset. So do not rely on them for accurate measurements, they only exist to aid in debugging.

Chapter 30

Virtual Router Redundancy (VRRP)

This chapter describes WeOS support for the Virtual Router Redundancy Protocol version 2 (VRRPv2)[19] and version 3 (VRRPv3)[25].

VRRP is a standard protocol to enable redundancy between a host and its router, in case the router goes down. VRRP can also be used for *load balancing* purposes.

VRRP provides router redundancy for regular (unicast) IP traffic by letting multiple routers share a virtual IP and MAC address. If the (master) router goes down, a *backup* router will automatically take over.

WeOS provides an optional feature, where the VRRP state (*master* or *backup*) is used to enable/disable *IP multicast routing* of incoming IP multicast packets. With this option enabled, the backup router will prevent the routing of (static) IP multicast routes in addition to IP unicast routing. See chapter 29 for information on support for static IP multicast routing in WeOS.

30.1 Introduction to WeOS VRRP support

The table below summarises VRRP support in WeOS.

Feature	Web	CLI	General Description
VRRP Instances	X	X	Sections 30.1.1-30.1.2
Virtual Router IDs (VRIDs)	X	X	Sections 30.1.1-30.1.2
Virtual Router IP Address	X	X	Sections 30.1.1-30.1.2
Virtual Router Priority	X	X	Sections 30.1.1-30.1.2
Static Priority	X	X	Sections 30.1.1-30.1.2
Dynamic Priority	X	X	Sections 30.1.1-30.1.2
Preemption control	X	X	Sections 30.1.1-30.1.2
Version Specific Settings			
VRRP versions (v2/v3)	X	X	Sections 30.1.2-30.1.3
Advertisement Interval	X	X	Sections 30.1.2-30.1.3
Regular (v2)	X	X	Sections 30.1.2-30.1.3
Fast (v3)	X	X	Sections 30.1.2-30.1.3
Message authentication (v2)	X	X	Section 30.1.4
Advanced Features			
Synchronisation Groups	X	X	Section 30.1.5
Multicast Routing Control	X	X	Section 30.1.6
Load balancing	X	X	Section 30.1.7

30.1.1 VRRP Overview

The primary objective of VRRP is to enable redundancy between a *host* and its *neighbour router*, i.e., you can deploy additional routers on an IP subnet as backup routers, and have one of the backup routers to automatically take over if the primary router fails. Fig. 30.1 can be used to illustrate the need for VRRP in such a scenario.

- A host will typically have an IP setting where the default gateway points to a specific router. An example is given in fig. 30.1a, where the host (H) will send all traffic towards the Internet via Router 1 (R1) with IP address

192.168.1.1. If R1 fails, the host will lose Internet connectivity even though a redundant path (R2) happens to exists.

- VRRP enables routers to share a virtual IP (VIP) address. The router with the highest priority acts as master for the VIP address, while the other routers are backups in case the master fails. Fig. 30.1b illustrates the use of VRRP. R1 and R2 are both responsible for the VIP address (192.168.1.3), with R1 as master since it has higher priority (150>100). If R1 goes down, R2 will become master of the VIP address and communication can automatically resume. Note that the default gateway of the host is configured to the VIP address.

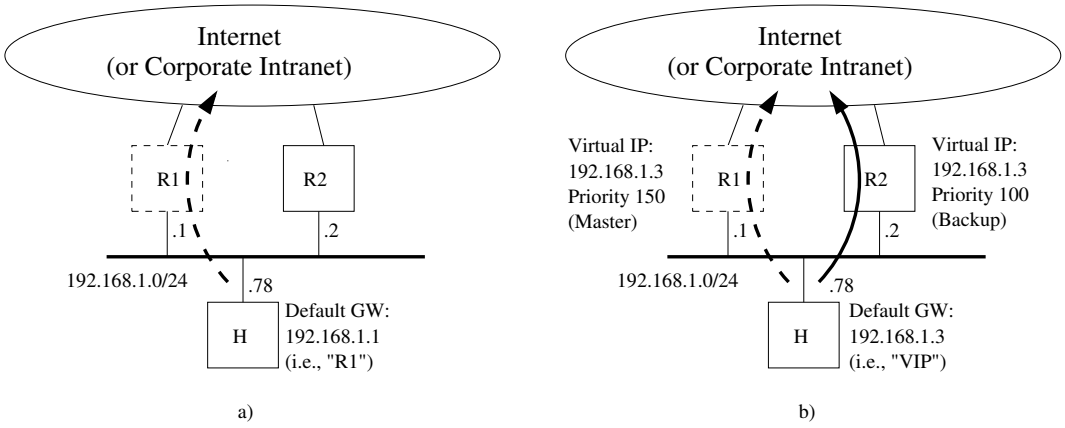


Figure 30.1: Illustrating the need for VRRP to support redundancy: a) Host (H) loses connectivity when Router 1 (R1) fails. b) Host (H) can continue to communicate even though Router 1 (R1) fails, since VRRP enables Router 2 (R2) to take over.

Note
VRRP enables a host to have redundant routers. For redundancy ”router to router”, dynamic routing protocols such as OSPF (chapter 27) or RIP (chapter 28) can be used.

30.1.2 Common VRRP parameters

Some common VRRP parameters are listed below:

1. *VRRP instance:* WeOS allows you to configure up to 16 VRRP instances per unit. Each instance will operate on a (VLAN) *interface* (e.g., *vlan1*) and be

assigned a virtual router identifier (VRID), see item 2 below.

> **Note**
> The ”VRRP instance number” is a parameter only used by WeOS for internal book keeping, e.g., when establishing VRRP synchronisation groups (section 30.1.5). The VRRP instance number is not exchanged in any VRRP message.

2. *Virtual Router Identifier (VRID):* Each instance is assigned a virtual router instance identifier (VRID) in range 0-255. All routers on a LAN, acting as virtual routers for a specific virtual IP address, must be configured with the same VRID. That is, R1 and R2 in fig. 30.1b should have the same VRID, e.g., ”33”.

   > **Note**
   > As of WeOS v4.17.1, a specific VRID (such as ”33”) can only be used once per WeOS unit. Using the same VRID in a second VRRP instance is not possible on a WeOS unit, not even on another LAN.

3. *Virtual IP address (VIP):* WeOS allows you to configure one VIP address per VRRP instance. When designing your network there are some restrictions to consider when selecting the VIP address.

   - *Select VIP in correct IP subnet:* The VIP address should be in the same IP subnet as the regular IP address assigned to the interface (e.g., the VIP address in fig. 30.1b is 192.168.1.3, which is in the same subnet as R1’s and R2’s IP addresses on that subnet).
   - *Select VIP not ”owned” by any router:* Although it is possible to use an address assigned to (i.e., owned by) a router as the VIP address, it is recommended that a separate IP address is used.

     Consider the example in fig. 30.1b): According to the recommendation, the chosen VIP address (”192.168.1.3”) is separate from the addresses assigned to R1 (”192.168.1.1”) and R2 (”192.168.1.2”).

     Although discouraged, it would have been possible to chose ”192.168.1.1” as VIP address. Being the *owner* of the address, R1 must in that case be configured with priority 255, with dynamic priority disabled. More information on VRRP priority is found in item 5 below.

4. *Advertisement interval:* In VRRP, the master will announce its presence by sending VRRP Advertisements on a certain interval. For VRRPv2 the inter-

val can be configured in range 1-255 seconds. VRRPv3 allows sub-second intervals (in steps of 100 ms) in range 0.1-40 seconds. All VRRP routers associated with the same VRID must use the same VRRP version (see section 30.1.3), and must have the same advertisement interval setting.

A low VRRP advertisement interval gives faster fail-over (the time to detect that a master is down is roughly 3 times the advertisement interval).

Default advertisement interval: **1 (second)**

5. *VRRP Priority:* The VRRP priority parameter is used to define which router should become master of the VIP address when multiple routers are available. (If two routers with the same priority transitions to master state, the router with the highest IP address will win the election.)

   The priority can be configured in range 1-255, where the value ”255” should be used if (and only if) the router is also the *owner* of the VIP address (see the Note in item 3 above). Default priority: **100**

   WeOS supports *dynamic VRRP priority*. E.g., if the master router loses its Internet connection it should lower its priority dynamically (or even decline to be master), this to allow for a backup router to take over immediately. For example, if R1 in fig. 30.1b would lose its upstream connection, it could lower its priority to 30, whereby R2 would could take over if preemption is enabled.

   In WeOS, dynamic VRRP priority is configured by mapping the status of an event trigger, typically a *ping trigger* (see section 24.1) to a priority *adjustment* value.

   If a router is the *owner* of the VIP, it should be configured with priority ”255”, with dynamic priority disabled.

6. *VRRP Preemption:* The VRRP master election is not controlled by the priority setting alone; there is also a *preemption* parameter, which enables you to select to have a *deterministic* master election procedure (highest priority always becomes master), or a *sticky* behaviour where the elected master router would keep its role even when another router with higher priority later appears on the network. With *preemption* disabled, the second router would refrain from taking over as long as the current master continuous to send advertisements.

   The exception to this is if the new router connected to the subnet is the VIP address *owner* (priority 255); the VIP owner will always preempt an existing master.

When preemption is enabled, an optional preemption delay parameter can be configured (default 0 seconds), which determines how long the router should wait until preemption is activated. Default: **Disabled**

> **Note**
> When the instance belongs to a synchronized group, the instance with the shortest preemption delay will be used.

> **Note**
> Preemption only occurs when starting or restarting a higher priority backup router, e.g. if a link down event occurs preemption will not be used.

A sample VRRP configuration for R1 in fig. 30.1b is shown below:

Example

```
router vrrp 1
        iface vlan2
        address 192.168.1.3
        vrid 33
        priority 150
        end
```

30.1.3 Selecting VRRP version (VRRPv2 or VRRPv3)

WeOS supports VRRP version 2 and version 3.The additions to version 3 is shorter advertisement interval (faster failover) and IPv6 support (not supported in WeOS). Authentication has been removed completely in version 3 since it was considered to not provide any real security. It is mandatory that the master and the backup routers uses the same VRRP version. Default: **VRRPv2**

30.1.4 Authentication (VRRPv2 only)

> **Warning**
> Use of VRRP authentication is discouraged[11], as it may cause more harm than help.

For VRRPv2, WeOS supports a simple form of VRRP message authentication, enabling the inclusion of a plain-text password in the VRRP advertisements[19].

To avoid that multiple master routers appear on an IP subnet, a WeOS VRRP router will refrain from becoming master if it hears another router with mismatching VRRP authentication information.

30.1.5 VRRP Synchronisation Groups

VRRP synchronisation is a function to keep the VRRP role (master vs backup) the same for different VRRP instances on the same unit, see fig. 30.2.

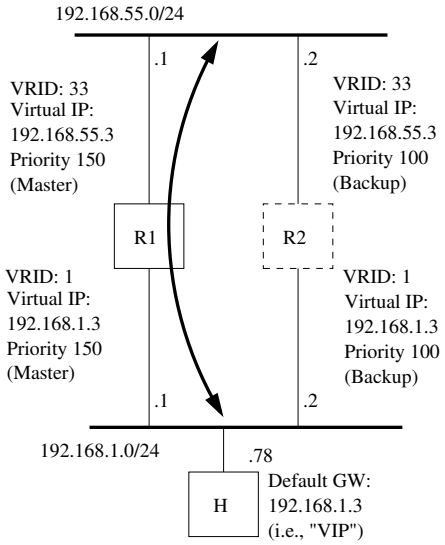


Figure 30.2: Illustrating a topology using synchronised groups. Both instances on R1 will always remain in master state as long no fault is detected (e.g. link down). On fault R1 will become backup on both instances and R2 will become master for both instances.

A synchronisation group consists of two VRRP instances. These two instances should be active on different VLAN network interfaces, e.g. VRID 1 on interface vlan1 can be synchronized with VRID 33 on interface vlan2. The VRRP instances on a unit will only take the master role if it considers itself to have the highest VRRP priority for both instances. If one of the VRRP instances in the synchronisation group would transistion to backup state (e.g. link down), the other instance will also change state to backup, i.e. the instances in the synchronisation group will always have the same state.

30.1.6 VRRP Control of static IP Multicast Routing

When using static multicast routing and VRRP a problem that can occur is that the multicast packets will get duplicated. This can be avoided by using the VRRP multicast routing control. When using this feature, only the master router will forward incoming multicast traffic from the configured VRRP interface while the backup router will prevent the packets from being forwarded.

Note
The setting is applied per interface. It is not recommended to configure more than one instance per interface as this will lead to unpredictable results.

30.1.7 Load sharing

It is possible to use VRRP for load sharing between routers, and still provide redundancy, by having the routers acting as backup for each other. Fig. 30.3 shows a load sharing example. Here the VIP addresses reside within the same IP subnet. However, since WeOS supports multi-netting, the VIP addresses could be on different IP subnets.

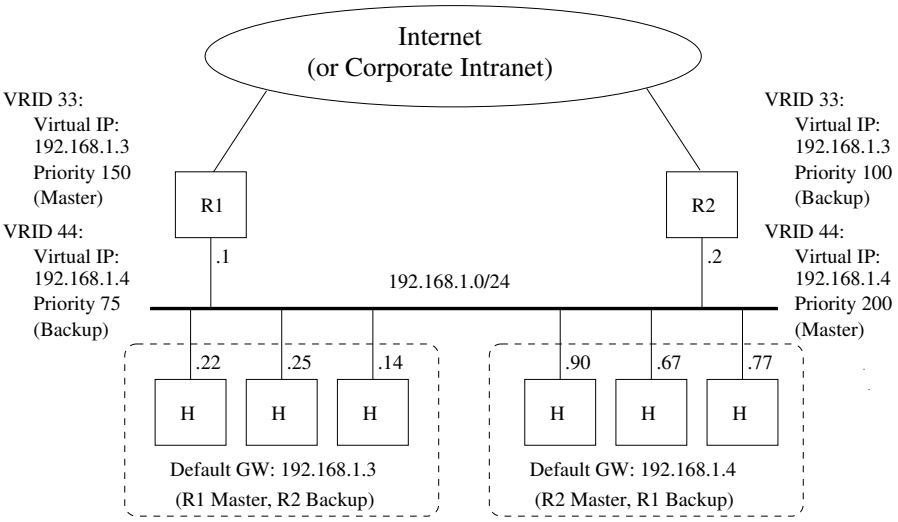


Figure 30.3: Example setup where R1 and R2 share the load from IP subnet 192.168.1.0/24, and using VRRP to backup each other.

30.2 Managing VRRP via the web interface

Menu path: Configuration ⇒ Routing ⇒ VRRP

The main VRRP configuration page lists the currently configured VRRP instances on all interfaces.

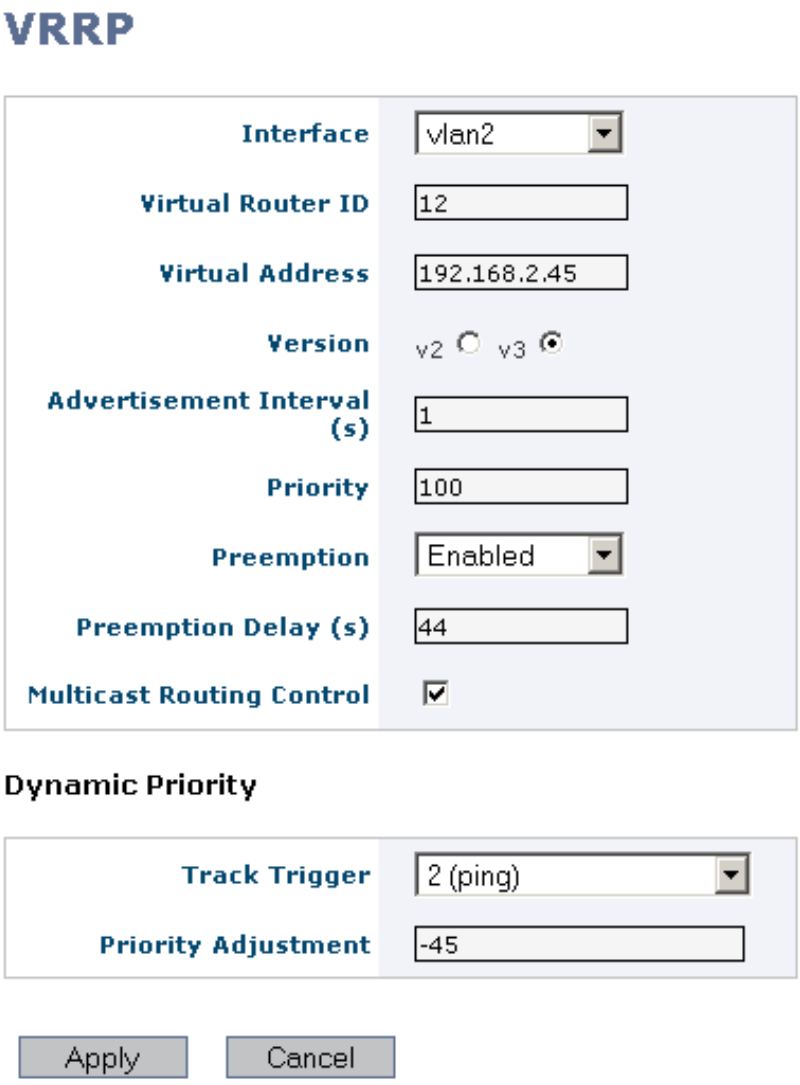
Grouping	To work with groups for synchronised fail-over, select two instances or a group for grouping/ungrouping. A group is displayed with a **[** linking the grouped instances, and common background colour.
Interface	The interface on which to listen for VRRP information and act as gateway. Only VLAN interfaces may be selected.
VRID	Virtual Router ID. A unique ID common to those routers that will provide redundancy..
Edit	Click this icon to edit a VRRP instance.
Delete	Click this icon to remove a VRRP instance. You will be asked to acknowledge the removal before it is actually executed.
Button New	lick this button to create a new VRRP instance.

Continued on next page

	Continued from previous page
Button `Group`	For synchronised fail-over - first select two ungrouped VRRP instances and then click this button to group the instances.
Button `Ungroup`	For synchronised fail-over - first select one group of VRRP instances and then click this button to ungroup the instances. They will be left as two individual instances that has to be removed separately.

30.2.1 Create a new VRRP instance using the web interface

Menu path: Configuration ⇒ Routing ⇒ VRRP ⇒ **New**



Interface	The interface on which to listen for VRRP information and act as gateway. Only VLAN interfaces may be selected.

Continued on next page

	Continued from previous page
Virtual Router ID	A unique ID common to those routers that will provide redundancy.
Virtual Address	A virtual address that the routers will use when providing the gateway support. The VIP address should be in the same IP subnet as the regular IP address assigned to the interface
Version	VRRP version to use (v2 or v3).
Advertisement Interval	The interval in seconds how often a VRRP advertisement message will be sent out. Allowed values: v2: 1-255 seconds v3: 0.1-40 seconds, in 100 msec intervals between 0.1 and 1.0 (default: 1).
Advertisement Interval	The interval in seconds how often a VRRP advertisement message will be sent out. Allowed values: 1-255 seconds (default: 1)
Priority	A number used for election of current gateway. A higher number means a higher chance to become elected. If two routers has the same priority in an election, the router with the highest IP address will win. The value 255 should be used if (and only if) the router is also the owner of the virtual IP address. Allowed values: 1-255 seconds (default: 100)
Preemption	Enable/disable preemption and, if enabled, set a preemption delay. Preemption allows an elected router to remain as master for a time period If the new router is the virtual IP address owner (priority 255), it will always become the master. Default: Disabled
Multicast Routing Control	Let VRRP control multicast routing. If checked, multicast routing will be disabled automatically for this instance when entering BACKUP state. Only one VRRP instance per interface may be configured for controlling multicast routing. The checkbox is disabled if another instance is in control.

For more information on the different settings, see section 30.1.1.

30.2.1.1 Dynamic Priority

Track Trigger	If not disabled, the alarm trigger selected will, if triggered, add the priority adjustment value to the router priority.
Priority Adjustment	A positive or negative number to add to the priority when the alarm has triggered. Allowed values: -255 to 255.

For more information on the different settings, see section 30.1.1.

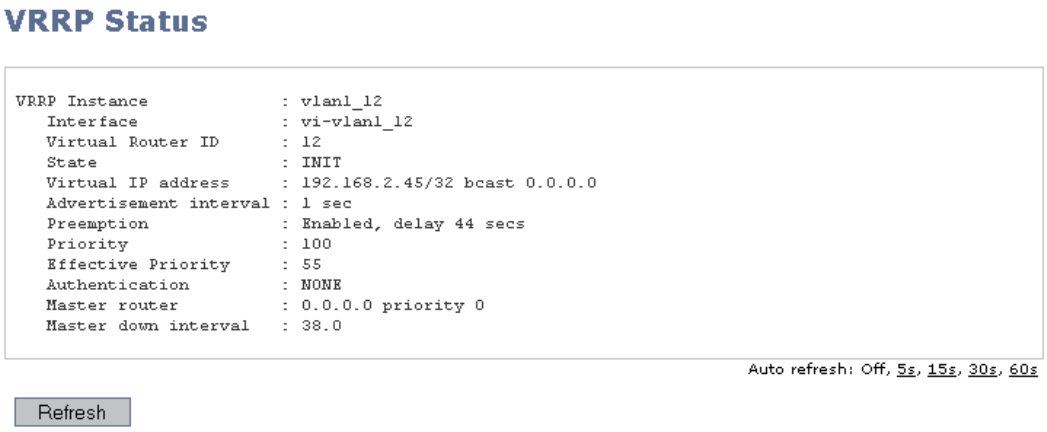
30.2.2 Edit VRRP settings using the web interface

Menu path: Configuration ⇒ Routing ⇒ VRRP ⇒ 

For description of fields, see section 30.2.1.

30.2.3 VRRP Status Page

Menu path: Status ⇒ Routing ⇒ VRRP



Show the status of all configured VRRP instances.

30.3 Managing VRRP via the CLI

The VRRP CLI syntax has been changed from an approach where VRRP was configured *per (VLAN) interface* to an approach where VRRP instances are configured as a common router service. Entering the configuration via both methods has been supported since WeOS v4.9.x. When storing the configuration, WeOS v4.13.x uses the new (router service) style.

Command	Default	Section
Configure VRRP Settings		
router		Sec. 26.3.1
[no] vrrp <INSTANCEID>		Sec. 30.3.1
[no] iface <IFNAME>		Sec. 30.3.2
[no] vrid <VRID>		Sec. 30.3.3
[no] version <2|3>	2	Sec. 30.3.4
[no] address <ADDRESS>		Sec. 30.3.5
[no] interval <INTERVAL> [ms]	1	Sec. 30.3.6
[no] priority <1..255>	100	Sec. 30.3.7
[no] preempt [delay <0..1000>]	Disabled	Sec. 30.3.8
[no] auth <plain> <SECRET>	Disabled	Sec. 30.3.9
[no] track trigger <ID> adjust <DELTA>	Disabled	Sec. 30.3.10
[no] sync <INSTANCEID>	Diabled	Sec. 30.3.11
[no] mroute-ctrl	Disabled	Sec. 30.3.12
View VRRP Status		
show vrrp		Sec. 30.3.13

30.3.1 Create and Manage a VRRP Instance

Syntax `[no] vrrp <INSTANCEID>`

Context Router Protocol Configuration context

Usage Create, manage, or delete a VRRP instance. Use **"vrrp <INSTANCEID>"** to enter the VRRP Instance Configuration context of the specified VRRP in-

stance (INSTANCEID can be in the range 1-16). If the instance does not already exist, it will be created.

Use **"no vrrp <INSTANCEID>"** to remove a specific VRRP instance, or **"no vrrp"** to remove all configured VRRP instances.

At most 16 VRRP instances can be created per unit.

Use **"show vrrp [INSTANCE]"** to show summary of VRRP settings. Use **"show vrrp"** to list settings for all configured VRRP instances, and **"show vrrp INSTANCE"** to list settings for a specific VRRP instance.

Default values Disabled

30.3.2 Configure VRRP interface

Syntax `[no] iface <IFNAME>`

Context VRRP Instance Configuration context

Usage Configure VRRP interface.

An interface is a mandatory setting (**"no iface"** is an invalid setting).

Use **"show iface"** to show the configured interface for this VRRP instance.

Default values None

30.3.3 Configure Virtual Router ID

Syntax `[no] vrid <VRID>`

Context VRRP Instance Configuration context

Usage Set the virtual router identifier (VRID) used for the VRRP instance. The VRID must be unique per switch.

A virtual router identifier is a mandatory setting (**"no vrid"** is an invalid setting).

Use **"show vrid"** to show the configured virtual router ID (VRID) for this VRRP instance.

Default values None

30.3.4 Configure VRRP Version

Syntax `[no] version <2|3>`

Context VRRP Instance Configuration context

Usage Configure VRRP version to be used.

Use **"no version"** to return to the default version setting.

Use **"show version"** to show the configured version (2 or 3) for this VRRP instance.

Default values 2

30.3.5 Configure Virtual Address

Syntax `[no] address <ADDRESS>`

Context VRRP Instance Configuration context

Usage Set the virtual IP address (VIP address) used for the VRRP instance.

The VIP address should be within the same IP subnet as the regular IP address assigned to the interface (see section 19.6.3).

Only one VIP address can be configured per VRRP instance.

Use **"show address"** to show the configured virtual IP (VIP) address for this VRRP instance.

Default values Disabled

30.3.6 Configure VRRP Advertisement Interval

Syntax `[no] interval <1..MAX> | <100..MAX*1000> msec`

Context VRRP Instance Configuration context

Usage Configure VRRP advertisement interval in seconds or milliseconds. MAX (in syntax description) is depending on version and is 255 for version 2 and 40 for version 3.

For version 2 the allowed interval is <1..255> seconds and for version 3 the allowed interval is <0.1..40> seconds. To configure an interval that is a fraction of a second one must set the interval in milliseconds.

A small value enables faster fail-over.

Use **”no interval”** to return to the default interval setting.

Use **”show interval”** to show the configured advertisement interval for this VRRP instance.

Default values 1 (second)

Example In this example, the interval is set to 500 milliseconds. The setting is only valid for VRRP version 3.

Example

```
example:/config/#> router
example:/config/router/#> vrrp 33
example:/config/router/vrrp-33/#> interval 500 msec
example:/config/router/vrrp-33/#> leave
example:/#> copy running start
```

30.3.7 Configure VRRP Priority

Syntax `[no] priority <1..255>`

Context VRRP Instance Configuration context

Usage Configure VRRP priority. A high value increases the chance to become master of the VIP address (see also the **”preempt”** command in section 30.3.8).

Priority ”255” should be used if (and only if) this router is the *owner* of the IP address used as VIP address, i.e., if the VIP address is assigned as an IP address to this router’s interface (see section 19.6.3).

Use **”no priority”** to return to the default priority setting.

Use **”show priority”** to show the configured VRRP priority for this VRRP instance.

Default values 100

30.3.8 Enable or Disable VRRP Master Preemption

Syntax `[no] preempt [delay <0..1000>]`

Context VRRP Instance Configuration context

Usage Enable or disable VRRP master preemption. If enabled, this router will preempt an existing master if the current master has lower priority. (Note: The *owner* of a VIP address will always take over as master irrespective of the **"preempt"** setting.)

When preemption is enabled, the router will wait a time interval depending on the configured advertisement interval and a configurable preemption delay (seconds) before taking over as master.

> **Note**
> Preemption only occurs when starting or restarting a higher priority backup router, e.g. if a link down event occurs preemption will not be used.

> **Note**
> Note: When the instance belongs to a synchronized group, the instance with the shortest preemption delay will be used.

Use **"no preempt"** to prohibit this router to preempt an existing VRRP master.

Use **"show preempt"** to show the configured VRRP master preemption setting for this VRRP instance.

Default values Disabled (**"no preempt"**) When enabled, the delay defaults to 0 seconds.

30.3.9 Configure VRRP Message Authentication

Syntax `[no] auth <plain> <SECRET>`

Context VRRP Instance Configuration context

Usage Configure VRRP message authentication. Simple clear-text authentication is supported for VRRP version 2.

The associated secret can be 4-7 characters. Valid characters are ASCII characters 33-126, except '#' (ASCII 35).

Authentication is not available in VRRP version 3. Authentication will automatically be disabled if version 3 is configured. Use **"no auth"** to disable VRRP message authentication.

Use **"show auth"** to show the configured VRRP message authentication setting for this VRRP instance.

Default values Disabled

30.3.10 Configure VRRP Dynamic Priority

Syntax `[no] track trigger <ID> adjust <DELTA>`

Context VRRP Instance Configuration context

Usage Configure dynamic VRRP priority. The VRRP priority will be adjusted by the given delta value (-255 to 255) when the associated trigger reports "alarm" status. E.g., **"track trigger 2 adjust -100"** will decrease the VRRP priority by 100 when there is an alarm condition on trigger 2.

When a router is the owner of the VIP, i.e. configured with priority "255", the dynamic priority has no effect.

Use **"no track"** to remove (all) track entries defined for this VRRP instance. (As of WeOS v4.17.1, at most one **"track"** entry can be configured.)

Use **"show track"** to show the configured VRRP track entries, i.e., the dynamic VRRP priority setting.

Default values Disabled

Example In this example, this virtual router's priority is lowered from *150* to *50*, if the router cannot reach the host *192.168.3.11* through the (upstream) interface *vlan2*.

Example

```
example:/config/#> alarm
example:/config/alarm/#> trigger ping
example:/config/alarm/trigger-2/#> peer 192.168.3.11 outbound vlan2
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> end
example:/config/#> router
example:/config/router/#> vrrp 33
example:/config/router/vrrp-33/#> address 192.168.2.1
example:/config/router/vrrp-33/#> priority 150
example:/config/router/vrrp-33/#> track trigger 2 adjust -100
example:/config/router/vrrp-33/#> leave
example:/#> copy running start
```

30.3.11 Configure VRRP Synchronisation

Syntax `[no] sync <VRRP ID>`

Context VRRP Instance Configuration context

Usage Configure synchronization between two VRRP instances. This will specify a state monitoring between two VRRP instances. It guarantees that two VRRP instances remain in the same state. The synchronized instances monitor each other. Changing this parameter will change the same parameter on the corresponding instance.

Use **"no sync"** to remove synchronization for this instance, this will remove synchronization for the corresponding instance as well.

Use **"show sync"** to show the configured VRRP instance ID this instance is synchronized with.

Default values Disabled

Example In this example, virtual router instance 33 is synchronized with instance 35.

Example

```
example:/config/#> router
example:/config/router/#> vrrp 33
example:/config/router/vrrp-33/#> sync 35
example:/config/router/vrrp-33/#> leave
example:/#> copy running start
```

30.3.12 Configure VRRP Multicast Routing Control

Syntax `[no] mroute-ctrl`

Context VRRP Instance Configuration context

Usage Configure whether multicast traffic should be routed on a interface in BACKUP state. If enabled, muticast traffic will not be routed when VRRP is in BACKUP state.

Use **"no mroute-ctrl"** to remove multicast routing control for this instance.

Use **"show mroute-ctrl"** to show the configured VRRP multicast routing control setting for this instance.

Default values Disabled

30.3.13 Show VRRP Status

Syntax `show vrrp`

Context Admin Exec context

Usage Show the status of all configured VRRP instances.

Default values Not applicable

Chapter 31

Firewall Management

When connecting your network to the Internet (or any non-trusted network) a router with firewall functionality should be used. The firewall will protect against undesired access to your local servers, or other kinds of network intrusion from attackers on the Internet.

The WeOS firewall supports the following main features:

- *Packet filtering:* Packet filters enables you to control what traffic is allowed to pass through your router/firewall and what packets it should drop. Packet filter rules can also be specified to control access to services on your router.
- *Packet modification:* Packet modification makes it possible to modify packets that are routed through the router/firewall.
- *Network Address Translation (NAT):* The WeOS NAT functionality includes both *network address port translation* (NAPT) and *1-TO-1 NAT*.
- *Port forwarding:* Port forwarding is often used together with NAPT, and will then enable you to access servers in your private network from outside (e.g., from the Internet).

The WeOS firewall utilises *connection tracking*; a 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. Application level gateway (ALG) helper functions can be enabled to provide connection tracking of more complex protocols, such as FTP and SIP.

Section 31.1 describes the firewall functionality available in WeOS. Sections 31.2 and 31.3 cover firewall management via the Web Interface and via the CLI.

31.1 Overview

Table 31.1 summarises the supported firewall functionality. Sections 31.1.1-31.1.5 provide further information on the WeOS firewall support.

Feature	Web	CLI	General Description
Enable Firewall	X	X	Sections 31.1.1-31.1.2
Packet filtering			Sections 31.1.1-31.1.2
Enable Packet Filtering	X	X	Sections 31.1.1-31.1.2
Filtering Rules	X	X	Sections 31.1.1-31.1.2
Rule Reordering	X	X	Sections 31.1.1-31.1.2
Activate/Deactivate Rules	X	X	Sections 31.1.1-31.1.2
Default Forward Policy	X	X	Sections 31.1.1-31.1.2
Default Input Policy		X	Sections 31.1.1-31.1.2
Stateful Packet Inspection		X	Sections 31.1.1-31.1.2
Packet modification			Sections 31.1.1, 31.1.3
DSCP	X	X	Section 31.1.3.3
Network Address Translation			
NAPT	X	X	Sections 31.1.1, 31.1.4
1-TO-1 NAT	X	X	Sections 31.1.1, 31.1.4
Port Forwarding	X	X	Sections 31.1.1, 31.1.5
ALG Helpers	X	X	Section 31.1.1
Logging	X	X	Section 31.1.6
View Firewall Configuration	X	X	
View Firewall Status		X	

Table 31.1: Summary of Firewall functionality in WeOS

31.1.1 Firewall introduction

The WeOS firewall includes support for three related types of functionality:

- *Packet Filtering*: The packet filtering support is primarily used to control what traffic is allowed to be *routed* via the switch (forward filtering), but can also be used to control accessibility to services on the switch itself (input filtering).

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 configured to handle more complex protocols by enabling ALG helpers (see below).

WeOS supports up to 1024 filtering rules. The WeOS packet filtering support is further described in sections 31.1.2 and 31.1.2.3.

- *Packet modification*: WeOS currently supports one kind of packet modification:
  - *DSCP*: The Differentiated Services Code Point (DSCP) field of the IP header is used for classifying traffic in some environments. The value of this field can be modified by WeOS *when routing* the IP packets.

  WeOS supports up to 32 packet modifier rules. The WeOS packet modification support is further described in section 31.1.3.
- *Network Address Translation (NAT)*: WeOS supports two kinds of NAT support:
  - *NAPT*: NAPT is the most common NAT form, where a common (public) IP address is shared by a set of hosts in a *private* network. This form of NAT is sometimes referred to as IP Masquerading or port address translation (PAT). NAPT is often used together with *port forwarding*, see below.
  - *1-TO-1 NAT*: 1-TO-1 NAT enables you to translate a whole range of IP addresses to another set of addresses.

  WeOS supports up to 512 NAT rules. The WeOS NAT support is further described in section 31.1.4.
- *Port Forwarding*: Port forwarding is commonly used together with NAPT. With port forwarding a service (such as a Web Server) located in a *private* network, can be made accessible from the *public* network, typically from the Internet.
  WeOS supports up to 256 port forwarding rules. The WeOS port forwarding support is further described in section 31.1.5.

Some network protocols are more complex and therefore more difficult than others to handle by the connection tracking function in a firewall or NAT device. An example is FTP, which utilises a *control connection* to exchange information on TCP port numbers for *data connections* for the 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. As of WeOS v4.17.1 ALG helpers for FTP, TFTP, SIP, IRC, H323 and PPTP are supported. ALG helpers have some impact on the unit's routing performance, thus are by default disabled.

31.1.2 Packet Filtering

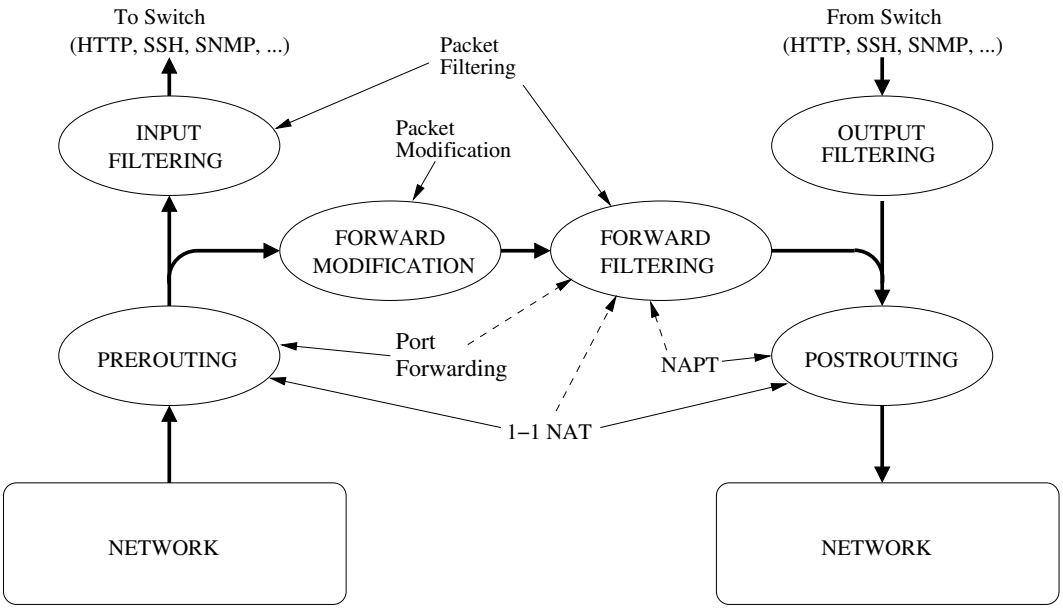


Figure 31.1: Overview of Firewall mechanism. Thick lines represent packet flows.

Fig. 31.1 presents an overview of the firewall mechanism, including the components for *packet filtering*, *packet modification*, *NAT*, and *port forwarding*.

The following sections provide a more in-depth description of the WeOS *packet filtering* functions.

- *Filtering chains (input, forward, output):* Filter rules can apply to
  - traffic destined to the switch (*input filtering*), e.g., HTTP traffic to manage the switch,
  - traffic forwarded/routed by the switch (*forward filtering*), or
  - traffic generated by the switch (*output filtering*).

The WeOS firewall supports *input* and *forward* filtering, but not *output* filtering. Section 31.1.2.1 gives more details on WeOS handling of filtering chains.

- *Configurable allow/deny filter rules:* The user can define filter rules to specify traffic to be *allowed* or *denied*, and the order of the configured rules. Incoming packets are evaluated against the filter rules – the first matching rule will decide how to treat the packet (*allow* or *deny*). Section 31.1.2.2 describes packet matching parameters for filter rules, and section 31.1.2.3 provides more information on filter evaluation order (both for *configured* filter rules and *implicit* filter rules described below).

  Default rules to allow "ping"
  When enabling the firewall, the user is offered to add a set of *default rules* - these rules allow ICMP packet to pass the *input filter*, thereby enabling operators to *ping* the unit after enabling the firewall. These rules are treated as any other configured rule, thus can be removed, etc.

- *Implicit filter rules:* The WeOS firewall implicitly adds firewall rules for services enabled on the unit, e.g., for DHCP, OSPF or DNS. The primary purpose of this is to simplify management of those services when the firewall is enabled. With a few exceptions, these *implicit rules* are evaluated **after** the *configured rules* (see above), thus, a user could override or complement the implicit rules by configuring additional filter rules. Below is a list of services associated with implicit filter rules.

  - IPsec VPN:

    * *IPsec signalling and data encapsulation:* If at least one IPsec tunnel is enabled, rules are implicitly added to allow IP protocol 50 (ESP), and UDP port 4500 (IKE/ESP for NAT traversal) to enter the unit on all interfaces.

    * *Allowing data to pass through tunnels:* For every IPsec VPN tunnel (see chapter 35) filter rules are implicitly added to the forward filter to *allow* between the *local subnet* and *remote subnet* defined for the VPN tunnel.

  As of WeOS v4.17.1, the implicit IPsec VPN rules are added **before** the configured filter rules (for performance reasons). Thus, the implicit IPsec VPN rules can **not** be overridden by rules configured by the user.

  - *Port Forwarding:* With port forwarding (section 31.1.5) it is possible to map incoming data to a given destination IP and (UDP/TCP) port to another destination IP/port when forwarding the packet. As shown in fig. 31.1 this mapping is conducted at the *pre-routing* stage of the packet processing. For every configured port forwarding rule, a filter rule is implicitly added to the forwarding filter to allow the packet to pass through the router. This is hinted by a *dashed arrow* in fig. 31.1.
  - *NAT:* Network address translation (section 31.1.4) involves ”translation operations” both in the pre-routing (”1-TO-1 NAT”) and in the post-routing post-routing stage (”1-TO-1 NAT” and ”NAPT”) as shown in fig. 31.1. For every configured NAT rule, an associated filter rule **can** be added to the forwarding filter to allow the packet to pass through the router. This is hinted by a *dashed arrow* in fig. 31.1.

    Note
    The user can choose if an associated filter rule should be added for each NAT rule or not. If disabled, the user needs to configure own filter rule(s) to make the data packets to pass through the firewall. See sections 31.1.4.1 and 31.1.4.2.3 for more information.

  - *Services:* Filter rules are implicitly added to to the *input filter* to allow packets for enabled services to enter the unit. This includes configurable services such as DHCP Server (chapter 22), Serial Over IP (chapter 39), VRRP (chapter 30), etc., where allow rules are added matching TCP/UDP port numbers, IP protocols, and/or incoming interfaces appropriate for the configured services. As the WeOS unit acts as a DNS forwarder, implicit allow rules to accept incoming DNS requests are also added.

- *Management interface:* The WeOS management interface feature (section 19.2.7) utilises firewall functionality to control which network interfaces the unit can be managed through.
- *Other filter rules:*
  - *Connection tracking (related/established):* The WeOS firewall will allow all packets associated with established connections, as well as packets related to established connections. This means that an a 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. Application level gateway (ALG)

helper functions can be enabled to provide connection tracking of more complex protocols, such as FTP and SIP.

For performance reasons, packets of related/established connections are evaluated early in the filter chains, thus cannot be overridden by filter rules configured by the user.

- *Stateful Packet Inspection (ability to drop packet of invalid state):* It is also possible to fine-tune the connection tracking behaviour to *drop* packets of *invalid*[1] state – this is done by enabling the *stateful packet inspection* (SPI) setting. In some situations that can be considered as a security enhancement, however, it may cause problems in topologies with asymmetric routing and is therefore disabled by default.
- *Default filter rules:* Packets not matching any filter rule will be handled according to the default filter policy. The default filter policy for the *input filter* and *forwarding filter* chains are configurable, see section 31.1.2.1.

31.1.2.1 Filtering chains (input, forward, output)

Fig. 31.1 presents an overview of the firewall mechanism including the filtering chains (input, forward and output). Packets are treated differently if they:

- *are destined to the switch*: Examples include HTTP/HTTPS, SSH, Telnet, and SNMP traffic used to manage the switch remotely, and ICMP (Ping) traffic to check if the switch is up or not. Such packets are subject to *pre-routing* and *input filtering* firewall mechanisms.
- *originate from switch*: This includes the same examples as above (HTTP/HTTPS, SSH, Telnet, SNMP, ICMP, etc.) with the difference that this is the packets from the switch instead of the packets to the switch. Such packets are subject to *output filtering* and *post-routing* firewall mechanisms, however WeOS does **not** include primitives to control *output filtering*.
- *are routed via the switch*: This includes traffic that is not destined for the switch or originate from the switch. Such packets are subject to *pre-routing*, *forward filtering* and *post-routing* firewall mechanisms.

As of WeOS v4.17.1, the selection of filter chain for configured filter rules is implicitly derived from the ”outbound interface” and ”destination IP Address/subnet”

[1] An example of a packet with an ”invalid” state is when a firewall sees a TCP ”SYN+ACK”, without having seen the preceding TCP ”SYN” in the other direction.

settings (see section 31.1.2.2) for the rule:

- *Apply rule to forwarding filter:* If ”outbound interface” **and/or** ”destination IP Address/subnet” are specified in the filter rule, it will apply to the ”Forwarding Filter” chain.
- *Apply rule to input filter:* If **neither** ”outbound interface” **nor** ”destination IP Address/subnet” are specified, the filter rule will apply to the ”Input Filter” chain.

WeOS does not support adding filter rules for the ”Output Filter” chain.

Associated with each filtering chain there is a default policy, defining what to do with packets that do not match any of the defined filter rules. When the firewall is enabled, the *default policies* for packet filtering are as follows:

- *Input Filtering*: **Deny**, i.e., packets to the switch are dropped unless they are explicitly allowed.
- *Forward Filtering*: **Deny**, i.e., when enabling the firewall no packets will be routed by the switch until such packet filter rules are defined.
- *Output Filtering*: **Accept**, i.e., there are no restrictions on the traffic originating from the switch.

31.1.2.2 Filter Rules Packet Matching

Packet filtering *allow* and *deny* rules can be specified to *match* IP packets based on the following filtering parameters:

- *Inbound Interface*: The interface where the packet comes in.
- *Outbound Interface*: The interface where the packet is sent out.
- *Source IP Address/Subnet*: The source IP address of the packet. This can be specified as a single IP address, or the rule could match a whole IP subnet.
- *Destination IP Address/Subnet*: The destination IP address of the packet. This can be specified as a single IP address, or the rule could match a whole IP subnet.
- *Protocol*: The *protocol* type of the IP payload. Typically TCP or UDP, but the filtering can also be made to match other protocols such as ICMP and ESP[2].

[2] See http://www.iana.org/assignments/protocol-numbers/ for a list of defined IP protocols.

- *Destination (UDP/TCP) Port*: When *protocol* is specified as UDP or TCP, the filter can match on the associated UDP/TCP port number(s).

As described in section 31.1.2.1 the filter setting for ”outbound interface” and ”destination IP Address/subnet” implicitly controls whether the rule will apply to the *input filter* or *forwarding filter*.

An incoming packet will be processed according to the *rules* defined for *input filter* when the packet is destined to the switch, or the rules defined for the *forwarding filter* when the packet is being routed through the switch. The list of rules is searched (in order) until a match is found; if no matching rule is found, the packet is treated according default policy of the chain.

For more information on the rule evaluation order in the input filter and forward filter, see section 31.1.2.3.

31.1.2.3 Rule Evaluation Order in Input and Forward Filters

When the firewall is enabled, incoming packets are subject to *input filtering* or *forward filtering* depending if the packet is destined to the switch itself, or if it should be routed to another network. Once the packet has been classified for the input or output filter chain, the list of that chain is traversed to find a matching rule. If a match is found, the packet will either be accepted or dropped depending on the type of matching rule (allow or deny). If no matching rule is found, the packet will be handled according to the default policy of the chain.

The filter rules are inserted in the list in a certain order; the same order as the packet matching evaluation is conducted. To view the current input and forward filter evaluation lists, use the command **”show firewall”** (see section 31.3.13) from the *Admin Exec* context. The order in which rules are inserted in the input and forward filters is described below.

31.1.2.3.1 Input Filter

1. *Established/Related:* Packets part of (or related) to established connections will be accepted. This rule is inserted first for performance reasons - the majority of all accepted packets will match this rule.
2. *Drop invalid:* If the stateful packet inspection (SPI) setting has been enabled, packets of invalid state will be dropped. (See section 31.1.2 for more information on what the SPI setting does.)

3. *VPN Rules:* If the WeOS unit is configured as VPN gateway, rules to accept IKE and ESP traffic are implicitly inserted here (UDP port 500 and 4500, and IP protocol 50).

4. *Configured Packet Filter Rules:* Then the configured packet filter rules are inserted, i.e., the configurable allow/deny rules described here in section 31.1.2. The *relative* order of these packet filter rules is configurable.

   As all packet rules are configured before the rules for ”Enabled Services” and ”Management Interfaces” (see below), the packet filter rules can be used to *override* those rules. E.g., if the *management interface configuration* has disabled SNMP management via interface *vlan1* (**”no management snmp”**, see section 19.6.6), a packet filtering rule allowing host *192.168.3.1* SNMP access (**”filter allow src 192.168.3.1 proto udp dport 161”**, see section 31.3.3) will have precedence, and thus allow SNMP management from that particular host even if the SNMP traffic enters via interface *vlan1*.

5. *Enabled Services:* Depending on what additional services are enabled in the configuration, additional allow rules will be inserted to enable those services to operate correctly. As of WeOS v4.17.1, this includes

   - DHCP Server: UDP port 67 is allowed for appropriate interfaces if a DHCP server is configured (see chapter 22).
   - OSPF: IP protocol 89 is allowed if the unit is configured to run OSPF for dynamic routing (see chapter 27).
   - RIP: UDP port 520 is allowed if the unit is configured to run RIP for dynamic routing (see chapter 28).
   - VRRP: IP protocol 112 is allowed for appropriate interfaces if VRRP is configured on the unit (see chapter 30).
   - Serial Over IP: If Serial Over IP is configured (Server, Peer or AT command mode), an allow rule according to the configured (UDP/TCP) port and interface is added (see chapter 39).
   - Modbus: If the unit is configured as a Modbus gateway (server mode), an allow rule according to the configured TCP port and interface is added (see chapter 40).
   - DNS: UDP/TCP port 53 is allowed on all interfaces as the WeOS unit acts as a DNS forwarder.

6. *Enabled Management Interfaces:* As described in section 19.2.7, an operator can use the *Management Interface* feature to enable/disable services per network interface. The management interface configuration is kept separate from the firewall configuration, but both configuration methods can affect the *Input Filter*. Allow rules for enabled management services are added per interface[3].
    - SSH: TCP port 22 is opened for interfaces where management via SSH has been enabled. (This also enables use of SCP for remote file access, see section 7.1.5.3).
    - Telnet: TCP port 23 is opened for interfaces where management via Telnet has been enabled.
    - HTTP: TCP port 80 is opened for interfaces where management via HTTP has been enabled.
    - HTTPS: TCP port 443 is opened for interfaces where management via HTTPS has been enabled.
    - SNMP: UDP port 161 is opened for interfaces where management via SNMP has been enabled.
    - (IPConfig:) If management via IPConfig service has been enabled, no corresponding allow rule is required - IPConfig protocol packets are instead filtered by other (lower-level) mechanisms in WeOS.
7. *Default Policy:* Packets not matching any of the rules above will be handled according the default policy for the input filter chain.

31.1.2.3.2 Forwarding Filter

1. *Packet modification:* Defined packet modifications are always performed before all filter rules, implicit and configured. Please see section 31.1.3 for additional details.
2. *Established/Related:* Packets part of (or related) to established connections will be accepted. This rule is put first of the forwarding filters for performance reasons - the majority of all accepted packets will match this rule.

[3] As of WeOS v4.17.1 ”allow” rules for *enabled* management services are added given that the ”Default policy” for the input filter is set to ”deny”. If the default policy is changed to ”allow”, then ”deny” rules for *disabled* management interfaces will be inserted instead.

3. *Drop invalid:* If the stateful packet inspection (SPI) setting has been enabled, packets of invalid state will be dropped. (See section 31.1.2 for more information on what the SPI setting does.)
4. *VPN Rules:* If the WeOS unit is configured as VPN gateway, rules to accept traffic between the local and remote subnets specified in the respective IPsec tunnel definitions are added to the forward filter. The reason for adding the implicit IPsec allow filter rules early in the evaluation order is to improve routing performance of VPN traffic. (In case you wish to limit the traffic to pass through the IPsec tunnel further, the recommendation is to update the IPsec tunnel definitions of local and remote subnet accordingly, see section 35.1.1.)
5. *Configured Packet Filter Rules:* Then the configured packet filter rules are inserted, i.e., the configurable allow/deny rules described here in section 31.1.2. The *relative* order of these packet filter rules is configurable.
6. *NAT and Port Forwarding Rules:* As described in section 31.1.2 implicit allow filter rules are added for every configured port forwarding rule.

   This is also true for NAT rules, however, here the user can choose whether the associated rule should be created or not (see sections 31.1.4.1 and 31.1.4.2.3). The internal order of the NAT rules can be changed, which also affects the order in which the associated filter rules are inserted in the forwarding filter chain.
7. *Default Policy:* Packets not matching any of the rules above will be handled according the default policy for the forwarding filter chain.

31.1.3 Packet modification

WeOS supports modification of packets that are *routed* through the router/firewall. In the firewall overview, fig. 31.1 in section 31.1.2, you can see that the modification is performed just before the forward filtering. Current limitations are that you can only modify the DSCP field of the IP header, and that modification is only possible for *forwarded* traffic, not for inbound or outbound local traffic.

Packet modification is specified as rules, similar to filters, and they are evaluated in the same order as they are listed. Opposite to filters (section 31.1.2), packet modification rules are non-terminating. This means that every rule will be evaluated for packets passing through, and packets may be modified more than once on its way through the modifier step.

31.1.3.1 Performance considerations

The *packet filtering* mechanism utilises the connection tracking mechanism to optimise handling for already established sessions, while *packet modification* rules can not use this connection tracking benefit. The modification rules will be evaluated for *every single forwarded packet* passing the router/firewall, which means that modification rules have a much *bigger performance impact* than filtering rules.

As using modifier rules decreases the total routing throughput of the router/firewall, you should use this feature with care and avoid adding unnecessary rules.

31.1.3.2 Packet modification matching

Much like packet filters, modification rules can have *match* parameters defining what traffic the rules apply to. The matching parameters are optional – if skipped the modifier rule runs for *ALL* packets.

These are the matching parameters that can be used:

- *Inbound Interface*: The interface where the packet comes in.
- *Outbound Interface*: The interface where the packet is sent out.
- *Source IP Address/Subnet*: The source IP address of the packet. This can be specified as a single IP address, or the rule could match a whole IP subnet.
- *Destination IP Address/Subnet*: The destination IP address of the packet. This can be specified as a single IP address, or the rule could match a whole IP subnet.
- *Protocol*: The *protocol* type of the IP payload. Typically TCP or UDP, but the filtering can also be made to match other protocols such as ICMP and ESP[4].
- *Destination (UDP/TCP) Port*: When *protocol* is specified as UDP or TCP, the filter can match on the associated UDP/TCP port number(s).

[4] See http://www.iana.org/assignments/protocol-numbers/ for a list of defined IP protocols.

31.1.3.3 Modification of the DSCP field

31.1.3.3.1 DSCP Introduction

DSCP, Differentiated Services Code Point (or Diffserv Code Point), is a standardised method for marking IP packets that they belong to a specific class of traffic. Its use in the IP header is specified in RFC 2474[26].

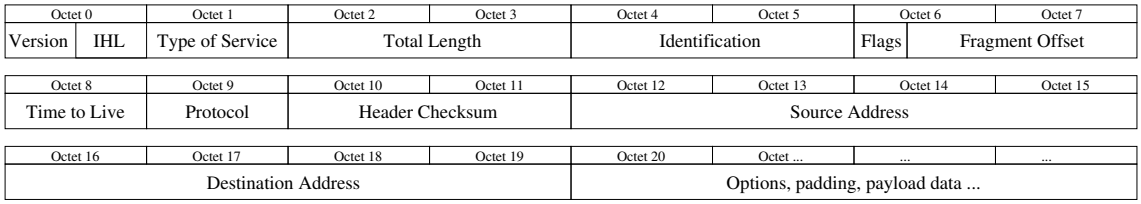


Figure 31.2: The IPv4 header

For the IPv4 header (RFC 791[29]), the ”Type of service” (or ToS) octet on offset 1 is used for carrying this kind of data. See fig. 31.2.

The IPv4 ToS octet has historically been used in different ways.

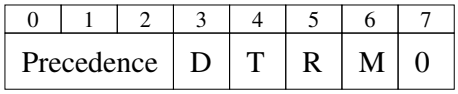


Figure 31.3: ToS bits according to RFC 791 + RFC 1349

The original definition of ToS in RFC 791 has 3 precedence bits, and bits 3-5 as flags for ”cost” aspects: ”Delay”, ”Throughput” and ”Reliability”. RFC 1349[2] updated ToS by adding the utilisation of bit 6 for ”Monetary cost”. See fig. 31.3.

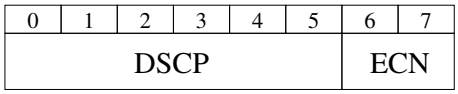


Figure 31.4: ToS bits according to RFC 2474 + RFC 3168

Later on, RFC 2474 redefined the use of the octet to carry DSCP information in the first 6 bits. RFC 2481[30] and its replacement RFC 3168[31] complement this by defining bits 6-7 for ”Enhanced Congestion Notification” (ECN), see fig. 31.4.

Both these conflicting interpretations are still in use today confusingly enough. The DSCP modification and the Layer-2 prioritising mechanisms (section 8.1.4) in WeOS are adapted to the RFC 2474 use.

31.1.3.3.2 Setting DSCP

WeOS can set the 6 DSCP bits in the IP ToS field with a modifier rule. The two last bits (Enhanced Congestion Notification) are not modified by this operation.

The decimal values 0-63 must be used when setting DSCP.

Several RFCs define standard DSCP values called ”Per-Hop Behaviors” or PHBs. WeOS does not support the PHB names for configuration, but the table below can be used to convert PHB names to the corresponding decimal values.

PHB Name	DSCP value
DF	0
CS1	8
AF11	10
AF12	12
AF13	14
CS2	16
AF21	18
AF22	20
AF23	22
CS3	24
AF31	26

PHB Name	DSCP value
AF32	28
AF33	30
CS4	32
AF41	34
AF42	36
AF43	38
CS5	40
VA	44
EF	46
CS6	48
CS7	56

31.1.3.3.3 DSCP Adjust priority

There is an additional parameter called ”adjust priority” that can be added to a DSCP modifier rule. This parameter enables adjustment of the router’s internal packet priority handling inline with the modified DSCP value. Furthermore, if traffic is routed out on a port that has tagged VLAN, this will affect the IEEE 802.1p priority field in the outbound packets.

This is useful in some scenarios when the DSCP is overridden. The priority adjustment function is made to mimic the behaviour of the Layer-2 priority support when configured in the IP ToS/DiffServ mode, as described in chapter 8, section 8.1.4.

Enabling this flag will introduce more work for the CPU inside the WeOS unit for every packet that is modified. As this decreases the maximum routing performance, it should only be enabled when necessary.

31.1.4 Network Address Translation

WeOS supports two kinds of NAT: NAPT (section 31.1.4.1) and 1-to-1 (section 31.1.4.2).

31.1.4.1 NAPT style NAT

NAPT, or ”Network Address and Port Translation” enables hosts on a private network to share an Internet connection with a single public IP address. NAPT is also known as IP Masquerading or PAT (Port Address Translation) in the Cisco world.

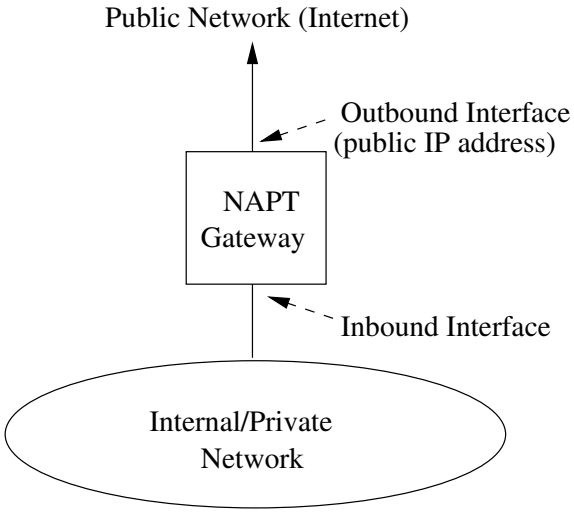


Figure 31.5: NAPT gateway providing access to the Internet. All hosts in the private network share a single public IP address.

When configuring a NAPT rule, you need to specify the *outbound interface*[5]. The appropriate rule will then be added to the *post-routing* step (see fig. 31.1) handling the address translation. A rule is also needed in the *forward filtering* chain to enable the forwarding (routing) of traffic, and that can be added automatically by using the **”addfilter”** option as shown in the example below (here we assume that the interface ”Outbound/Public” side is named ”vlan2”.

Example

```
example:/config/ip/firewall/#>  nat type napt out vlan2 addfilter
```

[5]Appropriate interface IP settings must be configured, and IP routing must also be enabled, see chapter 19.

The resulting firewall allow rule is shown below:

Example

```
example:/#> show firewall
=== Forwarding Packet Filter Rules ========================================
Forwarding Policy DROP
target     prot in     out     source               destination
...
ACCEPT     all  any    vlan2   anywhere             anywhere
...
```

Connection tracking will ensure that packets in the reverse direction (from the Internet to the private network) are accepted and managed properly.

31.1.4.2 1-to-1 style NAT

1-to-1 NAT, also called Full NAT, maps an entire network block in a one-to-one fashion.

31.1.4.2.1 Forward 1-to-1 NAT

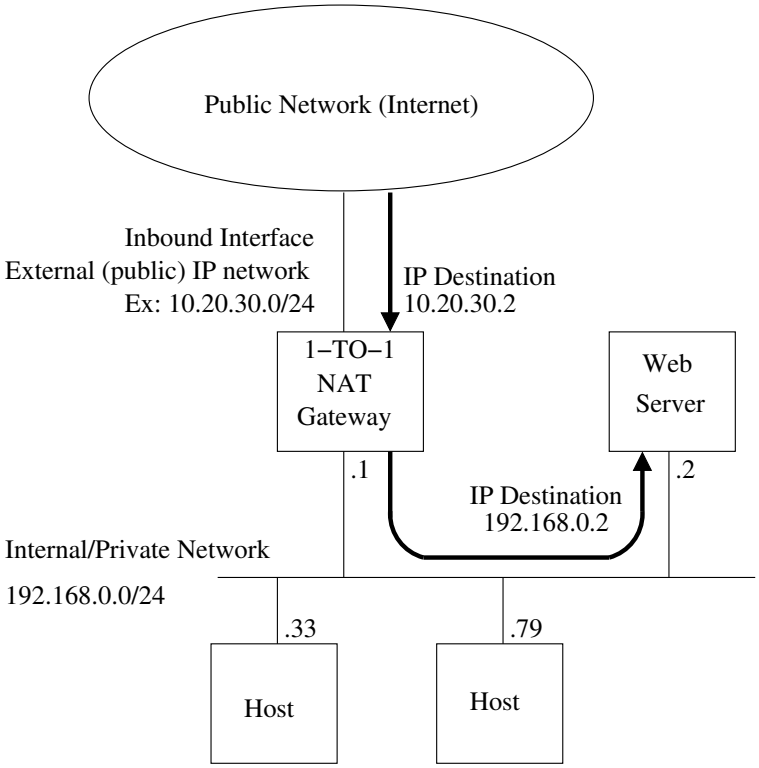


Figure 31.6: 1-to-1 NAT mapping external IP addresses to internal addresses.

A 1-to-1 NAT rule is defined by an inbound interface and two network blocks, the externally (publicly) visible network block and the internal block (typically private IP addresses). IP packets entering the router through the inbound interface targeted to the external network will be transformed so they become targeted to the internal block instead (see fig. 31.6). Packets going to the first IP in the external block will be mapped so they go to the first IP in the internal block, packets to the second external IP to the second internal IP, and so on. This one-to-one mapping requires that the external and internal network blocks are of the exact same size.

1-to-1 NAT mapping is done in the *pre-routing* step in the firewall (see fig. 31.1). This means (for inbound packets affected by a 1-to-1 NAT rule) that the destination IP address is changed to another IP address **before** routing is done and before rules in the *input filtering* and *forward filtering* chains are evaluated. Make sure that you only use the internal network block (called ”new destination” in the web configuration and ”to-dst” in CLI config) in routing and filtering as the external network is not visible inside the unit.

31.1.4.2.2 Reverse 1-to-1 NAT

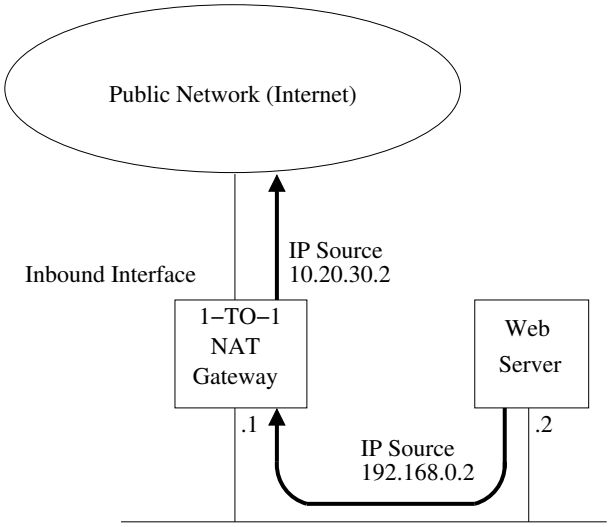


Figure 31.7: Reverse 1-to-1 NAT mapping

1-to-1 NAT is bi-directional which means that the NAT works in the reverse direction too. A request coming from an internal IP will be transformed so it appears to come from the external net when leaving the router through the configured ”inbound” interface (see fig. 31.7).

In this case the translation of the IP source address will be performed in the *post-routing* chain (fig. 31.1), just before packets leave the router. This means that the original internal network IP will be matched as source in any *forward filtering* and *output filtering* rules. The external addresses will not be visible here similar to the forward direction NAT.

31.1.4.2.3 1-1 NAT and implicit firewall rules Consider the sample network setup shown in figs. 31.6 and 31.7. Assuming the ”inbound” interface is named ”vlan2”, then the ”1-to-1” NAT rule could be achieved with the following CLI command.

Example

```
# Example with implicit firewall rule

example:/config/ip/firewall/#> nat type 1-to-1 in vlan2 dst 10.20.30.0/24
to-dst 192.168.2.0/24 addfilter
```

The **”addfilter”** attribute will add implicit firewall rules to allow forward traffic (fig. 31.6) and reverse traffic (fig. 31.7) to automatically pass through the firewall. One rule is created in each direction, as shown below.

Example

```
example:/#> show firewall
...
=== Forwarding Packet Filter Rules ========================================
Forwarding Policy DROP
target     prot in     out     source               destination
...
ACCEPT     all  vlan2  any     anywhere             192.168.2.0/24
ACCEPT     all  any    vlan2   192.168.2.0/24       anywhere
...
```

Using the ”addfilter” makes it easy to get your NAT-traffic through the firewall in either direction. But in cases where there are security concerns, such as when the ”inbound” interface is located on the public Internet, use of the **”addfilter”** option for ”1-to-1 NAT” is too permissive. Instead you could add explicit firewall rules to allow traffic according to your specific requirements. An example is shown below where traffic is only allowed to be *initiated* from the private network (i.e., the ”reverse” direction as shown in fig. 31.7). Note that the **”nat”** command does not include the **”addfilter”** option here.

Example

```
# Example with explicit firewall rule instead of implicit

example:/config/ip/firewall/#> nat type 1-to-1 in vlan2 dst 10.20.30.0/24
to-dst 192.168.2.0/24
example:/config/ip/firewall/#> filter allow out vlan2 src 192.168.2.0/24
```

The resulting firewall rule is shown below.

Example

```
example:/#> show firewall
...
=== Forwarding Packet Filter Rules ========================================
Forwarding Policy DROP
target     prot in     out     source               destination
...
ACCEPT     all  any    vlan2   192.168.2.0/24       anywhere
...
```

31.1.4.2.4 Proxy ARP and 1-to-1 NAT

WeOS 1-to-1 NAT includes a *proxy ARP* mechanism, which makes the WeOS unit answer on ARP requests for the external network specified in the configuration (the **"dst"** parameter in the CLI or **Destination Address(es)** field in the Web interface). The router will only answer on ARP requests originating from the network connected to the inbound interface (CLI: **"in"** parameter, Web: **Incoming Interface**). This makes it possible to use 1-to-1 NAT to pick up traffic to a specific subnet from within a larger network without the need of explicit routing settings.

An example is shown in fig. 31.8: You have a subnet 10.0.0.0/16 set on your external LAN, and want to use 1-to-1 NAT to take care of the specific subnets 10.0.1.0/24, 10.0.2.0/24 and 10.0.3.0/24, which should be translated and routed to the inside of the Router1, Router2 and Router3 respectively. In this case, hosts at the external LAN, such as the management PC (10.0.0.99), will use ARP when they want to reach something within the 10.0.0/16 range. If the PC sends an ARP Request for 10.0.1.33 (PLC3), WeOS Router1 will respond and announce its own MAC address in the ARP reply. Traffic from the management PC (and other hosts on the external network) to 10.0.1.33 (PLC3) will be sent to Router1, which performs 1-to-1 NAT (10.0.1.33⇒192.168.1.33) before forwarding the packets towards PLC3.

Proxy ARP removes the need for explicit routing in some scenarios, but if you are setting up a purely routed configuration, proxy ARP might not be useful, and in some special cases even undesirable. For these special scenarios it is possible to disable Proxy ARP for a 1-to-1 NAT rule. This is done by specifying the CLI keyword **"noarp"** or by un-checking the **Proxy ARP** checkbox in the Web. See sections 31.2.2.2 (Web) and 31.3.5 (CLI) for configuration details.

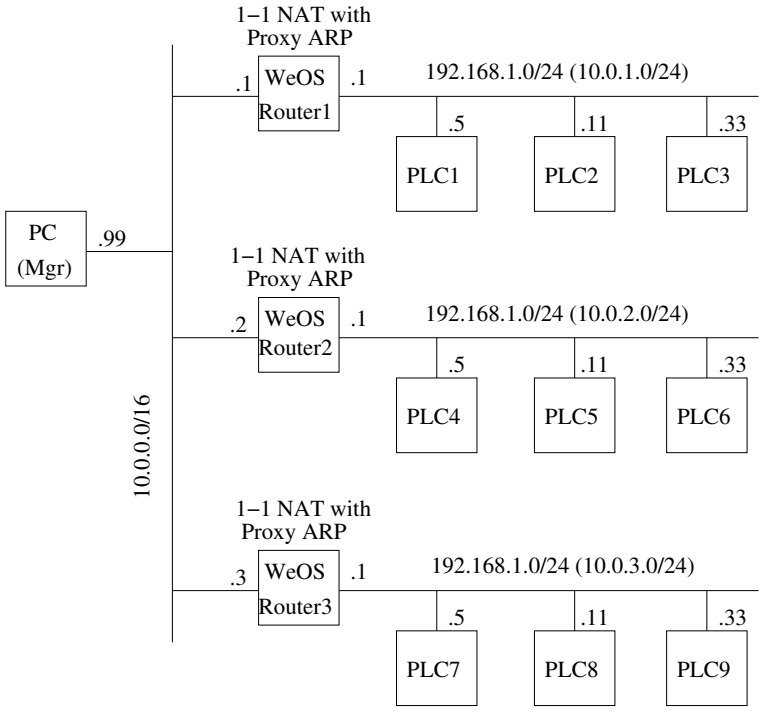


Figure 31.8: Use of proxy ARP with 1-to-1 NAT. The Management PC can reach the PLCs without explicit routes to networks 10.0.1.0/24, 10.0.2.0/24 or 10.0.3.0/24.

31.1.4.3 NAT and IP Multicast

Chapter 29 describes WeOS support for IP multicast routing. Combining NAT and IP multicast routing is **not** generally supported, although there exist some specific use cases which work as of WeOS v4.17.1. Furthermore, when using NAT for IP multicast traffic, the address translation only applies to the source IP address of the multicast packet (the source address is a unicast IP address).

31.1.5 Port Forwarding

Port Forwarding is commonly used together with NAPT, to enable access from the Internet to a server inside the private network. Fig. 31.9 shows a typical setup when *port forwarding* is useful:

- The switch acts as a NAT/NAPT gateway to the Internet: routing is enabled (see section 19.1) and a NAPT rule defining the external (outbound) interface has been configured (see section 31.1.4).
- A Web Server on the ”internal” network serves users on the Internet: A port forwarding rule has been added to allow users on the Internet to initiate connections to the Web server on host 192.168.0.2 (TCP port 80).

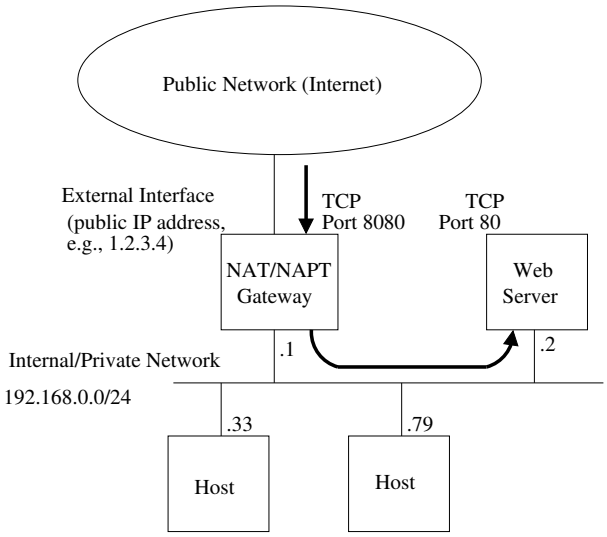


Figure 31.9: Use of port forwarding to enable Internet hosts to access a Web server inside the private network via a NAT/NAPT gateway.

With port forwarding, users on the Internet will connect to the internal Web Server as if it was running on the NAT/NAPT gateway, i.e., users on the Internet will connect to the Web server using the public IP address (here 1.2.3.4) and TCP port number (here 8080), without knowing that the traffic is forwarded to a server inside the internal network.

Configuration of port forwarding rules include the following parameters:

- *Inbound Interface*: Packets which are subject to port forwarding should come in on the specified interface. In the example network shown in fig. 31.9, this would be the *external interface*, i.e., the attached to the Internet.
- *Inbound Port (Range)*: Defines the range of TCP/UDP port numbers, which are to be mapped by this rule. In the example in fig. 31.9 Internet hosts would reach the Web server using TCP port 8080.
- *Source IP Address/Subnet*: Optional argument limiting the port forwarding rule to concern a limited set of Internet hosts.
- *Destination IP Address*: Specifies the IP address of the private server, i.e., where packets are to be sent. The Web server in in fig. 31.9 has IP address *192.168.0.2*.
- *Destination Port (Range)* Specifies which TCP/UDP port number(s) to use on the in the forwarded packet. The default is to use the same port number(s) as on the inbound interface. In the example, the Web server on the internal server uses TCP port 80. Note that only single port forwards can change the destination port so that it is different from the original inbound port. Forwarding of a range of ports always keep the port numbers. Multiple single port forwarding rules can be used to form a range in case the destination port numbers must be changed.
- *Transport Protocol (TCP/UDP)*: Specify if this rule applies to TCP, UDP or both. In the example, the rule applies only to TCP.

31.1.6 Firewall Logging

The WeOS firewall supports logging for monitoring and debugging purposes.

Firewall logging is done to the kernel log file **kern.log**, and to a remote syslog if configured. Internal system information will also be written to this file during (re)boot of the system, and some configuration changes may also add information to this log.

This log file can be viewed from the web interface via the **”View Log”** function under the menu: **”Maintenance”**. It can also be viewed in the CLI with the command **”show log://kern.log”**. For more information about log files and configuration of remote syslog, please see chapter 25.

Details about configuration options can be found in section 31.2 (Web), and section 31.3 (CLI).

31.1.6.1 Enabling logging for firewall rules

Logging is enabled for individual rules in the firewall.

Logging is possible for packet filtering rules (both allow and deny), for NAT rules (both NAPT and 1-to-1 types) and for port forwarding rules.

Logging is currently not possible for the packet modify operation, however traffic that is modified by packet modify rules is also passing through the forward filtering chain (see fig. 31.1). It is possible to simulate logging for packet modify by adding a filter allow rule in the forward chain with the same matching condition as the modify rule, and enable logging for that filtering rule.

An entry is added to the log file when an IP packet hits a specific rule with logging enabled. Note that **only the first packet in a connection will be logged**. Subsequent packets or return traffic packets belonging to the same session will not be logged (that would quickly overflow the logs).

Logging enabled for packet filter “deny” rules behave different though, and EVERY packet hitting such a rule will be logged.

31.1.6.2 Settings for rate limitation

The firewall logging system has a rate limitation functionality, preventing excessive amount of log entries to be created upon problems. This will reduce problems due to malicious traffic from outside or inside the network, so called “denial of service” attacks (or DOS attacks), port scannings or similar. It will also avoid problems by excessive logging caused by bad configuration or malfunctioning units in the network causing traffic storms.

The limitation is configured as a maximum rate of log entries per time unit. The time units available are: second, minute, hour or day.

The configuration: “10 per second”, means just that, max 10 log entries will be written to the log file each second.

The rate is continous. This means that the allowance of log entries will be evenly distributed over the time unit. An example: “60 per hour” will allow 60 entries per hour, but distributed evenly as max one log entry per minute.

This makes a rate of “1 per second” to be exactly the same as “60 per minute” and “3600 per hour” (also “86400 per day”, but that can not be configured as the biggest permitted value for any unit is 10000)

It is **not** possible to set non-continous rates like: 100 entries per calendar day etc.

If the rate limit of log entries is reached, the logging system **will instantly begin throwing away excessive log entries**. The logging will not be buffered or delivered later to the log file.

The rate limitation can be disabled through configuration, but this will open up for potential problems with malicious attacks or storms, therefore **it is not recommended that you disable the limitation mechanism**. But you can and should adjust the limit to fit your needs.

Firewall logging can also be disabled on a system level. Nothing will be logged even if there is logging configured for individual firewall rules.

The default rate limitation will be set to “5 per second” when the firewall is enabled through the web or CLI.

31.1.6.3 Firewall log format

WeOS uses the Linux Netfilter logging mechanism. The standard Netfilter log format is used for recorded entries.

Log entries will be prefixed with the type of rule that was hit, and will always be one of: FW-ALLOW, FW-DENY, FW-NAPT, FW-1TO1 or FW-PF (port forwarding).

Remember that the kernel log is shared with other types of logging. The prefixes are a good way to find the relevant log entries in the file.

You will not see exactly which firewall rule that triggered a log entry, only the type of it. This can be a problem if you use many rules with logging enabled. However, the information provided in the log should be enough to figure out what specific rule was causing it.

A rule position number or some other helping reference to the specific rule may be added in a later release of WeOS.

Here is an example of a kernel log entry generated when a filter ALLOW rule is hit:

Jan 15 14:44:49 example kernel: FW-ALLOW: IN=vlan1 OUT=vlan2 MAC=00:07:7c:10:de:c1:00:80:c8:3c:25:b7:08:00:45:00:00:54:c9:84 SRC=192.168.2.10 DST=192.168.3.100 LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=51588 DF PROTO=ICMP TYPE=8 CODE=0 ID=10941 SEQ=1

The same log entry line broken down in parts:

Log text part	**Explanation**
Jan 15 14:44:49	Timestamp
example	The system host name
kernel:	Identifies origin, kernel.log
FW-ALLOW:	This originates from a firewall filter “allow” rule
IN=vlan1	Inbound interface “vlan1”, may be empty for NAT rules
OUT=vlan2	Outbound interface “vlan2”
MAC= 00:07:7c:10:de:c1: 00:80:c8:3c:25:b7: 08:00: 45:00:00:54:c9:84	This is the first part from the ethernet packet (this field may be empty for some rules) The first part is destination MAC address This part is the source MAC address Ethertype, 08:00 is IP More data, first part of the IP header
SRC=192.168.2.10 DST=192.168.3.100	Source IP address, always the original IP before any NAT transformation Destination IP address, before NAT
LEN=84 TOS=0x00 PREC=0x00 TTL=63 ID=51588 DF	Packet length and other IP header options
PROTO=ICMP	The IP protocol
TYPE=8 CODE=0 ID=10941 SEQ=1	The rest is protocol specific data and flags, in this specific case an ICMP ping request

Here are example entries for the other types:

Jan 15 12:45:25 example kernel: FW-NAPT: IN= OUT=vlan1 SRC=192.168.2.200 DST=192.168.2.10 LEN=94 TOS=0x00 PREC=0x00 TTL=64 ID=59200 DF PROTO=UDP SPT=514 DPT=514 LEN=74

Jan 15 14:45:12 example kernel: FW-1TO1: IN=vlan1 OUT= MAC=00:07:7c:10:de:c1:00:80:c8:3c:25:b7:08:00:45:00:00:3c:bd:4b SRC=192.168.2.10 DST=192.168.2.100 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=48459 DF PROTO=TCP SPT=55301 DPT=80 WINDOW=14600 RES=0x00 SYN URGP=0

Jan 15 14:45:29 example kernel: FW-PF: IN=vlan1 OUT= MAC=00:07:7c:10:de:c1:00:80:c8:3c:25:b7:08:00:45:00:00:3c:ca:59 SRC=192.168.2.10 DST=192.168.2.200 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=51801 DF PROTO=TCP SPT=55631 DPT=8080 WINDOW=14600 RES=0x00 SYN URGP=0

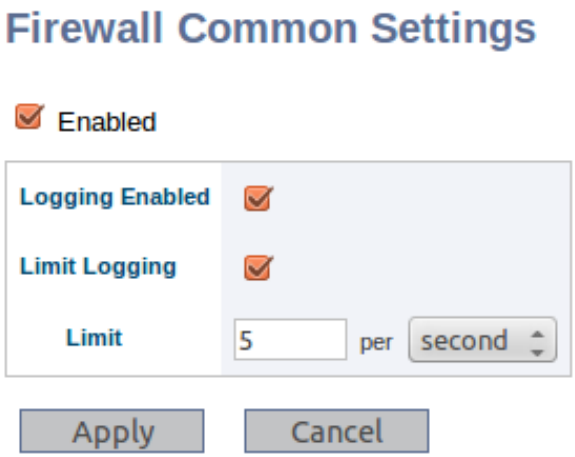
Jan 15 14:49:16 example kernel: FW-DENY: IN=vlan1 OUT= MAC=00:07:7c:10:de:c1:00:80:c8:3c:25:b7:08:00:45:00:00:1c:4a:ca SRC=192.168.2.10 DST=192.168.2.200 LEN=28 TOS=0x00 PREC=0x00 TTL=64 ID=19146 PROTO=UDP SPT=2702 DPT=2000 LEN=8

31.2 Firewall Management via the Web Interface

Menu path: Configuration ⇒ Firewall ⇒ Common

On the firewall common settings page you may enable or disable the firewall.

When disabling the firewall all rules will be lost. A confirmation is required if you try to disable the firewall to not loose rules by accident.

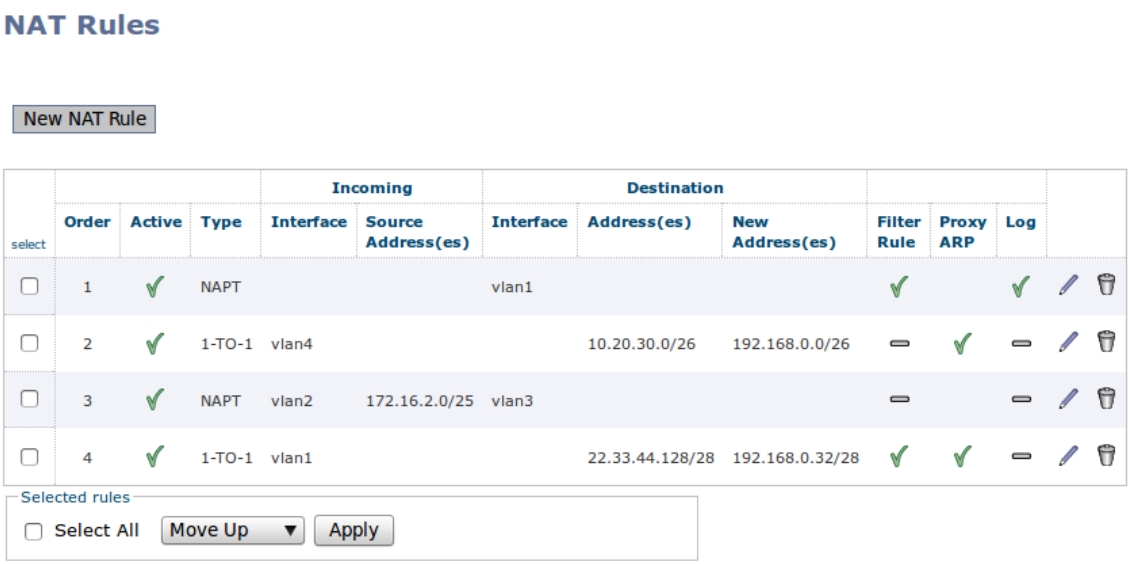


Enabled	Check this box to enable firewall functionality. **Note:** When disabling the firewall, the firewall is stopped and all existing *NAT* rules, *Port Forwarding* rules, *Packet Filter* rules and *Packet Modify* rules are deleted.
Logging Enabled	Check to enable logging for the firewall. This is a master control enabling the logging feature. **Note:** you also need to enable logging on individual firewall rules for anything to be logged.
Limit Logging	Check to enable rate limitation of the logging. The limit is set in the input boxes below. **Warning:** Disabling the limitation may lead to lots of data being logged. This can in a short time fill up the log files.
Limit	Set the threshold rate value and time unit for the limitation. See section 31.1.6 for information about how the limitation operates.

31.2.1 NAT Rules

Menu path: Configuration ⇒ Firewall ⇒ NAT

On the Firewall NAT configuration page you are presented to the list of current NAT rules. (If the firewall function is disabled or no rules have been created you will not see any list, but be presented to an information message.)



New Nat Rule	Click this button to create a new NAT rule. You will be presented to a form where you can configure the new rule.
Select	Check this box to select one or a set of rules for group rule management. Check the *Select all* box at the bottom of the page to select all rules.
Order	The order in which the rules will be applied. When using a JavaScript enabled browser, it is possible to select one or more rules and perform an action on multiple rules, see below. If not using a JavaScript enabled browser, there will be a set of arrows available to move rules up or down to change the order of application.
Active	A green check-mark means the rule is active, and a dash means it is inactive.

Continued on next page

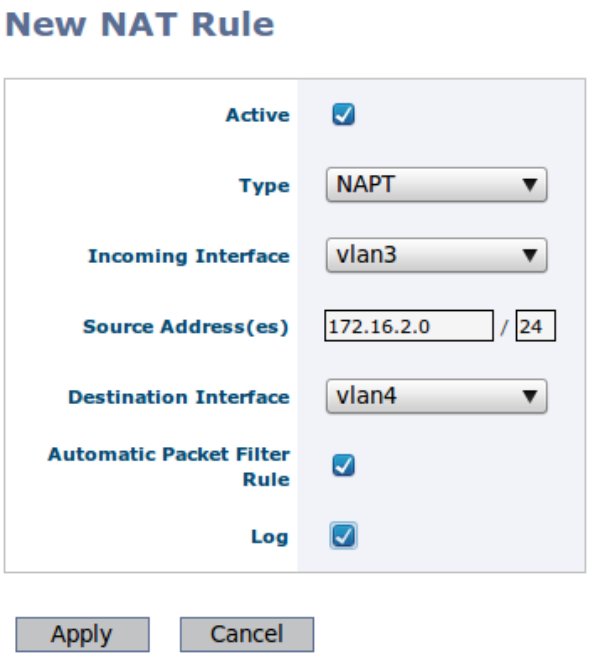
	Continued from previous page
Type	The NAT type for this rule: NAPT or 1-TO-1
Incoming Interface	The inbound interface for packets that should be NATed
Source Address(es)	The IP address and subnet mask (CIDR) for matching the source address of packets
Destination Interface	The outbound interface.
Destination Address(es)	The IP address and subnet mask (CIDR) for matching the destination address of packets
New Address(es)	The target IP address and subnet mask (CIDR) for 1-TO-1 NAT
Filter Rule	If automatic forwarding filter rules are created for this rule. A green check-mark means yes and a dash means no.
Proxy ARP	If Proxy ARP is enabled for a 1-to-1 NAT rule. A green check-mark means yes and a dash means no.
Log	Controls if a match on this rule should be logged in the kernel log file. Nothing will be logged unless logging is also enabled under the common firewall settings.
Edit	Click this icon to edit a NAT rule.
Delete	Click this icon to remove a NAT rule. You will be asked to acknowledge the removal before it is actually executed.
Selected Rules	Selected rules may be modified by selecting the rules to modify and select the modification action in the drop-down list and then click the **Apply** button.

31.2.2 New NAT Rule

Menu path: Configuration ⇒ Firewall ⇒ NAT ⇒ **New NAT Rule**

In the **New NAT Rule** configuration page you can specify a new NAT rule. This page exists in two views depending on what NAT type you want to create. When you enter this page initially, the ”NAPT” type is pre-selected. Change the type to ”1-TO-1” to see the other view. If you have disabled JavaScript you will only see one view with all fields from both NAPT and 1-TO-1 together.

31.2.2.1 New NAT Rule - NAPT view

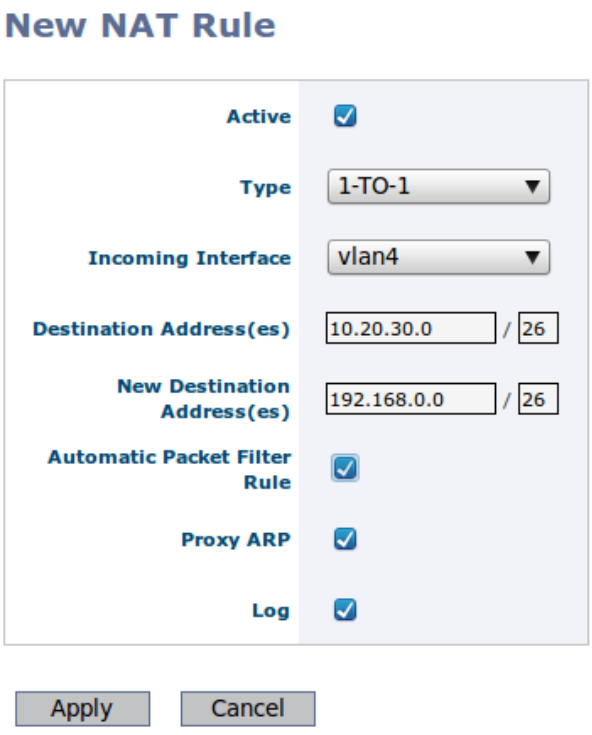


Active	Rule is active if checked.
Type	NAPT. If you change to 1-TO-1 NAT, the view will change. See section 31.2.2.2.
Incoming Interface	Optional. The interface connected to your subnet whose addresses you want to translate (the interface to your internal/private network).
Source Address(es)	Optional. The IP address and subnet mask (CIDR) identifying the IP subnet where this NAT rule should be applied.
Destination Interface	Mandatory. The interface that should represent all IP addresses on the subnet of the **internal interface**. This is the external/public interface, typically the interface connected to the Internet.

Continued on next page

	Continued from previous page
Automatic Packet Filter Rule	Keep as checked if you want an automatically created rule in the firewall *forwarding filter* allowing packets that matches this NAT rule. This rule is invisible in the filter configuration. Uncheck it if you want to set up your own rules for controlling traffic.
Log	Controls if a match on this rule should be logged in the kernel log file. Nothing will be logged unless logging is also enabled under the common firewall settings.

31.2.2.2 New NAT Rule - 1-TO-1 NAT view



Active	Rule is active if checked.
Type	1-TO-1. If you change to NAPT, the view will change. See section 31.2.2.1.

Continued on next page

	Continued from previous page
Incoming Interface	Mandatory. The inbound interface where traffic arrives to the router
Destination Address(es)	Mandatory. The original external IP address and subnet mask (CIDR) that should be NATed
New Destination Address(es)	Mandatory. The new internal IP address and subnet mask (CIDR) set by the NAT
Automatic Packet Filter Rule	Check if you want automatically created rules in the firewall *forwarding filter* allowing packets that matches this NAT rule. Rules will be created for both forward direction and for the reverse direction. Keep unchecked if you want to set up your own rules for controlling traffic.
Proxy ARP	Check to enable ARP proxying for the *Destination Address(es)* on the *Incoming Interface*. You should have this enabled in most cases.
Log	Controls if a match on this rule should be logged in the kernel log file. Nothing will be logged unless logging is also enabled under the common firewall settings.

31.2.3 Edit NAT Rule

Menu path: Configuration ⇒ Firewall ⇒ NAT ⇒ 

In the **`Edit NAT Rule`** configuration page you can change an existing NAT rule.

See section 31.2.2 for description of editable fields.

31.2.4 Port Forwarding Rules

Menu path: Configuration ⇒ Firewall ⇒ Port Forwarding

Port forwarding is e.g. used to give external units access to specific services in a subnet hidden by NAT/NAPT. If the firewall is disabled or no rules have been created you will see no list, but be presented to an information message.

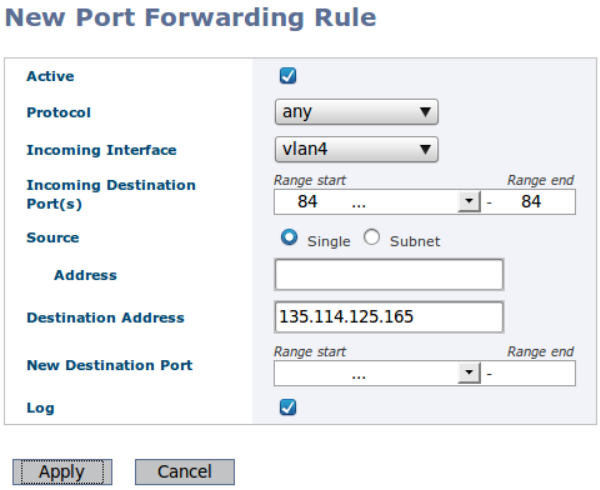
New Forwarding Rule	Click this button to create a new port forwarding rule. You will be presented to a form where you can configure the new rule.
Select	Check this box to select one or a set of rules for group rule management. Check the *Select all* box at the bottom of the page to select all rules.
Order	The order in which the rules will be applied. When using a JavaScript enabled browser, it is possible to select one or more rules and perform an action on multiple rules, see below. If not using a JavaScript enabled browser, there will be a set of arrows available to move rules up or down to change the order of application.
Active	A green check-mark means the rule is active, and a dash means it is inactive.

Continued on next page

	Continued from previous page
Protocol	Traffic may be filtered on transport layer protocol. Available are TCP and UDP.
Incoming Interface	The interface from which inbound traffic should be allowed.
Incoming Destination Port	The range of transport layer ports to match. E.g. 80 for standard web-server access.
Incoming Source Address(es)	Optional. The source IP address(es) of packets allowed to be forwarded. Either a single address, or a subnet. Subnet mask is displayed in CIDR notation (prefix length).
Destination Address	The destination IP address to which the packets will be forwarded.
Destination New Port	If another port or set of ports are used by the destination host for the service you can map the port(s) by entering another port or set of ports. Number of ports must match the number of incoming destination ports. Empty means that the incoming destination port will be used. **Note:** New destination port can only be set for single ports. Multi-port ranges can not be remapped to a new port range. You must use multiple single-port mappings to achieve this.
Log	Controls if a match on this rule should be logged in the kernel log file. Nothing will be logged unless logging is also enabled under the common firewall settings.
Edit	Click this icon to edit a port forwarding rule.
Delete	Click this icon to remove a port forwarding rule. You will be asked to acknowledge the removal before it is actually executed.
Selected Rules	Selected rules may be modified by selecting the rules to modify and select the modification action in the drop-down list and then click the **Apply** button.

31.2.5 New Port Forwarding Rule

Menu path: Configuration ⇒ Firewall ⇒ Port Forwarding ⇒ **`New Forwarding Rule`**



Active	Rule is active if checked.
Protocol	Mandatory. Traffic may be filtered on transport layer protocol. Available are TCP and UDP. Choose *any* to allow both TCP and UDP packets.
Incoming Interface	Mandatory. The interface from which inbound traffic should be allowed.
Incoming Destination Port(s)	Mandatory. The range of transport layer ports to match. E.g. 80 for standard web-server access. If JavaScript is enabled, the range start may be selected in the drop down.
Source	Optional. The source IP address(es) of packets allowed to be forwarded. Either a single address, or a subnet. If single is selected, enter a single address. If subnet is selected a netmask (e.g. 255.255.255.0) must also be entered to define the subnet. If you have a JavaScript enabled browser the netmask field will not be displayed unless you check the subnet radio button.
Destination Address	Mandatory. The destination IP address to which the packets will be forwarded.

Continued on next page

	Continued from previous page
New Destination Port	Optional. If another port or set of ports are used by the destination host for the service you can map the port(s) by entering another port or set of ports. Number of ports must match the number of incoming destination ports. Empty means that the incoming destination port will be used. **Note:** New destination port can only be set for single ports. Multi-port ranges can not be remapped to a new port range. You must use multiple single-port mappings to achieve this. If JavaScript is enabled, the range start may be selected in the drop down.
Log	Controls if a match on this rule should be logged in the kernel log file. Nothing will be logged unless logging is also enabled under the common firewall settings.

31.2.6 Edit Port Forwarding Rule

Menu path: Configuration ⇒ Firewall ⇒ Port Forwarding ⇒ 

In the **`Edit Port Forwarding Rule`** configuration page you can change an existing port forwarding rule.

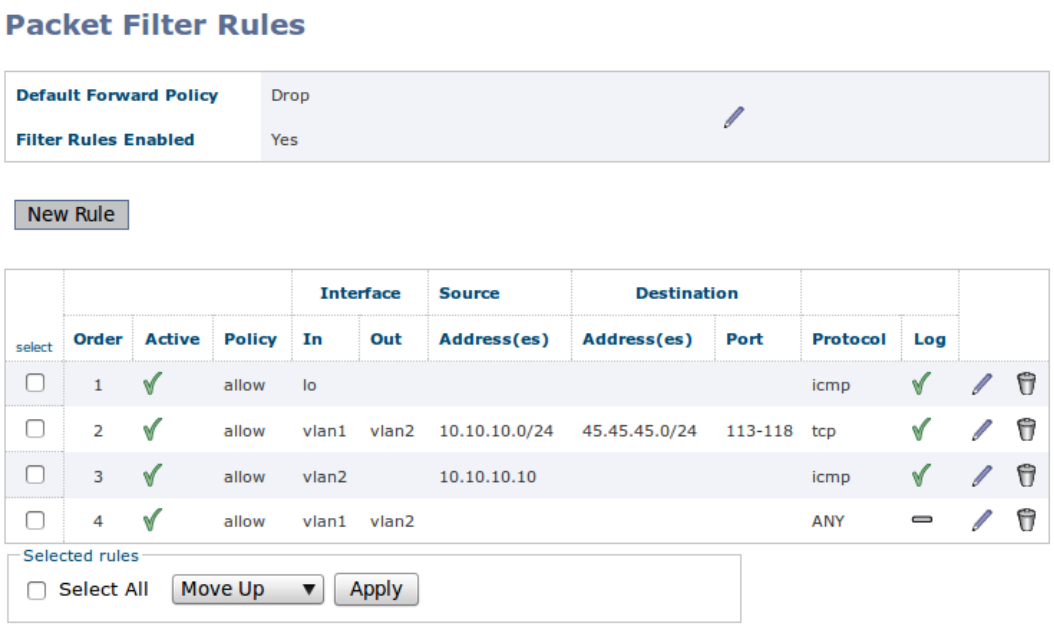
See section 31.2.5 for description of editable fields.

31.2.7 Packet Filter Rules

Menu path: Configuration ⇒ Firewall ⇒ Packet Filter

Packet filter rules are set up to allow traffic to pass through the firewall. Traffic is by default denied, except for a set of default allow rules created.

If the firewall is disabled or no rules have been created you will see no list, but be presented to an information message.



Default Forward Policy	The policy defines how to handle data for which no matching rule can be found. The forward chain controls traffic passing through the switch, not traffic destined to the switch itself. Possible values are: **Allow** Packets will be allowed through. **Drop** Packets will be dropped and no other actions are taken.
Filter Rules Enabled	Yes means rules are active. No means rules are deactivated and all traffic is allowed through. Individual deactivation of rules override when this setting is yes (active).
Edit	Click this icon to edit the global settings.

Continued on next page

Continued from previous page

New Rule	Click this button to create a new packet filter rule. You will be presented to a form where you can configure the new rule.
Select	Check this box to select one or a set of rules for group rule management. Check the *Select all* box at the bottom of the page to select all rules.
Order	The order in which the rules will be applied. When using a JavaScript enabled browser, it is possible to select one or more rules and perform an action on multiple rules, see below. If not using a JavaScript enabled browser, there will be a set of arrows available to move rules up or down to change the order of application.
Active	A green check-mark means the rule is active, and a dash means it is inactive.
Policy	The type of rule, *Allow* or *Deny*.
In Interface	The rule will be applied to traffic entering on this interface.
Out Interface	The rule will be applied to traffic exiting on this interface. If neither *Out Interface* nor *Destination Address* (see below) are specified, the rule will apply to the INPUT chain, i.e., traffic destined to the switch itself (ICMP pings, SSH management, etc.).
Source Address(es)	The rule will be applied to traffic originating from a source with this specific IP-address or an IP-address in the specified subnet.
Destination Address(es)	The rule will be applied to traffic destined to this specific IP-address or to an IP-address in the specified subnet. If neither *Out Interface* (see above) nor *Destination Address* are specified, the rule will apply to the INPUT chain, i.e., traffic destined to the switch itself (ICMP pings, SSH management, etc.).
Destination Port	The rule will be applied to traffic destined to this set of (UDP/TCP) ports.
Protocol	The rule will be applied to traffic using this protocol. Select the protocol name or enter the protocol number. If *ANY* the rule will be applied for all protocol types.

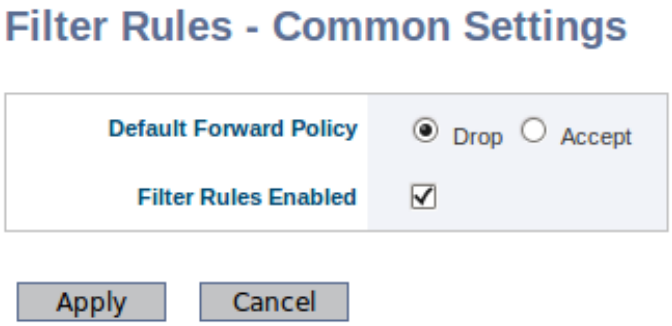
Continued on next page

	Continued from previous page
Log	Controls if a match on this rule should be logged in the kernel log file. Nothing will be logged unless logging is also enabled under the common firewall settings.
Edit	Click this icon to edit a packet filter rule.
Delete	Click this icon to remove a packet rule. You will be asked to acknowledge the removal before it is actually executed.
Selected Rules	Selected rules may be modified by selecting the rules to modify and select the modification action in the drop-down list and then click the **Apply** button.

31.2.8 Edit Common Packet Filter Settings

Menu path: Configuration ⇒ Firewall ⇒ Packet Filter ⇒ (Common Settings)

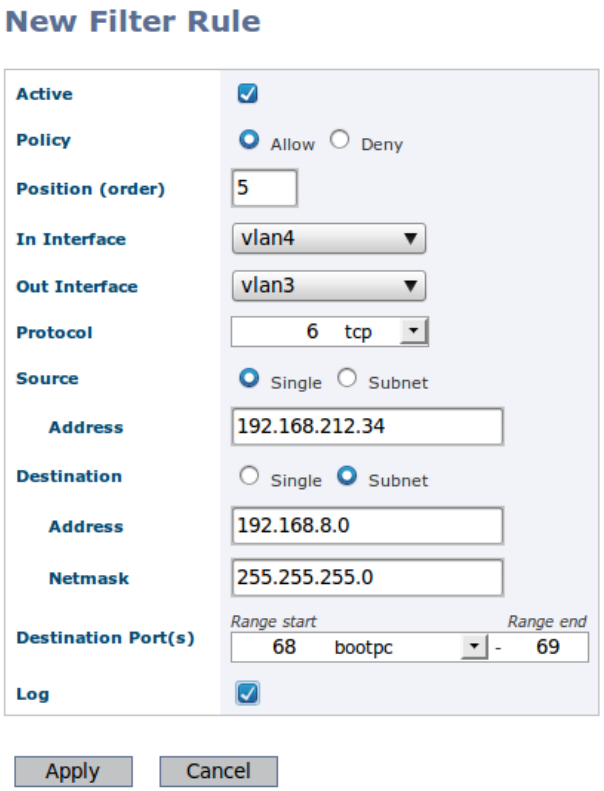
Here you may change the common settings for the packet filter rules.



Default Forward Policy	The policy defines how to handle data for which no matching rule can be found. The forward chain controls traffic passing through the switch, not traffic destined to the switch itself. Possible values are: **Allow** Packets will be allowed through. **Drop** Packets will be dropped and no other actions are taken. Select the policy by clicking the radio button.
Filter Rules Enabled	Check the box to activate the rules, or uncheck to deactivate the rules. Deactivation means all traffic is allowed through (policy is changed to *allow*).

31.2.9 New Packet Filter Rule

Menu path: Configuration ⇒ Firewall ⇒ Packet Filter ⇒ **`New Rule`**



Active	Rule is active if checked.
Policy	Choose Allow/Deny to select if this rule should allow or deny traffic.
Position (order)	The position in the list defining in what order rules will be applied. Defaults to last position. Change the value to insert this rule in another position.
In Interface	The rule will be applied to traffic entering on this interface.

Continued on next page

	Continued from previous page
Out Interface	The rule will be applied to traffic exiting on this interface. If neither *Out Interface* nor *Destination Address* (see below) are specified, the rule will apply to the INPUT chain, i.e., traffic destined to the switch itself (ICMP pings, SSH management, etc.).
Protocol	The rule will be applied to traffic using this protocol. Select IP protocol in drop-down or enter the protocol number to specify for which protocol to apply this rule (see also *Destination Port* option below). Select *any* to allow traffic from any IP Protocol (ICMP, TCP, UDP, . . .) through.
Source Address(es)	The rule will be applied to traffic originating from a source with this specific IP-address or an IP-address in the specified subnet. Select *Single* and enter the single source address into the address field. Select *Subnet* and enter an address into the address field and a subnet mask into the *Netmask* field.
Destination Address(es)	The rule will be applied to traffic destined to this specific IP-address or to an IP-address in the specified subnet. Select *Single* and enter the single source address into the address field. Select *Subnet* and enter an address into the address field and a subnet mask into the *Netmask* field. If neither *Out Interface* (see above) nor *Destination Address* are specified, the rule will apply to the INPUT chain, i.e., traffic destined to the switch itself (ICMP pings, SSH management, etc.).
Destination Port	The rule will be applied to traffic destined to this set of (UDP/TCP) ports. If JavaScript is enabled, the range start may be selected in the drop down. Only valid if *Protocol* TCP or UDP has been selected (see above).
Log	Controls if a match on this rule should be logged in the kernel log file. Nothing will be logged unless logging is also enabled under the common firewall settings. **Note:** Logging differs in behavior between policy *Accept* and *Deny*. See section 31.1.6 for more details.

31.2.10 Edit Packet Filter Rule

Menu path: Configuration ⇒ Firewall ⇒ Filter ⇒ 

In the **`Edit Packet Filter Rule`** configuration page you can change an existing packet filter rule.

See section 31.2.9 for description of editable fields.

31.2.11 Packet Modify Rules

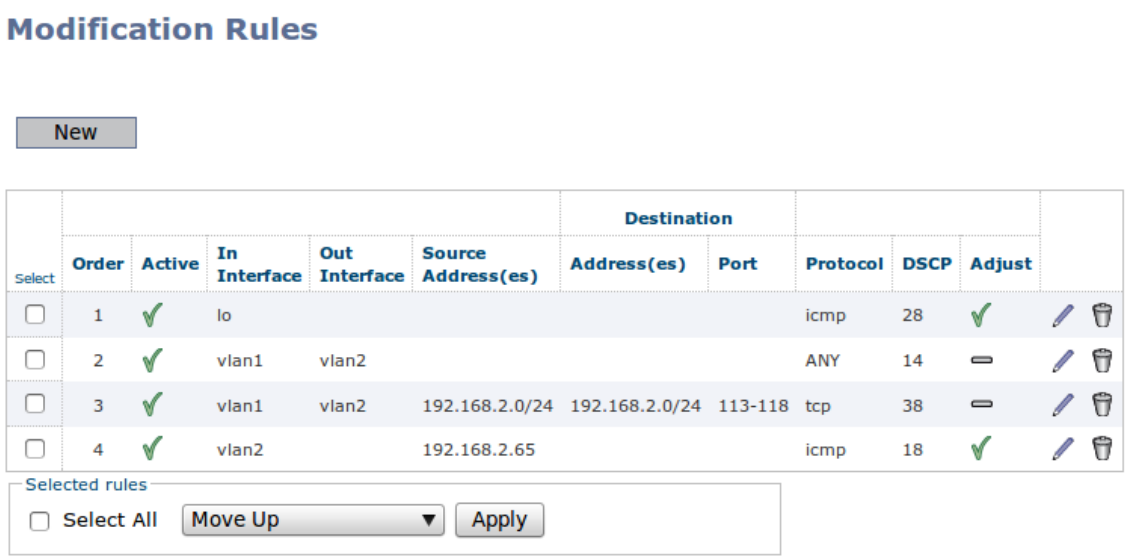
Menu path: Configuration ⇒ Firewall ⇒ Modify

Modify rules are set up to change the priority of packets passing through the firewall.

- Rules are evaluated in the listed order from the top and downwards.
- Rules are only used if the configured parameters match.
- A matching rule will result in the DSCP field in the packets being changed to the configured value, and the next rule is then evaluated. The final value will thus be from the last matching rule.

Optionally the *adjust priority* adjusts the (internal) priority handling of the packet inline with to the new DSCP value. In addition, the VLAN tag priority will be set accordingly if the packet egresses the switch tagged.

If the firewall is disabled or no rules have been created you will see no list, but be presented to an information message.



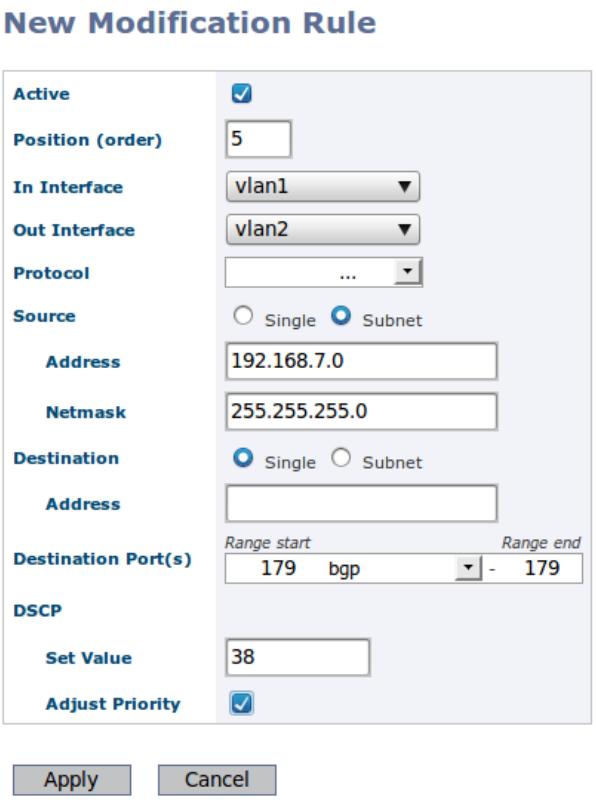
New	Click this button to create a new modify rule. You will be presented to a form where you can configure the new rule.

Continued on next page

	Continued from previous page
Order	The order in which the rules will be applied. When using a JavaScript enabled browser, it is possible to select one or more rules and perform an action on multiple rules, see below. If not using a JavaScript enabled browser, there will be a set of arrows available to move rules up or down to change the order of application.
Active	A green check-mark means the rule is active, and a dash means it is inactive.
In Interface	The rule will be applied to traffic entering on this interface.
Out Interface	The rule will be applied to traffic exiting on this interface.
Source Address(es)	The rule will be applied to traffic originating from a source with this specific IP-address or an IP-address in the specified subnet.
Destination Address(es)	The rule will be applied to traffic destined to this specific IP-address or to an IP-address in the specified subnet.
Destination Port	The rule will be applied to traffic destined to this set of (UDP/TCP) ports.
Protocol	The rule will be applied to traffic using this protocol. Select the protocol name or enter the protocol number. If *ANY* the rule will be applied for all protocol types.
DSCP	The DSCP value to be set for packets matching this rule.
Adjust	Indicates if the modified DSCP value should be used for switch internal prioritising and applied to VLAN-priority on tagged packets. A green check-mark means yes and a dash means no.
Edit	Click this icon to edit a modify rule.
Delete	Click this icon to remove a modify rule. You will be asked to acknowledge the removal before it is actually executed.
Selected Rules	Selected rules may be modified by selecting the rules to modify and select the modification action in the drop-down list and then click the **`Apply`** button.

31.2.12 New Modify Rule

Menu path: Configuration ⇒ Firewall ⇒ Modify ⇒ **New**



Active	Rule is active if checked.
Position (order)	The position in the list defining in what order rules will be applied. Defaults to last position. Change the value to insert this rule in another position.
In Interface	The rule will be applied to traffic entering on this interface.
Out Interface	The rule will be applied to traffic exiting on this interface.

Continued on next page

	Continued from previous page
Protocol	The rule will be applied to traffic using this protocol. Select IP protocol in drop-down or enter the protocol number to specify for which protocol to match with this rule (see also *Destination Port* option below). Select *any* to match any IP Protocol (ICMP, TCP, UDP, . . .).
Source Address(es)	The rule will be applied to traffic originating from a source with this specific IP-address or an IP-address in the specified subnet. Select *Single* and enter the single source address into the address field. Select *Subnet* and enter an address into the address field and a subnet mask into the *Netmask* field.
Destination Address(es)	The rule will be applied to traffic destined to this specific IP-address or to an IP-address in the specified subnet. Select *Single* and enter the single source address into the address field. Select *Subnet* and enter an address into the address field and a subnet mask into the *Netmask* field.
Destination Port	The rule will be applied to traffic destined to this set of (UDP/TCP) ports. If JavaScript is enabled, the range start may be selected in the drop down. Only valid if *Protocol* TCP or UDP has been selected (see above).
DSCP - Set Value	The DSCP value to be set for packets matching this rule. Valid values 0-63.
DSCP Adjust Priority	Indicates if the modified DSCP value should be used for switch internal prioritising and applied to VLAN-priority on tagged packets. Check to enable.

31.2.13 Edit Modify Rule

Menu path: Configuration ⇒ Firewall ⇒ Modify ⇒ 

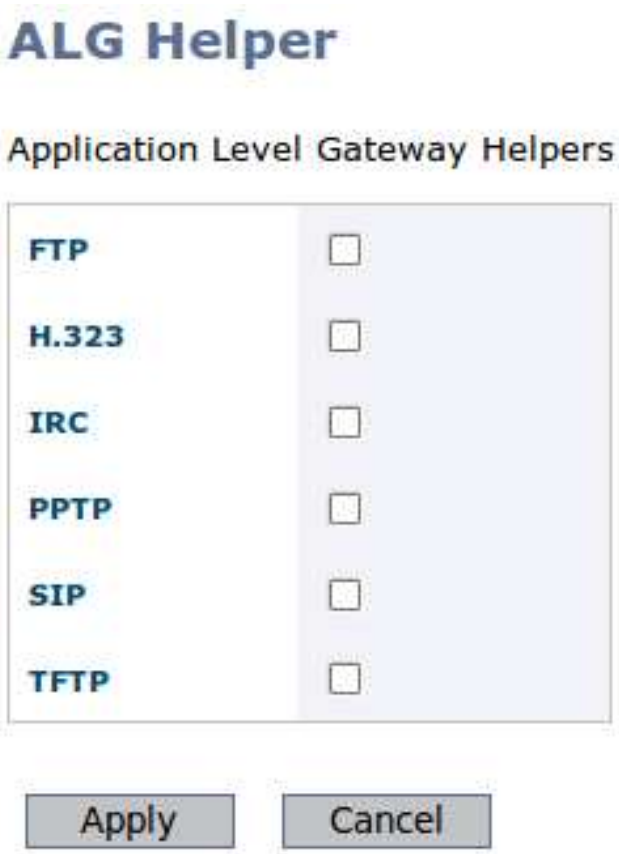
In the **`Edit Modification Rule`** configuration page you can change an existing modify rule.
It is also possible to move the rule to a certain position in the list by changing the *Position (order)* field. The rule will be inserted on requested position and the rule currently on the position will be shifted down.

See section 31.2.12 for description of editable fields.

31.2.14 Configure ALG Helpers

Menu path: Configuration ⇒ Firewall ⇒ ALG Helper

In the **ALG Helper** configuration page you can activate Application Level Gateway (ALG) Helpers in the firewall.



Check the box for the ALG helper to activate.

See section 31.1.1 for description of ALG helpers.

31.3 Firewall Management via the CLI

Command	Default	Section
Configure Firewall Settings		
[no] firewall	Disabled	Section 31.3.1
[no] enable	Enabled	Section 31.3.2
[no] filter [pos <NUM>] <allow|deny> [in <IFNAME>] [out <IFNAME>] [src <ADDR[/LEN]>] [dst <ADDR[/LEN]>] [dport <RANGE>] [proto <NAME|NUM>] [passive] [log]		Section 31.3.3
[no] modify [pos <NUM>] [match [in <IFNAME>] [out <IFNAME>] [src <ADDR[/LEN]>] [dst <ADDR[/LEN]>] [proto <NAME|NUM>] [dport <RANGE>]] set dscp <NUM> [adjust-prio] [passive]		Section 31.3.4
[no] nat [<NUM>] type <NAPT|1-TO-1> [in <IFNAME>] [out <IFNAME>] [src <ADDR[/LEN]>] [dst <ADDR[/LEN]>] [to-dst <ADDR[/LEN]>] [addfilter] [noarp] [passive] [log]		Section 31.3.5
[no] port-forward in <IFNAME>:<PORTRANGE> [src <ADDR/LEN>] dst <ADDR>[:PORTRANGE] [proto <tcp|udp>] [passive] [log]		Section 31.3.6
[no] alg <ftp|tftp|sip|irc|h323|pptp>	Disabled	Section 31.3.7
[no] spi	Disabled	Section 31.3.8
policy [forward|input] <deny|allow>	Deny	Section 31.3.9
move [filter|modify|nat|port-forward] <FROM> <TO>		Section 31.3.10
[no] passive [filter|modify|nat|port-forward] <POS>		Section 31.3.11
[no] log limit (none | <entries>/(second|minute|hour|day))		Section 31.3.12
[no] log [filter|nat|port-forward] <POS>		Section 31.3.12
View Firewall Status		
show firewall		Section 31.3.13

31.3.1 Managing the Firewall

Syntax `[no] firewall`

Context IP Configuration context

Usage Enter the Firewall Configuration context. This will enable the firewall (unless it is already enabled).

Use **”no firewall”** to disable the firewall, and to delete all existing *NAT*, *Port Forwarding*, *Packet filter (allow/deny)*, and ALG helper rules.

Use **”show firewall”** to show the firewall configuration. If the firewall is enabled, the list of currently configured Packet filtering, Modify, NAT and Port forwarding rules are presented. Also available as **”show”** command within the Firewall Configuration context.

Default values Disabled.

31.3.2 Enable Packet Filter Rules

Syntax `[no] enable`

Context Firewall Configuration context

Usage Enable/disable packet filtering. This setting affects the activation of packet filtering (allow/deny) rules, and the activation of the default policies. Modify, NAT, Port Forwarding, and ALG helper rules are unaffected (they are always enabled).

Use **”enable”** to (re)activate *all* configured packet filtering (allow/deny) rules and the configured default policies for the input and forward filter.

Use **”no enable”** to deactivate *all* the configured packet filtering (allow/deny) rules. Default forward policy will be *accept* and default input policy will be *drop*. ICMP will be allowed on the ingress filter.

Use **”show enable”** to show whether the configured packet filters are enabled or disabled.

It is also possible to activate/deactivate individual allow/deny rules (as well as NAT and port forwarding rules), see section 31.3.11.

Default values Enabled

31.3.3 Configure Packet Filter Rule

Syntax `[no] filter [pos <NUM>] <allow|deny> [in <IFNAME>] [out <IFNAME>] [src <ADDR[/LEN]>] [dst <ADDR[/LEN]>] [dport <PORTRANGE>] [proto <NAME|NUM>] [passive] [log]`

Context Firewall Configuration context

Usage Add or delete a packet filter *allow* or *deny* rule.

- *Rule maintenance parameters (insert position, activate/deactivate or delete rule)*:
  - Allow and deny rules are inserted (and thus evaluated) in a certain order in the input or forward filter. The **”pos <NUM>”** parameter controls at what position in the rule order this packet filter rule should be inserted, or when it comes to removing a rule, which packet filter rule to remove. The order is kept compact (see ”Delete rule” below). Use the **”show filter”** command to list the current packet filter rule list and their position numbers. Examples:
    - *Insert rule*: Use, e.g., **”filter pos 4 allow in vlan2”** will insert an *allow rule* at a specific position (here position 4) in the list of packet filter rules. The rule previously at position 4 will now have position 5, and so on.

      If no position argument is given, the packet filter rule will be inserted last in the list. The position of a command can be modified using the **”move”** command (see section 31.3.10).
    - *Delete rule*: Use, e.g., **”no filter pos 5”** to delete the packet filter rule (allow or deny) at a specific position (here position 5) in the list of packet filter rules. The rule previously at position 6 will now have position 5, and so on, keeping the list compact.

      A rule can also be deleted by using the *no*-form of the filter specification, e.g., the rule **”filter deny in vlan1 out vlan2”** can be deleted by the command **”no filter deny in vlan1 out vlan2”**.
  - The **”passive”** parameter specify that this rule is created as inactive. It will be shown in config but not used. To enable use

”passive” command, see section 31.3.11.

- The **”log”** parameter enables logging for traffic that matches this filter rule. Nothing will however be logged if logging is enabled here but disabled under the common settings. See section 31.3.12. **Note:** Logging differs in behavior between policy *Accept* and *Deny*. See section 31.1.6 for more details.

- *Filter specification parameters*:
  - The first parameter is mandatory and select the action type **”allow”** or **”deny”**.
  - The **”in <IFNAME>”** and **”src <ADDR[/LEN]>”** are used to match the inbound interface and source IP address of a packet. If the **”LEN”** parameter is omitted the **”src <ADDR/LEN>]”** argument will match a single source IP address. If included it will match a whole IP subnet.
  - Include the **”out <IFNAME>”** and/or **”dst <ADDR[/LEN]>”** arguments to define a FORWARDING rule (i.e., packets being routed through the switch). If both the **”out <IFNAME>”** and the **”dst <ADDR[/LEN]>”** arguments are omitted, the rule will apply to the INPUT chain, i.e., traffic destined to the switch itself (ICMP pings, SSH management, etc.).
    The **”out <IFNAME>”** argument is used to match the outbound interface of a packet.

    Use the **”dst <ADDR[/LEN]>”** to match a single destination IP address or whole subnet. If both the **”out <IFNAME>”** and the **”dst <ADDR[/LEN]>”** arguments are omitted, the rule will apply to the INPUT chain, i.e., traffic destined to the switch itself (ICMP pings, SSH management, etc.).
  - Use the **”proto <NAME|NUM>”** to match the IP protocol name, e.g., *tcp*, *udp* or *icmp*. It is also possible to specify the protocol’s assigned number, see http://www.iana.org/assignments/protocol-numbers/.
  - Use the **”dport <PORTRANGE>”** argument to specify a UDP or TCP port number or port range (ex: 1000-1010). This argument is only valid if **”proto udp”** or **”proto tcp”** is included.

Default values Not applicable.

31.3.4 Configure Packet Modify Rule

Syntax `[no] modify [pos <NUM>] [passive]`
`[match [in <IFNAME>] [out <IFNAME>]`
`[src <ADDR[/LEN]>] [dst <ADDR[/LEN]>]`
`[proto <NAME|NUM>] [dport <PORTRANGE>] ]`
`set dscp <VALUE> [adjust-prio]`

Context Firewall Configuration context

Usage Add or delete a modify rule to change the DSCP bits in the IP header for routed traffic.

- *Rule maintenance parameters (insert position, activate/deactivate or delete rule)*:
  - Modifier rules are inserted and evaluated in order. The **”pos <NUM>”** parameter controls at what position in the rule order this modify rule should be inserted, or when it comes to removing a rule, which rule to remove. The order is kept compact (see ”Delete rule” below). Use the **”show modify”** command to list the current modifier rule list and their position numbers. Examples:
    - *Insert rule*: Use, e.g., **”modify pos 4 match in vlan2 set dscp 30”** will insert a modifier rule at position 4 in the list of modifier rules. The rule previously at position 4 will now have position 5, and so on.

      If no position argument is given, the modifier rule will be inserted last in the list. The position of a command can be modified using the **”move”** command (see section 31.3.10).
    - *Delete rule*: Use, e.g., **”no modify pos 5”** to delete the modifier rule at position 5 from the list of modifier rules. The rule previously at position 6 will now have position 5, and so on, keeping the list compact.

      A rule can also be deleted by using the *no*-form, e.g., the rule **”modify match in vlan1 out vlan2 set dscp 0”** can be deleted by the command **”no modify match in vlan1 out vlan2 set dscp 0”**.
  - The **”passive”** parameter specify that this rule is created as inactive. It will be shown in config but not used. To enable use **”passive”** command, see section 31.3.11.

- *Matching parameters*:
  Matching parameters are optional. If you do not specify matching, all routed packets will have the DSCP field set. Matching is enabled with the **”match”** keyword followed by one or more of the filters described below:
  - The **”in <IFNAME>”** and **”out <IFNAME>”** are used to match on the inbound interface or the outbound interface.
  - **”src <ADDR[/LEN]>”** and **”dst <ADDR[/LEN]>”** match on IP source or IP destination. The **”LEN”** parameter is used to define an IP subnet, and if it is omitted it will only match a specific single IP address.
  - Use the **”proto <NAME|NUM>”** to match on traffic with a specific IP protocol. You can use the name, e.g., *tcp*, *udp* or *icmp*, or the protocol's assigned number (see http://www.iana.org/assignments/protocol-numbers/).
  - Use the **”dport <PORTRANGE>”** argument to specify a UDP or TCP port number or port range (ex: 1000-1010). This argument is only valid if **”proto udp”** or **”proto tcp”** is included.
- *Setting parameters*:
  - Use **”set dscp <VALUE>”** to define the DSCP value to be set on all packets matching the parameters described above. The value must be provided as a decimal number in the range 0-63.
  - Add parameter **”<adjust-prio>”** if the internal priority of the packet also should be updated in addition to the change of the DSCP field. The internal priority is used to determine what network queue to use in WeOS networking and hardware. Avoid using this option if not necessary, as it introduces additional work for the CPU in the unit, reducing total performance of the system.

Default values Not applicable.

31.3.5 Configure NAT Rule

Syntax
```
[no] nat [<POS>] [type <napt|1-to-1>] [in <IFNAME>]
    [out <IFNAME>] [src <ADDR[/LEN]>] [dst <ADDR[/LEN]>]
    [to-dst <ADDR[/LEN]>] [addfilter] [noarp] [passive] [log]
```

Context Firewall Configuration context

Usage Add or delete a NAT rule.

- *Add a NAPT NAT rule*
  These keywords are available for creating NAPT rules:
  - **”type napt”**. Select NAPT.
  - **”out <IFNAME>”**. Mandatory. 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.
  - **”in <IFNAME>”**. Optional. Specify that packets must arrive from this interface for this rule to apply.
  - **”src <ADDR[/LEN]>”**. Optional. Specify that packets must originate from a specific IP subnet for this rule to apply.
  - **”addfilter”**. If set, an automatic (invisible) packet filter rule will be created in the *forward filtering* chain allowing packets matching this NAT rule. Do not set this option if you want to manage forwarding rules yourself.
  - **”passive”**. Specify that this rule is created as inactive. It will be shown in config but not used. To enable use **”passive”** command, see section 31.3.11.
  - **”log”**. Enables logging for traffic that matches this NAT rule. Nothing will however be logged if logging is enabled here but disabled under the common settings. See section 31.3.12.
- *Add a 1-to-1 NAT rule*
  These keywords are available for creating 1-to-1 NAT rules:
  - **”type 1-to-1”**. Select 1-to-1 NAT.
  - **”in <IFNAME>”**. Mandatory. The inbound interface used for 1-to-1 NAT.
  - **”dst <ADDR[/LEN]>”**. Mandatory. Packets arriving on the inbound interface and has the IP destination within this subnet will be NATed.
  - **”to-dst <ADDR[/LEN]>”**. Mandatory. The new destination IP network for the NAT. Must be of exact same size as the **”dst”** network.
  - **”addfilter”**. If set, automatic (invisible) packet filter rules will be created in the *forward filtering* chain allowing packets matching this

NAT rule. Rules are created for both the forward and reverse direction (see section 31.1.4.2). Do not set this option if you want to manage forwarding rules yourself.

- **”noarp”**. Specify to disable ARP proxying for this rule. (see section 31.1.4.2 for details).
- **”passive”**. Specify that this rule is created as inactive. It will be shown in config but not used. To enable use **”passive”** command, see section 31.3.11.
- **”log”**. Enables logging for traffic that matches this NAT rule. Nothing will however be logged if logging is enabled here but disabled under the common settings. See section 31.3.12.

- *Delete a NAT rule*
  Use the command **”no nat <POS>”** to delete a specific NAT rule on the position POS as shown with the command **”show”** or **”show nat”**. Delete all NAT rules with **”no nat”**.

Use **”show nat”** to show configured *NAT* rules.

Default values Addresses without subnet lengths will be considered to be of length /32 i.e. as a single IP address.

31.3.6 Configure Port Forwarding Rule

Syntax `[no] port-forward in <IFNAME>:<PORTRANGE> [src <IPADDRESS/LEN>] dst <IPADDRESS>[:PORTRANGE] [proto <tcp|udp>] [passive] [log]`

Context Firewall Configuration context

Usage Add/delete a Port Forwarding rule. This is commonly used when the switch is acting as NAT gateway, see section 31.3.5. E.g., **”port-forward in vlan1:80 dst 10.0.0.2 proto tcp”** to forward all web traffic coming in on interface *vlan1* to the Web server at IP address 10.0.0.2 (port 80).

- The argument **”<IFNAME>:<PORTRANGE>”** specifies incoming interface, and what port or port range to match.
- Use the **”[src <IPADDRESS[/LEN]>]”** to match a single source IP address or whole subnet.
- Use the **”dst <IPADDRESS>[:PORTRANGE]”** to specify where the packets should be forwarded. If the **”PORTRANGE”** parameter is omitted, the

same port range as specified in the **”<IFNAME>:<PORTRANGE>”** argument is used.

- Use the **”[proto <tcp|udp>]”** to specify if the rule applies to TCP or UDP. If omitted, the rule applies to both.
- The **”passive”** parameter specify that this rule is created as inactive. It will be shown in config but not used. To enable use **”passive”** command, see section 31.3.11.
- The **”log”** parameter enables logging for traffic that matches this port forwarding rule. Nothing will however be logged if logging is enabled here but disabled under the common settings. See section 31.3.12. Use **”show port-forward”** to show configured *port forwarding* rules.

Default values Not appliable.

31.3.7 Configure Application Level Gateway (ALG) Helpers

Syntax `[no] alg <ftp|tftp|sip|irc|h323|pptp>`

Context Firewall Configuration context

Usage Enable/disable ALG helper for a protocol, e.g., use **”alg ftp”** to make your firewall or NAT gateway handle FTP traffic appropriately.

Use **”no alg <PROTO>”** to remove an enabled ALG helper for the given protocol, or use **”no alg”** to remove all enabled ALG helpers.

Use **”show alg”** to show list of protocols for which ALG helpers have been enabled.

Default values Disabled.

31.3.8 Configure Stateful Packet Inspection

Syntax `[no] spi`

Context Firewall Configuration context

Usage Stateful packet inspection will drop packet that are in an invalid state. An example of a packet with an ”invalid” state is when a firewall sees a TCP ”SYN+ACK”, without having seen the preceding TCP ”SYN” in the other direction.

For a true firewall it is generally a good idea to enable stateful packet inspection. However, due to potential problems with asymmetric routing, the default is to have this setting disabled.

Use **”show spi”** to show if stateful inspection is enabled or disabled.

Default values Disabled.

31.3.9 Configure Forwarding and Input Default Policies

Syntax `policy [forward|input] <allow|deny>`

Context Firewall Configuration context

Usage Configure the default policy for *forward filtering* and *input filtering*. By default, the command applies to the *forwarding filter*, e.g., **”policy allow”** will set the default policy for forward filtering to **”allow”**.

Use **”show policy”** to show configured default policies for the *forwarding filter* and the *input filter*.

Default values Deny (that is, both the forwarding filter and the input filter by default drop packets lacking a matching *allow* rule.)

31.3.10 Reorder/Move a Packet Filter, Modify, NAT or Port Forwarding Rule

Syntax `move [<filter|modify|nat|port-forward>] <FROM_POS> <TO_POS>`

Context Firewall Configuration context

Usage Change the position (reorder) a rule in the **”filter”**, **”modify”**, **”nat”** or **”port-forward”** table, e.g., use **”move filter 6 3”** to move the filter rule (allow/deny) at position ”6” to position ”3”. The filter rule previously at position ”3” ends up at position ”4”, and so on. Similarly, **”move modify 3 6”** will move the modify rule at position ”3” to position ”6”; the rule previously at position ”6” ends up at position ”5” and so on.

The tables are kept compact. Specifying a **”TO_POS”** beyond the highest number in that table is equal to moving it to the last position in the table.

If no table is specified, the move operation applies to the **”filter”** table, i.e., **”move 6 3”** is equivalent to **”move filter 6 3”**.

Examples

Example

```
example:/config/ip/firewall/#>  show filter
001 filter allow in vlan1 out vlan2
002 filter allow in vlan1 out vlan3
003 filter deny in vlan1 out vlan2 proto icmp
example:/config/ip/firewall/#>  move filter 3 1
example:/config/ip/firewall/#>  show filter
001 filter deny in vlan1 out vlan2 proto icmp
002 filter allow in vlan1 out vlan2
003 filter allow in vlan1 out vlan3
```

31.3.11 Activate/Deactivate a Packet Filter, Modify, NAT, or Port Forwarding Rule

Syntax `[no] passive [<filter|modify|nat|port-forward>] <POS>`

Context Firewall Configuration context

Usage Activate or deactivate a packet filter (allow/deny) rule, a modify rule, a NAT rule, or a port forwarding rule. E.g., use **”passive filter 4”** to deactivate the packet filter rule at position ”4”.

Use commands **”show filter”**, **”show modify”**, **”show nat”** or **”show port-forward”** to display the current list of rules for that specific type.

Use the ”no”-form to activate a previously deactivated rule, e.g., **”no passive modify 4”** activates modify rule ”4”.

Examples

Example

```
example:/config/ip/firewall/#>  show filter
001 filter allow in vlan1 proto icmp
002 filter allow in vlan2 proto icmp
003 filter deny in vlan1 out vlan2 proto icmp
004 filter allow in vlan1 out vlan2
example:/config/ip/firewall/#>  passive filter 3
example:/config/ip/firewall/#>  show filter
001 filter allow in vlan1 proto icmp
002 filter allow in vlan2 proto icmp
003 filter deny in vlan1 out vlan2 proto icmp passive
004 filter allow in vlan1 out vlan2
example:/config/ip/firewall/#>  no passive filter 3
example:/config/ip/firewall/#>  show filter
001 filter allow in vlan1 proto icmp
002 filter allow in vlan2 proto icmp
003 filter deny in vlan1 out vlan2 proto icmp
004 filter allow in vlan1 out vlan2
```

31.3.12 Configuration of firewall logging

This command has two uses, [1] to configure logging (and limit), and [2] to toggle the log flag on firewall rules.

Syntax 1 `[no] log limit ( none | <entries>/(second|minute|hour|day) )`

Syntax 2 `[no] log [filter|nat|port-forward] <POS>`

Context Firewall Configuration context

Usage 1 Enable/disable firewall logging and set rate limitation of firewall log entries. This is a master control enabling the logging feature.
A rate limit must be provided or “none” to disable limit, i.e. log everything. The limit is set as a number followed by a slash character “/” and a time unit. The time unit is one of “second”, “minute”, “hour” or “day”. See section 31.1.6 for information about how limitation operates.

All firewall logging is disabled by using the command: **”no log”**

Use **”show log”** to show if firewall logging is enabled or disabled, and the rate limitation setting.

Note
Besides enabling logging with this command, you also need to enable logging on individual firewall rules for anything to be logged.

Warning
Enabling logging and disabling the limitation may lead to lots of data being logged. This can in a short time fill up the log files.

Usage 2 Enable/disable logging for an existing individual packet filter, NAT or port forwarding rule. E.g., use **”log filter 4”** to enable logging for the packet filter rule at position ”4”.

Use commands **”show filter”**, **”show nat”** or **”show port-forward”** to display the current list of rules for that specific type. Rules containing the keyword “log” has logging enabled.

Use the ”no”-form to disable logging for an existing rule, e.g., **”no log nat 2”** disables logging for the NAT rule at position ”2”.

Logging can not be enabled for packet modify rules.

Default values Logging is enabled by default when the firewall is enabled, however no automatically created firewall rule will have the log parameter enabled by default. The default logging limit is set at 5 entries per second.

Examples with usage 1

Example

```
example:/config/ip/firewall/#>  log limit 100/day
example:/config/ip/firewall/#>  show log
Logging is Enabled, limited to 100 entries/day
example:/config/ip/firewall/#>  log limit none
example:/config/ip/firewall/#>  show log
Logging is Enabled, no rate limitation
example:/config/ip/firewall/#>  no log
example:/config/ip/firewall/#>  show log
Logging is Disabled
```

Examples with usage 2

Example

```
example:/config/ip/firewall/#>  show filter
001 filter allow in vlan1 proto icmp
002 filter allow in vlan2 proto icmp
003 filter deny in vlan1 out vlan2 proto icmp
004 filter allow in vlan1 out vlan2
example:/config/ip/firewall/#>  log filter 2
example:/config/ip/firewall/#>  show filter
001 filter allow in vlan1 proto icmp
002 filter allow in vlan2 proto icmp log
003 filter deny in vlan1 out vlan2 proto icmp
004 filter allow in vlan1 out vlan2
example:/config/ip/firewall/#>  no log filter 2
example:/config/ip/firewall/#>  show filter
001 filter allow in vlan1 proto icmp
002 filter allow in vlan2 proto icmp
003 filter deny in vlan1 out vlan2 proto icmp
004 filter allow in vlan1 out vlan2
```

31.3.13 View Firewall Status

Syntax `show firewall`

Context Admin Exec context

Usage Show current NAT rules, Port Forwarding rules, policies and entries in the Input and Forwarding Filters and Modifier rules. In addition, management interface configuration (see section 19.2.7) will appear as entries in the *Input Filter*.

Default values Not applicable.

Part IV

Virtual Private Networks and Tunnels

Chapter 32

Overview of WeOS VPN and Tunnel support

This chapter introduces WeOS support for virtual private networks (VPNs), IPsec and SSL VPN, as well as support for tunneling/point-to-point functionality (GRE and PPP). Although GRE and PPP can be used as part of VPNs, they can also be used as standalone features, e.g., to setup IP communication over a serial link.

32.1 WeOS support for VPNs

As shown in fig. 32.1, a WeOS switch can act as a VPN gateway in NETWORK-NETWORK and HOST-NETWORK scenarios. Configured as a VPN gateway, it can be used to securely connect branch office networks with a central office network, or to serve individual users wishing to ”dial in” securely over the Internet to the central office network, with their PC connected at some remote site. The data traffic will be protected by encrypted tunnels when sent over the Internet. A WeOS unit supports at most 25 simultaneous VPN tunnels.

WeOS provides two flavours of VPN support, which both support NETWORK-NETWORK and HOST-NETWORK VPN scenarios.

- *IPsec VPNs:* WeOS supports IPsec VPNs with IKEv1 (shared key and certificates) for authentication, and ESP for encapsulation of encrypted IP packets.
- SSL VPN: The WeOS SSL VPN support is based on OpenVPN[1].

[1] http://www.openvpn.net

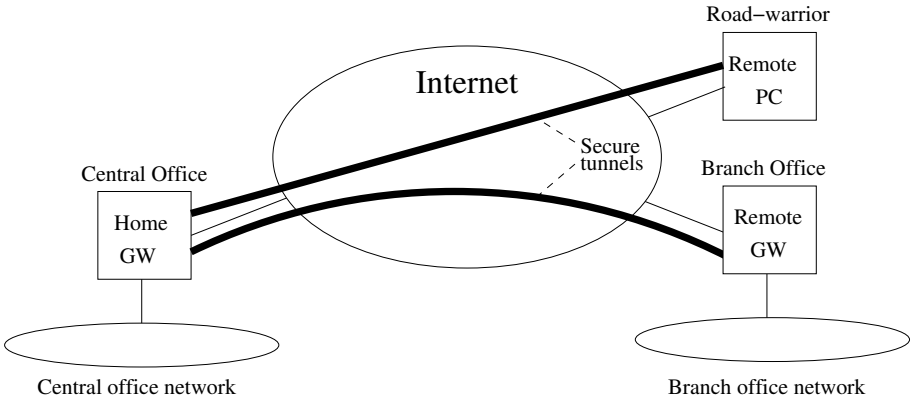


Figure 32.1: IPsec VPN tunnels can be used to securely connect hosts and networks over the Internet.

Both IPsec and SSL VPNs offer high level security. SSL VPNs are commonly considered easier to configure, and is often preferred to setup VPNs through firewalls managed by an external organisation. In other situations IPsec is the only choice, as it may be mandated by the customers.

Both SSL and IPsec VPNs are able to carry encrypted IP traffic. The WeOS SSL VPN is also able to carry encrypted Ethernet traffic, however, as of WeOS v4.17.1 this traffic can be *routed* but not *bridged*. Support for bridged SSL VPNs is planned, but not yet supported. IPsec VPNs are further described in chapter 35 and SSL VPNs are covered in chapter 36.

32.2 Tunneling using PPP

WeOS supports PPP over serial ports (as PPP client and server), and PPP over Ethernet (PPPoE) as client. PPP support is further described in chapter 33.

32.3 Tunneling using GRE

WeOS provides support for GRE tunnels (IP over GRE), which is useful in scenarios IPsec VPNs and OSPF are used to provide secure and redundant connectivity between branch offices and a central office. WeOS GRE support is covered in chapter 34.

Chapter 33

Point-to-Point Protocol (PPP) Connections

WeOS provides two types of PPP services:

- PPPoE (Ethernet/DSL): WeOS supports PPPoE client services on LAN. The PPPoE client operates on Ethernet and DSL ports (SHDSL, ADSL, VDSL) associated with a VLAN network interface.
- PPP over Serial Port: On serial ports, WeOS supports PPP dial in/out services with or without external modem.

This chapter describes PPP support in WeOS in general, with focus on how to create PPP instances, and configuration of low-level PPP settings for PPPoEad PPP over serial ports. PPP shares some functionality with other WeOS services, thus additional information relevant for PPP configuration is found at the following locations:

- General Interface settings: A network interface will be created for each PPP instance. Configuration of network interfaces is described in chapter 19.
- PPPoE on Falcon (xDSL): Section 11.2.1 provides useful information when using PPPoE on a Falcon xDSL router.
- Serial Port Settings: When running PPP over a serial port, there general serial port settings to be carried out in addition to the PPP settings described in this chapter. Configuration of serial port is described in chapter 38.
- Peer authentication: To authenticate the peer side of the PPP connection a local PPP user database is used. Configuration of local user databases is

covered in chapter 21.

33.1 Overview of PPP Instance Properties and Management Features

Feature	Web	CLI	General Description
Link types			
Ethernet (PPPoE client)	X	X	Section 33.1.1-33.1.3
Serial/modem	X	X	Section 33.1.1-33.1.2, and Section 33.1.4
PPP Link Establishment			
MRU negotiation	X	X	Section 33.1.2
PPP authentication			
Protocols: PAP,CHAP, . . .	X	X	Section 33.1.2, 33.1.5
Username/password	X	X	Section 33.1.2, 33.1.5
Peer authentication	X	X	Section 33.1.2, 33.1.5, and Chapter 21
MPPE Encryption	X	X	Section 33.1.2, 33.1.6
IP/Interface			
Address Assignment	X	X	Section 33.1.7
Proxy ARP	X	X	-”-
On demand dialing	X	X	-”-
Other interface settings (default route, etc.)	X	X	Chapter 19

33.1.1 Introduction to PPP

The Point-to-Point Protocol (PPP)[36] is a common data link protocol for point-to-point links. PPP is able to carry different kinds of layer-3 protocols, and can be used in several contexts. WeOS supports IP (IPv4) service over PPP for the following link types[1]:

- *PPP over Serial Link:* PPP can be used as data link protocol over serial links, e.g., by connecting to units directly via a serial (null-modem) cable, or over

[1] Future releases of WeOS may support additional PPP modes, such as PPTP and L2TP for VPN dial-in services.

a PSTN by use of modems.

- *PPP over Ethernet:* PPP can be used on Ethernet (or DSL) by use of the PPP over Ethernet (PPPoE) protocol[23]. WeOS provides a PPPoE client service, which is commonly used when connecting to an ISP via an xDSL connection.

As of WeOS v4.17.1, WeOS units can establish a single PPP connection using PPPoE, and a single PPP connection over serial port[2].

33.1.2 Phases in the PPP connection establishment

The two units establishing a PPP connection are referred to as *peers* in PPP terminology[36]. Here we will either denote them as *PPP peers*, or as *PPP client* and *PPP server* when referring the unit *initiating* the connection (i.e., dial-out) or the unit *waiting* for an incoming call (i.e., dial-in) respectively.

Establishment of a PPP connection is divided into several phases, as shown in fig. 33.1:

- *"Low-level" link establishment (Pre-PPP):* Before a PPP connection can be established, a point-to-point "link" must exist, either as a physical link (serial line), or as a logical link (PPP over Ethernet or PPTP/L2TP[3]).
  - *PPPoE:* To create a point-to-point connection over an Ethernet, the PPPoE protocol is used. Once the PPPoE handshake has finished, the PPP Link Establishment phase can start. See section 33.1.3 for more information on PPPoE specific settings.
  - *PPP over Serial Port:* For PPP over serial link, it is enough to configure the serial ports and the external modem (if used) to establish the physical link. No pre-PPP phase exists – the PPP Link Establishment phase starts immediately. See section 33.1.4 for additional information on how to configure the serial port and modem.
- *PPP Link Establishment Phase:* Once the point-to-point link is up, the PPP peers start to exchange PPP Link Control Protocol (LCP) messages. LCP is used to negotiate general settings, which are *independent of the network layer protocol(s)* used on top, e.g., the maximum receive unit (MRU), or what authentication protocol to use (if any). LCP is also used by the PPP peers to

[2] PPP can be run over the serial port of the WeOS serial port, but not via its console port. See section 1.5.1 for information on WeOS units equipped with serial ports.
[3] As of WeOS v4.17.1, PPTP or L2TP are not yet supported.

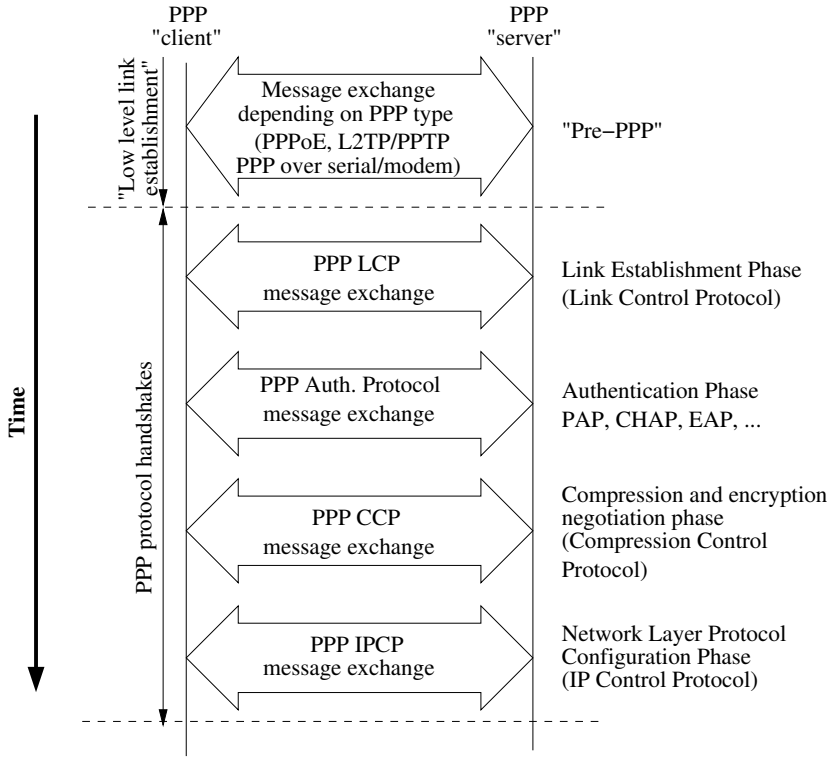


Figure 33.1: PPP Connection Establishment Phases

send LCP Echo Request/Reply messages, to verify connectivity once the link is up. As of WeOS v4.17.1 the LCP Echo Interval is 20 seconds (fixed), and the link is considered down after failing to receive three LCP responses.

- *PPP Authentication Phase:* During the Link Establishment phase, the peers can negotiate the use (and type) of authentication. See section 33.1.5 for more information on WeOS support for PPP authentication.
- *Compression and Encryption Negotiation Phase:* After the Link Establishment and Authentication phases, the PPP peers can use the PPP compression control protocol (CCP[32]) to negotiate link layer compression or encryption (typically the Microsoft Point-To-Point Encryption (MPPE) Protocol[27]). See section 33.1.6 for more information on WeOS support for PPP encryption.

  As of WeOS v4.17.1 PPP link layer compression is not supported.
- *Network Control Protocol Phase:* Once the link has been established via LCP, and the optional authentication and compression handshakes are car-

ried out, PPP can start to negotiate network level settings via one or more network layer protocols. Here the PPP IP Control Protocol (IPCP[24]) is used to negotiate IP Settings. Acting as PPP client, WeOS units will use IPCP to acquire an IP address for the PPP interface, as well as its domain name server(s).

Note
The domain name servers learnt via IPCP will only be used if the PPP interface has lowest *admin distance* (see section 19.2.6), and if no static domain name server is configured. Similarly, the peer will only be used as default gateway if the PPP interface has lowest *admin distance* and if no static default route has been configured.

33.1.3 PPP over Ethernet (PPPoE)

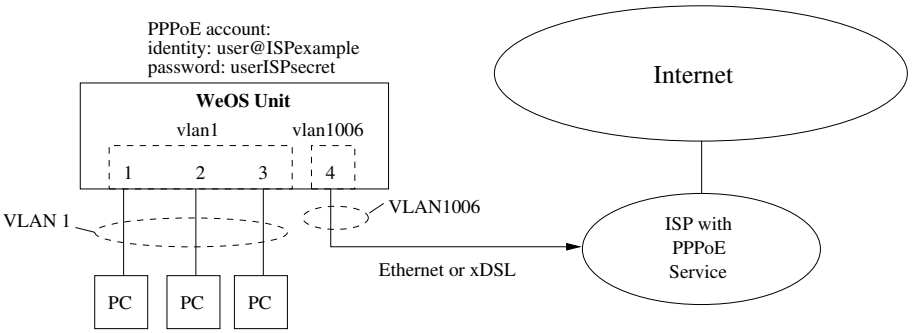


Figure 33.2: Example where WeOS unit routes traffic to Internet using PPPoE.

PPPoE is a protocol to establish a PPP connection over an Ethernet network. It is commonly used when connecting to an ISP over an xDSL or Ethernet connection, since PPPoE enables the use of PPP's features for user authentication and dynamic IP assignment. Fig. 33.2 shows a sample setup.

To configure PPPoE in WeOS you need to specify the following:

- The VLAN interface to run PPPoE over, i.e., the VLAN your upstreams xDSL or Ethernet port is associated with. In fig. 33.2 interface *vlan1006* is used.
- The *identity* and *password* assigned to you by your ISP (this is the PAP/CHAP username and password mentioned in section 33.1.2). In fig. 33.2 identity *user@ISPexample* and password *userISPsecret* are used.

- (Optionally) Some access network are shared between multiple ISPs. In order to connect to the PPPoE Server of your ISP, you then need to fill in the *service name* provided by your ISP. This step can typically be skipped.

Section 11.2.1 provides additional information, which is useful when setting up PPPoE on a Falcon xDSL router.

33.1.4 PPP over Serial Port

PPP over a Serial Port is in WeOS configured in the Modem context. For details of the configuration of the actual serial port see chapter 38.

The Serial PPP can be set up in 4 modes.

- *Null modem:* Two devices can be connected directly using a *null modem* cable (a serial cable where transmit and receive are cross-linked).

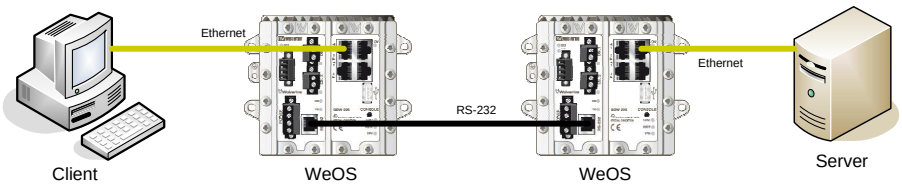


Figure 33.3: PPP - Null modem setup example

To setup a null modem PPP is simple. Select *null modem* as mode on both sides, and change the *local IP address* on one side in the PPP context.

- *Dial in:* Allows for a remote client to dial in to the device over a PSTN or leased line.

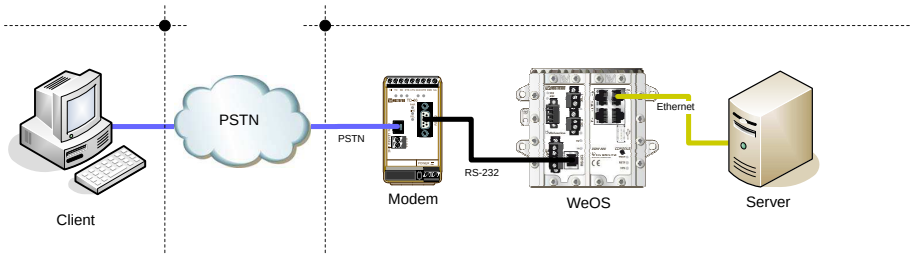


Figure 33.4: PPP - Dial in setup example

To setup a null modem PPP is simple. Select Null modem as mode on both sides and change Local IP on one side in the PPP context.

- *Dial out:* Allows for a local device to establish a PPP connection on demand over a PSTN or leased line.

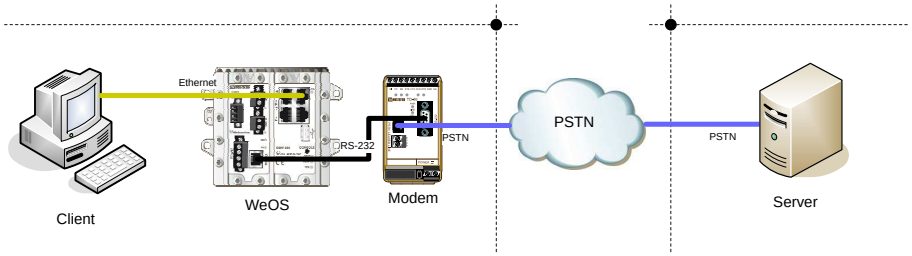


Figure 33.5: PPP - Dial out setup example

- *Dial in & out:* A combination of Dial in and out. Allows for both connections on demand and for incoming requests.

To setup a modem connected to the unit, WeOS provides an *initialisation string*. This AT-sequence is transmitted to the modem during system configuration. The default string is **ATE0Q0V1&C1&S0S0=0&W** (see table 33.1 for more information).

The initialisation string is not used when connecting via *null modem* cable.

The *dial string* is the AT-sequence that starts the connection attempt. A typical dial string is **ATDnnn** where *nnn* is the phone number to dial. If the modem uses a leased line the dial string typically is **ATD**.

E0	Disable Echo
Q0	Set Quite mode off
V1	Set Verbose mode on
&C1	Set DCD to follow the state of a carrier
&S0	Set DSR signal to follow mode (data or command)
S0=0	Disable auto-answer
&W	Save settings

Table 33.1: Breakdown of the initialisation string.

The dial string is not used when connecting via a *null modem* cable.

33.1.5 PPP Authentication Support

PPP enables you to authenticate yourself to your peer. This is typically needed when using PPPoE to connect to your ISP. To accomplish this,you add your *credentials* (identity/username and password) to you PPP configuration.

PPP also enables you to authenticate your peer, which is useful when you provide a *dial-in* service, but can also be used for *dial-out*. As of WeOS v4.17.1, a *local* database list (see section 21.1.2) can be configured with credentials of authorised peers. Later releases of WeOS may include PPP support for backend authentication, e.g., via RADIUS. Peer authentication is by default *Disabled*.

WeOS supports authentication using the password authentication protocol (PAP[20]) and challenge handshake authentication protocol (CHAP), including regular CHAP[35], MS-CHAP[56] and MS-CHAPv2[55]. By default all authentication protocols are available, but it is possible to specify which protocol(s) to use[4]. In WeOS the same set of authentication protocols are available for authenticating yourself to the peer as for the peer to authenticate to you.

When using MPPE to encrypt your PPP session (see section 33.1.6), use of MS-CHAPv2 or MS-CHAP is required.

[4] If more than one protocol are available, a WeOS unit will propose protocols in the following preference order: CHAP, MS-CHAPv2, MS-CHAP, and finally PAP.

33.1.6 PPP Encryption Support

WeOS provides support for the Microsoft Point-To-Point Encryption (MPPE) Protocol[27]), either with 40 or 128 bit key lengths. By enabling MPPE you achieve a basic level of protection of your PPP session. However, to reach a higher level of security it is recommended to use IPsec VPNs or SSL VPNs (OpenVPN) as described in chapters 35 and 36.

Use of MPPE requires that either MS-CHAPv2 or MS-CHAP are used for authentication, see section 33.1.5. MPPE is *disabled* by default.

33.1.7 IP and PPP network interfaces

Configuration of IP settings of PPP interfaces is handled somewhat differently as compared to other network interfaces in WeOS. The main reason is that PPP contains more options related to IP settings.

The following PPP related IP or interface settings are configured in the *Modem* or *PPPoE* contexts . Most important are the *local* and *remote* IP address settings:

- *Local IP address:* Your local IP address can either be assigned dynamically by the peer, or you can assign a static IP address for your PPP interface.
- *Remote IP address:* You can either assign an IP address to your peer, or accept the peer to use an IP address chosen by itself.
- *Proxy ARP:* A WeOS unit will by default apply *proxy ARP* to its PPP connections. With *proxy ARP* enabled for a PPP connection, the WeOS unit will check if the PPP peer's IP address matches any local IP subnet. The unit will then respond to ARP requests for the peer's IP address on that local VLAN.

  E.g., if the remote PPP address is *10.1.0.10*, and this matches the subnet of the local interface *vlan1* with address *10.1.0.2/24*, the WeOS unit will respond to ARP requests for *10.1.0.10* on *vlan1*.
- *On demand dialing:* PPP interfaces are commonly brought up immediately. However, in some use cases it is preferred to only have the PPP connection up when the units are actively sending traffic. The connection is brought up when there is traffic to be routed through that path, and brought down after a configurable idle timeout. To get traffic routed through the PPP interface (and bring it up) you can use a static route. A static 0.0.0.0/0 route to the PPP interface sets it as default.

On demand dialing is only applicable in PPP scenarios where the unit is acting as *client*, i.e., dialing out to a PPP *server*. On demand dialing is disabled by default.

Below is an example where the local address of a PPP null modem interface is set to *192.168.5.1* and the address *192.168.5.2* is assigned to the peer.

Example

```
example:/#> configure
example:/config/#> modem 0
Creating modem 0
Dial-mode: Null-modem
Serial port: 2
example:/config/modem-0/#> address 192.168.5.1
example:/config/modem-0/#> remote-address 192.168.5.2
example:/config/modem-0/#> end
example:/config/#> end
Stopping DHCP/DNS Server ................................ [ OK ]
Starting DHCP/DNS Server ................................ [ OK ]
Starting Modem link monitor ............................. [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> copy running startup
example:/#>
```

For every PPP connection there is an associated PPP interface (e.g., **”modem0”** or **”pppoe0”**), and these interfaces are treated as regular interfaces in WeOS with additional configuration options, see chapter 19. In particular, if you wish to learn your *default route* and *DNS servers* dynamically from your PPP peer, you should give your PPP interface *admin distance* ”1”, see section 19.2.6.

Below is an example where a PPP null-modem connection is configured to get its IP address, default route and name servers from its peer. In addition, here management of the unit through this PPP interface is limited to HTTPS.

Example

```
example:/#> configure
example:/config/#> modem 0
Creating modem 0
Dial-mode: Null-modem
Serial port: 2
example:/config/modem-0/#> no address
example:/config/modem-0/#> no remote-address
example:/config/modem-0/#> end
example:/config/#> iface modem0
example:/config/iface-modem0/#> distance 1
example:/config/iface-modem0/#> no management
example:/config/iface-modem0/#> management https
example:/config/iface-modem0/#> end
example:/config/#> end
Starting Modem link monitor ............................... [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
example:/#> copy running startup
example:/#>
```

33.2 Managing PPP settings via the web interface

The Web interface provides configuration of PPP connections, both for PPPoE (sections 33.2.1-33.2.2 and for PPP over modem/serial port (sections 33.2.3-33.2.4).

33.2.1 PPPoE overview

Menu path: Configuration ⇒ PPP ⇒ PPPoE

Figure 33.6: PPP settings overview

Click on the Edit icon ( ) to edit the settings of a specific PPPoE instance.

33.2.2 Edit PPPoE Settings

Menu path: Configuration ⇒ PPP ⇒ PPPoE ⇒ 

On this page you can change the settings for PPP connections.

The page has two views, a simple view (fig. 33.7) and an advanced view (fig. 33.8).

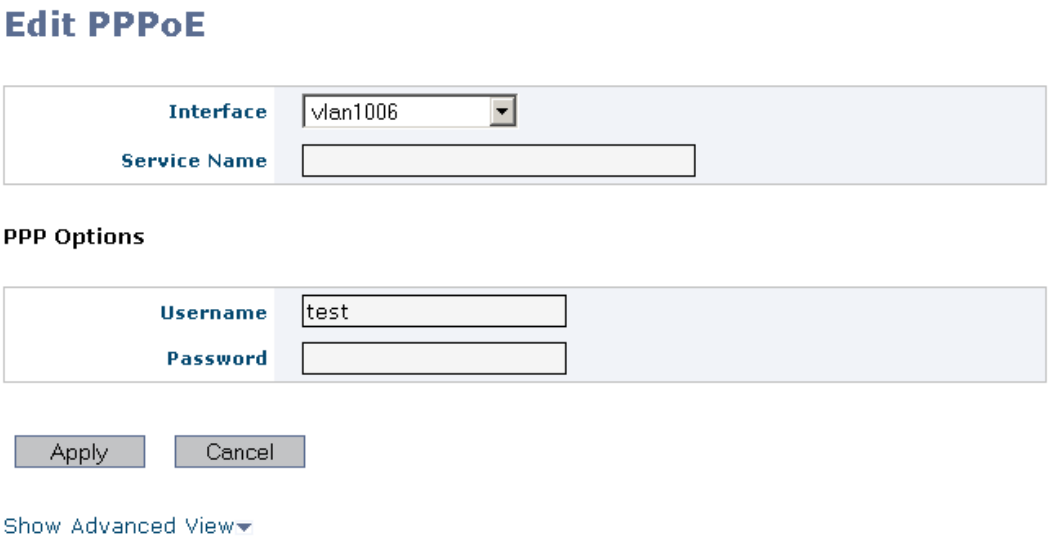


Figure 33.7: PPPoE edit page

Type	Type of PPP link
Interface	Interface for binding of PPP link.
Username	Username for authenticating against the peer
Password	Password for authenticating against the peer
Local IP	The Local IP for this link
Remote IP	The Remote IP for this link
Peer Authentication	Enable authentication of peers
Authentication Protocol	Select authentication protocol(s)
Crypto	Select link encryption
Dial-on-demand	Enable Dial-on-demand and sets disconnect time-out
MRU Negotiation	Enable maximum receive unit (MRU) negotiation

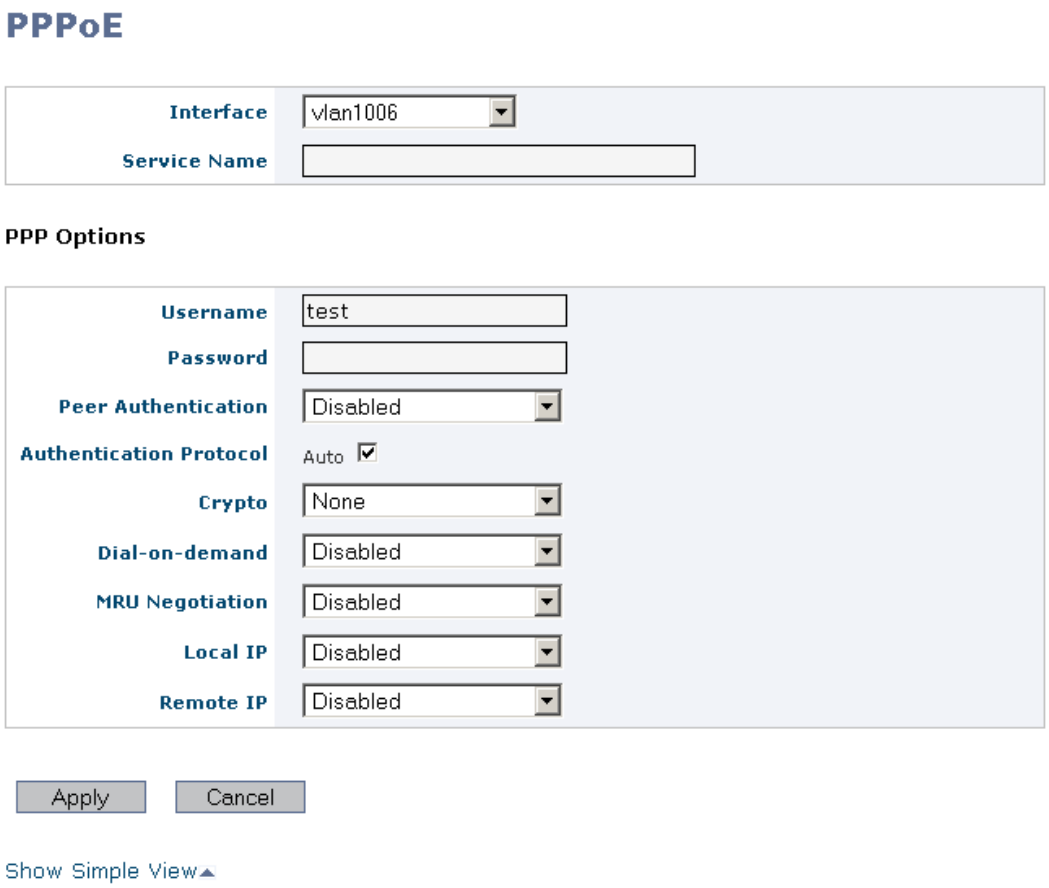


Figure 33.8: PPPoE advanced edit page

33.2.3 Modem overview

Menu path: Configuration ⇒ PPP ⇒ Modem

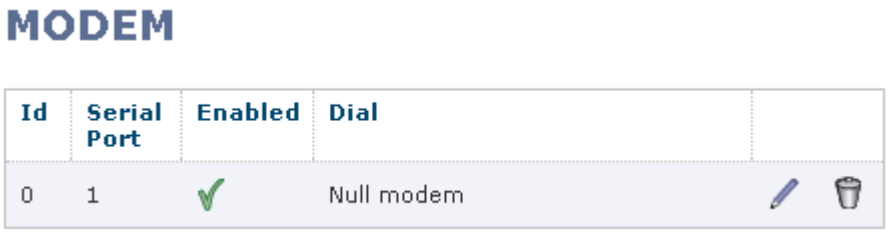


Figure 33.9: Modem settings overview

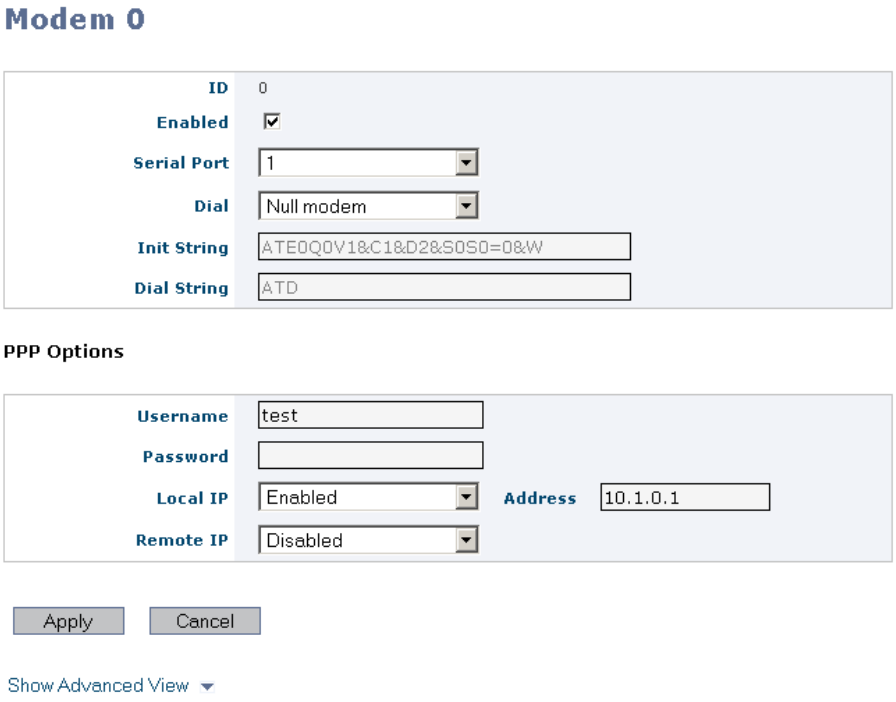
Click on the Edit icon ( ) to edit the settings of a specific Modem instance.

33.2.4 Edit Modem Settings

Menu path: Configuration ⇒ PPP ⇒ Modem ⇒ 

On this page you can change the settings for Modem connections.

The page has two views, a simple view and an advanced view.

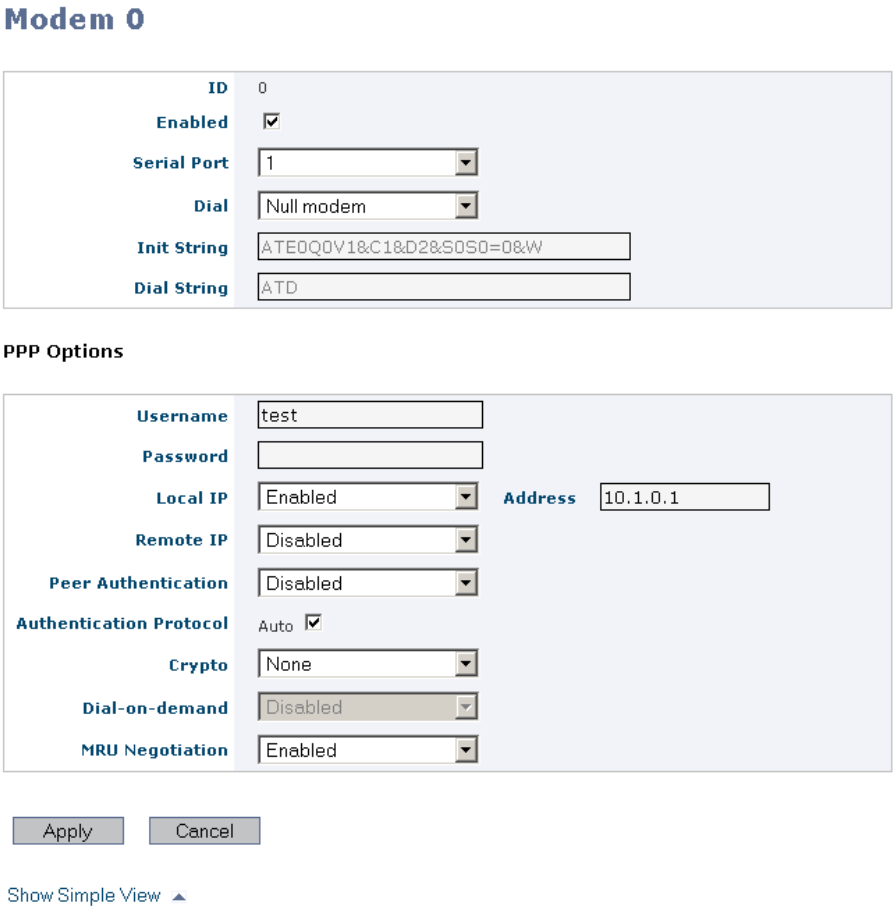


ID	The instance id
Enabled	Enable/disable this instance
Serial Port	The serial port to use
Dial	Set the mode of the modem.
Init String	Set the AT-sequence to intialize a modem
Dial String	Set the AT-sequence to dial the remote host
Username	Username for authenticating against the peer
Password	Password for authenticating against the peer
Local IP	The Local IP for this link
Remote IP	The Remote IP for this link

Continued on next page

	Continued from previous page
Peer Authentication	Enable authentication of peers
Authentication Protocol	Select authentication protocol(s)
Crypto	Select link encryption
Dial-on-demand	Enable Dial-on-demand and sets disconnect time-out
MRU Negotiation	Enable maximum receive unit (MRU) negotiation

Modem advanced edit page



33.3 Managing PPP settings via the CLI

Table 33.3 shows CLI commands related to *PPPoE management*.

For *PPP over serial port* (and modems) there are two tables to summarise the CLI commands: Table 33.5 presents commands the CLI commands in a generic way, while Table 33.6 describes CLI commands and default values depending on the *dial mode* (**”dial in”**, **”dial out”**, **”dial in,out”** and **”no dial”** (i.e., ”null-modem”)).

Command	Default	Section
Basic and advanced settings for PPPoE		
[no] pppoe <ID>		Sec. 33.3.1
[no] iface <IFNAME>	Disabled	Sec. 33.3.2
[no] service-name <SERVICE-NAME>	Disabled	Sec. 33.3.3
[no] enable	Enabled	Sec. 33.3.9
[no] identity <USERNAME> password <PASSWORD>	Disabled	Sec. 33.3.10
[no] ppp-advanced		Sec. 33.3.11
[no] address <IPV4ADDR>	Auto	Sec. 33.3.12
[no] remote-address <IPV4ADDR>	Auto	Sec. 33.3.13
[no] auth-proto <pap, . . . >	Disabled	Sec. 33.3.14
[no] aaa-method local-db <ID>	Disabled	Sec. 33.3.15
[no] crypto <mppe-40 | mppe-128>	Disabled	Sec. 33.3.16
[no] proxy-arp	Enabled	Sec. 33.3.17
[no] demand <IDLE-TIMEOUT>	Disabled	Sec. 33.3.18
[no] mru	Enabled	Sec. 33.3.19
Configure Peer Authentication Lists		
aaa		
[no] local-db <ID> [plain]		Sec. 21.3.3
. . .		

Table 33.3: CLI setting relevant for PPPoE management. All PPP settings are available in the ”ppp-advanced” subcontext. The most common PPP settings are (also) available in the main ”pppoe” context.

Command	Default	Section
Basic and advanced settings for PPP over serial/modem		
[no] modem <ID>		Sec. 33.3.4
[no] port <SERIALPORT>	”First Free”	Sec. 33.3.5
[no] dial <in, out>	Null-modem	Sec. 33.3.6
[no] init-string <STRING>	ATE...	Sec. 33.3.7
[no] dial-string <STRING>	ATD	Sec. 33.3.8
[no] enable	Enabled	Sec. 33.3.9
[no] identity <USERNAME> password <PASSWORD>	Disabled	Sec. 33.3.10
[no] ppp-advanced		Sec. 33.3.11
[no] address <IPV4ADDR>	Differs	Sec. 33.3.12
[no] remote-address <IPV4ADDR>	Differs	Sec. 33.3.13
[no] auth-proto <pap, ...>	Disabled	Sec. 33.3.14
[no] aaa-method local-db <ID>	Disabled	Sec. 33.3.15
[no] crypto <mppe-40 | mppe-128>	Disabled	Sec. 33.3.16
[no] proxy-arp	Enabled	Sec. 33.3.17
[no] demand <IDLE-TIMEOUT>	Disabled	Sec. 33.3.18
[no] mru	Enabled	Sec. 33.3.19
Serial Port Settings		
port serial <SERIALPORT>		Sec. 38.3.1
[no] speed <50-2000000>	115200	Sec. 38.3.2
...		
Configure Peer Authentication Lists		
aaa		
[no] local-db <ID> [plain]		Sec. 21.3.3
...		

Table 33.5: CLI setting relevant for management of PPP over serial ports with or without external modem. All PPP settings are available in the ”ppp-advanced” subcontext. The most common PPP settings are (also) available in the main ”modem” context.

Command	Comment	Section
No-dial/Null-modem defaults and contexts		
modem <ID>		
no dial	Null-modem	Sec. 33.3.6
address 10.1.0.1	Default Local IP	Sec. 33.3.12
no remote-address	Accept Remote IP	Sec. 33.3.13
(no init-string)	N/A	Sec. 33.3.6
(no dial-string)	N/A	Sec. 33.3.8
ppp-advanced		
(no demand)	N/A	Sec. 33.3.18
Dial-In specific defaults and contexts		
modem <ID>		
dial in	Dial-In	Sec. 33.3.6
address 10.1.0.1	Default Local IP	Sec. 33.3.12
remote-address 10.1.0.2	Default Remote IP	Sec. 33.3.13
no aaa-method	Basic	Sec. 33.3.15
ppp-advanced		
(no demand)	N/A	Sec. 33.3.18
Dial-Out specific defaults and contexts		
modem <ID>		
dial out	Dial-Out	Sec. 33.3.6
no address	Dynamic Local IP	Sec. 33.3.12
no remote-address	Accept Remote IP	Sec. 33.3.13
ppp-advanced		
no demand	Advanced	Sec. 33.3.18
Dial-In/Out specific defaults and contexts		
modem <ID>		
dial in,out	Dial-In/Out	Sec. 33.3.6
address 10.1.0.1	Default Local IP	Sec. 33.3.12
remote-address 10.1.0.2	Default Remote IP	Sec. 33.3.13
no aaa-method	Basic	Sec. 33.3.15
ppp-advanced		
demand 600	Enabled	Sec. 33.3.18

Table 33.6: Summary of differences in *default settings* and in the split between *basic* and *advanced PPP settings* for different dial modes.

33.3.1 Managing PPPoE connections

Syntax `[no] pppoe <ID>`

Context Global Configuration context

Usage Enter the PPPoE configuration context of the given PPPoE instance ID. If this is a new PPPoE instance, the PPP instance will be created first upon leaving the PPP context with *end* or *leave*. An associated network interface *pppoe<ID>* (e.g., *pppoe0*) will be created (see chapter 19).

Use **”no pppoe <ID>”** to remove an existing PPP instance, or no pppoe to remove all PPP instances.

As of WeOS v4.17.1 only a single PPPoE instance (ID ”0”) is supported.

Default values Not applicable.

33.3.2 PPPoE VLAN Interface Setting

Syntax `[no] iface <IFNAME>`

Context PPPoE Configuration context

Usage Set the (VLAN) network interface where this PPPoE instance should operate, e.g., **”iface vlan10”**.

Use **”show iface”** to check the interface setting for this PPPoE instance.

Default values None defined

33.3.3 PPPoE Service Name

Syntax `[no] service-name <SERVICE-NAME>`

Context PPPoE Configuration context

Usage ISP name or a class of service configured on PPP.

Use **”show service-name”** to check the service name setting for this PPPoE instance.

Default values Disabled (**”no service-name”**)

33.3.4 PPP Modem: Managing PPP over Serial Port and Modem

Syntax `[no] modem <ID>`

Context Global Configuration context

Usage Enter the PPP Modem configuration context of the given modem instance ID (defaults to ID ”0”). If this is a new modem instance, the modem instance will be created first upon leaving the modem context with *end* or *leave*. An associated network interface *modem<ID>* (e.g., *modem0*) will be created (see chapter 19).

Use **”no modem <ID>”** to remove an existing modem instance, or no modem to remove all modem instances.

As of WeOS v4.17.1 only a single modem instance (ID ”0”) is supported.

Default values Not applicable.

33.3.5 PPP Modem Serial Port

Syntax `[no] port <PORT>`

Context PPP Modem Configuration context

Usage Serial port connected to external modem or null modem cable.

Use **”show port”** to view which port will be used for communication.

Default values The first (lexicographically) enabled and available port will be chosen when creating a new modem instance. If there are no eligible ports, it will be disabled (**”no port”**).

33.3.6 PPP Modem Dial Mode

Syntax `[no] dial <in out>`

Context PPP Modem Configuration context

Usage Dial mode with external modem or null modem mode.

- **”no dial”**: Null modem mode.
- **”dial in”**: Dial-in. WeOS will respond to incoming calls.

- **”dial out”**: Dial-out. WeOS will initiate outgoing calls.
- **”dial in out”**: Combined dial-in/out. WeOS will be able to respond to incoming calls and initiate outgoing calls.

Use **”show dial”** to view the current dial mode.

Default values Null modem (**”no dial”**).

33.3.7 PPP Modem Initialisation String

Syntax `[no] init-string <AT-CMD>`

Context PPP Modem Configuration context

Usage AT command sequence used to initialise an external modem.

This option is only available when using an external modem, it is not applicable in a null modem setup.

Use **”show init-string”** to view the current initialisation string.

Default values `ATE0Q0V1&C1&S0S0=0&W`

33.3.8 PPP Modem Dial String

Syntax `[no] dial-string <AT-CMD>`

Context PPP Modem Configuration context

Usage AT command sequence used to make an outgoing call.

This option is only available when acting as an initiator, making outgoing calls using an external modem. In other words when dial mode is one of **”dial out”** or **”dial in out”**.

Use **”show dial-string”** to view the current dial string.

Default values `ATD`

33.3.9 PPP Enable

Syntax `[no] enable`

Context Generic PPP setting (PPPoE Configuration and PPP Modem Configuration contexts)

Usage Enable, or disable this PPP link.

Use **”`show enable`”** to check if this PPP instance is enabled or not.

Default values Enabled

33.3.10 PPP Credentials (Username and Password)

Syntax `[no] identity <USERNAME> password <PASSWORD>`

Context Generic PPP setting (PPPoE Configuration and PPP Modem Configuration contexts)

Usage PPP credentials, i.e., your username and password for the PPP connection. This information is used to authenticate you to the peer end of the PPP connection, typically your ISP.

(For information on how to authenticate your peer, see Sec. 33.3.15.)

Default values Disabled (**”`no identity`”**)

33.3.11 PPP Advanced Context

Syntax `[no] ppp-advanced`

Context Generic PPP setting (PPPoE Configuration and PPP Modem Configuration contexts)

Usage Enter the PPP Advanced Configuration context. This contexts holds all PPP settings applicable for this type of PPP context, while only the most common settings are available in the generic PPPoE Configuration and PPP Modem Configuration contexts) above. See tables 33.3-33.6 for more information.

33.3.12 PPP Local Address Setting

Syntax `[no] address <ADDRESS>`

Context Generic PPP setting (PPPoE Configuration and PPP Modem Configuration contexts)

Usage Set the local IP address for this PPP link.

Use **”show address”** to view the currently set address.

Default values Based on the link type and ID, for more details see section 33.1.7.

33.3.13 PPP Remote/Peer Address Setting

Syntax `[no] remote-address <ADDRESS>`

Context PPP Advanced Configuration context (also as generic PPP setting in PPP Modem Configuration context)

Usage Set the remote/peer IP address for this PPP link.

Use **”show address”** to view the currently set address.

Default values Based on the link type and ID, for more details see section 33.1.7.

33.3.14 PPP Authentication Protocols

Syntax `[no] auth-proto <pap chap mschap mschap-v2 | auto>`

Context PPP Advanced Configuration context

Usage Specify the allowed authentication protocols.

Use **”show auth-proto”** to view the currently allowed protocols.

Default values Auto, see section 33.1.5 for more details.

Example

Example

```
# only accept/agree to use pap
example:/config/pppoe-0/ppp-advanced/#>  auth-proto pap
example:/config/pppoe-0/ppp-advanced/#>

# accept/agree to use pap or chap
example:/config/pppoe-0/ppp-advanced/#>  auth-proto pap chap
example:/config/pppoe-0/ppp-advanced/#>
```

33.3.15 PPP Peer Authentication Method

Syntax `[no] aaa-method local-db <ID>`

Context PPP Advanced Configuration context (also as generic PPP setting in PPP Modem Configuration context for *dial-in* and *dial-in/out* modes).

Usage Specify the method used for peer authentication.

WeOS supports using local user databases for peer authentication. To create a local database see section 21.1.2.

Use **"`show aaa-method`"** to view the currently used peer authentication.

Default values Disabled.

33.3.16 PPP MPPE Crypto Settings

Syntax `[no] crypto <mppe-40 | mppe-128>`

Context PPP Advanced Configuration context

Usage Set the PPP link encryption.

Must only be used in combination with a one-way authenticated connection using some form of CHAP authentication (CHAP/MS-CHAP/MS-CHAPv2). See section 33.1.6 for more information.

Use **"`show crypto`"** to view the currently set encryption.

Default values Disabled.

33.3.17 PPP Proxy-ARP Settings

Syntax `[no] proxy-arp`

Context PPP Advanced Configuration context

Usage Enable or disable proxy ARP for this PPP link.

When **"`proxy-arp`"** is enabled, WeOS will proxy ARP requests for the peer's address under the following conditions:

- The peer has an address that belongs to the same subnet as the interface on which the ARP request is received.

- The aforementioned interface is up at the time when the PPP link is established.

Use **”show proxy-arp”** to view the current setting.

Default values Enabled.

33.3.18 PPP Dial-on-demand

Syntax `[no] demand <IDLE-TIMEOUT>`

Context PPP Advanced Configuration context

Usage Dial-on-demand, disconnect after idle timeout in seconds.

Use **”show demand”** to check the dial-on-demand setting for this PPP instance.

Default values Disabled (**”no demand”**)

33.3.19 PPP MRU

Syntax `[no] mru`

Context PPP Advanced Configuration context

Usage Enable maximum receive unit (MRU) negotiation.

If enabled, MRU parameters will be negotiated with the peer during the PPP link establishment phase.

The unit will use the PPP interface MTU value (configured or automatic) as the MRU presented to the peer.

A received MRU parameter from the peer will be acknowledged. The PPP interface MTU will be set in run-time to the lowest of the MTU value and the received MRU value.

See chapter 19 for information about MTU.

Use **”no mru”** to disable the MRU negotiation. No MRU parameter will be sent to the peer during the PPP link establishment phase, and any MRU parameter received from the peer will be rejected.

Use **”show mru”** to check the MRU setting for this PPP instance.

Default values Enabled (**”mru”**)

Chapter 34

GRE tunnels

WeOS supports Generic Routing Encapsulation (GRE) tunnels for IP over IP encapsulation. This chapter describes GRE tunnelling support in WeOS, while information on IP related settings of GRE interfaces is found in chapter 19.

34.1 Overview of GRE tunnel Properties and Management Features

Feature	Web	CLI	General Description
GRE Configuration			
Enable/disable GRE tunnel	X	X	
Tunnel Endpoints	X	X	Sections 34.1.1, 34.1.2
Tunnel TTL	X	X	Section 34.1.3
Outbound Interface	X	X	Section 34.1.4
GRE Status			
Show GRE Tunnel Status		X	

34.1.1 Introduction to GRE tunnels

GRE is an encapsulation method for tunnelling data packets over the IP protocol, and is specified in RFC 2784[8]. GRE can encapsulate arbitrary data packets, but

the most common use is to encapsulate IP packets, creating an *IP over IP* tunnel. WeOS only supports the encapsulation of IP packets in GRE.

GRE works by adding a special (GRE) header in front of the encapsulated packet containing a checksum[1], payload type (0x800 for IP) and some flags. The GRE header is preceded by an *outer* IP header used to route the packet between the tunnelling endpoints.

The GRE protocol is stateless. 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.

You can add security, if needed, by using GRE within an IPsec VPN tunnel (chapter 35) *or* by using some kind of secure protocol (such as HTTPS or SSH) for the data routed through the tunnel.

GRE tunnels are configured in two steps. First you need to define the tunnel with its endpoints and other related settings (described further sections 34.1.2-34.1.4). By configuring the tunnel, a new (GRE) network interface is created automatically. The second step is to configure the created GRE interface. See chapter 19 for information about configuring interfaces, including the *GRE interfaces*.

34.1.2 Defining GRE tunnel endpoints

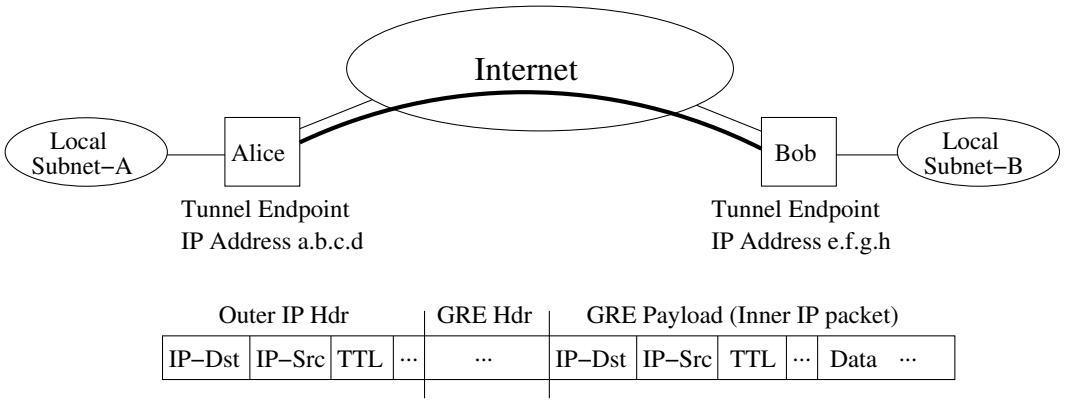


Figure 34.1: GRE tunnel example.

[1]The GRE checksum is optional. WeOS does not include a checksum in transmitted GRE packets

Fig. 34.1 shows a GRE tunnel example. The IP addresses in the *outer* IP header are the tunnel endpoints (*a.b.c.d* and *e.f.g.h*). The selection of IP addresses when defining GRE tunnel endpoints depends on the use case. Two common examples are described further in this section:

- *Generic use of GRE tunnels:* GRE can be used as a generic *IP-in-IP* tunnel. E.g., if Alice and Bob are NAT gateways, a GRE tunnel can be used to tunnel traffic between the local subnets (subnet-A and subnet-B.) The GRE tunnel endpoints (*a.b.c.d* and *e.f.g.h*) should be routeable IP addresses, and would typically be the public addresses of Alice and Bob (i.e., the Alice's and Bob's IP addresses on their respective interface towards the Internet).

- *Using GRE together with IPsec:* GRE can be used together with IPsec to enable an IPsec VPN to carry dynamic routing protocols such as OSPF. This enables the creation of robust IPsec VPNs capable of automatic failover to a redundant path if one connection fails. As of WeOS v4.17.1 redundant VPN solutions can be achieved by running two VPN gateways (IPsec, GRE, and OSPF) at each site as shown in fig. 34.2.

  In this case the IP addresses used for GRE tunnel endpoints should not be publicly routeable. Instead the IP address *a.b.c.d'* used by Alice1 would typically be an address within local subnet-A. To avoid problems when the local interface goes up/down, Alice1 could assign IP address *a.b.c.d'* as a secondary address to her loopback interface.

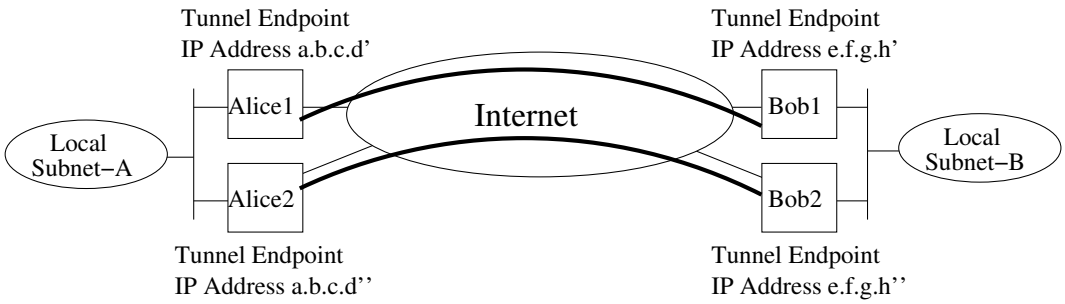


Figure 34.2: Redundant VPN solutions can be achieved by running two VPN gateways (IPsec/GRE/OSPF) at each site.

34.1.3 TTL of outer IP header

The TTL of the outer IP header (see fig. 34.1) is by default set equal to the TTL of the IP packet carried inside. It is possible to configure a specific TTL for the outer header for each GRE tunnel.

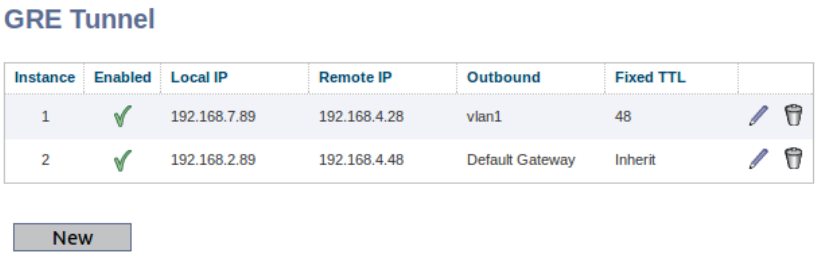
34.1.4 Restricting outbound interface for GRE traffic

By default, GRE traffic will be sent out through the interface leading towards the remote tunnel endpoint. The outbound interface is then selected on a per-packet basis by consulting the routing table (just like any other IP packet). It is also possible to configure the GRE tunnel to only allow traffic to go out via a specific network interface.

34.2 Managing GRE settings via the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ GRE

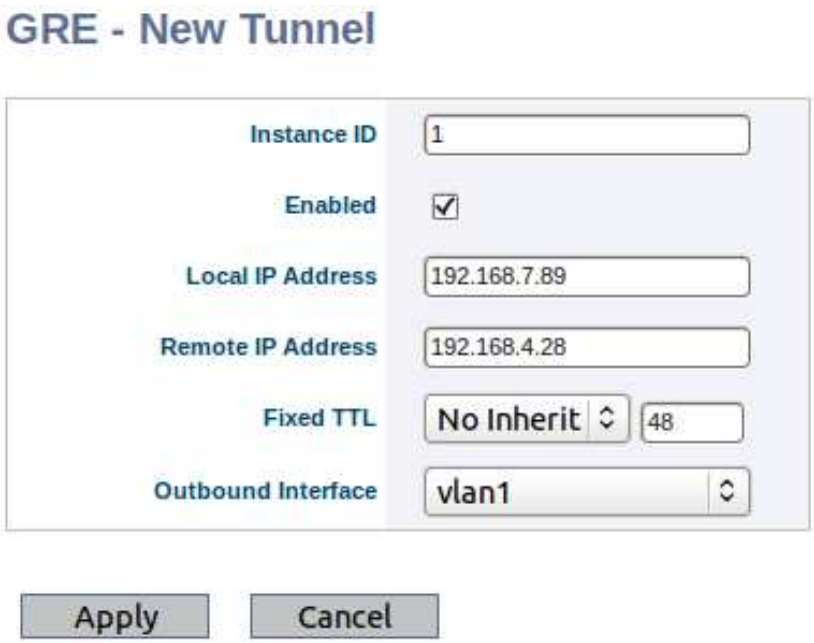
The main GRE configuration page lists the currently configured GRE instances.



Instance	A unique instance identifier for the GRE tunnel.
Enabled	In the overview table, a green check-box indicates the GRE tunnel is enabled, and a dash indicates disabled.
Local IP	The IP address assigned to the local endpoint of the GRE tunnel.
Remote IP	The IP address assigned to the remote endpoint of the GRE tunnel.
Outbound	The interface that will be used to send/receive GRE packets for this tunnel.
Fixed TTL	The TTL (Time to Live) to set on packets sent. If Inherit, the TTL value of the outbound interface will be used. Otherwise (No Inherit), the TTL value has to be assigned.
Edit	Click this icon to edit a GRE instance.
Delete	Click this icon to remove a GRE instance. You will be asked to acknowledge the removal before it is actually executed.
New	Click this button to create a new GRE instance.

34.2.1 Create a new GRE instance using the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ GRE ⇒ **New**



For description of fields, see section 34.2.

34.2.2 Edit GRE settings using the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ GRE ⇒ 

For description of fields, see section 34.2.

The Instance ID cannot be changed after creation.

34.3 Managing GRE settings via the CLI

The table below shows GRE management features available via the CLI.

Command	Default	Section
Configure GRE settings		
tunnel		Section 35.3.1
[no] gre <ID>		Section 34.3.1
[no] enable	Enabled	Section 34.3.2
[no] local <IPADDR>	Empty	Section 34.3.3
[no] remote <IPADDR>	Empty	Section 34.3.4
[no] outbound <IFNAME>	Auto	Section 34.3.5
[no] ttl <TTL>	Auto	Section 34.3.6
Show GRE Status		
show tunnel gre [ID]		Section 34.3.7

34.3.1 Managing GRE tunnels

Syntax `[no] gre <ID>` where ID is a number greater or equal to 1

Context Tunnel Configuration context

Usage Create, delete, or modify a GRE tunnel.

Use **”gre <ID>”** to create a new GRE tunnel, or to enter the GRE Tunnel Configuration context of an existing GRE tunnel. The ID affects the name of the interface created for this tunnel. Example: ID as 1 will create an interface named ”gre1”.

Use **”no gre <ID>”** to remove a specific GRE tunnel, or **”no gre”** to remove all configured GRE tunnels. This will also remove the corresponding interfaces AND their configurations!

Use **”show [gre [ID]]”** command within the Tunnel Configuration context. Also available as **”show”** command within the GRE Tunnel Configuration context, and as **”show tunnel [gre [ID]]”** within the Global Configuration context.

Default values Not applicable.

34.3.2 Enable/disable a GRE tunnel

Syntax `[no] enable`

Context GRE Tunnel Configuration context

Usage Enable, or disable this GRE tunnel.

Use **”enable”** to enable and **”no enable”** to disable a GRE tunnel.

Use **”show enable”** to show whether this GRE tunnel is enabled or disabled.

Default values Enabled

34.3.3 Local endpoint IP

Syntax `[no] local <IPADDR>`

Context GRE Tunnel Configuration context

Usage Set the local endpoint IP for the GRE packets in this tunnel.

This IP together with the remote endpoint IP will be used for the outer GRE packets carrying the tunnelled traffic. Data going out trough the tunnel from this node will be encapsulated in a GRE datagram using the local IP as source IP and the remote IP as destination IP. Incoming GRE datagrams with destination IP matching local IP and the source IP matching remote IP will be considered belonging to this GRE tunnel.

Use **”show local”** to show the configured local endpoint IP for this tunnel.

Default values None

34.3.4 Remote endpoint IP

Syntax `[no] remote <IPADDR>`

Context GRE Tunnel Configuration context

Usage Set the remote endpoint IP for the GRE packets in this tunnel.

This setting is used together with the local endpoint IP to specify the outer GRE packets. More info in section 34.3.3.

Use **”show remote”** to show the configured remote endpoint IP for this tunnel.

Default values None

34.3.5 Outbound interface

Syntax `[no] outbound <IFNAME>`

Context GRE Tunnel Configuration context

Usage Set the outbound interface of this tunnel.

Use this to set a specific interface that will be used for sending and receiving the GRE packets for this tunnel instance.

Use **”no outbound”** to automatically select the interface leading to the *default gateway* as outbound interface.

Use **”show outbound”** to show the configured outbound interface for this tunnel. **”Default Gateway”** is shown if the interface leading to the default gateway should be used as outbound interface.

Default values Auto (**”no outbound”**)

34.3.6 Time to Live setting

Syntax `[no] ttl <TTL>`

Context GRE Tunnel Configuration context

Usage Set the Time to Live parameter for the GRE packets

Use this to set a specific Time to Live (TTL) value that will be used in the IP header for the outbound GRE packets for this tunnel instance.

Use **”no ttl”** to use the TTL defined for the interface where the GRE packets are routed out.

Use **”show ttl”** to show the configured TTL value for this tunnel.

Default values Inherit (**”no ttl”**)

34.3.7 Show GRE Tunnel Status

Syntax `show tunnel gre [ID]`

Context Admin Exec context.

Usage Show the status for all or for a specific GRE tunnel.

Default values If no tunnel ID is specified, the status of all tunnels is shown.

Chapter 35

IPsec VPNs

WeOS provides virtual private network (VPN) support via IPsec VPNs. A WeOS switch can act as a VPN gateway in NETWORK-NETWORK and HOST-NETWORK scenarios. Configured as a VPN gateway, it can be used to securely connect branch office networks with a central office network, or to serve individual users wishing to ”dial in” securely over the Internet to the central office network, with their PC connected at some remote site. The data traffic will be protected by encrypted tunnels when sent over the Internet. A WeOS unit supports at most 25 simultaneous IPsec tunnels.

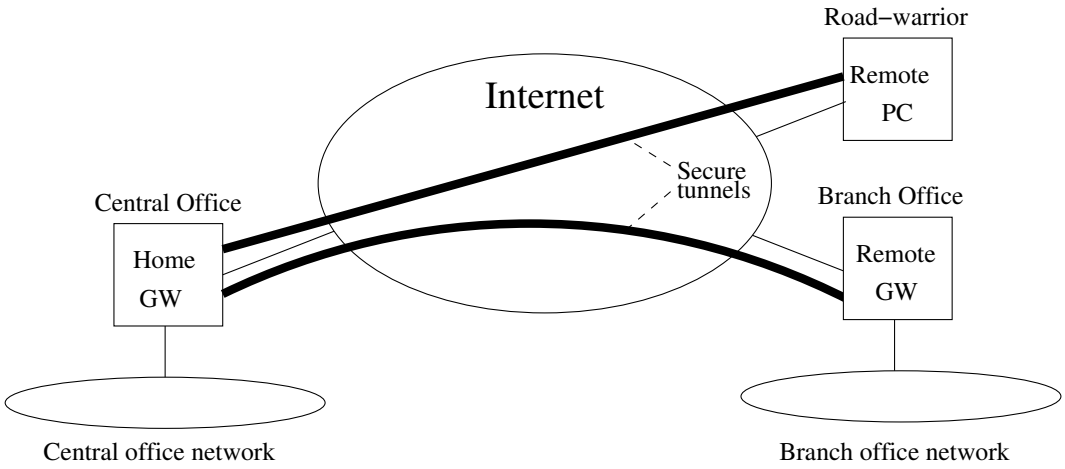


Figure 35.1: IPsec VPN tunnels can be used to securely connect hosts and networks over the Internet.

35.1 Overview of IPsec VPN Management Features

Feature	Web	CLI	General Description
VPN Configuration			
Add/Delete IPsec VPN tunnels	X	X	Section 35.1.1
Local/Remote Subnets	X	X	-"-
Local/Remote Protocol & Port		X	
Outbound Interface	X	X	-"-
NAT Traversal	X	X	-"-
IKEv1	X	X	Section 35.1.2
Role (Initiator/Responder)	X	X	-"-
Mode (Main/Aggressive)	X	X	Sections 35.1.2 and 35.1.6.1
IKE Authentication	X	X	Section 35.1.2
Pre-shared Key	X	X	Sections 35.1.2 and 35.1.6
Certificates	X	X	Sections 35.1.2 and 35.1.7
IKE Cipher Suite	X	X	Section 35.1.2
Identity	X	X	-"-
ESP Cipher Suite	X	X	-"-
Perfect Forward Secrecy	X	X	Section 35.1.3
MTU Override	X	X	Section 35.1.4
Dead Peer Detection	X	X	Section 35.1.5
VPN Status			
Show IPsec Tunnel Status	X	X	

35.1.1 Introduction to IPsec VPNs

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 NETWORK-NETWORK VPNs, and 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.

We use fig. 35.2 to explain some VPN related terminology.

- *Peers:* The two VPN gateways (Alice and Bob) are referred to as IPsec peers. The peers constitute the end-points of the secure tunnel. One of the peers

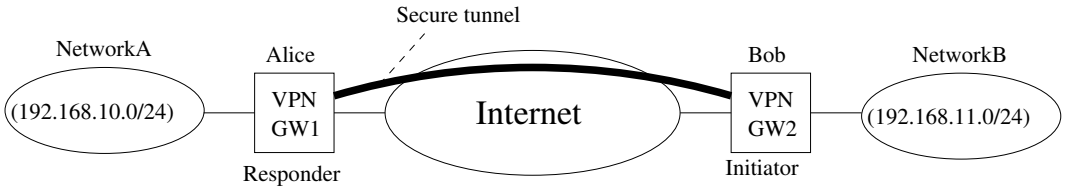


Figure 35.2: By establishing a secure IPsec Tunnel between the VPN gateways (Alice and Bob), traffic between Network-A and Network-B will be protected when sent across the Internet.

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 initiating the tunnel establishment by contacting the other peer - the *responder*. In fig. 35.2 we have assumed that Alice is the responder and Bob is the initiator.

  A WeOS switch configured as a VPN gateway is able to act both as *responder* (default) and as *initiator*.

- *NAT-traversal, Peer IP addresses and DDNS:* In order to act as a responder, Alice must be assigned a *public* (routable) IP address on its interface towards the Internet. Thus, Alice generally cannot be located behind a NAT gateway, since the initiator (Bob) would not be able to initiate the tunnel. Bob will need to know Alice's IP address (or domain name) in order to know where to send the tunnel establishment messages. If Alice is assigned a fixed IP address, Bob can choose between using Alice's IP address or her domain name. But if Alice gets her address dynamically (e.g., via DHCP), Bob should use her domain name to establish the contact. WeOS supports dynamic DNS (DDNS), thus Alice can dynamically register her current IP address, see section 19.3.3.

  The initiator (Bob) does not need to be assigned a public IP address. Bob is able to establish the tunnel even if he is located behind a NAT gateway, given that *NAT-traversal* (NAT-T) is enabled both in Alice's and Bob's VPN configurations.

  Furthermore, it is not mandatory for Alice to know Bob's IP address beforehand. It is possible to configure the VPN tunnel such that Bob could connect to the Internet at various locations and still be able to establish the VPN tunnel. This is commonly referred to as Bob being a *road warrior*.

- *Local and Remote Subnet:* Each peer will define what traffic should be al-

lowed to pass through the established tunnel. Each peer will define the local and remote subnet, and all traffic between these subnets is sent securely through the tunnel. To secure all traffic between networks ”A” and ”B”, Alice would define *192.168.10.0/24* as *local subnet*, and *192.168.11.0/24* as *remote subnet* in the tunnel configuration. Bob would do the opposite, i.e., define *192.168.11.0/24* as *local subnet*, and *192.168.10.0/24* as *remote subnet*.

More advanced settings for the local and remote subnet parameters are possible, e.g., it is possible to configure the tunnel so that all traffic from Network B is sent through the tunnel (i.e., not only the traffic heading for Network A).

- *Outbound interface:* The *outbound interface* denotes the interface, and implicitly the IP address, a VPN gateway uses to tunnel the traffic through, and to communicate with its peer. In fig. 35.2 Alice *outbound interface* would be her interface towards the Internet (and the same goes for Bob).

  By default, the *outbound interface* is set to the interface leading to the *default gateway* (see section 19.3).

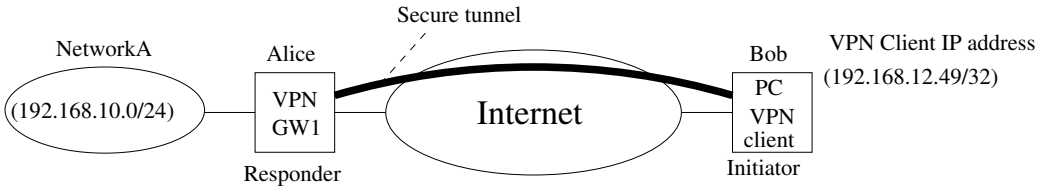


Figure 35.3: IPsec VPNs can be used to provide secure connections between individual hosts and a network behind a VPN gateway, a HOST-NETWORK VPN.

Another common use case is shown in fig. 35.3. In this case Bob is an individual host, i.e., a PC with VPN client software installed. A WeOS switch is able to act as VPN gateway in HOST-NETWORK scenarios. The host (Bob) should be assigned a VPN client IP address (*192.168.12.49* in fig. 35.3), which is used to communicate with the hosts in Network-A. For Alice the configuration is very similar to the NETWORK-NETWORK example above, with the main difference being that her remote-subnet defines an individual IP address (*192.168.12.49/32*, i.e., *netmask 255.255.255.255*) instead of a network. As in the NETWORK-NETWORK use case, Bob’s PC can be configured as a *road warrior* connecting from different IP addresses, and with NAT-T enabled he can connect from behind a NAT gateway.

35.1.2 Authenticated Keying using Internet Key Exchange (IKE)

As part of the IPsec VPN tunnel establishment Alice and Bob will use the IKE (Internet Key Exchange) protocol to authenticate each other and create necessary session keys to protect the data traffic. WeOS supports IKE version 1 (IKEv1) with authentication through *pre-shared keys* (PSK) or *certificates* (RSA signature keys using X.509 certificates). In IKEv1 there are two authentication handshakes (phase-1 and phase-2):

- IKE phase-1 handshake: In this document the IKE phase-1 handshake is simply referred to as the *IKE handshake*. In the IKE handshake Alice and Bob identify themselves and use their configured PSK or certificates to authenticate each other. When configuring an IPsec tunnel, the identities of the peers should be defined. Five methods are provided:
  - Distinguished name (ID_DER_ASN1_DN): (Only applicable for certificate based authentication). The distinguished name (DN) of an X.509 certificate, e.g., ”/C=US/O=ACME/CN=foobar” can be used as identification. The DN string can also be specified in LDAP style (e.g., ”C=US, O=ACME, CN=foobar”). The responder would typically use wild-card (e.g., ”C=US, O=ACME, CN=*”) to allow multiple road-warriors to establish *tunnel sessions* via a single *tunnel configuration*.
  - IP Address (ID_IPV4_ADDR): If the IP address of the peer is known, it can be used to identify it. When using *main mode* with PSK (*main* and *aggressive* modes are explained later in this section) this is the only option. When using IP address as IKE identity, WeOS allows you to specify either an IP address or a domain name, which is then *resolved* via DNS.
  - Domain name (ID_FQDN): The identification can be specified as the domain name of the peer. When specifying *type* ”domain name”, the entered identity value (e.g., *foobar.example.com*) is sent *as is*, i.e., it is **not** resolved to an IP address. Therefore, the domain name identification type could be used as a general user name, such as *foobar*.
  - Email style (ID_USER_FQDN): The identification can be specified in email address style, e.g., *foobar@example.com*.
  - Key identification (ID_KEY_ID): (Only applicable for PSK based authentication) With the key identification type, the identification can be entered as an opaque byte stream. As with the domain name type, the key identification type can be used to enter a general user name, such

as *foobar*.

The IKE handshake also creates the necessary credentials for the following ESP handshake.

- IKE phase-2 handshake: In this document the IKE phase-2 handshake is referred to as the *ESP handshake*. In the ESP handshake the *cipher suite* for the VPN tunnel is negotiated as well as the *session keys* used to encrypt and integrity protect the data send through the tunnel.

The user can also specify whether the IKE handshake should use the *main* (default) or *aggressive* mode. Not all combinations are supported:

- Pre-shared key: With PSK authentication, either *main* or *aggressive* mode can be used. However, due to limitations in IKEv1, PSK with main mode can only be used with IP address as identity, which in turn implies that the initiator must have a fixed IP address (no road-warrior).
- Certificates: As of WeOS v4.17.1, certificate based authentication is only supported in main mode.

A summary of supported combinations is shown below. *IKEv1 main mode with certificates* is recommended.

IKE Phase-1 handshake	**Authentication Method**	
	Certificate	**Pre-shared Key**
Main mode	**Recommended** Supports Road-warrior and fixed setups	Fixed setups No road-warrior
Aggressive mode	**Not supported**	Supports Road-warrior and fixed setups

Both for the IKE and ESP handshakes the user can specify which cryptographic protocols to use. The following algorithms are supported by WeOS:

- *Encryption algorithm:* Supported encryption algorithms are *3DES* and *AES* (key length 128 and 256 bits).
- *Message authentication/integrity:* Supported hash algorithms for message authentication are *MD5*, and *SHA-1*.
- *Diffie-Hellman groups:* Supported Diffie-Hellman groups are 1024 (DH group 2), 1536 (DH group 5), 2048 (DH group 14), 3072 (DH group 15) and 4096 (DH group 16).

These Diffie-Hellman key exchange groups are supported and are configurable for both IKE and ESP (for PFS) individually.

When using IKE *main* mode, Alice and Bob can be configured to automatically negotiate a suitable cipher suite. When using *aggressive* mode, Alice and Bob should be configured to use a specific cipher suite (same at both sides). When aggressive mode is selected, WeOS by default uses the suite *AES128-SHA1-DH1024*.

35.1.3 Perfect Forward Secrecy

Perfect Forward Secrecy (PFS) refers to the property that if an ESP session key is compromised, the attacker will only get access to the data protected by that single key. Previous and later session keys will not be revealed just because that single key was compromised, thus data encrypted by those keys is still protected.

Note
This setting is not supported by all IPsec implementations. It is however recommended to have it enabled, on both sides of the connection.

PFS uses Diffie-Hellman to exchange new session keys. The Diffie-Hellman group can be automatically selected or manually configured.

PFS with automatic Diffie-Hellman group selection is enabled by default on all new tunnels.

If you are unsure what do to, you can safely disable PFS. If the IPsec daemon receives a request with PFS, it will allow it despite PFS being disabled or not.

35.1.4 Data encapsulation and encryption

IPsec specifies two modes to encapsulate the data, a *transport* and a *tunnel* mode. WeOS IPsec VPN only supports the *tunnel* mode. In the tunnel mode, the original IP packets are encapsulated within another IP packet as shown in fig. 35.4.

In IPsec there is also the choice by protecting the data using *AH* (Authentication Header), and ESP (Encapsulating Security Payload) formats. WeOS only supports ESP, which is the format to use to achieve both data *encryption* and *integrity* protection.

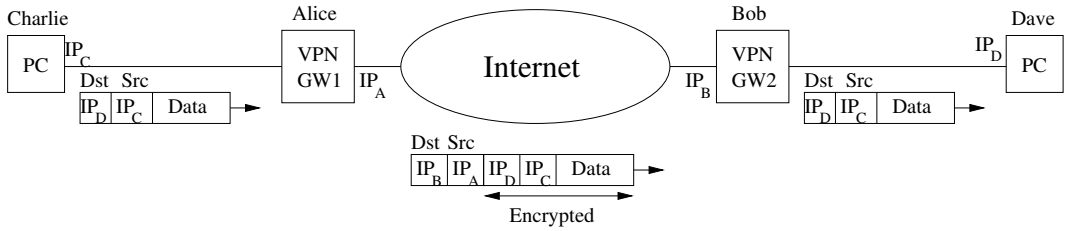


Figure 35.4: IPsec tunnel mode encapsulation. The ”inner” IP header holds the original IP addresses of Charlie and Dave, and the outer IP header contains the addresses of the VPN gateways Alice and Bob.

In order to send encapsulated data more efficiently over the Internet an operator can tune the maximum transmission unit (MTU) for VPN tunnels. By default the MTU for VPN tunnels is set to 1419 bytes.

35.1.5 Dead Peer Detection

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. There are three configuration options for the DPD action:

- *Restart:* An initiator should try to reestablish an IPsec tunnel by restarting the IKE handshake.
- *Hold:* A responder can chose the *Hold* DPD action. This is often the preferred option in a NETWORK-NETWORK VPN scenario (see fig. 35.2).
- *Clear:* A responder can also chose the *Clear* DPD action. This is the preferred option if the HOST-NETWORK VPN scenario, i.e., if the initiator is a single road warrior (see fig. 35.3), but *Clear* may also be used in a NETWORK-NETWORK VPN scenario.

As of WeOS v4.17.1 a VPN gateway configured as initiator will use DPD action *restart* by default, while a responder by default uses DPD action *clear*.

Two additional DPD parameters can be configured:

- DPD Delay: The DPD delay is the interval between DPD probing messages sent by a VPN gateway.
- DPD Timeout: If a period corresponding to the DPD timeout elapses without getting any response on the DPD probe messages, the VPN gateway considers the peer to be down.

The DPD settings can be configured individually on each peer. It is even possible to disable DPD on one of the peers - that peer will still respond to DPD probing messages from the other peer.

35.1.6 Examples of using IPsec VPN with PSK

This section illustrates configuration steps when configuring IPsec VPNs using IKE authentication with pre-shared key (PSKs).

Fig. 35.5 shows a sample IPsec VPN topology which can be used to illustrate VPN configuration steps. This is the same topology as shown in the NET-NET example in fig. 35.2, but with some more details on the inbound and outbound interface of each VPN gateway.

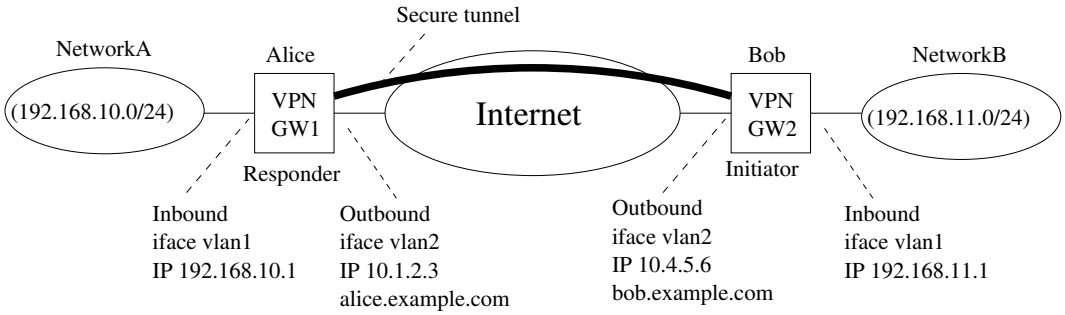


Figure 35.5: Example VPN topology used to illustrate configuration steps.

We have two VPN gateways, Alice and Bob, which are used to establish a secure VPN tunnel between the central office network (192.168.10.0/24) and the branch office network (192.168.11.0/24).

When using pre-shared key authentication, we first need to determine if Bob's outbound interface has a fixed address or not. This affects the choice of IKE *main mode* or *aggressive mode*, as discussed in section 35.1.6.1. Sections 35.1.6.2 and 35.1.6.3 explain the configuration steps if *aggressive mode* or *main mode* is used.

35.1.6.1 Selecting Aggressive or Main Mode?

An IPsec tunnel must specify whether IKE should operate in *main mode* or in *aggressive mode* (in WeOS v4.17.1 *main mode* is used by default).

As mentioned in section 35.1.2, the IKE *main mode* with PSK authentication is limited to IP address as peer identification. This in turn means that IKE *aggressive mode* should be used if the *initiator's IP address is not fixed*, e.g., if Bob may change location (road warrior), or if he is using DHCP to acquire his address on the outbound interface. For a description of establishing the VPN topology in fig. 35.5 with IKE *aggressive mode*, see section 35.1.6.2.

On the other hand, if Bob has a fixed IP address, the setup in fig. 35.5 could be established either with IKE *main mode* or *aggressive mode*. *Main mode* is somewhat simpler to configure, and is described in section 35.1.6.3.

35.1.6.2 Aggressive Mode Configuration

Below you find hints on how to configure the *initiator* (Bob) and *responder* (Alice) in IKE aggressive mode. Note: this is just an example; several alternatives exist.

Many VPN settings can be configured in the same way on the *responder* (Alice) and the *initiator* (Bob):

- VPN instance number: This number is of local significance only, i.e., it can differ on Alice and Bob. In the Web configuration, it is simplest to accept the suggested value.
- Enable the VPN tunnel: Yes (default)
- Outbound interface: Default gateway (or "vlan2")
- Aggressive mode: Yes
- IKE (phase-1) cipher suite: With aggressive mode, a specific cipher suite must be specified (auto-mode is not possible). Simplest is to use the default settings: AES-128 for encryption, SHA1 for authentication, and group DH 2 (1024) for the Diffie-Hellman exchange.
- Pre-shared secret: The common password, e.g., "TopSecret123!", which should be known only by Alice and Bob.
- ESP cipher suite: With aggressive mode, a specific cipher suite must be specified (auto-mode is not possible). Simplest is to use the default settings:

AES-128 for encryption, SHA1 for authentication, and automatic Diffie-Hellman group (for PFS)

- Enable PFS: Yes.
- DPD Delay: 30 seconds (default)
- DPD Timeout: 120 seconds (default)

Responder specific settings (Alice):

- Remote Peer: Any (not necessary to know the IP address of Bob)
- Local subnet: 192.168.10.0; netmask: 255.255.255.0
- Remote subnet: 192.168.11.0; netmask: 255.255.255.0
- Role: Responder (no initiator)
- Local-id: Type ”Name (DNS/User)”, Identifier ”Alice”
- Remote-id: Type ”Name (DNS/User)”, Identifier ”Bob”
- DPD Action: Clear

Initiator specific settings (Bob):

- Remote Peer: 10.1.2.3 (or *alice.example.com*)
- Local subnet: 192.168.11.0; netmask: 255.255.255.0
- Remote subnet: 192.168.10.0; netmask: 255.255.255.0
- Role: Initiator
- Local-id: Type ”Name (DNS/User)”, Identifier ”Bob”
- Remote-id: Type ”Name (DNS/User)”, Identifier ”Alice”
- DPD Action: Restart

35.1.6.3 Main Mode Configuration

Below you find hints on how to configure the *initiator* (Bob) and *responder* (Alice) in IKE main mode. Note: this is just an example; several alternatives exist.

Many VPN settings can be configured in the same way on the *responder* (Alice) and the *initiator* (Bob):

- VPN instance number: This number is of local significance only, i.e., it can differ on Alice and Bob. In the Web configuration, it is simplest to accept the suggested value.
- Enable the VPN tunnel: Yes (default)
- Outbound interface: Default gateway (or ”vlan2”)
- Aggressive mode: No (i.e., use main mode)
- IKE (phase-1) cipher suite: Auto (simplest)
- Pre-shared secret: The common password, e.g., ”TopSecret123!”, which should be known only by Alice and Bob.
- ESP cipher suite: Auto (simplest)
- Enable PFS: Yes.
- DPD Delay: 30 seconds (default)
- DPD Timeout: 120 seconds (default)

Responder specific settings (Alice):

- Remote Peer: 10.4.5.6 (”Any” can not be used; Domain name *bob.example.com* can not be used either.)
- Local subnet: 192.168.10.0; netmask: 255.255.255.0
- Remote subnet: 192.168.11.0; netmask: 255.255.255.0
- Role: Responder (no initiator)
- Local-id: Auto (or type ”IP Address”, Identifier ”10.1.2.3”)
- Remote-id: Auto (or type ”IP Address”, Identifier ”10.4.5.6”)
- DPD Action: Hold

Initiator specific settings (Bob):

- Remote Peer: 10.1.2.3 (or *alice.example.com*)
- Local subnet: 192.168.11.0; netmask: 255.255.255.0
- Remote subnet: 192.168.10.0; netmask: 255.255.255.0
- Role: Initiator
- Local-id: Auto (or type ”IP Address”, Identifier ”10.4.5.6”)

- Remote-id: Auto (or type ”IP Address”, Identifier ”10.1.2.3” or ”alice.example.com”)
- DPD Action: Restart

35.1.7 Use of certificates for IKE authentication

WeOS supports IKE authentication via *certificates* and *pre-shared keys* (PSKs), with certificate based authentication as *recommended* method. While PSK based authentication can be somewhat simpler to configure, certificate based authentication is often considered more secure, and makes it easier to manage setups with multiple road-warriors.

This section provides additional hints when using certificate based authentication of IPsec tunnels in WeOS.

1. *Load/import certificates:* To use certificates for IKE based authentication you must first create/acquire certificates and private keys, and load them onto your WeOS unit(s). See section 7.1.8 for more information on loading/importing certificates onto your WeOS unit.
2. *Use case and PKI model:* What certificates to load onto your WeOS unit will depend on your specific use case. Three common use cases supported by WeOS.
   - *Common CA:* Alice (IPsec Responder, typically a VPN Gateway), Bob (IPsec Initiator/VPN PC client or gateway) use a common CA. This would be a typical scenario when a company wish to allow their employees or branch offices to connect securely to the central office. See section 35.1.7.1 for more information.
   - *Different CAs:* Alice and Bob have certificates issued by different CAs. This would be a typical scenario when you wish to communicate securely between units of different organisations. See section 35.1.7.2 for more information.
   - *Trusted Peer:* Alice and Bob can import each others certificates. This approach does **not** require Alice and Bob to install each others CA certificates. In a way this case is similar to using PSKs, although a bit more secure. See section 35.1.7.3 for more information.
3. *Verify/set time on unit:* As certificates are valid for a certain time period (start time and end time), it is important that the date/time is set correctly on your WeOS unit. You can set the time manually (see chapter 20), but

it is recommended to use SNTP/NTP (see sections 19.3.2, 19.5.2 (Web), and 19.7.22 (CLI)) as the date/time can be reset to Unix epoch (January 1, 1970) if left without power for some time.

4. *Defining local and remote IKE identities:* For Alice and Bob to identify each other using certificates, use of *Distinguished Name*(ID_DER_ASN1_DN) is recommended. As stated in section 35.1.2, identity methods *domain name* (ID_FQDN), *email* (ID_USER_FQDN), and *IP address* (ID_IPV4_ADDR) are possible too, but requires the specific identity to be included as *subjectAltName* in the certificate. E.g., if Bob wish to wish to identify himself as *bob@example.com* (email style), his certificate needs to include "subjectAltName=email:bob@example.com", and he should set **"local-id email bob@example.com"** in his IPsec tunnel configuration. Correspondingly, Alice would set **"remote-id email bob@example.com""** in her IPsec tunnel configuration.

   For examples using *Distinguished Name* as identity, see sections 35.1.7.1-35.1.7.3.

   Using "auto" for the local-id setting (**"no local-id"**) together with certificate based authentication means that Alice will identify herself with the *ID_DER_ASN1_DN* method, and automatically extract her DN string value from her certificate.

   Warning on using "auto" mode for "remote-id"

   As of WeOS v4.17.1 use of "auto" mode for "remote-id" together with certificate authentication is discouraged. That option may change behaviour or even be removed in future versions of WeOS, thus its use will pose risks when doing future upgrades. (Use of "auto" mode with PSK authentication is fine, though).

   Further details: when using certificates in WeOS v4.17.1, if Alice uses "auto"-mode to identify Bob (**"no remote-id"**) WeOS will expect Bob to identify himself using method:

   - "ID_DER_ASN1_DN" when no peer IP address or domain name is set (she considers Bob to be a road-warrior (**"no peer"**)). Furthermore, there will be no restriction on **what** DN string Bob presents as long as his certificate is valid and issued by a trusted CA.
   - "ID_IPV4_ADDR" when a peer IP address or domain name is set (e.g., **"peer 1.2.3.4"**). Thus, in this case Bob would have to include the corresponding IP address in the certificate (e.g., "subjectAltName=IP:1.2.3.4")

and set his local-id accordingly (**”local-id inet 1.2.3.4”**).

5. *Defining local and remote IP subnets:* By using DN strings with common name (CN) wild-card, a VPN gateway can easily serve multiple road-warriors using a single IPsec tunnel. E.g., if Alice (IPsec Responder/VPN Gateway) use DN string, *C=US, O=ACME, CN=*”* as remote-id, it would match certificates with different CNs (e.g., Bob or Charlie) as long as the other relative distinguished names (RDNs), here *C=US, O=ACME*, of the presented certificate would match.

   However, if Alice is to allow multiple VPN peers to connect via a single tunnel definition, she should allow each peer to have a *local subnet* (or *virtual IP*) corresponding to *a part* of her configured *remote subnet*, i.e, her remote subnet should be shared by Bob, Charlie or any other valid peer. An example is shown in the figure below, where Alice has declared her remote subnet *10.0.2.0/24* as *shared* to allow Bob, Charlie and Dave to connect.

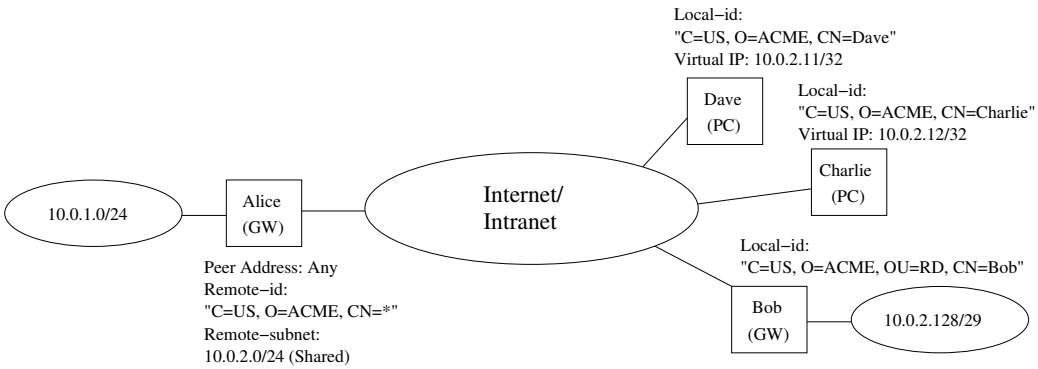


Figure 35.6: By defining the remote subnet as ”shared”, one IPsec tunnel definition at the responder (Alice) can serve multiple initiators (Bob, Charlie, and Dave).

35.1.7.1 Common CA: IKE certificates within an organisation

When a company wish to use IPsec with certificate authentication within their organisation, all entities (IPsec VPN gateways and users of VPN clients) can have their certificate issued by the *same* CA. The CA can either be operated by the company itself, or an external (professional) CA organisation.

In this user scenario, a VPN unit such as Alice will have to upload/import

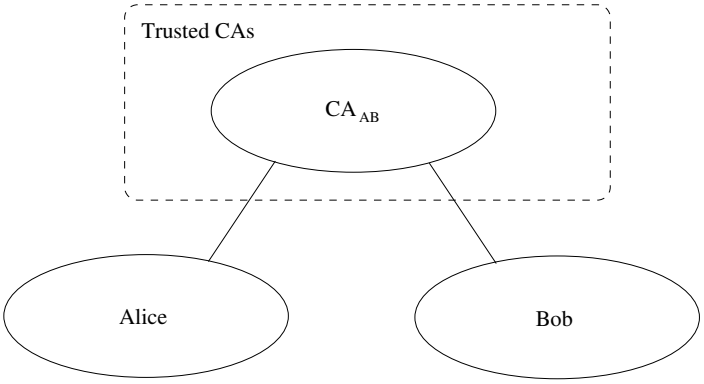


Figure 35.7: Alice and Bob have certificates issued by the same CA (e.g., their company CA). In this PKI model, Alice uploads the certificate of her CA, and trusts any certificate issued by that CA.

- the certificate of her CA (CA_{AB}),
- her own certificate (AliceCert), and
- the private key associated with her certificate.

This is typically done by importing a password protected *PKCS#12 bundle*, holding both these certificates and the private key (see section 7.1.8 for more information on certificate management).

If we consider the sample setup in fig. 35.6, the certificates of Alice, Bob, Charlie, and Dave could all be issued by the same CA. Below we see sample WeOS CLI syntax for Alice's and Bob's VPN configuration, as well as some comments.

- *Local-id:* The local-id strings are not necessary here; using the 'auto' mode (**"no local-id"**) is sufficient, since the default is to use the DN string of the local certificate in certificate authentication mode is used (**"method cert"**).
- *Shared remote-subnet:* As Bob's local subnet (10.0.2.128/29) only defines a subset of the remote subnet defined by Alice (10.0.2.0/24), she has added the keyword **"shared"**.
- *Remote CA:* The setting **"remote-ca same"** enforces the restriction that Alice will verify that Bob's certificate is issued by the same CA as her certificate (and vice versa). This is the default setting, and may not be shown in your configuration file. See sections 35.1.7.2 and 35.1.7.3 for alternative settings.

- *Remote Cert:* In this scenario, Alice would accept all initiators (Bob, Charlie, Dave, etc.) with a certificate issued by their common CA, and where the DN string matches "C=US, O=ACME, CN=*". The remote certificate only needs to be specified in the *trusted peer* use case, see section 35.1.7.3. The default setting is **"no remote-cert"**, thus this line may not be shown in your configuration file.
- *Peer IP address*: Alice is configured to accept initiators irrespective of their IP address. Bob needs to be configured with Alice's "Internet" IP address or domain name as peer (here *10.10.1.2*; not shown in fig. 35.6).

Example

Alice's Configuration

```
tunnel
  ipsec 0
    enable
    no aggressive
    pfs
    no ike
    no esp
    no peer
    no outbound
    local-id dn "C=US, O=ACME, CN=Alice"
    remote-id dn "C=US, O=ACME, CN=*"
    local-subnet 10.0.1.0/24
    remote-subnet 10.0.2.0/24 shared
    method cert
    local-cert AliceCert
    no remote-cert
    remote-ca same
    no initiator
    dpd-action clear
    dpd-delay 30
    dpd-timeout 120
    sa-lifetime 28800
    ike-lifetime 3600
    end
  end
```

Bob's Configuration

```
tunnel
  ipsec 0
    enable
    no aggressive
    pfs
    no ike
    no esp
    peer 10.10.1.2
    no outbound
    local-id dn "C=US, O=ACME, CN=Bob"
    remote-id dn "C=US, O=ACME, CN=Alice"
    local-subnet 10.0.2.128/29
    remote-subnet 10.0.1.0/24
    method cert
    local-cert BobCert
    no remote-cert
    remote-ca same
    initiator
    dpd-action restart
    dpd-delay 30
    dpd-timeout 120
    sa-lifetime 28800
    ike-lifetime 3600
    end
  end
```

35.1.7.2 Different CAs: IKE certificates with multiple organisations

As of WeOS v4.17.1, this use case can only be configured via the CLI.

To use IPsec to establish secure tunnels between users or units of different organisations, Alice and Bob will usually have certificates issued by *different* CAs. In this case, Alice would upload/import Bob's CA certificate (C_B), and would thereby trusted all certificates issued by Bob's CA.

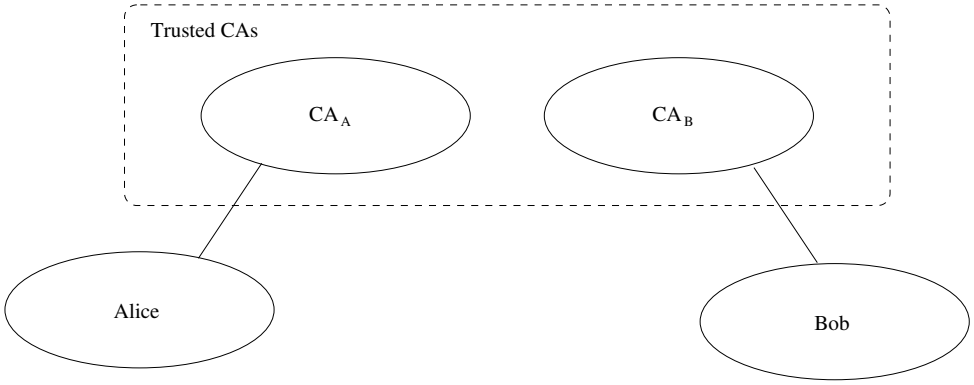


Figure 35.8: Alice and Bob have certificates issued by different CAs (e.g., their respective company CA). In this PKI model, Alice uploads the certificate of her CA (CA_A), and Bob's CA (CA_B), and trusts any certificate issued by either of them CA.

In this user scenario, a VPN unit such as Alice will have to upload/import

- the certificate of her CA (CA_A),
- the certificate of Bob's CA (CA_B),
- her own certificate (AliceCert), and
- the private key associated with her certificate.

Alice would typically upload/import her private key, her CA and own certificates as a password protected *PKCS#12 bundle*, while Bob's CA certificate could be uploaded/imported as a PEM file. See section 7.1.8 for more information on certificate management).

If we consider the sample setup in fig. 35.6, the certificates of Alice and Bob would now be issued by different CAs. Below we see sample WeOS CLI syntax for Alice's and Bob's VPN configuration, as well as some comments.

- *Remote CA:* The setting **"remote-ca dn 'C=US, O=FOOBAR, CN=FoobarCA'"** in Alice's configuration restricts initiators to have certificates issued by the "FoobarCA" (Bob's CA). An alternative would be to use the setting **"remote-ca any"**, which would allow initiators with valid certificates issued by any CA trusted by Alice.

  Correspondingly, Bob is configured to only trust certificates issued by "AcmeCA" (Alice's CA).

  As of WeOS v4.17.1, the *Remote CA* setting is only configurable via the CLI,

thus this use case cannot be configured via the Web interface. However, a similar service can be achieved via the *trusted peer* use case, see section 35.1.7.3.

- For comments on other settings, see the related example in section 35.1.7.1.

Example

Alice's Configuration

```
tunnel
  ipsec 0
    enable
    no aggressive
    pfs
    no ike
    no esp
    no peer
    no outbound
    local-id dn "C=US, O=ACME, CN=Alice"
    remote-id dn "C=US, O=FOOBAR, CN=*"
    local-subnet 10.0.1.0/24
    remote-subnet 10.0.2.0/24 shared
    method cert
    local-cert AliceCert
    no remote-cert
    remote-ca dn "C=US, O=FOOBAR, CN=FoobarCA"
    no initiator
    dpd-action clear
    dpd-delay 30
    dpd-timeout 120
    sa-lifetime 28800
    ike-lifetime 3600
    end
  end
```

Bob's Configuration

```
tunnel
  ipsec 0
    enable
    no aggressive
    pfs
    no ike
    no esp
    peer 10.10.1.2
    no outbound
    local-id dn "C=US, O=FOOBAR, CN=Bob"
    remote-id dn "C=US, O=ACME, CN=Alice"
    local-subnet 10.0.2.128/29
    remote-subnet 10.0.1.0/24
    method cert
    local-cert BobCert
    no remote-cert
    remote-ca dn "C=US, O=ACME, CN=AcmeCA"
    initiator
    dpd-action restart
    dpd-delay 30
    dpd-timeout 120
    sa-lifetime 28800
    ike-lifetime 3600
    end
  end
```

35.1.7.3 IKE with trusted peer certificates

As an alternative to installing trusted CA certificates, Alice and Bob can import each others certificates and use as *trusted peers*.

In this user scenario, a VPN unit such as Alice will have to upload/import

- Bob's certificate (BobCert),
- her own certificate (AliceCert), and
- the private key associated with her certificate.

In most cases Alice would also import her CA certificate (CA_A), although this is not required for this trust model. Typically she would then upload/import her private

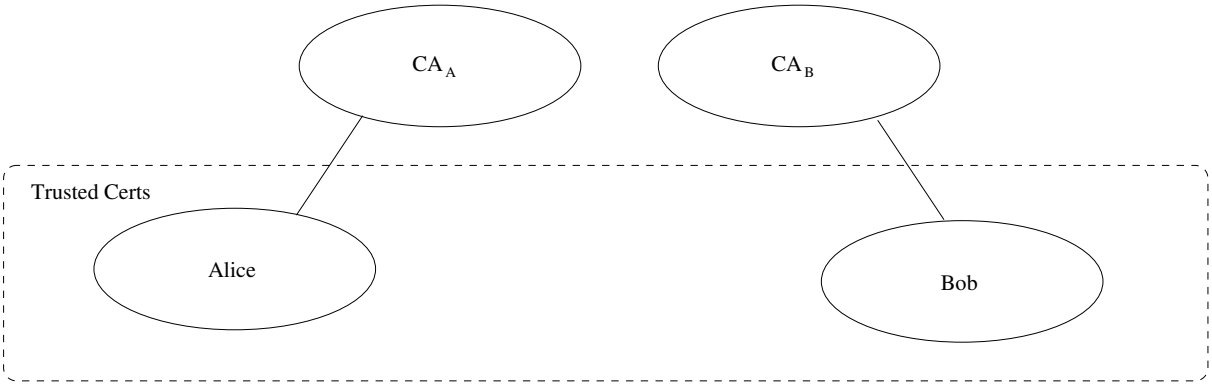


Figure 35.9: Alice and Bob have imported each others certificates as trusted peers. In this case Alice and Bob do not need to install/import CA certificates.

key, her CA and own certificates as a password protected *PKCS#12 bundle*, while Bob's certificate could be uploaded/imported as a PEM file. See section 7.1.8 for more information on certificate management).

Note
Although this trust model does not require Alice or Bob to install any CA certificates, WeOS still requires their certificates to be issued by some CA, i.e., the *Issuer* and *Subject* of the certificate cannot be the same.

The configuration example below is loosely based on sample setup in fig. 35.6. However, as this tunnel configuration is only intended for Alice and Bob, we have restricted the *remote-id* and *remote-subnet* settings on Alice side. Furthermore, we have let Alice and Bob have certificates of different CAs, to make the example more general.

- *Local-id:* Local-id could use "auto" mode (**"no local-id"**). That is simpler than defining the DN string explicitly as done below.
- *Remote-id:* As of WeOS v4.17.1, Remote-id can **not** use "auto" mode (**"no remote-id"**). That may change in future versions of WeOS.
- *Remote CA:* The *remote-ca* setting does **not** apply when a remote certificate is specified, thus is not shown in the example.

Example

Alice's Configuration

```
tunnel
  ipsec 0
    enable
    no aggressive
    pfs
    no ike
    no esp
    no peer
    no outbound
    local-id dn "C=US, O=ACME, CN=Alice"
    remote-id dn "C=US, O=FOOBAR, CN=Bob"
    local-subnet 10.0.1.0/24
    remote-subnet 10.0.2.0/29
    method cert
    local-cert AliceCert
    remote-cert BobCert
    no initiator
    dpd-action clear
    dpd-delay 30
    dpd-timeout 120
    sa-lifetime 28800
    ike-lifetime 3600
    end
  end
```

Bob's Configuration

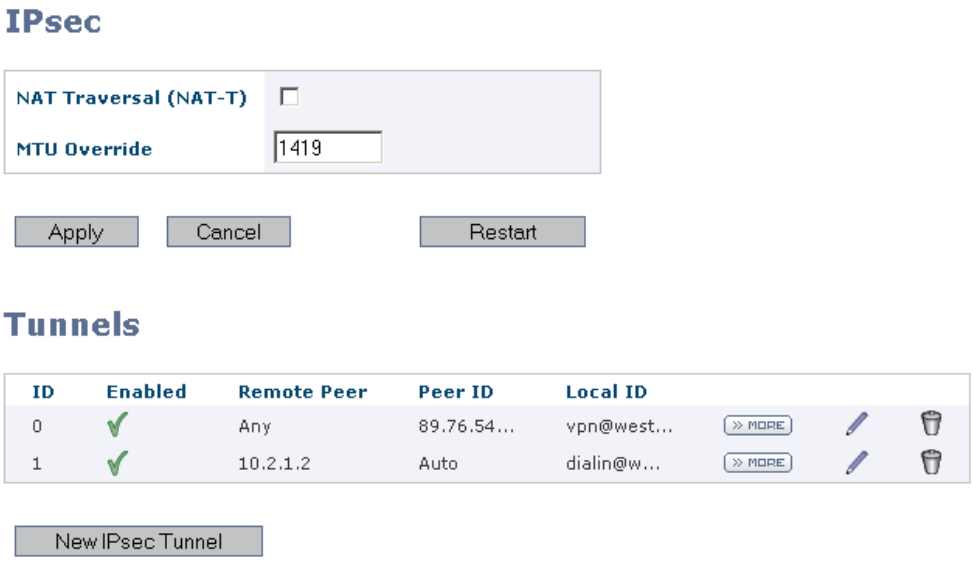
```
tunnel
  ipsec 0
    enable
    no aggressive
    pfs
    no ike
    no esp
    peer 10.10.1.2
    no outbound
    local-id dn "C=US, O=FOOBAR, CN=Bob"
    remote-id dn "C=US, O=ACME, CN=Alice"
    local-subnet 10.0.2.128/29
    remote-subnet 10.0.1.0/24
    method cert
    local-cert BobCert
    remote-cert AliceCert
    initiator
    dpd-action restart
    dpd-delay 30
    dpd-timeout 120
    sa-lifetime 28800
    ike-lifetime 3600
    end
  end
```

35.2 Managing VPN settings via the web interface

35.2.1 Manage IPsec VPN via the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ IPsec

The main IPsec VPN configuration pages contains two parts: the top part lists general IPsec settings applying to all ports, the bottom part shows a list of currently configured IPsec tunnels.



General IPsec settings:

NAT Traversal (NAT-T)	Enable NAT traversal support by checking the check box, disable NAT traversal support by un-checking the checkbox. The NAT-traversal setting will apply to all IPsec tunnels. NAT Traversal can cause inter-operability problems with some IPsec clients, so the default setting is disabled. However, when NAT-T is enabled it only kicks in when the server and client detects they are being NAT'ed. So in most cases it is a safe option to set.
MTU Override	Specify the maximum transfer unit for IPsec packets. The setting affects all IPsec tunnels.
Restart	Click this button to restart the IPsec daemon. All IPsec tunnels will be torn down and restarted.

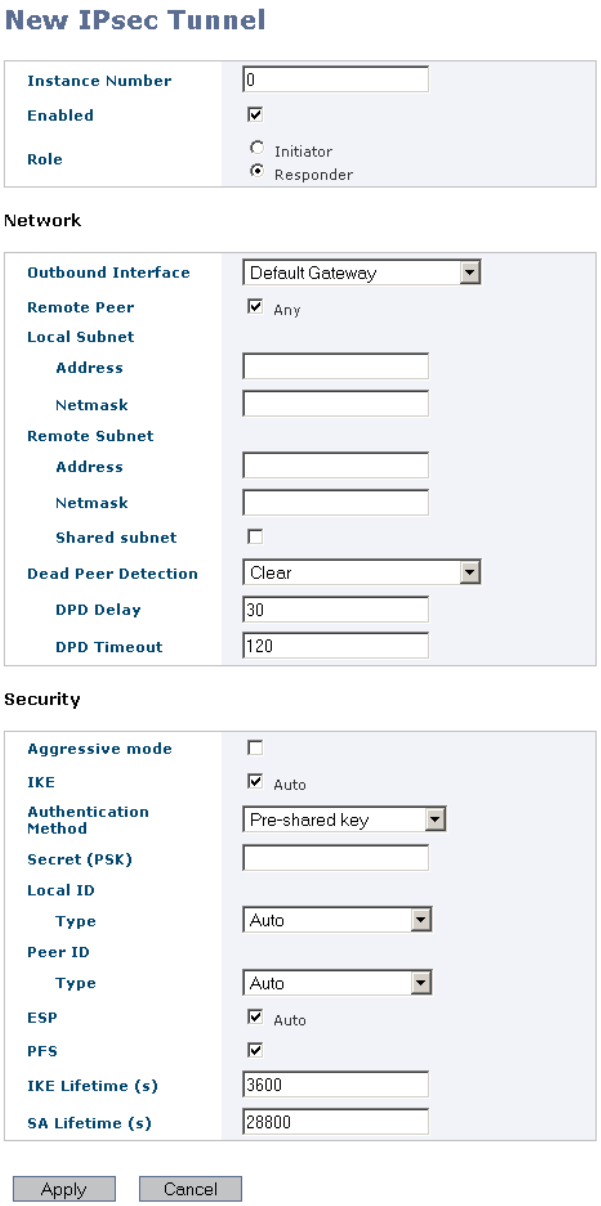
The list shows currently configured IPsec tunnels, and displays some of the tunnel settings.

ID	The IPsec tunnel index. Each configured IPsec tunnel is identified by a number for maintenance purposes. This ID is of local significance only.
Enabled	A green check-mark means enabled and a dash means disabled.
Remote Peer	The IP address or domain name of the remote peer. **Any** is shown if the remote peer is allowed to connect from any IP address.
Peer ID	The Name/E-mail/Key/IP used for matching the identify of the remote peer. **Auto** is shown if any peer ID is accepted.
Local ID	The Name/E-mail/Key/IP used to identify ourselves to the remote peer. **Auto** means that the IP of the outbound interface is used as ID.
MORE **More**	Show the details of this tunnel by hovering the pointer over this button. This is only available if you have JavaScript enabled in your browser.
Edit	Click this icon to edit the settings of a VPN tunnel.
Delete	Click this icon to remove a VPN tunnel. **Note:** Tunnels which are not intended to be used should either be *deleted* or *disabled* (section 35.2.2).

35.2.2 Configure new IPsec tunnel via the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ IPsec ⇒ **`New IPsec Tunnel`**

When clicking the **`New IPsec Tunnel`** button the window to configure a new IPsec tunnel appears.



General part:

Instance number	The IPsec tunnel index. Each configured IPsec tunnel is identified by a number for maintenance purposes. This ID is of local significance only.
Enabled	A tunnel can be configured as **Enabled** or **Disabled**. **Note:** Tunnels which are not intended to be used should either be *deleted* (section 35.2.1) or *disabled*.
Role	Configure the VPN gateway to act as *Initiator* or *Responder* of the VPN tunnel.

Network part:

Outbound Interface	The outbound interface for this tunnel. The interface can either be stated explicitly (e.g., **vlan3**) or implicitly as the interface leading to the **Default Gateway**.
Remote Peer Any (Checkbox)	Click the **Any** checkbox if the remote peer can connect from any IP address. This is typically the case if the remote peer is a *road warrior*, who may use different addresses every time he/she connects. A VPN gateway should only consider setting **Remote Peer** to **Any** if it is acting as **Responder** (i.e., when the remote peer is acting as **Initiator**). Un-check the **Any** checkbox to specify a specific IP address (or domain name) for the remote host, see the item below.
Remote Peer Address/Name	The IP address (e.g., **1.2.3.4**) or domain name (e.g., **foobar.example.com**) of the remote peer. This option is required if the node is acting as **Initiator** of the VPN tunnel. This option is only possible to set if the **Any** checkbox is *un-checked*.
Local Subnet Address & Netmask	The **Address** (e.g. **192.168.10.0**) and **Netmask** (e.g., **255.255.255.0**) define the local subnet. 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. If no local subnet is specified, only traffic to/from the IP address of the **Outbound Interface** will be allowed through the tunnel.

Continued on next page

Continued from previous page

Remote Subnet Address & Netmask, & Shared Subnet (Checkbox)	The **Address** (e.g. **192.168.11.0**) and **Netmask** (e.g., **255.255.255.0**) define the remote subnet. Only traffic to 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. In case the remote peer is a PC (see fig. 35.3), specify the PC's VPN client IP address (e.g., **192.168.12.49**) as **Address**, and **255.255.255.255** as **Netmask**. If no remote subnet is specified, only traffic to/from the IP address of the **Remote Peer** will be allowed through the tunnel. On a *responder*, you can specify that the remote subnet configured is *shared* by multiple initiators by setting the **Shared subnet** checkbox. The local subnet of each initiator must be within the range specified by the responder's remote subnet. By un-checking the **Shared subnet**, there can only be one initiator for this tunnel configuration, and its local subnet must match the responder's remote subnet.
Dead Peer Detection	The DPD Action. The DPD action defines how the VPN gateway should react when the peer is determined to be unreachable (i.e., "dead").
DPD Delay	The DPD delay is the interval between DPD probing messages sent by this VPN gateway. (The DPD delay setting on the two peers are independent, thus they may differ.)
DPD Timeout	If a period corresponding to the DPD timeout elapses without getting any response on the DPD probe messages, the VPN gateway considers the peer to be down.

Security part:

Aggressive Mode	Configure whether this VPN tunnel should use *aggressive* or *main* mode for the IKE handshake. Checking the **`Aggressive mode`** checkbox specifies use of *aggressive* mode; un-checking the checkbox means specifies use of *main* mode. For Certificate based authentication, only *main* mode can be used. For PSK either *main* or *aggressive* mode can be used.
IKE Auto (Checkbox)	The cipher suite to use for the IKE handshake can either be negotiated automatically between the peers, or a specific suite can be configured manually. Check the **`Auto`** checkbox to specify cipher auto-negotiation; un-check the checkbox to specify an IKE cipher suite manually (see below). Note: Cipher auto-negotiation is only valid with main mode IKE. In case of aggressive mode, a specific IKE cipher suite must be configured (see below).
IKE Encryption, Authentication & DH-Group	Configure the encryption algorithm, message authentication algorithm and Diffie-Hellman group to use for the IKE handshake. This option is only possible to set if the IKE **`Auto`** checkbox is *un-checked*.
Authentication Method	Select between PSK and Certificate based IKE authentication.
Secret	The pre-shared secret (PSK) password string used to protect the IKE handshake. The password string should consist of at least 8 characters and at most 63 characters. Valid characters are ASCII characters 33-126, except '#' (ASCII 35).
Local Certificate	Label of local certificate (and associated private key). Mandatory when IKE authentication is based on certificates.
Remote Certificate	Label of remote (peer) certificate. Only used for *trusted peer* scenarios, see section 35.1.7.3.

Continued on next page

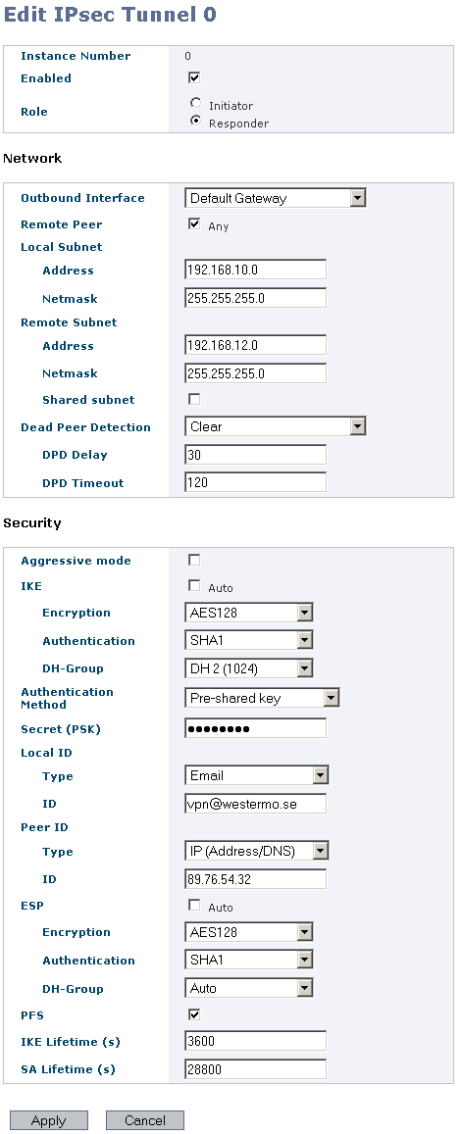
	Continued from previous page
Local ID Type & ID	The identity used by the VPN gateway during the IKE handshake. Typically the **Name(DNS/User)** type with a simple ID text string (e.g., **alice**) can be used to identify the VPN gateway. For more details on available identification types and ID values, see section 35.1.2. If **Auto** is selected, the local-id will be of type **IP Address** (for PSK authentication), using the IP address of the specified **Outbound interface** as identity. For certificate authentication, **Auto** implies a local-id of type **Distinguished Name**, using the subject string of the local certificate as identity.
Peer ID Type & ID	The identity used by the peer VPN gateway during the IKE handshake. Typically the **Name(DNS/User** type with a simple ID text string (e.g., **bob**) can be used to identify the peer VPN gateway. For more details on available identification types and ID values, see section 35.1.2. If **Auto** is selected, the **Peer ID** will be of type **IP Address** (for PSK authentication), using the IP address from the **Remote Peer Address/Name** field as identity (a domain name will be resolved to an IP address). For certificate authentication, **Auto** is discouraged for the **Peer ID**, see section 35.1.7 for details.
ESP Auto (Checkbox)	The cipher suite to use for the ESP handshake can either be negotiated automatically between the peers, or a specific suite can be configured manually. Check the **Auto** checkbox to specify cipher auto-negotiation; uncheck the checkbox to specify an ESP cipher suite and Diffie-Hellman group manually (see below). Note: ESP cipher auto-negotiation is only valid with main mode IKE. In case of aggressive mode, a specific ESP cipher suite must be configured (see below).
	Continued on next page

	Continued from previous page
ESP Encryption, Authentication & DH-Group	Configure the encryption algorithm, message authentication algorithm, and the Diffie-Hellman group to use for the ESP handshake and PFS. This option is only possible to set if the ESP **Auto** checkbox is *un-checked*.
PFS	Enable the Perfect Forward Secrecy (PFS) extension. PFS uses Diffie-Hellman for key exchange. The DH group is configured together with the ESP settings.
IKE Lifetime(s)	The maximum lifetime of the IKE (Phase 1) SA in seconds. Default is 3600 (1h).
SE Lifetime(s)	The maximum lifetime of the ESP (Phase 2) SA in seconds. Default is 28800 (8h).

35.2.3 Edit existing IPsec tunnel via the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ IPsec ⇒  (IPsec Tunnel)

By clicking the **Edit** button in the list of IPsec tunnels, you reach the **Edit IPsec Tunnel** page, as shown below.

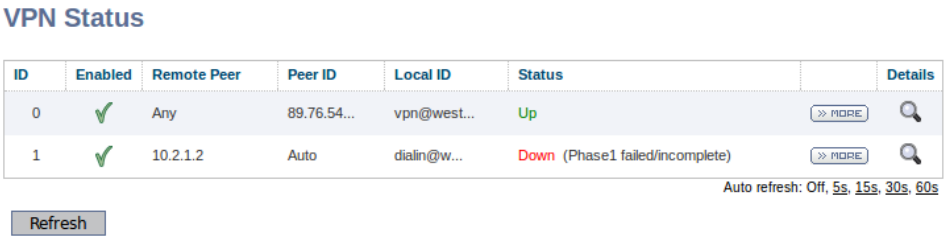


For information on the available configuration items, see section 35.2.2.

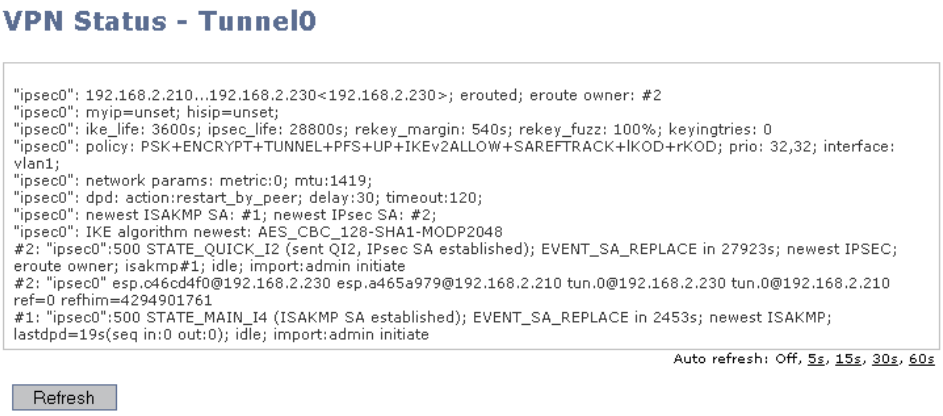
35.2.4 View IPsec Tunnel Status

Menu path: Status ⇒ VPN & Tunnel ⇒ IPsec

The **VPN Status** page lists the status of configured IPsec tunnels.



Click the **Details** symbol for a specific tunnel to see more verbose status information.



Configured settings can also be seen by hovering the pointer over the **More** button » MORE (you need JavaScript enabled it your browser to see this information).

35.3 Managing VPN settings via the CLI

The table below shows VPN management features available via the CLI.

Command	Default	Section
Configure VPN Settings		
tunnel		Section 35.3.1
[no] ipsec-nat-traversal	Disabled	Section 35.3.2
[no] ipsec-mtu-override <BYTES>	1419	Section 35.3.3
[no] ipsec <INDEX>		Section 35.3.4
[no] enable	Enabled	Section 35.3.5
[no] aggressive	Main mode	Section 35.3.6
[no] pfs	Enabled	Section 35.3.7
[no] ike crypto <3des|aes128|. . . > auth <md5|sha1> dh <1024|. . . >	Auto	Section 35.3.8
[no] esp crypto <3des|aes128|. . . > auth <md5|sha1> dh <auto|. . . >	Auto	Section 35.3.9
[no] method <psk|cert>	PSK	Section 35.3.10
[no] secret <PASSWORD>	Empty	Section 35.3.11
[no] local-cert <LABEL>	Disabled	Section 35.3.12
[no] remote-cert <LABEL>	Disabled	Section 35.3.13
[no] remote-ca <same|any| dn <DNSTRING>>	Same	Section 35.3.14
[no] peer <IPADDR|FQDN>	Any	Section 35.3.15
[no] outbound <IFACE>	Auto	Section 35.3.16
[no] local-id <inet <IPADDR|DOMAIN>| name <DOMAIN|USER> | email <USER@DOMAIN> | key <ID> | dn <DNSTRING>>	Auto	Section 35.3.17
[no] remote-id <inet <IPADDR|DOMAIN>| name <DOMAIN|USER> | email <USER@DOMAIN> | key <ID> | dn <DNSTRING>>	Auto	Section 35.3.18
[no] local-subnet <SUBNET/LEN | SUBNET NETMASK>	Auto	Section 35.3.19

Continued on next page

Continued from previous page

Command	Default	Section
[no] remote-subnet <SUBNET/LEN | SUBNET NETMASK> [shared]	Auto	Section 35.3.20
[no] local-protocol <PROTO> [port <PORT>]	Disabled	Section 35.3.21
[no] remote-protocol <PROTO> [port <PORT>]	Disabled	Section 35.3.22
[no] initiator	Responder	Section 35.3.23
[no] dpd-action <clear|hold|restart>	Clear/Restart	Section 35.3.24
[no] dpd-delay <SECONDS>	30	Section 35.3.25
[no] dpd-timeout <SECONDS>	120	Section 35.3.26
[no] ike-lifetime <SECONDS[s]|...>	1h	Section 35.3.27
[no] sa-lifetime <SECONDS[s]|...>	8h	Section 35.3.28
Show VPN Status		
show tunnel ipsec [ID]		Section 35.3.29

35.3.1 Managing Tunnels

Syntax `tunnel`

Context Global Configuration context

Usage Use the **"tunnel"** command to enter the Tunnel Configuration context.

Use **"show tunnel"** to list configured VPN tunnels (also available as **"show"** command within the Tunnel Configuration context.

Default values Not applicable.

35.3.2 Enable/disable IPsec NAT Traversal

Syntax `[no] ipsec-nat-traversal`

Context Tunnel Configuration context

Usage Enable or disable NAT-T for *all* IPsec tunnels. NAT Traversal can cause inter-operability problems with some IPsec clients, so the default setting is disabled.

However, when NAT-T is enabled it only kicks in when the server and client detects they are being NAT'ed. So in most cases it is a safe option to set.

Use **"ipsec-nat-traversal"** to enable and **"no ipsec-nat-traversal"** to disable NAT traversal.

Use **"show ipsec-nat-traversal"** to show whether IPsec NAT traversal is enabled or disabled.

Default values Disabled (**"no ipsec-nat-traversal"**)

35.3.3 Configure IP tunnel MTU

Syntax `[no] ipsec-mtu-override <BYTES>`

Context Tunnel Configuration context

Usage Override default MTU for *all* IPsec tunnels.

Use **"ipsec-mtu-override <BYTES>"** to specify a specific MTU value to use for all IPsec tunnels. Use **"no ipsec-mtu-override"** to return to the default setting.

Use **"show ipsec-mtu-override"** to show the configured IPsec MTU value.

Default values 1419 (bytes)

35.3.4 Managing IPsec VPN Tunnels

Syntax `[no] ipsec <INDEX>` where INDEX is a number greater or equal to 0.

Context Tunnel Configuration context

Usage Create, delete, or modify an IPsec VPN tunnel. Use **"ipsec <INDEX>"** to create a new IPsec tunnel, or to enter the configuration context of an existing IPsec tunnel. (To find the index of configured tunnels, use **"show tunnel"** as described in section 35.3.1,)

Use **"no ipsec <INDEX>"** to remove a specific IPsec VPN tunnel, or **"no ipsec"** to remove all configured IPsec VPN tunnels.

Use **"show ipsec <INDEX>"** to show all settings of a specific IPsec tunnel (also available as **"show"** command within the IPsec Configuration context).

> **Note**
> Tunnels which are not intended to be used should either be *deleted* or *disabled* (section 35.3.5).

Default values Not applicable.

35.3.5 Enable/disable an IPsec VPN tunnel

Syntax `[no] enable`

Context IPsec Configuration context

Usage Enable or disable an IPsec VPN tunnel. A disabled tunnel will be deactivated, but keeps its configuration settings.

Use **"enable"** to enable and **"no enable"** to disable an IPsec VPN tunnel.

Use **"show enable"** to show whether this IPsec tunnel is enabled or disabled.

> **Note**
> Tunnels which are not intended to be used should either be *deleted* (section 35.3.4) or *disabled*.

Default values Enabled

35.3.6 IKE phase-1 aggressive or main mode

Syntax `[no] aggressive`

Context IPsec Configuration context

Usage Select aggressive or main mode for the IKE phase-1 handshake.

Use **"aggressive"** to select aggressive mode, and **"no aggressive"** to select main mode.

Use **"show aggressive"** to show whether this IPsec tunnel is configured to use IKE *aggressive* or *main* mode. **"Enabled"** means *aggressive* mode, while **"Disabled"** means *main* mode.

Default values Disabled (**"no aggressive"**, i.e., *main* mode is use by default.)

35.3.7 Enable/disable Perfect Forward Secrecy

Syntax `[no] pfs`

Context IPsec Configuration context

Usage Enable or disable Perfect Forward Secrecy for this IPsec tunnel. Protects previous key exchanges even if the current one is compromised.

> **Note**
> This setting is not supported by all IPsec implementations. It is however recommended to have it enabled, on both sides of the connection.

If you are unsure what do to, you can safely disable PFS. If the IPsec daemon receives a request with PFS, it will allow it despite how your having disabled it here, because there is absolutely no reason not to use PFS if it is available.

Use **"pfs"** to enable and **"no pfs"** to disable perfect forward secrecy.

Use **"show pfs"** to show whether *perfect forward secrecy* is enabled or disabled for this tunnel.

Default values Enabled (**"pfs"**)

35.3.8 Configure allowed crypto algorithms for IKE phase-1

Syntax `[no] ike crypto <3des|aes128|...> auth <md5|sha1> dh <1024|...>`

Context IPsec Configuration context

Usage Set IKE phase-1 handshake. Configure what security suite to use to protect the IKE authentication handshake. Here the security suite consists of three parameters:

- *Encryption algorithm:* Supported encryption algorithms are *3des*, *aes128*, *aes192* and *aes256*.
- *Message authentication/integrity:* Supported hash algorithms for message authentication are *md5*, and *sha1*.

- *Diffie-Hellman groups:* Supported Diffie-Hellman groups are 1024 (DH group 2), 1536 (DH group 5), 2048 (DH group 14), 3072 (DH group 15), 4096 (DH group 16), 6144 (DH group 17) and 8192 (DH group 18).

By specifying an IKE suite, e.g., **”ike crypto aes256 auth sha1 dh 2048”** you will ensure that this suite is used to secure the IKE handshake - if the remote side does not support this suite, the handshake will fail.

Use **”no ike”** to specify the *automatic* security suite negotiation. When configured as an *initiator*, this means that all combinations will be tried (starting by offering a set of suites with either AES-128 or 3DES for encryption, SHA1 or MD5 for authentication, and DH groups 1024, 1536 and 2048). When configured as a *responder* any combination of the listed algorithms will be accepted.

Use **”show ike”** to show the configured IKE Cipher suite for this tunnel, i.e., encryption algorithm, message authentication algorithm, and Diffie-Hellman group. **”Auto”** is shown if the VPN gateway is configured to auto-negotiate what IKE cipher suite to use.

Default values Auto (**”no ike”**)

> **Note**
> If *aggressive* mode is selected for the IKE phase-1 handshake, the default security suite for IKE phase-1 negotiation is set to ”AES128-SHA1-DH1024” (**”esp crypto aes128 auth sha1 dh 1024”**).

Examples The following example show the output when AES-128 is used for encryption, SHA-1 for message authentication, and Diffie-Hellman group 1024.

> **Example**
> ```
> example:/config/tunnel/ipsec-0/#> show ike
> AES128-SHA1-1024
> example:/config/tunnel/ipsec-0/#>
> ```

35.3.9 Configure allowed crypto algorithms for ESP

Syntax `[no] esp crypto <3des|aes128|...> auth <md5|sha1> dh <auto|...>`

Context IPsec Configuration context

Usage Set IKE Phase-2 hand shake negotiation. Configure what security suite ESP should use to protect the *data traffic* in the established VPN tunnel. Here the security suite consists of two parameters:

- *Encryption algorithm:* Supported encryption algorithms are *3des*, *aes128*, *aes192* and *aes256*.
- *Message authentication/integrity:* Supported hash algorithms for message authentication are *md5*, and *sha1*.
- *Diffie-Hellman group for PFS:* The Diffie-Hellman group can be negotiated automatically, or a preferred group can be selected by hand. Supported Diffie-Hellman groups are 1024 (DH group 2), 1536 (DH group 5), 2048 (DH group 14), 3072 (DH group 15), 4096 (DH group 16), 6144 (DH group 17) and 8192 (DH group 18).

By specifying an ESP suite, e.g., **”esp crypto aes256 auth sha1 dh 1024”** you will ensure that this suite is used to secure the data traffic in the established IPsec ESP tunnel. IKE phase-1 handshake - if the remote side does not support this suite, the handshake will fail.

Use **”no esp”** to specify the *automatic* security suite negotiation. When configured as an *initiator*, this means that all combinations will be tried. When configured as a *responder* any combination of the listed algorithms will be accepted.

Use **”show esp”** to show the configured ESP Cipher suite for this tunnel. **”Auto”** is shown if the VPN gateway is configured to auto-negotiate what ESP cipher suite to use.

Default values Auto (**”no esp”**)

> **Note**
> If *aggressive* mode is selected for the IKE phase-1 handshake, the default security suite for IKE phase-2 negotiation is set to ”AES128-SHA1-AUTO” (**”esp crypto aes128 auth sha1 dh auto”**).

35.3.10 Select Pre-shared Secret or Certificate based authentication

Syntax `[no] method <psk|cert>`

Context IPsec Configuration context

Usage Select Pre-shared secret or Certificate based IKE authentication. Use **”method psk”** to use pre-shared secret authentication (default), or **”method cert”** to use certificates for IKE authentication.

”no method” will return to default setting **”method psk”**.

Use **”show method”** to show whether IKE authentication is configured to use PSK or certificate.

Default values Pre-shared Secret (`method psk`)

35.3.11 Configure IPsec Pre-shared Secret

Syntax `[no] secret <PASSWORD>`

Context IPsec Configuration context (Only valid when **”method psk”** is set.)

Usage Set pre-shared key (shared secret). The password string should consist of at least 8 characters and at most 63 characters.

Valid characters are ASCII characters 33-126, except ’#’ (ASCII 35).

Use **”no secret”** to remove a configured pre-shared secret.

Use **”show secret”** to show the configured pre-shared secret (PSK) for this tunnel.

Default values Empty

35.3.12 Select Local Certificate

Syntax `[no] local-cert <LABEL>`

Context IPsec Configuration context (Only valid when **”method cert”** is set.)

Usage Select local certificate (and associated private key), i.e., the certificate by which this unit will authenticate itself. The **”LABEL”** is the reference of the certificate when imported to the WeOS unit.

This setting is required when **”method cert”** is set.

Use **”no local-cert”** to remove the selection of local certificate.

Use **”show local-cert”** to show the local certificate setting.

Default values Disabled

35.3.13 Select Remote Certificate

Syntax `[no] remote-cert <LABEL>`

Context IPsec Configuration context (Only valid when **"method cert"** is set.)

Usage Select remote certificate, if the certificate of the trusted peer has been imported to this WeOS unit.

The **"LABEL"** is the reference of the certificate when imported to the WeOS unit.

Use **"no remote-cert"** to remove the selection of remote certificate.

Use **"show remote-cert"** to show the remote certificate setting.

Default values Disabled

35.3.14 Manage Remote CA restrictions

Syntax `[no] remote-ca <same|any|dn <DNSTRING>>`

Context IPsec Configuration context (Only valid when **"method cert"** and **"no remote-cert"** are set.)

Usage Define restrictions of the peer's CA. By default, the peer is required use a certificate issued by the same CA as this unit (**"same"**).

Use **"remote-ca any"** to allow peers with a certificate issued by any of the CAs trusted by this unit. It is also possible to only accept peers with certificates issued by a specific CA (among the ones trusted by this unit) by the **"remote-ca dn <DNSTRING>"**setting.

"no remote-ca" will return to the default setting (**"remote-ca same"**).

Use **"show remote-ca"** to show the remote CA setting.

Default values Same (**"remote-ca same"**)

35.3.15 Specify IP Address/domain name of remote unit

Syntax `[no] peer <IPADDR|FQDN>`

Context IPsec Configuration context

Usage Set peer IP address, or DNS domain name. When acting as initiator, the peer setting defines the remote server to connect to. As responder it can be used to allow a single client or not.

Use **”no peer”** to allow connections from any client.

Use **”show peer”** to show the configured *peer IP address* or *peer domain name*. **”Any”** is shown if the peer can connect from any IP address.

Default values Any

35.3.16 Configure Outbound Interface

Syntax `[no] outbound <IFACE>`

Context IPsec Configuration context

Usage Set the outbound interface of this tunnel.

Use **”no outbound”** to automatically select the interface leading to the *default gateway* as outbound interface.

Use **”show outbound”** to show the configured *outbound interface* for this tunnel. **”Default Gateway”** is shown if the interface leading to the default gateway should be used as outbound interface.

See section 35.1.1 for more information on the outbound interface.

Default values Auto (**”no outbound”**)

35.3.17 Configure Local Identifier

Syntax `[no] local-id <inet <IPADDR|DOMAIN> | name <DOMAIN|USER> | email <USER@DOMAIN> | key <ID> | dn <DNSTRING>>`

Context IPsec Configuration context

Usage 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.

For more details on available identification types and ID values, see section 35.1.2.

If **"no local-id"** is selected for PSK authentication, the local-id will be of type **"inet"** (IPv4 address), using the IP address of the *Outbound interface* (see section 35.3.16) as identity. For certificate authentication, **"no local-id"** implies a local-id of type *Distinguished Name*, using the subject string of the local certificate as identity.

Use **"show local-id"** to show the configured *local identifier* for this tunnel, i.e., both the local-id *type* and the local-id *value*. **"Auto"** is shown if the local identifier is assigned as type **"inet"** with the IP address of the *outbound interface* as value.

Default values Auto (**"no local-id"**)

35.3.18 Configure Remote Identifier

Syntax `[no] local-id <inet <IPADDR|DOMAIN> | name <DOMAIN|USER> | email <USER@DOMAIN> | key <ID> | dn <DNSTRING>>`

Context IPsec Configuration context

Usage Set the identifier (type and value) for the peer VPN gateway. The remote-id is used by the peer VPN gateway during the IKE handshake. Typically the **"name"** type with a simple ID text string (e.g., **"bob"**) can be used to identify the peer VPN gateway.

For more details on available identification types and ID values, see section 35.1.2.

If **"no remote-id"** is selected for PSK authentication, the **"remote-id"** will be of type **"inet"** (IPv4 address), using the IP address from the configured *Peer* (see section 35.3.15) as identity. A peer domain name will be resolved to an IP address.

For certificate authentication, **Auto** is discouraged for the **Peer ID**, see section 35.1.7 for details.

Use **"show remote-id"** to show the configured *remote identifier* for this tunnel, i.e., both the remote-id *type* and the remote-id *value*. **"Auto"** is shown if the local identifier is assigned as type **"inet"** with the IP address of the *peer* as value.

Default values Auto (**"no remote-id"**)

35.3.19 Configure Local Subnet

Syntax `[no] local-subnet <SUBNET/LEN | SUBNET NETMASK>`

Context IPsec Configuration context

Usage Set the local subnet of this 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.

If **"no local-subnet"** is specified, only traffic to/from the IP address of the *outbound interface* will be allowed through the tunnel.

Use **"show local-subnet"** to show the configured *local subnet* for this tunnel. **"None"** is shown if no local subnet has been configured.

Default values None (**"no local-subnet"**)

35.3.20 Configure Remote Subnet

Syntax `[no] remote-subnet <SUBNET/LEN | SUBNET NETMASK> [shared]`

Context IPsec Configuration context

Usage Set the remote subnet of this 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.

In case the remote peer is a PC (see fig. 35.3), specify the PC's VPN client IP address with a "/32" prefix length, e.g., **"192.168.12.49/32"**.

If **"no remote-subnet"** is specified, only traffic to/from the IP address of the *Peer* will be allowed through the tunnel.

On a *responder*, you can specify that the remote subnet configured is *shared* by multiple initiators by setting the **"shared"** keyword (default disabled). The local subnet of each initiator must be within the range specified by the responder's remote subnet. Without the **"shared"** keyword, there can only be one initiator for this tunnel configuration, and its local subnet must match the responder's remote subnet.

Use **”show remote-subnet”** to show the configured *remote subnet* for this tunnel. **”None”** is shown if no remote subnet has been configured.

Default values None (**”no remote-subnet”**)

35.3.21 Configure Local IP Protocol and UDP/TCP port

Syntax `[no] local-protocol <PROTOCOL> [port <PORT>]`

Context IPsec Configuration context

Usage Allowed transmitted IP protocol, and (TCP/UDP) port over this connection. This setting must match in both ends of the tunnel for the tunnel to start. **”PROTOCOL”** is IP protocol specified as a number (0-255), or by name. If protocol is TCP(6) or UDP(17), the traffic can further match specific (TCP/UDP) port number for transmitted packets (**”port <PORT>”**).

If **”no local-protocol”** is specified, all IP protocols are allow.

Use **”show local-protocol”** to show the local IP protocol and UDP/TCP port settings for this tunnel.

Default values Disabled (**”no local-protocol”**), i.e., all local IP protocols allowed.

35.3.22 Configure Remote IP Protocol and UDP/TCP port

Syntax `[no] remote-protocol <PROTOCOL> [port <PORT>]`

Context IPsec Configuration context

Usage Allowed received IP protocol, and (TCP/UDP) port over this connection. This setting must match in both ends of the tunnel for the tunnel to start. **”PROTOCOL”** is IP protocol specified as a number (0-255) or by name. If protocol is TCP(6) or UDP(17), the traffic can further match specific (TCP/UDP) port number for received packets (**”port <PORT>”**).

If **”no remote-protocol”** is specified, all IP protocols are allow.

Use **”show remote-protocol”** to show the remote IP protocol and UDP/TCP port settings for this tunnel.

Default values Disabled (**”no remote-protocol”**), i.e., all local IP protocols allowed.

35.3.23 Configure Initiator/Responder Setting

Syntax `[no] initiator`

Context IPsec Configuration context

Usage Select whether the VPN gateway should act as initiator or responder of this IPsec tunnel.

Use **”initiator”** to make the VPN gateway act as *initiator*, and **”no initiator”** to make it act as responder.

Use **”show initiator”** to show whether the VPN gateway acts as *Initiator* or *Responder* for this tunnel.

Default values Responder (**”no initiator”**)

35.3.24 Configure Dead Peer Detection Action

Syntax `[no] dpd-action <clear|hold|restart>`

Context IPsec Configuration context

Usage Set the DPD action for this VPN gateway. The DPD action defines how the VPN gateway should react when the peer is determined to be unreachable (i.e., ”dead”).

Use **”no dpd-action”** to disable the DPD mechanism on this VPN gateway. When disabled, this VPN gateway will not probe the peer to check if it is down, however, this VPN gateway will still respond to DPD probing messages from the peer. That is, it is possible for the peer to the DPD mechanism successfully even though DPD is disabled on this side.

Use **”show dpd-action”** to show the configured DPD action setting. **”off”** is shown if DPD has been disabled on this VPN gateway.

For more information on DPD action settings, see section 35.1.5.

Default values This depends on the role of this VPN gateway.

- *Initiator:* If this VPN gateway is the initiator of the tunnel, the DPD action is by default set to *restart* (**”dpd-action restart”**)
- *Responder:* If this VPN gateway is the initiator of the tunnel, the DPD action is by default set to *clear* (**”dpd-action clear”**)

35.3.25 Configure Dead Peer Detection Delay

Syntax `[no] dpd-delay <SECONDS>`

Context IPsec Configuration context

Usage Set the DPD probing interval. The DPD delay is the interval between DPD probing messages sent by this VPN gateway. (The DPD delay setting on the two peers are independent, thus they may differ.)

Use **"no dpd-delay"** to return to the default setting.

Use **"show dpd-delay"** to show the configured DPD delay setting (in seconds).

Default values 30 (seconds)

35.3.26 Configure Dead Peer Detection Timeout

Syntax `[no] dpd-timeout <SECONDS>`

Context IPsec Configuration context

Usage Set the DPD timeout. If a period corresponding to the DPD timeout elapses without getting any response on the DPD probe messages, the VPN gateway considers the peer to be down.

Use **"no dpd-timeout"** to return to the default setting.

Use **"show dpd-timeout"** to show the configured DPD timeout setting (in seconds).

Default values 120 (seconds)

35.3.27 Configure IKE Lifetime

Syntax `[no] ike-lifetime <SECONDS[s] | MINUTESm | HOURSh | DAYSd>`

Context IPsec Configuration context

Usage Set the IKE (phase 1) security association lifetime. When this time has passed, a new phase 1 negotiation will be initiated. The remote peer may use a different value. In that case, the peer with the lowest timeout will initiate the renegotiation first.

Use **”no ike-lifetime”** to return to the default setting.

Use **”show ike-lifetime”** to show the configured IKE (phase 1) security association lifetime setting (in seconds).

Default values 3600 seconds (1h)

35.3.28 Configure SA (ESP) Lifetime

Syntax `[no] sa-lifetime <SECONDS[s] | MINUTESm | HOURSh | DAYSd>`

Context IPsec Configuration context

Usage Set the ESP (phase 2) security association lifetime. When this time has passed, a new phase 2 negotiation will be initiated. The remote peer may use a different value. In that case, the peer with the lowest timeout will initiate the renegotiation first.

Use **”no sa-lifetime”** to return to the default setting.

Use **”show sa-lifetime”** to show the configured ESP (phase 2) security association lifetime setting (in seconds).

Default values 28800 seconds (8h)

35.3.29 Show IPsec Tunnel Status

Syntax `show tunnel ipsec [ID]`

Context Admin Execcontext.

Usage Show the status for all or for a specific IPsec tunnel.

Default values If no tunnel ID is specified, the status of all tunnels is shown.

Chapter 36

SSL VPN

This chapter describes the WeOS SSL VPN support. The WeOS SSL VPN is based on OpenVPN[1], and WeOS units can act both as SSL VPN server and client. With the WeOS unit configured an SSL VPN server gateway with layer-3 VPN interface it is typically used in HOST-NET VPN scenarios serving where various SSL VPN (OpenVPN) clients can connect to the WeOS unit. The unit can also be used both as VPN server and client gateways in a NET-NET VPN scenario, using layer-2 VPN interfaces.

As of WeOS v4.17.1, bridged LAN VPNs (i.e., bridging a layer-2 SSL VPN interface with regular Ethernet ports on a VLAN) is not supported. Such support is planned, but not yet implemented.

36.1 Overview of SSL VPN Management Features

Table 36.1 summarises the SSL VPN features available in WeOS. These features are further explored in the following sections.

36.1.1 Introduction to SSL VPN

In an SSL VPN we have a VPN Server Gateway (Alice) providing secure access to a protected network (e.g., a central office network) to one or more VPN Clients (Bob) connecting over an unsecure network such as the Internet. Bob could be

[1] OpenVPN home page, http://openvpn.net (March 2014).

Feature	Web	CLI	General Description
SSL VPN Configuration			
Role (Server/Client)	X	X	Section 36.1.1
Transport settings (UDP/TCP)	X	X	Section 36.1.2
SSL Tunnel Network Settings	X	X	Section 36.1.3
Type (Layer2/Layer3)	X	X	-”-
IP address settings	X	X	-”-
Address pool	X	X	-”-
Pushed networks	X	X	-”-
Traffic between clients	X	X	-”-
SSL Tunnel Security	X	X	Section 36.1.4
Authentication	X	X	-”-
Certificates	X	X	-”-
Username/password	X	X	-”-
Local-DB (Server)	X	X	-”-
RADIUS (Server)	X	X	-”-
AAA identity (Client)	X	X	-”-
Cipher settings	X	X	-”-
TLS authentication settings	X	X	-”-
Other SSL tunnel setting	X	X	Section 36.1.5
Keepalive	X	X	-”-
Compression	X	X	-”-
SSL VPN Status			
Show SSL VPN Status	X	X	
Related Settings			Section 36.1.6
Routing (static/dynamic)	X	X	See also chapters 26-28
Firewall and NAT	X	X	See also chapter 31

Table 36.1: Summary of SSL VPN features

a single host (a Host-NET SSL VPN) as shown in fig. 36.1, or Bob could itself be a VPN gateway with a local network behind (a NET-NET SSL VPN) as shown in fig. 36.2.

We refer to Alice as a VPN Server, as she waits for VPN Clients to establish VPN connections. Bob is the VPN client initiating the connection establishment.

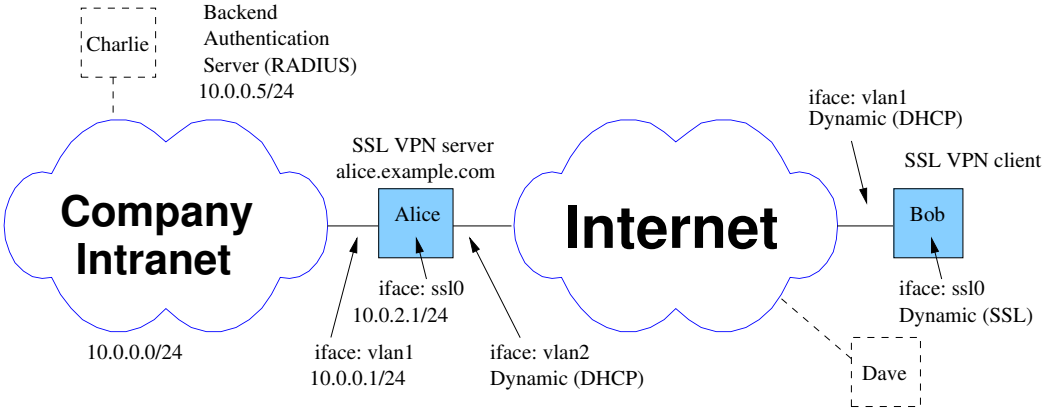


Figure 36.1: SSL Host-NET setup. One or more SSL Clients (”roadwarriors” Bob and Dave) can access the company private network via the SSL Server Gateway (Alice).

The VPN server (Alice) may be reachable via a fixed IP address on her upstream interface. But if Alice acquires her IP address dynamically from her ISP, it is recommended that Alice use Dynamic DNS (DDNS) to bind her IP address to a domain name, see section 19.3.3. The VPN client (Bob) would then use Alice domain name when initiating the SSL tunnel (*alice.example.com* in figs. 36.1 and 36.2).

Example

```
bob:/config/#> tunnel
bob:/config/tunnel/#> ssl 0
bob:/config/tunnel/ssl-0/#> no server
bob:/config/tunnel/ssl-0/#> peer alice.example.com
bob:/config/tunnel/ssl-0/#> end
bob:/config/tunnel/#>
```

36.1.2 Tunnel Transport Settings

The WeOS SSL support assumes that there is an SSL Server unit and an SSL Client unit, where the client (Bob) initiates the VPN connection to the server (Alice). The SSL tunnel can be carried over UDP or TCP. By default UDP transport is used, with UDP port number *1194*.

In case the Bob is located behind a firewall, which outgoing traffic for UDP port 1194, an alternative can be to configure Alice and Bob to use TCP transport with

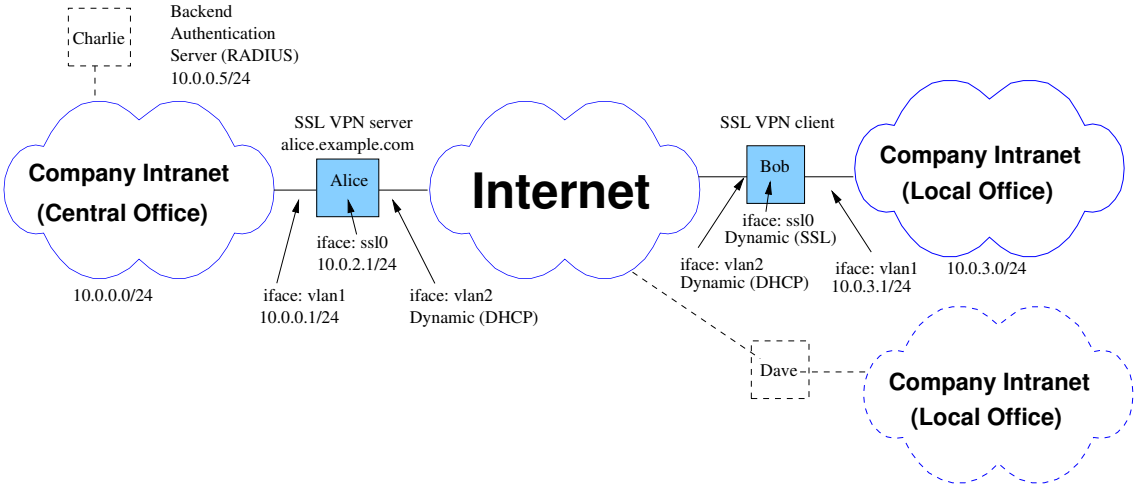


Figure 36.2: SSL NET-NET setup. One or more SSL Client Gateway(s) (Bob) can establish an SSL tunnel to the SSL Server Gateway, and provide a NET-NET VPN solution between the central office and branch office networks.

TCP port 443. This port is typically used for HTTPS traffic, and most firewalls will therefore allow such traffic to pass.

Note

As of WeOS v4.17.1, if you configure the your SSL server (Alice) to listen to TCP port 443, you should either disable Alice' web server or configure her web server to listen for HTTPS at another port.

An example where Alice listens for SSL connections on TCP port 443 is given below.

Example

```
alice:/config/#> web
alice:/config/web/#> ssl-port 8443
alice:/config/web/#> end
alice:/config/#> tunnel
alice:/config/tunnel/#> ssl 0
alice:/config/tunnel/ssl-0/#> protocol tcp
alice:/config/tunnel/ssl-0/#> port 443
alice:/config/tunnel/ssl-0/#> leave
alice:/#>
```

36.1.3 SSL Network Settings

For the SSL tunnel, Alice and Bob will have an SSL network interface (with names such as *ssl0*), which can be assigned an IP address, and be used as other network interfaces when it comes to routing and firewall settings, etc. The SSL interface can either be a layer-2 or layer-3 interface, see section 36.1.3.1 for more information.

Multiple clients (Bob and Dave) can connect to the same server. The clients and the server forms a virtual *subnet topology*[2].

- IP assignment: Alice, Bob, and Dave will each have an IP address within this virtual subnet. See section 36.1.3.2 for information on how to assign IP addresses at the server and client side. That section also touches upon related settings, such as *domain name server* and IP routes.
- Client to client communication: It is possible for two SSL clients to communicate with each other. This is enabled by default, see section 36.1.3.4 for more information.

36.1.3.1 Selecting layer-2 or layer-3 VPN interfaces

The SSL network interface can either be a *layer-3* interface or a *layer-2* interface.

- *Layer-3 interface (IP):* By default, WeOS SSL tunnels have layer-3 interfaces. This simplifies setting up a *HOST-NET* solution (see fig. 36.1) with the WeOS unit as SSL VPN Gateway, since many SSL VPN clients use layer-3 interfaces by default.
- *Layer-2 interface (LAN):* Layer-2 SSL interfaces have MAC addresses, just like other LAN interfaces in WeOS. As of WeOS v4.17.1 layer-2 is the recommended interface type when using SSL in *NET-NET* setups (see fig. 36.1). Dynamic routing protocols such as OSPF (chapter 27 and RIP chapter 28) can be used on layer-2 SSL interfaces.

[2] Although other topologies are possible for layer-3 SSL interfaces, current WeOS support is limited to the *subnet* topology. For more information on other possible SSL topologies not yet supported by WeOS (*p2p* and *net30*), see http://openvpn.net.

Note
As of WeOS v4.17.1, the layer-2 SSL interfaces can **not** be added to VLANs, i.e., it is not yet possible to bridge traffic between the SSL tunnel and the Ethernet or DSL ports on your WeOS unit. Such support is planned, but not yet implemented.

Below is an example of configuring the SSL interface type to layer-2 at Alice in fig. 36.2.

Example

```
alice:/config/#> tunnel
alice:/config/tunnel/#> ssl 0
alice:/config/tunnel/ssl-0/#> type layer2
alice:/config/tunnel/ssl-0/#> leave
alice:/#>
```

36.1.3.2 IP address and other SSL interface settings

In WeOS, the SSL VPN server (Alice) will always have a statically assigned address, while the SSL client (Bob) can either be assigned his SSL address statically or acquire it dynamically as part of the SSL tunnel establishment. Similar to other network interfaces, it is also possible to assign secondary IP addresses (section 19.2.5) to SSL interfaces.

- *Static IP addresses:* By default SSL interfaces are configured for static IP address assignment, but without any address defined. An example for Alice in fig. 36.1 is shown below.

  Example

  ```
  alice:/config/#> iface ssl0
  alice:/config/iface-ssl0/#> inet static
  alice:/config/iface-ssl0/#> address 10.0.2.1/24
  alice:/config/iface-ssl0/#> leave
  alice:/#>
  ```

- *Dynamic IP addresses:* Alice could hand out addresses dynamically to Bob and other SSL clients. To do this she should define the address pool to assign addresses from, see below

Example

```
alice:/config/#> tunnel
alice:/config/tunnel/#> ssl 0
alice:/config/tunnel/ssl-0/#> pool start 10.0.2.100 end 10.0.2.110
alice:/config/tunnel/ssl-0/#> leave
alice:/#>
```

An optional ”netmask” parameter can be added to the **”pool”** command, if the netmask for the clients should be smaller than the netmask of Alice SSL interface (set to **”/24”** in the example above).

Bob configures his SSL interface for dynamic address assignment:

Example

```
bob:/config/#> iface ssl0
bob:/config/iface-ssl0/#> inet dynamic
bob:/config/iface-ssl0/#> leave
bob:/#>
```

36.1.3.3 Other settings assigned by SSL server

The SSL server (Alice) can push the following settings to the client (Bob):

- Network route: In the HOST-NET setup (fig. 36.1), Alice would typically push a route to the central office subnet using the **”push-network 10.0.0.0/24”** setting. Up to 10 subnets can be pushed.
- Compression setting: The data compression setting (see section 36.1.5) at the server and client must match. Therefore the compression setting at the Alice is implicitly pushed to Bob. See also section 36.1.5.

Bob can decline using these settings offered by Alice, by using the **”no pull”** command. This does not affect Bob’s IP address assignment, which is instead controlled via interface settings as described in section 36.1.3.2.

Note

It is *not* possible to push routes from client to server. In the NET-NET setup (fig. 36.2) Alice would either configure a static route to Bob’s local network, or RIP or OSPF to exchange routes dynamically.

36.1.3.4 Managing traffic between VPN clients (hosts or gateways)

Traffic between VPN clients (Bob and Dave in figs. 36.1 and 36.2) will go via the VPN Server (Alice), and will by default be handled by the WeOS firewall at Alice. To allow client-client communication, there are two alternatives:

- *Add "allow" rule in firewall:* (for layer-3 tunnels) The VPN Server Gateway can add a appropriate *filter allow* rule for the given SSL interface. An example is given below. Note that *ssl0* is used both as *incoming* and *outgoing* interface.

  Example

  ```
  alice:/config/#> ip
  alice:/config/ip/#> firewall
  alice:/config/ip/firewall/#> filter allow in ssl0 out ssl0
  alice:/config/ip/firewall/#> leave
  alice:/#>
  ```

- *Enable client-to-client communication without involving the firewall:* (for layer-2 or layer-3 tunnels) With this setting, the VPN gateway (Alice) will forward packets between clients without involving her firewall.

  Example

  ```
  alice:/config/#> tunnel
  alice:/config/tunnel/#> ssl 0
  alice:/config/tunnel/ssl-0/#> client-to-client
  alice:/config/tunnel/ssl-0/#> leave
  alice:/#>
  ```

Note

When using a NET-NET setup (layer-2 VPN) with multiple VPN client gateways (Bob and Dave in fig. 36.2), then the **"client-to-client"** setting must be enabled at the VPN server (Alice) to enable traffic **between** the local office networks (networks behind Bob and Dave). As of WeOS v4.17.1 the alternative to enable the traffic via the firewall at Alice does not work for layer-2 VPNs.

36.1.4 SSL Security Settings

SSL security settings include *authentication* settings for tunnel establishment, and *cipher suite* settings (encryption and per packet authentication algorithms)

for the SSL tunnel.

36.1.4.1 Authentication of SSL users

WeOS units primarily relies on certificates for authentication of Alice and Bob. In addition, the server (Alice) can require Bob to provide username and password, which she can match in a local database, or towards a backend authentication (RADIUS) server (see *Charlie* in figs. 36.1 and 36.2).

Alice and Bob needs to upload their respective certificate and private key, as well as the certificate a CA they trust. Typically, a simple PKI model is used where Alice and Bob have their certificates issued by the same Certificate Authority (CA), see fig. 36.3.

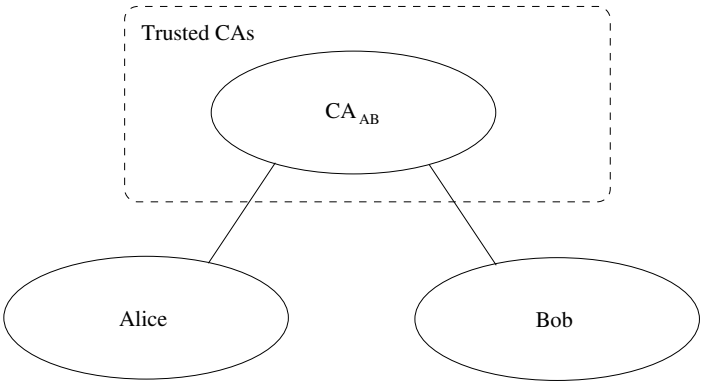


Figure 36.3: Alice and Bob have certificates issued by the same CA (e.g., their company CA). In this PKI model, Alice uploads the certificate of her CA, and trusts any certificate issued by that CA.

To generate certificates and private keys for Alice and Bob, you can e.g., use the *Easy-RSA* tools provided by OpenVPN[3]. The easiest way to upload certificates and keys to your WeOS unit(s) is via the WeOS web, see chapter 7.2.6 for more information. An example of the alternative to use the CLI to download to download a PKCS bundle (including Alice' certificate, private key and CA certificate) is shown below.

[3] OpenVPN home page, http://openvpn.net (March 2014).

Example

```
alice:/#> cert import pkcs password "AliceSecret" scp://foo@10.0.0.5/home/foo/alice.p12
Downloading alice.p12 from scp://foo...
foo@10.0.0.5's password:
alice.p12                                     100% 3064     3.0KB/s   00:00
Importing certificate alice...
OK
alice:/#> show cert
Type  Label           Common Name                    Expires
================================================================================
Pub   alice           MyServer                       Nov 26 13:35:42 2023 GMT
CA    alice           MyCA                           Nov 26 13:34:19 2023 GMT
Key   alice
alice:/#>
```

With the certificates installed on your WeOS unit, you can configure your SSL tunnel to use them by referring to their label, see the example for Alice below. Until she has configured what certificates to use as her own certificate and her CA certificate, the CLI will give warning messages.

Example

```
alice:/config/#> tunnel
alice:/config/tunnel/#> ssl 0
Creating new SSL tunnel 0, check your settings before activating the tunnel!
ssl0: Invalid settings: No certificate selected.

alice:/config/tunnel/ssl-0/#> certificate alice
ssl0: Invalid settings: No CA certificate selected.

alice:/config/tunnel/ssl-0/#> ca-certificate alice
alice:/config/tunnel/ssl-0/#> leave
alice:/#>
```

With the simple PKI model supported by WeOS (see fig. 36.3), Alice will accept connections from any VPN client presenting a valid certificate issued by her configured CA. (Similarly, Bob (and other VPN clients) will accept certificates presented by the VPN gateway if issued by the CA he has configured.)

36.1.4.1.1 Multiple VPN clients sharing the same certificate: Typically, each VPN client will have a unique certificate issued by their CA, but it is also possible for multiple VPN clients (Bob and Dave) to be configured with the same certificate. In this case the VPN gateway (Alice) must have the **"duplicate-cn"** (duplicate common name) setting enabled. If this setting is *enabled*, she will accept multiple parallel VPN sessions from clients with the certificate, but if it is *disabled* (default) she will tear down an existing VPN session if a new session is

established with the same certificate; she interprets that as if Bob has moved to a new location.

36.1.4.1.2 Use of username and password to authenticate clients: It is possible for Alice to use a second step authentication by requiring the VPN clients to provide a username and password (in addition to certificate). The example below shows an example of the credentials at the VPN client (Bob):

Example

```
bob:/config/#> tunnel ssl 0
bob:/config/tunnel/ssl-0/#> identity bob password builder
bob:/config/tunnel/ssl-0/#> leave
bob:/#>
```

Alice will either check these credentials against a local user database or towards a backend RADIUS server. Examples for both alternatives are shown below.

- *Local Database:* Configuration at the VPN Gateway (Alice)

  Example

  ```
  alice:/config/#> aaa
  alice:/config/aaa/#> local-db 1
  Creating new local db 1
  alice:/config/aaa/local-db-1/#> description openvpn-users
  alice:/config/aaa/local-db-1/#> username bob builder
  alice:/config/aaa/local-db-1/#> show
  Type               : plain
  Description        : openvpn-users
  Number of users    : 1

  Username    Password
  ----------  ----------
  bob         builder
  alice:/config/aaa/local-db-1/#> end
  alice:/config/aaa/#> end
  alice:/config/#> tunnel ssl 0
  alice:/config/tunnel/ssl-0/#> aaa-method local-db 1
  alice:/config/tunnel/ssl-0/#> leave
  alice:/#>
  ```

- *Backend RADIUS server:* Configuration at Alice (VPN Gateway)

Example

```
alice:/config/#> aaa
alice:/config/aaa/#> remote-server 1
Creating new remote server 1
alice:/config/aaa/remote-server-1/#> address 10.0.0.5
alice:/config/aaa/remote-server-1/#> password str4wb3rry
alice:/config/aaa/remote-server-1/#> end
alice:/config/aaa/#> end
alice:/config/#> tunnel ssl 0
alice:/config/tunnel/ssl-0/#> aaa-method remote-server 1
alice:/config/tunnel/ssl-0/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

And you also need to setup a RADIUS server (in the example above it is assumed to be located at 10.0.0.5 as in figs. 36.1 and 36.2). An example is to use a FreeRADIUS server, see http://www.freeradius.org for more information. Some hints are given below.

Example

```
In  /etc/freeradius/clients.conf:
client 10.0.0.1
shortname = 10.0.0.1
secret = str4wb3rry
nastype     = other


In  /etc/freeradius/users:
bob           Cleartext-password := "builder"
```

36.1.4.2 Cipher Suite Settings

To protect the SSL tunnel, you can chose between a set of data encryption and integrity protection alternatives:

- *Encryption:* WeOS supports various encryption alternatives based on Blowfish, DES and AES. Default is Blowfish (BF-CBC).
- *Message Authentication*: WeOS supports SHA1 and MD5 for message authentication (message integrity). Default is SHA1.

The session keys used for encryption and message integrity is derived as part of the authentication handshake at tunnel establishment. These session keys are renegotiated at a regular interval, which is controlled by the **"renegotiation-timeout"** setting (default 3600 seconds). The lowest timeout value configured by the client

or server is used for the SSL VPN session.

36.1.4.3 TLS Authentication Settings

WeOS supports an optional extra authentication of the SSL (TLS) tunnel by using something called “TLS Authentication”. This is an extra signature and encryption step performed with a static fixed key. This is done on all control packets for the tunnel, but not for the tunnelled data going through the tunnel (this data is encrypted already with the negotiated ciphers and keys). All control packets, including initial communication, received by the SSL VPN server will be checked and decrypted by this mechanism, and packets that does not match will be discarded immediately.

This extra authentication step makes an SSL VPN server less sensitive to DDOS attacks, especially when combined with using the UDP protocol for the tunnel. The server side software will not waste temporary memory by allocating connection data structures (TLS contexts, security associations, etc.) for bad incoming calls.

Using TLS Authentication and UDP together makes the VPN server to be completely quiet if (an attacker’s) packets arrive and are not signed by the correct key. Port scanning utilities will not detect the server in this mode.

TLS Authentication works for TCP as well, and has some of the benefits similar to the UDP mode, but the server network stack need to reply to the incoming TCP SYN packet to get a connection before it can determine if the key is valid. A port scanning utility will therefore be able to detect a server in TCP mode, and a heavy DDOS attack may potentially fill up all available connection slots on the server (socket memory/file descriptors).

TLS Authentication requires that a special OpenVPN Static key is imported into the system. The exact same key must be used on both ends of the tunnel for it to connect.

An OpenVPN static key can be generated with a computer with OpenVPN installed. Below is an example command line when using Linux:

Example

```
linux:~/>  openvpn --genkey --secret ta-example.key
```

To import the key, use the “cert” command. For details about the certificate store and operations, see chapter 7.2.6.

Example

```
alice:/#> cert import ovpn type key label mylabel ftp://192.168.2.10/ta-example.key
Downloading ta-example.key from ftp://192.168.2.10...
Connecting to 192.168.2.10:21 (192.168.2.10:21)
ta-example.key       100% |******************************|   636   0:00:00 ETA
Importing certificate mylabel...
OK
alice:/#>
```

An imported key label is referred from the tunnel configuration, and there is an optional direction setting that can be used together with the key.

The key direction setting is either “0” or “1”, and if it is used, the opposite sides of the tunnel need to have different settings for this parameter.

Commonly “0” is used on the server side, and “1” for the client side, but the opposite will also work. A specific key direction is optional, the default is to have the key work in both directions (bi-directional).

Example

```
alice:/config/#> tunnel
alice:/config/tunnel/#> ssl 0
alice:/config/tunnel/ssl-0/#> tls-auth label mylabel direction 0
alice:/config/tunnel/ssl-0/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

36.1.5 Other SSL tunnel settings

WeOS provides some additional SSL VPN settings:

- *Keepalive:* The **”keepalive”** setting is used (1) to keep session state in intermediate firewalls and NAT gateways (”ping” messages are send at a configurable interval when no data is transmitted), and (2) to restart the tunnel if the connection has gone down or if the server domain name resolves to a new IP address. Thus the *keepalive* setting also resembles a dead-peer-detection mechanism. Default is to send ”pings” at a 10 second interval, and to restart the tunnel (including DNS lookup) after 60 seconds if no response is received. The setting at the VPN server is pushed to the connecting VPN clients.
- *Data compression:* WeOS supports LZO compression for the SSL tunnel. When LZO compression is enabled, you can select to always compress or you set the **”adaptive”** mode when compression is dynamically turned on

and off by measuring its *usefulness* (if the data transmitted is determined to already be sufficiently compressed, additional LZO compression is disabled). Default setting is **"compression adaptive"**, i.e., compression is enabled in adaptive mode.

Note
As of WeOS v4.17.1, the compression setting at the VPN client and VPN server must match.

36.1.6 Related settings

An SSL tunnel is represented as a network interface in WeOS, and can be configured for routing, NAT and Firewall as other network interfaces. Additional hints on routing and firewall/NAT settings when using SSL VPNs are provided in the following sections.

36.1.6.1 Routing and SSL VPNs

In HOST-NET setups (fig. 36.1, the VPN server typically *pushes* routing information for relevant IP subnets to the VPN clients during tunnel establishment (see also section 36.1.3.3).

Below some other aspects of routing and SSL VPNs are listed:

- Blackhole routes: To ensure that traffic intended to be sent encrypted via your SSL tunnel is dropped by your VPN client or server when the tunnel is down, you can use *blackhole routes* (section 26.1.4.3). An example for Alice in fig. 36.2 is shown below, but a similar configuration can be used at the VPN client (Bob).

  Example
  ```
  alice:/config/#> ip
  alice:/config/ip/#> route 10.0.0.0/16 null0 200
  alice:/config/ip/#> leave
  Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
  alice:/#>
  ```

- Routing in NET-NET use case: In the NET-NET setup shown in fig. 36.2 there are some different alternatives for Alice and Bob to learn about the routes available at the peer side.

– *Dynamic routing:* It is possible to use dynamic routing protocols such as OSFP (chapter 27) or RIP (chapter 28) at Alice and Bob to exchange routes. Below is a sample RIP configuration at Alice (fig. 36.2), but the setup would be the same at the VPN client (Bob).

Example

```
alice:/config/#> router
alice:/config/router/#> rip
Activating RIP with default settings, type 'abort' to cancel.
alice:/config/router/rip/#> network vlan1
alice:/config/router/rip/#> network ssl0
alice:/config/router/rip/#> leave
Configuration activated. Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

– *Static routing:* When using static routing to route traffic between the office networks in fig. 36.2, Bob should be configured with a static IP address on his SSL interface (rather than acquiring it dynamically as described in section 36.1.3.2). Here we assume that Bob's SSL interface has IP address *10.0.2.2/24*. In addition, Bob can disable routes or addresses pushed by Alice using the **"no pull"** setting.

Example

```
bob:/config/#> iface ssl0
bob:/config/iface-ssl0/#> inet static
bob:/config/iface-ssl0/#> address 10.0.2.2/24
bob:/config/iface-ssl0/#> end
bob:/config/#> tunnel
bob:/config/tunnel/#> ssl 0
bob:/config/tunnel/ssl-0/#> no pull
bob:/config/tunnel/ssl-0/#> end
bob:/config/tunnel/#>
```

Alice and Bob can then setup static routes to their respective networks. Here we show how Alice defines a static route to the office network at Bob (10.0.3.0/24).

Example

```
alice:/config/#> ip
alice:/config/ip/#> route 10.0.3.0/24 10.0.2.2
alice:/config/ip/#>
```

Bob would add a corresponding static route to the central office subnet (**"route 10.0.0.0/24 10.0.2.1"**).

36.1.6.2 Firewall and NAT

VPN clients and servers typically have their firewall enabled. To allow the intended traffic to flow through the tunnel, suitable *filter allow* rules should be added to your the VPN units. An example for the VPN gateway (Alice) in figs. 36.1 and 36.2 is shown below:

Example

```
alice:/config/#> ip
alice:/config/ip/#> firewall
alice:/config/ip/firewall/#> filter allow in ssl0 out vlan1
alice:/config/ip/firewall/#> filter allow in vlan1 out ssl0
alice:/config/ip/firewall/#> leave
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

The VPN gateway (Alice) is typically used as a NAT gateway towards the Internet (interface *vlan2* in figs. 36.1 and 36.2. Below in an example of NAT configuration, where *ping* (ICMP) and DNS requests are blocked on the upstream Interface (vlan2).

Example

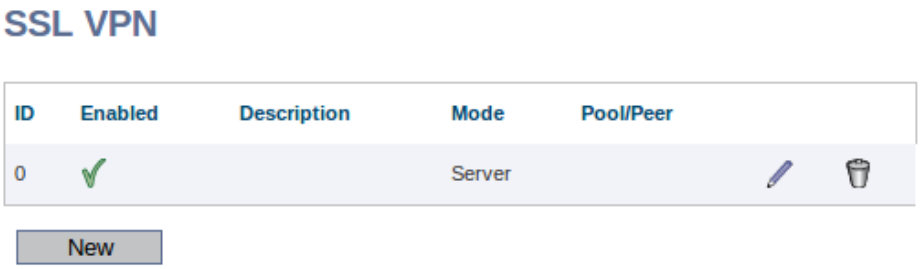
```
alice:/config/ip/firewall/#> nat type napt out vlan2 addfilter
alice:/config/ip/firewall/#> filter deny in vlan2 proto udp dport 53
alice:/config/ip/firewall/#> filter deny in vlan2 proto tcp dport 53
alice:/config/ip/firewall/#> filter deny in vlan2 proto icmp
alice:/config/ip/firewall/#> filter allow proto icmp
alice:/config/ip/firewall/#> leave
Starting ZeroConf IPv4 link-local daemon ..................... [ OK ]
Configuration activated.  Remember "copy run start" to save to flash (NVRAM).
alice:/#>
```

36.2 Managing SSL VPN settings via the web interface

36.2.1 Manage SSL VPN via the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ SSL VPN

The main SSL VPN configuration pages a list of currently configured SSL VPN tunnels.



The list shows currently configured SSL VPN tunnels, and displays some of the tunnel settings.

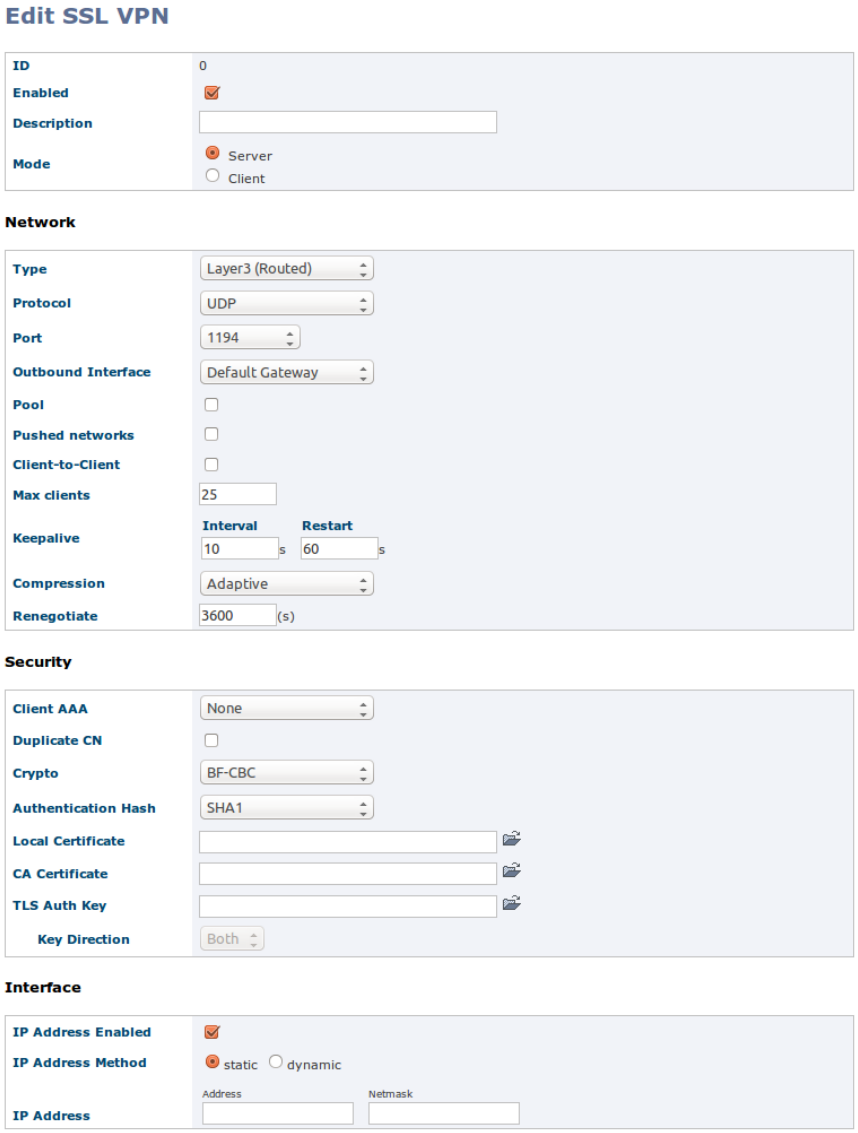
ID	The tunnel index. Each configured tunnel is identified by a number for maintenance purposes. This ID is of local significance only.
Enabled	A green check-mark means enabled and a dash means disabled.
Description	A description for the tunnel.
Mode	Client or Server mode
Edit	Click this icon to edit the settings of a VPN tunnel.
Delete	Click this icon to remove a VPN tunnel. **Note:** Tunnels which are not intended to be used should either be *deleted* or *disabled* (section 36.2.2).

36.2.2 Configure new or existing SSL VPN tunnel via the web interface

Menu path: Configuration ⇒ VPN & Tunnel ⇒ SSL VPN ⇒ **New**

Menu path: Configuration ⇒ VPN & Tunnel ⇒ SSL VPN ⇒  (Instance)

When clicking the **New** button the window to configure a new SSL VPN tunnel appears. To edit an existing tunnel, click on the **Edit** button for the tunnel.



General part:

Instance number	The tunnel index. Each configured tunnel is identified by a number for maintenance purposes. This ID is of local significance only.
Enabled	A tunnel can be configured as **`Enabled`** or **`Disabled`**. **Note:** Tunnels which are not intended to be used should either be *deleted* (section 36.2.1) or *disabled*.
Description	A descriptive text for this tunnel.
Mode	Client or Server mode

Network part:

Type	Set the tunnel to be in Layer2 (Bridged) or Layer3 (Routed) mode. Layer2 is often described as TAP (network tap) and Layer3 as TUN (network tunnel)
Protocol	Protocol to encapsulate the traffic in. TCP or UDP
Port	TCP/UDP listen port.
Outbound Interface	Outbound interface. The tunnel will only connect through the specified interface. If no outbound is specified, the interface which is connected to the default gateway will be used.
Pool (server mode)	IP address to be pushed to all clients connecting to us o netmask is not possible to set when type is Layer3 (routed) and is mandatory in layer3 (bridged). If not set in layer3 mode, the default is to set according to IP class of the start address.
Pushed networks (server mode)	Define networks to push towards all clients.
Client-to-Client (server mode)	Allow clients to communicate with each other.
Max clients (server mode)	How many clients should max be possible to connect to this tunnel if more then this tries to connect, they will be rejected.

Continued on next page

	Continued from previous page
Remote peer (client mode)	Remote peer IP address, or DNS domain name. When acting as client, the peer setting defines the remote server to connect to. As server it can be used to allow a single client or not. Use 'no peer' to allow connections from ANY client
Pull (client mode)	Allow pushed network routes from the server.
Keepalive	Send keep-alive probes over the tunnel to make sure that stateful firewalls gets updated as expected, they is only sent as long as there is no traffic on the tunnel. Interval - The interval to send probes, if there are not traffic on the tunnel Restart - Force restart of the ping probe, this will force reload of DNS for example, this is very useful when dealing with DynDNS (section 19.3.3).
Compression	Set preferred compression setting to be used on the tunnel. In client mode this can be overridden by the server.
Renegotiate	Set the renegotiation time for the data channel, this can be set on both the client and the server, if so, the lowest value will be used.

Security part:

Client AAA (server mode)	Enable authentication of clients
Identity (client mode)	Provide authentication when connecting to the server
Duplicate CN	Allow multiple clients to connect with the same Common Name, without this option it will disconnect a current client when a new connects with the same Common Name.
Crypto	Selects crypto cipher to use.
Authentication Hash	Selects authentication hash to use
Local Certificate	Local certificate to use, including private key
CA Certificate	CA certificate used for signing our certificate, without private key.
TLS Auth Key	TLS authentication key for extended security
Key Direction	Direction for TLS authentication key

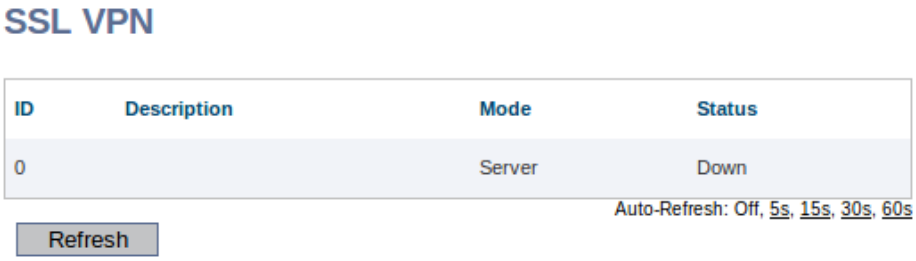
Interface part:

IP Address Enabled	Enable IPv4 address on the SSL interface
IP Address Method	Select Method for IPv4 address, static or DHCP
IP Address	The IP address for the SSL interface

36.2.3 View SSL VPN Tunnel Status

Menu path: Status ⇒ VPN & Tunnel ⇒ SSL VPN

The **SSL VPN Status** page lists the status of configured SSL VPN tunnels.



36.3 Managing SSL VPN settings via the CLI

The WeOS unit can be configured as SSL VPN server gateway (waiting for clients to connect), or as an SSL VPN client (initiating connections). We start out by shown the CLI commands available when configuring an SSL VPN *server* gateway (**”[no] server”** command set to **”server”**).

Command	Default	Section
General SSL VPN Server Gateway Settings		
tunnel		Section 35.3.1
[no] ssl <INDEX>		Section 36.3.1
server	Server	Section 36.3.2
[no] enable	Enabled	Section 36.3.3
[no] description <STRING>	empty	Section 36.3.4
[no] type <layer2|layer3>	layer3	Section 36.3.5
[no] pool start <IPADDR> <num <NUM>|end <IPADDR>> [netmask NETMASK]	Disabled	Section 36.3.6
[no] push-network <NETWORK/LEN>	Disabled	Section 36.3.7
Authentication Settings		
[no] certificate	Empty	Section 36.3.9
[no] ca-certificate	Empty	Section 36.3.10
[no] tls-auth label <KEY LABEL> [direction <0|1>]	Empty	Section 36.3.11
[no] aaa-method <remote-server <ID>| local <ID>>	Disabled	Section 36.3.12
Data Security Settings		
[no] crypto <aes128-cbc|. . . >	aes128-cbc	Section 36.3.14
[no] auth <sha1|md5>	sha1	Section 36.3.15
Additional/Advanced Settings		
[no] protocol <tcp|udp>	UDP	Section 36.3.16
[no] port	1194	Section 36.3.17
[no] outbound	Auto	Section 36.3.18
[no] keepalive <interval <SEC>	interval 10	Section 36.3.19

Continued on next page

Continued from previous page

Command	Default	Section
restart <SEC>>	restart 60	
[no] compression [adaptive]	Adaptive	Section 36.3.20
[no] renegotiation-timeout <SECONDS>	3600	Section 36.3.21
[no] client-to-client	Disabled	Section 36.3.22
[no] duplicate-cn	Disabled	Section 36.3.23
Show SSL VPN Status		
show tunnel ssl [ID]		Section 36.3.25
See also (Interface and Firewall Settings)		
iface ssl<ID> inet <static|dynamic|dhcp>	Dynamic (SSL)	Sec. 19.6.1
Various Interface settings	. . .	See Sec. 19.6
ip		
[no] firewall	Disabled	Section 31.3.1
Various Firewall/NAT settings	. . .	See Sec. 31.3

The table below shows the available CLI commands when configuring the WeOS unit as SSL client (**"[no] server"** command set to **"no server"**).

Command	Default	Section
General SSL VPN Settings		
tunnel		Section 35.3.1
[no] ssl <INDEX>		Section 36.3.1
no server	Server	Section 36.3.2
[no] enable	Enabled	Section 36.3.3
[no] description <STRING>	empty	Section 36.3.4
[no] type <layer2|layer3>	layer3	Section 36.3.5
[no] peer <ADDRESS|DOMAIN>	empty	Section 36.3.8
Authentication Settings		
[no] certificate	Empty	Section 36.3.9
[no] ca-certificate	Empty	Section 36.3.10
[no] tls-auth label <KEY LABEL>	Empty	Section 36.3.11

Continued on next page

Continued from previous page

Command	Default	Section
[direction <0|1>]		
[no] identity <USERNAME> password <PASSWORD>	Disabled	Section 36.3.13
Data Security Settings		
[no] crypto <aes128-cbc|. . . >	aes128-cbc	Section 36.3.14
[no] auth <sha1|md5>	sha1	Section 36.3.15
Additional/Advanced Settings		
[no] protocol <tcp|udp>	UDP	Section 36.3.16
[no] port	1194	Section 36.3.17
[no] outbound	Auto	Section 36.3.18
[no] keepalive <interval <SEC> restart <SEC>>	interval 10 restart 60	Section 36.3.19
[no] compression [adaptive]	Adaptive	Section 36.3.20
[no] renegotiation-timeout <SECONDS>	3600	Section 36.3.21
[no] pull	Enabled	Section 36.3.24
Show SSL VPN Status		
show tunnel ssl [ID]		Section 36.3.25
See also (Interface and Firewall Settings)		
iface ssl<ID> inet <static|dynamic|dhcp>	Dynamic (SSL)	Sec. 19.6.1
Various Interface settings	. . .	See Sec. 19.6
ip		
[no] firewall	Disabled	Section 31.3.1
Various Firewall/NAT settings	. . .	See Sec. 31.3

36.3.1 Managing SSL VPN Tunnels

Syntax `[no] ssl <INDEX>` where INDEX is a number greater or equal to 0.

Context Tunnel Configuration context

Usage Create, delete, or modify an SSL VPN tunnel. Use **”ssl <INDEX>”** to create a new SSL tunnel, or to enter the configuration context of an existing SSL tunnel. (To find the index of configured tunnels, use **”show tunnel”** as described in section 35.3.1,)

Use **”no ssl <INDEX>”** to remove a specific SSL VPN tunnel, or **”no ssl”** to remove all configured SSL VPN tunnels.

Use **”show ssl <INDEX>”** to show all settings of a specific SSL tunnel (also available as **”show”** command within the SSL VPN Configuration context).

> **Note**
> Tunnels which are not intended to be used should either be *deleted* or *disabled* (section 36.3.3).

Default values Not applicable.

36.3.2 Change tunnel mode (Server/Client)

Syntax `[no] server`

Context SSL VPN Configuration context

Usage Set the tunnel in server or client mode, use **”no server”** for client mode.

Default values Server

36.3.3 Enable/disable a SSL VPN tunnel

Syntax `[no] enable`

Context SSL VPN Configuration context

Usage Enable or disable a SSL VPN tunnel. A disabled tunnel will be deactivated, but keeps its configuration settings.

Use **”enable”** to enable and **”no enable”** to disable an SSL VPN tunnel.

Use **”show enable”** to show whether this SSL VPN tunnel is enabled or disabled.

Note
Tunnels which are not intended to be used should either be *deleted* (section 36.3.1) or *disabled*.

Default values Enabled

36.3.4 SSL VPN Description Setting

Syntax `[no] description <STRING>`

Context SSL VPN Configuration context

Usage Set or remove the SSL VPN description string.
Use **"description <STRING>"** to set a description for this database.

Use **"no description"** to remove the current description.
Use citation marks around the string if you want to have a description containing space characters.

To view the current description, use **"show description"**.

Default values Empty.

Examples

Example

```
example:/config/tunnel/ssl-19/#> description secrets
 or ...
example:/config/tunnel/ssl-19/#> description ''Office tunnel''
```

36.3.5 Configure tunnel type

Syntax `[no] type <layer2|layer3>`

Context SSL VPN Configuration context

Usage Change which type of tunnel you want to use, select layer2 (sometimes called bridged) or layer3 (sometimes called routed). **"no type"** reset to layer3.

Default values layer3

36.3.6 Configure an address pool

Syntax `[no] pool start <IPADDR> <num <NUM> | end <IPADDR>> [netmask NETMASK]`

Context SSL VPN Configuration context (Only valid when server)

Usage Auto configure all clients connecting to us, if netmask is omitted it will be set to the default mask for the address class for the start address.

> **Note**
> The address of the server interface will be untouched, you will need to configure it manually from the interface context for the ssl-interface Sec. 19.6.1.

> **Example**
> `example:/config/tunnel/ssl-19/#>` **pool 192.168.253.2 num 10**

Default values Disabled

36.3.7 Push networks to connecting clients

Syntax `[no] push-network <NETWORK/LEN>`

Context SSL VPN Configuration context (Only valid when server)

Usage This is a part of the auto-configuration of the clients, push networks (Max is 10) to the clients, these routes will automatically be set as routes to us as long as the client has **”pull”** enabled.

Default values Disabled

36.3.8 Change remote peer

Syntax `[no] peer <ADDRESS|DOMAIN>`

Context SSL VPN Configuration context (Only valid when client)

Usage Set the peer for the client to connect to.

Default values Disabled

36.3.9 Select local certificate

Syntax `[no] certificate <LABEL>`

Context SSL VPN Configuration context

Usage Select local certificate (and associated private key), i.e., the certificate by which this unit will authenticate itself. The **”LABEL”** is the reference of the certificate when imported to the WeOS unit. The certificate must be signed of by the CA certificate set in Section 36.3.10 Use **”show certificate”** to show the local certificate setting.

Default values Empty

36.3.10 Select CA certificate

Syntax `[no] ca-certificate <LABEL>`

Context SSL VPN Configuration context

Usage Select CA certificate, i.e., the certificate by which this unit will authenticate itself. The **”LABEL”** is the reference of the certificate when imported to the WeOS unit. Use **”show ca-certificate”** to show the CA certificate setting.

Default values Empty

36.3.11 Enable TLS authentication

Syntax `[no] tls-auth label <KEY LABEL> [direction <0|1>]`

Context SSL VPN Configuration context

Usage Enable TLS authentication. **”KEY LABEL”** is the label of an OpenVPN key to be used for authentication. The direction is optional and not setting it means to use the key in both directions (bi-directionally).

Default values Empty (disabled)

36.3.12 Configure AAA remote authentication

Syntax `[no] aaa-method <remote-server | local> <ID>`

Context SSL VPN Configuration context

Usage Require an extra authentication after the certificate exchange. Require to first create a remote-server or a local user database in the AAA context. Section 21.3

Example

```
example:/config/tunnel/ssl-19/#> aaa-method local 1
 or ...
example:/config/tunnel/ssl-19/#> aaa-method remote-server 1
```

Default values Disabled

36.3.13 Configure authentication identity

Syntax `[no] identity <USERNAME> password <PASSWORD>`

Context SSL VPN Configuration context (Only valid when client)

Usage This is only required if the server is configured to require an extra authentication layer after the certificate exchange. Section 36.3.12

Example

```
example:/config/tunnel/ssl-19/#> identity user1 password secrets
```

Default values Disabled

36.3.14 Change cryptographic cipher

Syntax `[no] crypto <<bf-cbc|des-ede3-cbc|aes128-cbc|aes192-cbc|aes256-cbc»`

Context SSL VPN Configuration context

Usage Set the crypto to use, must match on both the client and the server. **"no crypto"** disables all encryption, all traffic will pass over the tunnel unencrypted.

Default values aes128-cbc

36.3.15 Change authentication hash

Syntax `[no] auth <sha1|md5>`

Context SSL VPN Configuration context Authenticate packets with HMAC using message digest. Use **"no auth"** to disable the authentication hash.

Default values sha1

36.3.16 Configure protocol

Syntax `[no] protocol <tcp|udp>`

Context SSL VPN Configuration context

Usage Select the protocol to encapsulate the traffic in.

Default values UDP

36.3.17 Configure port

Syntax `[no] port <PORT>`

Context SSL VPN Configuration context

Usage In client mode, this selects the port to connect to on the server, in server mode, this selects which port to listen for incoming connections on.

> **Note**
> A neat function when using SSL VPN is to listen on TCP (Section 36.3.16) port 443, this will allow the tunnel to pass almost all firewalls, since the traffic will look like it is HTTPS. To achieve this in server mode you will have to move HTTPS on the WeOS unit to a separate port. See Section 7.3.45.

Default values 1194

36.3.18 Configure Outbound Interface

Syntax `[no] outbound <IFACE>`

Context SSL VPN Configuration context

Usage Set the outbound interface of this tunnel.

Use **"no outbound"** to automatically select the interface leading to the *default gateway* as outbound interface.

Use **"show outbound"** to show the configured *outbound interface* for this tunnel. **"Default Gateway"** is shown if the interface leading to the default gateway should be used as outbound interface.

Default values Auto (**"no outbound"**)

36.3.19 Change keepalive settings

Syntax `[no] keepalive <interval <SEC> restart <SEC>>`

Context SSL VPN Configuration context

Usage Send keepalive probes over the tunnel to make sure that stateful firewalls gets updated as expected, they is only sent as long as there is no traffic on the tunnel.

- interval - The interval to send probes, if there are not traffic on
- restart - Force restart of the ping probe, this will force reload of DNS for example, this is very useful when dealing with DynDNS (section 19.3.3).

Note: In server mode, this settings will also be pushed to the clients, if **"pull"** is enabled in the clients, they will not need to configure keepalive settings.

Use **"show keepalive"** to view current keepalive settings.

Default values interval 10 restart 60

36.3.20 Configure compression settings

Syntax `[no] compression [adaptive]`

Context SSL VPN Configuration context

Usage Toggle compression settings, **"no compression"** will disable all compression. **"compression adaptive"** will result in that SSL VPN tries to find out if the traffic is encrypted, if not it will encrypt it. This will have performance

penalty if all traffic already is encrypted. This setting must match on client and server to get the traffic going. In server mode, this setting will also be pushed to the clients.

Default values Adaptive

36.3.21 Change renegotiation timeout

Syntax `[no] renegotiation-timeout <SECONDS>`

Context SSL VPN Configuration context

Usage Set the renegotiation time for the data channel, this can be set on both the client and the server, if so, the lowest value will be used. To disable renegotiation use **"no renegotiation-timeout"** on both ends.

Default values 3600 seconds

36.3.22 Change client to client communication

Syntax `[no] client-to-client`

Context SSL VPN Configuration context (Only valid when server)

Usage If enabled all clients will be able to communicated with each other.

> **Note**
> No traffic will be passed through the normal network stack, e.g firewall rules will not be possible. If you want the possible to set firewall rules per client you have to create multiple server instance and route between the instances.

Default values Disabled

36.3.23 Allow/deny clients with the same CN

Syntax `[no] duplicate-cn`

Context SSL VPN Configuration context (Only valid when server)

Usage The normal behaviour is to deny clients which connect with a CN (common name) that is already connected. Enabling this setting will allow the second connection.

> **Note**
> This is a serious security risk, use only if you know what you are doing, you should look to combine this with an aaa-method (Section 36.3.12)

Default values Disabled

36.3.24 Change pull settings

Syntax `[no] pull`

Context SSL VPN Configuration context (Only valid when client)

Usage In client mode the client may receive routes and ip address from the server. When setting **"`no pull`"** all these settings the server tries to push, will be discarded.

Default values Enabled

36.3.25 Show SSL Tunnel Status

Syntax `show tunnel ssl [ID]`

Context Admin Exec context.

Usage Show the status for all or for a specific SSL tunnel.

Default values If no tunnel ID is specified, the status of all SSL tunnels is shown.

Chapter 37

WeConnect

This chapter describes the WeOS support for the Westermo WeConnect service. Westermo WeConnect is a centralised on-line connectivity service offered by Westermo as a separate product (not normally included in the purchase of a WeOS product).

The idea of the service is to connect equipment and networks through the Internet in an easy way, but at the same time safe and encrypted using standard VPN features.

The secured networks set up by WeConnect can be used in many ways such as remote management, interconnection of remote network locations, centralised logging and alerts, emergency access, etc.

WeConnect is managed with an on-line web portal. In this portal you are able to define your virtual secure networks and create VPN configurations for WeOS units and other clients and nodes. For more information about the service and how to sign up for it, please visit Westermo's home page at http://www.westermo.com/.

In WeOS, WeConnect is set up using an installer that takes you through some easy steps that takes care of the configuration for you.

You only need the Secure Network Code and the One Time Password (OTP) for your unit from the WeConnect web portal to be able to run the installer. WeOS will use the Code and OTP to make an encrypted download of the configuration and certificates from the on-line portal service. The VPN will automatically connect when the installer procedure is completed and your unit instantly becomes part of your secured network.

WeConnect utilises these standard features in WeOS for its operations:

- *SSL VPN* - This is used for the encrypted tunnel that connects to your WeConnect secure network.
- *RIP* - This protocol is run inside the SSL VPN tunnel to receive routes from other units and networks in the secure network. It also announces the local networks on your unit so they can be reached remotely.
- *Firewall* - Automatic forward rules for the WeConnect SSL VPN tunnel are added. It is recommended that you use the firewall, but not mandatory.

The SSL VPN tunnel is run in UDP mode. This makes the WeConnect service perform well on most types of Internet connections. There is no requirement of fixed public IP number for your unit, and accessing the Internet via external firewalls and NAT will work in most cases.

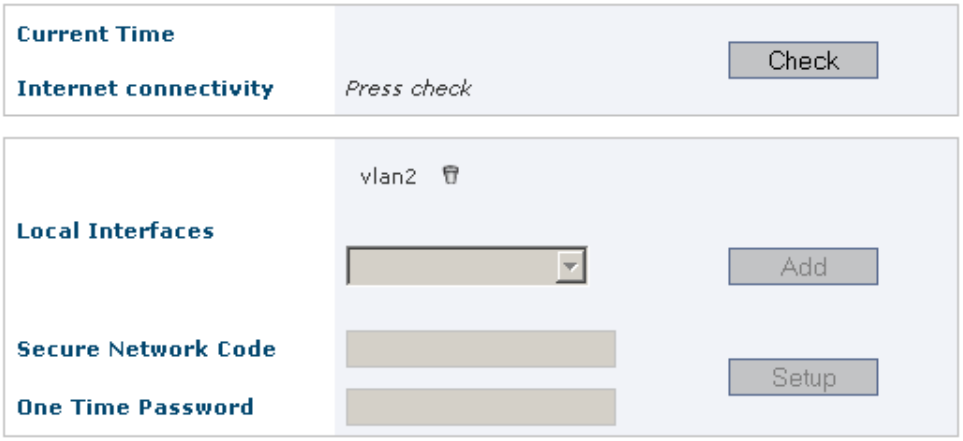
Note

WeConnect is using the IPv4 networks **198.18.0.0/16** and **198.19.0.0/16** internally for its operation. You can not use these networks, or subnets within these networks, for other purposes on your WeOS unit while using WeConnect.

37.1 Installing WeConnect via the Web

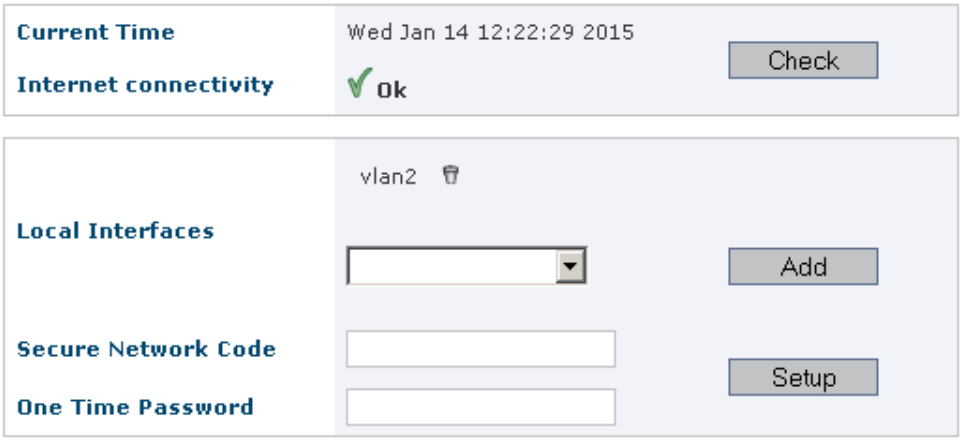
Menu path: WeConnect

When you enter the WeConnect installer, you will be greeted by an introductory text and these input fields:



The first thing to do is to click the **Check** button. This will test the connectivity to Internet and the WeConnect portal, and to check that the local time on your unit is properly set.

If all goes well with the check, the rest of the input fields will be enabled:

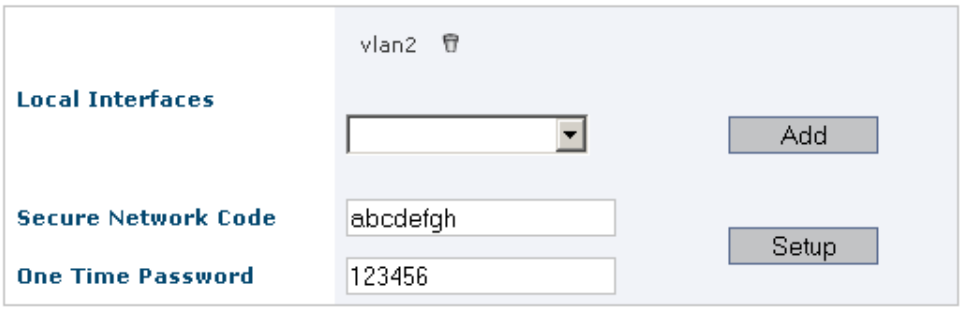


Please see the troubleshooting section if you get stuck on an error message dur-

ing this guide (section 37.3).

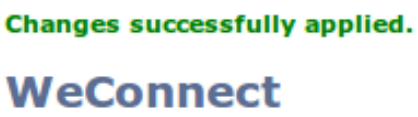
By default, all VLAN interfaces except for the interface that connects to Internet are added and exported into WeConnect (in the example above, vlan2 will be exported). You can remove and add interfaces manually using the delete icon, and the interface select menu together with the **Add** button.

Enter the **Secure Network Code** and **One Time Password** obtained from the WeConnect on-line portal:



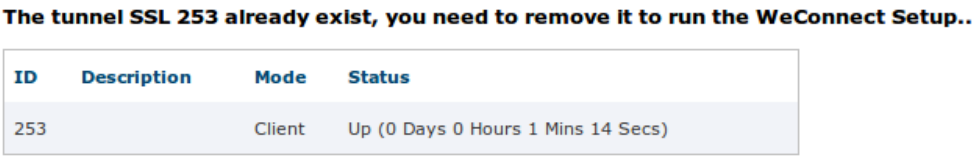
Click **Setup**. This downloads and installs the configuration and certificates that enables WeConnect on your unit.

This is shown at the top of the page if everything went OK:



The setup is now complete, and the unit will connect to your secure network automatically.

After installing WeConnect you can go to the WeConnect page again via the menu to see the status of the SSL VPN tunnel. Example, with the tunnel up:



The warning message shown above the status information does not indicate an error, but serves as a notification that WeConnect is already set up and the installer can not be run again.

WeConnect uses the SSL VPN tunnel ID 253, and you can also find tunnel status via the ordinary status menu ”Status ⇒ VPN & Tunnel ⇒ SSL”, see section 36.2.3.

37.2 Installing WeConnect via the CLI

WeConnect in the CLI is not set up in the normal configuration mode, but as a command of its own (**”weconnect”**) in the Admin Exec context.

Example

```
example:/#>  weconnect

============================
Welcome to Westermo WeConnect!
============================
WeConnect allows secure remote access to both the network behind the WeOS
devices and the devices themselves. WeConnect solves the complexity of
managing VPNs over the internet.

With WeConnect users can easily and securely connect to any IP-device on
the network using their normal PC, smartphone or tablet.

If you do not yet have an account, contact your local Westermo reseller or
visit http://www.westermo.com/ for further information.

This installation procedure will download configurations and certificates
for connecting to the Westermo WeConnect service.

Certificates will be installed and the current running configuration will
be changed and saved as startup configuration as part of this procedure.

Do you want to continue (y/N)?
```

Enter **”y”** here to continue.

At this point the installer will test the connectivity to Internet and that the WeConnect portal can be reached. The local time on your unit is also checked so that it is properly set. Please see the troubleshooting section if you get stuck on an error message during the install (section 37.3).

If all goes well with the checks, you will be asked to enter the **Secure Network Code** and **One Time Password** obtained from the WeConnect on-line portal.

Example

```
Please enter the identification information provided by the WeConnect web
portal.

  Secure Network Code:  abcdefgh
  One Time Password:  123456

Please specify a list of the interfaces that will be shared over WeConnect.
 [vlan2, vlan3]:
```

You are asked to enter which interfaces that should be exported into WeConnect. By default, all VLAN interfaces except for the interface that connects to Internet are suggested inside square brackets at the prompt (in the example above: vlan2 and vlan3). If the suggested list is OK, just press enter to continue.

If you want to add or remove any interface to the list, you should manually enter the *whole list* of interfaces that you want to be exported into WeConnect, separated with commas.

To export interfaces vlan3 and gre2 (but not vlan2):

Example

```
Please specify a list of the interfaces that will be shared over WeConnect.
 [vlan2, vlan3]:  vlan3, gre2
```

After this, the configuration and certificates for WeConnect will be downloaded and installed on your unit.

Example

```
Downloading and installing configuration and certificates. Please wait...
Installation OK

WeConnect installation complete!

The SSL tunnel status can be viewed with the command:
  show tunnel ssl 253

The WeOS configuration was changed as part of the installation.
Run ‘‘copy run start’’ to save to flash (NVRAM).
Starting RIP daemon ..................................... [ OK ]
Starting SSL tunnel daemon .............................. [ OK ]
```

The configuration is changed but is *not* saved permanently.

Note
It is possible at this point to undo the configuration by rebooting, or copying a saved configuration to “running-config”, but this will not remove the installed certificate for the VPN tunnel. The certificate needs to be manually deleted to completely undo the install. (See section 7.2.6 for more information on certificate management.)

Do not forget to save the settings

Example

```
example:/#> copy running-config startup-config
```

The setup is now complete, and the unit will connect to your secure network automatically. Use the command **”show tunnel ssl 253”** to see the status of the tunnel.

37.3 Troubleshooting

These are error messages that you may get while running the installer.

37.3.1 At least two interfaces needed

You will need at least two interfaces for WeConnect to work. One of the interfaces is the uplink used to connect to Internet. You will not have much use of WeConnect if you do not have at least one additional interface that is exported to the secure network.

37.3.2 Unable to connect to the WeConnect servers

This message usually means that you can not reach the Internet. You need to configure your unit so that it has Internet access in some way. Please check that you have got a DHCP lease, or if configuring IP settings manually, that you have a proper IP address and netmask, a default route, and a name server (DNS) configured.

The download of configuration and certificates is using HTTPS (TCP port 443) to access the on-line servers. If you are behind a firewall, please make sure it does not block this port for outbound connections. (And while you are at it, you should also check that outbound UDP traffic to port 1194 is allowed as that will be used for the SSL VPN tunnel later.)

37.3.3 The system time is incorrect

The SSL VPN functionality is using certificates for authentication. These certificates are valid for a defined time range (WeConnect certificates are normally valid for several years). It is important that the system time is correctly set to the current time as the tunnel will not be established if the time is outside the validity time range of the certificate.

A WeOS unit that has been stored for a long time without power attached may reset the internal clock. The internal power source (similar to a battery) is slowly drained.

You need to manually set the current time in the unit, or even better, configure NTP to automatically set the time from a time server.

37.3.4 The connection to the WeConnect download service was interrupted

This error message is shown if one of these things occured:

- You entered incorrect information for Secure Network Code or One Time Password.
- The One Time Password had already been used before.
- The Internet connection went away before or during the file transfer.
- The downloaded file was corrupted during the transfer.
- The WeConnect servers had some kind of problem.

Please check that you have the correct ID and password, and re-run the WeConnect installation. If the information seems OK but still does not work, you can try generating a new one time password in the WeConnect portal and re-run the installation.

37.3.5 RIP is already configured

WeConnect uses RIP for handling routes inside a secure network. RIP must not be configured previously when running the installer. You may add your own RIP settings after the installer, but this is not recommended. All routes picked up by RIP will propagate to the WeConnect secure network. If you need dynamic routing for other purposes, please consider using OSPF for that, and keep RIP exclusive for WeConnect.

You may also get this error message if WeConnect is already installed.

37.3.6 The tunnel SSL 253 already exist

This means that WeConnect is already installed. The special tunnel number 253 is used exclusively for WeConnect.

37.3.7 TFTP is very slow over WeConnect

The traffic sent from one node is put into a VPN tunnel that is terminating at Westermo's WeConnect servers on the Internet. It is then re-routed to the target node via the target node's VPN tunnel. This causes a high latency for the traffic going back and forth via this service. High latency is very harmful for TFTP performance as explained in the note in section 7.1.1.1.2.

A WeOS firmware update using a PKG file, using TFTP and going via WeConnect typically takes several hours due to the latency. Please avoid using TFTP at all over WeConnect. Other protocols such as FTP, HTTP and SSH use TCP and these protocols work a lot better for transfering files.

37.3.8 I need to remove or reinstall WeConnect

To reinstall, you should first remove the old settings, and then run the installer again.

There is no automated way of removing WeConnect, it has to be done manually. Easiest way is to do a factory reset on your unit, but sometimes that is not an option.

Restoring an old configuration that was saved before you installed WeConnect is one way. But this procedure will not remove the WeConnect certificates and keys. You need to manually remove these.

If you need to manually remove WeConnect but keep all other configuration as it is, you should do these steps:

- Delete the SSL VPN tunnel with ID 253.
- Delete RIP configuration.
- Delete WeConnect related certificates and keys.
- If you have the firewall enabled, you may need to remove firewall forward filter rules that are related to interface ssl253.

Part V

Serial Port Management and Applications

Chapter 38

Serial Port Management

This chapter describes serial port features and management support in WeOS, thus only apply to WeOS products equipped with serial port(s). WeOS products can be equipped with three types of serial ports:

- *RS-232 serial port:* Serial ports that support RS-232.
- *"Combo" RS-232/422/485 serial ports:* Serial ports capable of using RS-232, RS-422 and RS-485 as serial protocol. The user can select the type of serial protocol to run.
- *"Combo" RS-422/485 serial ports:* Serial ports capable of using RS-422 and RS-485 as serial protocol. The user can select the type of serial protocol to run.

For details on the serial port capabilities and pin-out of your specific WeOS product, please see the User Guide of the product (section 1.5).

The serial port can be used by several different serial applications:

- *Serial extender*: The WeOS units with serial ports can be used to extend a serial connection over a TCP/IP network. Chapter 39 provides information on WeOS *Serial Over IP* support.
- *Modem replacement:* WeOS units are able to interpret AT commands, the can be used to replace modems. The *modem replacement* support is integrated with WeOS Serial Over IP functions, and is described in chapter 39.
- *Modbus gateway:* WeOS units can act as gateway between Modbus units on the serial port and Modbus units on TCP/IP networks. Chapter 40 describes WeOS Modbus gateway support.

- *Microlok gateway:* WeOS units can act as gateway between Microlok units on the serial port and Microlok units on UDP/IP networks. Chapter 41 describes WeOS Microlok gateway support.
- *PPP network interface:* For WeOS units with proper software level, the serial port can be used to establish PPP connections, with or without external modem. See chapter 33 for more information.

38.1 Overview of Serial Port Management

The table below presents the serial port management features in WeOS.

Feature	Web	CLI	General Description
Enable/disable Serial Port[1]	X	X	
Select type (RS-232/422/485)[2]	X	X	
Speed	X	X	Section 38.1.1
Data bits	X	X	-”-
Parity	X	X	-”-
Stop bits	X	X	-”-
Hardware flow control[3]	X	X	Section 38.1.2
Software flow control	X	X	Section 38.1.3
Termination[2]	X	X	

Note
For details on RS-422 or RS-485 pinouts, see the User Guide of your specific product (listed in Chapter 1). For background information on RS-422 and RS-485 technology and applications, see the Westermo Handbook 5.0[53] (pages 29-30).

38.1.1 Serial Port Settings

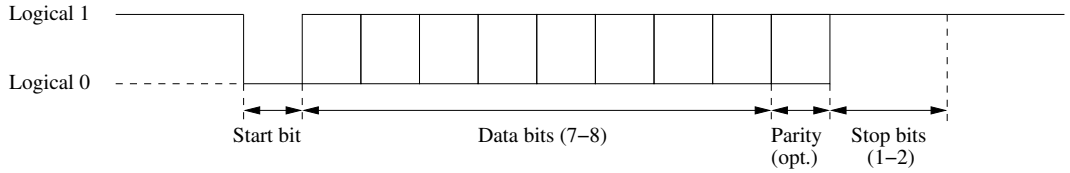
The serial port settings include the following parameters:

[1]Only applicable to ”combo” RS-232/422/485 ports. ”Regular” serial ports only supporting RS-232, as well as ”combo” RS-422/485 ports, are always enabled.
[2]Only applicable for ports with RS-422/485 support, either the ”combo” RS-232/422/485 or the ”combo” RS-422/485
[3]Hardware flow-control is only applicable for ports running in RS-232 mode.

- *Speed:* Set serial port data rate (bits/s).
  - Possible data rates for RS-232 (and RS-422/485) are: 50, 75, 110, 134, 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bits/s
  - Additional data rates for RS-422/485 : 230400, 315000, 460800, 500000, 576000, 615000, 921600, 1000000, 1152000, 1500000, 2000000 bits/s

  Default: **115200 bits/s**

- *Data character:*



  - *Data bits:* Number of data bits per character. Possible values are 7-8 bits. Default: **8 data bits**
  - *Parity:* Parity error detection setting. Possible settings are *none* (no parity checking), *even* and *odd* parity checking. When configured to use even (or odd) parity, an additional bit (the parity bit) is transmitted after the data bits to enforce that an even (or odd, respectively) number of 1's are sent, thereby enabling the receiver to detect single bit errors. Default: **No parity**
  - *Stop bits:* Number of stop bits. Possible values are 1 and 2 bits. The stop bits define the interval until the next character can be transmitted, and are sent as logical 1 (compare with the *start bit*, which is sent as a logical 0). Default: **1 stop bit**

- *Flow control*
  - *Hardware flow control:* (RS-232 only) Hardware flow control using RTS/CTS. Explained further in section 38.1.2. Default: **Disabled**
  - *Software flow control:* Software flow control using XON/XOFF. Explained further in section 38.1.3. Default: **Disabled**

38.1.2 Hardware flow control using RTS/CTS

RS-232 serial ports can use the request to send (RTS) and clear to send (CTS) pins to enforce flow control over the serial line. The DTE will assert the RTS to indicate to the DCE that it has data to send, and the DCE will respond by asserting the CTS when it is ready to receive data.

Similarly, the DCE asserts the CTS when it has data to send, and the DTE will respond by asserting RTS to give the DCE permission to send. The extension to allow the flow-control to work both ways is referred to as *RTS/CTS handshaking* and was not included in the original RS-232 standard.

Serial ports on WeOS devices are typically RS-232 ports using RJ-45 sockets (EIA/TIA-561) in DCE mode, as shown in fig. 38.1 (for a definite description of the serial port on your Westermo device, see the associated product User Guide).

Signal	Acronym	Dir (DCE)	Nb
Request To Send	RTS	In	8
Clear To Send	CTS	Out	7
Transmitted Data	TD	In	6
Received Data	RD	Out	5
Signal Ground	SG		4
Data Terminal Ready	DTR	In	3
Data Carrier Detect	DCD	Out	2
Data Set Ready	DSR	Out	1

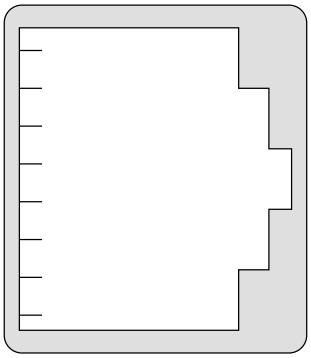


Figure 38.1: Typical RS-232 serial port on WeOS devices – RJ-45 socket (EIA/TIA-561) in DCE mode.

38.1.3 Software flow control using XON/XOFF

An alternative to hardware flow control is to use software flow control, which does not require the presence of the RTS and CTS pins. With software flow control (XON/XOFF) the receiver can stop the sender by transmitting a special character (XOFF, ASCII 19) over the data line. Once the receiver is ready to receive more data it transmits an XON character (ASCII 17).

38.2 Managing serial ports via the web interface

The Web interface provides configuration of serial ports.

38.2.1 Serial ports overview

Menu path: Configuration ⇒ Serial ⇒ Port

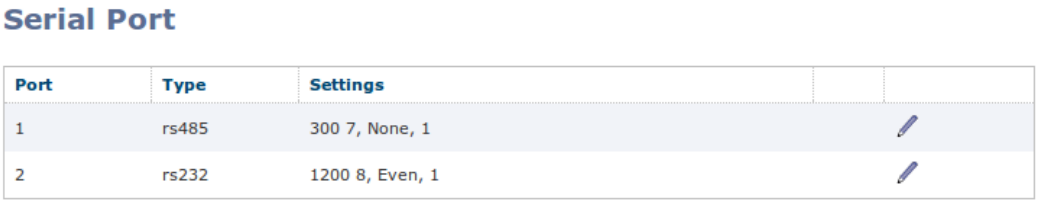
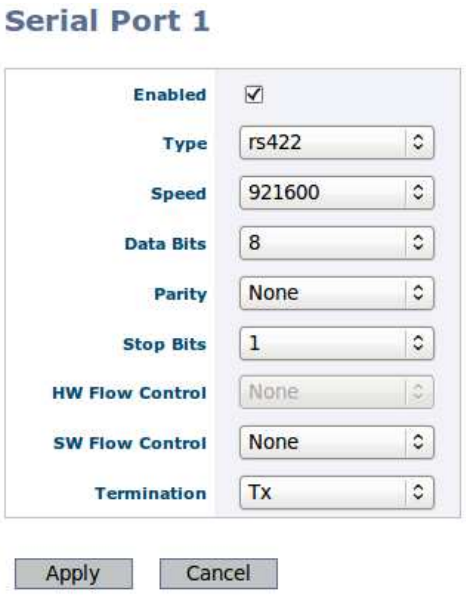


Figure 38.2: Serial port configuration settings overview

38.2.2 Edit Serial Port Settings

Menu path: Configuration ⇒ Serial ⇒ Port⇒ 

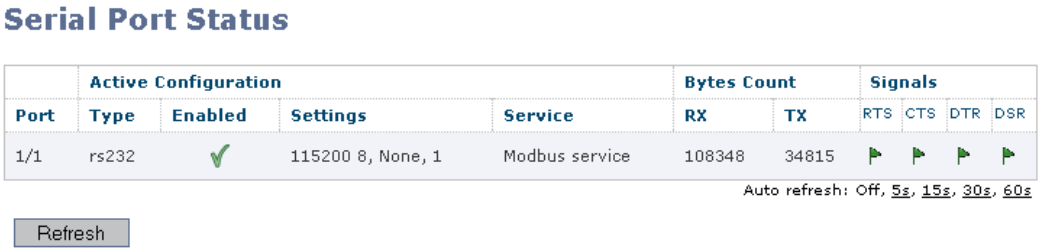


On this page you can change the settings for the serial port.

Type	Enable/disable the serial port by checking/unchecking this checkbox.
Type	Select serial interface type. Selections only available if multiple interface types are supported by hardware.
Speed	Set serial port data rate.
Data bits	Set the number of data bits
Parity	Set parity error detection
Stop bits	Set the number of stop bits
HW flow control	Enable/disable hardware flow control using RTS/CTS
SW flow control	Enable/disable software flow control using XON/XOFF
Termination	Select serial interface termination for RS422 and RS485.

38.2.3 Serial Port Status

Menu path: Status ⇒ Serial ⇒ Port



Port	The port label
Type	Type of port, rs-232, rs-422 or rs-485
Enabled	Shows if the port is enabled
Settings	Show active port settings
Service	Shows what service controls the port.
Byte count	Displays the total byte count, in (RX) and out (TX)
Signals	Displays the status signals, in (CTS, DSR) and out (RTS, DTR)
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click Off to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.

38.3 Managing serial ports via the CLI interface

The table below shows serial port management features available via the CLI.

Command	Default	Section
Configure common serial port settings		
port [serial|. . .] <PORTLIST>		Section 38.3.1
[no] speed <50-2000000>	115200	Section 38.3.2
[no] databits <7-8>	8	Section 38.3.3
[no] parity	Disabled	Section 38.3.4
[no] stopbits	1	Section 38.3.5
[no] xonxoff	Disabled	Section 38.3.6
Configure settings specific to RS-232		
[no] rtscts	Disabled	Section 38.3.7
Configure settings specific to RS-232/422/485 and RS-422/485 ports combo ports		
[no] type <rs232|rs422|rs485>	*Differs*[1]	Section 38.3.8
[no] terminate [rx] [tx]	Disabled	Section 38.3.9
Configure settings specific to RS-232/422/485 combo ports		
[no] enable	Disabled	Section 38.3.10
Show serial port status		
show port serial [PORTLIST]		Section 38.3.11

[1] Default for RS-232/422/485 ”combo” ports is **”rs232”**, while default for RS-422/485 ”combo” ports is **”rs485”**.

38.3.1 Managing serial port settings

Syntax `port [serial|...] <PORTLIST>`

Context Global Configuration context

Usage Enter the Serial Port Configuration context for the given port.

A **"PORTLIST"** is a comma separated list of ranges of serial ports without intermediate spaces, e.g., **"1/1,1/2"** on a *slotted* product, or **"1-3,5"** on a *non-slotted* product.

The port qualifier keyword **"serial"** is not needed if the numbers in the **"PORTLIST"** are unique to serial ports.

Use **"show port serial <PORTID>"** to show all configuration settings for a given serial port (also available as **"show"** command within the Serial Port Configuration context).

For a more general description of the **"port"** command, see section 8.3.1.

Default value Not applicable.

38.3.2 Setting port speed

Syntax `[no] speed <300-2000000>`

Context Serial Port Configuration context

Usage Set serial port data rate.

- Possible data rates for RS-232 (and RS-422/485) are: 50, 75, 110, 134, 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bits/s
- Additional data rates for RS-422/485 : 230400, 315000, 460800, 500000, 576000, 615000, 921600, 1000000, 1152000, 1500000, 2000000 bits/s

Use **"no speed"** to reset the serial port data rate to the default setting.

Use **"show speed"** to show the serial port speed setting.

Default value 115200 (bits/s)

38.3.3 Setting number of data bits

Syntax `[no] databits <7-8>`

Context Serial Port Configuration context

Usage Set the number number of data bits.

Use **"no databits"** to reset the number of data bits to the default setting.

Use **"show databits"** to show the configured number of databits for this serial port.

Default value 8

38.3.4 Setting parity error detection

Syntax `[no] parity <odd|even>`

Context Serial Port Configuration context

Usage Set parity error detection. Use command **"parity odd"** to specify *odd* parity, or **"parity even"** to specify *even* parity on this serial port.

Use **"no parity"** to disable parity checking on this port.

Use **"show parity"** to show the parity checking setting for this port: *None* (i.e., Disabled), *Odd*, or *Even*.

Default value Disabled (`no parity`).

38.3.5 Setting number of stop bits

Syntax `[no] stopbits <1|2>`

Context Serial Port Configuration context

Usage Set the number number of stop bits (1 or 2).

Use **"no stopbits"** reset the number of stop bits to the default setting.

Use **"show stopbits"** to show the configured number of stopbits for this serial port.

Default value 1

38.3.6 Setting Software flow control (XON/XOFF)

Syntax `[no] xonxoff`

Context Serial Port Configuration context

Usage Enable/disable software flow control using XON/XOFF

Use **"show xonxoff"** to show the software flow control setting (XON/XOFF) for this serial port.

Default value Disabled (`no xonxoff`)

38.3.7 Setting Hardware flow control (RTS/CTS)

Syntax `[no] rtscts`

Context Serial Port Configuration context (only applicable when the serial port is operating in RS-232 mode)

Usage Enable/disable hardware flow control using RTS/CTS.

Use **"show rtscts"** to show the hardware flow control setting (RTS/CTS) for this serial port.

Default value Disabled (`no rtscts`)

38.3.8 Selecting serial protocol for a serial (combo) port

Syntax `[no] type <rs232|rs422|rs485>`

Context Serial Port Configuration context (Only applicable to "combo" *RS-232/422/485* and *RS-422/485* serial ports.)

Usage Select serial protocol to use for a serial (combo) port. Use, e.g., **"type rs422"** to make a port operate in RS-422 mode. **"no type"** resets to the default type (RS-232 for RS-232/422/485 ports and RS-485 for RS-422/485 ports).

Use **"show type"** to show whether the serial (combo) port is configured in RS-232, RS-422, or RS-485 mode. (This is only of major interest for combo *RS-232/422/485* and *RS-422/485* serial ports - a "regular" RS-232" serial port cannot be set to anything but RS-232.)

Default value *RS-232* for ”combo” RS-232/422/485 ports and *RS-485* for ”combo” RS-422/485 ports.

38.3.9 Configure serial port termination

Syntax `[no] terminate [rx] [tx]`

Context Serial Port Configuration context (*RS-422 and RS-485* modes only)

Usage Configure serial port termination setting. The termination setting is only applicable for ports operating in RS-422 or RS-485 mode.

- *RS-422 (4-wire):* Use **”terminate rx”** to enable termination on the receive pair, **”terminate tx”** to enable termination on the transmit pair, and **”terminate rx,tx”** to enable termination on both the receive and transmit pairs. **”no terminate”** will disable termination.
- *RS-485 (2-wire):* Use **”terminate”** to enable termination and **”no terminate”** to disable termination.

Use **”show ”** to shether termination is enabled or disabled. In RS-422 mode, the port pairs, for which termination is enabled, are listed (RX, TX, or both RX and TX).

Default value Disabled (`no terminate`)

38.3.10 Enabling/disabling a serial (RS-232/422/485 ”combo”) port

Syntax `[no] enable`

Context Serial Port Configuration context

Usage Enable/disable a serial port. (Only applicable to ”combo” *RS-232/422/485* serial ports. Regular RS-232 ports and ”combo” *RS-422/485* serial ports are always enabled.)

Use **”show enable”** to show whether the ”combo” RS-232/422/485 serial port is enabled or disabled.

Default value Disabled (`no enable`)

38.3.11 Show Serial Port Status

Syntax `show port serial [PORTLIST]`

Context Admin Exec context

Usage Show status of one or all serial ports. Use **"show port serial"** to list status *summary* of all serial ports on the unit. Use **"show port serial PORLIST"** (e.g., *show port serial 1* to list *detailed* status information on a specific serial port (or list of ports).

Default value If no PORTID is given, a status *summary* of all serial ports is listed.

Chapter 39

Serial Over IP

This chapter describes the *Serial Over IP* application available on WeOS products equipped with a serial port. *Serial over IP* enables you to:

- extend an existing serial communication channel over an intermediate IP network.
- create a virtual serial port for remote access from a PC.
- replace an analog modem with a WeOS unit (AT command mode).

For information on serial port configuration (data rate, data bits, etc.), see chapter 38.

39.1 Overview of Serial Over IP

An overview of Serial Over IP features in WeOS is presented in the table below.

Feature	Web	CLI	General Description
Mode (server, client, peer, or AT command)	X	X	Sections 39.1.1-39.1.2
Protocol Extensions	X	X	Sections 39.1.1.3 and 39.1.2
Packing of Data	X	X	Sections 39.1.2 and 39.1.3
Frame separator	X	X	Sections 39.1.2 and 39.1.3
Frame size	X	X	Sections 39.1.2 and 39.1.3
Frame delay	X	X	Sections 39.1.2 and 39.1.3

Continued on next page

Continued from previous page

Feature	Web	CLI	General Description
Select Serial Port	X	X	Section 39.1.2
Addressing/Port Settings	X	X	-”-
Receiving (incl. multicast)	X	X	-”-
Sending (incl. multicast)	X	X	-”-
Modem Replacement / AT command interpreter	X	X	Sections 39.1.1.4 and 39.1.2, Sections 39.1.4 and 39.1.5

39.1.1 Serial Over IP introduction

The *Serial Over IP* application can be used in several ways, but the use cases can be divided into four typical applications:

- Serial point-to-point
- Serial one-to-many (typically a Master-slaves application)
- PC access to remote serial devices.
- Modem replacement

39.1.1.1 Point-to-point

In this way two serial devices can communicate over an IP network. It can be set up either as a client-server configuration using TCP, and as two peers using UDP.

Figure 39.1: Serial Point-to-point link

39.1.1.2 One-to-many

This allows one serial device (typically a master) to communicate with multiple serial devices using UDP transport. It can be set up as IP broadcast, IP multicast, or via multiple IP unicast streams.

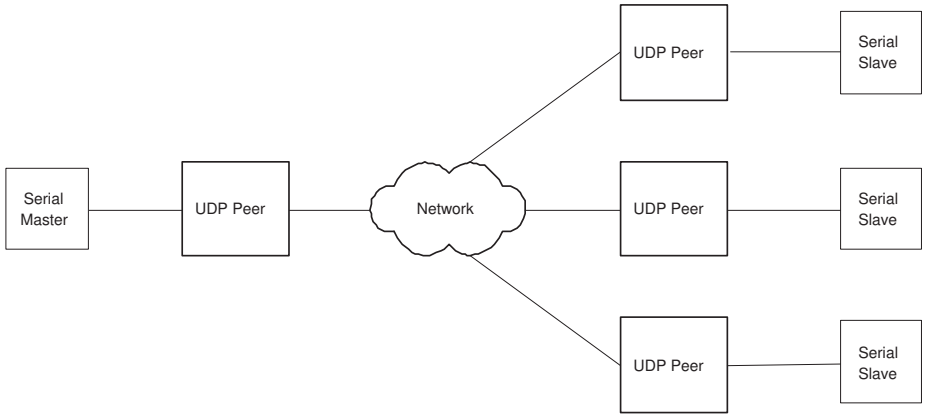


Figure 39.2: Serial one to many

39.1.1.3 Serial Port Redirector (Virtual Serial Port)

By using a serial port redirector software, an application can access remote serial devices as if the where directly connected to the PC. Westermo provides a OEM version Serial/IP®[1] that allows up to 10 virtual serial port to be created. Note: the OEM version of Serial/IP® requires that telnet protocol extension is enabled to verify the license. There is also a possibility for an application to directly connect to the Serial Over IP.

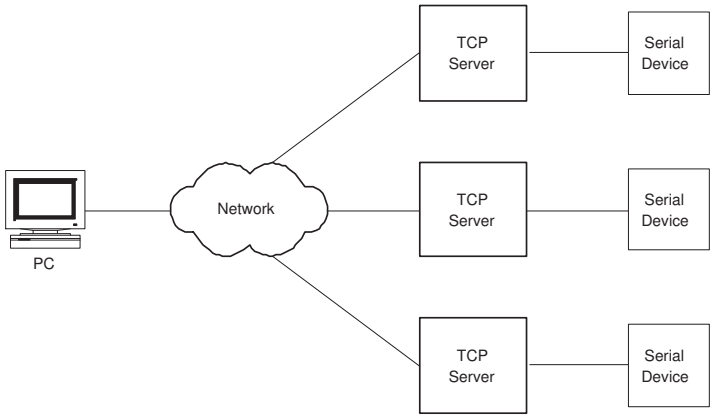
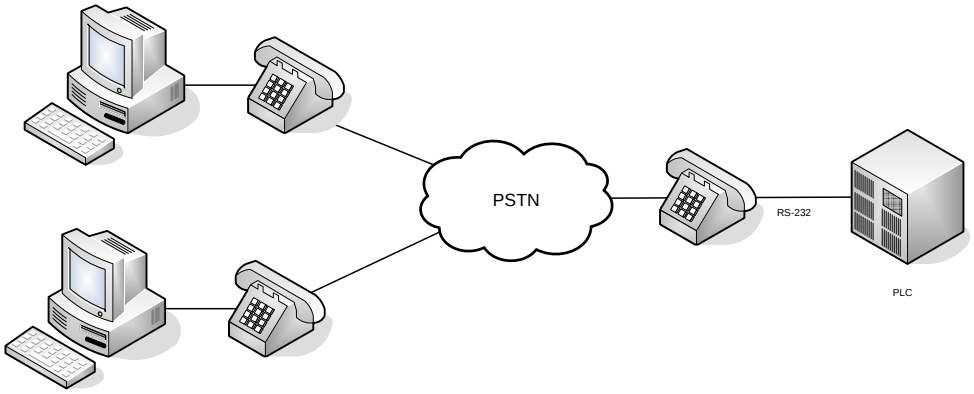


Figure 39.3: Using a serial port redirector software

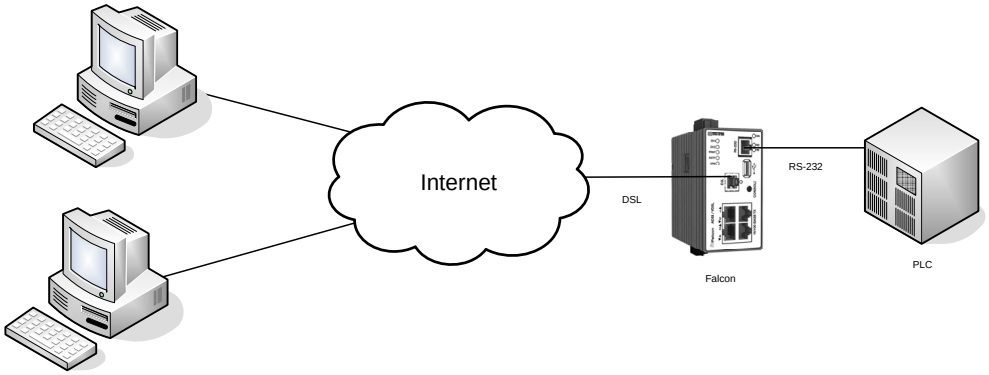
[1] Serial/IP is a registered trademark of Tactical Software LLC.

39.1.1.4 Modem replacement / AT command interpreter

The AT Command mode in Serial Over IP is designed to be used with old legacy serial equipments that historically used PSTN or leased line modems.



Serial Over IP allows serial equipments to use Internet instead.



To enable an easy transitions from old modems to new DSL modems, the Serial Over IP can be controlled with AT commands.

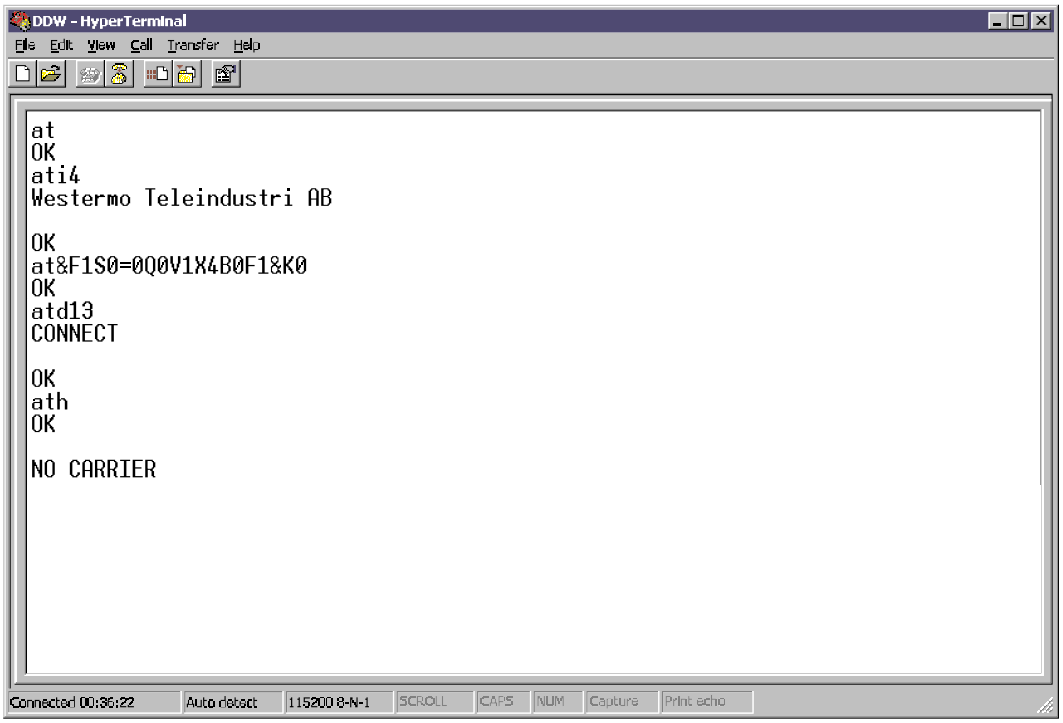


Figure 39.4: Terminal connected to a DDW-226

39.1.2 Serial Over IP settings

- *General settings:*
  - *Mode:* Set operating mode.
    - *Server:* This is the default setting. In *server* mode, the unit will act as a TCP server, and listen for incoming call establishments. Only a single client can connect to the serial port at time. This mode can be used both in *point-to-point* serial extension (section 39.1.1.1) and *serial port redirector* (section 39.1.1.3) applications.
    - *Client:* In *client* mode, the unit will act as a TCP client, and initiate a connection to a remote TCP server. This mode can be used in the *point-to-point* serial extension application, see section 39.1.1.1.
    - *Peer:* In *peer* mode, UDP will be used for serial data transportation. This mode can be used both in the *point-to-point* (section 39.1.1.1)

and *one-to-many* (section 39.1.1.2) serial extension applications.

In the point-to-point case both peers will specify the IP address of the remote peer as the *destination*. For the *one-to-many* case there are many addressing options, see the item on *Addressing information* below.

* *AT Command:* In *AT Command* mode the connection is controlled by AT commands. It operates both as a TCP server and a TCP client. The connection is establish by entering ATD<NUMBER>. The number must be mapped to an IP address. See *Modem replacement* (section 39.1.1.4) for details.

Default: **Server**

- *Protocol Extensions:* Enable protocol extensions, e.g. RFC2217 Telnet extensions[5]. Needed to verify OEM licence of Serial/IP®[2]. As of WeOS v4.17.1, there is no other use of this setting, e.g., there is no support for configuring serial port settings (datarate, number of databits, etc.) via telnet extensions. Such support is planned, but not yet implemented.
- *Serial Port:* Select which serial port to use. Default: **Disabled**

- *Data Packing Settings*: (see section 39.1.3 for further explanation)
  - *Frame separator character:* Define frame separator character, if any. Any 8-bit ASCII character, 0-255, can be used. Default: **Disabled**
  - *Maximum Frame Size:* Define maximum frame size in number of bytes. Allowed values are in range 1-1460 (bytes). Default: **1000 (bytes)**
  - *Maximum Frame Delay:* Define maximum frame delay in milliseconds. Allowed values are in range 1-2550 (ms). Default: **20 (milliseconds)**
- *Addressing information:*
  - *Listen:* Define the local interface to accept incoming Serial Over IP traffic, and the (UDP/TCP) port to listen to. This setting is only applicable in *Server*, *Peer* and *AT Command* modes.

    The default is to listen to UDP/TCP port 9000, and to accept traffic on any interface. The *Listen* setting can be used to *limit* incoming Serial Over IP traffic to a specific interface[3], and/or to define a specific

[2] Serial/IP is a registered trademark of Tactical Software LLC.
[3] For more fine-grain control to limit Serial Over IP traffic, see section 31.1.2 (Packet Filtering) in the Firewall chapter.

(UDP/TCP) port to listen to.

In *Server* and in AT Command modes, the default is to listen to TCP port 9000, and to accept Serial Over IP traffic on any interface. In *Server* mode, an *additional* listening port may be set to allow support for e.g. failover. See setting *Secondary Listen Port* below.

In *Peer* mode, the default is to listen to UDP port 9000, and to accept serial over IP traffic on any interface. IP Unicast, broadcast[4], and multicast (for defined multicast groups, see below) are accepted in *Peer* mode.

Default: **Interface: "Any"** (incoming traffic accepted via any interface), and **(UDP/TCP) port: 9000**

- *Secondary Listen Port:* Additional local TCP port to listen to. This setting is only applicable in (TCP) *Server* mode.

  For more information, see setting *Secondary mode* below.

  Default: **Disabled**

- *Secondary Mode:* If an additional local TCP port is set up (see *Secondary Listen Port* above), the concurrent access to the serial port (collision handling) is controlled by altering this option.

  Available options are:

  * *Slotted:* Where the access to the serial port is interleaved between the TCP-sessions. When data is received from a TCP-session, the serial port is reserved for the TCP-session for a timeslot with a length of *timeout* ms. Data received during this time period is sent to the TCP-session with the reservation only. If data is received from the secondary TCP-session, the data will be read when the time period has ended.

    If data is received from the serial port after the timeout and no new data is available from the TCP-sessions, this data will be forwarded to the last TCP-session that was active.

    If data is available from both TCP-sessions at exactly the same time, data is first received from the primary session, then the secondary.

[4] Both IP subnet broadcast packets (e.g., 192.168.1.255 on a 192.168.1.0/24 network), and data link IP broadcast (255.255.255.255) are accepted if received on the appropriate interface.

* *Failover:* Where the primary TCP-session has exclusive access to the serial port until disconnected. Secondary TCP-session data will be handled when the primary session has disconnected, i.e., act as *backup*. When the primary TCP-session comes up again, it will get back the exclusive access to the serial port once data is received on the primary TCP-session.

  When the secondary TCP-session is in backup state, incoming data on the secondary-TCP session is ignored.

* *Raw:* In this case, data from both TCP-sessions are directly forwarded onto the serial port without any handling. Data received from the serial port is returned to both TCP-sessions. **Note:** This means that the applications connecting to the TCP ports have to take full responsibility for the concurrent access to the serial port.

Default: **Slotted, timeout 600 ms**

Configuring a secondary (TCP server) listen port in WeOS, corresponds to the *Dual TCP* feature found in Westermo EDW-100 units. The table below translates between WeOS and EDW-100 settings.

WeOS		**EDW-100**		
Secondary Mode	**Timeout**	**Dual TCP**	**Dual TCP Priority**	**Response Timeout**
Slotted	50-65535 (ms)	Enabled	Disabled	50-65535 (ms)
Raw	N/A	Enabled	Disabled	0 (ms)
Failover	N/A	Enabled	Enabled	Empty

– *Multicast group:* Multicast group to *receive* data from. This is only applicable in *Peer* (UDP) mode. IP multicast addresses are in the following range: 224.0.0.0-239.255.255.255.

  When configured, the unit will accept packets to the stated multicast address, when received on the interface and (UDP) port declared in the *Listen* setting. Note: the unit will still accept unicast and broadcast packets as described in the *Listen* item above.

  Default: **Disabled**

– *Destination/peer:* IP address and (UDP/TCP) port numbers to relay data to/from. This setting is only applicable in *Client* and *Peer* modes. In Client mode, the destination address should be the IP address of the

(remote) Server.

In Peer mode, it is possible to specify one or more destinations/peers (maximum 32), and the address can be IP unicast, broadcast[5], or multicast [6].

Default: **Disabled**

- *Dynamic peer:* Peer mode only. When dynamic peer[7] is enabled, data will be sent to the source of the latest incoming data. (Until there is incoming data, the unit will send data to the (first) configured destination/peer.)

  Default: **Disabled**

- *DTR Control:* The DTR (Data Terminal Ready) Control setting is only applicable for RS-232 serial ports, and when Serial Over IP is configured in TCP Server or TCP Client mode. With *DTR Control* enabled, the activation of the TCP session is controlled by the status of the DTR control line:

  - *TCP Server:* With Serial Over IP in TCP server mode, the server will await activation of DTR before it accepts remote TCP clients to establish a connection.

  - *TCP Client:* With Serial Over IP in TCP client mode, the client will not make any TCP connection attempts as long as the DTR is inactive.

  - *AT Command:* With Serial Over IP in AT Command mode, an active connection will be closed if the DTR signal is dropped. When establishing connections the DTR signal is ignored.

  With DTR Control enabled, both TCP clients and servers will close an established connection if the DTR becomes inactive. Furthermore, with DTR Control enabled, data arriving on the serial port will be dropped while DTR is inactive.

  Default: **Disabled**

[5] Sending to the data link IP broadcast (255.255.255.255) will only work if the unit has a default gateway configured (see section 19.5.2). IP subnet broadcast (e.g., 192.168.1.255) is preferred.

[6] Sending data to a multicast address will only work if the unit has a default gateway configured (see section 19.5.2).

[7] The ”dynamic peer” setting is referred to as ”latest calling” in Westermo EDW-100.

39.1.3 Packing Algorithm

When data arrives at the serial port of the WeOS unit, one of the following criteria must be fulfilled before the serial data is encapsulated into a UDP/TCP packet and sent over the network.

- *Frame separator character detected:* A frame separator character can be defined. The serial data buffered will be sent over the network when this character is detected,e.g., ”13” for Carriage return). Any 8-bit ASCII character, 0-255, can be used.
- *Maximum Frame Size Reached:* A maximum frame size must be defined. This is the maximum number of serial data bytes that will be carried in each UDP/TCP frame. When the maximum number of bytes is buffered, the packet will be transmitted over the network. Allowed values are in range 1-1460 (bytes). Values above 255 are approximate.
- *Maximum Frame Delay Reached:* A maximum frame delay can be defined. This is the time, after the last received character in the buffer, the WeOS unit will wait until the buffered serial data is sent over the network. Allowed values are in range 1-2550 ms; If *maximum frame delay* is used with low data rates (see section 38.1.1), it should be set to at least one ”character time”.

39.1.4 AT Commands

The AT command interpreter has a limited AT command set. Table 39.1 lists the supported AT commands and their function.

In addition to the AT commands listed in table 39.1, WeOS units accept the set AT commands listed in table 39.2, however, they only respond ”OK”. Any combination of these commands and ’+’, ’?’, ’&’,’%’, or ’\’ are valid.

If a specific answer is required *user strings* can be defined, see section 39.1.5.

39.1.5 AT Command profile settings

- *Control settings:* The serial command prompt can be set to not echo back characters (ATE overrides this). The result codes can be set to either numeric verbose (default) mode (ATV overrides this). These settings are stored

ATD<phone number>	Dials a "phone number"
ATH	Hang-up a connection
ATA	Answer a call
ATO	Go on-line, from on-line command mode
ATE<0|1>	Set Echo on/off
ATQ<0|1>	Set Quiet mode on/off
ATV<0|1>	Set Verbose mode on/off
ATI<x>	Show identification
+++	The escape sequence.
A/	Repeat the last entered command

Table 39.1: Supported AT commands.

ATB	Always responds OK
ATC	”
ATF	”
ATG	”
ATH	”
ATJ	”
ATK	”
ATL	”
ATM	”
ATN	”
ATP	”
ATR	”
ATS	”
ATT	”
ATU	”
ATX	”
ATY	”
ATZ	”

Table 39.2: Set of AT commands where WeOS units respond ”OK”.

in the configuration database in contrast to ATE and ATV command which are not stored. The auto-answer function can be set to off (0) or numbers of RINGS (default 1) before answering the call.

- *Sync connect settings:* If sync connect is enabled the local and remote side

will synchronize the connect sequence, and regularly verify the connection. If an established connection can't be verified the connection will be taken down after a time specified by the timeout setting.

- *Message settings:* The OK, CONNECT, DISCONNECT and ERROR message can be customised if desired. Both verbose and numeric value can be changed.
- *Map settings:* To be able to dial, the phone number must be mapped to an IP-address and port. ATD is equivalent with ATD0, mapping number 0 to an IP-address allows for connecting with ATD without number. Up to 250 map entries can be configured.
- *User strings:* If the serial equipment requires a specific answer to an initialisation string, user answer strings can be defined. The will not perform any function except printing out the desired text. The answer can be with line breaks, use backslash ('\') as line break. The answer can be up to 64 characters long including line breaks.

39.2 Managing Serial Over IP via the web interface

The Web interface provides configuration of the Serial Over IP.

39.2.1 Serial Over IP overview

Menu path: Configuration ⇒ Serial ⇒ Serial Over IP

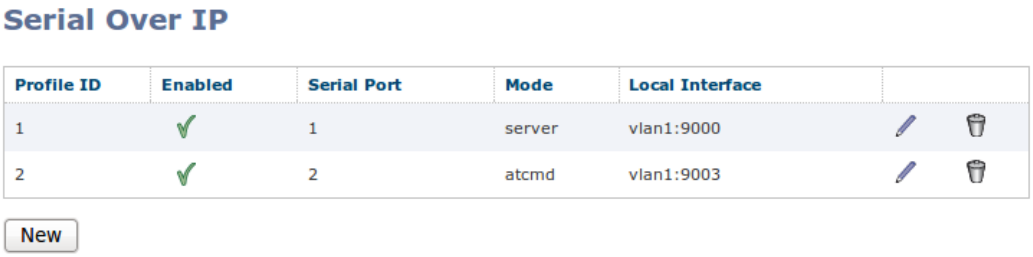


Figure 39.5: Serial Over IP configuration settings overview

Profile ID	A Unique identifier for the Serial Over IP instance. Automatically generated,
Enabled	If disabled, the instance will not be started (i.e the instance will not listen for data nor send data). A green check-mark means the instance is enabled, and a dash means it is disabled.
Serial Port	The serial port to which the Serial Over IP instance is connected.
Mode	The mode (**`Server (TCP)`**, **`Client (TCP)`**, **`Peer (UDP)`**, or **`AT Command`** for this Serial Over IP instance.
Local Interface	The interface to accept incoming Serial Over IP traffic, and the TCP/UDP port to listen to. Only applicable in **`Server`**, **`Peer`**, and **`AT Command`** modes.
Edit	Click this icon to edit a Serial Over IP instance.
Delete	Click this icon to remove a Serial Over IP instance.
New	Click this button to create a new Serial Over IP instance. You will be presented to a form where you can configure the new instance. One instance can be created for each serial port found on the device.

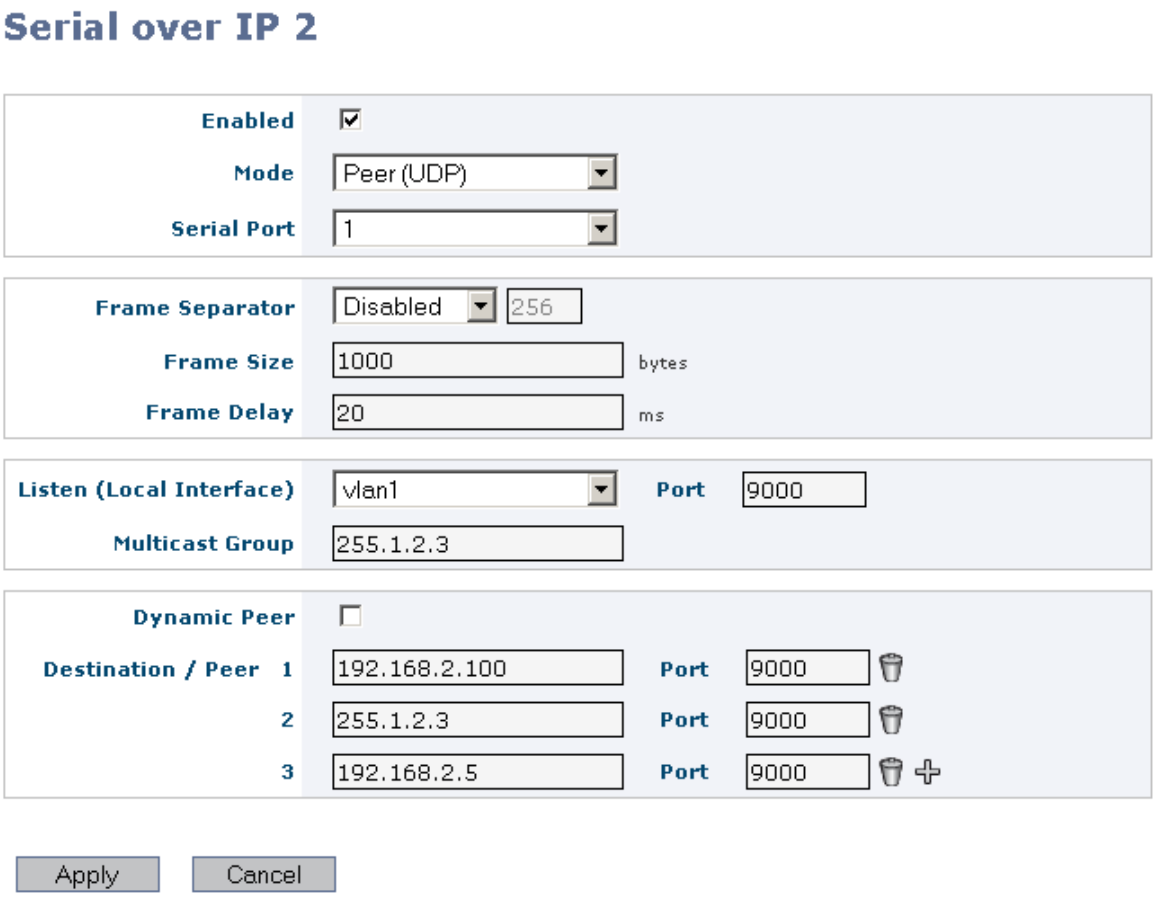
39.2.2 New Serial Over IP Instance

Menu path: Configuration ⇒ Serial ⇒ Serial Over IP⇒ **New**

When clicking the **New** button, the edit page will be displayed. For field descriptions, see section 39.2.3 below.

39.2.3 Edit Serial Over IP Settings

Menu path: Configuration ⇒ Serial ⇒ Serial Over IP⇒ 



On this page you can change the settings for Serial Over IP.

If parameter is applicable in a certain mode is denoted with a character according to:

- S - TCP-Server
- C - TCP-Client
- P - Peer (UDP)
- A - AT Command

Enabled	All	Check the box to enable the Serial Over IP instance, uncheck to disable. If disabled, the instance will not be started.
Mode	All	Set operating mode: Server (TCP), Client (TCP), Peer (UDP), or AT Command.
Protocol Extensions	S	Disable or chose appropriate protocol extension.
DTR Control	S,C,A	Enable DTR control on the connection.
Serial Port	All	Serial port to use.
AT Command Profile	A	Select a command profile for this AT command instance. See Section 39.1.5 for more information.
Frame separator	All	Enable/Disable and define frame separator character if enabled.
Frame size	All	Define maximum frame size in characters.
Frame delay	All	Define maximum frame delay in milliseconds.
Listen (local end)	S, P	The interface to accept incoming Serial Over IP traffic, and the TCP/UDP port to listen to. Default Interface: Any. Default port: 9000
Secondary Listen Port	S	Additional local TCP port to listen to. Will listen on the same interface as set in *Listen* option.Default: Disabled (empty)
Secondary Mode	S	How to handle concurrent access to serial port. See Section 39.1.2 for available modes.Default: Slotted
Secondary Mode Timeout	S	Timeout for *Secondary mode* slotted.Default: 600 ms
Multicast group	P	Multicast address to listen to.
Dynamic peer	P	Enable/disable dynamic peer[1].

Continued on next page

		Continued from previous page
Destination/peer	C, P	IP address and (UDP/TCP) port for remote peer(s)/ destinations. In Peer Mode, several destination/peer entries can be configured, and the destination address can be unicast, broadcast or multicast. Default port: 9000

39.2.4 AT Command Profiles overview

Menu path: Configuration ⇒ Serial ⇒ Serial Over IP ⇒ AT Command

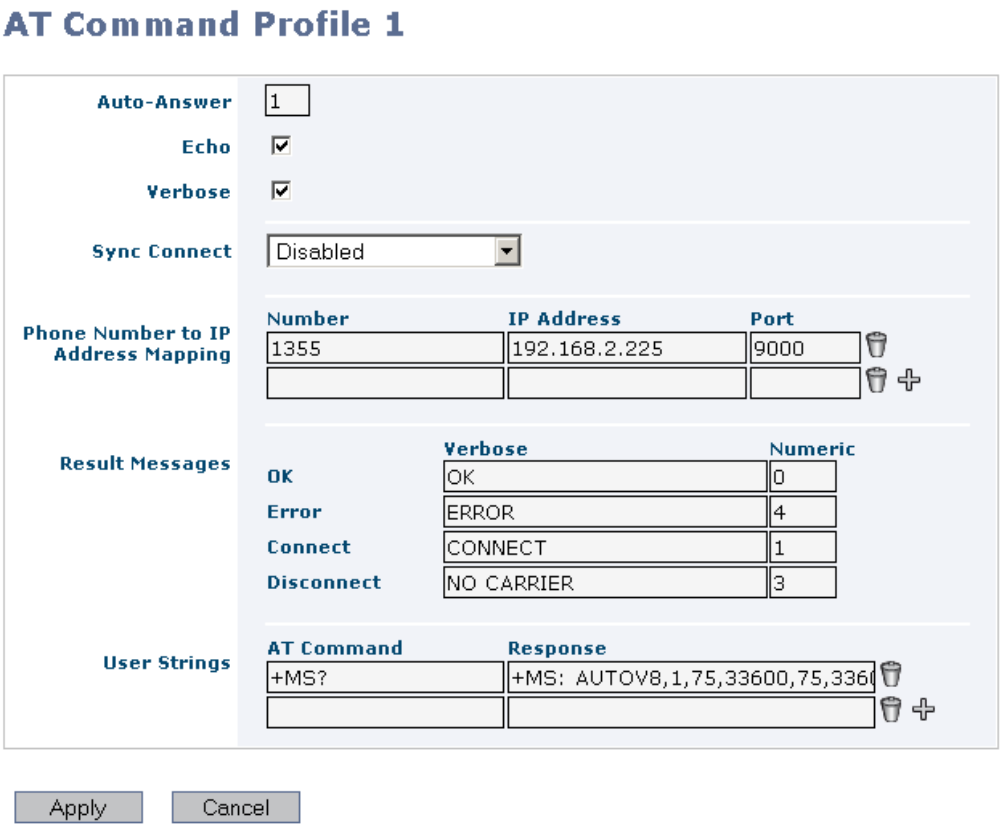
Click on the Edit icon ( ) to edit the settings of a specific profile.

[1]The ”dynamic peer” setting is referred to as ”latest calling” in Westermo EDW-100.

39.2.5 Edit AT Command Profile

Menu path: Configuration ⇒ Serial ⇒ Serial Over IP ⇒ AT Command⇒ 

On this page you can change the settings for an existing profile or when creating a new profile.

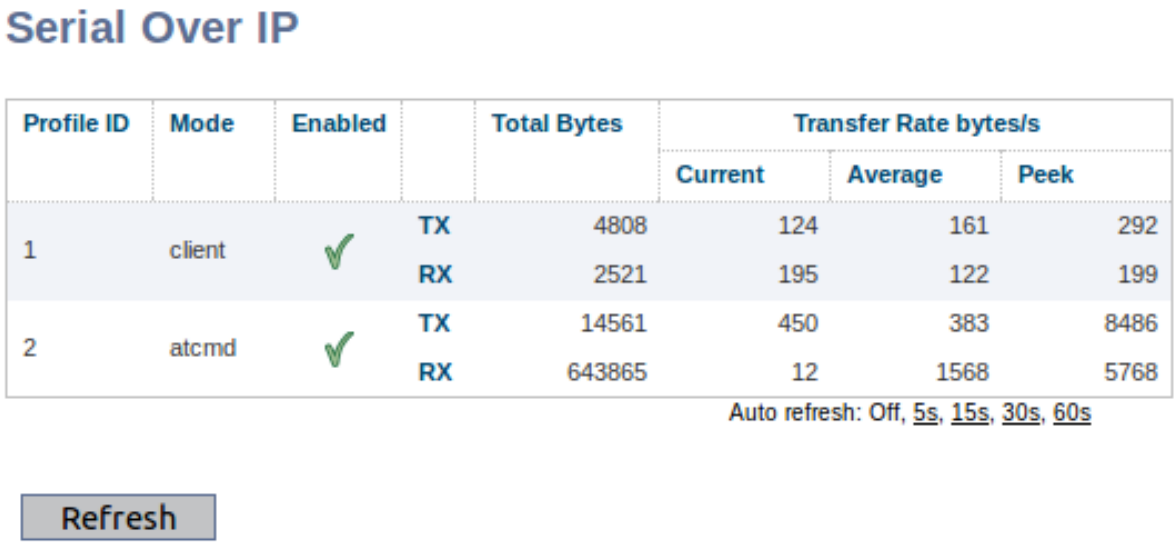


Auto-Answer	Set auto-answer, answer after number of rings. Set this to 0 for no auto-answer. Default: 1 .
Echo	Enables/disables echo (ATE overrides this). Default: enabled
Verbose	Enables/disables verbose result messages (ATV overrides this). Default: enabled
Sync Connect	Enables/disables synchronized connections. Default: disabled
Sync Connect Timeout	Configures the timeout for synchronized connections. Default: 60s
Mapping	Maps phone number to IP-addresses and ports.
OK-Message	Configure the OK message, verbose text and numeric value. Default-verbose: OK, Default-numeric: 0
Error-Message	Configure the Error message, verbose text and numeric value. Default-verbose: ERROR, default-numeric: 4
Connect-Message	Configure the Connect message, verbose text and numeric value. Default-verbose: CONNECT, Default-numeric: 1
Disconnect-Message	Configure the Disconnect message, verbose text and numeric value. Default-verbose: NO CARRIER, Default-numeric: 3
User strings	Configures user strings and answer. The answer can be up to 64 characters long including line breaks.

39.2.6 Serial Over IP Status

Menu path: Status ⇒ Serial ⇒ Serial Over IP

On this page you can see status for the Serial over IP profiles.



Profile ID	The Serial Over IP profile ID .
Enabled	Indicates if the profile is enabled. A green check-mark means the instance is enabled, and a dash means it is disabled.
Total Bytes	Total number of bytes sent/received (TX/RX) by this profile.
Transfer Rate, Current	Current transfer rate sending/receiving (TX/RX) for this profile.
Transfer Rate, Average	Average transfer rate sending/receiving (TX/RX) for this profile.
Transfer Rate, Peek	Highest transfer rate sending/receiving (TX/RX) for this profile.

39.3 Managing Serial Over IP via the CLI interface

The table below shows Serial Over IP management features available via the CLI.

Command	Default	Section
Configure Serial Over IP settings		
seroip		Section 39.3.1
Settings common to all modes		
[no] mode <server|client|peer|atcmd>	server	Section 39.3.2
[no] port <SERIAL-PORT>	Disabled	Section 39.3.3
[no] frame-separator <0-255>	Disabled	Section 39.3.4
[no] frame-delay <1-2550>	20	Section 39.3.5
[no] frame-size <1-1460>	1000	Section 39.3.6
Additional Server mode (TCP) settings		
[no] listen <IFACE[:PORT]>	Disabled	Section 39.3.7
[no] protocol <raw|telnet>	Disabled	Section 39.3.8
[no] listen-secondary port <PORT>	Disabled	Section 39.3.9
[no] secondary-mode <failover| raw|slotted <TIMEOUT>>	slotted 600	Section 39.3.10
[no] dtr-control	Disabled	Section 39.3.11
Additional Client mode (TCP) settings		
[no] peer <ADDRESS[:PORT] [,ADDRESS:PORT,...]>	Disabled	Section 39.3.12
[no] dtr-control	Disabled	Section 39.3.11
Additional Peer mode (UDP) settings		
[no] listen <IFACE[:PORT]>	Disabled	Section 39.3.7
[no] peer <ADDRESS[:PORT] [,ADDRESS:PORT,...]>	Disabled	Section 39.3.12
[no] mcast-group <ADDRESS>	Disabled	Section 39.3.13
[no] dynamic-peer	Disabled	Section 39.3.14
Additional AT Command mode settings		
[no] listen <IFACE[:PORT]>	Disabled	Section 39.3.7

Continued on next page

Continued from previous page

Command	Default	Section
[no] atcmd-set <ID>	Disabled	Section 39.3.15
Show Serial Over IP settings		
seroip		
show		Section 39.3.16
show mode		Section 39.3.17
show port		Section 39.3.18
show protocol		Section 39.3.19
show atcmd-set		Section 39.3.20
show listen		Section 39.3.21
show listen-secondary		Section 39.3.22
show secondary-mode		Section 39.3.23
show dtr-control		Section 39.3.24
show mcast-group		Section 39.3.25
show frame-separator		Section 39.3.26
show frame-delay		Section 39.3.27
show frame-size		Section 39.3.28
show peer		Section 39.3.29
show dynamic-peer		Section 39.3.30
Configure AT Command profiles		
[no] atcmd [ID]	1	Section 39.3.31
[no] auto-answer <COUNT>	1	Section 39.3.32
[no] echo	Enabled	Section 39.3.33
[no] verbose	Enabled	Section 39.3.34
[no] sync-connect	Disabled	Section 39.3.35
[no] sync-timeout <TIMEOUT>	60 s	Section 39.3.36
[no] ok-message text <MESSAGE>	OK	Section 39.3.37
code <CODE>	0	
[no] error-message text <MESSAGE>	ERROR	Section 39.3.38
code <CODE>	4	
[no] connect-message text	CONNECT	Section 39.3.39
<MESSAGE> code <CODE>	1	
[no] disconnect-message text	NO CARRIER	Section 39.3.40

Continued on next page

Continued from previous page

Command	Default	Section
<MESSAGE> code <CODE>	3	
[no] map number <PSTN-NUM> to <ADDRESS[:PORT]>	9000	Section 39.3.41
[no] user-message command <COMMAND> text <MESSAGE>		Section 39.3.42
Show AT Command profile		
atcmd [ID]	1	
show auto-answer		
show echo		
show verbose		
show sync-connect		
show sync-timeout		
show ok-message		
show error-message		
show connect-message		
show disconnect-message		
show map		
show user-message		

39.3.1 Managing Serial Over IP settings

Syntax `seroip`

Context *Global Configuration* context

Usage Enter the Serial Over IP configuration context.

Default values Not applicable.

Error messages None defined yet.

39.3.2 Setting Mode

Syntax `[no] mode <server|client|peer|atcmd>`

Context *seroip* context

Usage Set Serial Over IP mode.

Default values server

Error messages None defined yet.

39.3.3 Setting Serial Port

Syntax `[no] port <SERIAL-PORT>`

Context *seroip* context

Usage Set serial port

Default values Disabled (**”no port”**)

Error messages None defined yet.

39.3.4 Setting Frame Separator

Syntax `[no] frame-separator <0-255>`

Context *seroip* context

Usage Define frame separator character, if any. Any 8-bit ASCII character, 0-255, can be used.

Use **”no frame-separator”** to disable frame separator checking in the packing algorithm.

Default values Disabled (**”no frame-separator”**)

Error messages None defined yet.

39.3.5 Setting Frame Delay

Syntax `[no] frame-delay <1-2550>`

Context *seroip* context

Usage Define maximum frame delay in milliseconds.

Use **”no frame-delay”** to disable maximum delay checking in the packing algorithm.

Default values 20 (milliseconds)

Error messages None defined yet.

39.3.6 Setting Frame Size

Syntax `[no] frame-size <1-1460>`

Context *seroip* context

Usage Define maximum frame size in bytes (this is part of the packing algorithm).

Use **”no frame-size”** to reset the maximum frame size to the default value.

Default values 1000 (bytes)

Error messages None defined yet.

39.3.7 Setting listen interface and port

Syntax `[no] listen <IFACE[:PORT]>`

Context *seroip* context

Usage Set local interface and (UDP/TCP) port to listen to.

Specify an interface to limit incoming traffic to the stated interface, or **”any”** to accept traffic via any interface.

Specify **”PORT”** to configure a specific (TCP/UDP) port to listen to.

Use **”no listen”** to accept traffic for (TCP/UDP) port 9000 on any interface.

Default values Any interface, (UDP/TCP) port 9000 (**”no listen”**)

Error messages None defined yet.

39.3.8 Setting Protocol Extensions

Syntax `[no] protocol <raw|telnet>`

Context *seroip* context

Usage Set protocol extensions. This is only applicable in *server* mode (section 39.3.2).

When accessing the serial port with Westermo's OEM version of Serial/IP®[8] the protocol extension setting should be **"protocol telnet"**.

Use **"no protocol"** (or **"protocol raw"**) to disable protocol extensions.

Default values Disabled (**"no protocol"**)

Error messages None defined yet.

39.3.9 Setting secondary listen port

Syntax `[no] listen-secondary port <PORT>`

Context *seroip* context

Usage Additional local TCP port to listen to. Will listen on the same interface as set in *Listen* option.Only applicable in TCP-Server mode.

Default values Disabled (**"no listen-secondary"**).

Error messages None defined yet.

39.3.10 Setting secondary mode

Syntax `[no] secondary-mode <failover|raw|slotted <TIMEOUT>>`

Context *seroip* context

Usage How to handle concurrent access to serial port. See Section 39.1.2 for available modes.Only applicable in TCP-Server mode.

Default values Slotted, timeout 600 ms (**"no secondary-mode"**).

Error messages None defined yet.

[8] Serial/IP is a registered trademark of Tactical Software LLC.

39.3.11 Enable/disable DTR Control

Syntax `[no] dtr-control`

Context *seroip* context (Only applicable in TCP Client and TCP Server modes, and only for RS-232 serial ports)

Usage Use **"dtr-control"** to activate DTR control. With DTR Control enabled, the activation of the TCP session is controlled by the status of the RS-232 DTR control line:

- *TCP Server:* With Serial Over IP in TCP server mode, the server will await activation of DTR before it accepts remote TCP clients to establish a connection.
- *TCP Client:* With Serial Over IP in TCP client mode, the client will not make any TCP connection attempts as long as the DTR is inactive.

With DTR Control enabled, both TCP clients and servers will close an established connection if the DTR becomes inactive.

Use **"no dtr-control"** to disable DTR control.

Default values Disabled

Error messages None defined yet.

39.3.12 Setting peer address and port

Syntax `[no] peer <ADDRESS[:PORT][,ADDRESS:PORT,...]>`

Context *seroip* context

Usage Remote destinations/peer(s) to relay data to/from. Note, this is only used in client or peer mode. If PORT is omitted the default port 9000 will be used.

Default values Disabled (**"no peer"**)

Error messages None defined yet.

39.3.13 Setting multicast group

Syntax `[no] mcast-group <ADDRESS>`

Context *seroip* context

Usage Multicast group to listen on. Note, this is only used in peer mode.

Default values Disabled (**”no mcast-group”**)

Error messages None defined yet.

39.3.14 Enabling Dynamic-peer

Syntax `[no] dynamic-peer`

Context *seroip* context

Usage Enable/disable dynamic-peer (The ”dynamic peer” setting is referred to as ”latest calling” in Westermo EDW-100.)

Default values Disabled

Error messages None defined yet.

39.3.15 Setting AT Command Set

Syntax `[no] atcmd-set <ID>`

Context *seroip* context

Usage Select AT command set. This setting is only valid when Serial Over IP is configured in AT command mode, see section 39.3.2.

Use **”atcmd-set <ID>”** to select an AT command set. AT command sets are created via the **”atcmd <ID>”** command, see section 39.3.31. (The AT command set identifier (ID) can take values in range 1-5.)

Use **”no atcmd-set”** to remove (disable) an AT command set selection.

Default values Disabled (**”no atcmd-set”**)

Error messages None defined yet.

39.3.16 Show All Settings of a Serial Over IP

Syntax `show`

Context *seroip* context

Usage Show all configuration settings for Serial Over IP.

Default value Not applicable.

39.3.17 Show Serial Over IP Mode Setting

Syntax `show mode`

Context *seroip* context

Usage Show the Serial Over IP mode setting

Default value Not applicable.

39.3.18 Show Selected Serial Port

Syntax `show port`

Context *seroip* context

Usage Show what serial port is selected (if any) for this Serial Over IP instance.

Default value Not applicable.

39.3.19 Show Serial Over IP Protocol extensions Setting

Syntax `show protocol`

Context *seroip* context

Usage Show the serial Over IP protocol extensions setting

Default value Not applicable.

39.3.20 Show Selected AT Command Set

Syntax `show atcmd-set`

Context *seroip* context

Usage Show the ID of the selected AT command set, or ”Disabled” if no AT command set is selected.

Default value Not applicable.

39.3.21 Show Serial Over IP Listen Setting

Syntax `show listen`

Context *seroip* context

Usage Show the Serial Over IP listen setting

Default value Not applicable.

39.3.22 Show Serial Over IP Secondary Listen Setting

Syntax `show listen-secondary`

Context *seroip* context

Usage Show whether Serial Over IP is configured to listen on a secondary port or not. Only applicable in TCP-Server mode.

Default value Not applicable.

39.3.23 Show Serial Over IP Secondary Mode

Syntax `show secondary-mode`

Context *seroip* context

Usage Show the mode controlling concurrent access to serial port. Only applicable in TCP-Server mode, and if a secondary listen port has been configured (see section 39.3.9).

Default value Not applicable.

39.3.24 Show Serial Over IP DTR Control Setting

Syntax `show dtr-control`

Context *seroip* context

Usage Show whether DTR control has been enabled or not. Only applicable in TCP Client or TCP Server mode.

Default value Not applicable.

39.3.25 Show Serial Over IP Multicast group Setting

Syntax `show mcast-group`

Context *seroip* context

Usage Show the serial Over IP multicast group setting

Default value Not applicable.

39.3.26 Show Serial Over IP Frame Separator Setting

Syntax `show frame-separator`

Context *seroip* context

Usage Show the Serial Over IP frame separator setting

Default value Not applicable.

39.3.27 Show Serial Over IP Frame Delay Setting

Syntax `show frame-delay`

Context *seroip* context

Usage Show the Serial Over IP frame delay setting

Default value Not applicable.

39.3.28 Show Serial Over IP Frame Size Setting

Syntax `show frame-size`

Context *seroip* context

Usage Show the Serial Over IP frame size setting

Default value Not applicable.

39.3.29 Show Serial Over IP Peer Setting

Syntax `show peer`

Context *seroip* context

Usage Show the Serial Over IP peer setting

Default value Not applicable.

39.3.30 Show Serial Over IP Dynamic Peer Setting

Syntax `show dynamic-peer`

Context *seroip* context

Usage Show the Serial Over IP dynamic-peer setting. (The ”dynamic peer” setting is referred to as ”latest calling” in Westermo EDW-100.)

Default value Not applicable.

39.3.31 Managing AT Command Profiles

Syntax `[no] atcmd [ID]`

Context *Global Configuration* context

Usage Enter/create the AT Command profile context of the given AT Profile instance ID. If this is a new instance, the instance will be created first upon leaving the AT Command profile context with *end* or *leave*.

Use **”no atcmd <ID>”** to remove an existing AT Command profile instance, or **”no atcmd”** to remove all profiles.

Default values 1 (i.e., running **”atcmd”** will enter/create AT command set 1. However, running **”no atcmd”** will remove all configured AT command sets.)

Error messages None defined yet.

39.3.32 Auto-Answer

Syntax `[no] auto-answer <COUNT>`

Context *atcmd* context

Usage Set auto-answer, answer after number of rings. Use command **”auto-answer 0”** for no auto-answer.

Use command **”no auto-answer”** to reset to the default setting.

Default values 1

Error messages None defined yet.

39.3.33 Echo

Syntax `[no] echo`

Context *atcmd* context

Usage Enables/disables echo (ATE overrides this)

Default values Enabled

Error messages None defined yet.

39.3.34 Verbose

Syntax `[no] verbose`

Context *atcmd* context

Usage Enables/disables verbose result messages (ATV overrides this). When disabled only the numeric codes for (the below) messages are displayed.

Default values Enabled

Error messages None defined yet.

39.3.35 Sync Connection

Syntax `[no] sync-connect`

Context *atcmd* context

Usage Enables/disables synchronized connections

Default values Disabled

Error messages None defined yet.

39.3.36 Sync Connection Timeout

Syntax `[no] sync-timeout <TIMEOUT>`

Context *atcmd* context

Usage Set the timeout for synchronized connections

Default values 60 s

Error messages None defined yet.

39.3.37 OK-Message

Syntax `[no] ok-message text <MESSAGE> [code <CODE>]`

Context *atcmd* context

Usage Configure the OK message, verbose mode text and numeric code. Omitting the code will use the default value. If verbose (above) is disabled the numeric code is displayed.

Default values Text: OK, Code: 10

Error messages None defined yet.

39.3.38 Error-Message

Syntax `[no] error-message text <MESSAGE> [code <CODE>]`

Context *atcmd* context

Usage Configure the Error message, verbose mode text and numeric code. Omitting the code will use the default value. If verbose (above) is disabled the numeric code is displayed.

Default values Text: ERROR, Code: 4

Error messages None defined yet.

39.3.39 Connect-Message

Syntax `[no] connect-message text <MESSAGE> [code <CODE>]`

Context *atcmd* context

Usage Configure the Connect message, verbose mode text and numeric code. Omitting the code will use the default value. If verbose (above) is disabled the numeric code is displayed.

Default values Text: CONNECT, Code: 1

Error messages None defined yet.

39.3.40 Disconnect-Message

Syntax `[no] disconnect-message text <MESSAGE> [code <CODE>]`

Context *atcmd* context

Usage Configure the Disconnect message, verbose mode text and numeric code. Omitting the code will use the default value. If verbose (above) is disabled the numeric code is displayed.

Default values Text: NO CARRIER, Code: 3

Error messages None defined yet.

39.3.41 Map

Syntax `[no] map number <PSTN-NUMBER> to <ADDRESS[:PORT]>`

Context *atcmd* context

Usage Map PSTN number to a remote server IP address.

Default values 9000 (Default port number is **”9000”**)

Error messages None defined yet.

39.3.42 User-Message

Syntax `[no] user-message command <COMMAND> text <MESSAGE>`

Context *atcmd* context

Usage Configures custom user strings and answer. The answer can be up to 64 characters long including line breaks.

Default values N/A

Error messages None defined yet.

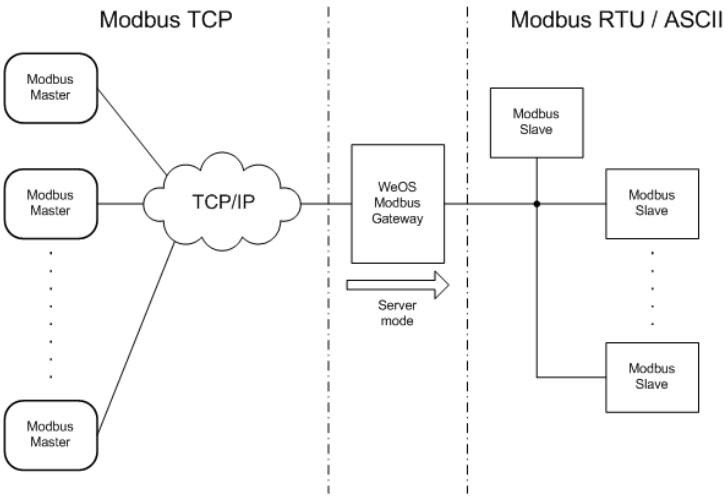
Chapter 40

Modbus Gateway

This chapter describes the *Modbus Gateway* application available on WeOS products equipped with a serial port. The Modbus Gateway is used for interconnecting a Modbus/TCP network with a Modbus/RTU or a Modbus/ASCII network.

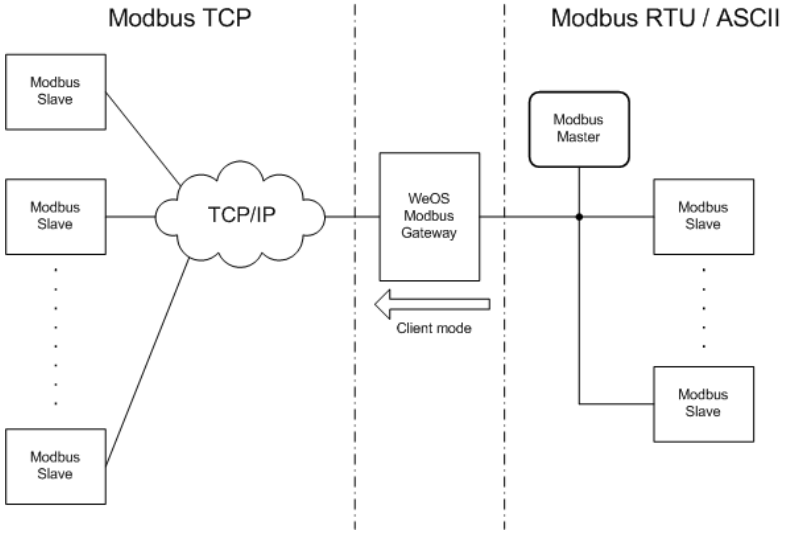
The Modbus Gateway has two operational modes, *server* and *client*:

- *Server:*[1] Allowing one or many Modbus/TCP masters to connect to one or many serial Modbus slaves. In this mode the gateway acts as Modbus/TCP slave on the TCP/IP side and a Modbus Master on the serial side. All incoming Modbus/TCP requests are converted into either Modbus/RTU or Modbus/ASCII requests on the serial side.



[1] In *server* mode, the Modbus Gateway acts as a server on the TCP/IP side (TCP Server).

- *Client:*[2] Allowing one serial Modbus Master to connect to one or many Modbus/TCP slaves. In this mode the Gateway acts as a Modbus Master on the TCP/IP side and a Modbus Slave on the serial side. All Modbus/RTU requests (or Modbus/ASCII requests) on the serial side are converted into Modbus/TCP requests on the TCP/IP side.



[2] In *client* mode, the Modbus Gateway acts as a client on the TCP/IP side (TCP Client).

40.1 Managing Modbus Gateway via the web interface

The Web interface provides configuration of all Modbus Gateway Settings.

40.1.1 Modbus Gateway Overview

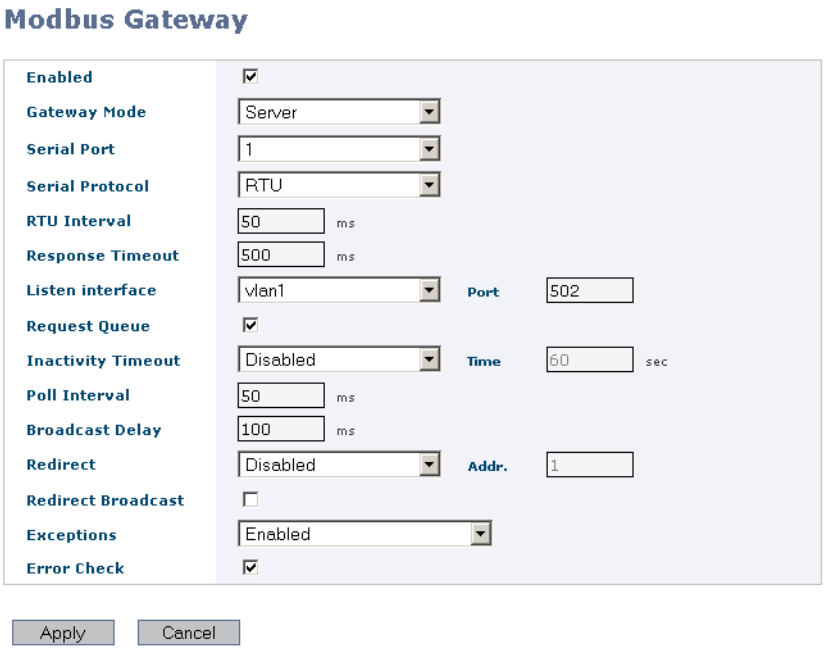
Menu path: Configuration ⇒ Serial ⇒ Modbus

If no Modbus Gateway is configured, click the **New** button to create a Modbus Gateway and you will be presented to the edit page described in section 40.1.2. Otherwise, the Modbus Gateway will be presented in a short overview.

40.1.2 Edit Modbus Gateway Settings

Menu path: Configuration ⇒ Serial ⇒ Modbus ⇒ **New**

Menu path: Configuration ⇒ Serial ⇒ Modbus ⇒ 



Configuration of Modbus Gateway in Server Mode

Modbus settings common to Client and Server Modes

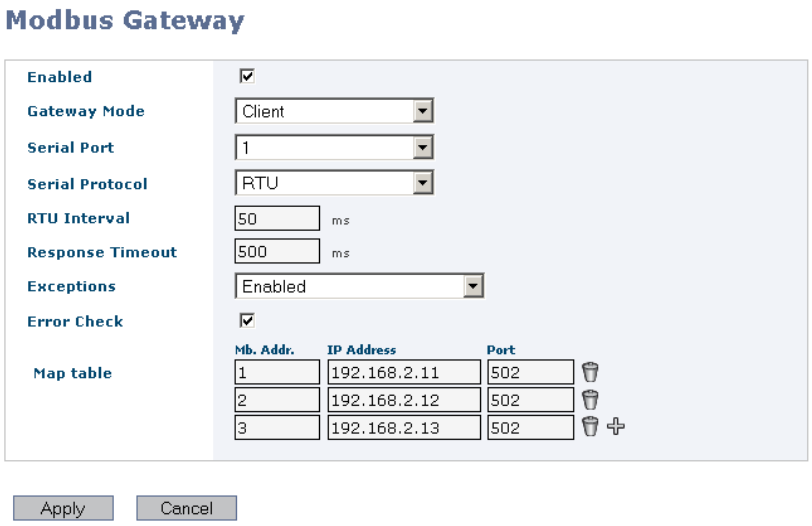
Gateway Mode	Configures the Modbus Gateway mode: (TCP) *Server* or (TCP) *Client*
Serial Port	Selects the Serial Port
Serial Protocol	Configures Serial protocol (RTU or ASCII)
RTU Interval	Configures the RTU-interval if RTU is selected as serial protocol.
ASCII Timeout	Configures the ASCII timeout if ASCII is selected as serial protocol.
Response Timeout	Configures the Response timeout
Error Check	Enables or disabled CRC/LRC error check

Continued on next page

	Continued from previous page
Exceptions	Configures Modbus Exceptions. With Modbus Exceptions enabled, the Modbus Gateway will react upon and respond to certain error conditions, e.g., if a Gateway in *client* mode receives a (serial) Modbus message addressed to Modbus unit not found in its *Map table*. If exception handling is desired, except for this particular situation, the gateway can be configured to ignore exceptions for unknown units.

Modbus Server specific settings

Listen interface	Configures the listen Interface and port.
Poll Interval	Configures the Poll Interval
Inactivity Timeout	Configures the Inactivity timer
Broadcast Delay	Configures the Broadcast delay
Redirect	Configures the Modbus Address redirection
Redirect Broadcast	Enables/Disables redirect of Modbus broadcasts
Request Queue	Enables/Disables the Request Queue



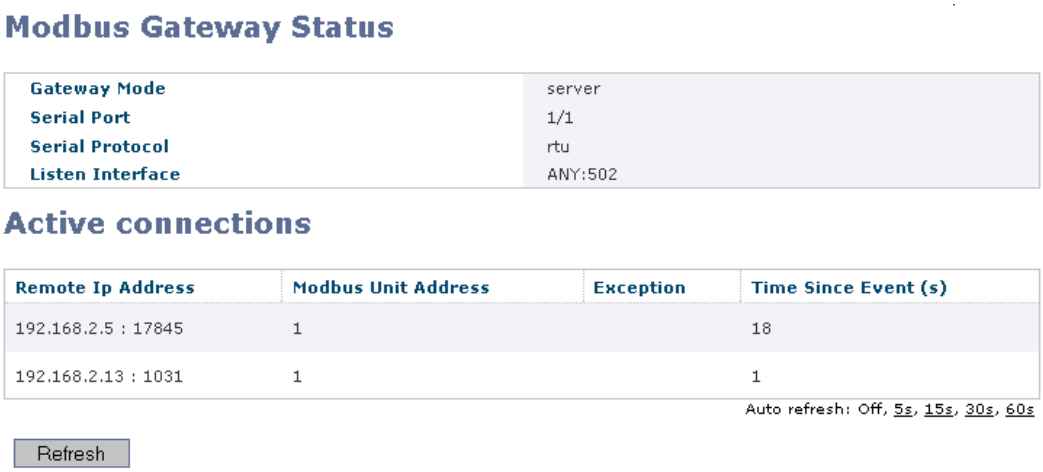
Configuration of Modbus Gateway in Client Mode

Modbus Client specific settings

Map table	Configures the mapping of Modbus unit number to IP address and TCP port number of Modbus/TCP slaves.

40.1.3 Modbus Gateway Status Page

Menu path: Status ⇒ Serial ⇒ Modbus



Gateway Mode	Current operation mode of the gateway, **Server** or **Client**.
Serial Port	The serial port in use.
Serial Protocol	Serial protocol, **RTU** or **ASCII**.
Listen Interface	Listen interface and port (only in server mode).
Active connections	This list shows active and open sessions. The gateway can only handle 16 open TCP sessions; if a new connection is needed, the oldest session will be closed.
Auto Refresh	Click on a value to make the page reload with updated statistics automatically every 5, 15, 30 or 60 seconds. Click **Off** to turn off auto refresh.
Refresh	Click on this button to reload with updated statistics.

40.2 Managing Modbus Gateway via the CLI interface

The table below shows Modbus Gateway management features available via the CLI.

Command	Default	Section
Configure Modbus Gateway settings		
modbus		Section 40.2.1
[no] mode <server|client>	server	Section 40.2.2
[no] port <SERIAL-PORT>	Disabled	Section 40.2.3
[no] serial-protocol <rtu|ascii>	rtu	Section 40.2.4
[no] listen <IFACE> [port <PORT>]	Disabled	Section 40.2.5
[no] rtu-interval <MILLISECONDS>	50	Section 40.2.7
[no] ascii-timeout <MILLISECONDS>	1000	Section 40.2.6
[no] poll-interval <MILLISECONDS>	50	Section 40.2.8
[no] error-check	Enabled	Section 40.2.9
[no] inactivity-timeout <SECONDS>	Disabled	Section 40.2.10
[no] response-timeout <MILLISECONDS>	500	Section 40.2.11
[no] broadcast-delay <MILLISECONDS>	100	Section 40.2.12
[no] redirect <MODBUS-ADDR>	Disabled	Section 40.2.13
[no] redirect-broadcast	Disabled	Section 40.2.14
[no] request-queue	Enabled	Section 40.2.15
[no] exceptions [ignore-unknown]	Enabled	Section 40.2.16
[no] map unit <MODBUS-ADDR> address <IPADDRESS>[:<PORT>]	N/A	Section 40.2.17
Show Modbus Gateway settings		
modbus		
show		Section 40.2.18
show mode		Section 40.2.19
show port		Section 40.2.20
show serial-protocol		Section 40.2.21
show listen		Section 40.2.22
show rtu-interval		Section 40.2.23
show ascii-timeout		Section 40.2.24
show poll-interval		Section 40.2.25
show error-check		Section 40.2.26

Continued on next page

Continued from previous page

Command	Default	Section
show inactivity-timeout		Section 40.2.27
show response-timeout		Section 40.2.28
show broadcast-delay		Section 40.2.29
show redirect		Section 40.2.30
show redirect-broadcast		Section 40.2.31
show request-queue		Section 40.2.32
show exceptions		Section 40.2.33
show map		Section 40.2.34
Show Modbus Gateway Status		
show modbus		Section 40.2.35

40.2.1 Managing Modbus Gateway settings

Syntax `modbus`

Context *Global Configuration* context

Usage Enter the Modbus Gateway configuration context. As of WeOS v4.17.1 a single Modbus Gateway instance is supported.

Default values Not applicable.

Error messages None defined yet.

40.2.2 Setting Mode

Syntax `[no] mode <server|client>`

Context *modbus* context

Usage Set Modbus Gateway mode (*server* or *client*).

Default values server

Error messages None defined yet.

40.2.3 Setting Serial Port

Syntax `[no] port <SERIAL-PORT>`

Context *modbus* context

Usage Set serial port, e.g., use **"port 1"** to select serial port 1 on a single slot unit, or **"port 1/1"** to select serial port 1 in slot 1 of a slotted WeOS unit.

You can use the **"show serial"** command in Admin Exec (see section 38.3.11) mode to list information your serial ports, including the serial port numbers.

Default values Disabled (**"no port"**)

Error messages None defined yet.

40.2.4 Setting Serial Protocol

Syntax `[no] serial-protocol <rtu|ascii>`

Context *modbus* context

Usage Set Serial protocol (*RTU* or ASCII).

Default values rtu

Error messages None defined yet.

40.2.5 Setting listen interface and port

Syntax `[no] listen <IFNAME> [port <PORT>]`

Context *modbus* context (*server* mode only)

Usage Setting local interface and TCP port to listen to. Acceptable port range is in range <0-65335>, where **"port 0"** results in using the default port number (502).

Only applicable when configuring the Modbus Gateway in **"server"** mode (see section 40.2.2).

Default values Disabled (**"no listen"**) When enabled, the default port is 502.

Error messages None defined yet.

40.2.6 Setting ASCII Timeout

Syntax `[no] ascii-timeout <MILLISECONDS>`

Context *modbus* context

Usage Set ASCII Timeout, i.e., the maximum time between two frames in ASCII mode. Allowed range is <10-7050> (milliseconds).

Use **"`no ascii-timeout`"** to reset the timeout to the default value.

The **"`ascii-timeout`"** command is only applicable when configuring the Modbus Gateway to use **"`ascii`"** as serial protocol (see section 40.2.4).

Default values 1000

Error messages None defined yet.

40.2.7 Setting RTU Interval

Syntax `[no] rtu-interval <MILLISECONDS>`

Context *modbus* context

Usage Set RTU Interval, i.e., the minimum time between two frames in RTU mode. Allowed range is <10-7050> (milliseconds).

Use **"`no rtu-interval`"** to reset the timeout to the default value.

The **"`rtu-interval`"** command is only applicable when configuring the Modbus Gateway to use **"`rtu`"** as serial protocol (see section 40.2.4).

Default values 50

Error messages None defined yet.

40.2.8 Setting Poll Interval

Syntax `[no] poll-interval <MILLISECONDS>`

Context *modbus* context (*server* mode only)

Usage Set Poll Interval. Allowed values are in range <10-65535> (milliseconds).

Only applicable when configuring the Modbus Gateway in **”server”** mode (see section 40.2.2).

Default values 50

Error messages None defined yet.

40.2.9 Setting Error Check

Syntax `[no] error-check`

Context *modbus* context

Usage Enabled/disables Error Check.

Default values Enabled

Error messages None defined yet.

40.2.10 Setting Inactivity Timeout

Syntax `[no] inactivity-timeout <SECONDS>`

Context *modbus* context (*server* mode only)

Usage Set Inactivity Timeout. Allowed values are in range <10-65535> (seconds).

Only applicable when configuring the Modbus Gateway in **”server”** mode (see section 40.2.2).

Default values Disabled

Error messages None defined yet.

40.2.11 Setting Response Timeout

Syntax `[no] response-timeout <MILLISECONDS>`

Context *modbus* context

Usage Set Response Timeout. Allowed values are in range <1-65535> (milliseconds).

Default values 500

Error messages None defined yet.

40.2.12 Setting Broadcast Delay

Syntax `[no] broadcast-delay <MILLISECONDS>`

Context *modbus* context (*server* mode only)

Usage Set Broadcast Delay, i.e., time to wait after transmitting an broadcast. Allowed values are in range <0-65535> (milliseconds).

Only applicable when configuring the Modbus Gateway in **"server"** mode (see section 40.2.2).

Default values 100

Error messages None defined yet.

40.2.13 Setting Redirect

Syntax `[no] redirect <MODBUS-ADDR>`

Context *modbus* context (*server* mode only)

Usage Set redirect, i.e., redirect all Modbus addresses to this address. Allowed values are in range <1-255> (seconds)

Only applicable when configuring the Modbus Gateway in **"server"** mode (see section 40.2.2).

Default values Disabled

Error messages None defined yet.

40.2.14 Setting Redirect Broadcast

Syntax `[no] redirect-broadcast` (*server* mode only)

Context *modbus* context

Usage Set redirect broadcast, i.e., redirect all Modbus broadcasts to Modbus address 1.

Only applicable when configuring the Modbus Gateway in **”server”** mode (see section 40.2.2).

Default values Disabled

Error messages None defined yet.

40.2.15 Setting Request Queue

Syntax `[no] request-queue`

Context *modbus* context (*server* mode only)

Usage Enabled/disables Request Queue.

Only applicable when configuring the Modbus Gateway in **”server”** mode (see section 40.2.2).

Default values Enabled

Error messages None defined yet.

40.2.16 Setting Exceptions

Syntax `[no] exceptions [ignore-unknown]`

Context *modbus* context

Usage Enable/disables handling of Modbus exceptions. With Modbus Exceptions enabled, the Modbus Gateway will react upon and respond to certain error conditions, e.g., if a Gateway in *client* mode receives a (serial) Modbus message addressed to Modbus unit not found in its *Map table*. If exception handling is desired, except for this particular situation, the gateway can be configured to ignore exceptions for unknown units (**”exceptions ignore-unknown”**).

Use **”no exceptions”** to disable exception handling.

Default values Enabled (all exceptions enabled)

Error messages None defined yet.

40.2.17 Managing Mapping of Modbus units to IP hosts

Syntax `[no] unit <MODBUS-ADDR> address <ADDRESS>[:<PORT>]`

Context *modbus* context (*client* mode only)

Usage Setup Modbus IP map.

Only applicable when configuring the Modbus Gateway in **"client"** mode (see section 40.2.2).

Default values N/A

Error messages None defined yet.

40.2.18 Show Modbus Gateway Setting

Syntax `show`

Context *modbus* context

Usage Show Modbus Gateway Setting

Default value Not applicable.

40.2.19 Show Modbus Gateway Mode Setting

Syntax `show mode`

Context *modbus* context

Usage Show Modbus Gateway Mode Setting

Default value Not applicable.

40.2.20 Show Serial Port Setting

Syntax `show port`

Context *modbus* context

Usage Show Modbus Gateway Mode Setting

Default value Not applicable.

40.2.21 Show Serial Protocol Setting

Syntax `show serial-protocol`

Context *modbus* context

Usage Show Modbus Gateway Mode Setting

Default value Not applicable.

40.2.22 Show Modbus Gateway Listen Setting

Syntax `show listen`

Context *modbus* context

Usage Show Modbus Gateway Listen Setting

Default value Not applicable.

40.2.23 Show RTU Interval Setting

Syntax `show rtu-interval`

Context *modbus* context

Usage Show Modbus Gateway RTU Interval Setting

Default value Not applicable.

40.2.24 Show ASCII Timeout Setting

Syntax `show ascii-timeout`

Context *modbus* context

Usage Show Modbus Gateway Ascii Timeout Setting

Default value Not applicable.

40.2.25 Show Poll Interval Setting

Syntax `show poll-interval`

Context *modbus* context

Usage Show Modbus Gateway Poll Interval Setting

Default value Not applicable.

40.2.26 Show Error Check Setting

Syntax `show error-check`

Context *modbus* context

Usage Show Modbus Gateway Error Check Setting

Default value Not applicable.

40.2.27 Show Inactivity Timeout Setting

Syntax `show inactivity-timeout`

Context *modbus* context

Usage Show Modbus Gateway Inactivity Timeout Setting

Default value Not applicable.

40.2.28 Show Response Timeout Setting

Syntax `show response-timeout`

Context *modbus* context

Usage Show Modbus Gateway Response Timeout Setting

Default value Not applicable.

40.2.29 Show Broadcast Delay Setting

Syntax `show broadcast-delay`

Context *modbus* context

Usage Show Modbus Gateway Broadcast Delay Setting

Default value Not applicable.

40.2.30 Show Redirect Setting

Syntax `show redirect`

Context *modbus* context

Usage Show Modbus Gateway Redirect Setting

Default value Not applicable.

40.2.31 Show Redirect Broadcast Setting

Syntax `show redirect-broadcast`

Context *modbus* context

Usage Show Modbus Gateway Redirect Broadcast Setting

Default value Not applicable.

40.2.32 Show Request Queue Setting

Syntax `show request-queue`

Context *modbus* context

Usage Show Modbus Gateway Request Queue Setting

Default value Not applicable.

40.2.33 Show Exceptions Setting

Syntax `show exceptions`

Context *modbus* context

Usage Show Modbus Gateway Exceptions Setting

Default value Not applicable.

40.2.34 Show Modbus Unit to IP Host Settings

Syntax `show map`

Context *modbus* context

Usage Show Modbus Gateway Map Setting

Default value Not applicable.

40.2.35 Show Modbus Gateway status

Syntax `show modbus`

Context *Admin Exec* context.

Usage Show Modbus Gateway status information

Default values Not applicable.

Error messages None defined yet.

Example

Example

```
example:/#> show modbus
Modbus Gateway Enabled    : Yes, running as PID 542
Mode                      : client
Serial port               : 1
Serial protocol           : rtu

Remote IP addr             Modbus Addr     Exception
=====================================================
192.168.2.5 : 502            10
example:/#>
```

Chapter 41

MicroLok II Gateway

This chapter describes the MicroLok II®[1] Gateway application available on WeOS products equipped with a serial port. The MicroLok Gateway is used for interconnecting MicroLok (serial) networks over an IP network. It provides a MicroLok Address Lookup table, which defines MicroLok Stations on the local serial ports and at remote gateways. A filtering mechanism ensures that only data belonging to established MicroLok sessions are forwarded over the IP network. Established sessions are supervised and an alarm can be indicated whenever one or more of these sessions are down.

41.1 Overview of MicroLok Gateway Properties and Management Features

Feature	Web	CLI	General Description
Enable/disable MicroLok Gateway	X	X	
Set Gateway Listen Interface/UDP-port	X	X	Sec. 41.1.1.1
Set Key-On/Key-Off/Grant Delay	X	X	Sec. 41.1.1.1
Manage MicroLok Address Lookup Table	X	X	Sec. 41.1.1.2
View MicroLok Status	X	X	Sec. 41.1.2
MicroLok Packet Monitoring	X		
Manage MicroLok Alarm	X	X	Chapter 24
Manage Serial Port Settings	X	X	Chapter 38

[1] MicroLok is a registered trademark of Ansaldo STS USA.

41.1.1 Introduction to MicroLok and the WeOS MicroLok Gateway

The MicroLok II Peer Protocol[3] provides a service where multiple communication sessions between MicroLok Stations can be multiplexed over serial lines. The WeOS MicroLok Gateway can be used to extend the MicroLok network over an IP network, by encapsulation the data packets from the serial side into IP/UDP packets. The packets will be received by a remote MicroLok Gateway, which will decapsulate the IP/UDP packets before forwarding on its serial side.

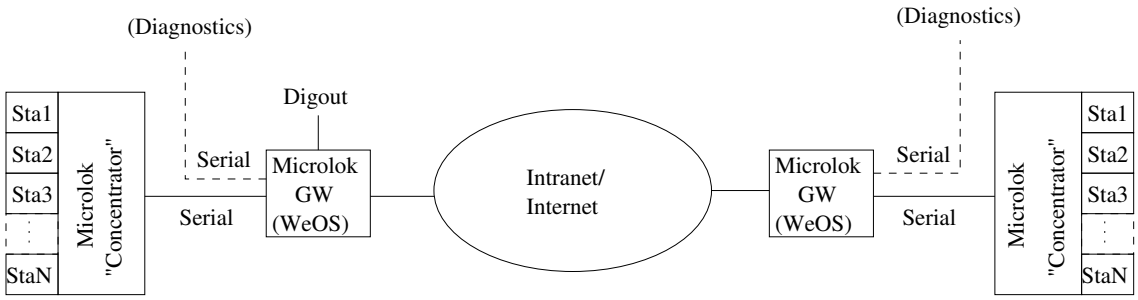


Figure 41.1: Use of MicroLok Gateways to extend a MicroLok network.

MicroLok Diagnostic data is assumed to use the WeOS Serial Over IP service (see chapter 39) on a different serial port or a different WeOS unit, and is not covered by the MicroLok Gateway functionality specified here. (The dashed lines in Fig. 41.1 illustrates the use of a separate serial port for diagnostic data.)

The WeOS MicroLok gateway listens for MicroLok packets on its serial interface, and forwards them as IP/UDP packets to a remote MicroLok Gateway if a match is found in its MicroLok Address Table. Similarly, the gateway listens for MicroLok IP/UDP packets, and forwards them on its serial port. General MicroLok settings for IP interface and UDP port, etc., is described further in section 41.1.1.1.

The example topology in fig. 41.1 can be extended to use more than two MicroLok gateways. And WeOS units with more than one serial port is able to connect local MicroLok stations on all its serial ports. Further information on connecting local and remote MicroLok stations is given in section 41.1.1.2 when describing the address lookup table.

The MicroLok gateway will keeps track of initialised MicroLok sessions, and will drop MicroLok packets not being part of an established session. The gateway will also monitor established sessions, and is able to signal an alarm if one or more sessions are determined down, see section 41.1.2.

41.1.1.1 General MicroLok Gateway Parameters

As of WeOS v4.17.1 a single MicroLok gateway instance is supported. You can specify which network interface, and which UDP port the gateway should listen on. By default, the gateway will listen for IP/UDP MicroLok packets on *any* network interface and UDP port 60000.

Firewall rules will be added automatically for the specified interface and UDP port. These rules will accept IP/UDP MicroLok packets *coming in on the specified listening interface* and destined for the given UDP port. This automatic firewall rule will suffice in most use cases (if further firewall tweaking is needed, see chapter 31).

The general MicroLok settings also includes a set of MicroLok specific timeouts applying to MicroLok communication on all local serial ports: *Key-On delay*, *Key-Off delay* and *Grant delay*. These timeouts are all disabled by default, which should be fine in most use cases.

41.1.1.2 The MicroLok Address Lookup Table

The MicroLok Gateway contains an address lookup table. As of WeOS v4.17.1 the table can hold up to 64 address entries, and each entry containing the following items:

- MicroLok Station Address: A two-byte MicroLok address, given as a hexadecimal number, e.g., **”001f”**.
- A local or remote destination: The location of the given MicroLok station address is either specified as a local serial port, e.g., **”serial 1”**, or as a remote gateway (IP address and optionally a UDP port number), e.g., **”remote 192.168.2.1:12345”** for a remote gateway with IP address 192.168.2.1 listening on UDP port 12345. Either a local serial port or a remote gateway needs to be specified for an address entry to be active.

  When remote gateway is specified, the default UDP port number is 60000.
- Station Description: An optional 15 byte text string to describe the given MicroLok station. Disabled by default.
- Session Timeout: Timeout used to determine if a session established with this MicroLok station is down. The session timeout can be in range 500-600000 (milliseconds). Disabled by default.

The MicroLok Gateway can be configured to give an alarm if one or more of the established sessions is down, see sec:microlok-status-alarm.

Note
Serial port settings (rate, start bits, stop bits, parity) etc. is configured in the general serial port contexts, see sections 38.2 (Web) and 38.3 (CLI). Note that MicroLok stations typically operate with speeds in range 300-38400 bits/s, while serial ports on WeOS units default to rate 115200 bit/s, and should therefore be changed.

41.1.2 MicroLok Session Status and Alarm Handling

The MicroLok Gateway keeps track of established MicroLok sessions, and filters out data packets not belonging to an established session. (MicroLok session *initialisation* packets are always forwarded by the gateway, but MicroLok *data* packets not matching an established session will be dropped.) It is possible to view the MicroLok session status (section 41.2.3 for Web and section 41.3.9 for CLI), to list which sessions that are UP, which are DOWN, and which are in INIT state:

- Status UP: A session is considered up when the initialisation handshake between the two stations has finished, and where data and control traffic is exchanged within the given session timeout for these stations.
- Status DOWN: For a session to be listed as DOWN, it must first have been UP. If no data or control traffic for the session is detected within the given session timeout it is considered down.

  Sessions which are DOWN will have their data packets filtered. To get status UP, a new initialisation handshake is needed.
- Status INIT: New sessions which have not yet finished the initialisation handshake are listed as INIT.

When the MicroLok gateway service starts, the list of *current sessions* is empty. (The same is true if a MicroLok Gateway configuration change occurs, as this implies a restart of the gateway process.) As new sessions are established the list of *current MicroLok sessions* grows. As of WeOS v4.17.1, up to 64 MicroLok sessions are supported. When one or more of the sessions in the current list is determined DOWN, a MicroLok alarm will be raised. The alarm will indicate failure state until all sessions in the current list have got status UP. To use this MicroLok summary alarm function, you need to create a *MicroLok alarm trigger*,

and map this trigger to a suitable *alarm action*. Thereby you can use, e.g., digital out to indicate if there is a problem with any of the MicroLok sessions, as hinted in fig. 41.1. For more information on how to set up MicroLok alarm triggers and alarm actions, see chapter 24. A brief CLI example is given in section 24.3.2.11.

41.2 Managing MicroLok Gateway via the web interface

The Web interface provides configuration of all MicroLok Gateway Settings.

41.2.1 MicroLok Gateway Overview

Menu path: Configuration ⇒ Serial ⇒ MicroLok

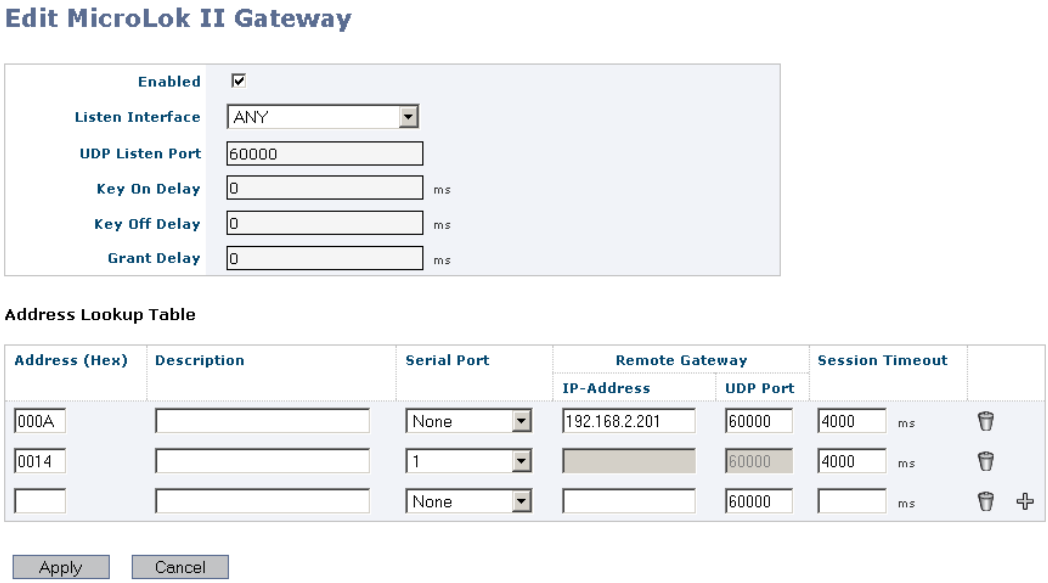
If no MicroLok Gateway is configured, the page below will be displayed. Click the **New** button to create a MicroLok Gateway and you will be presented to the edit page described in section 41.2.2.

If a MicroLok Gateway is configured, the short overview page below will be displayed.

41.2.2 Edit MicroLok Gateway Settings

Menu path: Configuration ⇒ Serial ⇒ MicroLok ⇒ **New**, or

Menu path: Configuration ⇒ Serial ⇒ MicroLok ⇒ 



MicroLok gateway settings

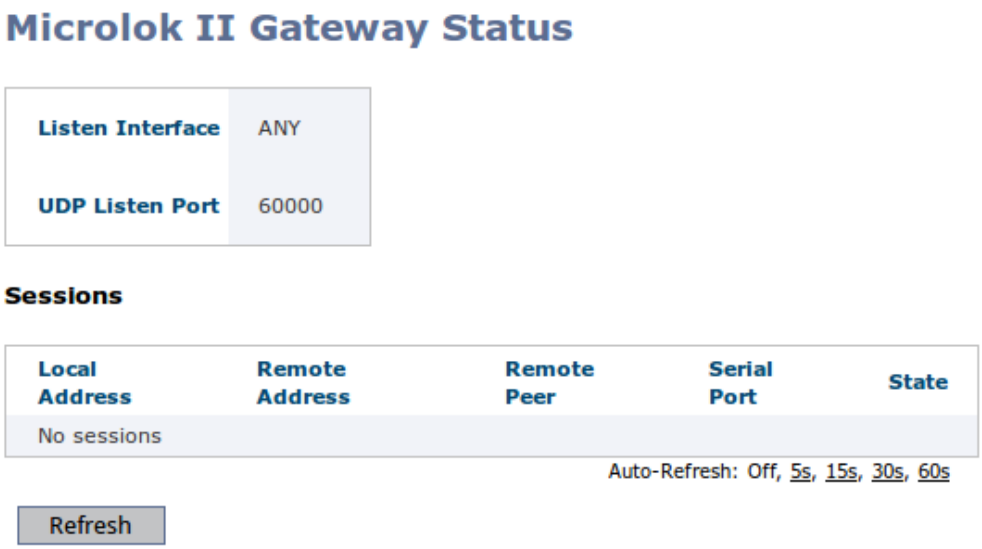
Enable	Enable or disable MicroLok II gateway instance
Listen Interface	Setting listen interface address.
UDP Listen Port	Setting UDP listen port. Acceptable port range is in range <0-65335>, where **"port 0"** results in using the default port number (60000).
Key On Delay	Set Key-On signal delay in milliseconds, i.e. key-on-delay 100. To disable this function set value to 0 (zero). Allowed range is <1-1000> (milliseconds).
Key Off Delay	Set Key-Off signal delay in milliseconds, i.e. key-off-delay 100. To disable this function set value to 0 (zero). Allowed range is <1-1000> (milliseconds).
Grant Delay	Set Grant signal delay in milliseconds, i.e. grant-delay 100. To disable this function set value to 0 (zero). Allowed range is <500-600000> (milliseconds).

MicroLok gateway address table settings

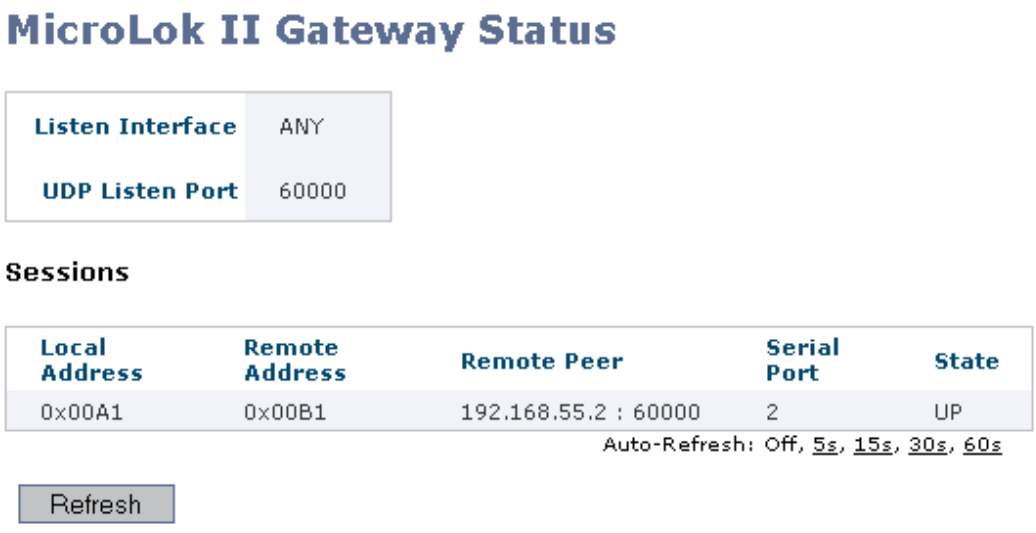
Address	Setting for the MicroLok address.
Description	Description of the MicroLok address table setting.
Serial Port	Setting for the local serial port.
Remote Gateway	Setting for the remote MicroLok peer address.
Session Timeout	Setting for the session timeout (will be rounded to nearest 100 ms).

41.2.3 MicroLok Gateway Status Page

Menu path: Status ⇒ Serial ⇒ MicroLok (no active sessions)



Menu path: Status ⇒ Serial ⇒ MicroLok (active sessions)



Listen Interface	Listen interface.
UDP Listen Port	Listen port.

41.3 Managing MicroLok Gateway via the CLI interface

The table below shows MicroLok Gateway management features available via the CLI.

Command	Default	Section
Configure MicroLok Gateway settings		
[no] microlok		Section 41.3.1
[no] enable	Enabled	Section 41.3.2
[no] listen <any|IFNAME>[:PORT]	any:60000	Section 41.3.3
[no] key-on-delay <MSEC>	Disabled	Section 41.3.4
[no] key-off-delay <MSEC>	Disabled	Section 41.3.5
[no] grant-delay <MSEC>	Disabled	Section 41.3.6
[no] map station [dec] <MICROLOK-ADDR> <remote <IPADDRESS>[:<PORT>] | <serial <SERIALPORT>> [session-timeout <MSEC>] [description <STRING>]		Section 41.3.7
MicroLok Gateway Status and Packet Monitoring		
show microlok		Section 41.3.9
microlok		Section 41.3.10
[no] monitor		Section 41.3.11

41.3.1 Managing MicroLok Gateway settings

Syntax `no microlok`

Context Global Configuration context

Usage Enter the MicroLok Gateway configuration context. A gateway instance will be created, unless it already exists. As of WeOS v4.17.1 a single MicroLok Gateway instance is supported.

Use **”`no microlok`”** to remove your Gateway instance(s).

Use **”show microlok”** (from Global Configuration context) to list summary information for configured MicroLok gateway instances. Use **”show”** (within the Microlok Gateway Configuration context) to list detailed information on a specific MicroLok gateway instance.

Default values Not applicable.

Example

```
example:/config/#> show microlok
Microlok II Gateway
ID  Enabled    Serial     Listen Port
================================================================================
 1  Enabled    1          60000
example:/config/#> microlok
example:/config/microlok-1/#> show
Microlok II gateway
Status              : Enabled
Listen              : ANY port 60000
Key On Delay        : Disabled
Key Off Delay       : Disabled
Grant Delay         : Disabled

Address Lookup Table (ALT):
--------------------------------------------------------------------------------
Addr  Remote Peer            Serial  Description       Session Timeout (ms)
0010  N/A                    1       siteA-ctrl3       Disabled
0020  192.168.3.3:60000      N/A     siteB-ctrl1       Disabled

example:/config/microlok-1/#>
```

41.3.2 Setting Enable

Syntax `[no] enable`

Context Microlok Gateway Configuration context

Usage Use **”enable”** to enable and **”no enable”** to disable a MicroLok II gateway instance.

Use **”show enable”** to show whether this gateway instance is enabled or disabled.

Default values Enabled

41.3.3 Setting listen interface and port

Syntax `[no] listen <any|IFNAME>[:PORT]`

Context Microlok Gateway Configuration context

Usage Setting local interface and UDP port to listen to, e.g., **"listen vlan1:45678"** to listen on interface *vlan1* and UDP port *45678*. Keyword **"any"** can be used to listen on all interfaces.

Acceptable port range is <1-65335>, where default port number is 60000.

Use **"no listen"** to reset to default settings (listen on any interface, UDP port 60000).

Use **"show listen"** to list the current setting.

Default values Any interface, UDP port 60000.

41.3.4 Setting key-on-delay

Syntax `[no] key-on-delay <MILLISECONDS>`

Context Microlok Gateway Configuration context

Usage Set Key-On signal delay in milliseconds, e.g., **"key-on-delay 100"**. Allowed range is <1-1000> (milliseconds). Use **"no key-on-delay"** or **"key-on-delay 0"** to disable this function.

Use **"show key-on-delay"** to list the current Key-On signal delay setting.

Default values Disabled (0)

41.3.5 Setting key-off-delay

Syntax `[no] key-off-delay <MILLISECONDS>`

Context Microlok Gateway Configuration context

Usage Set Key-Off signal delay in milliseconds, e.g. key-off-delay 100. Allowed range is <1-1000> (milliseconds). Use **"no key-off-delay"** or **"key-off-delay 0"** to disable this function.

Use **"show key-off-delay"** to list the current Key-Off signal delay setting.

Default values Disabled (0)

41.3.6 Setting grant-delay

Syntax `[no] grant-delay <MILLISECONDS>`

Context Microlok Gateway Configuration context

Usage Set Grant signal delay in milliseconds, e.g. **”grant-delay 100”**. Allowed range is <1-1000> (milliseconds). Use **”no grant-delay”** or **”grant-delay 0”** to disable this function.

Use **”show grant-delay”** to list the current Grant signal delay setting.

Default values Disabled (0)

41.3.7 Managing Mapping of MicroLok units to IP hosts

Syntax `[no] map station [dec] <MICROLOK-ADDR> <serial <SERIALPORT> | remote <IPADDRESS[:UDPPORT]>> [session-timeout <ms>] [description <STRING>]`

Context Microlok Gateway Configuration context

Usage Configure the MicroLok Address Lookup Table entries. Up to 64 MicroLok Station entries can be added.

- **MicroLok addresses:** MicroLok addresses are given in hexadecimal form (range: 0-ffff), e.g., **”map station ff serial 1”** or **”map station 0xff serial 1”**. Add the keyword **”dec”** to interpret an address as decimal, e.g., **”map station dec 30 serial 1”** is equivalent to **”map station 1e serial 1”**.
- **Local or Remote Station:** An address entry is only active if a (local) serial port or a (remote) gateway IP address is set:
  - For local MicroLok addresses the serial port is set, e.g., **”map station 10 serial 1”**
  - For remote MicroLok addresses the remote gateway IP address (and optionally the UDP port) is set, e.g., **”map station 20 remote 192.168.3.1:34567”**. Default UDP port is 60000.

- **Session Timeout:** The session timeout will be rounded to nearest 100 ms. Valid range: 500-600000 (ms). Default: Disabled.
- **Description:** The optional description string (15 characters) can be used to provide information on the MicroLok station. Default: Disabled

Use **”no map station MICROLOK-ADDRESS”** (e.g., **”no map station 10”**) to remove a specific station entry, and **”no map”** to remove all station entries from the MicroLok address lookup table.

Default values For remote MicroLok stations, default UDP port is 60000.

41.3.8 Show MicroLok Gateway Setting

Syntax `show`

Context Microlok Gateway Configuration context

Usage Show MicroLok Gateway Setting

Default value Not applicable.

41.3.9 Show MicroLok Gateway status

Syntax `show microlok`

Context Admin Exec context.

Usage Show MicroLok Gateway status information (also available as **”show”** command within the Microlok Gateway Status context.

Default values Not applicable.

Example

```
example:/#> show microlok
Microlok II Gateway Enabled    : Yes, running as PID 615
Listen Interface               : ANY
UDP Listen Port                : 60000

Local          Remote
Station Serial Station Gateway                State
=================================================
0014    1      000A    192.168.2.204:60000    UP
```

41.3.10 Enter MicroLok Gateway Status Context

Syntax `microlok`

Context Admin Exec context.

Usage Enter Microlok Gateway Status context. From this context, you can show MicroLok Gateway status (**"show"**, see also section 41.3.9) and enable/disable MicroLok packet monitoring (see section 41.3.11).

Default values Not applicable.

41.3.11 Enable MicroLok Packet Monitoring

Syntax `[no] monitor`

Context Microlok Gateway Status context.

Usage MicroLok packets being exchanged can be displayed on the WeOS console. Use **"monitor"** to enable monitoring of MicroLok packets, and **"no monitor"** to disable it.

Enabling MicroLok packet monitoring is a debugging facility, not a configuration setting.

Default values Not applicable.

Example

```
example:/#> microlok
example:/microlok#> monitor
Enabling Microlok II Gateway monitor.
example:/microlok#>

### WeOS Microlok II monitor
MSG Status: RCV (00a4)  DEST ADDR: 000a  SRC ADDR: 0014
MSG TYPE: 01  SNDMSN: 00  RCVMSN: 00  07//06//34 18:54:82
f4 00 0a 00 14 00 00 01 00 a4 07 06 22 12 36 52 1b 00 01 30 38 2e 35 30
00 38 30 2f d6 2f a2 3e 92 94 7d 01 01 01 08 01 01 01 08 01 c5 43 be 2c
f6


### WeOS Microlok II monitor
MSG Status: XMT (0021)  DEST ADDR: 0014  SRC ADDR: 000a
MSG TYPE: 06  SNDMSN: 36  RCVMSN: 7e  07//06//34 18:54:82
f4 00 14 00 0a 36 7e 06 00 21 07 06 22 12 36 52 b4 0f 20 66 f6


### WeOS Microlok II monitor
MSG Status: RCV (0021)  DEST ADDR: 000a  SRC ADDR: 0014
MSG TYPE: 06  SNDMSN: 7e  RCVMSN: 36  07//06//34 18:54:82
f4 00 0a 00 14 7e 36 06 00 21 07 06 22 12 36 52 1e 11 3e 48 f6

example:/#>
example:/#> no monitor
Disabling Microlok II Gateway monitor.
```

Part VI

Appendixes

Acronyms and abbreviations

3DES	Triple DES
AAA	Authentication, Authorisation and Accounting
AH	Authentication Header
ASCII	American Standard Code for Information Interchange
AES	Advanced Encryption Standard
AVT	Adaptive VLAN Trunking (Westermo propriatary dynamic VLAN function)
CA	Certificate Authority
CIDR	Classless Inter-Domain Routing
CLI	Command Line Interface
CN	Common Name (X.509 certificate term)
CPU	Central Processing Unit
dBm	Power ratio in dB referenced to 1 mW. (Used for DDM SFP optic power representation.)
DDM	Digital Diagnostics Montitoring
DDNS	Dynamic DNS
DDOS	Distributed Denial of Service (Attack)
DES	Data Encryption Standard
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DN	Distinguished Name (X.509 certificate term)
DNS	Domain Name System
DOM	Digital Optics Monitoring
DPD	Dead Peer Detection
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
EAP	Extensible Authentication Protocol

ECN	Enhanced Congestion Notification
ESP	Encapsulating Security Payload
FRNT	Fast Reconfiguration of Network Topology
GRE	Generic Routing Encapsulation
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP (HTTP over SSL/TLS)
I/O	Input/Output
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
IKEv1	IKE version 1
IP	Internet Protocol
IPsec	IP Security
IPv4	IP version 4
IPv6	IP version 6
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LED	Light Emitting Diode
LFF	Link Fault Forward
LLDP	Link Layer Discovery Protocol
MD5	Message Digest 5
MIB	Management Information Base
MTU	Maximum Transfer Unit
NAPT	Network Address and Port Translation
NAT	Network Address Translation
NAT-T	NAT Traversal
NBMA	Non-Broadcast Multiple Access
NTP	Network Time Protcol
OID	Object Identifier
OSPF	Open Shortest Path First
PAF	PME Aggregation Function (SHDSL link bonding)
PC	Personal Computer
PEM	Privacy Enhanced Mail (X.509 certificate term)
PFS	Perfect Forward Secrecy
PHB	Per-Hop Behaviour
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PNAC	Port-based Network Access Control
PPP	Point to Point Protocol

RIP	Routing Information Protocol
RAM	Random Access Memory
RDN	Relative Distinguished Name (X.509 certificate term)
RMON	Remote Monitoring
RSA	Rivest, Shamir, and Adleman (public key encryption algorithm)
SHDSL	Symmetric High-speed Digital Subscriber Line
SFP	Small Form-factor Pluggable (transceiver module)
SHA	Secure Hash Algorithm
SHA-1	Secure Hash Algorithm 1
SNMP	Simple Network Management Protocol
SNR	Signal to Noise Ratio
SNTP	Simple NTP
SSH	Secure SHell
SSL	Secure Socket Layer
TLS	Transport Layer Security
ToS	Type of Service
USB	Universal Serial Bus
VFS	Virtual File System
VIP	Virtual IP Address (VRRP)
VLAN	Virtual LAN
VPN	Virtual Private Network
VRID	Virtual Router Identifier (VRRP)
VRRP	Virtual Router Redundancy Protocol
WAN	Wide Area Network
WeOS	Westermo OS

Bibliography

[1] S. Alexander and R. Droms. DHCP Options and BOOTP Vendor Extensions. rfc 2132, IETF, March 1997.

[2] P. Almquist. Type of Service in the Internet Protocol Suite. rfc 1349, IETF, July 1992.

[3] MicroLok II Peer Protocol Application Guidelines. Ansaldo STS, SM 9726, Rev. 2, May 2009.

[4] M. Christensen, K. Kimball, and F. Solensky. Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches. rfc 4541, IETF, May 2006.

[5] G. Clark. Telnet Com Port Control Option. rfc 2217, IETF, October 1997.

[6] S.E. Deering. Host extensions for IP multicasting. rfc 1112, IETF, August 1989.

[7] R. Droms. Dynamic Host Configuration Protocol. rfc 2131, IETF, March 1997.

[8] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. Generic Routing Encapsulation (GRE). rfc 2784, IETF, March 2000.

[9] D. Grossman and J. Heinanen. Multiprotocol Encapsulation over ATM Adaptation Layer 5. rfc 2684, IETF, September 1999.

[10] C.L. Hedrick. Routing Information Protocol. rfc 1058, IETF, June 1988.

[11] R. Hinden and Ed. Virtual Router Redundancy Protocol (VRRP). rfc 3768, IETF, April 2004.

[12] IEEE 802.1AB Station and Media Access Control Connectivity Discovery. IEEE Standard for Local and metropolitan area networks, 2005.

[13] IEEE 802.1AX Link Aggregation. IEEE Standard for Local and metropolitan area networks, 2008.

[14] IEEE 802.1Q: Virtual Bridged Local Area Networks. IEEE Standard for Local and metropolitan area networks, 2005.

[15] IEEE 802.1X: Port-Based Network Access Control. IEEE Standard for Local and metropolitan area networks, 2001.

[16] IEEE 802.3af. Amendment: Data Terminal Equipment (DTE) Power via Media Dependent Interface (MDI). IEEE Standard for Local and metropolitan area networks, Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2003.

[17] IEEE 802.3at. Amendment: Data Terminal Equipment (DTE) Power via Media Dependent Interface (MDI) Enhancements. IEEE Standard for Local and metropolitan area networks, Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2009.

[18] R. Johnson, J. Kumarasamy, K. Kinnear, and M. Stapp. DHCP Server Identifier Override Suboption. rfc 5107, IETF, February 2008.

[19] S. Knight, D. Weaver, D. Whipple, R. Hinden, D. Mitzel, P. Hunt, P. Higginson, M. Shand, and A. Lindem. Virtual Router Redundancy Protocol. rfc 2338, IETF, April 1998.

[20] B. Lloyd and W. Simpson. PPP Authentication Protocols. rfc 1334, IETF, October 1992.

[21] G. Malkin. RIP Version 2. rfc 2453, IETF, November 1998.

[22] G. Malkin and A. Harkin. TFTP Option Extension. rfc 2347, IETF, May 1998.

[23] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler. A Method for Transmitting PPP Over Ethernet (PPPoE). rfc 2516, IETF, February 1999.

[24] G. McGregor. The PPP Internet Protocol Control Protocol (IPCP). rfc 1332, IETF, May 1992.

[25] S. Nadas and Ed. Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6. rfc 5798, IETF, March 2010.

[26] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. rfc 2474, IETF, December 1998.

[27] G. Pall and G. Zorn. Microsoft Point-To-Point Encryption (MPPE) Protocol. rfc 3078, IETF, March 2001.

[28] M. Patrick. DHCP Relay Agent Information Option. rfc 3046, IETF, January 2001.

[29] J. Postel. Internet Protocol. rfc 0791, IETF, September 1981.

[30] K. Ramakrishnan and S. Floyd. A Proposal to add Explicit Congestion Notification (ECN) to IP. rfc 2481, IETF, January 1999.

[31] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. rfc 3168, IETF, September 2001.

[32] D. Rand. The PPP Compression Control Protocol (CCP). rfc 1962, IETF, June 1996.

[33] C. Rigney, W. Willats, and P. Calhoun. RADIUS Extensions. rfc 2869, IETF, June 2000.

[34] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS). rfc 2865, IETF, June 2000.

[35] W. Simpson. PPP Challenge Handshake Authentication Protocol (CHAP). rfc 1994, IETF, August 1996.

[36] W. Simpson and Ed. The Point-to-Point Protocol (PPP). rfc 1661, IETF, July 1994.

[37] DDW-142 User Guide. Westermo Teleindustri AB, Doc. number 6642-2250X. Wolverine Series, See http://www.westermo.com for updates.

[38] DDW-142-485 User Guide. Westermo Teleindustri AB, Doc. number 6642-2251X. Wolverine Series, See http://www.westermo.com for updates.

[39] DDW-225 User Guide. Westermo Teleindustri AB, Doc. number 6642-2230X. Wolverine Series, See http://www.westermo.com for updates.

[40] DDW-226 User Guide. Westermo Teleindustri AB, Doc. number 6642-2240X. Wolverine Series, See http://www.westermo.com for updates.

[41] Falcon FDV-206-1D1S User Guide. Westermo Teleindustri AB, Doc. number 6660-220X. See http://www.westermo.com for updates.

[42] Lynx L106/206-F2G User Guide. Westermo Teleindustri AB, Doc. number 6643-225X. See http://www.westermo.com for updates.

[43] Lynx DSS L108/208-F2G-S2 User Guide. Westermo Teleindustri AB, Doc. number 6643-222X. See http://www.westermo.com for updates.

[44] Lynx DSS L105/205-S1 User Guide. Westermo Teleindustri AB, Doc. number 6643-223X. See http://www.westermo.com for updates.

[45] Lynx DSS L106/206-S2 User Guide. Westermo Teleindustri AB, Doc. number 6643-224X. See http://www.westermo.com for updates.

[46] Lynx L110/210 User Guide. Westermo Teleindustri AB, Doc. number 6643-221X. See http://www.westermo.com for updates.

[47] RedFox Industrial User Guide. Westermo Teleindustri AB, Doc. number 6641-2230X. RedFox Series, See http://www.westermo.com for updates.

[48] RedFox Industrial User Guide. Westermo Teleindustri AB, Doc. number 6641-2231X. RedFox Series, See http://www.westermo.com for updates.

[49] RedFox Industrial Rack User Guide. Westermo Teleindustri AB, Doc. number 6641-2281X. RedFox Industrial Rack Series, See http://www.westermo.com for updates.

[50] RedFox Rail User Guide. Westermo Teleindustri AB, Doc. number 6641-222X. See http://www.westermo.com for updates.

[51] Viper-12 User Guide. Westermo Teleindustri AB, Doc. number 6641-224X. See http://www.westermo.com for updates.

[52] Viper-12 PoE User Guide. Westermo Teleindustri AB, Doc. number 6641-225X. See http://www.westermo.com for updates.

[53] Westermo Handbook 5.0 - Industrial Data Communication: Theoretical and General Applications. Westermo Teleindustri AB, Available at http://www.westermo.com (Accessed February 2014), 2004.

[54] WeConfig User Guide. Westermo Teleindustri AB, Doc. number 4100-2200X. See http://www.westermo.com for updates.

[55] G. Zorn. Microsoft PPP CHAP Extensions, Version 2. rfc 2759, IETF, January 2000.

[56] G. Zorn and S. Cobb. Microsoft PPP CHAP Extensions. rfc 2433, IETF, October 1998.

Index

