GW2027 and GW2028 Series User Manual

Issue: 1.4
Date: 01 February 2018

1 Introduction

This user manual describes the features and how to configure the Virtual Access GW2027 and GW2028 Series routers.

The GW2027 Series and GW208 Series are a range of versatile 3G/4G LTE/CDMA450 wireless routers suitable for a variety of business and industrial deployments. The compact and rugged structure makes a suitable product for deployments in M2M applications such as CCTV, ATM, telemetry, SCADA, retail (POS), digital signage, and intelligent traffic systems. The product line supports the following radio access technologies: HSPA+, HSPA, UMTS, EDGE, CDMA450, GPRS and GSM.

3G is the third generation of mobile phone standards and technology. It is based on the International Telecommunication Union (ITU) family of standards under the International Mobile Telecommunications programme, IMT-2000.

4G is a mobile communications standard intended to replace 3G, allowing wireless internet access at a much higher speed.

3G and 4G technologies enable network operators to offer users a wider range of more advanced services, while achieving greater network capacity through improved spectral efficiency. Services include wide-area wireless voice telephony, video calls, and broadband wireless data, all in a mobile environment.

1.1 Document scope

This document covers the GW2027 and GW2028 Series routers. For general references, we refer to the GW2028 Series throughout. Feature variations between GW2027 Series and GW2028 Series are described in separate sections.

GW2027 Series router features:

GW2027:	2 x Ethernet, 3G, 4G/LTE, CDMA450, single RS232 and single RS485, Digital I/O **Note**: the second input is either RS232 or RS485 and is specified at time of ordering and fixed in manufacturing.

GW2028 Series router features:

GW2028:	4 x Ethernet, 3G, 4G/LTE, CDMA450,single RS232 and single RS485, Digital I/O **Note**: the second input is either RS232 or RS485 and is specified at time of ordering and fixed in manufacturing.

1.2 Using this documentation

You can configure your router using either the router's web interface or via the command line using UCI commands. Each chapter explains first the web interface settings, followed by how to configure the router using UCI. The web interface screens are shown along with a path to the screen for example, 'In the top menu, select **Service -> SNMP**.' followed by a screen grab.

After the screen grab there is an information table that describes each of the screen's fields.

1.2.1 Information tables

We use information tables to show the different ways to configure the router using the router's web and command line. The left-hand column shows three options:

- **Web:** refers the command on the router's web page,
- **UCI:** shows the specific UCI command, and
- **Opt:** shows the package option.

The right-hand column shows a description field that describes the feature's field or command and shows any options for that feature.

Some features have a drop-down menu and the options are described in a table within the description column. The default value is shown in a grey cell.

Values for enabling and disabling a feature are varied throughout the web interface, for example, 1/0; Yes/No; True/False; check/uncheck a radio button. In the table descriptions, we use **0** to denote Disable and **1** to denote Enable.

Some configuration sections can be defined more than once. An example of this is the routing table where multiple routes can exist and all are named 'route'. For these sections, the UCI command will have a code value [**0**] or [**x**] (where x is the section number) to identify the section.

Web Field / UCI / Package Option	Description
Web: Metric UCI: network.@route[0].metric Opt: metric	Specifies the route metric to use.

Note: these sections can be given a label for identification when using UCI or package options.

```
network.@route[0]=route
network.@route[0].metric=0
```

can be witten as:

```
network.routename=route
network.routename.metric=0
```

However the documentation usually assumes that a section label is not configured.

The table below shows fields from a variety of chapters to illustrate the explanations above.

Web Field / UCI / Package Option	Description
Web: Enable UCI: cesop.main.enable Opt: enable	Enables CESoPSN services. 0 — Disabled. 1 — Enabled.
Web: Syslog Severity UCI: cesop.main.severity Opt: log_severity	Selects the severity used for logging events CESoPSN in syslog. The following levels are available. 0 — Emergency 1 — Alert 2 — Critical 3 — Error 4 — Warning 5 — Notice 6 — Informational 7 — Debug
Web: Agent Address UCI: snmpd.agent[0].agentaddress Opt: agentaddress	Specifies the address(es) and port(s) on which the agent should listen. [(udp|tcp):]port[@address][,...]

Table 1: Example of an information table

1.2.2 Definitions

Throughout the document, we use the host name ‘VA_router’ to cover all router models.

UCI commands and package option examples are shown in the following format:

```
root@VA_router:~# vacmd show current config
```

1.2.3 Diagnostics

Diagnostics are explained at the end of each feature’s chapter.

1.2.4 UCI commands

For detailed information on using UCI commands, read chapters ‘Router File Structure’ and ‘Using Command Line Interface’.

2 GW2027 and GW2028 hardware specification

2.1.1 GW2027 Series router

Figure 1: The GW2027 series router

GW2027: 2 x Ethernet, 3G, 4G/LTE, CDMA450, single RS232 and single RS485, digital I/O, dual SIM, metal case

Note: the second input is either RS232 or RS485 and is software selectable.

2.1.2 GW2028 Series router

Figure 2: The GW2028 series router

GW2028: 4 x Ethernet, 3G, 4G/LTE, CDMA450, single RS232 and single RS485, digital I/O, dual SIM, metal case

Note: the second input is either RS232 or RS485 and is software selectable.

2.2 GW2027 hardware features

- Dual SIM sockets
- Dual antenna SMA connectors
- Two 10/100 Mbps Ethernet ports
- Optional 1 or 2 RS232 ports
- Optional 4KV isolation ports
- Optional RS485 port
- SIM cover

2.3 GW2028 hardware features

- Dual SIM sockets
- Dual antenna SMA connectors
- Up to eight 10/100 Mbps Ethernet ports
- Optional 1 or 2 RS232 ports
- Optional 4KV isolation ports
- Optional RS485 port
- SIM cover

2.4 Serial ports on the GW2027 and GW2028 Series router

The asynchronous serial ports are named:

- Port 0: '/dev/ttySC0'
- Port 1: '/dev/ttySC1'

Each serial port has a number of configurable settings, such as baud rate, word size, parity, flow control mode, etc.

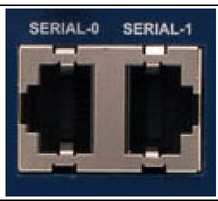


Figure 3: Serial ports on the GW2027 and GW2028 series router

2.4.1.1 RS232 pinout for the GW2027 and GW2028 Series router

Pin	Name	Direction
1	RTS	Out
2	DTR	Out
3	TX Data	Out
4	GND	-
5	GND	-
6	RX Data	In
7	DSR	In
8	CTS	In

2.4.1.2 RS485 pinout for the GW2027 and GW2028 Series router

	Half Duplex Mode		Full Duplex Mode	
Pin	Name	Direction (From GW2027 or GW2028 router)	Pin	Name
1	-	-	1	-
2	-	-	2	-
3	-	-	3	-
4	GND	-	4	GND
5	GND	-	5	GND
6	Tx1/Rx1+	In/Out	6	Tx1/Rx1+
7	Tx1/Rx1-	In/Out	7	Tx1/Rx1-
8	-	-	8	-

2.4.1.3 GPIOs on the GW2027 and GW2028 Series router: digital inputs

Pin	Name	Direction	Description
1	Input 0+	In	Isolated positive input for Digital Input 0
2	Input 0-	In	Isolated negative input for Digital Input 0
3	Input 1+	In	Isolated positive input for Digital Input 1
4	Input 1-	In	Isolated negative input for Digital Input 1
5	5V0	Out	Non isolated 5V supply for Digital Input 0
6	5V1	Out	Non isolated 5V supply for Digital Input 1
7	GND	-	Non isolated ground terminal for Digital Inputs
8	GND	-	Non isolated ground terminal for Digital Inputs

The maximum input voltage for the Digital Inputs is 9V.

The maximum input current for the Digital Inputs is 60 mA.

2.4.1.4 GPIOs on the GW2027 and GW2028 Series router: digital outputs

Pin	Name	Direction	Description
1	Output N/O	-	Digital Output, normally open
2	Output Com	-	Digital Output, common
3	Output N/C	-	Digital Output, normally open

The maximum voltage for the Digital Output is 30V DC.

The maximum current for the Digital Output is 2A.

2.5 GSM and LTE technology

- 4G LTE
- HSPA+
- EDGE/GPRS
- Download up to 21 Mbps
- Upload up to 5.76 Mbps
- 2100/1900/1800/900/850/450 MHz bands

2.6 Power supply

WARNING

Only properly trained service personnel should remove or install power supplies.

Do not touch bare parts inside the enclosure: there may be hazardous energy levels.

The user is responsible for checking equipment ratings, operating instructions and installation instructions before commissioning or maintenance.

The user is responsible for ensuring the equipment is installed, operated and used for its intended function in the manner specified by Virtual Access. Failure to do so may invalidate safety features of the equipment.

2.6.1 Power supply symbols

Symbol	Publication	Description
	IEC 60417-5031 (2002-10)	Direct current
	IEC 60417-5032 (2002-10)	Alternating current
	IEC 60417-5033 (2002-10)	Both direct and alternating current
	IEC 60417-5017 (2006-08)	Earth (ground) terminal

Table 2: power supply symbols

2.7 GW2027 and GW2028 Series router power supply

- 9V-59DC
- Power consumption: 5W

2.8 GW2027 and GW2028 Series router environmental conditions

The following environmental conditions apply to all GW2027 and GW2028 Series routers.

- Rated IP2X when mounted in normal position of use
- Rated pollution degree 2 when mounted in normal position of use
- Rated insulation class III when mounted in normal position of use

2.9 GW2027 Series router dimensions

GW2027 Series unit size:	40W 115D 160H
GW2027 Series unit size:	500g

2.10 GW2028 Series router dimensions

GW2028 Series unit size:	52W 116D 157H
GW2028 Series unit weight:	500g

2.11 GW2027 Series router compliance

The GW2027 Series router is compliant and tested to the following standards:

Safety	EN 60950-1: 2006, + A11:2009 + A1 2010 + A12:2011 + A2:2013
EMC	EN55022 and EN55024 for more specific details please read the GW2027 datasheet.
Environmental	ETSI 300 019-1-3 Sinusoidal Vibration and Shock ETSI 300 019-2-3 Random Vibration.

2.12 GW2028 Series router compliance

The GW2028 Series router is compliant and tested to the following standards:

Safety	EN 60950-1: 2006, + A11:2009 + A1 2010 + A12:2011 + A2:2013
EMC	EN55022 and EN55024 for more specific details please read the GW2028 datasheet.
Environmental	ETSI 300 019-1-3 Sinusoidal Vibration and Shock ETSI 300 019-2-3 Random Vibration.

2.13 Operating temperature range

The operating temperature range depends on the router’s type of power supply.

GW2027	-20°C to 70°C	DIN rail PSU
GW2028	-20°C to 70°C	DIN rail PSU

2.14 Antenna

The GW2027 and GW2028 Series router has two SMA connectors for connection of two antennas for antenna diversity. Antenna diversity helps improve the quality of a wireless link by mitigating problems associated with multipath interference.

2.15 Components

To enable and configure connections on your router, it must be correctly installed.

The GW2027 and GW2028 Series router contains an internal web server that you use for configurations. Before you can access the internal web server and start the configuration, ensure the components are correctly connected and that your PC has the correct networking setup.

2.15.1 GW2028 and GW2027 Series router components

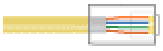
1 x GW2028 or GW2027 Series router (models vary)	
1 x Ethernet cable. RJ45 connector at both ends.	
1 x PSU	
1 x antenna	

Table 3: GW2027 and GW2028 Series router components

2.16 Inserting a SIM card

1. Ensure the unit is powered off.
2. Hold the SIM 1 card with the chip side facing down and the cut corner front left.
3. Gently push the SIM card into SIM slot 1 until it clicks in.
4. If using SIM 2 then hold the SIM with the cut corner front right
5. Gently push the SIM card into SIM slot 2 until it clicks in.

2.17 Connecting the SIM lock

Connect the SIM lock using the Allen key provided.

2.17.1 Connecting cables

Connect one end of the Ethernet cable into port A and the other end to your PC or switch.

2.18 Connecting the antenna

If you are only connecting one antenna, screw the antenna into the MAIN SMA connector.

If you are using two antennas, screw the main antenna into the MAIN SMA connector and the secondary antenna into the AUX SMA connector.

2.19 Mounting the GW2028 Series router on to a DIN rail

The GW2027 and GW2028 Series router is supplied with DIN rail clips attached.

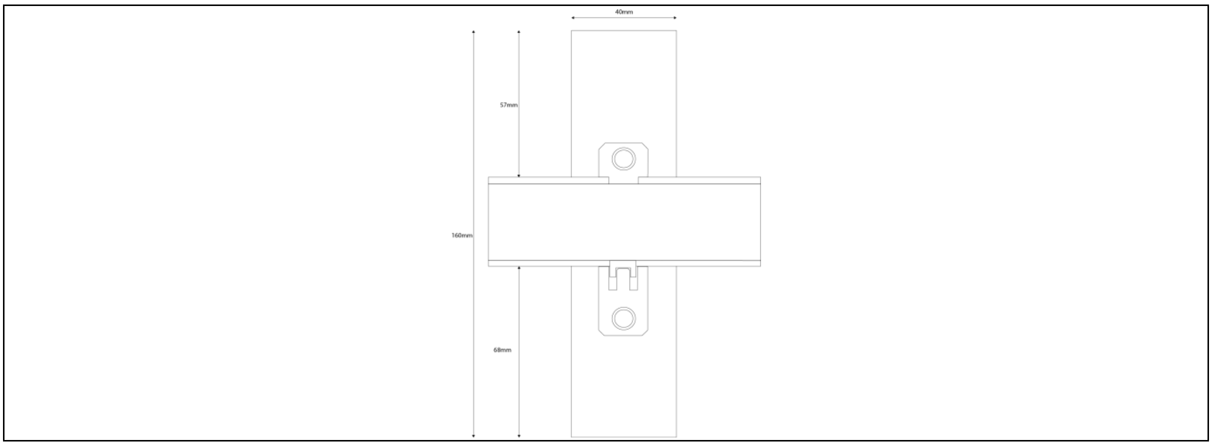


Figure 4: DIN rail position on the rear of a GW2028 series router

1. Offer the unit down on to the DIN rail. Hold the unit at a slight angle and slide the top teeth of the DIN rail clip onto the DIN rail.
2. When the top teeth and DIN rail are connected, push the unit down and back in one motion until the DIN rail clip clicks into the DIN rail.
3. The click sound means the box has engaged on the DIN rail and is secure.

2.20 Releasing the GW2024P Series router from a DIN rail

1. Insert a screwdriver or similar implement into the extender clip on the DIN rail clip and pull up to release the top spring of the DIN rail clip.
2. Pull the unit up and away from the DIN rail to remove it from the DIN rail.

2.21 Powering up the GW2027 or GW2028 Series router

GW2027 and GW2028 Series routers are supplied with a plug in terminal connector or optional external power supply.

1. Wire the 12V DC input to the appropriate DC supply in accordance with local regulations.
2. Plug the terminal connector into the GW2027 or GW2028 Series router.

2.22 Reset button

The reset button is used to request a system reset.

When you press the reset button all LEDs turn on simultaneously. The length of time you hold the reset button will determine its behaviour.

Press Duration	PWR/CONFIG LED behaviour	Router Behaviour on depress
0-3 seconds	On	Normal reset to running config. No special LED activity.
Between 3 and 15 seconds	Flashing slowly	Releasing between 3-15 seconds switches the router back to factory configuration.
Between 15 and 20 seconds	On	Releasing between 15-20 seconds performs a normal reset to running config.
Between 20 seconds and 30 seconds	Flashing faster	Releasing between 20-30 seconds reboots the router in recovery mode.
Over 30 seconds	On	Releasing after 30 seconds performs a normal reset.

Table 4: GW2027 and GW2028 Series router reset behaviour

2.23 Recovery mode

Recovery mode is a fail-safe mode where the router can load a default configuration from the routers firmware. If your router goes into recovery mode, all config files are kept intact. After the next reboot, the router will revert to the previous config file.

You can use recovery mode to manipulate the config files, but should only be used if all other configs files are corrupt. If your router has entered recovery mode, contact your local reseller for access information.

3 GW2027 and GW2028 Series LED behaviour

3.1 Main LED behaviour

The GW2028 Series router has single colour LEDs for Power, Config, SIM1, SIM2 and signal strength. When the router is powered on, the LED is green.

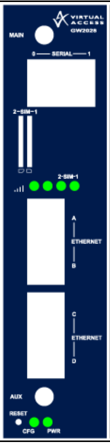


Figure 5: Example of LED activity

The possible LED states are:

- Off
- Flashing slowing
- Flashing quickly
- On

Booting		The GW2028 takes approximately 2 minutes to boot up. During this time, the power LED flashes. Other LEDs display different diagnostic patterns during boot up. Booting is complete when the power LED stops flashing and stays on steady.
Power LED	On	Power connected.
	Off	No power/boot loader does not exist.
Config LED	On	Unit running a valid configuration file.
	Flashing slowly	Unit running in recovery mode (5 Hz).
	Flashing quickly	Unit running in factory configuration (2.5 Hz).
SIM LEDs	On	SIM selected and registered on the network.
	Off	Not selected or SIM not inserted.
	Flashing	SIM selected and not registered on the network.
Signal LEDs	None	PPP not connected or signal strength <= -113dBm.
	Bottom on, top off	Data connection up and signal strength <= -89dBm.
	Bottom off, top on	Data connection up and signal strength between -89dBm and -69dBm.
	Both on	Data connection up and signal strength >-69dBm.

Table 5: LED behaviour and descriptions

Note: when PPP is not connected, none of the signal LEDs will light regardless of signal strength.

3.2 Ethernet port LED behaviour

The Ethernet port has two LEDs: a LINK LED (green) and an ACT LED (amber). When looking at the port, the LED on the top is the LINK LED, and the ACT LED is on the bottom.

Figure 6: Ethernet LED activity

Link LED (green)	Off	No physical Ethernet link detected
	On	Physical Ethernet link detected
ACT LED (amber)	Off	No data is being transmitted/received over the link
	Flashing	Data is being transmitted/ received over the link

4 Factory configuration extraction from SIM card

Virtual Access routers have a feature to update the factory configuration from a SIM card. This allows you to change the factory configuration of a router when installing the SIM.

1. Make sure the SIM card you are inserting has the required configuration written on it.
2. Ensure the router is powered off.
3. Hold the SIM 1 card with the chip side facing down and the cut corner front left.
4. Gently push the SIM card into SIM slot 1 until it clicks in.
5. Power up the router.

Depending on the model, the power LED and/or the configuration LED flash as usual.

The SIM LED starts flashing. This indicates the application responsible for 3G and configuration extraction management is running. It also means the update of the configuration is happening.

When the update is finished, depending on the model, the power LED and/or the configuration LED blink alternatively and very fast for 20 seconds.

Note: factory configuration extraction is only supported on mobile modules that support phone book operations.

5 Accessing the router

Access the router through the web interface or by using SSH. By default, Telnet is disabled.

5.1 Configuration packages used

Package	Sections
dropbear	dropbear
system	main
uhttpd	main cert

5.2 Accessing the router over Ethernet using the web interface

DHCP is disabled by default, so if you do not receive an IP address via DHCP, assign a static IP to the PC that will be connected to the router.

PC IP address	192.168.100.100
Network mask	255.255.255.0
Default gateway	192.168.100.1

Assuming that the PC is connected to Port A on the router, in your internet browser, type in the default local IP address 192.168.100.1, and press **Enter**. The Authorization page appears.

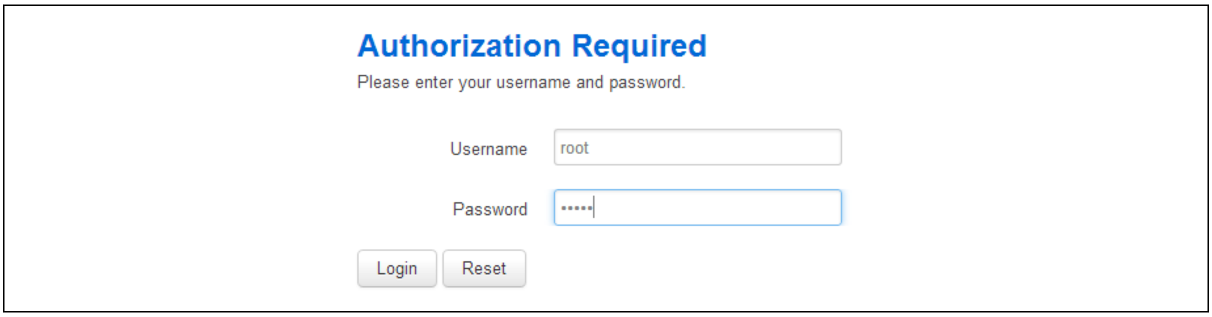


Figure 7: The login page

The password may vary depending on the factory configuration the router has been shipped with. The default settings are shown below. The username and password are case sensitive.

In the username field, type **root**.

In the Password field, type **admin**.

Click **Login**. The Status page appears.

5.3 Accessing the router over Ethernet using an SSH client

You can also access the router over Ethernet, using Secure Shell (SSH) and optionally over Telnet.

To access CLI over Ethernet start an SSH client and connect to the router's management IP address, on port **22: 192.168.100.1/24**.

On the first connection, you may be asked to confirm that you trust the host.

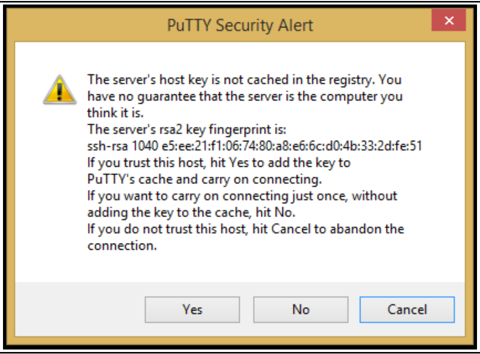


Figure 8: Confirming trust of the routers public key over SSH

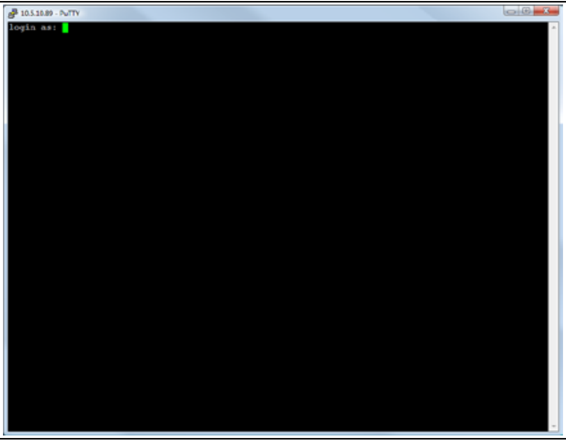


Figure 9: SSH CLI logon screen

In the SSH CLI logon screen, enter the default username and password.

Username: **root**

Password: **admin**

5.3.1 SCP (Secure Copy Protocol)

As part of accessing the router over SSH, you can also use SCP protocol. Use the same user authentication credentials as for SSH access. You can use SCP protocol to securely, manually transfer files from and to the router's SCP server.

No dedicated SPC client is supported; select the SCP client software of your own choice.

5.4 Accessing the router over Ethernet using a Telnet client

Telnet is disabled by default, when you enable Telnet, SSH is disabled.

To enable Telnet, enter:

```
root@VA_router: ~# /etc/init.d/dropbear disable
root@VA_router: ~# reboot -f
```

To re-enable SSH, enter:

```
root@VA_router: ~# /etc/init.d/dropbear enable
root@VA_router: ~# reboot -f
```

Note: As SSH is enabled by default, initial connection to the router to enable Telnet must be established over SSH.

5.5 Configuring the password

5.5.1 Configuration packages used

Package	Sections
system	main

5.6 Configuring the password using the web interface

To change your password, in the top menu click **System -> Administration**. The Administration page appears.

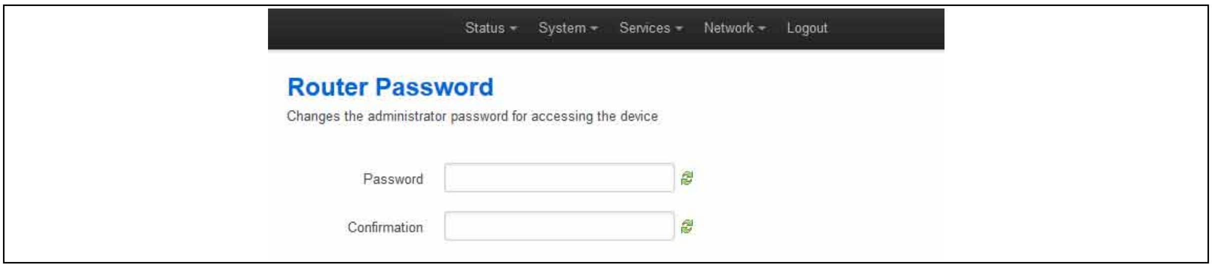


Figure 10: The router password section

In the Router Password section, type your new password in the password field and then retype the password in the confirmation field.

Scroll down the page and click **Save & Apply**.

Note: the username 'root' cannot be changed.

Web Field / UCI / Package Option	Description
Web: Password UCI: system.main.password Opt: password	Defines the root password. The password is displayed encrypted via the CLI using the 'hashpassword' option. UCI: system.main.hashpassword Opt: hashpassword

5.7 Configuring the password using UCI

The root password is displayed encrypted via the CLI using the hashpassword option.

```
root@VA_router:~# uci show system
system.main=system
system.main.hostname=VA_router
system.main.hashpassword=$1$jRX/x8A/$U5kLCMpi9dcahRhOl7eZV1
```

If you are changing the password using UCI, enter the new password in plain text using the password option.

```
root@VA_router:~# uci system.main.password=newpassword
root@VA_router:~# uci commit
```

The new password will take effect after reboot and will now be displayed in encrypted format via the hashpassword option.

5.8 Configuring the password using package options

The root password is displayed encrypted via the CLI using the hashpassword option.

```
root@VA_router:~# uci export system
package system

config system 'main'
      option hostname 'VA_router'
      option hashpassword '$1$wRYYiJOz$EeHN.GQcxXhRgNPVbqxVw
```

If you are changing the password using UCI, enter the new password in plain text using the password option.

```
package system

config system 'main'
      option hostname 'VA_router'
      option hashpassword '$1$wRYYiJOz$EeHN.GQcxXhRgNPVbqxVw
      option password ‘newpassword’
```

The new password will take effect after reboot and will now be displayed in encrypted format via the hashpassword option.

5.9 Accessing the device using RADIUS authentication

You can configure RADIUS authentication to access the router over SSH, web or local console interface.

```
package system

config system 'main'
        option hostname 'VirtualAccess'
        option timezone 'UTC'

config pam_auth
        option enabled 'yes'
        option pamservice 'login'
        option pammodule 'auth'
        option pamcontrol 'sufficient'
        option type 'radius'
        option servers '192.168.0.1:3333|test|20 192.168.2.5|secret|10'

config pam_auth
        option enabled 'yes'
        option pamservice 'sshd'
        option pammodule 'auth'
        option pamcontrol 'sufficient'          it checks package 
management_users
        option type 'radius'
        option servers '192.168.0.1:3333|test|20 192.168.2.5|secret|10'

config 'pam_auth'
        option enabled 'yes'
        option pamservice 'luci"
        option pammodule 'auth'
        option pamcontrol 'sufficient'
        option type 'radius'
        servers '192.168.0.1:3333|test|20 192.168.2.5|secret|10'
```

<table>
<tr><th>UCI / Package Option</th><th>Description</th></tr>
<tr><td>UCI: system.@pam_auth[0].enabled=yes
Opt: enabled</td><td>Enables and disables RADIUS configuration sections.<table><tr><td>yes</td><td>Enables following RADIUS configuration section.</td></tr><tr><td>no</td><td>Disables following RADIUS configuration section.</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamservice
Opt: pamservice</td><td>Selects the method which users should be authenticated by.<table><tr><td>login</td><td>User connecting over console cable.</td></tr><tr><td>sshd</td><td>User connecting over SSH.</td></tr><tr><td>luci</td><td>User connecting over web.</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamcontrol
Opt: pamcontrol</td><td>Specifies authentication behaviour after authentication fails or connection to RADIUS server is broken.<table><tr><td>Sufficient</td><td>First authenticates against remote RADIUS if password authentication fails then it tries local database (user defined in package management_users)</td></tr><tr><td>Required</td><td>If either authentication fails or RADIUS server is not reachable then user is not allowed to access the router.</td></tr><tr><td>[success=done new_authtok_reqd=done authinfo_unavail=ignore default=die]</td><td>Local database is only checked if RADIUS server is not reachable.</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pammodule.auth
Opt: pammodule</td><td>Enables user authentication.</td></tr>
<tr><td>UCI: system.@pam_auth[0].type.radius
Opt: type</td><td>Specifies the authentication method.</td></tr>
<tr><td>UCI: system.@pam_auth[0].servers
Opt: servers</td><td>Specifies the RADIUS server or multiple servers along with port number and password. The example below explains the syntax.
192.168.0.1:3333|test|20 192.168.2.5|secret|10</td></tr>
</table>

Table 6: Information table for RADIUS authentication

5.10 Accessing the device using TACACS+ authentication

TACACS+ authentication can be configured for accessing the router over SSH, web or local console interface.

```
package system

config system 'main'
        option hostname 'VirtualAccess'
        option timezone 'UTC'

config pam_auth
        option enabled 'yes'
        option pamservice 'sshd'
```

```
        option pammodule 'auth'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'

config pam_auth
        option enabled 'yes'
        option pamservice 'sshd'
        option pammodule 'account'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'
        option args 'service=ppp'

config pam_auth
        option enabled 'yes'
        option pamservice 'sshd'
        option pammodule 'session'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'
        option args 'service=ppp'

config pam_auth
        option enabled 'yes'
        option pamservice 'luci'
        option pammodule 'auth'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'

config pam_auth
        option enabled 'yes'
        option pamservice 'luci'
        option pammodule 'account'
        option pamcontrol 'sufficient'
        option type 'tacplus'
```

```
        option servers '192.168.0.1:49|secret'
        option args 'service=ppp'

config pam_auth
        option enabled 'yes'
        option pamservice 'luci'
        option pammodule 'session'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'
        option args 'service=ppp'
config pam_auth
        option enabled 'yes'
        option pamservice 'login'
        option pammodule 'auth'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'

config pam_auth
        option enabled 'yes'
        option pamservice 'login'
        option pammodule 'account'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'
        option args 'service=ppp'

config pam_auth
        option enabled 'yes'
        option pamservice 'login'
        option pammodule 'session'
        option pamcontrol 'sufficient'
        option type 'tacplus'
        option servers '192.168.0.1:49|secret'
        option args 'service=ppp'
```

<table>
<tr><th>UCI / Package Option</th><th>Description</th></tr>
<tr><td>UCI: system.@pam_auth[0].enabled=yes
Opt: enabled</td><td>Enables and disables TACACS configuration sections.<table><tr><td>yes</td><td>Enables following TACACS configuration section.</td></tr><tr><td>no</td><td>Disables following TACACS configuration section.</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamservice
Opt: pamservice</td><td>Selects the method which users should be authenticated by.<table><tr><td>login</td><td>User connecting over console cable.</td></tr><tr><td>sshd</td><td>User connecting over SSH.</td></tr><tr><td>luci</td><td>User connecting over web.</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamcontrol
Opt: pamcontrol</td><td>Specifies authentication behaviour after authentication fails or connection to TACACS server is broken.<table><tr><td>Sufficient</td><td>First authenticates against remote TACACS if password authentication fails then it tries local database (user defined in package management_users)</td></tr><tr><td>Required</td><td>If either authentication fails or TACACS server is not reachable then user is not allowed to access the router.</td></tr><tr><td>[success=done new_authtok_reqd=done authinfo_unavail=ignore default=die]</td><td>Local database is only checked if TACACS server is not reachable.</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pammodule.auth
Opt: pammodule</td><td>Selects which TACACS module this part of configuration relates to.<table><tr><td>auth</td><td>auth module provides the actual authentication and sets credentials</td></tr><tr><td>account</td><td>account module checks to make sure that access is allowed for the user</td></tr><tr><td>session</td><td>session module performs additional tasks which are needed to allow access</td></tr></table></td></tr>
<tr><td>system.@pam_auth[0].type=tacplus
Opt: type</td><td>Specifies the authentication method.</td></tr>
<tr><td>UCI: system.@pam_auth[0].servers
Opt: servers</td><td>Specifies the TACACS servers along with port number and password. The example below explains the syntax.
192.168.0.1:49|secret '</td></tr>
<tr><td>UCI: system.@pam_auth[1].args=service=ppp
Opt: args</td><td>Additional arguments to pass to TACACS serer.</td></tr>
</table>

Table7: Information table for TACACS authentication

5.11 SSH

SSH allows you to access remote machines over text-based shell sessions. SSH uses public key cryptography to create a secure connection. These connections allow you to issue commands remotely via a command line.

The router uses a package called Dropbear to configure the SSH server on the box. You can configure Dropbear via the web interface or through an SSH connection by editing the file stored on: /etc/config_name/dropbear.

5.11.1 Configuration packages used

Package	Sections
dropbear	dropbear

5.11.2 SSH access using the web interface

In the top menu, click **System -> Administration**. The Administration page appears. Scroll down to the SSH Access section.

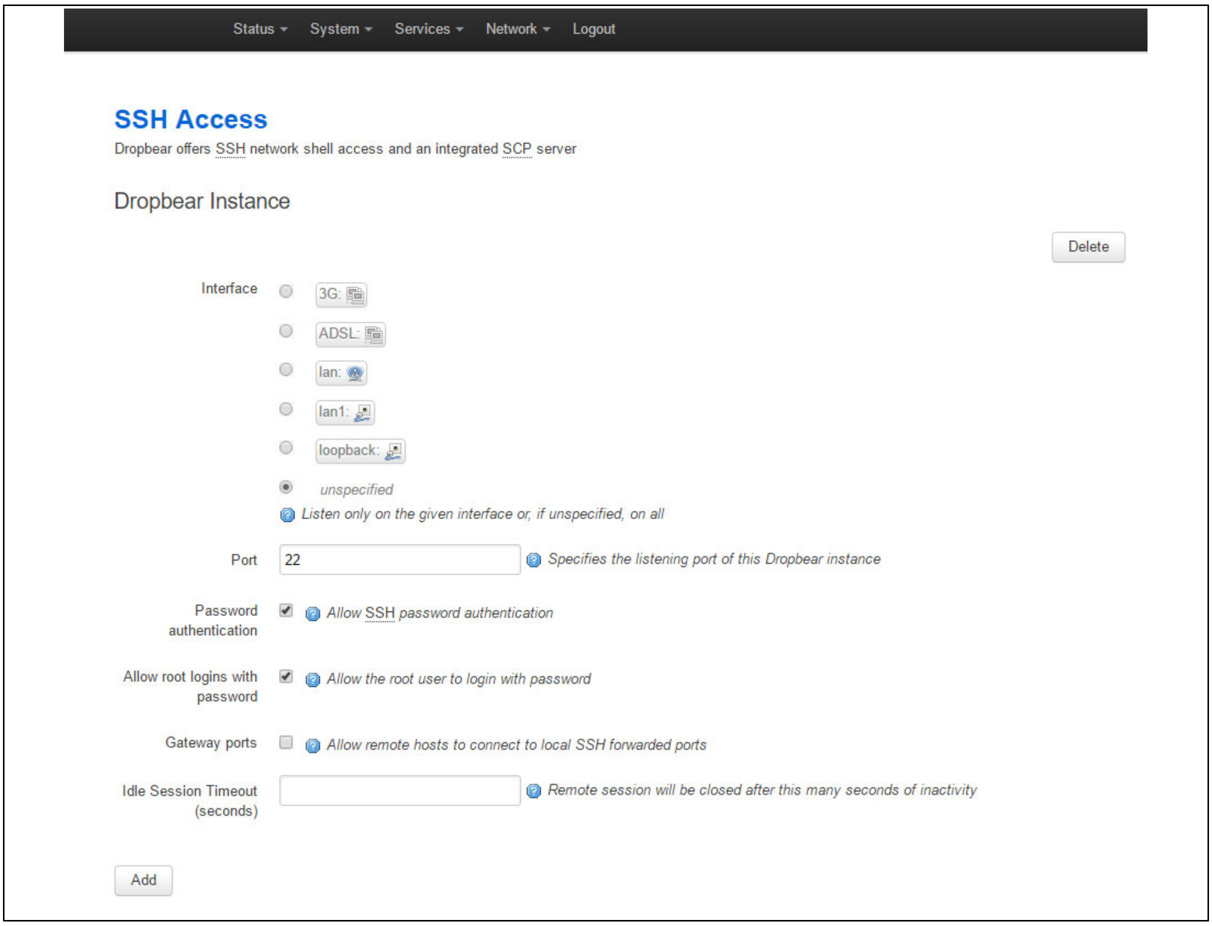


Figure 11: The SSH access section

Web Field / UCI / Package Option	Description
Basic settings	
Web: Interface UCI: dropbear.@dropbear[0].Interface Opt: interface	Listens only on the selected interface. If unspecified is checked, listens on all interfaces. All configured interfaces will be displayed via the web GUI. (unspecified): listens on all interfaces. Range: Configured interface names.
Web: Port UCI: dropbear.@dropbear[0].Port Opt: port	Specifies the listening port of the Dropbear instance. 22 Range: 0-65535
Web: Password authentication UCI: dropbear.@dropbear[0].PasswordAuth Opt: PasswordAuth	If enabled, allows SSH password authentication. 0: Disabled. 1: Enabled.
Web: Allow root logins with password UCI: dropbear.@dropbear[0].RootPasswordAuth Opt: RootPasswordAuth	Allows the root user to login with password. 0: Disabled. 1: Enabled.
Web: Gateway ports UCI: dropbear.@dropbear[0].GatewayPorts Opt: GatewayPorts	Allows remote hosts to connect to local SSH forwarded ports. 0: Disabled. 1: Enabled.
Web: Idle Session Timeout UCI: dropbear.@dropbear[0].IdleTimeout Opt: IdleTimeout	Defines the idle period where remote session will be closed after the allocated number of seconds of inactivity. 30: 30 seconds. Range
Web: n/a UCI: dropbear.@dropbear[0]. BannerFile Opt: BannerFile	Defines a banner file to be displayed during login. /etc/banner Range
Web: n/a UCI: dropbear.@dropbear[0].MaxLoginAttempts Opt: MaxLoginAttempts	Specifies maximum login failures before session terminates. 10 0-infinite

Table 7: Information table for SSH access settings

5.12 Package dropbear using UCI

```
root@VA_router:~# uci show dropbear
dropbear.@dropbear[0]=dropbear
dropbear.@dropbear[0].PasswordAuth=on
dropbear.@dropbear[0].RootPasswordAuth=on
dropbear.@dropbear[0].GatewayPorts=0
dropbear.@dropbear[0].IdleTimeout=30
dropbear.@dropbear[0].Port=22
dropbear.@dropbear[0].MaxLoginAttempts=3
Package dropbear using package options
```

```
root@VA_router:~# uci export dropbear
package dropbear
config dropbear'
        option PasswordAuth 'on'
        option RootPasswordAuth 'on'
        option Port '22'
        option GatewayPorts ‘0’
        option IdleTimeout ‘30’
        option MaxLoginAttempts '3'
```

5.13 Certs and private keys

Certificates are used to prove ownership of a public key. They contain information about the key, its owner’s ID, and the digital signature of an individual that has verified the content of the certificate.

In asymmetric cryptography, public keys are announced to the public, and a different private key is kept by the receiver. The public key is used to encrypt the message, and the private key is used to decrypt it.

To access certs and private keys, in the top menu, click **System -> Administration**. The Administration page appears. Scroll down to the Certs & Private Keys section.

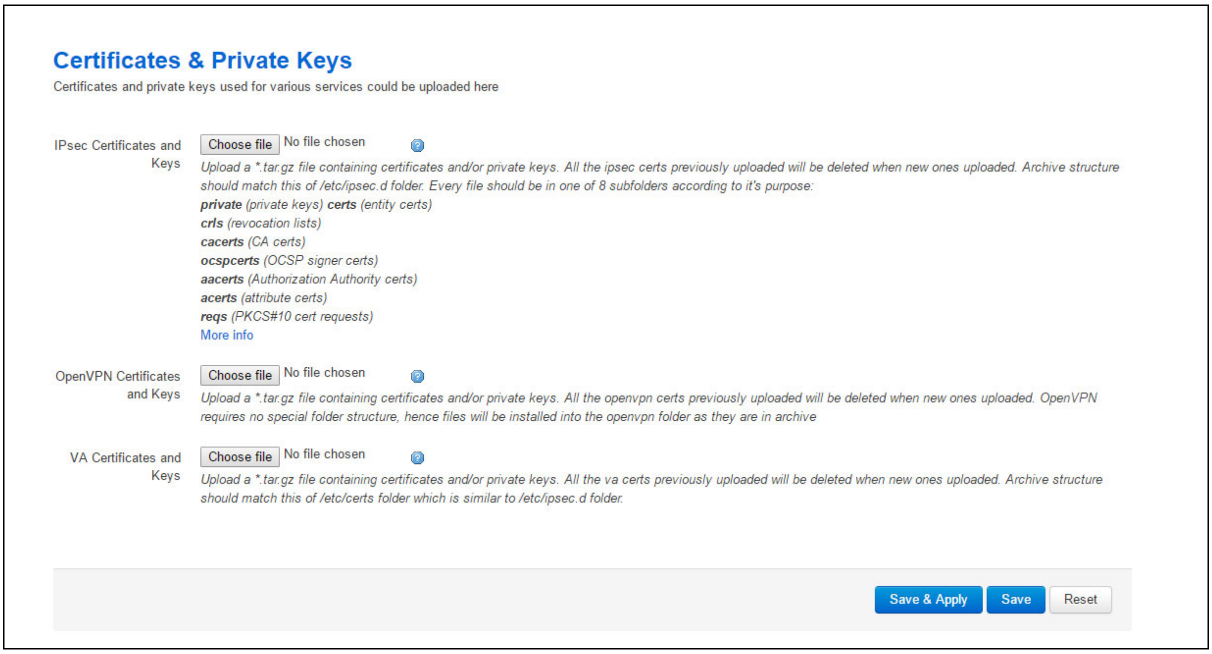


Figure 12: The certificates & private keys section

This section allows you to upload any certificates and keys that you may have stored. There is support for IPSec, OpenVPN and VA certificates and keys.

If you have generated your own SSH public keys, you can input them in the SSH Keys section, for SSH public key authentication.

Figure 13: The SSH-keys box

5.14 Configuring a router's web server

The router's web server is configured in package uhttpd. This file defines the behaviour of the server and default values for certificates generated for SSL operation. uhttpd supports multiple instances, that is, multiple listen ports, each with its own document root and other features, as well as cgi and lua. There are two sections defined:

Main: this uHTTPd section contains general server settings.

Cert: this section defines the default values for SSL certificates.

5.14.1 Configuration packages used

Package	**Sections**
uhttpd	main
	cert

To configure the router's HTTP server parameters, in the top menu, select **Services -> HTTP Server**. The HTTP Server page has two sections.

Main Settings	Server configurations
Certificate Settings	SSL certificates.

5.14.2 Main settings

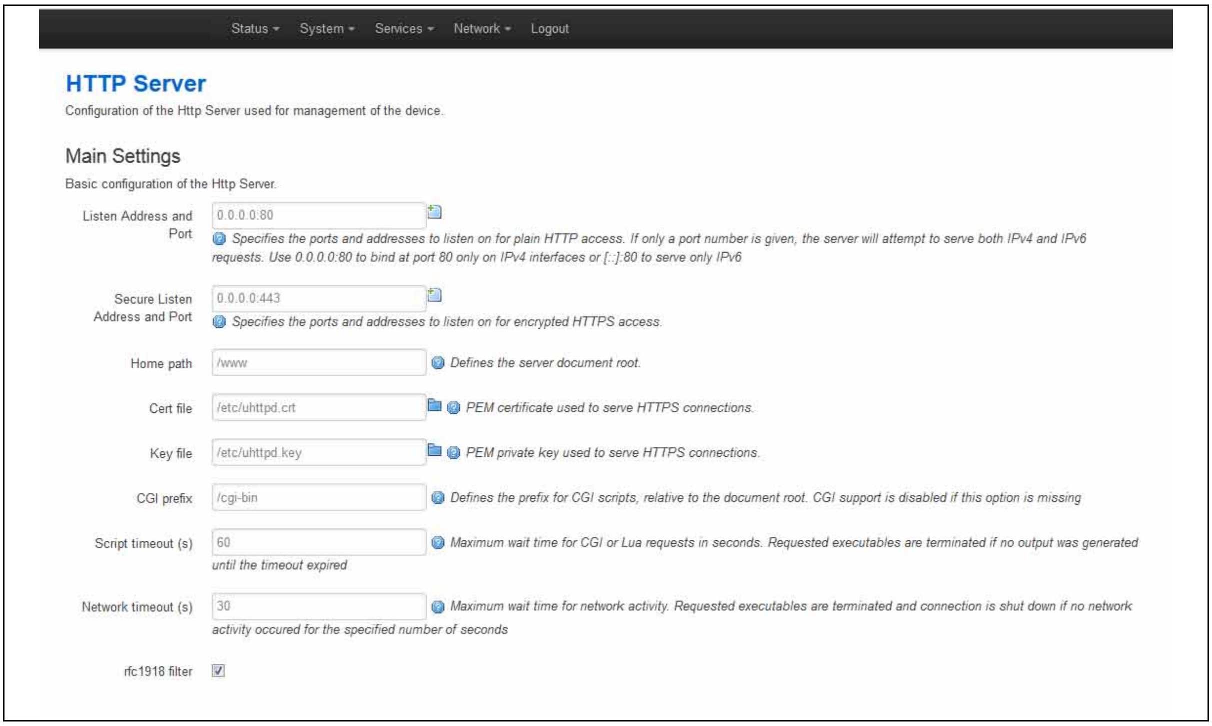


Figure 14: HTTP server settings

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Listen Address and Port
UCI: uhttpd.main.listen_http
Opt: list listen_http</td><td>Specifies the ports and addresses to listen on for plain HTTP access. If only a port number is given, the server will attempt to serve both IPv4 and IPv6 requests.
<table>
<tr><td>0.0.0.0:80</td><td>Bind at port 80 only on IPv4 interfaces.</td></tr>
<tr><td>[::]:80</td><td>Bind at port 80 only on IPv6 interfaces</td></tr>
<tr><td>Range</td><td>IP address and/or port</td></tr>
</table></td></tr>
<tr><td>Web: Secure Listen Address and Port
UCI: uhttpd.main.listen_https
Opt: list listen_https</td><td>Specifies the ports and address to listen on for encrypted HTTPS access. The format is the same as listen_http.
<table>
<tr><td>0.0.0.0:443</td><td>Bind at port 443 only</td></tr>
<tr><td>[::]:443</td><td></td></tr>
<tr><td>Range</td><td>IP address and/or port</td></tr>
</table></td></tr>
<tr><td>Web: Home path
UCI: uhttpd.main.home
Opt: home</td><td>Defines the server document root.
<table>
<tr><td>/www</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Cert file
UCI: uhttpd.main.cert
Opt: cert</td><td>ASN.1/DER certificate used to serve HTTPS connections. If no listen_https options are given the key options are ignored.
<table>
<tr><td>/etc/uhttpd.crt</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Key file
UCI: uhttpd.main.key
Opt: key</td><td>ASN.1/DER private key used to serve HTTPS connections. If no listen_https options are given the key options are ignored.
<table>
<tr><td>/etc/uhttpd.key</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
</table>

Web: CGI profile UCI: uhttpd.main.cgi_prefix Opt: cgi_prefix	Defines the prefix for CGI scripts, relative to the document root. CGI support is disabled if this option is missing. /cgi-bin | Range |
Web: N/A UCI: uhttpd.main.lua_prefix Opt: lua_prefix	Defines the prefix for dispatching requests to the embedded lua interpreter, relative to the document root. Lua support is disabled if this option is missing. /luci | Range |
Web: N/A UCI: uhttpd.main.lua_handler Opt: lua_handler	Specifies the lua handler script used to initialise the lua runtime on server start. /usr/lib/lua/luci/sgi/uhttpd.lua | Range |
Web: Script timeout UCI: uhttpd.main.script_timeout Opt: script_timeout	Sets the maximum wait time for CGI or lua requests in seconds. Requested executables are terminated if no output was generated. 60 | Range |
Web: Network timeout UCI: uhttpd.main.network_timeout Opt: network_timeout	Maximum wait time for network activity. Requested executables are terminated and connection is shut down if no network activity occured for the specified number of seconds. 30 | Range |
Web: N/A UCI: uhttpd.main.realm Opt: realm	Defines basic authentication realm when prompting the client for credentials (HTTP 400). OpenWrt | Range |
Web: N/A UCI: uhttpd.main.config Opt: config	Config file in Busybox httpd format for additional settings. Currently only used to specify basic auth areas. /etc/http.conf | Range |
Web: N/A UCI: uhttpd.main.index_page Opt: index_page	Index file to use for directories, for example, add index.php when using php. | Range |
Web: N/A UCI: httpd.main.error_page Opt: error_page	Virtual URL of file of CGI script to handle 404 requests. Must begin with '/' (forward slash). | Range |
Web: N/A UCI: uhttpd.main.no_symlinks Opt: no_symlinks	Does not follow symbolic links if enabled. 0 | Disabled. 1 | Enabled.
Web: N/A UCI: uhttpd.main.no_dirlists Opt: no_symlinks	Does not generate directory listings if enabled. 0 | Disabled. 1 | Enabled.

Web: rfc 1918 filter UCI: uhttpd.main.rfc1918_filter=1 Opt: rfc1918_filter	Enables option to reject requests from RFC1918 IPs to public server IPs (DNS rebinding counter measure). 0: Disabled. 1: Enabled.

Table 8: Information table for http server basic settings

5.14.3 HTTP server using UCI

Multiple sections of the type uhttpd may exist. The init script will launch one webserver instance per section.

A standard uhttpd configuration is shown below.

```
root@VA_router:~# uci show uhttpd
uhttpd.main=uhttpd
uhttpd.main.listen_http=0.0.0.0:80
uhttpd.main.listen_https=0.0.0.0:443
uhttpd.main.home=/www
uhttpd.main.rfc1918_filter=1
uhttpd.main.cert=/etc/uhttpd.crt
uhttpd.main.key=/etc/uhttpd.key
uhttpd.main.cgi_prefix=/cgi-bin
uhttpd.main.script_timeout=60
uhttpd.main.network_timeout=30
uhttpd.main.config=/etc/http.conf
HTTP server using package options
root@VA_router:~# uci export dropbear
config uhttpd 'main'
        list listen_http '0.0.0.0:80'
        list listen_https '0.0.0.0:443'
        option home '/www'
        option rfc1918_filter '1'
        option cert '/etc/uhttpd.crt'
        option key '/etc/uhttpd.key'
        option cgi_prefix '/cgi-bin'
        option script_timeout '60'
        option network_timeout '30'
        option config '/etc/http.conf'
```

5.14.4 HTTPs server certificate settings

To configure HTTPs server certificate settings, in the top menu, select **Services -> HTTP Server**. Scroll down to the Certificate Settings section.

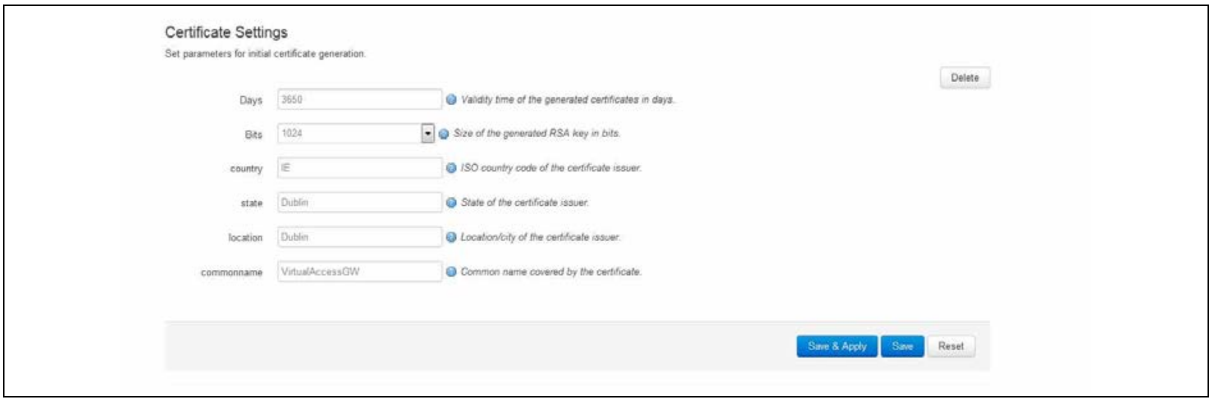


Figure 15: HTTP server certificate settings

Web Field/UCI/Package Option	Description
Web: Days UCI: uhttpd.px5g.days Opt: days	Validity time of the generated certificates in days. 730 Range
Web: Bits UCI: uhttpd.px5g.bits Opt: bits	Size of the generated RSA key in bits. 1024 Range
Web: Country UCI: uhttpd.px5g.country Opt: country	ISO code of the certificate issuer.
Web: State UCI: uhttpd.px5g.state Opt: state	State of the certificate issuer.
Web: Location UCI: uhttpd.px5g.location Opt: location	Location or city of the certificate user.
Web: Commonname UCI: uhttpd.commonname Opt: commonname	Common name covered by the certificate. For the purposes of secure Activation, this must be set to the serial number (Eth0 MAC address) of the device.

Table 9: Information table for HTTP server certificate settings

5.14.5 HTTPs server using UCI

```
root@VA_router:~# uci show uhttpd.px5g
uhttpd.px5g=cert
uhttpd.px5g.days=3650
uhttpd.px5g.bits=1024
uhttpd.px5g.country=IE
uhttpd.px5g.state=Dublin
uhttpd.px5g.location=Dublin
uhttpd.px5g.commonname=00E0C8000000
HTTPs server using package options
root@VA_router:~# uci export uhttpd
package uhttpdconfig 'cert' 'px5g'
      option 'days' '3650'
      option 'bits' '1024'
      option 'state' 'Dublin'

      option 'location' 'Dublin'
      option 'commonname' '00E0C8000000'
```

5.15 Basic authentication (httpd conf)

For backward compatibility reasons, uhttpd uses the file /etc/httpd.conf to define authentication areas and the associated usernames and passwords. This configuration file is not in UCI format.

Authentication realms are defined in the format prefix:username:password with one entry and a line break.

Prefix is the URL part covered by the realm, for example, cgi-bin to request basic auth for any CGI program.

Username specifies the username a client has to login with.

Password defines the secret password required to authenticate.

The password can be either in plain text format, MD5 encoded or in the form $p$user where the user refers to an account in /etc/shadow or /etc/passwd.

If you use p… format, uhttpd will compare the client provided password against the one stored in the shadow or passwd database.

5.16 Securing uhttpd

By default, uhttpd binds to 0.0.0.0 which also includes the WAN port of your router. To bind uhttpd to the LAN port only you have to change the listen_http and listen_https options to your LAN IP address.

To get your current LAN IP address, enter:

```
uci get network.lan.ipaddr
```

Then modify the configuration appropriately:

```
uci set uhttpd.main.listen_http='192.168.1.1:80'
uci set uhttpd.main.listen_https='192.168.1.1:443'

config 'uhttpd' 'main'
        list listen_http        192.168.1.1:80
        list listen_https       192.168.1.1:443
```

5.17 Displaying custom information via login screen

The login screen, by default, shows the hostname of the router in addition to the username and password prompt. However, the router can be configured to show some other basic information if required using a UDS script.

Note: this can only be configured via the command line.

5.17.1 Configuration packages used

Package	Sections
luci	main
uds	script

5.17.2 Configuring login screen custom information

The luci package option `login_page_info_template` is configured with the path to a UDS script that would render the required information on the right side of the login page.

The following example shows how to display serial number and mobile signal strength.

Note: this can only be configured via the command line.

Figure 16: Example login screen displaying serial and signal strength

5.17.2.1 Login screen custom information using UCI

```
root@VA_router:~# uci show luci
luci.main=core
luci.main.login_page_info_template=/tmp/uds/sysauth_template

root@VA_router:~# uci show uds
uds.sysauth_template=script
uds.sysauth_template.enabled=1
uds.sysauth_template.exec_type=none
uds.sysauth_template.fname=sysauth_template.htm
uds.sysauth_template.type=none
uds.sysauth_template.text=Serial: <%=pcdata(luci.version.serial)%><br/> <%
local sig = luci.dispatcher.uci.cursor_state():get("mobile", "3g_1_1",
"sig_dbm") or -113 sig = tonumber(sig) local hue = (sig + 113) * 2 local
hue = math.min(math.max(hue, 0), 120) %> Signal strength: <h3
style="color:hsl(<%=hue%>, 90%, 50%); display:inline;"><%=sig%></h3> dBm
```

5.17.2.2 Login screen custom information using package options

```
root@VA_router:~# uci export luci
package luci
config core ‘main’
      option login_page_info_template ‘/tmp/uds/sysauth_template’
root@VA_router:~# uci export uds
package uds
config script 'sysauth_template'
        option enabled '1'
        option exec_type 'none'
        option fname 'sysauth_template.htm'
        option type 'none'
        list text 'Serial: <%=pcdata(luci.version.serial)%><br/>'
        list text '<% local sig =
luci.dispatcher.uci.cursor_state():get("mobile", "3g_1_1", "sig_dbm") or -
113'
        list text 'sig = tonumber(sig)'
        list text 'local hue = (sig + 113) * 2'
        list text 'local hue = math.min(math.max(hue, 0), 120) %>'
        list text 'Signal strength: <h3 style="color:hsl(<%=hue%>, 90%,
50%); display:inline;"><%=sig%></h3> dBm
```

6 Router file structure

This section describes the file structure and location of essential directories and files on Virtual Access routers.

Throughout this document, we use information tables to show the different ways to configure the router using the router's web interface and command line interface (CLI).

When showing examples of the command line interface we use the host name 'VA_router' to indicate the system prompt. For example, the table below displays what the user should see when entering the command to show the current configuration in use on the router:

```
root@VA_router:~# va_config.sh
```

6.1 System information

General information about software and configuration used by the router is displayed on the Status page. To view the running configuration file status on the web interface, in the top menu, select **Status -> Overview**. This page also appears immediately after you have logged in.

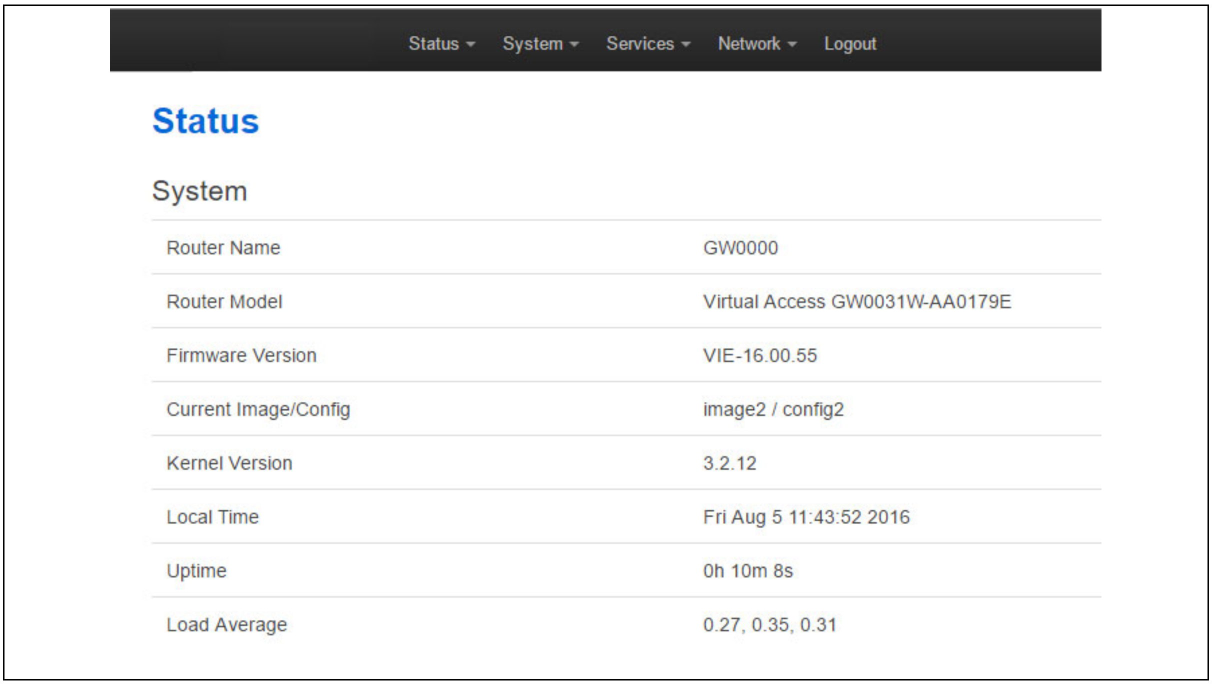


Figure 17: Example of the status page

System information is also available from the CLI if you enter the following command:

```
root@VA_router:~# va_vars.sh
```

The example below shows the output from the above command.

```
VA_SERIAL:              00E0C8121215
VA_MODEL:               GW0000
VA_ACTIVEIMAGE:         image2
VA_ACTIVECONFIG:        config1
VA_IMAGE1VER:           VIE-16.00.44
VA_IMAGE2VER:           VIE-16.00.44
```

6.2 Identify your software version

To check which software version your router is running, in the top menu, browse to **Status -> Overview**.

Figure 18: The status page showing a software version prior to 72.002

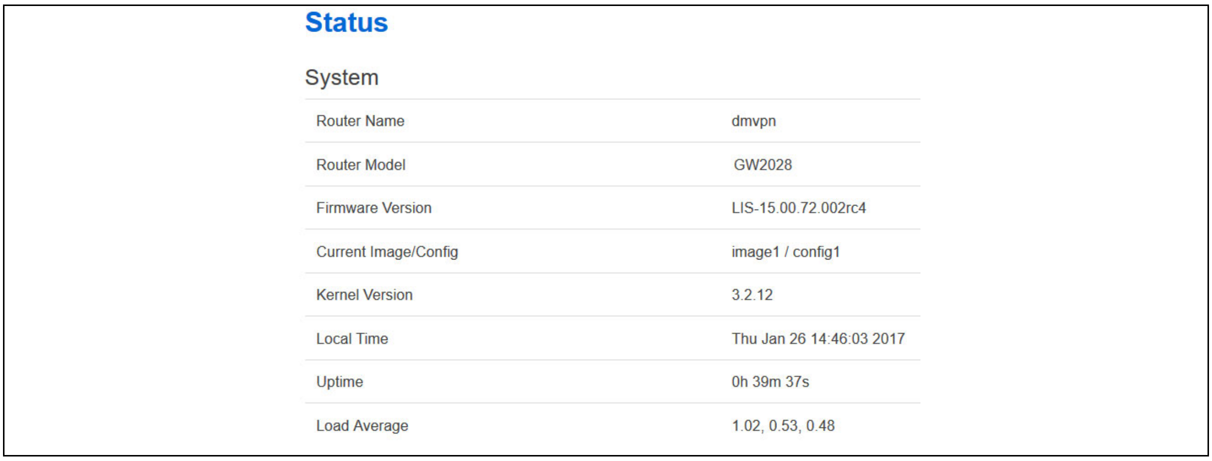


Figure 19: The status page showing software version 72.002

In the Firmware Version row, the first two digits of the firmware version identify the hardware platform, for example LIS-15; while the remaining digits: .00.72.002, show the software version.

6.3 Image files

The system allows for two firmware image files:

- image1, and
- image2

Two firmware images are supported to enable the system to rollback to a previous firmware version if the upgrade of one image fails.

The image names (image1, image2) themselves are symbols that point to different partitions in the overall file system. A special image name “altimage” exists which always points to the image that is not running.

The firmware upgrade system always downloads firmware to “altimage”.

6.4 Directory locations for UCI configuration files

Router configurations files are stored in folders on:

- /etc/factconf,
- /etc/config1, and
- /etc/config2

Multiple configuration files exist in each folder. Each configuration file contains configuration parameters for different areas of functionality in the system.

A symbolic link exists at /etc/config, which always points to one of factconf, config1 or config2 is the active configuration file.

Files that appear to be in /etc/config are actually in /etc/factconf|config1|config2 depending on which configuration is active.

If /etc/config is missing on start-up, for example on first boot, the links and directories are created with configuration files copied from /rom/etc/config/.

At any given time, only one of the configurations is the active configuration. The UCI system tool (Unified Configuration Interface) only acts upon the currently active configuration.

6.5 Viewing and changing current configuration

To show the configuration currently running, enter:

```
root@VA_router:~# va_config.sh
```

To show the configuration to run after the next reboot, enter:

```
root@VA_router:~# va_config.sh next
```

To set the configuration to run after the next reboot, enter:

```
root@VA_router:~# va_config.sh -s [factconf|config1|config2|altconfig]
```

6.6 Configuration file syntax

The configuration files consist of sections – or packages - that contain one or more config statements. These optional statements define actual values.

Below is an example of a simple configuration file.

```
package 'example'
config 'example' 'test'
        option   'string'       'some value'
        option   'boolean'      '1'
        list     'collection'   'first item'
        list     'collection'   'second item'
```

The config 'example' 'test' statement defines the start of a section with the type example and the name test.

Command	**Target**	**Description**
export	[<config>]	Exports the configuration in a machine readable format. It is used internally to evaluate configuration files as shell scripts.
import	[<config>]	Imports configuration files in UCI syntax.
add	<config> <section-type>	Adds an anonymous section of type-section type to the given configuration.
add_list	<config>.<section>.<option>=<string>	Adds the given string to an existing list option.
show	[<config>[.<section>[.<option>]]]	Shows the given option, section or configuration in compressed notation.
get	<config>.<section>[.<option>]	Gets the value of the given option or the type of the given section.
Set	<config>.<section>[.<option>]=<value>	Sets the value of the given option, or adds a new section with the type set to the given value.
delete	<config>[.<section[.<option>]]	Deletes the given section or option.

Table 1: Common commands, target and their descriptions

6.7 Managing configurations

6.7.1 Managing sets of configuration files using directory manipulation

Configurations can also be managed using directory manipulation.

To remove the contents of the current folder, enter:

```
root@VA_router:/etc/config1# rm –f *
```

Warning: the above command makes irreversible changes.

To remove the contents of a specific folder regardless of the current folder (config2), enter:

```
root@VA_router:/ # rm –f /etc/config1/*
```

Warning: the above command makes irreversible changes.

To copy the contents of one folder into another (config2 into config1), enter:

```
root@VA_router:/etc/config1# cp /etc/config2/* /etc/config1
```

6.8 Exporting a configuration file

If you have software versions prior to 72.002, to export a configuration file using the web interface, go to section 6.8.1.

If you have software version 72.002 or above, export a configuration file using the web interface go to section 6.8.2.

To export a configuration file using CLI, for any software version, go to section 6.8.3.

6.8.1 Exporting a configuration file using the web interface for software versions pre- 72.002

The current running configuration file may be exported using the web interface.

In the top menu, select **System > Backup/Flash Firmware**. The Flash operations page appears.

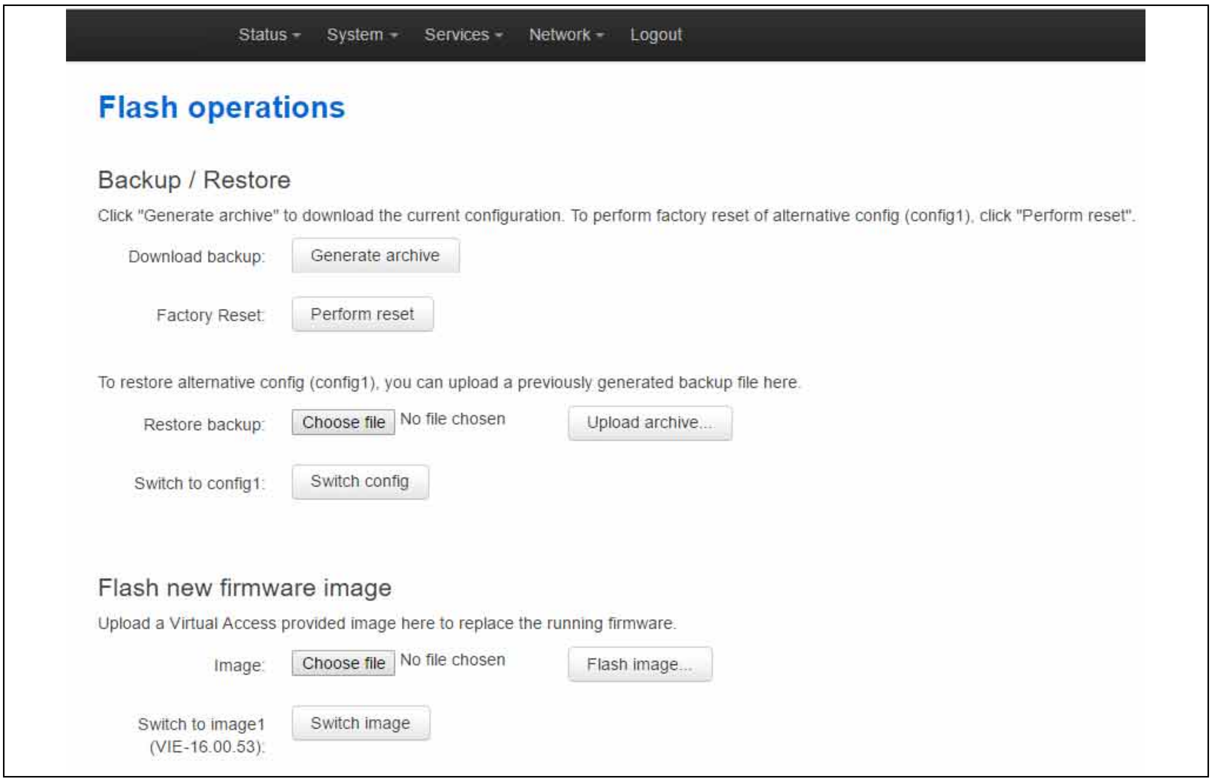


Figure 20: The flash operations page

In the Backup/Restore section, select **Generate Archive**.

6.8.2 Exporting a configuration file using the web interface for software version 72.002 and above

The current running configuration file may be exported using the web interface.

In the top menu, select **System > Flash Operations**. The Flash operations page appears.

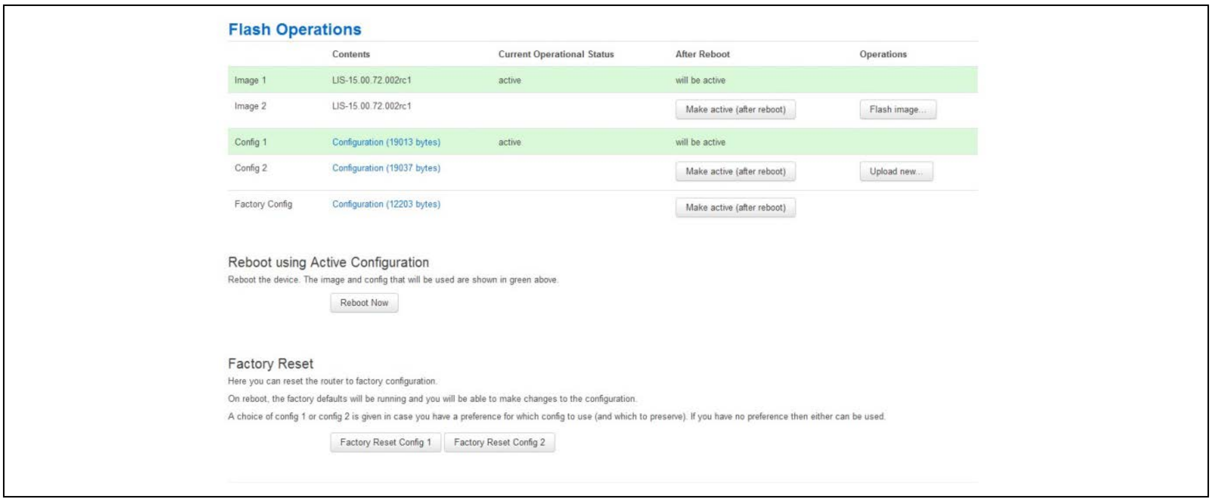


Figure 21: The flash operations page

In the **Flash Operation** section, click the configuration file in the Contents column to download it.

6.8.3 Exporting a configuration file using UCI

You can view any configuration file segment using UCI.

To export the running configuration file, enter:

```
root@VA_router:~# uci export
```

To export the factory configuration file, enter:

```
root@VA_router:~# uci –c /etc/factconf/ export
```

To export config1 or config2 configuration file, enter:

```
root@VA_router:~# uci –c /etc/config1/ export
root@VA_router:~# uci –c /etc/config2/ export
```

6.9 Importing a configuration file

If you have software versions prior to 72.002, to export a configuration file using the web interface, go to section 6.9.1.

If you have software version 72.002 or above, export a configuration file using the web interface go to section 6.9.2.

To export a configuration file using CLI, for any software version, go to section 6.9.3.

6.9.1 Importing a configuration file using the web interface for software versions pre- 72.002

You can import a configuration file to the alternate configuration segment using the web interface. This will automatically reboot the router into this configuration file.

In the top menu, select **System > Backup/Flash Firmware**. The Flash operations page appears.

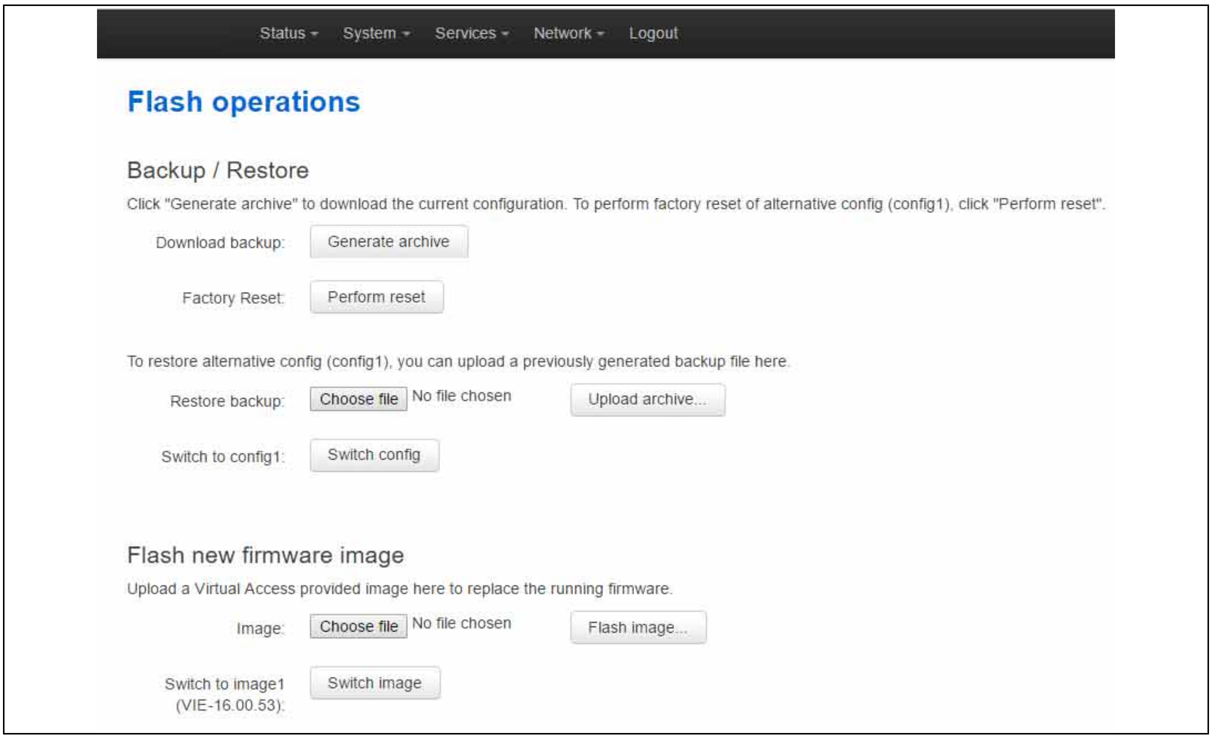


Figure 22: The flash operations page

Under Backup/Restore, choose **Restore Backup: Choose file**. Select the appropriate file and then click **Upload archive**.

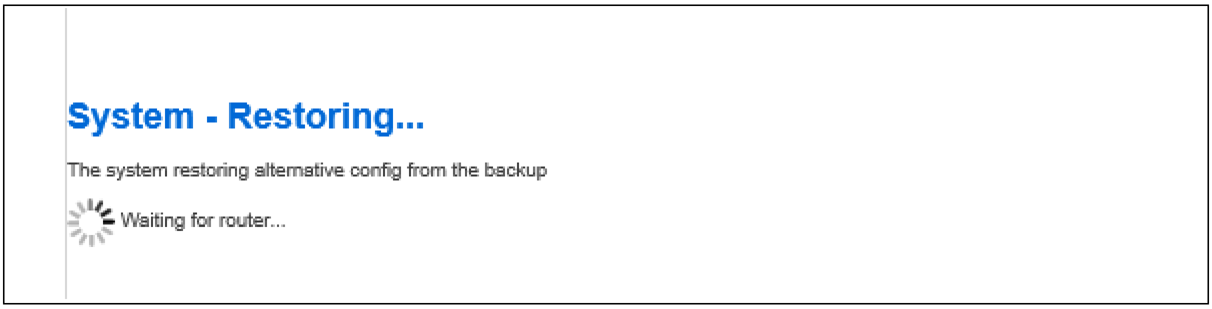


Figure 23: The system – restoring…page

When the 'waiting for router' icon disappears, the upgrade is complete, and the login homepage appears.

6.9.2 Importing a configuration file using the web interface for software version 72.002 and above

You can import a configuration file to the alternate configuration segment using the web interface.

In the top menu, select **System > Flash Operations**. The Flash operations page appears.

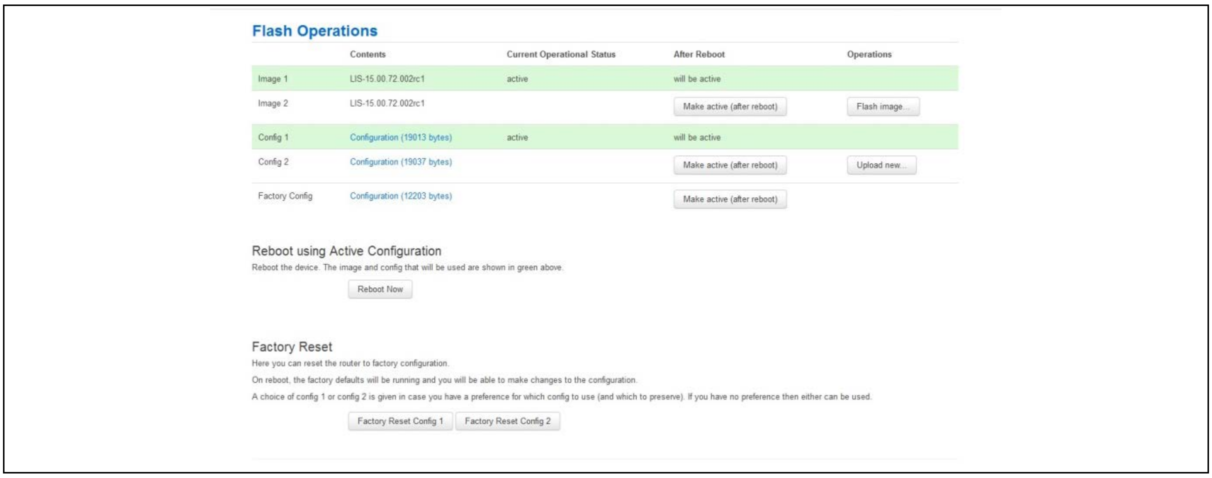


Figure 24: The flash operations page

In the Operations column, click **Upload new**. Select the appropriate file.

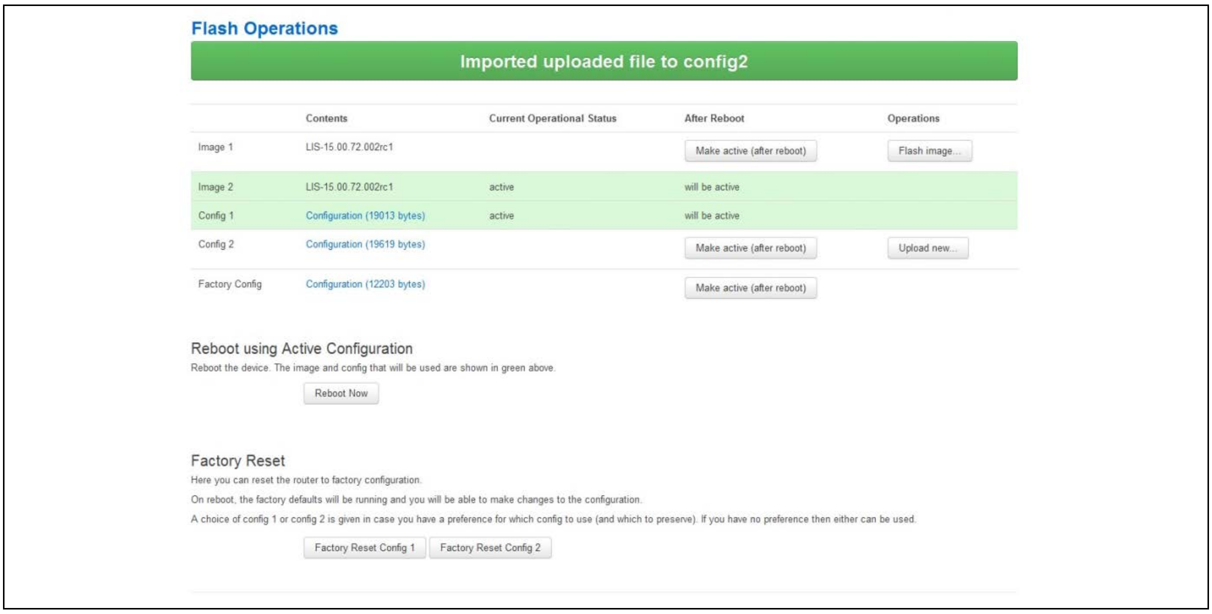


Figure 25: The flash operations succeed upload configuration page

If you select 'Flash image and do not reboot', the router will only run this configuration if you click **OK** to return to the Flash Operations page. There you can manually select **Made Active (after reboot)**. Then click **Reboot Now** in the 'Reboot using Active Configuration' section.

6.9.3 Importing a configuration file using UCI

You can import a configuration file to any file segment using UCI.

To import to config1, enter:

```
root@VA_router:~# uci –c /etc/config1/ import
<paste in config file>
<CTRL-D>
```

Note: it is very important that the config file is in the correct format otherwise it will not import correctly.

7 Using the Command Line Interface

This chapter explains how to view Virtual Access routers' log files and edit configuration files using a Command Line Interface (CLI) and the Unified Configuration Interface (UCI) system. Some commands may vary between router models.

7.1 Overview of some common commands

Virtual Access routers' system has an SSH server typically running on port 22.

The factconf default password for the root user is **admin**.

To change the factconf default password, enter:

```
root@VA_router:/# uci set system.main.password=”******”

root@VA_router:/# uci commit system
```

To reboot the system, enter:

```
root@VA_router:/# reboot
```

The system provides a Unix-like command line. Common Unix commands are available such as `ls, cd, cat, top, grep, tail, head, more` and `less`.

Typical pipe and redirect operators are also available, such as: `>, >>, <, |`

The system log can be viewed using any of the following commands:

```
root@VA_router:/# logread


root@VA_router:/# logread | tail


root@VA_router:/# logread –f
```

These commands will show the full log, end of the log (tail) and continuously (-f). Enter **Ctrl-C** to stop the continuous output from logread -f.

To view and edit configuration files, the system uses the Unified Configuration Interface (UCI) which is described further on in this chapter. This is the preferred method of editing configuration files. However, you can also view and edit these files using some of the standard Unix tools.

For example, to view a text or configuration file in the system, enter:

```
root@VA_router:/# cat /etc/passwd
```

The command output information shows the following, or similar output.

```
root:x:0:0:root:/root:/bin/ash
daemon:*:1:1:daemon:/var:/bin/false
ftp:*:55:55:ftp:/home/ftp:/bin/false
sftp:*:56:56:sftp:/var:/usr/lib/sftp-server
network:*:101:101:network:/var:/bin/false
nobody:*:65534:65534:nobody:/var:/bin/false
```

To view files in the current folder, enter:

```
root@VA_router:/# ls


 bin      etc      lib      opt      sbin     usr
 bkrepos  home     linuxrc  proc     sys      var
 dev      init     mnt      root     tmp      www
```

For more details add the -l argument:

```
root@VA_router:/# ls -l


drwxrwxr-x    2 root     root   642 Jul 16  2012 bin
drwxr-xr-x    5 root     root  1020 Jul  4 01:27 dev
drwxrwxr-x    1 root     root     0 Jul  3 18:41 etc
drwxr-xr-x    1 root     root     0 Jul  9  2012 lib
drwxr-xr-x    2 root     root     3 Jul 16  2012 mnt
drwxr-xr-x    7 root     root     0 Jan  1  1970 overlay
dr-xr-xr-x   58 root     root     0 Jan  1  1970 proc
drwxr-xr-x   16 root     root   223 Jul 16  2012 rom
drwxr-xr-x    1 root     root     0 Jul  3 22:53 root
drwxrwxr-x    2 root     root   612 Jul 16  2012 sbin
drwxr-xr-x   11 root     root     0 Jan  1  1970 sys
drwxrwxrwt   10 root     root   300 Jul  4 01:27 tmp
drwxr-xr-x    1 root     root     0 Jul  3 11:37 usr
lrwxrwxrwx    1 root     root     4 Jul 16  2012 var -> /tmp
drwxr-xr-x    4 root     root    67 Jul 16  2012 www
```

To change the current folder, enter **cd** followed by the desired path:

```
root@VA_router:/# cd /etc/config1
root@VA_router:/etc/config1#
```

Note: if the specified directory is actually a link to a directory, the real directory will be shown in the prompt.

To view scheduled jobs, enter:

```
root@VA_router:/# crontab –l


 0 * * * * slaupload 00FF5FF92752 TFTP 1 172.16.250.100 69
```

To view currently running processes, enter:

```
root@VA_router:/# ps


 PID  Uid     VmSize Stat Command
    1 root       356 S   init
    2 root           DW  [keventd]
    3 root           RWN [ksoftirqd_CPU0]
    4 root           SW  [kswapd]
    5 root           SW  [bdflush]
    6 root           SW  [kupdated]
    8 root           SW  [mtdblockd]
   89 root       344 S   logger -s -p 6 -t
   92 root       356 S   init
   93 root       348 S   syslogd -C 16
   94 root       300 S   klogd
  424 root       320 S   wifi up
  549 root       364 S   httpd -p 80 -h /www -r VA_router
  563 root       336 S   crond -c /etc/crontabs
 6712 root       392 S   /usr/sbin/dropbear
 6824 root       588 S   /usr/sbin/dropbear
 7296 root       444 S   -ash
  374 root       344 R   ps ax
  375 root       400 S   /bin/sh /sbin/hotplug button
  384 root       396 R   /bin/sh /sbin/hotplug button
  385 root           RW  [keventd]
```

To search for a process, enter: pgrep -fl '<process name or part of name>':

```
root@VA_router:/# pgrep -fl ‘wifi’


424 root        320 S   wifi up
```

To kill a process, enter the PID:

```
root@VA_router:~# kill 424
```

7.2 Using Unified Configuration Interface (UCI)

The system uses Unified Configuration Interface (UCI) for central configuration management. Most common and useful configuration settings can be accessed and configured using the UCI system.

UCI consists of a Command Line Utility (CLI), the files containing the actual configuration data, and scripts that take the configuration data and apply it to the proper parts of the system, such as the networking interfaces. Entering the command 'uci' on its own will display the list of valid arguments for the command and their format.

```
root@VA_router:/lib/config# uci
```

Usage: uci [<options>] <command> [<arguments>]

```
Commands:
export     [<config>]
import     [<config>]
changes    [<config>]
commit     [<config>]
add        <config> <section-type>
add_list   <config>.<section>.<option>=<string>
show       [<config>[.<section>[.<option>]]]
get        <config>.<section>[.<option>]
set        <config>.<section>[.<option>]=<value>
delete     <config>[.<section[.<option>]]
rename     <config>.<section>[.<option>]=<name>
revert     <config>[.<section>[.<option>]]
Options:
-c <path>  set the search path for config files (default: /etc/config)
-d <str>   set the delimiter for list values in uci show
-f <file>  use <file> as input instead of stdin
-m         when importing, merge data into an existing package
```

```
-n          name unnamed sections on export (default)
-N          don't name unnamed sections
-p <path>   add a search path for config change files
-P <path>   add a search path for config change files and use as default
-q          quiet mode (don't print error messages)
-s          force strict mode (stop on parser errors, default)

-S          disable strict mode
-X          do not use extended syntax on 'show'
```

The table below describes commands for the UCI command line and some further examples of how to use this utility.

Command	Target	Description
commit	[<config>]	Writes changes of the given configuration file, or if none is given, all configuration files, to the filesystem. All "uci set", "uci add", "uci rename" and "uci delete" commands are staged into a temporary location and written to flash at once with "uci commit". This is not needed after editing configuration files with a text editor, but for scripts, GUIs and other programs working directly with UCI files.
export	[<config>]	Exports the configuration in a UCI syntax and does validation.
import	[<config>]	Imports configuration files in UCI syntax.
changes	[<config>]	Lists staged changes to the given configuration file or if none given, all configuration files.
add	<config> <section-type>	Adds an anonymous section of type section-type to the given configuration.
add_list	<config>.<section>.<option>=<string>	Adds the given string to an existing list option.
show	[<config>[.<section>[.<option>]]]	Shows the given option, section or configuration in compressed notation.
get	<config>.<section>[.<option>]	Gets the value of the given option or the type of the given section.
set	<config>.<section>[.<option>]=<value>	Sets the value of the given option, or add a new section with the type set to the given value.
delete	<config>[.<section[.<option>]]	Deletes the given section or option.
rename	<config>.<section>[.<option>]=<name>	Renames the given option or section to the given name.
revert	<config>[.<section>[.<option>]]	Deletes staged changes to the given option, section or configuration file.

Table 10: Common commands, target and their descriptions

Note: all operations do not act directly on the configuration files. A commit command is required after you have finished your configuration.

```
root@VA_router:~# uci commit
```

7.2.1 Using uci commit to avoid router reboot

After changing the port, uhttpd listens on from 80 to 8080 in the file /etc/config/uhttpd; save it, then enter:

```
root@VA_router:~# uci commit uhttpd
```

Then enter:

```
root@VA_router:~# /etc/init.d/uhttpd restart
```

For this example, the router does not need to reboot as the changes take effect when the specified process is restarted.

7.2.2 Export a configuration

Using the uci export command it is possible to view the entire configuration of the router or a specific package. Using this method to view configurations does not show comments that are present in the configuration file:

```
root@VA_router:~# uci export httpd


package 'httpd'
config 'httpd'
option 'port' '80'
option 'home' '/www'
```

7.2.3 Show a configuration tree

The configuration tree format displays the full path to each option. This path can then be used to edit a specific option using the `uci set` command.

To show the configuration 'tree' for a given config, enter:

```
root@VA_router:/# uci show network


network.loopback=interface
network.loopback.ifname=lo
network.loopback.proto=static
network.loopback.ipaddr=127.0.0.1
network.loopback.netmask=255.0.0.0
network.lan=interface
network.lan.ifname=eth0
network.lan.proto=dhcp
network.wan=interface
network.wan.username=foo
```

```
network.wan.password=bar
network.wan.proto=3g
network.wan.device=/dev/ttyACM0
network.wan.service=umts
network.wan.auto=0
network.wan.apn=arkessa.com
network.@va_switch[0]=va_switch
network.@va_switch[0].eth0=A B C
network.@va_switch[0].eth1=D
```

It is also possible to display a limited subset of a configuration:

```
root@VA_router:/# uci show network.wan
network.wan=interface
network.wan.username=foo
network.wan.password=bar
network.wan.proto=3g
network.wan.device=/dev/ttyACM0
network.wan.service=umts
network.wan.auto=0
network.wan.apn=hs.vodafone.ie
```

7.2.4 Display just the value of an option

To display a specific value of an individual option within a package, enter:

```
root@VA_router:~# uci get httpd.@httpd[0].port
80
root@VA_router:~#
```

7.2.5 High level image commands

To show the image running currently, enter:

```
root@VA_router:~# vacmd show current image
```

To set the image to run on next reboot, enter:

```
root@VA_router:~# vacmd set next image [image1|image2|altimage]
root@VA_router:~# reboot
```

7.2.6 Format of multiple rules

When there are multiple rules next to each other, UCI uses array-like references for them. For example, if there are 8 NTP servers, UCI will let you reference their sections as `timeserver.@timeserver[0]` for the first section; or `timeserver.@timeserver[7]` for the last section.

You can also use negative indexes, such as `timeserver.@timeserver[-1]` '-1' means the last one, and '-2' means the second-to-last one. This is useful when appending new rules to the end of a list.

```
root@VA_router:/# uci show va_eventd
va_eventd.main=va_eventd
va_eventd.main.enabled=yes
va_eventd.main.event_queue_file=/tmp/event_buffer
va_eventd.main.event_queue_size=128K
va_eventd.@conn_tester[0]=conn_tester
va_eventd.@conn_tester[0].name=Pinger
va_eventd.@conn_tester[0].enabled=yes
va_eventd.@conn_tester[0].type=ping
va_eventd.@conn_tester[0].ping_dest_addr=192.168.250.100
va_eventd.@conn_tester[0].ping_success_duration_sec=5
va_eventd.@target[0]=target
va_eventd.@target[0].name=MonitorSyslog
va_eventd.@target[0].enabled=yes
va_eventd.@target[0].type=syslog
va_eventd.@target[0].target_addr=192.168.250.100
va_eventd.@target[0].conn_tester=Pinger
va_eventd.@target[0].suppress_duplicate_forwardings=no
va_eventd.@forwarding[0]=forwarding
va_eventd.@forwarding[0].enabled=yes
va_eventd.@forwarding[0].className=ethernet
va_eventd.@forwarding[0].target=MonitorSyslog
va_eventd.@forwarding[1]=forwarding
va_eventd.@forwarding[1].enabled=yes
va_eventd.@forwarding[1].className=auth
va_eventd.@forwarding[1].target=MonitorSyslog
va_eventd.@forwarding[2]=forwarding
va_eventd.@forwarding[2].enabled=yes
va_eventd.@forwarding[2].className=adsl
```

```
va_eventd.@forwarding[2].target=MonitorSyslog
va_eventd.@forwarding[3]=forwarding
va_eventd.@forwarding[3].enabled=yes
va_eventd.@forwarding[3].className=ppp
va_eventd.@forwarding[3].target=MonitorSyslog
```

7.3 Configuration files

The table below lists common package configuration files that can be edited using uci commands. Other configuration files may also be present depending on the specific options available on the Virtual Access router.

File	Description
Management	
/etc/config/autoload	Boot up Activation behaviour (typically used in factconf)
/etc/config/httpclient	Activator addresses and urls
/etc/config/monitor	Monitor details
Basic	
/etc/config/dropbear	SSH server options
/etc/config/dhcp	Dnsmasq configuration and DHCP settings
/etc/config/firewall	NAT, packet filter, port forwarding, etc.
/etc/config/network	Switch, interface, L2TP and route configuration
/etc/config/system	Misc. system settings including syslog
Other	
/etc/config/snmpd	SNMPd settings
/etc/config/uhttpd	Web server options (uHTTPd)
/etc/config/strongswan	IPSec settings

7.4 Configuration file syntax

The configuration files usually consist of one or more config statements, so-called sections with one or more option statements defining the actual values.

Below is an example of a simple configuration file.

```
package 'example'
config 'example' 'test'
        option   'string'      'some value'
        option   'boolean'     '1'
        list     'collection'  'first item'
        list     'collection'  'second item'
```

The config `'example' 'test'` statement defines the start of a section with the type example and the name test. There can also be so-called anonymous sections with only a type, but no name identifier. The type is important for the processing programs to decide how to treat the enclosed options.

The option `'string' 'some value'` and option `'boolean' '1'` lines define simple values within the section.

Note: there are no syntactical differences between text and boolean options. Per convention, boolean options may have one of the values '`0`', 'no', '`off`' or '`false`' to specify a false value or '`1`' , '`yes`', 'on' or '`true`' to specify a true value.

In the lines starting with a list keyword, an option with multiple values is defined. All list statements that share the same name collection in our example will be combined into a single list of values with the same order as in the configuration file.

The indentation of the option and list statements is a convention to improve the readability of the configuration file but it is not syntactically required.

Usually you do not need to enclose identifiers or values in quotes. Quotes are only required if the enclosed value contains spaces or tabs. Also it is legal to use double-quotes instead of single-quotes when typing configuration options.

All of the examples below are valid syntax.

```
option example value
option 'example' value
option example "value"
option "example"    'value'
option   'example' "value"
```

In contrast, the following examples are not valid syntax.

```
option 'example" "value'
```

Quotes are unbalanced.

```
option example some value with space
```

Missing quotes around the value.

It is important to note that identifiers and config file names may only contain the characters a-z, A-Z, 0-9 and _. However, option values may contain any character, as long they are properly quoted.

8 Upgrading router firmware

This chapter describes how to upgrade router firmware. The upgrade process is as follows:

- Firmware is transferred to the device.
- Firmware is checked to ensure there are no corruptions.
- Firmware is saved to persistent storage.
- Data in persistent storage is validated.

To avoid any unrecoverable errors during the process, you must follow several safety steps described in this chapter.

On successful completion of the process, you can restart the device running the new firmware.

8.1 Software versions

If you have software versions prior to 72.002, to upgrade firmware using the web interface, go to section 8.1.2.

If you have software version 72.002 or above, to upgrade firmware using the web interface go to section 8.1.3.

To upgrade firmware using CLI, for any software version, go to section 8.1.4.

8.1.1 Identify your software version

To check which software version your router is running, in the top menu, browse to **Status -> Overview**.

Figure 26: The status page showing a software version prior to 72.002

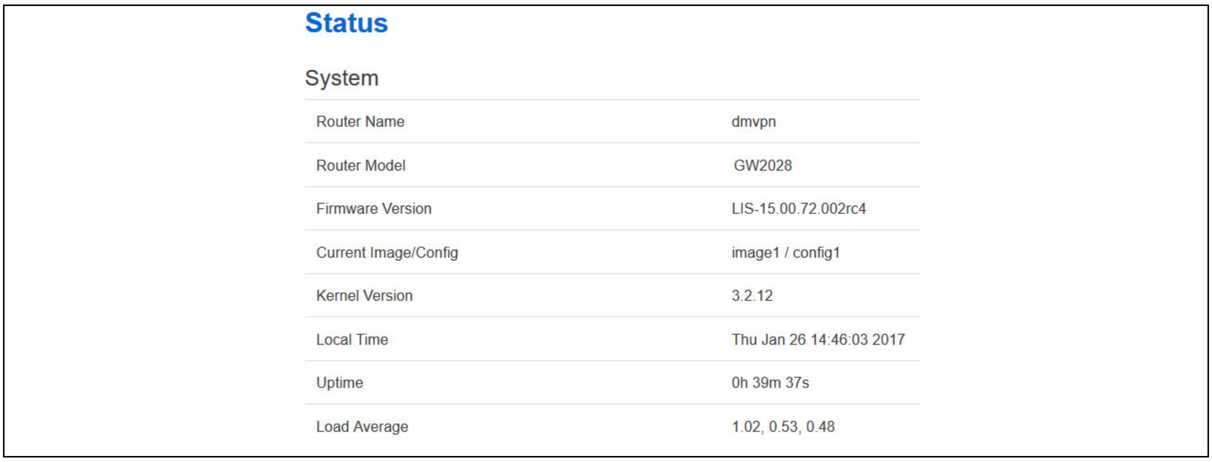


Figure 27: The status page showing software version 72.002

In the Firmware Version row, the first two digits of the firmware version identify the hardware platform, for example LIS-15; while the remaining digits: .00.72.002, show the software version.

8.1.2 Upgrading router firmware for software versions pre- 72.002

Copy the new firmware issued by Virtual Access to a PC connected to the router.

In the top menu, select **System tab -> Backup/Flash Firmware**. The Flash operations page appears.

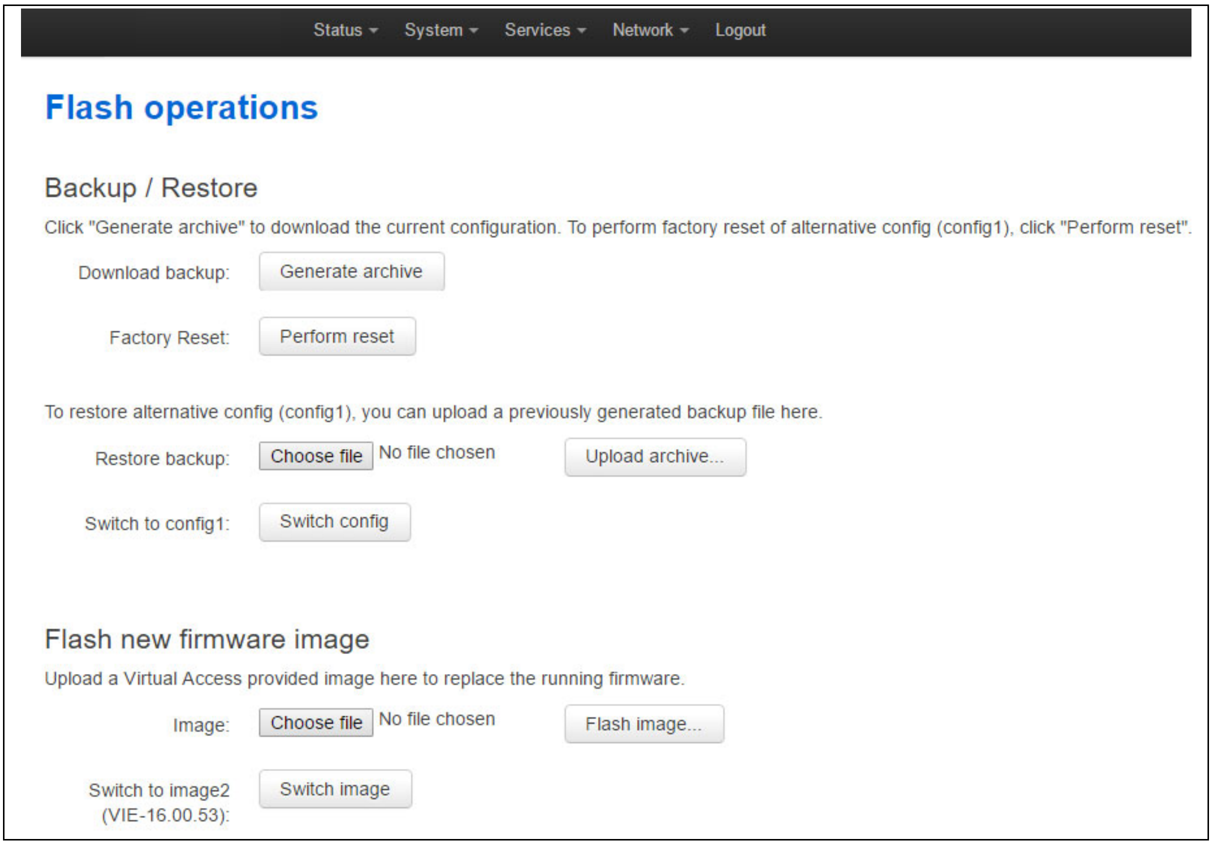


Figure 28: The flash operations page

Under Flash new firmware image, click **Choose File** or **Browse**.

Note: the button will vary depending on the browser you are using.

Select the appropriate image and then click **Flash Image**. The Flash Firmware – Verify page appears.

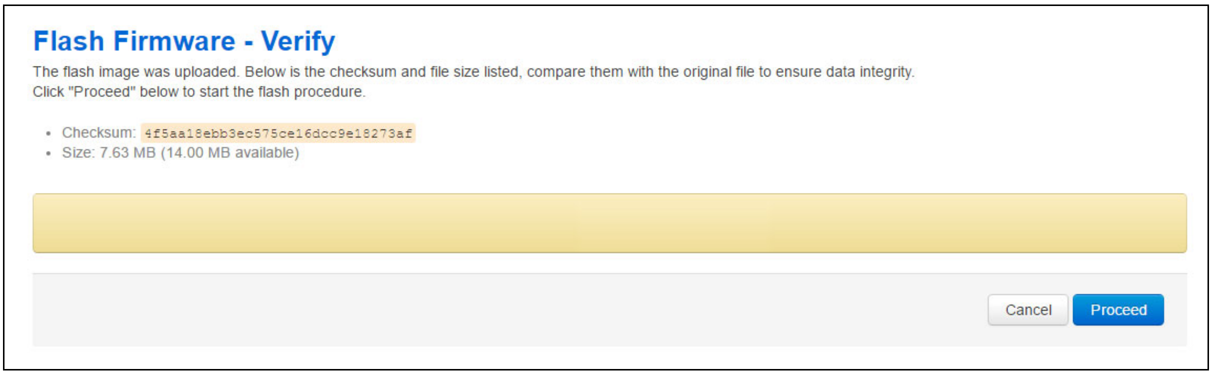


Figure 29: The flash firmware - verify page

Click **Proceed**. The System – Flashing… page appears.

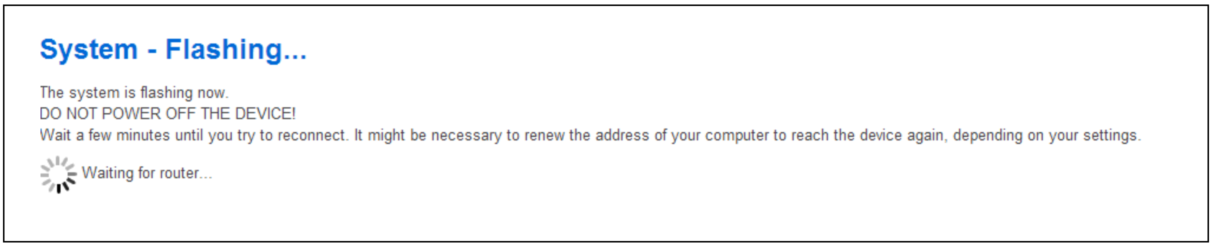


Figure 30: The system – flashing…page

When the 'waiting for router' icon disappears, the upgrade is complete, and the login homepage appears.

To verify that the router has been upgraded successfully, click **Status** in the top menu. The Firmware Version shows in the system list.

Figure 31: The system status list

8.1.3 Upgrading router firmware for software version 72.002 and above

Copy the new firmware issued by Virtual Access to a PC connected to the router.

In the top menu, select **System tab > Flash operations**. The Flash operations page appears.

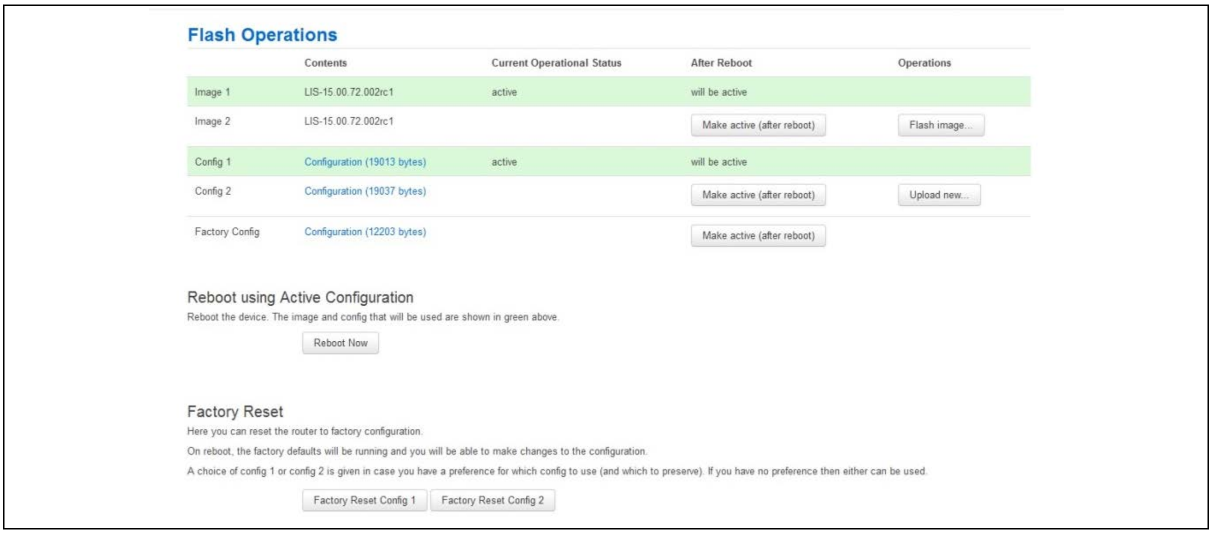


Figure 32: The flash operations page

Under Flash Operations, click **Flash Image**. Only the inactive image is available to flash.

Select the appropriate image and then wait until image has loaded.

Note: this process may take a while depending on the available connection speed.

When the image has loaded, the Update Firmware page appears.

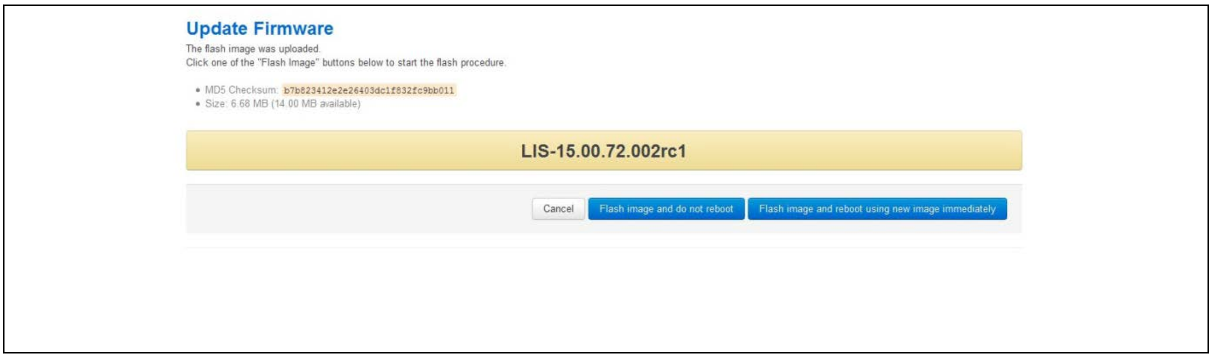


Figure 33: The flash firmware - verify page

Click either: **Flash image and do not reboot**, or **Flash image and reboot using new image immediately**. The 'Firmware update is being applied' message appears.

When the firmware update is complete, the Update Firmware page appears. There are various messages, depending on which option you selected, or if any corruptions have occurred.

8.1.4 Flash image and do not reboot option

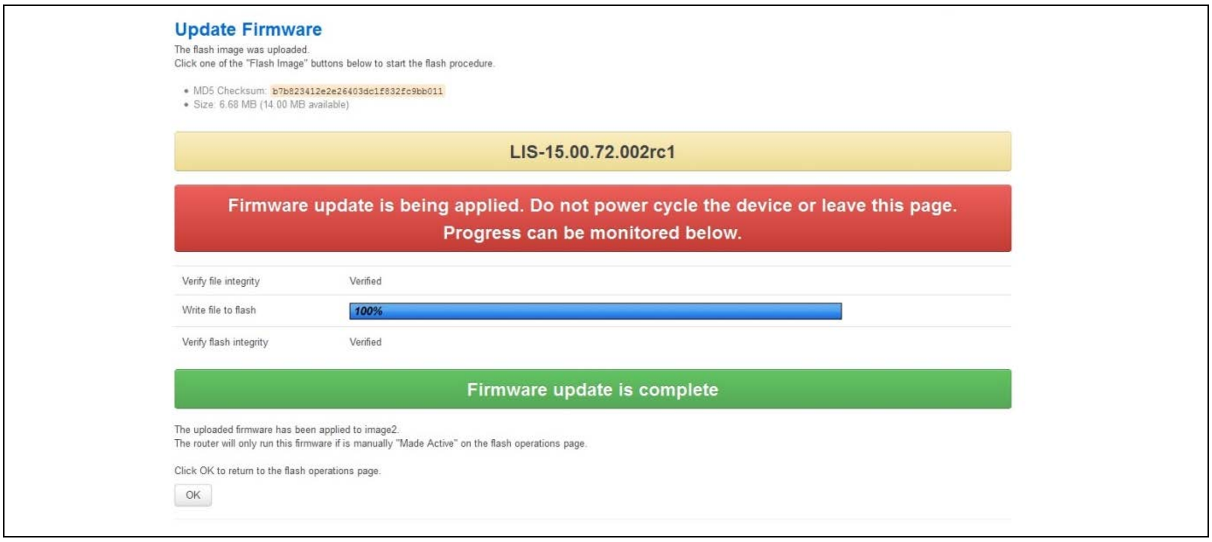


Figure 34: The firmware update page after '…do not reboot' option selected

If you select 'Flash image and do not reboot', the router will only run the firmware if you click **OK** to return to the Flash Operations page. There you can manually select **Made Active (after reboot)**. Then click **Reboot Now** in the 'Reboot using Active Configuration' section.

8.1.5 Update flash image and reboot using new image immediately option

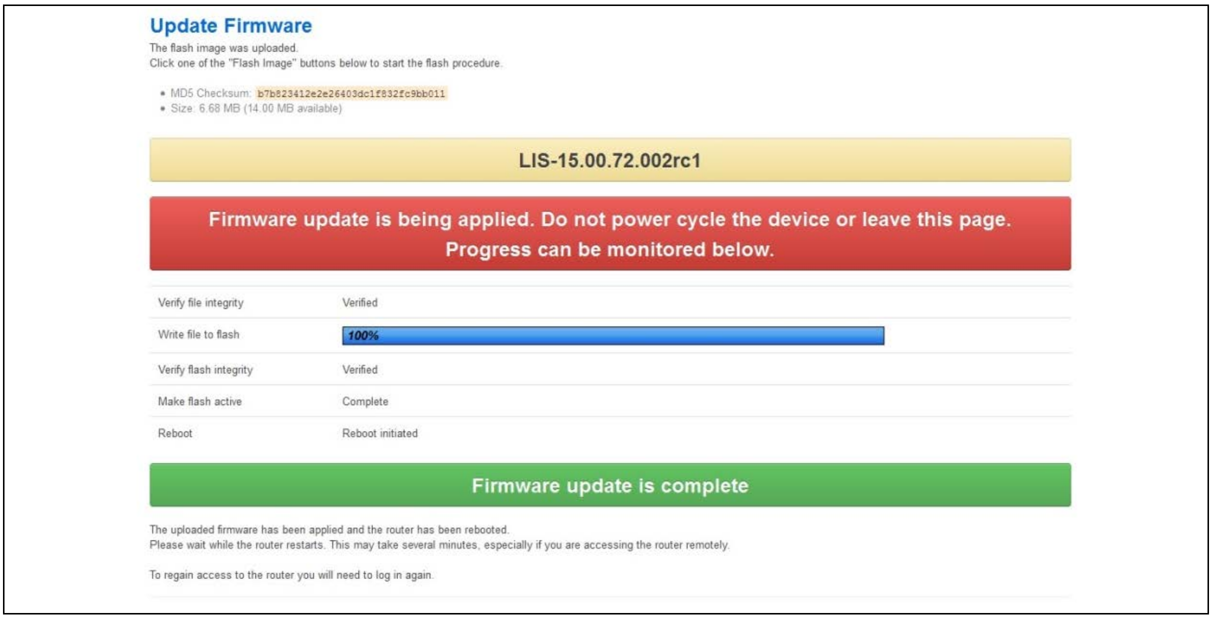


Figure 35: The firmware update page after 'update flash image and reboot…' option selected

If you select 'Update flash image and reboot using new image immediately' and the overall validation and flashing process has succeeded, the router will reboot immediately. To regain access to the router you must login again. If any part of the processes encounters an error the reboot does **not** occur and a report is given.

8.1.6 Possible file corruption

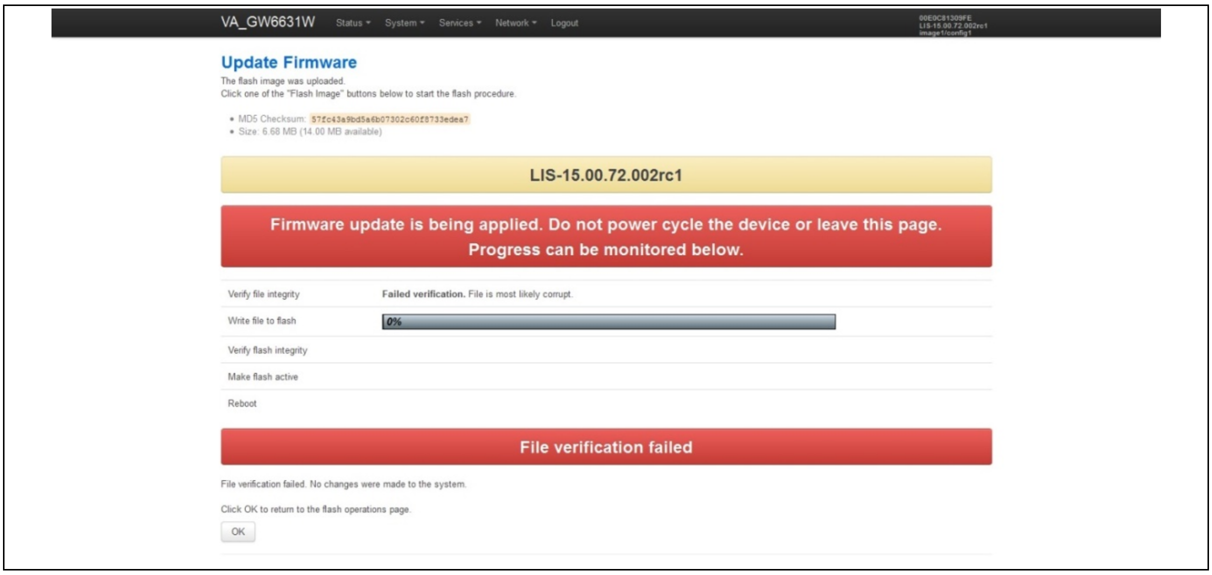


Figure 36: The firmware update failure page

In the unfortunate event that the firmware upgrade fails, the 'Failed verification File is most likely corrupt' or similar message will appear in the Verify file integrity row. No changes will be made to the system and the general message **File verification failed** appears.

8.1.7 Verify the firmware has been upgraded successfully

To check the firmware version, in the top menu, browse to **System -> Flash Operations**, or after router reboots, in the top menu, click **Status**. The Firmware Version shows in the system list and also in the right top corner of the menu bar.

Figure 37: The system status list showing current firmware version

8.2 Upgrading firmware using CLI

8.2.1 Transfer file to router

To upgrade firmware using CLI, you will need a TFTP server on a connected PC or SCP available.

Open up an SSH or Telnet session to the router.

Enter in the relevant username and password.

To access the temp folder, enter **cd /tmp**

Depending on the router's software version the following TFTP clients are available:

- atftp
- curl

To determine which is available on your router, enter:

```
which curl || which atftp
```

The output shows the available application:

```
/usr/bin/curl
```

ATFTP

Inline command usage:

```
atftp -g -r LIS-15.00.72.002.image –l /tmp/LIS-15.00.72.002.image x.x.x.x
```

where x.x.x.x is the IP address of your PC, **-g** is get operation and **-l** / **-r** are local and remote file name to store.

CURL

Inline command usage:

```
curl tftp://x.x.x.x/LIS-15.00.72.002.image -o /tmp/LIS-15.00.72.002.image
```

where x.x.x.x is the IP of your PC, **-o** is local file name to store.

SCP

Secure Copy (SCP) is a part of Secure Shell (SSH) and enables file transfers to the router using authentication and encryption. It is different to TFTP, which uses UDP, while SCP uses a TCP connection. On Unix machines, SCP is a standard part of the system; on Windows it requires an additional application.

The usage example below is for a Unix machine and therefore assumes the image file is in the current folder.

```
scp LIS-15.00.72.002.image root@x.x.x.x:/tmp/LIS-15.00.72.002.image
```

Where the first argument 'LIS-15.00.72.002.image' in SCP is the source and the second argument 'tmp/LIS-15.00.72.002.image' is the destination path, enter **root** as the username to connect to x.x.x.x IP address.

After you execute the above command you will be asked to provide a root password.

At this stage the output shows the process of copying the software file into destination directory.

```
root@192.168.100.1's password:
LIS-15.00.72.000.image                100%  6812KB   2.2MB/s        00:03
```

8.2.1.1 Image verification before flashing

To verify the integrity of the image, firmware version xx.yy.72.002 and later uses an image-check application.

Note: it is the user's responsibility to verify the image before starting to write the image to flash process.

To use the image-check on downloaded image, enter:

```
image-check /tmp/LIS-15.00.72.002.image
```

In the case of any image corruption, an appropriate error message appears:

```
Error: no SquashFS filesystem after CRC'd section - data length 3
Error: read failed, expected at least 3 more bytes
```

or similar.

Note: the image is valid only if no error message appears. This process is done automatically during Web UI firmware update.

Flashing

When downloaded firmware verification succeeds, the new image can be written to flash.

To write the image into the alternative image, enter:

```
mtd write LIS-15.00.72.002.image altimage
```

Note: this is an example, substitute the correct file name.

8.2.1.2 Flash verification after flashing

After the write process has finished, you must complete a post verification of the firmware.

To verify the checksum of downloaded firmware, enter:

```
va_image_csum.sh /tmp/LIS-15.00.72.002.image
```

The checksum of the downloaded binary is shown:

```
08761cd03e33c569873bcc24cf2b7389  7006920  LIS-15.00.72.002 This MD5
```

To verify the checksum of written firmware, enter:

```
va_image_csum.sh alt
```

After a while the checksum will be calculated:

```
Calculating checksum.........
```

```
08761cd03e33c569873bcc24cf2b7389  7006920  LIS-15.00.72.002 This MD5
```

Verify and compare the checksum with the MD5 sum of the downloaded image.

If the checksum of the written firmware in altimage matches the one from the downloaded image in /tmp, the new firmware has been programmed successfully.

8.2.1.3 Setup an alternative image

Provided the programming has succeeded, you can set it as the next image to use after reboot; enter:

```
vacmd set next image altimage
```

To reboot using the new firmware, enter:

```
reboot
```

9 System settings

The system section contains settings that apply to the most basic operation of the system, such as the host name, time zone, logging details, NTP server, language and style.

The host name appears in the top left hand corner of the interface menu. It also appears when you open a Telnet or SSH session.

Note: this document shows no host name in screen grabs. Throughout the document we use the host name 'VA_router'.

The system configuration contains a logging section for the configuration of a Syslog client.

9.1 Configuration package used

Package	Sections
system	main
	timeserver

9.2 Configuring system properties

To set your system properties, in the top menu, click **System**. There are four sections in the System page.

Section	Description
General settings	Configure host name, local time and time zone.
Logging	Configure a router to log to a server. You can configure a Syslog client in this section.
Language and Style	Configure the router's web language and style.
Time synchronization	Configure the NTP server in this section.

9.2.1 General settings

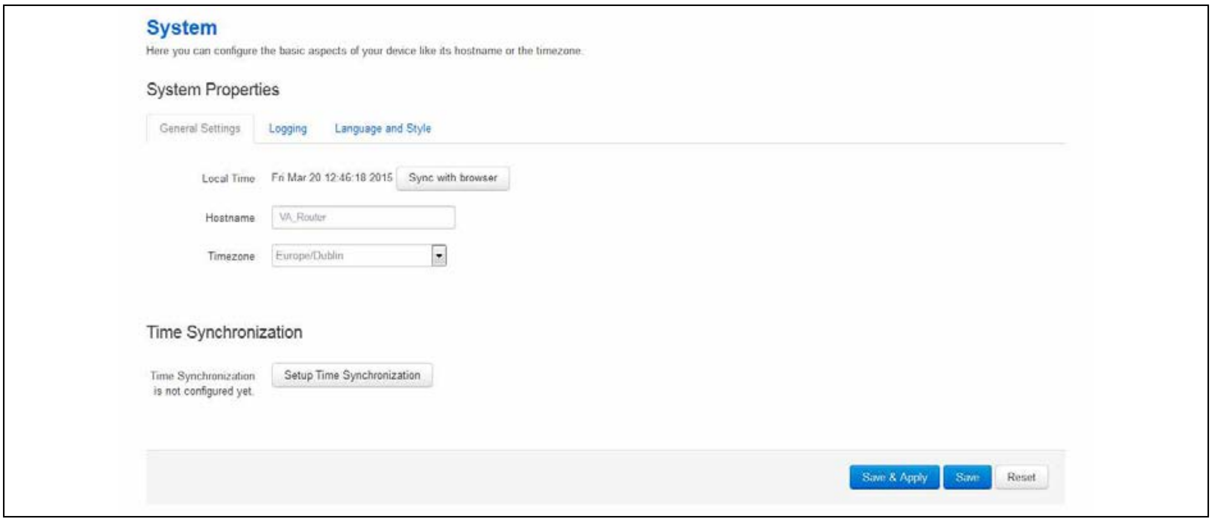


Figure 38: General settings in system properties

Web Field / UCI / Package Option	Description
Web: Local Time	Sets the local time and syncs with browser. You can manually configure on CLI, using: `date –s YYYY.MM.DD-hh:mm:ss`
Web: hostname UCI: system.main.hostname Opt: hostname	Specifies the hostname for this system.
Web: Timezone UCI: system.main.timezone Opt: timezone	Specifies the time zone that the date and time should be rendered in by default.
Web: n/a UCI: system.main.timezone Opt: time_save_interval_min	Defines the interval in minutes to store the local time for use on next reboot. 10m

Table 11: Information table for general settings section

9.2.2 Logging

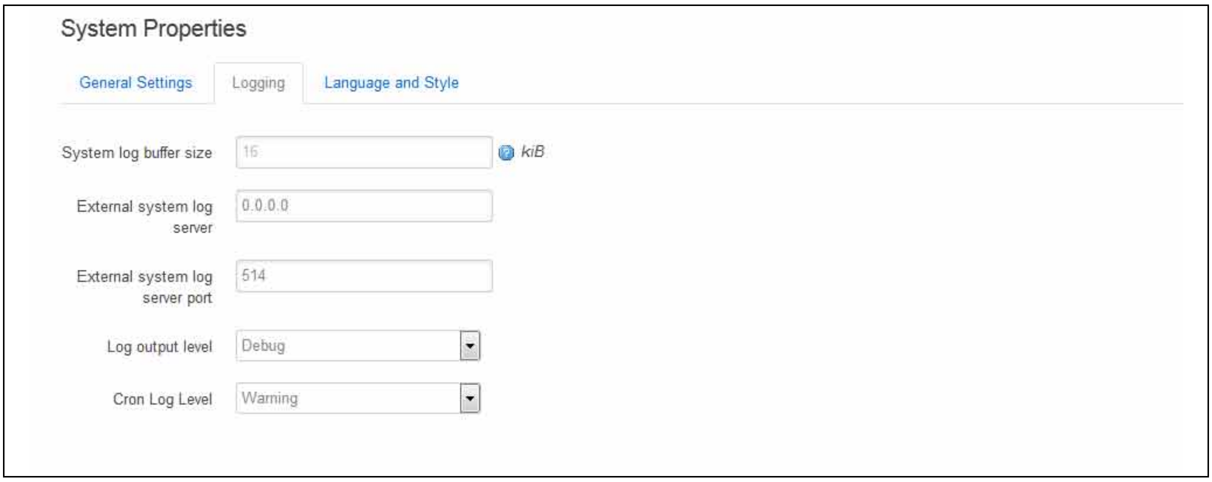


Figure 39: The logging section in system properties

Web Field / UCI / Package Option	Description
Web: System log buffer size UCI: system.main.log_size Opt: log_size	Log buffer size in KB. Range 16 16 KB
Web: External system log server UCI: system.main.log_ip Opt: log_ip	External syslog server IP address. Range 0.0.0.0
Web: External system log server port UCI: system.main.log_port Opt: log_port	External syslog server port number. Range 514

<table>
<tr><td>Web: Log output level
UCI: system.main.conloglevel
Opt: conloglevel</td><td>Sets the maximum log output level severity for system events. System events are written to the system log. Messages with a lower level or level equal to the configured level are displayed in the console using the logread command, or alternatively written to flash, if configured to do so.
<table>
<tr><th>Web value</th><th>Description</th><th>UCI</th></tr>
<tr><td>Debug</td><td>Information useful to developers for debugging the application.</td><td>8</td></tr>
<tr><td>Info</td><td>Normal operational messages that require no action.</td><td>7</td></tr>
<tr><td>Notice</td><td>Events that are unusual, but not error conditions.</td><td>6</td></tr>
<tr><td>Warning</td><td>May indicate that an error will occur if action is not taken.</td><td>5</td></tr>
<tr><td>Error</td><td>Error conditions</td><td>4</td></tr>
<tr><td>Critical</td><td>Critical conditions</td><td>3</td></tr>
<tr><td>Alert</td><td>Should be addressed immediately</td><td>2</td></tr>
<tr><td>Emergency</td><td>System is unusable</td><td>1</td></tr>
</table></td></tr>
<tr><td>Web: Cron Log Level
UCI: system.main.cronloglevel
Opt: cronloglevel</td><td>Sets the maximum log level for kernel messages to be logged to the console. Only messages with a level lower, or level equal to the configured level will be printed to the console.
<table>
<tr><th>Web value</th><th>Description</th><th>UCI</th></tr>
<tr><td>Normal</td><td>Normal operation messages</td><td>8</td></tr>
<tr><td>Warning</td><td>Error messages</td><td>9</td></tr>
<tr><td>Debug</td><td>Debug messages</td><td>5</td></tr>
</table></td></tr>
<tr><td>Web: n/a
UCI: system.main.log_file
Opt: log_file</td><td>Since logread is only small in size it can be beneficial to write system events to flash. This option defines the file path to write the events. Set to ‘root/syslog.messages’</td></tr>
<tr><td>Web: n/a
UCI: system.main.log_type
Opt: log_type</td><td>Defines whether to write the system events to a file rather than logread. Set to ‘file’ to write to the file configured under log_file option.</td></tr>
<tr><td>Web: n/a
UCI: system.main.log_file_count
Opt: log_file_count</td><td>Defines the number of archive syslog files to store in flash. When configured above to write to /root.syslog.messages files will be stored at /root/syslog.messages,x (where x starts at 0).
<table>
<tr><td>Range</td><td></td></tr>
<tr><td>1</td><td>Stores 1 archive log file in flash</td></tr>
</table></td></tr>
</table>

Table 12: Information table for the logging section

9.2.3 Language and style

Figure 40: The language and style section in system properties

Web Field / UCI / Package Option	Description
Language	Sets the language to 'auto' or 'English'. Auto English
Design	Sets the router's style.

Table 13: Information table for the language and style page

9.2.4 Time synchronization

The router time must be synchronised using NTP. The router can act as both an NTP client and an NTP server. It is enabled as an NTP client by default and individual interfaces can be configured to respond to NTP requests.

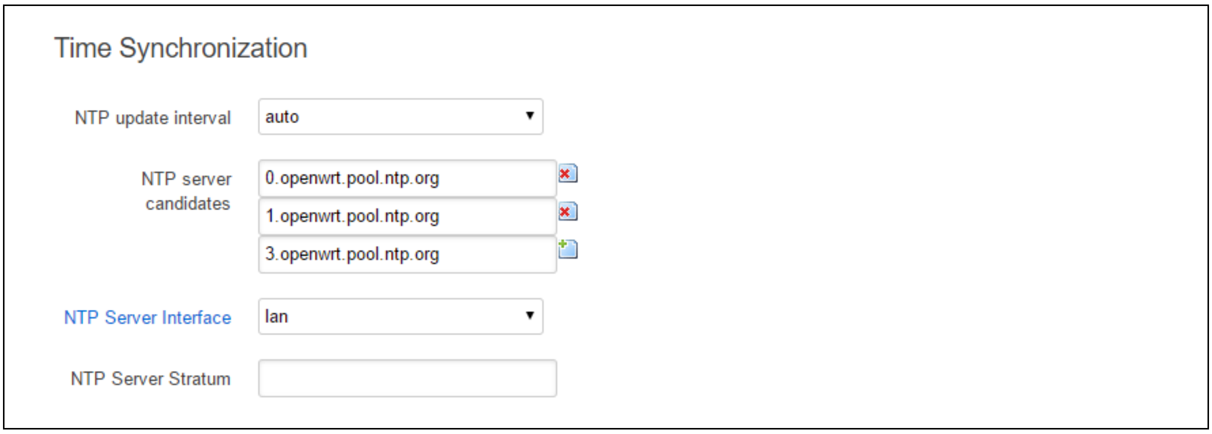


Figure 41: The time synchronization section in system properties

Web Field / UCI / Package Option	Description
Web: NTP update interval UCI: system.ntp.interval_hours Opt: interval_hours	Specifies interval of NTP requests in hours. Default value set to auto. Auto Range: auto; 1-23
Web: NTP server candidates UCI: system.ntp.server Opt: list server	Defines the list of NTP servers to poll the time from. If the list is empty, the built in NTP daemon is not started. Multiple servers can be configured and are separated by a space if using UCI. By default all fields are set to 0.0.0.0.
Web: NTP Server Interface UCI: system.ntp.listen Opt: listen	Defines a list of interfaces that respond to NTP requests. Interfaces should be delimited using space. Example: option listen 'LAN1 LAN2' Blank: Do not respond to NTP requests. Range
Web: NTP Server Stratum UCI: system.ntp.stratum Opt: stratum	Defines how far this NTP Server is from the reference clock. For example, an NTP server getting time directly from the reference clock will have a stratum of 1. In general, this should be left blank, which means that the router NTP Server will derive the stratum from the NTP dialogue. Blank: NTP server will derive stratum Range

Table 14: Information table for time synchronization section

9.2.5 System reboot

The router can be configured to reboot immediately, or scheduled to reboot a configured time in the future.

In the top menu, select **System -> Reboot**. The System page appears.

Ensure you have saved all your configuration changes before you reboot.

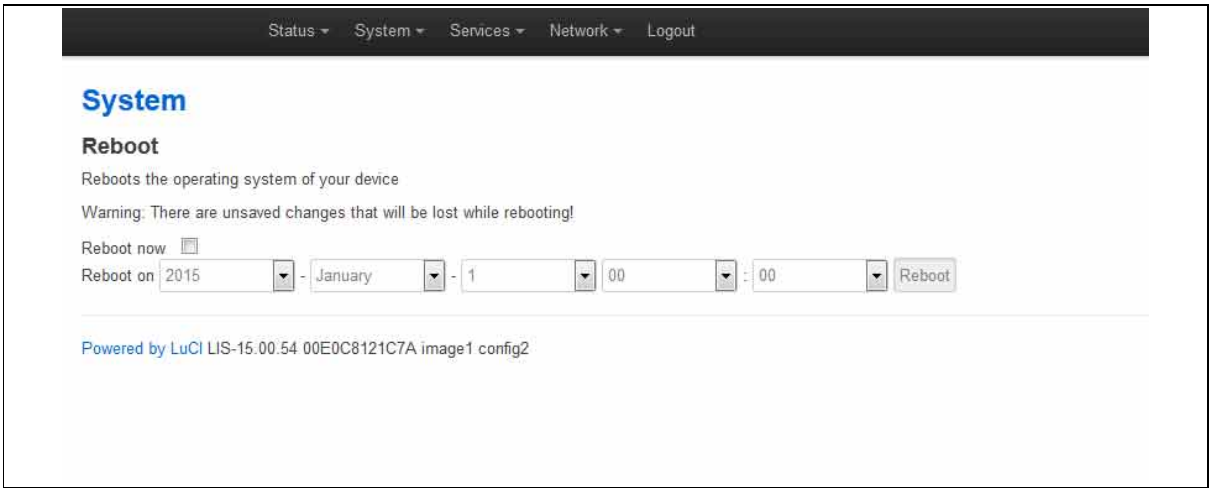


Figure 42: The reboot page

Check the **Reboot now** check box and then click **Reboot**.

9.3 System settings using UCI

```
root@VA_router:~# uci show system
system.main=system
system.main.hostname=VA_router
system.main.timezone=UTC
system.main.log_ip=1.1.1.1
system.main.log_port=514
system.main.conloglevel=8
system.main.cronloglevel=8
system.ntp.interval_hours=auto
system.ntp.server=0.VA_router.pool.ntp.org 10.10.10.10
System settings using package options
root@VA_router:~# uci export system
package 'system'

config 'system' 'main'
      option 'hostname' "VA_router"
      option 'timezone' "UTC"
```

```
      option 'log_ip' "1.1.1.1"
      option 'log_port' "514"
      option time_save_interval_min "10"
      option conloglevel '8'
      option cronloglevel '8'

config 'timeserver' 'ntp'
      option interval_hours 'auto'
      list server "0.VA_router.pool.ntp.org"
      list server ’10.10.10.10’
      option listen ‘LAN1 LAN2’
```

9.4 System diagnostics

9.4.1 System events

Events in the system have a class, sub class and severity. All events are written to the system log.

9.4.1.1 Logread

To view the system log, enter:

```
root@VA_router:~# logread
```

Shows the log.

```
root@VA_router:~# logread |tail
```

Shows end of the log.

```
root@VA_router:~# logread | more
```

Shows the log page by page.

```
root@VA_router:~# logread –f
```

Shows the log on an ongoing basis. To stop this option, press **ctrl-c**.

```
root@VA_router:~# logread –f &
```

Shows the log on an ongoing basis while in the background. This allows you to run other commands while still tracing the event logs. To stop this option, type **fg** to view the current jobs, then press **ctrl-c** to kill those jobs.

9.4.2 System events in flash

Since logread is only small in size it can be beneficial to write system events to flash. To do this you need to modify the system config under the system package. Set the options 'log_file', 'log_size' and 'log_type' as below:

```
root@VA_router:~# uci export system
package system
config system 'main'
        option hostname 'VA_router'
        option zonename 'UTC'
        option timezone 'GMT0'
        option conloglevel '8'
        option cronloglevel '8'
        option time_save_interval_hour '10'
        option log_hostname '%serial'
        option log_ip '1.1.1.1'
        option log_port '514'
        option log_file '/root/syslog.messages'
        option log_size '400'
        option log_type 'file'
```

The above commands will take effect after a reboot.

```
root@VA_router:~# cat /root/syslog.messages
```

Shows all the system events stored in flash.

```
root@VA_router:~# tail /root/syslog.messages
```

Shows end of the events stored flash.

```
root@VA_router:~# tail –f /root/syslog.messages &
```

Shows the log on an ongoing basis. To stop this option, press **ctrl-c**.

10 Configuring an Ethernet interface

This section describes how to configure an Ethernet interface including configuring the interface as a DHCP server, adding the interface to a firewall zone, mapping the physical switch ports and defining loopback interface.

10.1 Configuration packages used

Package	Sections
network	interface
	route
	va_switch
	alias
firewall	zone
dhcp	dhcp

10.2 Configuring an Ethernet interface using the web interface

To create and edit interfaces via the web interface, in the top menu, click **Network -> Interfaces**. The Interfaces overview page appears.

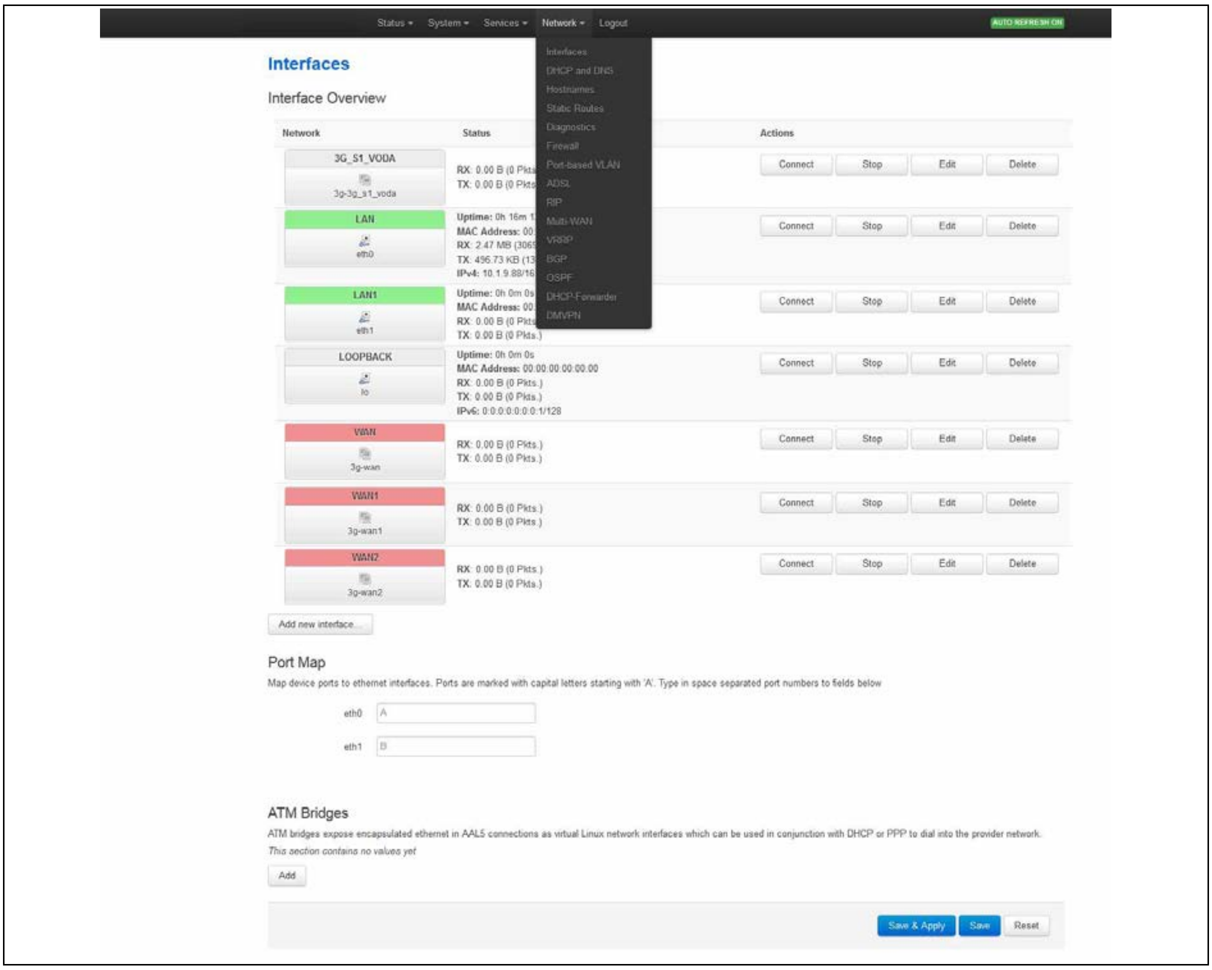


Figure 43: The interfaces overview page

There are three sections in the Interfaces page.

Section	Description
Interface Overview	Shows existing interfaces and their status. You can create new, and edit existing interfaces here.
Port Map	In this section you can map device ports to Ethernet interfaces. Ports are marked with capital letters starting with 'A'. Type in space-separated port character in the port map fields.
ATM Bridges	ATM bridges expose encapsulated Ethernet in AAL5 connections as virtual Linux network interfaces, which can be used in conjunction with DHCP or PPP to dial into the provider network.

10.2.1 Interface overview: editing an existing interface

To edit an existing interface, from the interface tabs at the top of the page, select the interface you wish to configure. Alternatively, click **Edit** in the interface's row.

10.2.2 Interface overview: creating a new interface

To create a new interface, in the Interface Overview section, click **Add new interface**. The Create Interface page appears.

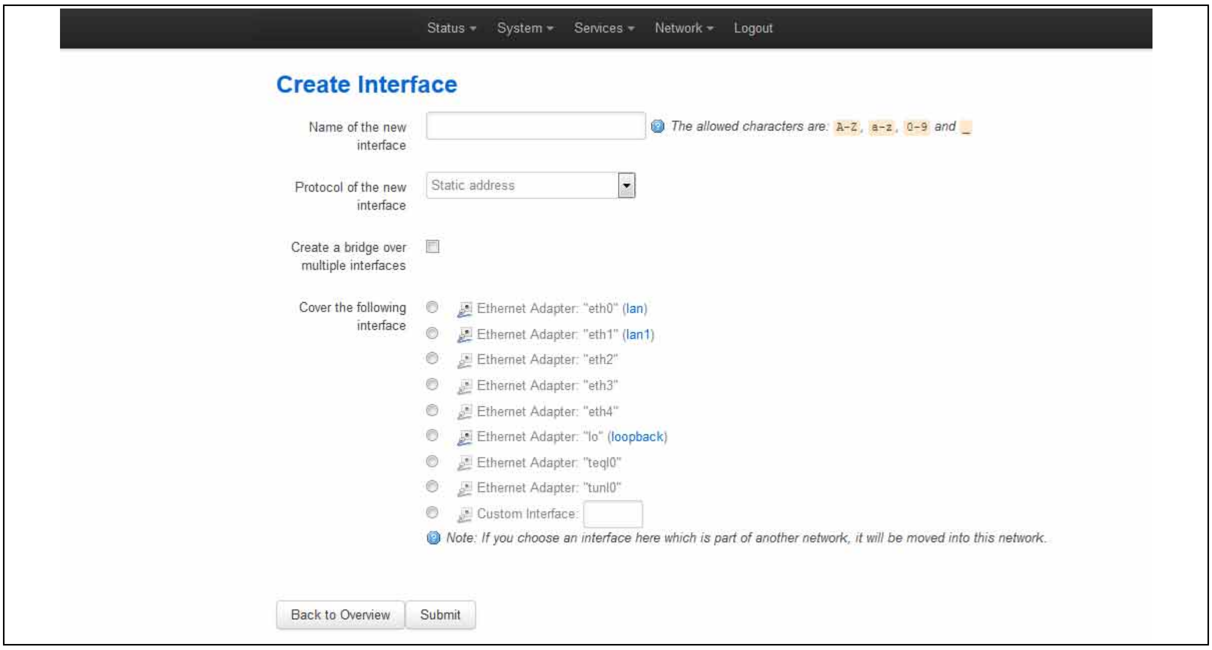


Figure 44: The create interface page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface
UCI: network.<if name>
Opt: config interface</td><td>Assigns a logical name to the interface. The network interface section will assign this name (<if name>).
Type the name of the new interface.
Allowed characters are A-Z, a-z, 0-9 and _</td></tr>
<tr><td>Web: Protocol of the new interface
UCI: network.<if name>.proto
Opt: proto</td><td>Specifies what protocol the interface will operate on. Select Static.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td></tr>
<tr><td>PPP</td><td>Point to Point Protocol</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: Create a bridge over multiple interfaces
UCI: network.<if name>.type
Opt: type</td><td>If you select this option, then the new logical interface created will act as a bridging interface between the chosen existing physical interfaces.
<table>
<tr><td>Empty</td><td></td></tr>
<tr><td>Bridge</td><td>Configures a bridge over multiple interfaces.</td></tr>
</table></td></tr>
<tr><td>Web: Cover the following interface
UCI: network.<if name>.ifname
Opt: ifname</td><td>Physical interface name to assign to this logical interface. If creating a bridge over multiple interfaces select two interfaces to bridge. When using uci the interface names should be separated by a space e.g. option ifname ‘eth2 eth3’</td></tr>
</table>

Table 15: Information table for the create new interface page

Click **Submit**. The Interface configuration page appears. There are three sections:

Section	Description
Common Configuration	Configure the interface settings such as protocol, IP address, gateway, netmask, custom DNS servers, MTU and firewall configuration.
IP-Aliases	Assigning multiple IP addresses to the interface
DHCP Server	Configuring DHCP server settings for this interface

10.2.3 Interface overview: common configuration

The common configuration section has four sub sections:

Section	Description
General Setup	Configure the basic interface settings such as protocol, IP address, gateway, netmask, custom DNS servers.
Advanced Settings	'Bring up on boot', 'Monitor interface state', Override MAC address, Override MTU and 'Use gateway metric'
Physical Settings	Bridge interfaces, VLAN PCP to SKB priority mapping,
Firewall settings	Assign a firewall zone to the interface

10.2.3.1 Common configuration – general setup

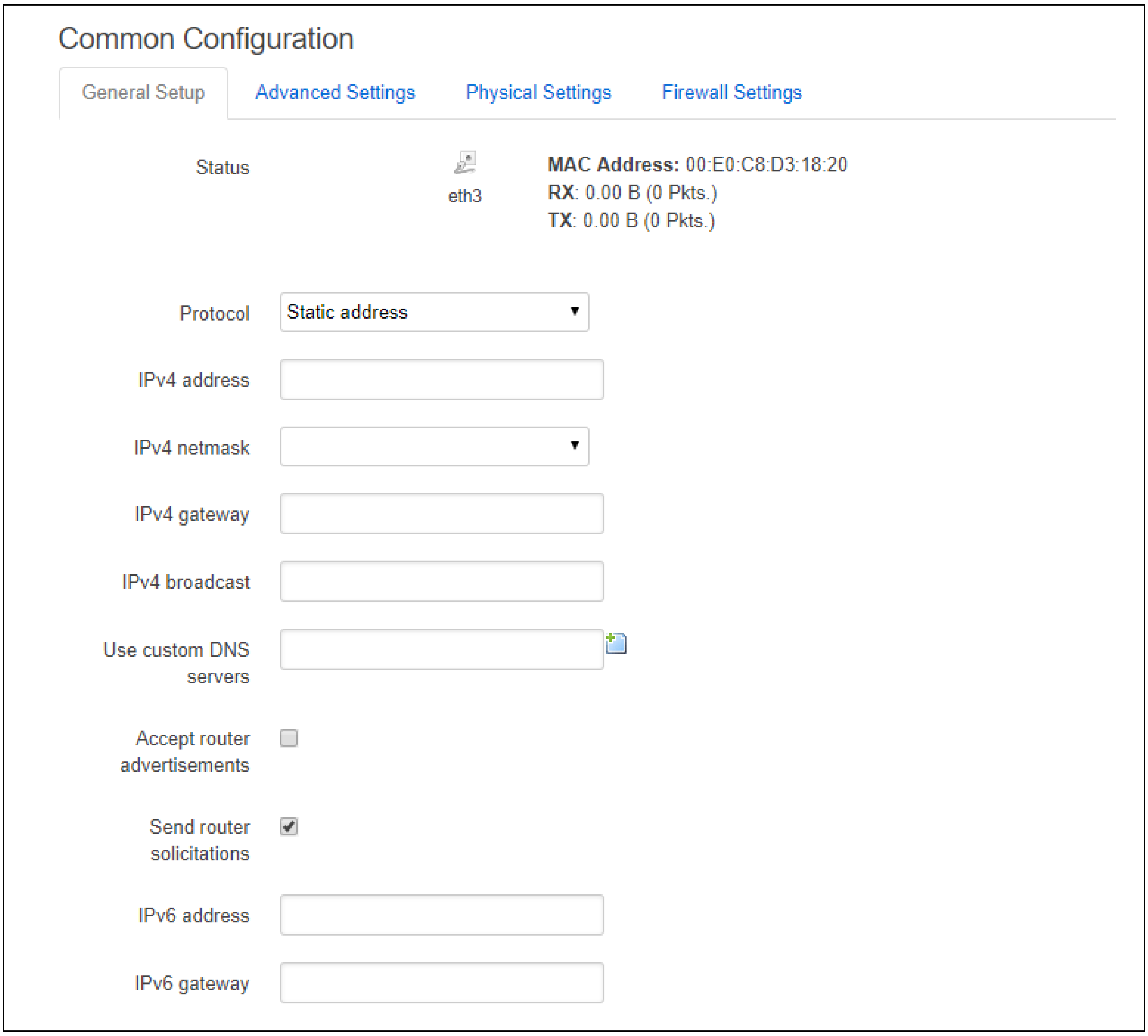


Figure 45: The Ethernet connection common configuration settings page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td colspan="2">General Setup</td></tr>
<tr><td>Web: Status</td><td>Shows the current status of the interface.</td></tr>
<tr><td>Web: Protocol
UCI: network.<if name>.proto
Opt: proto</td><td>Protocol type. The interface protocol may be one of the options shown below. The protocol selected in the previous step will be displayed as default but can be changed if required.
<table>
<tr><td>Option</td><td>Description</td></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol.</td></tr>
<tr><td>PPP</td><td>Point-to-Point protocol</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: IPv4 address
UCI: network.<if name>.ipaddr
Opt: ipaddr</td><td>The IPv4 address of the interface. This is optional if an IPv6 address is provided.</td></tr>
<tr><td>Web: IPv4 netmask
UCI: network.<if name>.netmask
Opt: netmask</td><td>Subnet mask to be applied to the IP address of this interface.</td></tr>
<tr><td>Web: IPv4 gateway
UCI: network.<if name>.gateway
Opt: gateway</td><td>IPv4 default gateway to assign to this interface (optional).</td></tr>
<tr><td>Web: IPv4 broadcast
UCI: network.<if name>.broadcast
Opt: broadcast</td><td>Broadcast address. This is automatically generated if no broadcast address is specified.</td></tr>
<tr><td>Web: Use custom DNS servers
UCI: network.<if name>.dns
Opt: list dns</td><td>List of DNS server IP addresses (optional). Multiple DNS Servers are separated by a space if using UCI.</td></tr>
<tr><td>Web: Accept router advertisements
UCI: network.<if name>.accept_ra
Opt: accept_ra</td><td>Specifies whether to accept IPv6 Router Advertisements on this interface (optional).
Note: default is 1 if protocol is set to DHCP, otherwise defaults to 0.</td></tr>
<tr><td>Web: Send router solicitations
UCI: network.<if name>.send_rs
Opt: send_rs</td><td>Specifies whether to send Router Solicitations on this interface (optional).
Note: defaults to 1 for Static protocol, otherwise defaults to 0.</td></tr>
<tr><td>Web: IPv6 address
UCI: network.<if name>.ip6addr
Opt: ip6addr</td><td>The IPv6 IP address of the interface. Optional if an IPv4 address is provided.
CIDR notation for the IPv6 address is required.</td></tr>
<tr><td>Web: IPv6 gateway
UCI: network.<if name>.ip6gw
Opt: ip6gw</td><td>Assign given IPv6 default gateway to this interface (optional).</td></tr>
</table>

Table 16: Information table for LAN interface common configuration settings

10.2.3.2 Common configuration: advanced settings

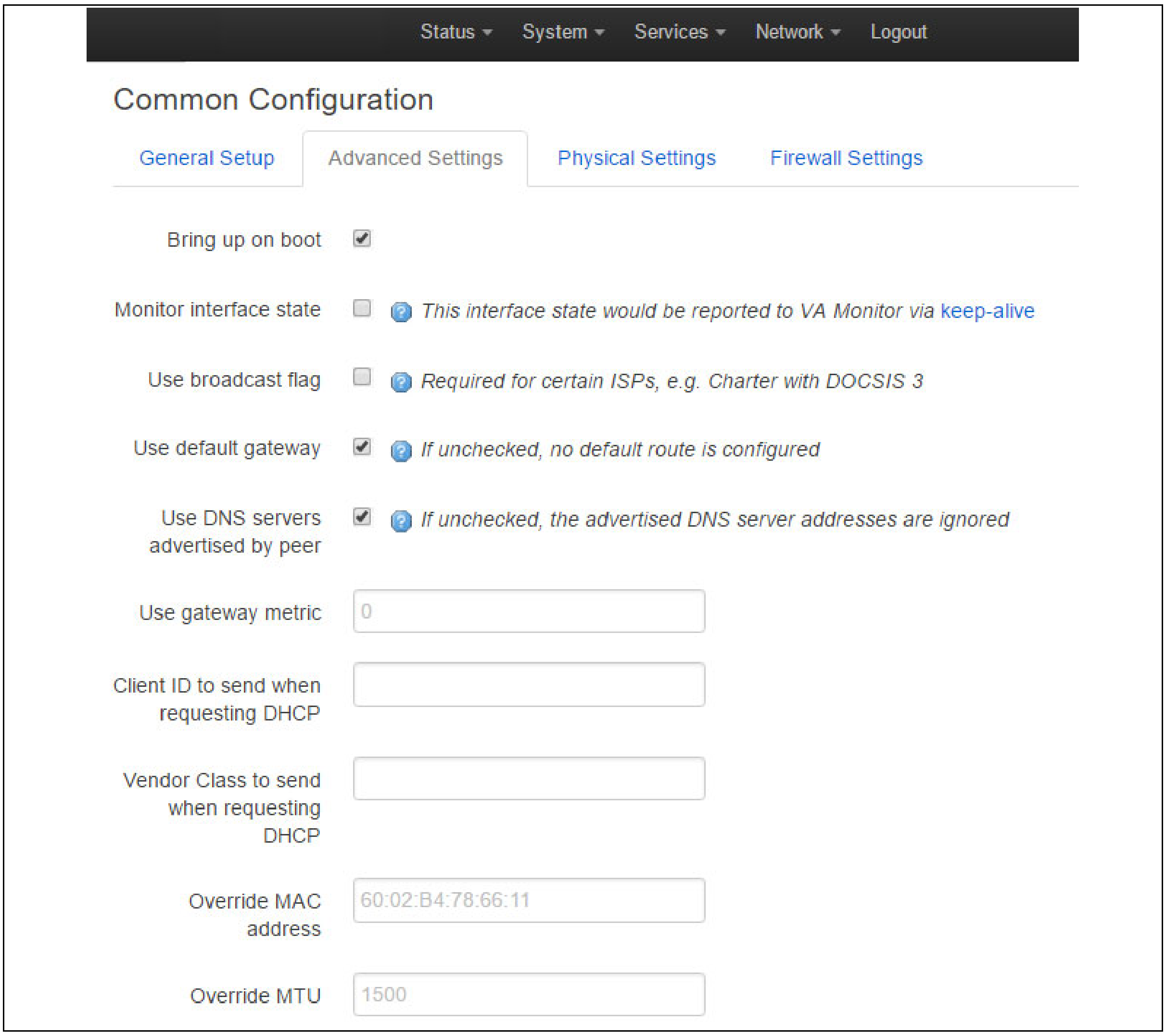


Figure 46: The Ethernet connection advanced settings page

Web Field / UCI / Package Option	Description
Web: Bring up on boot UCI: network.<if name>.auto Opt: auto	Enables the interface to connect automatically on boot up. 0 — Disabled. 1 — Enabled.
Web: Monitor interface state UCI: network.<if name>.monitored Opt: monitored	Enabled if status of interface is presented on Monitoring platform. 0 — Disabled. 1 — Enabled.
Web: Override MAC address UCI: network.<if name>.macaddr Opt: macaddr	Override the MAC address assigned to this interface. Must be in the form: hh:hh:hh:hh:hh:hh, where h is a hexadecimal number.
Web: Override MTU UCI: network.<if name>.mtu Opt: mtu	Defines the value to override the default MTU on this interface. 1500 — 1500 bytes
Web: Use gateway metric UCI: network.<if name>.metric Opt: metric	Specifies the default route metric to use for this interface (optional). 0 Range

<table>
<tr>
<td>Web: Dependant Interfaces
UCI: network.[..x..].dependants
Opt: dependants</td>
<td>Lists interfaces that are dependent on this parent interface. Dependant interfaces will go down when parent interface is down and will start or restart when parent interface starts.
Separate multiple interfaces by a space when using UCI.
Example: <code>option dependants 'PPPADSL MOBILE'</code>
This replaces the following previous options in child interfaces.
<table>
<tr><td>gre</td><td>option local_interface</td></tr>
<tr><td>lt2p</td><td>option src_ipaddr</td></tr>
<tr><td>iot</td><td>option wan1 wan2</td></tr>
<tr><td>6in4</td><td>option ipaddr</td></tr>
<tr><td>6to4</td><td>option ipaddr</td></tr>
</table>
</td>
</tr>
<tr>
<td>Web: SNMP Alias ifindex
UCI: network.[..x..].snmp_alias_ifindex
Opt: snmp_alias_ifindex</td>
<td>Defines a static SNMP interface alias index for this interface, that can be polled via the SNMP interface index (snmp_alias_ifindex+1000). See Configuring SNMP section for more information
<table>
<tr><td>Blank</td><td>No SNMP interface alias index</td></tr>
<tr><td>Range</td><td>0 - 4294966295</td></tr>
</table>
</td>
</tr>
</table>

Table 17: Information table for common configuration advanced settings

10.2.3.3 Common configuration: physical settings

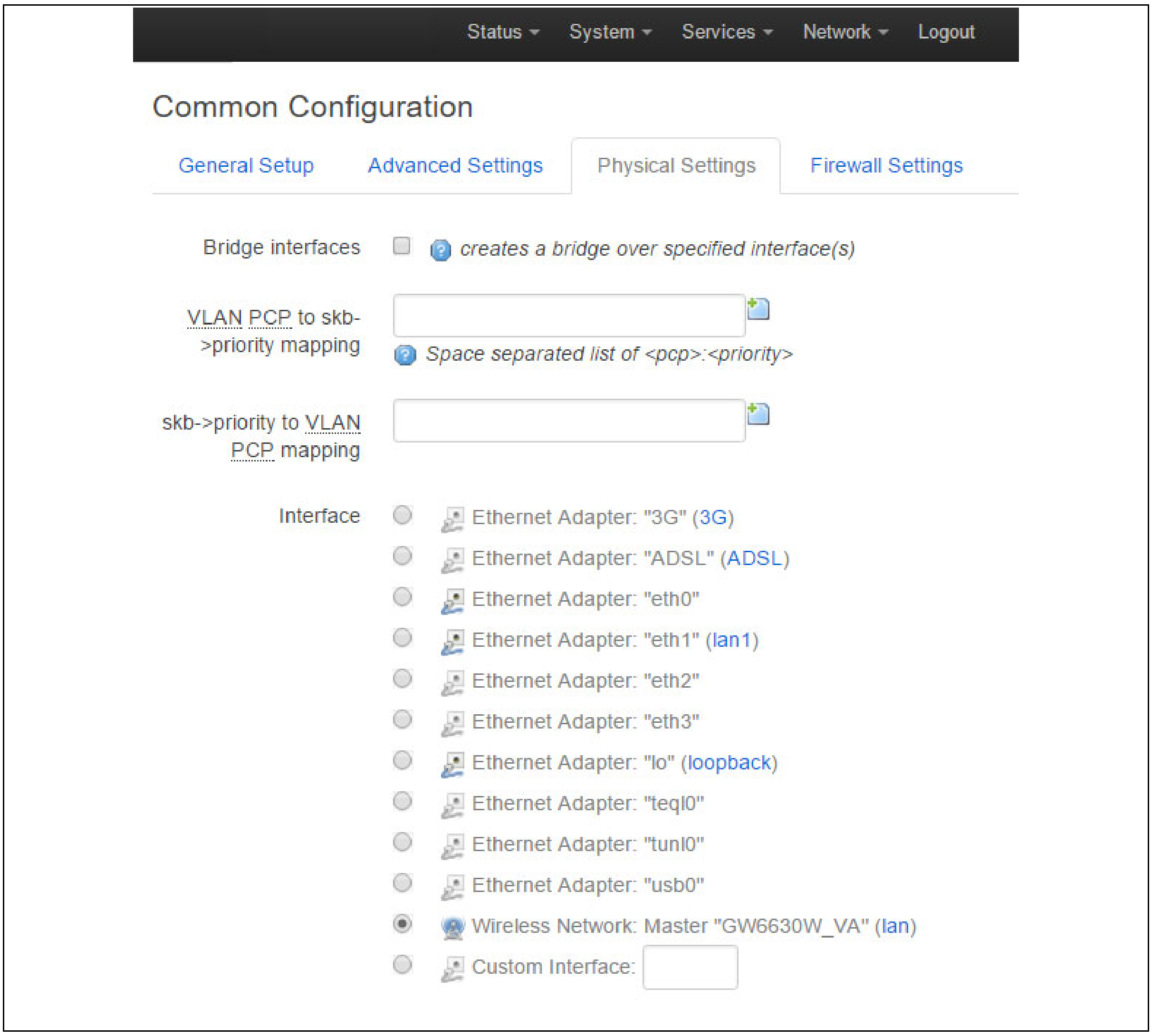


Figure 47: The common configuration physical settings page

Web Field / UCI / Package Option	Description
Web: Bridge interfaces UCI: network.<if name>.type Opt: type	Sets the interface to bridge over a specified interface(s). The physical interfaces can be selected from the list and are defined in network.<if name>.ifname. Empty Bridge: Configures a bridge over multiple interfaces.
Web: Enable STP UCI: network.<if name>.stp Opt: stp	Enable Spanning Tree Protocol. This option is only available when the Bridge Interfaces option is selected. 0: Disabled. 1: Enabled.
Web: VLAN PCP to skb>priority mapping UCI: network.<if name>.vlan_qos_map_ingress Opt: list vlan_qos_map_ingress	VLAN priority code point to socket buffer mapping. Multiple priority mappings are entered with a space between them when using UCI. Example: network.<if name>. vlan_qos_map_ingress =1:2 2:1
Web: skb priority to >VLAN PCP mapping UCI: network.<if name>.vlan_qos_map_egress Opt: list vlan_qos_map_egress	Socket buffer to VLAN priority code point mapping. Multiple priority mappings are entered with a space between them when using UCI. Example: network.<if name>. vlan_qos_map_egress =1:2 2:1
Web: Interface UCI: network.<if name>.ifname Opt: ifname	Physical interface to assign the logical interface to. If mapping multiple interfaces for bridging the interface names are separated by a space when using UCI and package options. Example: option ifname 'eth2 eth3' or network.<if name>.ifname=eth2 eth 3

Table 18: Information table for physical settings page

10.2.3.4 Loopback interfaces

Loopback interfaces are defined in exactly the same way as Ethernet interfaces. Please see section above.

Note: There is no software limitation as to how many loopback interfaces can exist on the router.

10.2.3.5 Common configuration: firewall settings

Use this section to select the firewall zone you want to assign to this interface.

Select **unspecified** to remove the interface from the associated zone or fill out the create field to define a new zone and attach the interface to it.

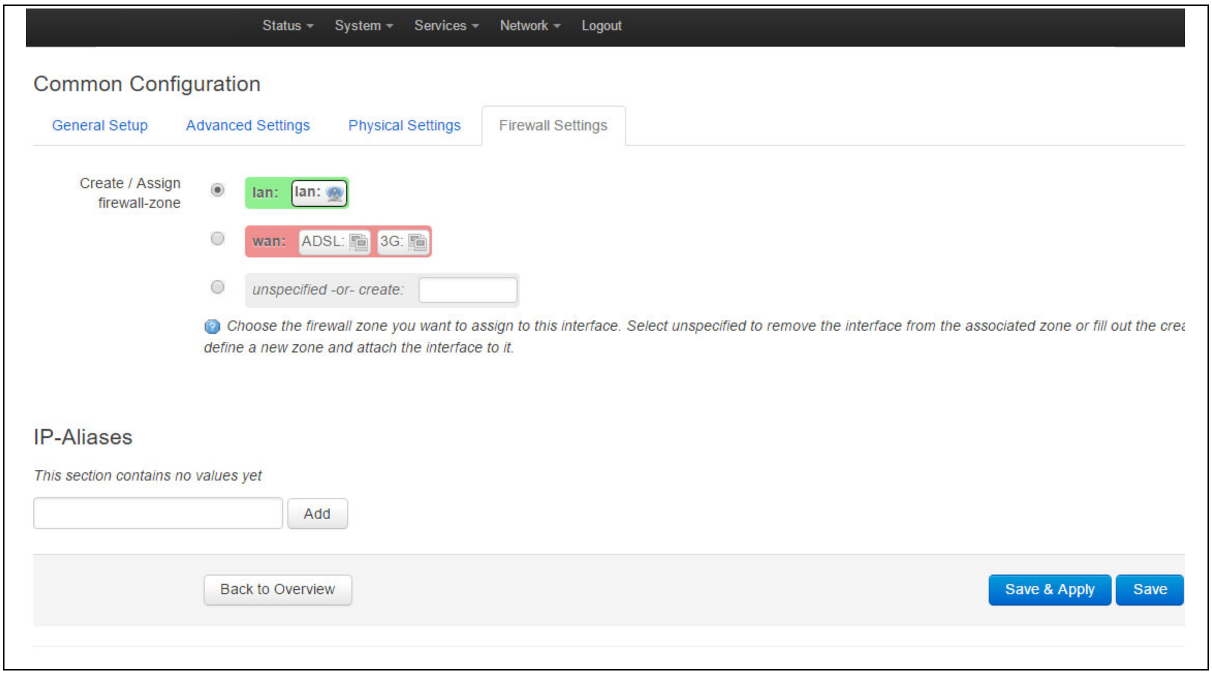


Figure 48: GRE firewall settings

10.2.4 Interface overview: IP-aliases

IP aliasing means associating more than one IP address to a network interface. You can assign multiple aliases.

10.2.4.1 IP-alias packages

Package	Sections
Network	alias

10.2.4.2 IP-alias using the web

To use IP-aliases, enter a name for the alias and click **Add**. This name will be assigned to the alias section for this IP-alias. In this example, we use the name 'ethalias1'.

Figure 49: The IP-Aliases section

Web Field / UCI / Package Option	Description
UCI: network.<alias name>=ifname Opt: config interface ‘aliasname’	Assigns the alias name.
UCI: network.<alias name>.interface Opt: interface	This maps the IP-Alias to the interface.
UCI: network.<alias name>.proto Opt: proto	This maps the interface protocol to the alias.

Table 19: Information table for IP-Aliases name assignment

The IP Aliases configuration options page appears. The IP-Alias is divided into two sub sections: general setup and advanced.

10.2.4.3 IP-aliases: general setup

Figure 50: The IP-Aliases general setup section

Web Field / UCI / Package Option	Description
Web: IPv4-Address UCI: network.<alias name>.ipaddr Opt: ipaddr	Defines the IP address for the IP alias.
Web: IPv4-Netmask UCI: network.<alias name>.netmask Opt: netmask	Defines the netmask for the IP alias.
Web: IPv4-Gateway UCI: network.<alias name>.gateway Opt: gateway	Defines the gateway for the IP alias.

Table 20: Information table for IP-Alias general setup page

10.2.4.4 IP-aliases: advanced settings

Figure 51: The IP-Aliases advanced settings section

Web Field / UCI / Package Option	Description
Web: IPv4-Broadcast UCI: network.<alias name>.bcast Opt: bcast	Defines the IP broadcast address for the IP alias.
Web: DNS-Server UCI: network.<alias name>.dns Opt: dns	Defines the DNS server for the IP alias.

Table 21: Information table for IP-Alias advanced settings page

10.2.5 Interface overview: DHCP server

Note: this option is only available for interfaces with a static IP address.

10.2.5.1 DHCP server: packages

Package	Sections
dhcp	dhcp

To assign a DHCP Server to the interface, click **Setup DHCP Server**.

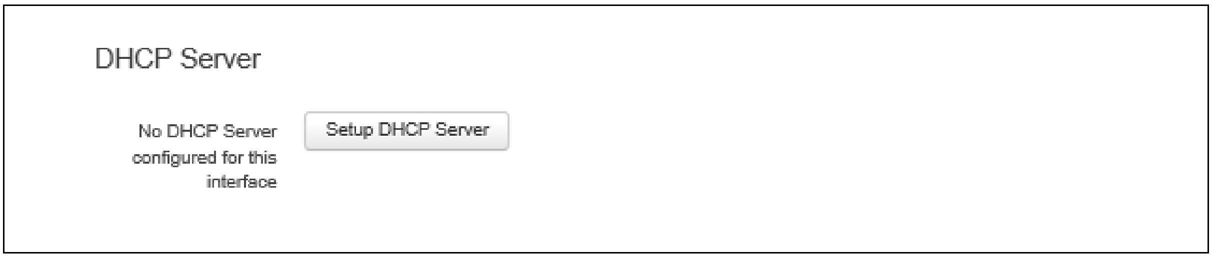


Figure 52: The DHCP Server settings section

The DHCP Server configuration options will appear. The DHCP Server is divided into two sub sections – general setup and advanced.

10.2.5.2 DHCP server: general setup

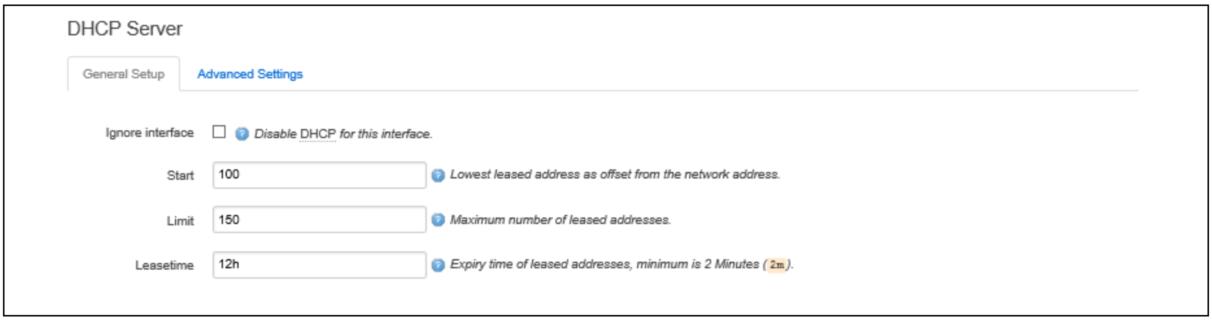


Figure 53: The DHCP server general setup section

Web Field / UCI / Package Option	Description
Web: Ignore interface UCI: dhcp.@dhcp[x].ignore Opt: ignore	Defines whether the DHCP pool should be enabled for this interface. If not specified for the DHCP pool then default is disabled i.e. dhcp pool enabled. 0 : Disabled. 1 : Enabled.
Web: n/a UCI: dhcp.@dhcp[x].start Opt: start	Defines the offset from the network address for the start of the DHCP pool. It may be greater than 255 to span subnets. 100 Range
Web: n/a UCI: dhcp.@dhcp[x].limit Opt: limit	Defines the offset from the network address for the end of the DHCP pool. 150 Range : 0 – 255
Web: n/a UCI: dhcp.@dhcp[x].leasetime Opt: leasetime	Defines the lease time of addresses handed out to clients, for example 12h or 30m. 12h : 12 hours Range

Table 22: Information table for DHCP server general setup page

10.2.5.3 DHCP server: advanced settings

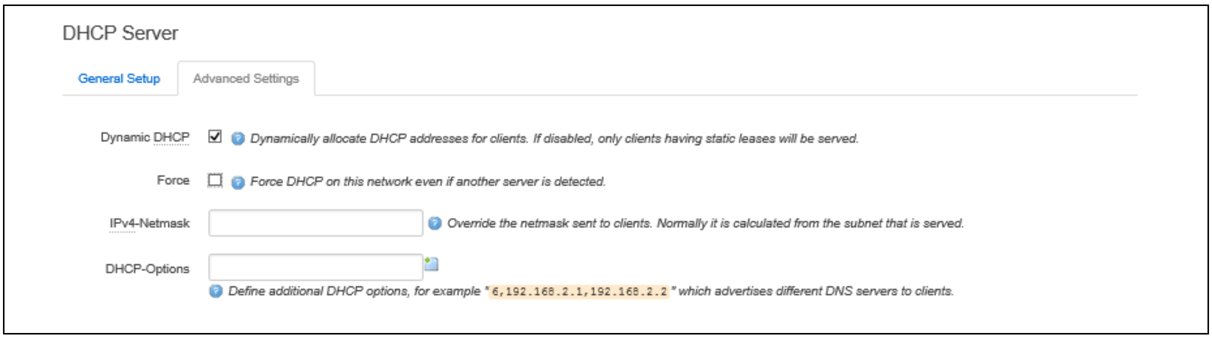


Figure 54: The DHCP server advanced settings section

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Dynamic DHCP
UCI: dhcp.@dhcp[x].dynamicdhcp
Opt: dynamicdhcp</td><td>Defines whether to allocate DHCP leases.<table><tr><td>1</td><td>Dynamically allocate leases.</td></tr><tr><td>0</td><td>Use /etc/ethers file for serving DHCP leases.</td></tr></table></td></tr>
<tr><td>Web: Force
UCI: dhcp.@dhcp[x].force
Opt: force</td><td>Forces DHCP serving on the specified interface even if another DHCP server is detected on the same network segment.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: DHCP-Options
UCI: dhcp.@dhcp[x].dhcp_option
Opt: list dhcp_option</td><td>Defines additional options to be added for this dhcp pool. For example with 'list dhcp_option 26,1470' or 'list dhcp_option mtu, 1470' you can assign a specific MTU per DHCP pool. Your client must accept the MTU option for this to work. Options that contain multiple vales should be separated by a space.
Example: list dhcp_option 6,192.168.2.1 192.168.2.2<table><tr><td></td><td>No options defined.</td></tr><tr><td>Syntax</td><td>Option_number, option_value</td></tr></table></td></tr>
<tr><td>Web: n/a
UCI: dhcp.@dhcp[x].networkid
Opt: networked</td><td>Assigns a network-id to all clients that obtain an IP address from this pool.</td></tr>
</table>

Table 23: Information table for DHCP advanced settings page

For more advanced configuration on the DHCP server, read ‘DHCP server and DNS configuration section.

10.3 Interface configuration using UCI

The configuration files are stored on **/etc/config/network, /etc/config/firewal**l and **/etc/config/dhcp**

```
root@VA_router:~# uci show network
      …..
network.newinterface=interface
network.newinterface.proto=static
network.newinterface.ifname=eth0
network.newinterface.monitored=0
network.newinterface.ipaddr=2.2.2.2
network.newinterface.netmask=255.255.255.0
network.newinterface.gateway=2.2.2.10
network.newinterface.broadcast=2.2.2.255
network.newinterface.vlan_qos_map_ingress=1:2 2:1
network.ethalias1=alias
network.ethalias1.proto=static
network.ethalias1.interface=newinterface
network.ethalias1.ipaddr=10.10.10.1
network.ethalias1.netmask=255.255.255.0
```

```
network.ethalias1.gateway=10.10.10.10
network.ethalias1.bcast=10.10.10.255
network.ethalias1.dns=8.8.8.8

root@VA_router:~# uci show firewall
      …..firewall.@zone[0]=zone
firewall.@zone[0].name=lan
firewall.@zone[0].input=ACCEPT
firewall.@zone[0].output=ACCEPT
firewall.@zone[0].forward=ACCEPT
firewall.@zone[0].network=lan newinterface

root@VA_router:~# uci show dhcp
      …
dhcp.@dhcp[0]=dhcp
dhcp.@dhcp[0].start=100
root@VA_router:~# uci show firewall
dhcp.@dhcp[0].leasetime=12h
dhcp.@dhcp[0].limit=150
dhcp.@dhcp[0].interface=newinterface
```

To change any of the above values use `uci set` command.

10.3.1 Interface common configuration using package options

The configuration files are stored on **/etc/config/network, /etc/config/firewall** and **/etc/config/dhcp**

```
root@VA_router:~# uci export network
package network
      ……
config interface 'newinterface'
        option proto 'static'
        option ifname 'eth0'
        option monitored '0'
        option ipaddr '2.2.2.2'
        option netmask '255.255.255.0'
        option gateway '2.2.2.10'
        option broadcast '2.2.2.255'
        list vlan_qos_map_ingress '1:2'
```

```
        list vlan_qos_map_ingress '2:1'

config alias 'ethalias1'
        option proto 'static'
        option interface 'newinterface'
        option ipaddr '10.10.10.1'
        option netmask '255.255.255.0'
        option gateway '10.10.10.10'
        option bcast '10.10.10.255'
        option dns '8.8.8.8'

root@VA_router:~# uci export firewall
package firewall
config zone
      option name 'lan'
      option input 'ACCEPT'
      option output 'ACCEPT'
      option forward 'ACCEPT'
      option network 'lan newinterface'

root@VA_router:~# uci export dhcp
package dhcp
      ……
config dhcp
        option start '100'
        option leasetime '12h'
        option limit '150'
        option interface 'newinterface'
```

To change any of the above values use `uci set` command.

10.3.2 Loopback interfaces

Loopback interfaces are defined in exactly the same way as Ethernet interfaces. Read the section above.

Note: There is no software limitation as to how many loopback interfaces can exist on the router.

An example showing a partial uci export of a loopback interface configuration is shown below.

```
root@VA_router:~# uci export network
      …..
config interface 'loopback'
        option proto 'static'
        option ifname 'lo'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'
```

10.4 Configuring port maps

10.5 Port map packages

Package	Sections
Network	va_switch

10.5.1 Configuring port map using the web interface

The new logical Ethernet interface needs to be mapped to a physical switch port. To configure the Ethernet switch physical port to logical interface mappings, go to the Port Map section at **Network->Interfaces**.

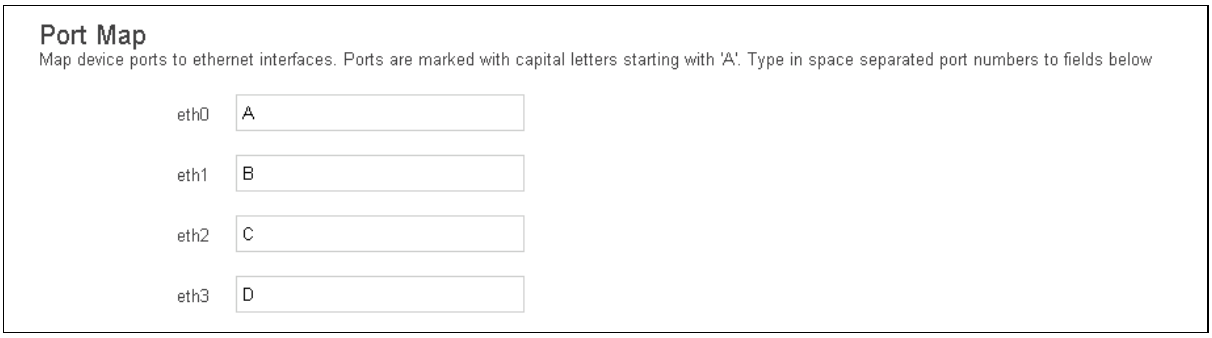


Figure 55: The Interface port map section

Web Field / UCI / Package Option	Description
Web: eth0 UCI: network.@va_switch[0].eth0 Opt: eth0	Defines eth0 physical switch port mapping. Must be entered in upper case. A — Eth0 assigned to switch port A B — Eth0 assigned to switch port B C — Eth0 assigned to switch port C D — Eth0 assigned to switch port C
Web: eth1 UCI: network.@va_switch[0].eth1 Opt: eth1	Defines eth1 physical switch port mapping. Must be entered in upper case. A — Eth1 assigned to switch port A B — Eth1 assigned to switch port B C — Eth1 assigned to switch port C D — Eth1 assigned to switch port C
Web: eth2 UCI: network.@va_switch[0].eth2 Opt: eth2	Defines eth0 physical switch port mapping. Must be entered in upper case. A — Eth2 assigned to switch port A B — Eth2 assigned to switch port B C — Eth2 assigned to switch port C D — Eth2 assigned to switch port C
Web: eth3 UCI: network.@va_switch[0].eth3 Opt: eth3	Defines eth0 physical switch port mapping. Must be entered in upper case. A — Eth3 assigned to switch port A B — Eth3 assigned to switch port B C — Eth3 assigned to switch port C D — Eth3 assigned to switch port C

Table 24: Information table for interface port map page

10.5.2 Configuring port maps using UCI

The configuration files are stored on **/etc/config/network**

```
root@VA_router:~# uci show network
      ……
network.@va_switch[0]=va_switch
network.@va_switch[0].eth0=A
network.@va_switch[0].eth1=B
network.@va_switch[0].eth2=C
network.@va_switch[0].eth3=D
```

To change any of the above values use `uci set` command.

10.5.3 Configuring port map using package options

The configuration files are stored on **/etc/config/network**

```
root@VA_router:~# uci export network
      …..
config va_switch
        option eth0 'A'
        option eth1 'B'
        option eth2 'C'
        option eth3 'D'
```

To change any of the above values use `uci set` command.

10.5.4 ATM bridges

The ATM bridges section is not used when configuring an Ethernet interface.

10.6 Interface diagnostics

10.6.1 Interfaces status

To show the current running interfaces, enter:

```
root@VA_router:~# ifconfig
3g-CDMA   Link encap:Point-to-Point Protocol
          inet addr:10.33.152.100  P-t-P:178.72.0.237  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1400  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:428 (428.0 B)  TX bytes:2986 (2.9 KiB)

eth0      Link encap:Ethernet  HWaddr 00:E0:C8:12:12:15
          inet addr:192.168.100.1  Bcast:192.168.100.255
Mask:255.255.255.0
          inet6 addr: fe80::2e0:c8ff:fe12:1215/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6645 errors:0 dropped:0 overruns:0 frame:0
          TX packets:523 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:569453 (556.1 KiB)  TX bytes:77306 (75.4 KiB)
```

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:385585 errors:0 dropped:0 overruns:0 frame:0
          TX packets:385585 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:43205140 (41.2 MiB)  TX bytes:43205140 (41.2 MiB)
```

To display a specific interface, enter:

```
root@VA_router:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:E0:C8:12:12:15
          inet addr:192.168.100.1  Bcast:192.168.100.255
Mask:255.255.255.0
          inet6 addr: fe80::2e0:c8ff:fe12:1215/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7710 errors:0 dropped:0 overruns:0 frame:0
          TX packets:535 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:647933 (632.7 KiB)  TX bytes:80978 (79.0 KiB)
```

10.6.2 ARP table status

To show the current ARP table of the router, enter:

```
root@GW7314:~# arp
? (10.67.253.141) at 30:30:41:30:43:36 [ether]  on eth8
? (10.47.48.1) at 0a:44:b2:06 [ether]  on gre-gre1
```

10.6.3 Route status

To show the current routing status, enter:

```
root@VA_router:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.100.0   *               255.255.255.0   U     0      0        0 eth0
```

Note: a route will only be displayed in the routing table when the interface is up.

11 Configuring VLAN

11.1 Maximum number of VLANs supported

Virtual Access' routers support up to 4095 VLANs.

11.2 Configuration package used

Package	Sections
Network	

11.3 Configuring VLAN using the web interface

11.3.1 Create a VLAN interface

To configure VLAN using the web interface, in the top menu, select **Network ->Interfaces**.

Click **Add** new interface. The Create Interface page appears.

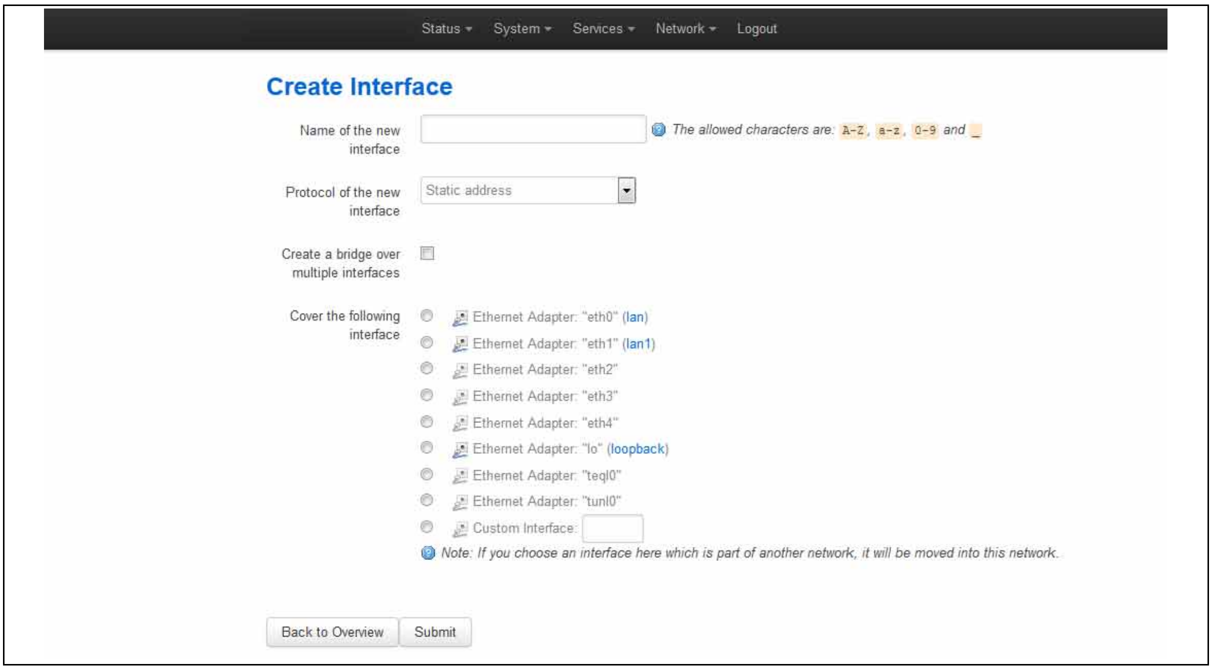


Figure 56: The create interface page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface
UCI: network.vlan1=interface
Opt: interface</td><td>Type the name of the new interface. For example, VLAN1.</td></tr>
<tr><td>Web: Protocol of the new interface
UCI: network.vlan_test.proto
Opt: proto</td><td>Protocol type. Select Static.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td></tr>
<tr><td>PPP</td><td>Point to Point Protocol</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: Create a bridge over multiple interfaces
UCI: network.vlan1.type
Opt: type</td><td>Create a bridge over multiple interfaces.</td></tr>
<tr><td>Web: Cover the following interface
UCI: network.vlan1.ifname
Opt: ifname</td><td>Check the Custom Interface radio button.
Enter a name, for example eth0.100. This will assign VLAN 100 to the eth0 interface.</td></tr>
</table>

Table 25: Information table for the create interface page

Click **Submit**. The Interfaces page for VLAN1 appears.

11.3.2 General setup: VLAN

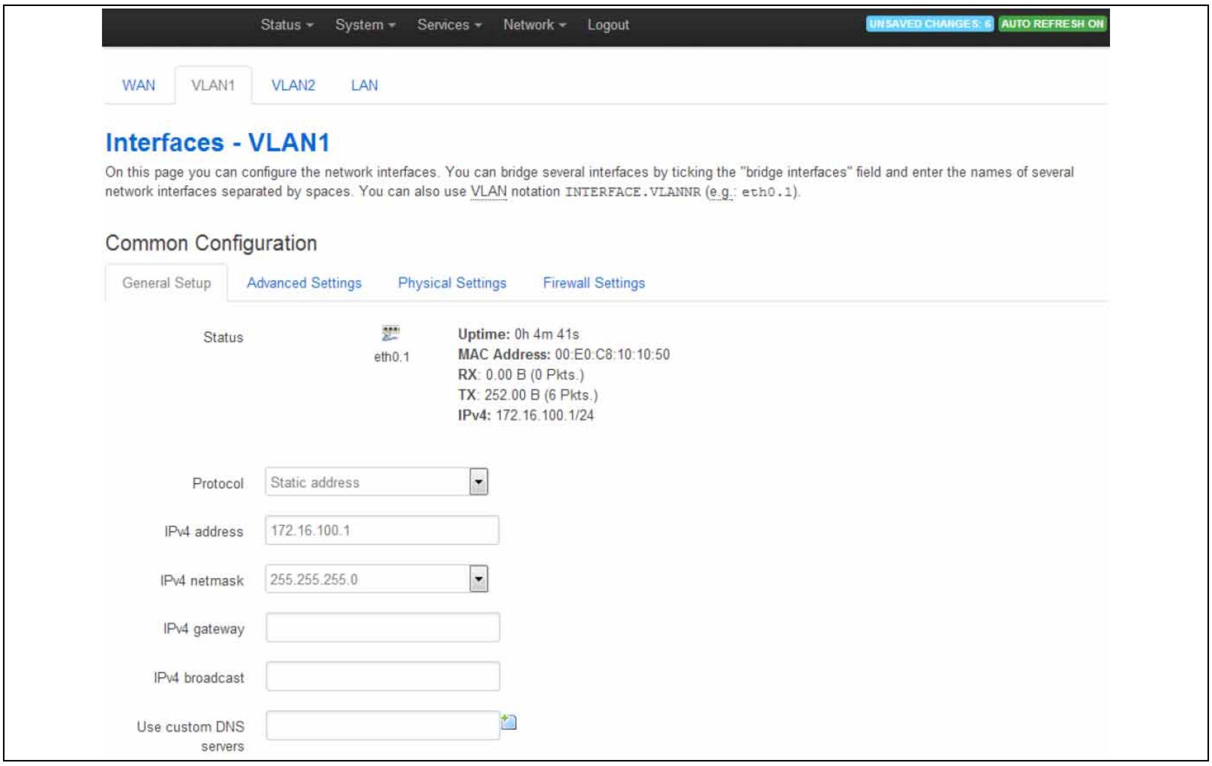


Figure 57: The VLAN 1 interface page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Protocol
UCI: network.VLAN1.proto
Opt: proto</td><td>Protocol type.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td></tr>
<tr><td>PPP</td><td>Point to Point Protocol</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: IPv4 address
UCI: network.VLAN1.ipaddr
Opt: ipaddr</td><td>The IPv4 address of the interface. This is optional if an IPv6 address is provided.</td></tr>
<tr><td>Web: IPv4 netmask
UCI: network.VLAN1.netmask
Opt: netmask</td><td>Subnet mask to be applied to the IP address of this interface.</td></tr>
</table>

Web: IPv4 gateway UCI: network.VLAN1.gateway Opt: gateway	IPv4 default gateway to assign to this interface (optional).
Web: Use custom DNS servers UCI: network.VLAN1.dns Opt: dns	List of DNS server IP addresses (optional).

Table 26: Information table for VLAN general settings

11.3.3 Firewall settings: VLAN

Use this section to select the firewall zone you want to assign to the VLAN interface.

Select **unspecified** to remove the interface from the associated zone or fill out the create field to define a new zone and attach the interface to it.

Figure 58: Firewall settings page

When you have added all the VLAN interfaces you require, click **Save & Apply**.

11.4 Viewing VLAN interface settings

To view the new VLAN interface settings, in the top menu, select **Network -> Interfaces**. The Interfaces Overview page appears.

The example below shows two VLAN interfaces configured.

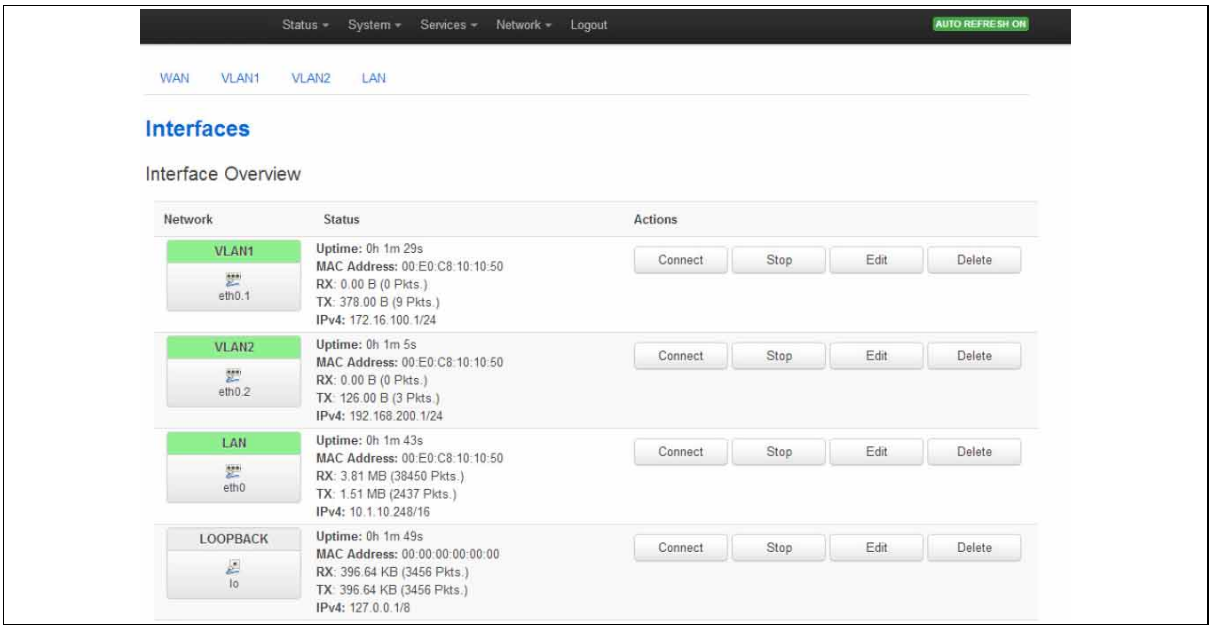


Figure 59: The interface overview page showing two VLAN interfaces

11.5 Configuring VLAN using the UCI interface

You can configure VLANs through CLI. The VLAN configuration file is stored on: **/etc/config/network**

```
# uci export network
package network
config interface 'vlan100'
        option proto 'static'
        option ifname 'eth0.100'
        option monitored '0'
        option ipaddr '192.168.100.1'
        option netmask '255.255.255.0'
        option gateway '192.168.100.10'
        option broadcast '192.168.100.255'
        option dns '8.8.8.8'
```

Modify these settings by running `uci set <parameter>` command.

When specifying the ifname ensure that it is written in dotted mode, that is, eth1.100 where eth1 is the physical interface assigned to VLAN tag 100.

Note: VLAN1 is, by default the native VLAN and will not be tagged.

12 Configuring a mobile connection

12.1 Configuration package used

Package	Sections
network	interface

12.2 Configuring a mobile connection using the web interface

Note: if you are creating multiple mobile interfaces, simply repeat the steps in this chapter for each interface. Multiple interfaces are required for dual SIM or multiple radio module scenarios. Configuring static routes and/or Multi-WAN can be used to manage these interfaces.

In the top menu, select **Network -> Interfaces**. The Interfaces Overview page appears.

12.2.1 Create a new mobile interface

To create a new mobile interface, in the Interface Overview section, click **Add new interface**. The Create Interface page appears. In the examples below, 3G has been used for the interface name.

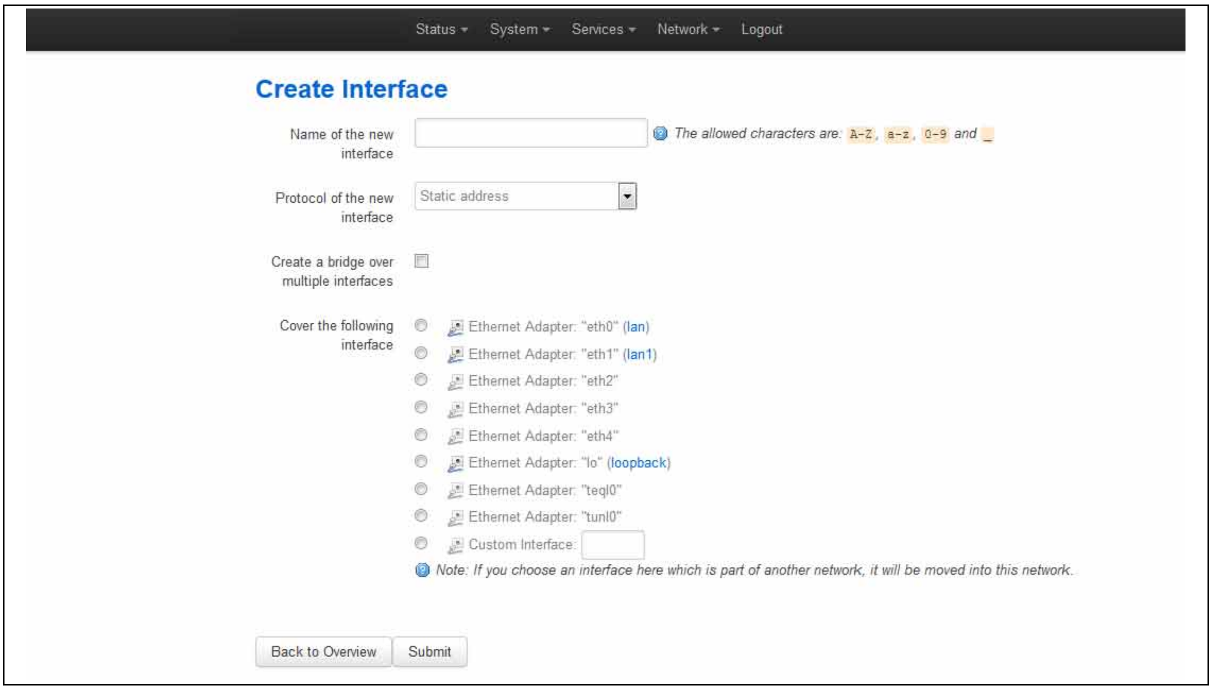


Figure 60: The create interface page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface
UCI: network.3G=interface
Opt: interface</td><td>Allowed characters are A-Z, a-z, 0-9 and _</td></tr>
<tr><td>Web: Protocol of the new interface
UCI: network.3G.proto
Opt: proto</td><td>Protocol type. Select LTE / UMTS / GPRS / EV-DO.
<table>
<tr><td>Option</td><td>Description</td></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4</td><td></td></tr>
<tr><td>IPv6-over-IPv4</td><td></td></tr>
<tr><td>GRE</td><td></td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol.</td></tr>
<tr><td>PPP</td><td></td></tr>
<tr><td>PPPoE</td><td></td></tr>
<tr><td>PPPoATM</td><td></td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: Create a bridge over multiple interfaces
UCI: network.3G.type
Opt: type</td><td>Enables bridge between two interfaces.
Not relevant when configuring a mobile interface.
<table>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
</table></td></tr>
<tr><td>Web: Cover the following interface
UCI: network.3G.ifname
Opt: ifname</td><td>Select interfaces for bridge connection.
Not relevant when configuring a mobile interface.</td></tr>
</table>

Table 27: Information table for the create interface page

Click **Submit**. The Common Configuration page appears. There are three sections in the mobile interface common configurations:

Section	Description
General Setup	Configure the basic interface settings such as protocol, service type, APN information, user name and password.
Advanced Settings	Set up more in-depth features such as initialisation timeout, LCP echo failure thresholds and inactivity timeouts.
Firewall settings	Assign a firewall zone to the connection.

12.2.1.1 Mobile interface: general setup

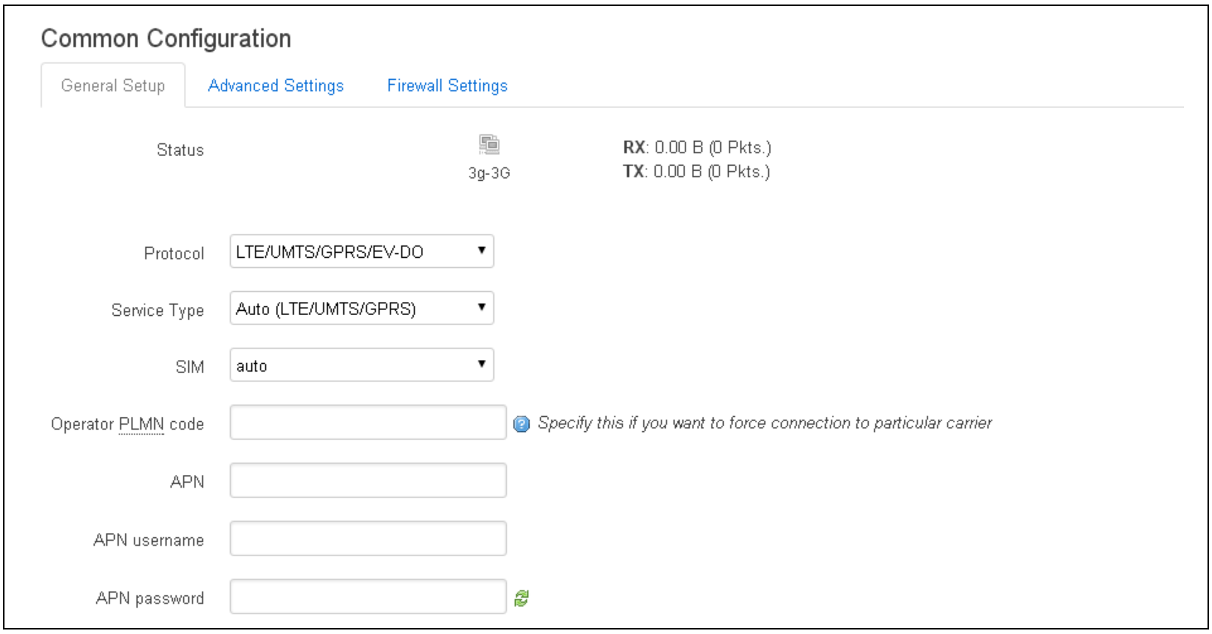


Figure 61: The common configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Status
UCI: n/a
Opt: n/a</td><td>Shows the current status of the interface.</td></tr>
<tr><td>Web: Protocol
UCI: network.3G.proto
Opt: proto</td><td>Protocol type. Select LTE / UMTS / GPRS / EV-DO.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>GRE</td><td></td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol.</td></tr>
<tr><td>PPP</td><td></td></tr>
<tr><td>PPPoE</td><td></td></tr>
<tr><td>PPPoATM</td><td></td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: Service Type
UCI: network.3G.service
Opt: service</td><td>Service type that will be used to connect to the network.
<table>
<tr><td>gprs_only</td><td>Allows GSM module to only connect to GPRS network.</td></tr>
<tr><td>lte_only</td><td>Allows GSM module to only connect to LTE network.</td></tr>
<tr><td>cdma</td><td>Allows GSM module to only connect to CDMA network.</td></tr>
<tr><td>auto</td><td>GSM module will automatically detect the best available technology code.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Web: n/a
UCI: network.3G.service_order
Opt: service_order</td><td>Defines a space separated list of services, in preferred order. Valid options are <code>gprs, umts, lte, auto</code>.
If no valid_service order is defined, then the configured Service Type is used. Example:
<code>network.3G.service_order="gprs umts lte auto"</code><table><tr><td>Blank</td><td>Use configured service type.</td></tr><tr><td>Range</td><td>gprs umts lte auto</td></tr></table></td></tr>
<tr><td>Web: Operator PLMN code
UCI: network.3G.operator
Opt: operator</td><td>Specifies an operator PLMN code to force the connection to a particular carrier. The PLMN code is identified as a combination of the MCC and the MNC.
Note: the operator option is used in conjunction with the operator format option <code>option opformat</code> which is used to define how the operator string is parsed. If configuring via the web GUI the op format is automatically set to '2' to indicate it is a PLMN code.
See below for alternative options for the operator format option.</td></tr>
<tr><td>Web: n/a
UCI: network.3G.opformat
Opt: opformat</td><td>Defines the operator format. We recommended you use PLMN code.
The operator is case sensitive so if using long or short character format it must match the operator exactly.
To see the current operator using SSH enter the command: cat /var/state/mobile or using the web mobile stats page at Status -> Mobile Stats.<table><tr><td>0</td><td>Long character format</td></tr><tr><td>1</td><td>Short character format</td></tr><tr><td>2</td><td>PLMN code</td></tr></table></td></tr>
<tr><td>Web: SIM
UCI: network.3G.sim
Opt: sim</td><td>Defines which SIM is used on this interface.<table><tr><td>1</td><td>SIM 1</td></tr><tr><td>2</td><td>SIM 2</td></tr><tr><td>any</td><td>Automatically detect</td></tr></table></td></tr>
<tr><td>Web: APN
UCI: network.3G.apn
Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: APN username
UCI: network.3G.username
Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: APN password
UCI: network.3G.password
Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: n/a
UCI: network.3G.retry_interval_sec
Opt: retry_interval_sec</td><td>Alllows you to specify exact integer or range that will be used to calculate random number to delay PPP connection.<table><tr><td>0</td><td>PPP will connect immediately, without any delay.</td></tr><tr><td>1-infinite</td><td>PPP will attempt to connect again after specified interval.</td></tr><tr><td>Range</td><td>PPP will attempt to connect within specified range. The exact interval is calculated randomly from specified range.
retry_interval_sec 60 180</td></tr></table></td></tr>
</table>

Table 28: Information table for common configuration settings

The Modem Configuration link at the bottom of the page is used for SIM pin code and SMS configuration. For more information, read the chapter 'Configuring mobile manager'.

12.2.1.2 Mobile interface: advanced settings

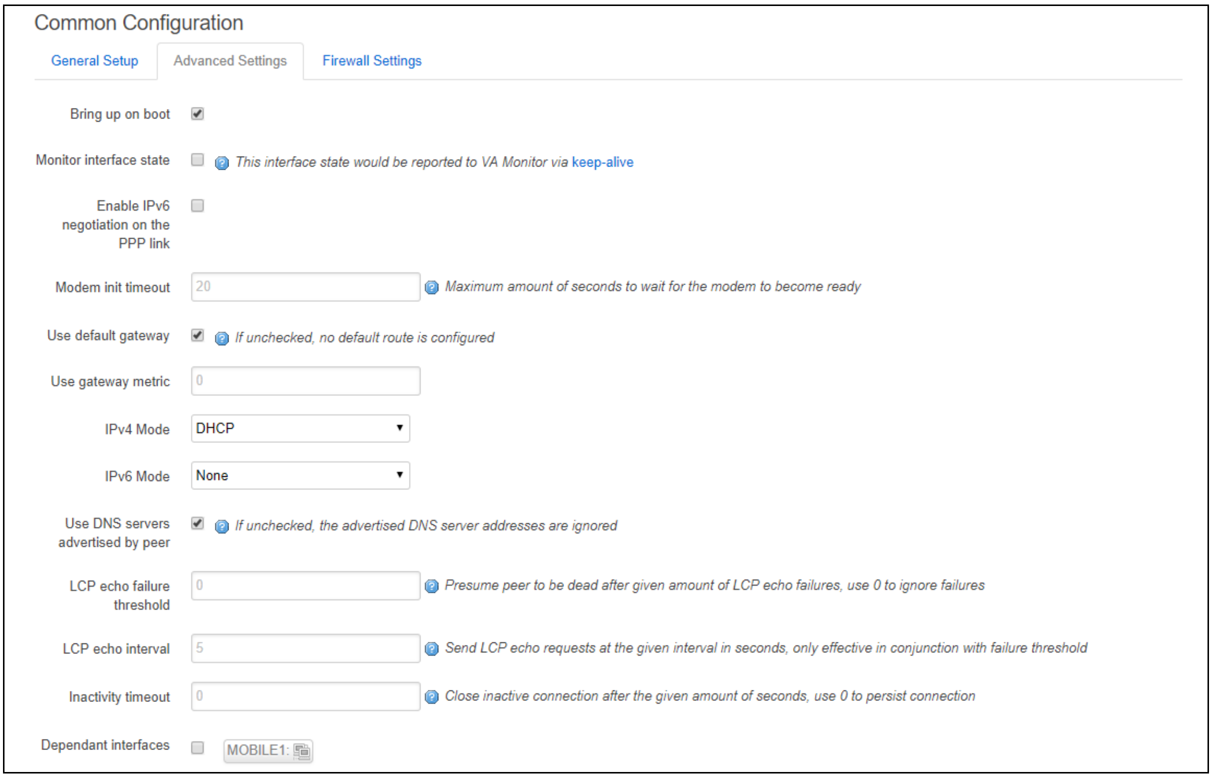


Figure 62: The advanced settings tab

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web:Bring up on boot
UCI: network.3G.auto
Opt: auto</td><td>Enables the interface to connect automatically on boot up or reconnect automatically when disconnected.</td></tr>
<tr><td>Web: Monitor interface state
UCI: network.3G.monitored
Opt: monitored</td><td>Enabled if status of interface is presented on Monitoring platform.<table><tr><td>0</td><td>Do not monitor interface.</td></tr><tr><td>1</td><td>Monitor interface.</td></tr></table></td></tr>
<tr><td>Web: Enable IPv6 negotiation on the PPP link
UCI: network.3G.ipv6
Opt: ipv6</td><td>Enables IPv6 routing on the interface.<table><tr><td>0</td><td>Do not enable IPv6.</td></tr><tr><td>1</td><td>Enable IPv6.</td></tr></table></td></tr>
<tr><td>Web: Modem int timeout
UCI: network.3G.maxwait
Opt: maxwait</td><td>Maximum amount of seconds to wait for the modem to become ready.<table><tr><td>20</td><td>Seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Use default gateway
UCI: network.3G.defaultroute
Opt: defaultroute</td><td>Enables this interface as a default route<table><tr><td>0</td><td>Do not use as a default route.</td></tr><tr><td>1</td><td>Use as a default route.</td></tr></table></td></tr>
<tr><td>Web: Use gateway metric
UCI: network.3G.metric
Opt: metric</td><td>Defines the metric for the default route. Lower number metrics are used first when route is up.<table><tr><td>0</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

<table>
<tr><td>Web: IPv4 Mode
UCI: network.3G.ipv4mode
Opt: ipv4mode</td><td>Defines the IPv4 address assignment approach for mobile interfaces in Ethernet Mode. Note: by default, mobile interfaces are in Ethernet mode.
<table>
<tr><th>Option</th><th>Description</th><th>UCI</th></tr>
<tr><td>None</td><td>No dynamic assignment</td><td>none</td></tr>
<tr><td>DHCP</td><td>DHCP address assignment</td><td>dhcp</td></tr>
</table></td></tr>
<tr><td>Web: IPv6 Mode
UCI: network.3G.ipv6mode
Opt: ipv6mode</td><td>Defines the IPv6 address assignment approach for mobile interfaces in Ethernet Mode.
Note: by default, mobile interfaces are in Ethernet mode.
<table>
<tr><th>Option</th><th>Description</th><th>UCI</th></tr>
<tr><td>None</td><td>No dynamic assignment</td><td>none</td></tr>
<tr><td>DHCPv6</td><td>DHCP address assignment</td><td>dhcp</td></tr>
<tr><td>RA</td><td>Router Advertisement (RA) assignment</td><td>ra</td></tr>
<tr><td>DHCPv6 after RA</td><td>Wait for RA, then start DHCP</td><td>ra_then_dhcp</td></tr>
</table></td></tr>
<tr><td>Web: Use DNS servers advertised by peer
UCI: network.3G.peerdns
Opt: peerdns</td><td>If unchecked, the advertised DNS server addresses are ignored.
<table>
<tr><td>0</td><td>Use static DNS</td></tr>
<tr><td>1</td><td>Use advertised DNS</td></tr>
</table></td></tr>
<tr><td>Web: Use custom DNS servers
UCI: network.3G.dns
Opt: dns</td><td>Specifies DNS server. Only available if Use DNS servers advertised by peer is unselected. When multiple DNS servers are required separate using space for UCI or option value. Example:
<code>uci set network.3G.dns='1.1.1.1 2.2.2.2'</code></td></tr>
<tr><td>Web: LCP echo failure threshold
UCI: network.3G.keepalive
Opt: keepalive</td><td>Presumes peer to be dead after a given amount of LCP echo failures, use 0 to ignore failures.
This command is used in conjunction with the LCP echo interval. The syntax is as follows: uci network.3G.keepalive=<echo failure threshold> <echo interval>
Example:
<code>uci set network.3G.keepalive=15 10</code></td></tr>
<tr><td>Web: LCP echo internal
UCI: network.3G.keepalive
Opt: keepalive</td><td>Send LCP echo requests at the given interval in seconds, only effective in conjunction with failure
This command is used in conjunction with the LCP echo failure threshold. The syntax is as follows: uci network.3G.keepalive=<echo failure threshold> <echo interval> Example:
<code>uci set network.3G.keepalive=15 10</code></td></tr>
<tr><td>Web: Inactivity timeout
UCI: network.3G.demand
Opt: demand</td><td>Closes an inactive connection after the given amount of seconds. Use 0 to persist connection.
<table>
<tr><td>0</td><td>Do not disconnect on inactivity</td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Dependant Interfaces
UCI: network.3G.dependants
Opt: dependants</td><td>Lists interfaces that are dependent on this parent interface. Dependant interfaces will go down when the parent interface is down and will start or restart when the parent interface starts.
Separate multiple interfaces by a space when using UCI. Example: <code>option dependants 'PPPADSL MOBILE'</code>
This replaces the following previous options in child interfaces.
<table>
<tr><td>gre</td><td>option local_interface</td></tr>
<tr><td>lt2p</td><td>option src_ipaddr</td></tr>
<tr><td>iot</td><td>option wan1 wan2</td></tr>
<tr><td>6in4</td><td>option ipaddr</td></tr>
<tr><td>6to4</td><td>option ipaddr</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Web: SNMP Alias ifindex
UCI: network.[..x..].snmp_alias_ifindex
Opt: snmp_alias_ifindex</td><td>Defines a static SNMP interface alias index for this interface that can be polled via the SNMP interface index. (snmp_alias_ifindex+1000). See Configuring SNMP section for more information.<table><tr><td>Blank</td><td>No SNMP interface alias index</td></tr><tr><td>Range</td><td>0 - 4294966295</td></tr></table></td></tr>
</table>

Table 29: Information table for general set up page

12.2.1.3 Mobile interface: firewall settings

Use this section to select the firewall zone you want to assign to the interface.

Select **unspecified** to remove the interface from the associated zone or fill out the create field to define a new zone and attach the interface to it.

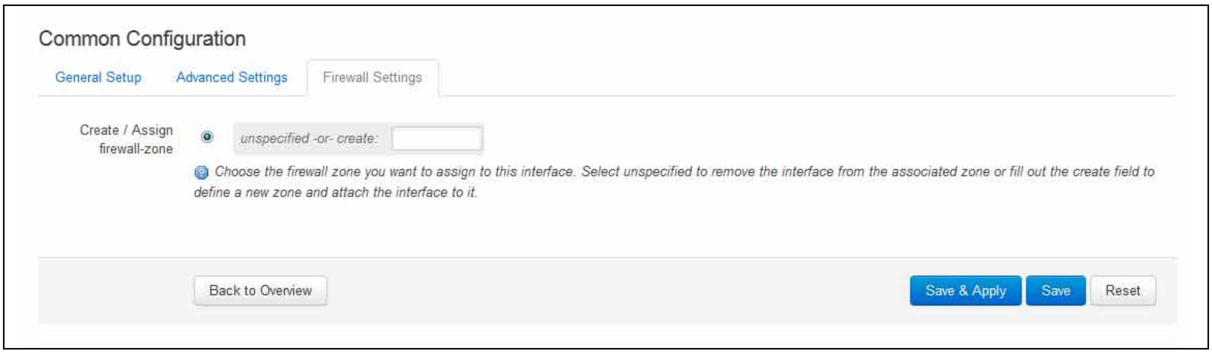


Figure 63: Firewall settings page

12.3 Configuring a mobile connection using CLI

12.3.1 UCI

To establish a basic mobile connection, enter:

```
root@VA_router:~# uci show network
network.3G=interface
network.3G.proto=3g
network.3G.monitored=0
network.3G.sim=any
network.3G.auto=1
network.3G.defaultroute=1
network.3G.metric=1
network.3G.service=autonetwork.3G.apn=test.apn
network.3G.username=username
network.3G.password=password
network.3G.ipv4mode=dhcp
network.3G.ipv6mode=none
```

12.3.2 Package options

```
root@VA_router:~#
package network

config interface '3G'
      option proto '3g'
      option monitored '0'
      option auto ‘1’
      option sim 'any'
      option defaultroute '1'
      option metric ‘1’       option service 'auto'
      option apn 'test.apn'
      option username ‘username’
      option password 'password'
      option ipv4mode ‘dhcp’
      option ipv6mode ‘none’
```

12.4 Diagnositcs

Note: the information presented on screen and data output using UCI depends on the actual mobile hardware being used. Therefore, the interfaces or output you see may differ from the samples shown here.

12.4.1 Mobile status via the web

To view mobile connectivity information, in the top menu, select **Status -> Mobile Information**. The Mobile Information page appears. The information presented depends on the actual mobile hardware used; it might therefore differ from the samples shown here.

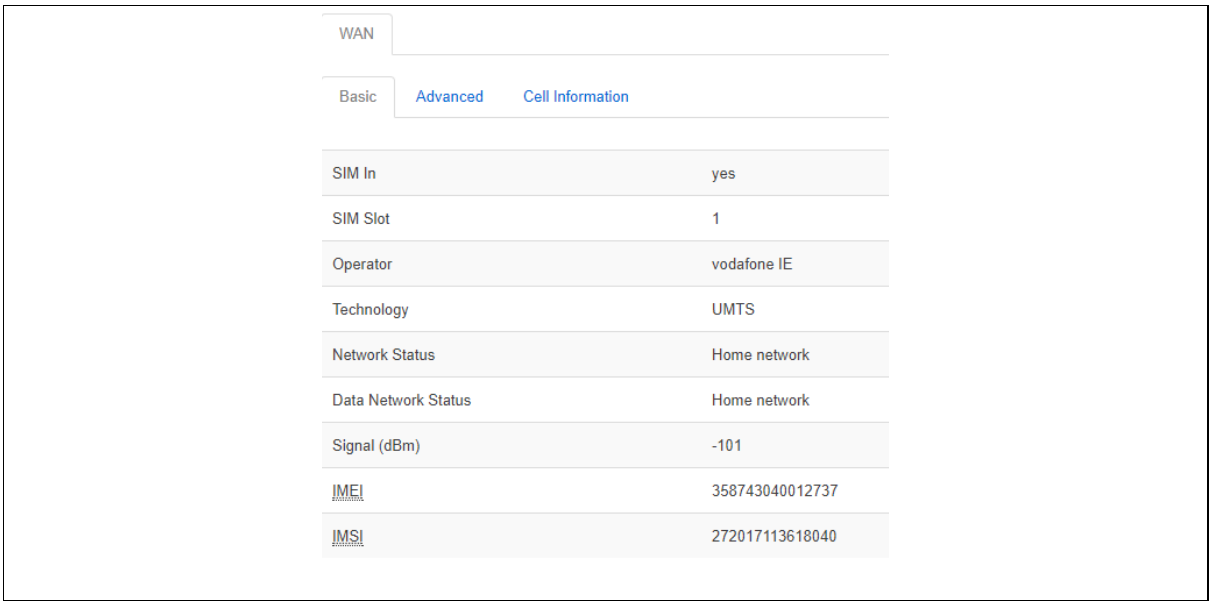


Figure 64: The mobile information page

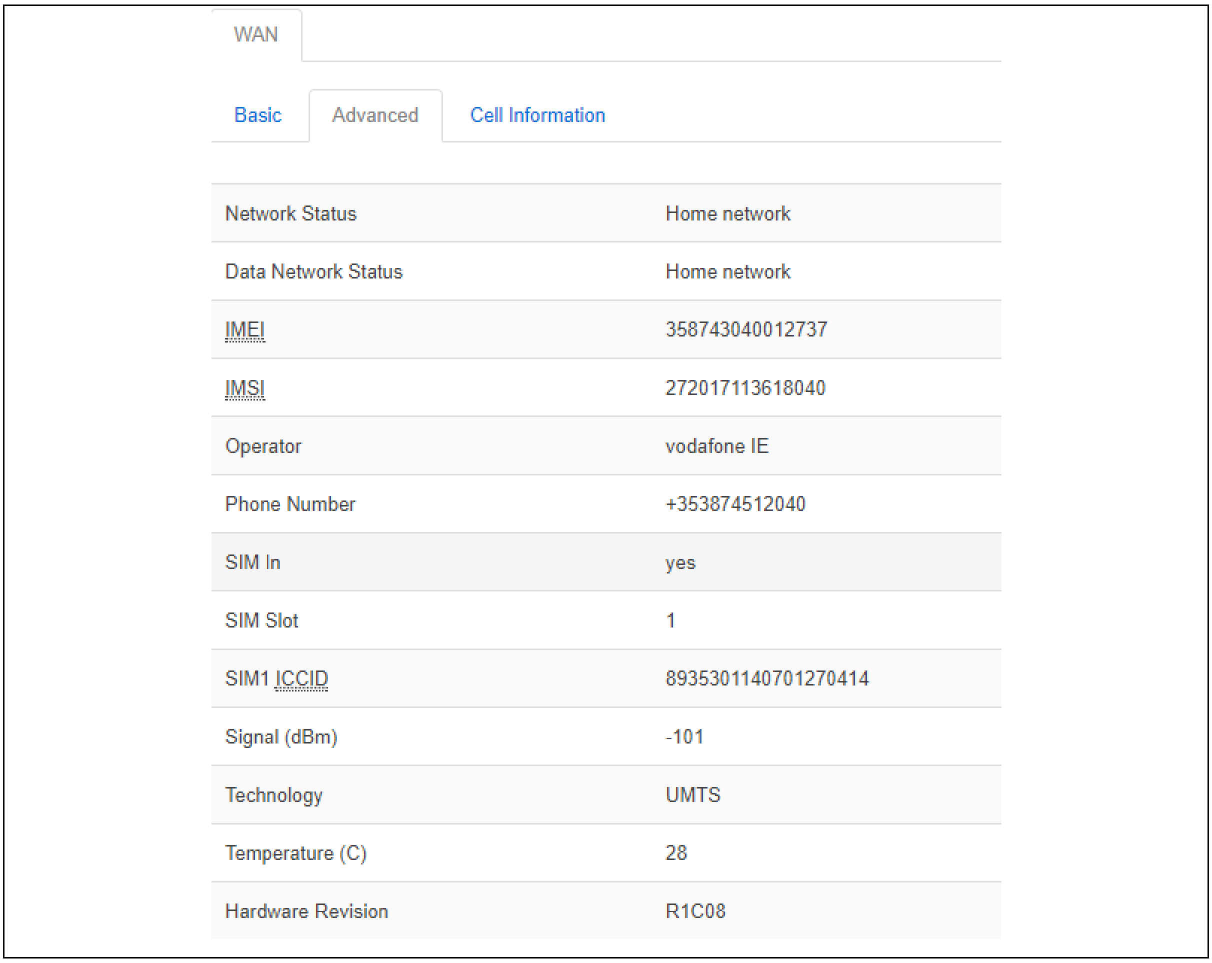


Figure 65: The advanced information page

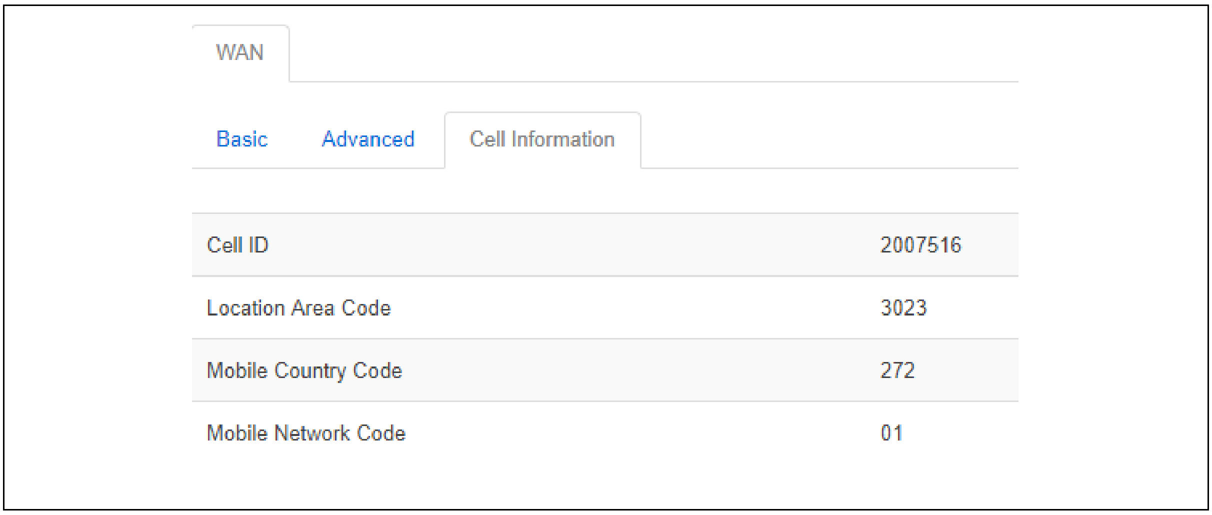


Figure 66: The cell information page

12.4.2 Mobile status using UCI

To display information and status of mobile interfaces such as 3G, 4G or CDMA, enter mobile_status:

```
root@VA_router:~# mobile_status

Mobile Interface     : WAN
Status               : idle
SIM In               : yes
SIM Slot             : 1
Operator             : vodafone IE
Technology           : UMTS
CS Network Status    : Home network
PS Network Status    : Home network
Signal (dBm)         : -107
IMEI                 : 358743040012737
IMSI                 : 272017113618040
```

For more advanced information, enter mobile_status –a:

```
root@ VA_router:~# mobile_status -a

Mobile Interface     : WAN
Status               : idle
CS Network Status    : Home network
```

```
PS Network Status   : Home network
IMEI                : 358743040012737
IMSI                : 272017113618040
Operator            : vodafone IE
Phone Number        : +353874512040
SIM In              : yes
SIM Slot            : 1
SIM1 ICCID          : 89353011407012704l4
Signal (dBm)        : -107
Technology          : UMTS
Temperature (C)     : 28
Hardware Revision   : R1C08
```

13 Configuring mobile manager

The Mobile Manager feature allows you to configure SIM settings.

13.1 Configuration package used

Package	Sections
mobile	Main
	Callers
	Roaming template

13.2 Configuring mobile manager using the web interface

Select **Services -> Mobile Manager**. The Mobile Manager page appears.

There are four sections in the mobile manager page:

Section	Description
Basic settings	Enable SMS, configure SIM pin code, select roaming SIM, collect ICCCIDs and set IMSI.
CDMA*	CDMA configuration.
Callers	Configure callers that can use SMS.
Roaming Interface Template	Configure Preferred Roaming List options.
*Option available only for Telit CE910-SL module.	

13.2.1 Mobile manager: basic settings

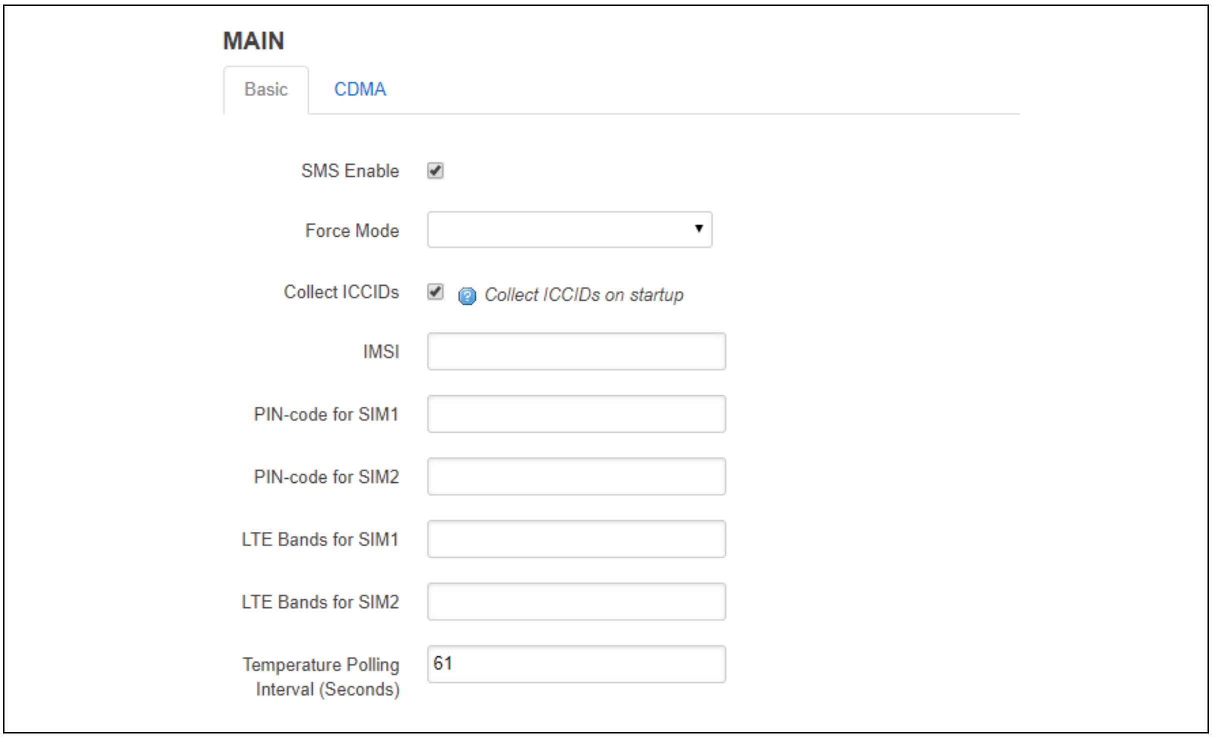


Figure 67: The mobile manager basic page

Web Field / UCI / Package Option	Description
Web: SMS Enable UCI: mobile.main.sms Opt: sms	Enables or disables SMS functionality. 0: Disabled. 1: Enabled.
Web: Force Mode UCI: mobile.main.force_mode Opt: force_mode	Defines whether to operate mobile modem in TTY or Ethernet mode. The mode will be dependent on the service provided by the mobile provider. In general, this is Ethernet mode (default). Blank: Ethernet mode (option not present). tty: Enable TTY mode.
Web: Collect ICCIDs UCI: mobile.main.init_get_iccids Opt: init_get_iccids	Enables or disables integrated circuit card identifier ICCID's collection functionality. If enabled then both SIM 1 and SIM 2 ICCIDs will be collected otherwise it will default to SIM 1. This will be displayed under mobile stats. 0: Disabled. 1: Enabled.
Web: IMSI UCI: mobile.main.imsi Opt: imsi	Allows the IMSI (International Mobile Subscriber Identity) to be changed. Default: Programmed in module. Digits: Up to 15 digits.
Web: PIN code for SIM1 UCI: mobile.main.sim1pin Opt: sim1pin	Depending on the SIM card specify the pin code for SIM 1. Blank Range: Depends on the SIM provider.
Web: PIN code for SIM2 UCI: mobile.main.sim2pin Opt: sim2pin	Depending on the SIM card specify the pin code for SIM 2. Blank Range: Depends on the SIM provider.
Web: LTE bands for SIM1 UCI: mobile.main.sim1_lte_bands Opt: sim1_lte_bands	Depending on the SIM card specify the LTE bands for SIM 1. Comma delimiter. Example: option sim1_lte_bands '3,20' Limits LTE bands to 3 and 20. **Note**: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200 and Asiatel. Blank Range: LTE bands range from 1 to 70.
Web: LTE bands for SIM2 UCI: mobile.main.sim2_lte_bands Opt:sim2_lte_bands	Depending on the SIM card specify the LTE bands for SIM 2. Comma delimiter. Example: option sim1_lte_bands '3,20' Limits LTE bands to 3 and 20. **Note**: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200 and Asiatel. Blank Range: LTE bands range from 1 to 70.
Web: Temperature Polling Interval UCI: mobile.main.temp_poll_interval_sec Opt: temp_poll_interval_sec	Defines the time in seconds to poll the mobile module for temperature. Set to **0** to disable. 61: 61 seconds. Range
Web: n/a UCI: mobile.main.disable_time Opt: disable_time	Defines whether to use time obtained from the mobile carrier to update the system clock when NTP is enabled. 0: Disabled. 1: Enabled.

Web: n/a UCI: mobile.main.service_order Opt: service_order	Defines a space separated list of services, in preferred order. Valid options are `gprs, umts, lte, auto`. If no valid_service order is defined, then the configured Service Type is used. Example: `mobile.main.service_order="gprs umts lte auto"` Blank: Use configured service type. Range: gprs umts lte auto.

Table 30: Information table for mobile manager basic settings

13.2.2 Mobile manager: CDMA settings

This configuration page is only supported for the Telit CE910-SL CDMA module.

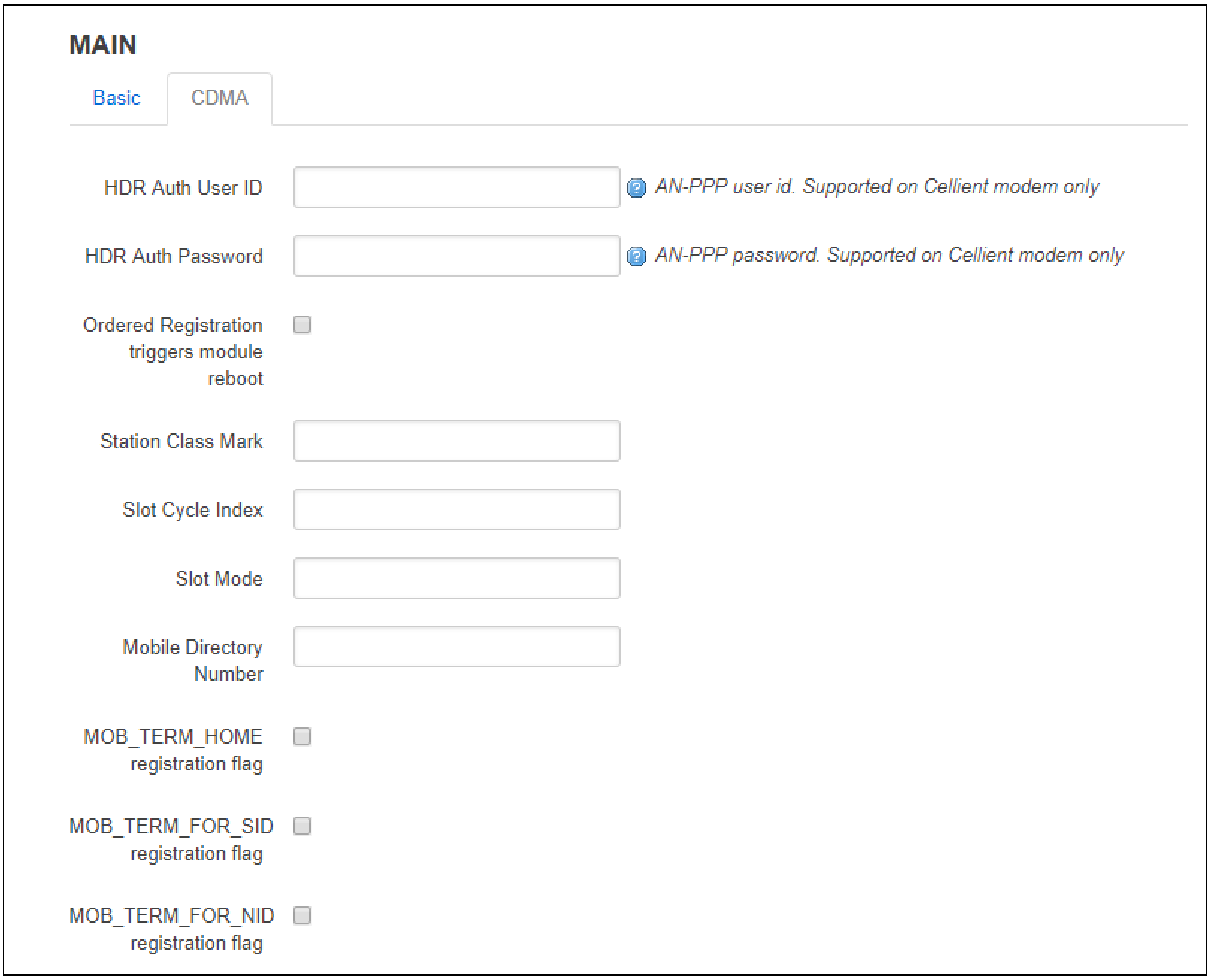


Figure 68: The mobile manager CDMA page

Web Field / UCI / Package Option	Description
Web: HDR Auth User ID UCI: mobile.main.hdr_userid Opt: hdr_userid	AN-PPP user ID. Supported on Cellient (CDMA) modem only. Blank | Range | Depends on the CDMA provider.
Web: HDR Auth User Password UCI: mobile.main.hdr_password Opt: hdr_password	AN-PPP password. Supported on Cellient (CDMA) modem only. Blank | Range | Depends on the CDMA provider.
Web: Ordered Registration triggers module reboot UCI: mobile.main. mobile.main.cdma_ordered_registration_reboot_enabled Opt: cdma_ordered_registration_reboot_enabled	Enables or disables rebooting the module after Order Registration command is received from a network. 0 | Disabled. 1 | Enabled.
Web: Station Class Mark UCI: mobile.main.cdma_station_class_mark Opt: cdma_station_class_mark	Allows the station class mark for the MS to be changed. 58 | 0-255 |
Web: Slot Cycle Index UCI: mobile.main.cdma_slot_cycle_index Opt: cdma_slot_cycle_index	The desired slot cycle index if different from the default. 2 | 0-7 |
Web: Slot Mode UCI: mobile.main.cdma_slot_mode Opt: cdma_slot_mode	Specifies the slot mode. 0 |
Web: Mobile Directory Number UCI: mobile.main.cdma_mobile_directory_number Opt: cdma_mobile_directory_number	Allows the mobile directory number (MDN) to be changed Default | Programmed in module. Digits | Up to 15 digits.
Web: MOB_TERM_HOME registration flag UCI: mobile.main. cdma_mob_term_home_registration_flag Opt: cdma_mob_term_home_registration_flag	The MOB_TERM_HOME registration flag 0 | Disabled. 1 | Enabled.
Web: MOB_TERM_FOR_SID registration flag UCI: mobile.main. cdma_mob_term_for_sid_registration_flag Opt: cdma_mob_term_for_sid_registration_flag	The MOB_TERM_FOR_SID registration flag 0 | Disabled. 1 | Enabled.
Web: MOB_TERM_FOR_NID registration flag UCI: mobile.main. cdma_mob_term_for_nid_registration_flag Opt: cdma_mob_term_for_nid_registration_flag	The MOB_TERM_FOR_NID registration flag 0 | Disabled. 1 | Enabled.
Web: Access Overload Control UCI: mobile.main.cdma_access_overload_control Opt: cdma_access_overload_control	Allows the access overload class to be changed Default | Programmed into module as part of IMSI Range | 0-7
Web: Preferred Serving System UCI: mobile.main.cdma_preferred_serving_system Opt: cdma_preferred_serving_system	The CDMA Preferred Serving System(A/B) 5 |
Web: Digital Analog Mode Preference UCI: cdma_digital_analog_mode_preference Opt: cdma_digital_analog_mode_preference	Digital/Analog Mode Preference. 4 |

Web: Primary Channel A UCI: mobile.main.cdma_primary_channel_a Opt: cdma_primary_channel_a.	Allows the primary channel (A) to be changed. 283 1-2016 Any band class 5 channel number.
Web: Primary Channel B UCI: mobile.main.cdma_primary_channel_b Opt: cdma_primary_channel_b	Allows the primary channel (B) to be changed. 384 1-2016 Any band class 5 channel number
Web: Secondary Channel A UCI: mobile.main.cdma_secondary_channel_a Opt: cdma_secondary_channel_a	Allows the secondary channel (A) to be changed. 691 1-2016 Any band class 5 channel number.
Web: Secondary Channel B UCI: mobile.main.cdma_secondary_channel_b Opt: cdma_secondary_channel_b	Allows the secondary channel (B) to be changed. 777 1-2016 Any band class 5 channel number.
Web: Preferred Forward & Reverse RC UCI: mobile.main.cdma_preferred_forward_and_reverse_rc Opt:cdma_preferred_forward_and_reverse_rc	The Preferred Forward & Reverse RC value, this takes the form "forward_rc,reverse_rc" Format: forward radio channel, reverse radio channel Default: 0,0
Web: SID-NID pairs UCI: mobile.main.cdma_sid_nid_pairs Opt:cdma_sid_nid_pairs	Allows specification of SID:NID pairs, this takes the form "SID1,NID1,SID2,NID2, … Format: SID1 (0-65535),NID (0-65535) Default: 0,65535

Table 31: Information table for mobile manager CDMA settings

13.2.3 Mobile manager: callers

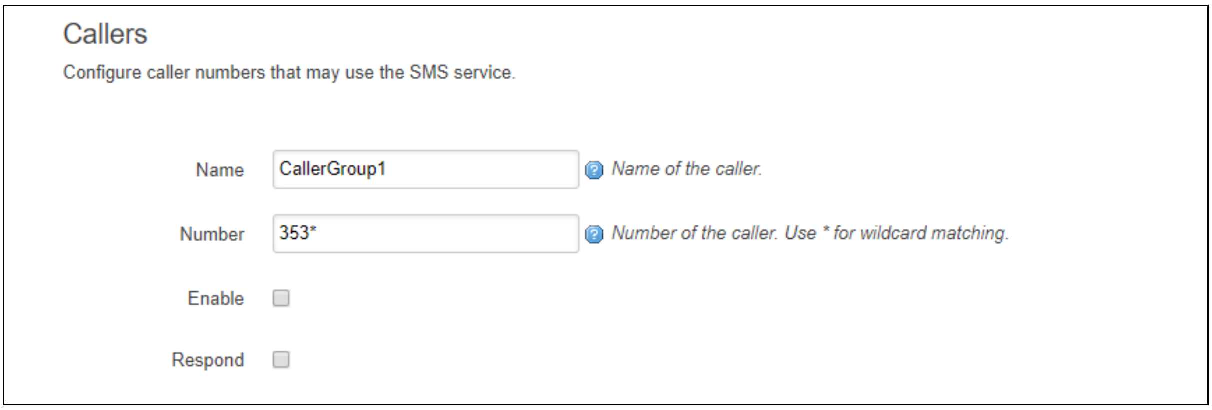


Figure 69: The mobile manager CDMA page

Web Field/UCI/Package Option	Description
Web: Name UCI: mobile.@caller[0].name Opt:name	Name assigned to the caller. Blank: Range: No limit
Web: Number UCI: mobile.@caller[0].number Opt:number	Number of the caller allowed to SMS the router. Add in specific caller numbers, or use the * wildcard symbol. Blank: Range: No limit Characters: Global value (*) is accepted; International value (+) is accepted
Web: Enable UCI: mobile.@caller[0].enabled Opt:enabled	Enables or disables incoming caller ID. 0: Disabled. 1: Enabled.
Web: Respond UCI: mobile.@caller[0].respond Opt: respond	If checked, the router will return an SMS. Select Respond if you want the router to reply. 0: Disabled. 1: Enabled.

Table 32: Information table for mobile manager callers settings

13.2.4 Mobile manager: roaming interface template

For more information on Roaming Interface Template configuration, read the chapter, 'Automatic Operator Selection'.

13.3 Configuring mobile manager using command line

13.3.1 Mobile manager using UCI

The configuration files for mobile manager are stored on **/etc/config/mobile**

The following example shows how to enable the SMS functionality to receive and respond from certain caller ID numbers.

```
root@VA_router:~# uci show mobile
uci set mobile.main=mobile
uci set mobile.main.sim1pin=0000
uci set mobile.main.sim2pin=0000
uci set mobile.main.roaming_sim=none
uci set mobile.main.sms=1
uci set mobile.main.hdr_password=5678
uci set mobile.main.hdr_userid=1234
uci set mobile.main.init_get_iccids=1
uci set mobile.@caller[0]=caller
uci set mobile.@caller[0].name=user1
uci set mobile.@caller[0].number=3538712345678
uci set mobile.@caller[0].enabled=1
```

```
uci set mobile.@caller[0].respond=1
uci set mobile.@caller[1]=caller
uci set mobile.@caller[1].name=user2
uci set mobile.@caller[1].number=3538723456789
uci set mobile.@caller[1].enabled=1
uci set mobile.@caller[1].respond=1
```

13.3.2 Mobile manager using package options

```
root@VA_router:~# uci export mobile
package mobile
config mobile 'main'
        option sim1pin '0000'
        option sim2pin '0000'
        option roaming_sim 'none'
        option sms '1'
        option hdr_password '5678'
        option hdr_userid '1234'
        option init_get_iccids '1'
config caller
        option name 'vasupport'
        option number '353871234567'
        option enabled '1'
        option respond '1'

config caller
        option name 'vasupport1'
        option number '353872345678'
        option enabled '1'
        option respond '1'
```

13.4 Monitoring SMS

You can monitor inbound SMS messages using the router's web browser or via an SSH session.

To monitor SMS using the web browser, login and select **Status >system log**.

Scroll to the bottom of the log to view the SMS message.

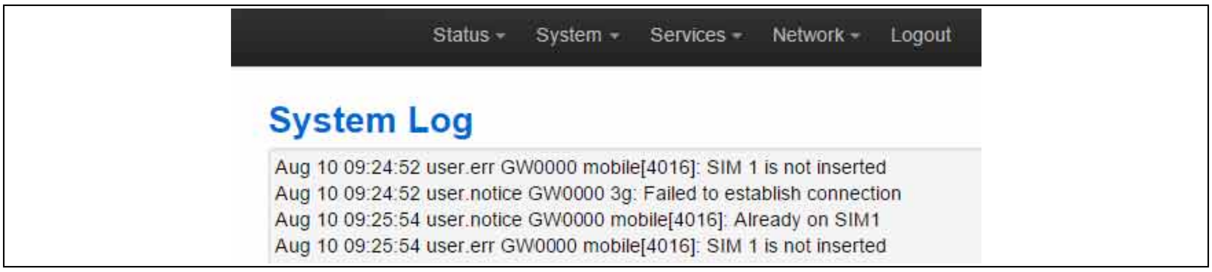


Figure 70: Example of output from system log

To monitor using SSH, login and enter:

```
logread –f &
```

An outgoing SMS message appears.

```
sendsms 353879876543 ‘hello’
root@VirtualAccess:~# Aug 10 16:29:11 user.notice VirtualAccess
mobile[1737]: Queue sms to 353879876543 "hello"
```

13.5 Sending SMS from the router

You can send an outgoing message via the command line using the following syntax:

```
sendsms 353879876543 ‘hello’
root@VirtualAccess:~# Aug 10 16:29:1 user.notice VirtualAccess
mobile[1737]: Queue sms to 353879876543 "hello"
```

13.6 Sending SMS to the router

The router can accept UCI show and set commands via SMS if the caller is enabled.

Note: commands are case sensitive.

An example would be to SMS the SIM card number by typing the following command on the phone and checking the SMS received from the router.

```
uci show mobile.@caller[0].number
```

14 Configuring a GRE interface

General Routing Encapsulation (GRE) is a tunnelling protocol used for encapsulation of other communication protocols inside point to point links over IP.

14.1 Configuration packages used

Package	Sections
network	interface

14.2 Creating a GRE connection using the web interface

To create GRE interfaces through the web interface, in the top menu, select **Network ->Interfaces**.

There are three sections in the Interfaces page.

Section	Description
Interface Overview	Shows existing interfaces and their status. You can create new, and edit existing interfaces here.
Port Map	In this section you can map device ports to Ethernet interfaces. Ports are marked with capital letters starting with 'A'. Type in space separated port numbers in the port map fields.
ATM Bridges	ATM bridges expose encapsulated Ethernet in AAL5 connections as virtual Linux network interfaces, which can be used in conjunction with DHCP or PPP to dial into the provider network.

In the Interface Overview section, click **Add new interface**. The Create Interface page appears.

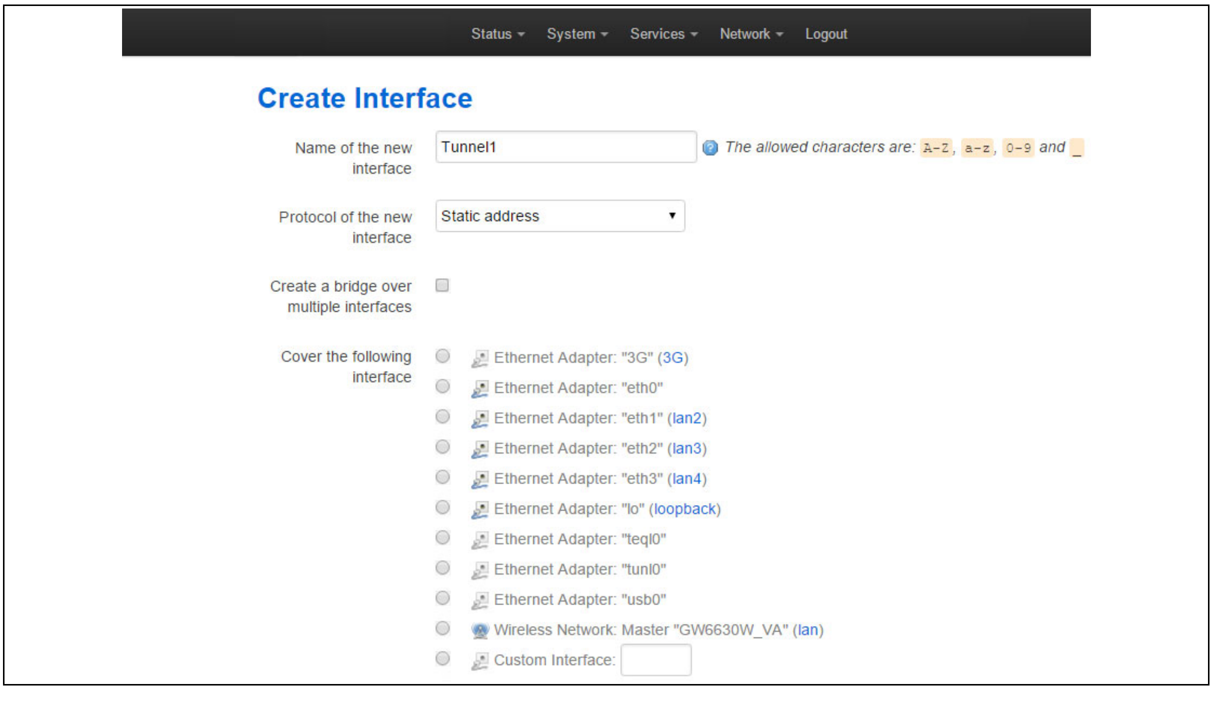


Figure 71: The create interface page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface
UCI: network. .<if name>
Opt: config interface</td><td>Assigns a logical name to the GRE tunnel, The network interface section will be assigned this name <if name>.
Type the name of the new interface.
Allowed characters are A-Z, a-z, 0-9 and _.
Must be less than 11 characters.</td></tr>
<tr><td>Web: Protocol of the new interface
UCI: network.<if name>.proto
Opt: proto</td><td>Specifies what protocol the interface will operate on. Select GRE.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td></tr>
<tr><td>PPP</td><td>Point-to-Point protocol</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: Create a bridge over multiple interfaces
UCI: network.<if name>
Opt: n/a</td><td>Not applicable for GRE.</td></tr>
<tr><td>Web: Cover the following interface
UCI: network.<if name>
Opt:n/a</td><td>Not applicable for GRE.</td></tr>
</table>

Table 33: Information table for the create new interface page

Click **Submit**. The Common Configuration page appears. There are three sections in the Common Configurations page.

Section	Description
General Setup	Configure the basic interface settings such as protocol, IP address, mask length, local interface, remote IP address, TTL, tunnel key and MTU.
Advanced Settings	'Bring up on boot' and 'monitor interface state' settings.
Firewall settings	Assign a firewall zone to the connection.

14.2.1 GRE connection: common configuration - general setup

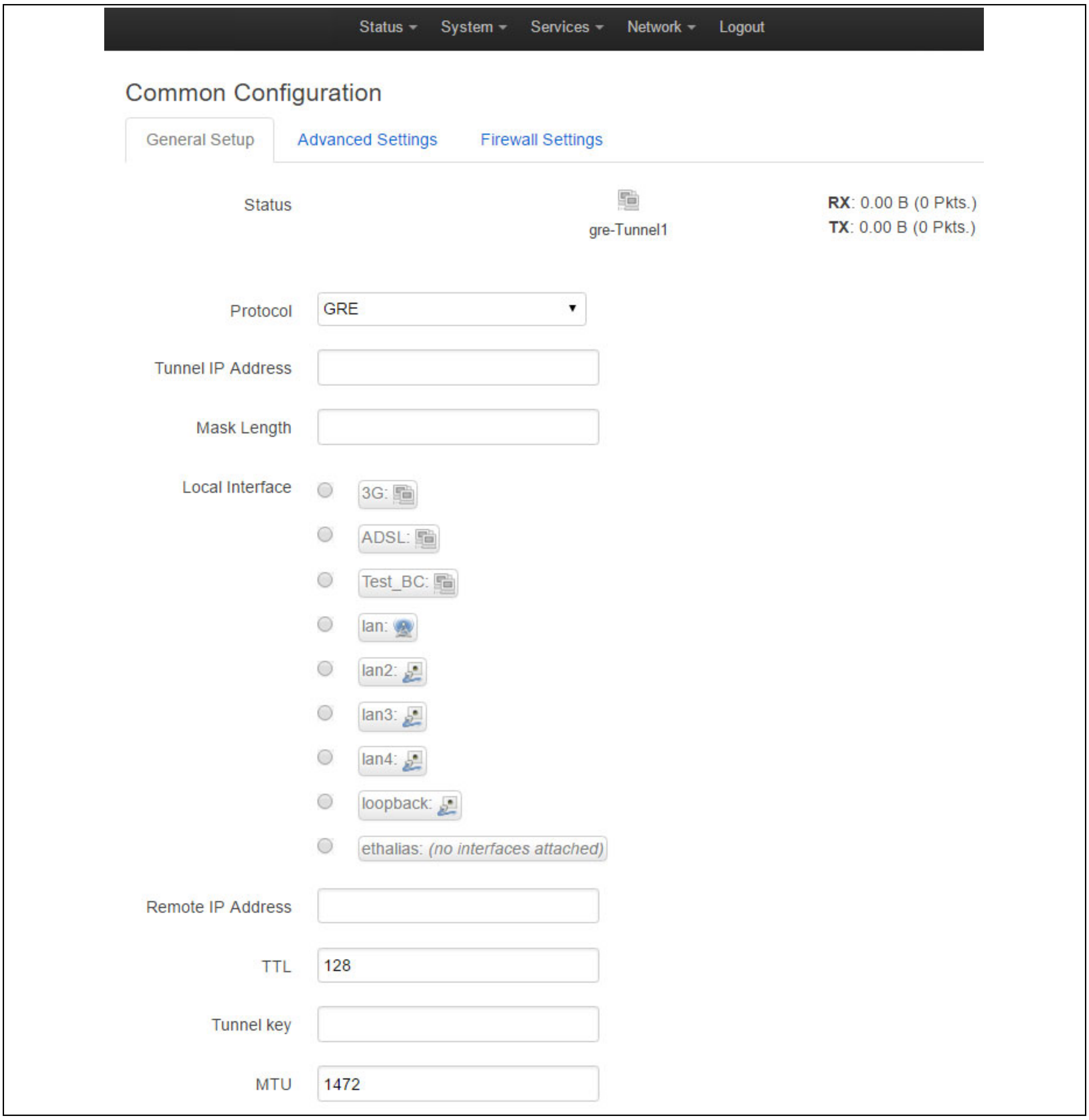


Figure 72: The GRE common configuration page

Web Field / UCI / Package Option	Description
Web: Protocol of the new interface UCI: network.<if name>.proto Opt: proto	Shows the protocol the interface will operate on. GRE should be currently selected.
Web: Tunnel IP Address UCI: network.<if name>.ipaddr Opt: ipaddr	Configures local IP address of the GRE interface.
Web: Mask Length UCI: network.<if name>.mask_length Opt: mask_length	Subnet mask, in CIDR notation, to be applied to the tunnel. Typically '30' for point-to-point tunnels. 24 Range: 0 - 30

Web: Local Interface UCI: network.<if name>.local_interface Opt: local_interface	Specifies which interface is going to be linked with the GRE tunnel interface (optional).
Web: Remote IP address UCI: network.<if name>.remote_ip Opt: remote_ip	For point to point tunnels specifies Remote IP address.
Web: TTL UCI: network.<if name>.ttl Opt: ttl	Sets Time-To-Live value on the interface. 128 Range
Web: Tunnel key UCI: network.<if name>.key Opt: key	Sets GRE tunnel ID key (optional). Usually an integer.
Web: MTU UCI: network.<if name>.mtu Opt: mtu	Configures MTU (maximum transmission unit) size of PDUs using this interface. 1472 Range

Table 34: Information table for GRE

14.2.2 GRE connection: common configuration-advanced settings

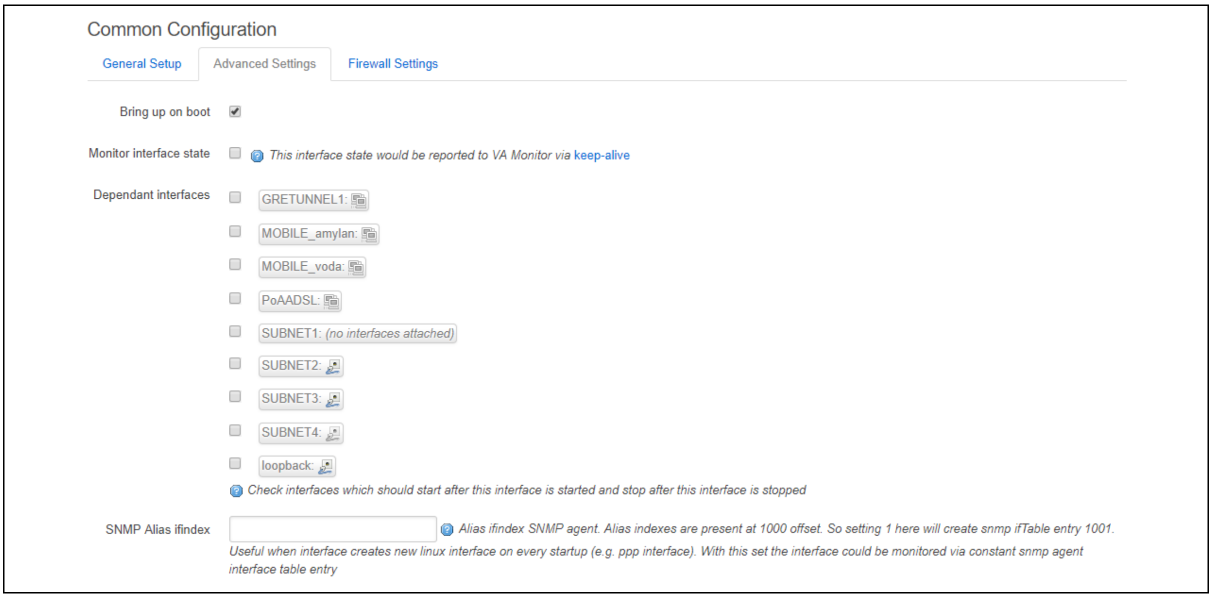


Figure 73: GRE advanced settings page

Web Field / UCI / Package Option	Description
Web: Bring up on boot UCI: network.<if name>.auto Opt: auto	Enables the interface to connect automatically on boot up. 0 — Disabled. 1 — Enabled.
Web: Monitor interface state UCI: network.<if name>.monitored Opt: monitored	Enabled if status of interface is presented on Monitoring platform. 0 — Disabled. 1 — Enabled.
Web: Dependant Interfaces UCI: network.[..x..].dependants Opt: dependants	Lists interfaces that are dependent on this parent interface. Dependant interfaces will go down when parent interface is down and will start or restart when parent interface starts. Separate multiple interfaces by a space when using UCI. Example: `option dependants 'PPPADSL MOBILE'` This replaces the following previous options in child interfaces. gre — option local_interface lt2p — option src_ipaddr iot — option wan1 wan2 6in4 — option ipaddr 6to4 — option ipaddr
Web: SNMP Alias ifindex UCI: network.[..x..].snmp_alias_ifindex Opt: snmp_alias_ifindex	Defines a static SNMP interface alias index for this interface, that can be polled via the SNMP interface index (*snmp_alias_ifindex+1000*). See *Configuring SNMP* section for more information Blank — No SNMP interface alias index Range — 0 - 4294966295

Table 35: Information table for GRE advanced settings

14.2.3 GRE connection: firewall settings

Use this section to select the firewall zone you want to assign to this interface.

Select **unspecified** to remove the interface from the associated zone or fill out the create field to define a new zone and attach the interface to it.

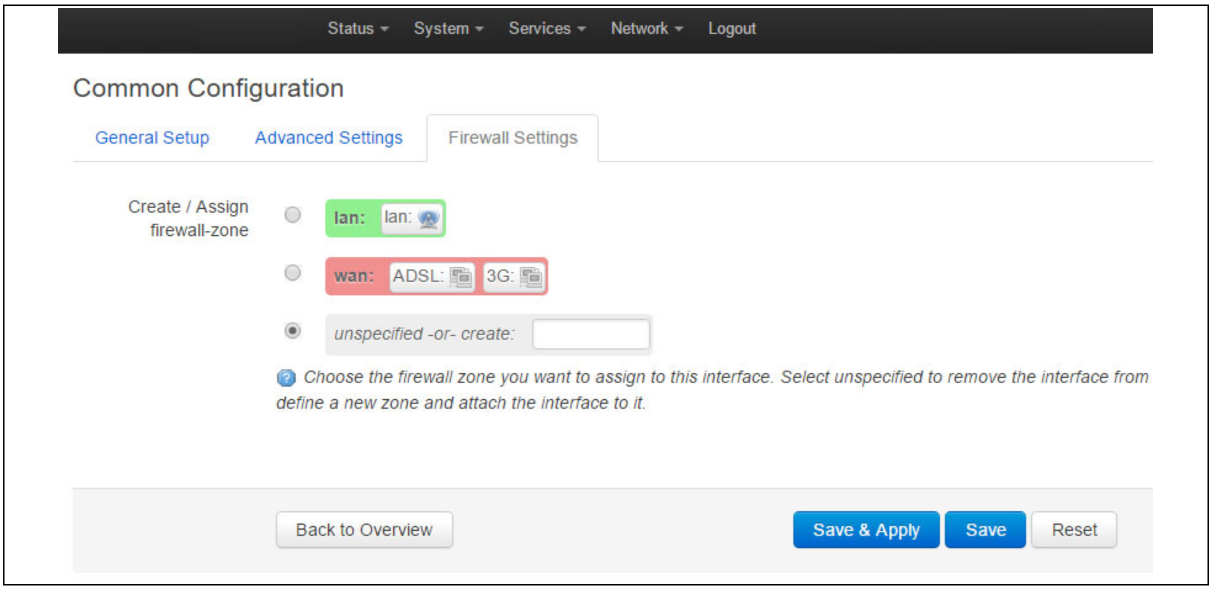


Figure 74: GRE firewall settings

Click **Save and Apply**. This will save the current settings and return you to the Interface Overview page. To configure further settings on the GRE interface select **EDIT** for the relevant GRE interface.

14.2.4 GRE connection: adding a static route

After you have configured the GRE interface, you must configure a static route to route the desired traffic over the GRE tunnel. To do this, browse to **Network->Static Routes**. For more information, read the chapter 'Configuring Static Routes'.

14.3 GRE configuration using command line

The configuration file is stored on **/etc/config/network**

For the examples below tunnel1 is used as the interface logical name.

14.4 GRE configuration using UCI

```
root@VA_router:~# uci show network
network.tunnel1=interface
network.tunnel1.proto=gre
network.tunnel1.monitored=0
network.tunnel1.ipaddr=172.255.255.2
network.tunnel1.mask_length=24
network.tunnel1.local_interface=wan
network.tunnel1.remote_ip=172.255.255.100
network.tunnel1.ttl=128
network.tunnel1.key=1234
network.tunnel1.mtu=1472
network.tunnel1.auto=1
```

14.5 GRE configuration using package options

```
root@VA_router:~# uci export network
config interface 'tunnel1'
      option proto 'gre'
      option monitored ‘0’
      option ipaddr '172.255.255.2'
      option mask_length '24'
      option local_interface 'wan'
      option remote_ip ‘172.255.255.100’
      option ttl '128'
```

```
      option key '1234'
      option mtu '1472'
      option auto ‘1’
```

To change any of the above values use `uci set` command.

14.6 GRE diagnostics

14.6.1 GRE interface status

To show the current running interfaces, enter:

```
root@VA_router:~# ifconfig
base0        Link encap:Ethernet  HWaddr 00:00:00:00:01:01
             inet6 addr: fe80::200:ff:fe00:101/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1504  Metric:1
             RX packets:39810 errors:0 dropped:0 overruns:0 frame:0
             TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:10889090 (10.3 MiB)  TX bytes:68820 (67.2 KiB)
eth4         Link encap:Ethernet  HWaddr 00:1E:10:1F:00:00
             inet addr:10.68.66.54  Bcast:10.68.66.55  Mask:255.255.255.252
             inet6 addr: fe80::21e:10ff:fe1f:0/64 Scope:Link
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:81 errors:0 dropped:0 overruns:0 frame:0
             TX packets:127 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:8308 (8.1 KiB)  TX bytes:12693 (12.3 KiB)
gre-Tunnel1 Link encap:UNSPEC  HWaddr 0A-44-42-36-DB-B0-00-48-00-00-00-00-
00-00-00-00
             inet addr:13.13.13.2  Mask:255.255.255.248
             inet6 addr: fe80::5efe:a44:4236/64 Scope:Link
             UP RUNNING MULTICAST  MTU:1472  Metric:1
             RX packets:7 errors:0 dropped:0 overruns:0 frame:0
             TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:912 (912.0 B)  TX bytes:884 (884.0 B)
lo           Link encap:Local Loopback
             inet addr:127.0.0.1  Mask:255.0.0.0
             inet6 addr: ::1/128 Scope:Host
```

```
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:1465 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1465 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:166202 (162.3 KiB)  TX bytes:166202 (162.3 KiB)
```

To display a specific GRE interface, enter `ifconfig gre-<if name>`:

```
root@VA_router:~# ifconfig gre-Tunnel1
gre-Tunnel1   Link encap:UNSPEC  HWaddr 0A-44-42-36-00-00-7F-E2-00-00-00-
00-00-00-00-00
              inet addr:13.13.13.2  Mask:255.255.255.248
              inet6 addr: fe80::5efe:a44:4236/64 Scope:Link
              UP RUNNING MULTICAST  MTU:1472  Metric:1
              RX packets:7 errors:0 dropped:0 overruns:0 frame:0
              TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:0
              RX bytes:912 (912.0 B)  TX bytes:8GRE route status
```

To show the current GRE route status, enter:

```
root@VA_router:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
0.0.0.0         10.68.66.53     0.0.0.0         UG    0      0      0 eth4
0.0.0.0         13.13.13.1      0.0.0.0         UG    1      0      0 gre-
Tunnel1
10.68.66.52     0.0.0.0         255.255.255.252 U     0      0      0 eth4
13.13.13.0      0.0.0.0         255.255.255.248 U     0      0      0 gre-
Tunnel1
172.19.101.3    13.13.13.1      255.255.255.255 UGH   0      0     0 gre-
Tunnel1
```

Note: a GRE route will only be displayed in the routing table when the interface is up.

15 Configuring static routes

It is possible to define arbitrary IPv4 routes on specific interfaces using route sections. As for aliases, multiple sections can be attached to an interface. These types of routes are most commonly known as static routes.

You can add static routes to the routing table to forward traffic to specific subnets when dynamic routing protocols are not used or they are not configured for such subnets. They can be created based on outgoing interface or next hop IP address.

15.1 Configuration package used

Package	Sections
network	route

15.2 Configuring static routes using the web interface

In the top menu, select **Network -> Static Routes**. The Routes page appears.

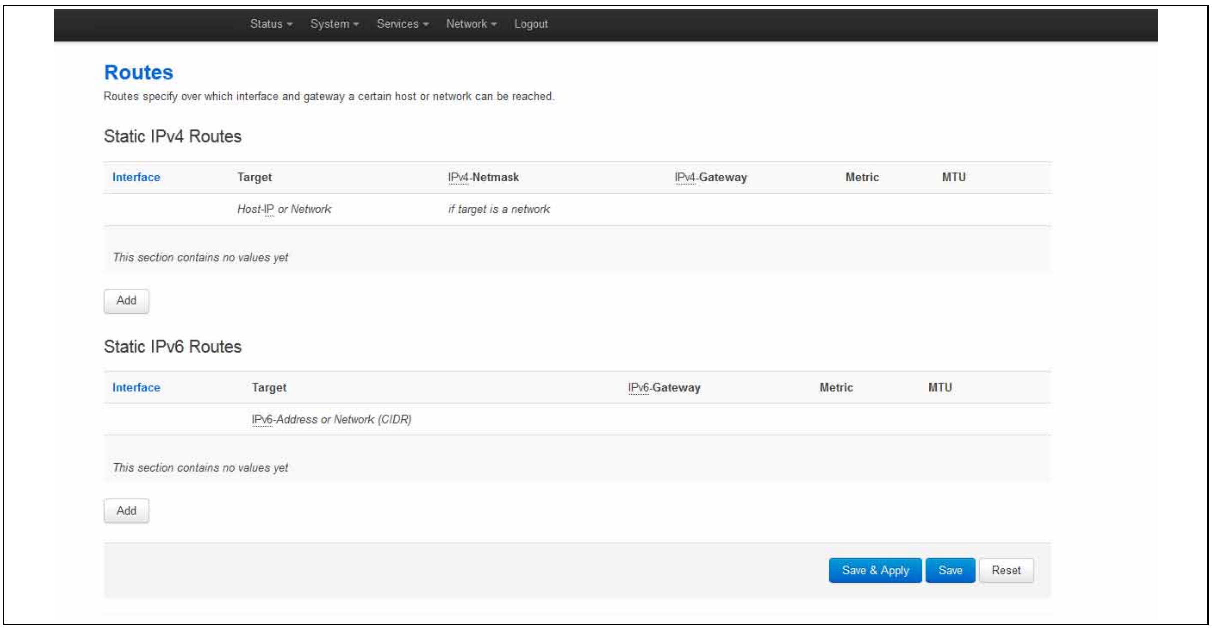


Figure 75: The routes page

In the IPv4 Routes section, click **Add**.

Web Field/UCI/Package Option	Description
Web: Interface UCI: network.@route[0].interface Opt: Interface	Specifies the logical interface name of the parent or master interface this route belongs to. It must refer to one of the defined interface sections.
Web: target UCI: network.@route[0].target Opt: target	Specifies the route network IP address.
Web: netmask UCI: network.@route[0].netmask Opt: netmask	Defines the route netmask. If omitted, 255.255.255.255 is assumed, which makes the target a host address.

Web: Gateway UCI: network.@route[0].gateway Opt: Gateway	Network gateway. If omitted, the gateway from the parent interface is taken. If set to 0.0.0.0 no gateway will be specified for the route.
Web: Metric UCI: network.@route[0].metric Opt: metric	Specifies the route metric to use. 0 Range
Web: MTU UCI: network.@route[0].mtu Opt:mtu	Defines a specific MTU for this route. If omitted, the MTU from the parent interface will be taken. Blank Range

Table 36: Information table for IPv4 static routes section

15.3 Configuring IPv6 routes using the web interface

You can also specify IPv6 routes by defining one or more IPv6 routes. In the IPv6 routes section, click **Add**.

Web Field / UCI / Package Option	Description
Web: Interface UCI: network.@route[1].interface Opt: interface	Specifies the logical interface name of the parent or master interface this route belongs to. It must refer to one of the defined interface sections.
Web: target UCI: network.@route[1].target Opt: target	Specifies the route network IP address, or subnet in CIDR notation: Eample: 2001:0DB8:100:F00:BA3::1/64
Web: Gateway UCI: network.@route[1].gateway Opt: Gateway	Network gateway. If omitted, the gateway from the parent interface is taken. If set to 0.0.0.0 no gateway will be specified for the route.
Web: Metric UCI: network.@route[1].metric Opt: metric	Specifies the route metric to use. 0 Range
Web: MTU UCI: network.@route[1].mtu Opt:mtu	Defines a specific MTU for this route. If omitted the MTU from the parent interface will be taken. Empty Range

Table 37: Information table for IPv6 routes

When you have made your changes, click **Save & Apply**.

15.4 Configuring routes using command line

By default all routes are named ‘route’, it is identified by @route then the route’s position in the package as a number. For example, for the first route in the package using UCI:

```
network.@route[0]=route
network.@route[0].interface=lan
```

Or using package options:

```
config route
        option 'interface' 'lan'
```

However, you can give a route a name if desired. For example, a route named ‘myroute’ will be `network.myroute`.

To define a named route using UCI, enter:

```
network.name_your_route=route
network.name_your_route.interface=lan
```

To define a named route using package options, enter:

```
config route 'name_your_route'
        option 'interface' 'lan'
```

15.5 IPv4 routes using UCI

The command line example routes in the subsections below do not have a configured name.

```
root@VA_router:~# uci show network
network.@route[0]=route
network.@route[0].interface=lan
network.@route[0].target=3.3.3.10
network.@route[0].netmask=255.255.255.255
network.@route[0].gateway=10.1.1.2
network.@route[0].metric=3
network.@route[0].mtu=1400
```

15.6 IPv4 routes using package options

```
root@VA_router:~# uci export network
package network
      ….
config route
        option interface 'lan'
        option target '2.2.2.2'
        option netmask '255.255.255.255'
        option gateway '192.168.100.1'
        option metric '1'
        option mtu '1500'
```

15.7 IPv6 routes using UCI

```
root@VA_router:~# uci show network
network.@route[1]=route
network.@route[1].interface=lan
network.@route[1].target=2001:0DB8:100:F00:BA3::1/64
network.@route[1].gateway=2001:0DB8:99::1
network.@route[1].metric=1
network.@route[1].mtu=1500
```

15.8 IPv6 routes using packages options

```
root@VA_router:~# uci export network
package network
      ….
config route
        option interface 'lan'
        option target '2001:0DB8:100:F00:BA3::1/64'
        option gateway '2001:0DB8:99::1'
      option metric ‘1’
        option mtu '1500'
```

15.9 Static routes diagnostics

15.9.1 Route status

To show the current routing status, enter:

```
root@VA_router:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.100.0   *               255.255.255.0   U     0      0        0 eth0
```

Note: a route will only be displayed in the routing table when the interface is up.

16 Configuring BGP (Border Gateway Protocol)

BGP is a protocol for exchanging routing information between gateway hosts, each with its own router, in a network of autonomous systems. BGP is often the protocol used between gateway hosts on the internet. The routing table contains a list of known routers, the addresses they can reach, and a cost metric associated with the path to each router so that the best available route is chosen.

16.1 Configuration package used

Package	Sections
bgpd	routing
	peer
	routemap

16.2 Configuring BGP using the web interface

In the top menu, select **Network -> BGP**. BGP configuration page appears. The page has three sections: Global Settings, BGP Neighbours and BGP Route Map.

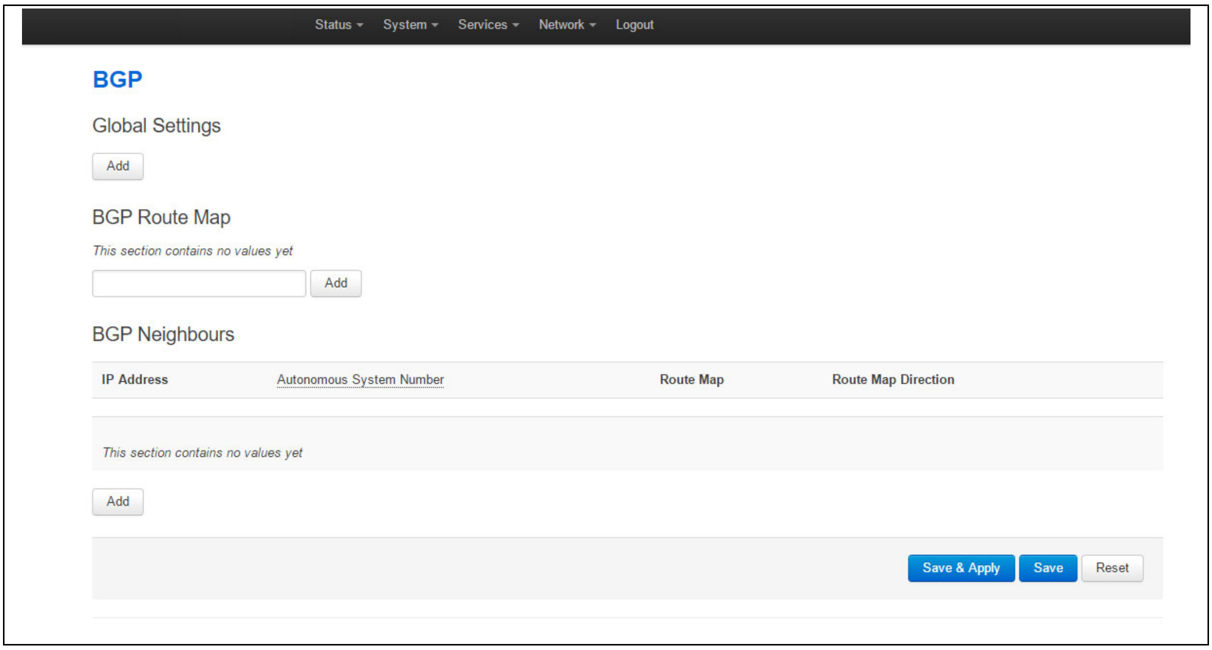


Figure 76: The BGP page

16.2.1 BGP global settings

To configure global BGP settings, click **Add.** The Global Settings page appears.

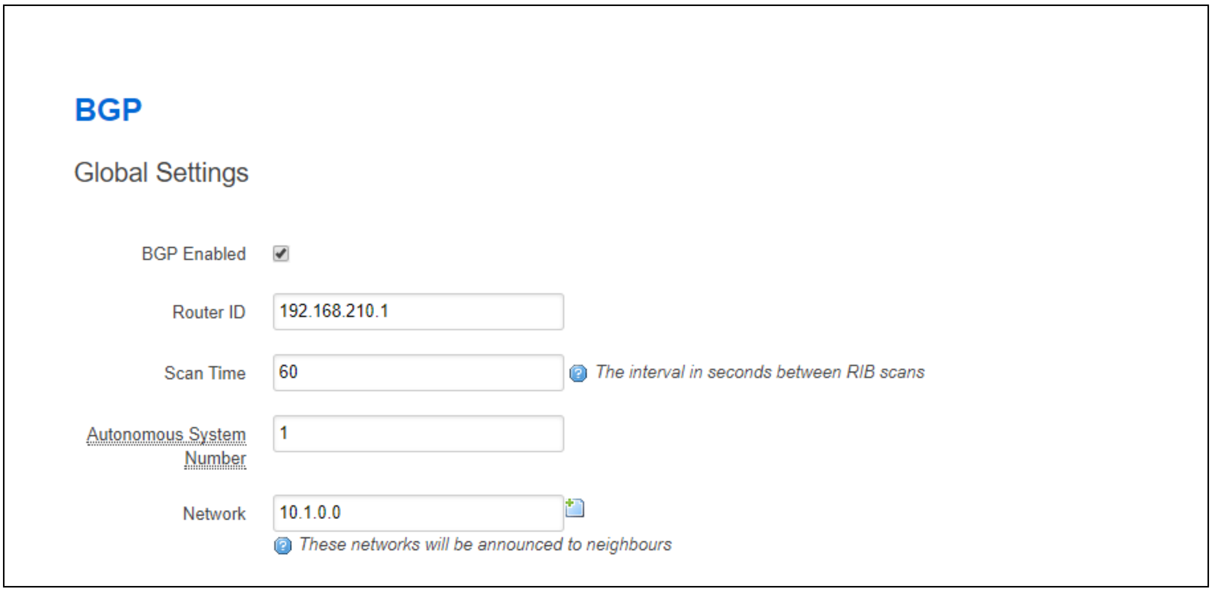


Figure 77: The BGP global settings page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: BGP Enabled
UCI: bgpd.bgpd.enabled
Opt: enabled</td><td>Enables or disables BGP protocol.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
<tr><td>Web: Router ID
UCI: bgpd.bgpd.router_id
Opt: router_id</td><td>Sets a Unique Router ID in 4 byte format 0.0.0.0.</td></tr>
<tr><td>Web: Scan Time
UCI: bgpd.bgpd.scan_time
Opt: scan_time</td><td>Defines the interval in seconds between RIB scans.<table><tr><td>60</td><td>60 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Autonomous System Number
UCI: bgpd.bgpd.asn
Opt: asn</td><td>Defines the ASN for the local router. Type in the ASN.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td>1-4294967295</td></tr></table></td></tr>
<tr><td>Web: Network
UCI: bgpd.bgpd.network
Opt: list network</td><td>Sets the list of networks that will be advertised to neighbours in prefix format 0.0.0.0/0. Separate multiple networks by a space using UCI. Ensure the network prefix matches the one shown in the routing table. See ‘Routes’ section below.</td></tr>
</table>

Table 38: Information table for BGP global settings

16.2.2 Optionally configure a BGP route map

Route maps provide a means to both filter and/or apply actions to a route. This allows a policy to be applied to routes. Route maps are an ordered list of route map entries each with a set of criteria that must be matched before specific attributes of the route are modified.

Scroll down to the BGP Route Map section.

Type in a name for the BGP route map name and then click **Add**. The ROUTEMAP configuration section appears. You can configure multiple route maps.

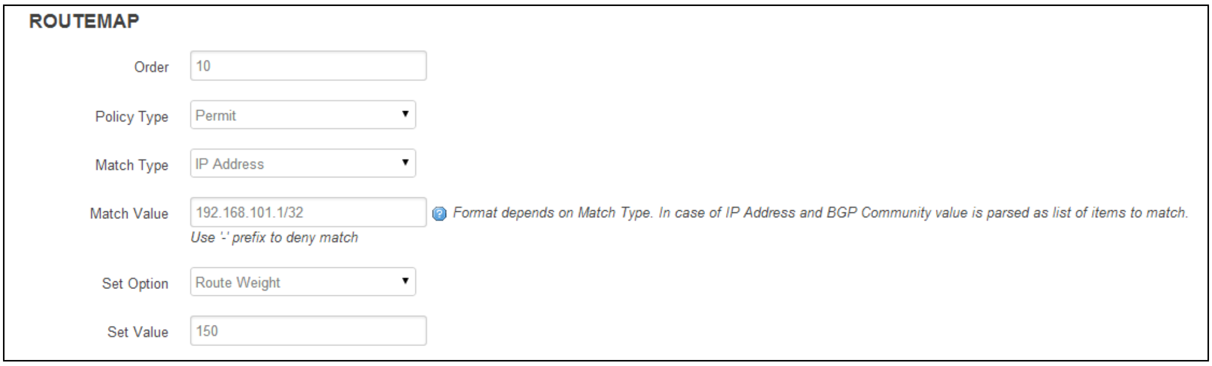


Figure 78: The routemap section

Web Field / UCI / Package Option	Description
Web: Order UCI: bgpd.ROUTEMAP.order Opt: order	Defines the Route Map order number. Blank: Range: 1-65535
Web: Policy Type UCI: bgpd.ROUTEMAP.permit Opt: permit	Defines the actions taken if the entry is matched. Deny: Denies the route. Permit: Permits the route so process the set actions for this entry.
Web: Match Type UCI: bgpd.ROUTEMAP.match_type Opt: match_type	Defines match type. Available options are as follows: IP address: Matches IP address. IP Next Hop: Matches next hop IP address. AS-Path: Matches AS-path. Route Metric: Matches route metric. BGP Community: Matches BGP community.
Web: Match value UCI: bgpd.ROUTEMAP.match Opt: match	Defines the value of the match type. Format depends on the Match Type selected. In the case of IP address and BGP Community values, the match value is parsed as a list of items to match.
Web: Set Option UCI: bgpd.ROUTEMAP.set_type Opt: set_type	Defines the set option to be processed on a match. Available options are shown below. None: IP Next Hop: Setting option for IP next hop. Local Preference: Setting option for Local Preference. Route Weight: Setting option for Route Weight. BGP MED: Setting option for BGP multi-exit discriminator (BGP metric). AS Path to Prepend: Setting option to prepend AS to AS path. BGP Community: Setting option for BGP community. IPv6 Next Hop Global: Setting option for IPv6 Next Hop Global. IPv6 Next Hop Local: Setting option for IPv6 Next Hop Local.
Web: Value UCI: bgpd.ROUTEMAP.set Opt: set	Defines the set value when a match occurs. Value format depends on the set option you have selected.

Table 39: Information table for routemap

16.2.3 Configure BGP neighbours

To configure BGP neighbours, in the BGP neighbours section, click **Add**. The BGP Neighbours page appears. Multiple BGP neighbours can be configured.

Figure 79: The BGP neighbours section

Web Field / UCI / Package Option	Description
Web: IP Address UCI: bgpd.@peer[0].ipaddr Opt: ipaddr	Sets the IP address of the neighbour.
Web: Autonomous System Number UCI: bgpd.@peer[0].asn Opt: asn	Sets the ASN of the remote peer. Blank: Range: 1-4294967295
Web: Route Map UCI: bgpd.@peer[0].route_map Opt: route_map	Sets route map name to use with this neighbour.
Web: Route Map Direction UCI: bgpd.@peer[0].route_map_in Opt: route_map_in	Defines the direction the route map should be applied. 1: In 0: Out

Table 40: Information table for BGP neighbours

16.3 Configuring BGP using UCI

You can also configure BGP using UCI. The configuration file is stored on **/etc/config/bgpd**

```
root@VA_router:~# uci show bgpd
bgpd.bgpd=routing
bgpd.bgpd.enabled=yes
bgpd.bgpd.router_id=3.3.3.3
bgpd.bgpd.asn=1
bgpd.bgpd.network=11.11.11.0/29 192.168.103.1/32
bgpd.@peer[0]=peer
bgpd.@peer[0].route_map_in=yes
bgpd.@peer[0].ipaddr=11.11.11.1
bgpd.@peer[0].asn=1
bgpd.@peer[0].route_map=ROUTEMAP
bgpd.ROUTEMAP=routemap
```

```
bgpd.ROUTEMAP.order=10
bgpd.ROUTEMAP.permit=yes
bgpd.ROUTEMAP.match_type=ip address
bgpd.ROUTEMAP.match=192.168.101.1/32
bgpd.ROUTEMAP.set_type=ip next-hop
bgpd.ROUTEMAP.set='192.168.101.2/32'
```

To change any of the above values use UCI set command.

16.4 Configuring BGP using packages options

```
root@VA_router:~# uci export bgpd
package bgpd
config routing 'bgpd'
        option enabled 'yes'
        option router_id '3.3.3.3'
        option asn '1'
        list network '11.11.11.0/29'
        list network '192.168.103.1/32'
config peer
        option route_map_in 'yes'
        option ipaddr '11.11.11.1'
        option asn '1'
        option route_map 'ROUTEMAP'

config routemap 'ROUTEMAP'
        option order '10'
        option permit 'yes'
        option match_type 'ip address'
        option match '192.168.101.1/32'
        option set_type 'ip next-hop'
        option set '192.168.101.2/32'
```

16.5 View routes statistics

To view routes statistics, in the top menu click **Status -> Routes**. The routing table appears.

Routes

The following rules are currently active on this system.

ARP

IPv4-Address	MAC-Address	Interface
192.168.210.100	50:b7:c3:0c:1e:4b	br-lan
10.1.1.124	d4:ae:52:cd:61:21	eth1
10.1.10.83	00:13:60:51:39:56	eth1

Active IPv4-Routes

Network	Target	IPv4-Gateway	Metric
wan	0.0.0.0/0	10.64.64.64	0
wan	0.0.0.0/0	10.64.64.64	1
LAN2	10.1.0.0/16	0.0.0.0	0
wan	10.64.64.64	0.0.0.0	0
LAN2	192.168.101.1	10.1.10.83	0
lan	192.168.210.0/24	0.0.0.0	0
wan	217.67.129.143	10.64.64.64	0

Active IPv6-Routes

Network	Target	IPv6-Gateway	Metric
loopback	0:0:0:0:0:0:0:0/0	0:0:0:0:0:0:0:0/0	FFFFFFFF
loopback	0:0:0:0:0:0:0:0/0	0:0:0:0:0:0:0:0/0	FFFFFFFF
loopback	0:0:0:0:0:0:0:1	0:0:0:0:0:0:0:0/0	00000000
LAN2	FF02:0:0:0:0:0:0:FB	0:0:0:0:0:0:0:0/0	00000000
(base0)	FF00:0:0:0:0:0:0:0/8	0:0:0:0:0:0:0:0/0	00000100
lan	FF00:0:0:0:0:0:0:0/8	0:0:0:0:0:0:0:0/0	00000100
LAN2	FF00:0:0:0:0:0:0:0/8	0:0:0:0:0:0:0:0/0	00000100
loopback	0:0:0:0:0:0:0:0/0	0:0:0:0:0:0:0:0/0	FFFFFFFF

Figure 80: The routing table

To view routes via the command line, enter:

```
root@support:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.1.0.0        0.0.0.0         255.255.0.0     U     0      0        0 br-lan2
```

17 Configuring OSPF (Open Shortest Path First)

17.1 Introduction

OSPF is a standardised Link State routing protocol, designed to scale efficiently to support larger networks. Link State protocols track the status and connection type of each link and produce a calculated metric based on these and other factors, including some set by the network administrator. Link State protocols will take a path which has more hops, but that uses a faster medium over a path using a slower medium with fewer hops.

- OSPF adheres to the following Link State characteristics:
- OSPF employs a hierarchical network design using areas.
- OSPF will form neighbour relationships with adjacent routers in the same area.
- Instead of advertising the distance to connected networks, OSPF advertises the status of directly connected links using Link-State Advertisements (LSAs).
- OSPF sends updates (LSAs) when there is a change to one of its links, and will only send the change in the update. LSAs are additionally refreshed every 30 minutes.
- OSPF traffic is multicast either to address 224.0.0.5 (all OSPF routers) or 224.0.0.6 (all designated routers).
- OSPF uses the Dijkstra Shortest Path First algorithm to determine the shortest path.
- OSPF is a classless protocol, and therefore supports variable Length Subnet Masks (VLSMs).

Other characteristics of OSPF include:

- OSPF supports only IP routing.
- OSPF routes have an administrative distance is 110.
- OSPF uses cost as its metric, which is computed based on the bandwidth of the link. OSPF has no hop-count limit.

The OSPF process builds and maintains three separate tables:

- **A neighbour table** containing a list of all neighbouring routers
- **A topology table** containing a list of all possible routes to all known networks within an area
- **A routing table** containing the best route for each known network

17.1.1 OSPF areas

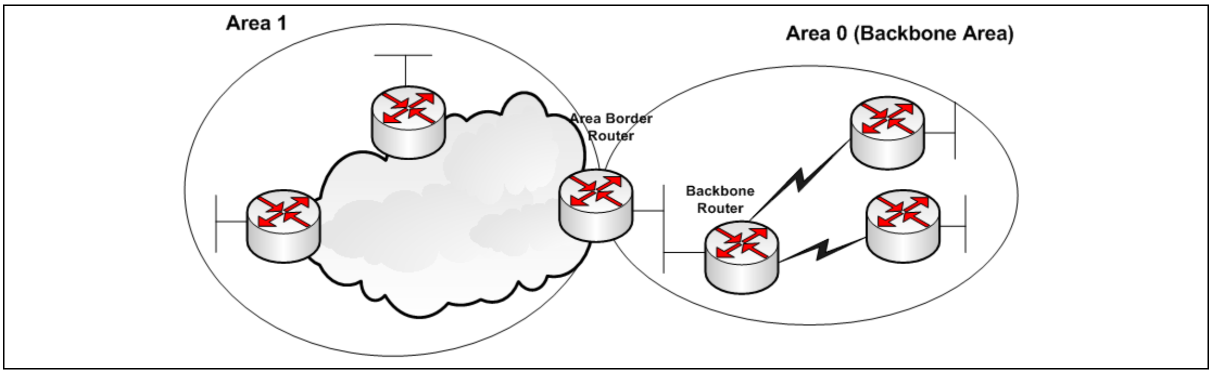


Figure 81: OSPF areas

OSPF has a number of features that allow it to scale well for larger networks. One of these features is OSPF areas. OSPF areas break up the topology so that routers in one area know less topology information about the subnets in the other area, and they do not know anything about the routers in the other area at all. With smaller topology databases, routers consume less memory and take less processing time to run SPF.

The Area Border Router (ABR) is the border between two areas. The ABR does not advertise full topology information about the part of the network in area 0 to routers in area 1. Instead the ABR advertises summary information about the subnets in area 0. Area 1 will just see a number of subnets reachable via area 0.

17.1.2 OSPF neighbours

OSPF forms neighbour relationships, called adjacencies, with other routers in the same Area by exchanging 'Hello' packets to multicast address 224.0.0.5. Only after an adjacency is formed can routers share routing information.

Each OSPF router is identified by a unique router ID. The router ID can be determined in one of three ways:

- The router ID can be manually specified.
- If not manually specified, the highest IP address configured on any Loopback interface on the router will become the router ID.
- If no loopback interface exists, the highest IP address configured on any physical interface will become the router ID.

By default, Hello packets are sent out OSPF-enabled interfaces every 10 seconds for broadcast and point-to-point interfaces, and 30 seconds for non-broadcast and point-to-multipoint interfaces.

OSPF also has a 'Dead Interval', which indicates how long a router will wait without hearing any hellos before announcing a neighbour as 'down'. The default setting for the Dead Interval is 40 seconds for broadcast and point-to-point interfaces, and 120 seconds for non-broadcast and point-to-multipoint interfaces. By default, the Dead Interval timer is four times the Hello interval.

OSPF routers will only become neighbours if the following parameters within a Hello packet are identical on each router:

- Area ID
- Area Type (stub, NSSA, etc.)
- Prefix
- Subnet Mask
- Hello Interval
- Dead Interval
- Network Type (broadcast, point-to-point, etc.)
- Authentication

The Hello packets also serve as keepalives to allow routers to quickly discover if a neighbour is down. Hello packets also contain a neighbour field that lists the router IDs of all neighbours the router is connected to. A neighbour table is constructed from the OSPF Hello packets, which includes the following information:

- The router ID of each neighbouring router
- The current ‘state’ of each neighbouring router
- The interface directly connecting to each neighbour
- The IP address of the remote interface of each neighbour

17.1.3 OSPF designated routers

In multi-access networks such as Ethernet, there is the possibility of many neighbour relationships on the same physical segment. This leads to a considerable amount of unnecessary Link State Advertisement (LSA) traffic. If a link of a router were to fail, it would flood this information to all neighbours. Each neighbour, in turn, would then flood that same information to all other neighbours. This is a waste of bandwidth and processor load.

To prevent this, OSPF will elect a Designated Router (DR) for each multi-access networks, accessed via multicast address 224.0.0.6. For redundancy purposes, a Backup Designated Router (BDR) is also elected.

OSPF routers will form adjacencies with the DR and BDR. If a change occurs to a link, the update is forwarded only to the DR, which then forwards it to all other routers. This greatly reduces the flooding of LSAs. DR and BDR elections are determined by a router’s OSPF priority, which is configured on a per-interface basis (a router can have interfaces in multiple multi-access networks). The router with the highest priority becomes the DR; second highest becomes the BDR. If there is a tie in priority, whichever router has the highest Router ID will become the DR.

17.1.4 OSPF neighbour states

Neighbour adjacencies will progress through several states, described in the table below.

<table>
<tr><th>State</th><th>Description</th></tr>
<tr><td>Down</td><td>Indicates that no Hellos have been heard from the neighbouring router</td></tr>
<tr><td>Init</td><td>Indicates a Hello packet has been heard from the neighbour, but two-way communication has not yet been initialized.</td></tr>
<tr><td>2-Way</td><td>Indicates that bidirectional communication has been established.
Recall that Hello packets contain a neighbour field. Thus, communication is considered 2-Way once a router sees its own Router ID in its neighbour’s Hello Packet. Designated and Backup Designated Routers are elected at this stage.</td></tr>
<tr><td>ExStart</td><td>Indicates that the routers are preparing to share link state information. Master/slave relationships are formed between routers to determine who will begin the exchange.</td></tr>
<tr><td>Exchange</td><td>Indicates that the routers are exchanging Database Descriptors (DBDs). DBDs contain a description of the router’s Topology Database. A router will examine a neighbour’s DBD to determine if it has information to share.</td></tr>
<tr><td>Loading</td><td>Indicates the routers are finally exchanging Link State Advertisements, containing information about all links connected to each router. Essentially, routers are sharing their topology tables with each other.</td></tr>
<tr><td>Full</td><td>Indicates that the routers are fully synchronized. The topology table of all routers in the area should now be identical. Depending on the role of the neighbour, the state may appear as:
<table>
<tr><td>Full/DR</td><td>Indicating that the neighbour is a Designated Router (DR)</td></tr>
<tr><td>Full/BDR</td><td>Indicating that the neighbour is a Backup Designated Router (BDR)</td></tr>
<tr><td>Full/DROther</td><td>Indicating that the neighbour is neither the DR nor BDR. On a multi-access network, OSPF routers will only form Full adjacencies with DRs and BDRs. Non-DRs and non-BDRs will still form adjacencies, but will remain in a 2-Way State. This is normal OSPF behaviour.</td></tr>
</table></td></tr>
</table>

Table 41: Neighbour adjacency states

17.1.5 OSPF network types

OSPF’s functionality is different across several different network topology types.

State	Description
Broadcast Multi-Access	Indicates a topology where broadcast occurs. Examples include Ethernet, Token Ring and ATM. OSPF characteristics are: OSPF will elect DRs and BDRs Traffic to DRs and BDRs is multicast to 224.0.0.6. Traffic from DRs and BDRs to other routers is multicast to 224.0.0.5 Neighbours do not need to be manually specified.
Point-to-Point	Indicates a topology where two routers are directly connected. An example would be a point-to-point T1. OSPF characteristics are: OSPF will not elect DRs and BDRs All OSPF traffic is multicast to 224.0.0.5 Neighbours do not need to be manually specified
Point-to-Multipoint	Indicates a topology where one interface can connect to multiple destinations. Each connection between a source and destination is treated as a point-to-point link. An example would be point to Point-to-Multipoint Frame Relay. OSPF characteristics are: OSPF will not elect DRs and BDRs. All OSPF traffic is multicast to 224.0.0.5. Neighbours do not need to be manually specified.

Non-broadcast Multi-access Network (NBMA)	Indicates a topology where one interface can connect to multiple destinations; however, broadcasts cannot be sent across a NBMA network. An example would be Frame Relay. OSPF characteristics are: OSPF will elect DRs and BDRs. OSPF neighbours must be manually defined, so all OSPF traffic is unicast instead of multicast. **Note**: on non-broadcast networks, neighbours must be manually specified, as multicast Hello's are not allowed

Table 42: OSPF functionality over different topology types

17.1.6 The OSPF hierarchy

OSPF is a hierarchical system that separates an autonomous system into individual areas. OSPF traffic can either be:

- intra-area (within one area),
- inter-area (between separate areas), or
- external (from another AS).

OSPF routers build a topology database of all links within their area, and all routers within an area will have an identical topology database. Routing updates between these routers will only contain information about links local to their area. Limiting the topology database to include only the local area conserves bandwidth and reduces CPU loads.

Area 0 is required for OSPF to function, and is considered the backbone area. As a rule, all other areas must have a connection into area 0, though this rule can be bypassed using virtual links. Area 0 is often referred to as the transit area to connect all other areas.

OSPF routers can belong to multiple areas, and therefore contain separate topology databases or each area. These routers are known as Area Border Routers (ABRs).

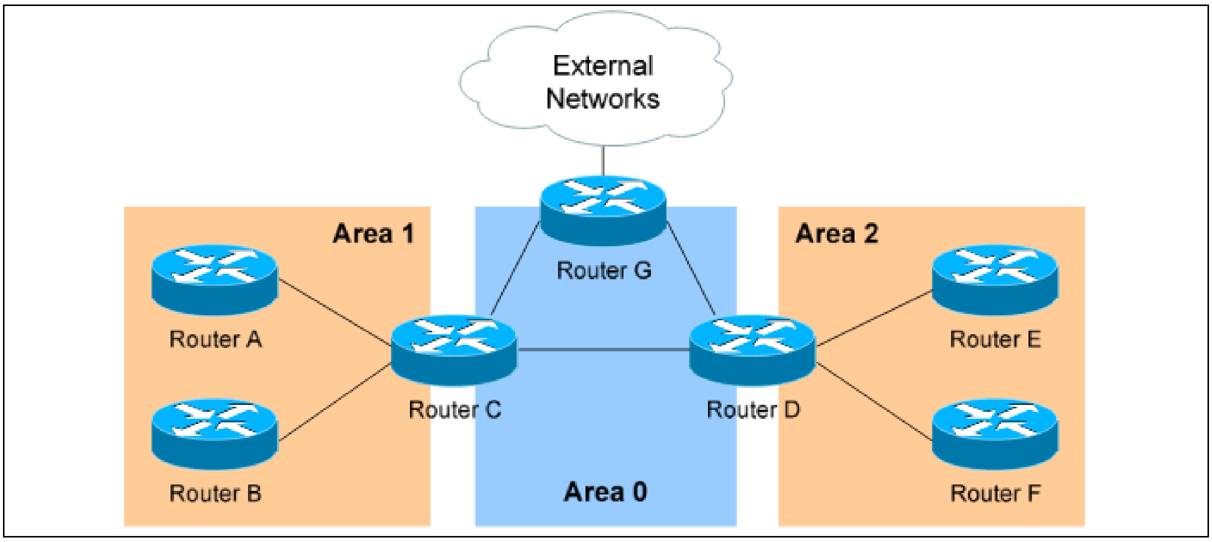


Figure 82: OSPF hierarchy

In the above example three areas exist: Area 0, Area 1, and Area 2.

Area 0 is the backbone area for this autonomous system.

Both Area 1 and Area 2 must directly connect to Area 0. Routers A and B belong fully to Area 1, while Routers E and F belong fully to Area 2. These are known as Internal Routers.

Router C belongs to both Area 0 and Area 1; so it is an ABR. Because it has an interface in Area 0, it can also be considered a Backbone Router (BR). The same can be said for Router D, as it belongs to both Area 0 and Area 2.

Router G also belongs to Area 0 however it also has a connection to the internet, which is outside this autonomous system. This makes Router G an Autonomous System Border Router (ASBR).

A router can become an ASBR in one of two ways:

- By connecting to a separate Autonomous System, such as the internet
- By redistributing another routing protocol into the OSPF process.

ASBRs provide access to external networks. OSPF defines two types of external routes, as shown in the table below.

Type 2 (E2)	Includes only the external cost to the destination network. External cost is the metric being advertised from outside the OSPF domain. This is the default type assigned to external routes.
Type 1 (E1)	Includes both the external cost, and the internal cost to reach the ASBR, to determine the total metric to reach the destination network. Type 1 routes are always preferred over Type 2 routes to the same destination.

Table 43: Types of external routes

17.1.7 OSPF router types

The four separate OSPF router types are shown in the table below.

Route Type	**Description**
Internal Router	All router interfaces belong to only one area.
Area Border Router (ABR)	Have interfaces in at least two separate areas.
Backbone Router	Have at least one interface in area 0.
Autonomous System Border Router (ABR)	Have a connection to a separate autonomous system.

17.2 Configuration package used

Package	**Sections**
ospfd	routing network interface

17.3 Configuring OSPF using the web interface

Select **Network -> OSPF**. The OSPF page appears.

There are three sections in the OSPF page:

Section	**Description**
Global Settings	Enables OSPF and configures the OSPF routing section containing global configuration parameters. The web automatically names the routing section ospfd
Topology Configuration	Configures the network sections.
Interfaces Configuration	Configures the interface sections. Defines interface configuration for OSPF and interface specific parameters

17.3.1 Global settings

The Global Settings section configures the ospfd routing section. The web automatically names the routing section 'ospfd'.

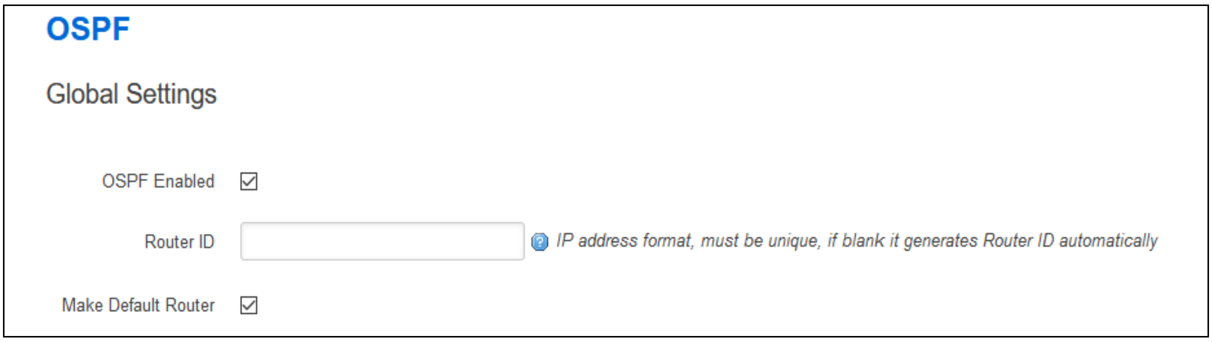


Figure 83: The OSPF global settings configuration page

Web Field / UCI / Package Option	**Description**
Web: OSPF Enabled UCI: ospfd.ospfd.enabled Opt: enabled	Enables OSPF advertisements on router. 0 — Disabled. 1 — Enabled.
Web: Router ID UCI: ospfd.ospfd.router_id Opt: router_id	This sets the Router ID of the OSPF process. The Router ID may be an IP address of the router, but need not be - it can be any arbitrary 32bit number. However it MUST be unique within the entire OSPF domain to the OSPF speaker. If one is not specified, then ospfd will obtain a router-ID automatically from the zebra daemon. Empty Range
Web: Make Default Router UCI: ospfd.ospfd.default_info_originate Opt: default_info_originate	Defines whether to originate an AS-External (type-5) LSA describing a default route into all external-routing capable areas, of the specified metric and metric type. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: ospfd.ospfd.vty_enabled Opt: vty_enabled	Enable vty for OSPFd (telnet to localhost:2604)

Table 44: Information table for OSPF global settings

17.3.2 Topology configuration

The Topology section configures the ospfd network section. This section specifies the OSPF enabled interface(s). The router can provide network information to the other OSPF routers via this interface.

Note: to advertise OSPF on an interface, the network mask prefix length for the topology configuration statement for the desired interface advertisement must be equal or smaller (IE. larger network) than the network mask prefix length for the interface.

For example, the topology configuration statement in the screenshot below does not enable OSPF on an interface with address 12.1.1.1/23, but it would on an interface with address 12.1.1.129/25.

Figure 84: The OSPF topology configuration page

Web Field / UCI / Package Option	Description
Web: Network UCI: ospfd.@network[0].ip_addr Opt: ip_addr	Specify the IP address for OSPF enabled interface. Format: A.B.C.D
Web: Mask Length UCI: ospfd.@network[0].mask_length Opt: mask_length	Specify the mask length for OSPF enabled interface. The mask length should be entered in CIDR notation.
Web: Area UCI: ospfd.@network[0].area Opt: area	Specify the area number for OSPF enabled interface.
Web: Stub Area UCI: ospfd.@network[0].stub_area Opt: stub_area	Only for non-backbone areas. Configure the area to be a stub area. That is, an area where no router originates routes external to OSPF and hence an area where all external routes are via the ABR(s). ABRs for such an area do not need to pass AS-External LSAs (type-5s) or ASBR-Summary LSAs (type-4) into the area. They need only pass Network-Summary (type-3) LSAs into such an area, along with a default-route summary. 0 — Disabled. 1 — Enabled.

Table 45: Information table for OSPF topology configuration

17.3.3 Interfaces configuration

The Interfaces section contains settings to configure the OSPF interface. It defines interface configuration for OSPF and interface specific parameters.

OSPFv2 allows packets to be authenticated using either an insecure plain text password, included with the packet, or by a more secure MD5 based HMAC (keyed-Hashing for Message AuthentiCation). Enabling authentication prevents routes being updated by unauthenticated remote routers, but still can allow routes, that is, the entire OSPF routing table, to be queried remotely, potentially by anyone on the internet, via OSPFv1.

This section defines key_chains to be used for MD5 authentication.

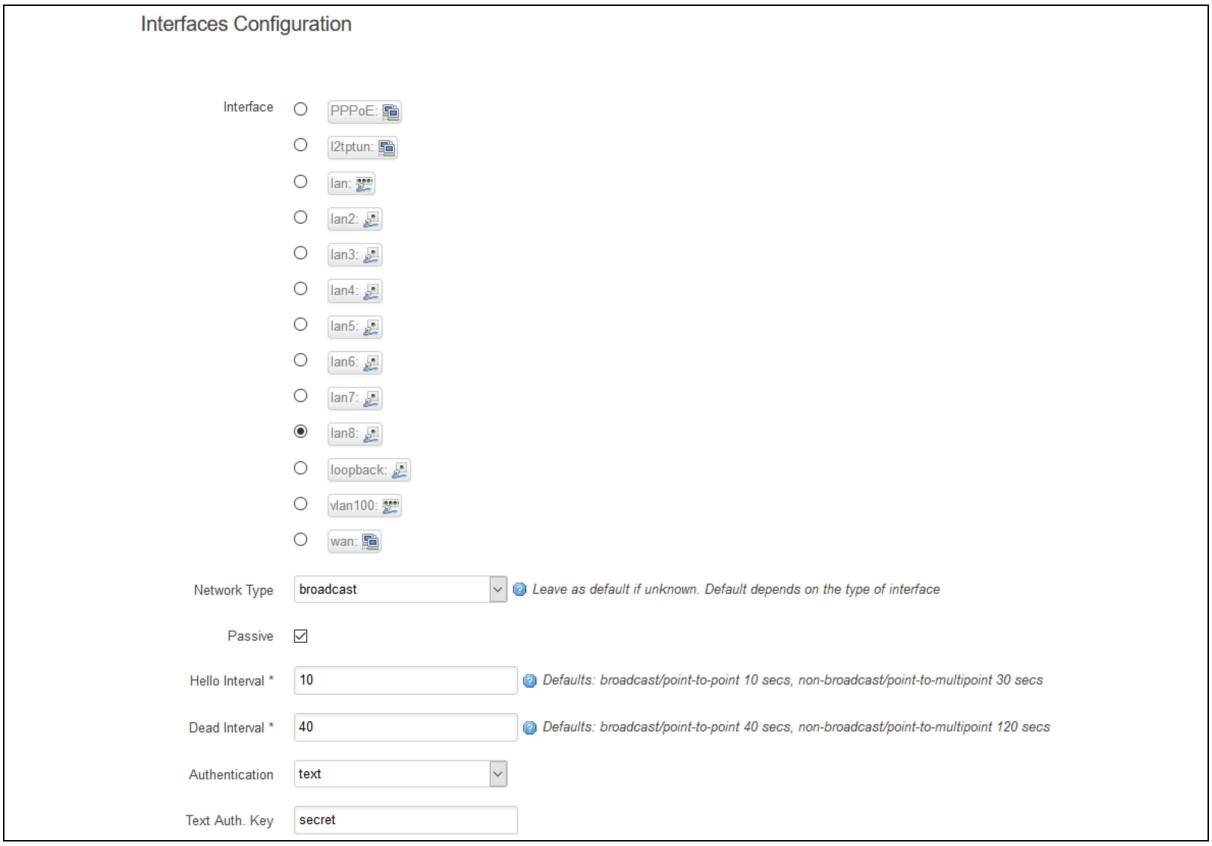


Figure 85: The OSPF interfaces configuration section

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Interface
UCI: ospfd.@interface[0].ospf_interface
Opt: ospf_interface</td><td>Defines the interface name</td></tr>
<tr><td>Web: Network Type
UCI: ospfd.@interface[0].network_type
Opt: network_type</td><td>Defines network type for specified interface.<table><tr><td>Default</td><td>Autodetect: it will be broadcast. If broadcast is not supported on that interface then use point-to-point.</td></tr><tr><td>broadcast</td><td></td></tr><tr><td>non-broadcast</td><td></td></tr><tr><td>point-to-point</td><td></td></tr><tr><td>point-to-multipoint</td><td></td></tr></table></td></tr>
<tr><td>Web: Passive
UCI: ospfd.@interface[0].passive
Opt: passive</td><td>Do not send hello packets on the given interface, but do advertise the interface as a stub link in the router-LSA (Link State Advertisement) for this router.
This allows you to advertise addresses on such connected interfaces without having to originate AS-External/Type-5 LSAs, which have global flooding scope, as would occur if connected addresses were redistributed into OSPF. This is the only way to advertise non-OSPF links into stub areas.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
</table>

<table>
<tr><td>Web: Hello Interval
UCI: ospfd.@interface[0].hello_interval
Opt: hello_interval</td><td>Defines the number of seconds for the Hello Interval timer value.
A Hello packet will be sent every timer value seconds on the specified interface. This value must be the same for all routers attached to a common network.
The default is every 10 seconds for broadcast and point-to-point interfaces, and 30 seconds for non-broadcast and point-to-multipoint interfaces.<table><tr><td>10</td><td>10 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Dead Interval
UCI: ospfd.@interface[0].dead_interval
Opt: dead_interval</td><td>Defines the number of seconds for the Dead Interval timer value used for Wait Timer and Inactivity Timer. This value must be the same for all routers attached to a common network.
The default is 40 seconds for broadcast and point-to-point interfaces, and 120 seconds for non-broadcast and point-to-multipoint interfaces. By default, the Dead Interval timer is four times the Hello interval.<table><tr><td>40</td><td>40 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Authentication
UCI: ospfd.@interface[0].auth_mode
Opt: auth_mode</td><td>OSPFv2 (only) allows packets to be authenticated via either an insecure plain text password, included with the packet, or via a more secure MD5 based HMAC (keyed-Hashing for Message AuthentiCation). Enabling authentication prevents routes being updated by unauthenticated remote routers, but still can allow routes, that is, the entire OSPF routing table to be queried remotely, potentially by anyone on the internet, via OSPFv1.<table><tr><td>no</td><td>Default value. No authentication.</td></tr><tr><td>md5</td><td>Set the interface with OSPF MD5 authentication.</td></tr><tr><td>text</td><td>Set the interface with OSPF simple password authentication.</td></tr></table></td></tr>
<tr><td>Web: Text Auth. Key
UCI: ospfd.@interface[0].text_auth_key
Opt: text_auth_key</td><td>This command sets authentication string for text authentication. text_auth_key option can have length up to 8 characters.
Displayed only when Authentication is set to text.</td></tr>
<tr><td>Web: Key ID
UCI: ospfd.@interface[0].key_id
Opt: key_id</td><td>Specifies key ID. Must be unique and match at both ends.
Displayed only when Authentication is set to MD5.</td></tr>
<tr><td>Web: MD5 Auth. Key
UCI: ospfd.@interface[0].md5_auth_key
Opt: md5_auth_key</td><td>Specify Keyed MD5 chain.
Displayed only when Authentication is set to MD5.</td></tr>
</table>

Table 46: Information table for OSPF interface commands

17.4 Configuring OSPF using the command line

OSPF is configured under the ospfd package /etc/config/ospfd.

There are three config sections: ospfd, interface and network.

You can configure multiple interface and network sections.

By default, all OSPF interface instances are named interface, instances are identified by `@interface` then the interface position in the package as a number. For example, for the first interface in the package using UCI:

```
ospfd.@interface[0]=interface
ospfd.@interface[0].ospf_interface=lan
```

Or using package options:

```
config interface
      option ospf_interface 'lan'
```

By default, all OSPF network instances are named network, it is identified by @network then the interface position in the package as a number. For example, for the first network in the package using UCI:

```
ospfd.@network[0]=network
ospfd.@network[0].ip_addr=12.1.1.1
```

Or using package options:

```
config network
      option ip_addr '12.1.1.1'
```

17.5 OSPF using UCI

```
root@VA_router:~# uci show ospfd
ospfd.ospfd=routing
ospfd.ospfd.enabled=yes
ospfd.ospfd.default_info_originate=yes
ospfd.ospfd.router_id=1.2.3.4
ospfd.@network[0]=network
ospfd.@network[0].ip_addr=12.1.1.1
ospfd.@network[0].mask_length=24
ospfd.@network[0].area=0
ospfd.@network[0].stub_area=yes
ospfd.@interface[0]=interface
ospfd.@interface[0].ospf_interface=lan8
ospfd.@interface[0].hello_interval=10
ospfd.@interface[0].dead_interval=40
ospfd.@interface[0].network_type=broadcast
ospfd.@interface[0].passive=yes
ospfd.@interface[0].auth_mode=text
ospfd.@interface[0].text_auth_key=secret
ospfd.@interface[1]=interface
ospfd.@interface[1].ospf_interface=lan7
ospfd.@interface[1].network_type=point-to-point
ospfd.@interface[1].passive=no
```

```
ospfd.@interface[1].hello_interval=30
ospfd.@interface[1].dead_interval=120
ospfd.@interface[1].auth_mode=md5
ospfd.@interface[1].key_id=1
ospfd.@interface[1].md5_auth_key=test
```

17.6 OSPF using package options

```
root@VA_router:~# uci export ospfd
package ospfd

config routing 'ospfd'
        option enabled 'yes'
        option default_info_originate 'yes'
        option router_id '1.2.3.4'

config network
        option ip_addr '12.1.1.1'
        option mask_length '24'
        option area '0'
        option stub_area 'yes'

config interface
        option ospf_interface 'lan8'
        option hello_interval '10'
        option dead_interval '40'
        option network_type 'broadcast'
        option passive 'yes'
        option auth_mode 'text'
        option text_auth_key 'secret'

config interface
        option ospf_interface 'lan7'
        option network_type 'point-to-point'
        option passive 'no'
        option hello_interval '30'
        option dead_interval '120'
```

```
        option auth_mode 'md5'
        option key_id '1'
        option md5_auth_key 'test'
```

17.7 OSPF diagnostics

17.7.1 Route status

To show the current routing status, enter:

```
root@VA_router:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.206.4.65     0.0.0.0         UG    1      0        0 usb0
10.1.0.0        0.0.0.0         255.255.0.0     U     0      0        0 eth1
10.206.4.64     0.0.0.0         255.255.255.252 U     0      0        0 usb0
11.11.11.0      0.0.0.0         255.255.255.248 U     0      0        0 gre-
GRE
89.101.154.151  10.206.4.65     255.255.255.255 UGH   0      0        0 usb0
192.168.100.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.101.1   11.11.11.1      255.255.255.255 UGH   11     0        0 gre-
GRE
192.168.104.1   11.11.11.4      255.255.255.255 UGH   20     0        0 gre-
GRE
```

Note: a route will only be displayed in the routing table when the interface is up.

17.7.2 Tracing OSPF packets

Typically, OSPF uses IP as its transport protocol. The well-known IP protocol type for OSPF traffic is 0x59. To trace OSPF packets on any interface on the router, enter: `tcpdump -i any -n proto ospf &`

```
root@VA_router:~# tcpdump -i any -n proto ospf &
root@VA_router:~# tcpdump: verbose output suppressed, use -v or -vv for
full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535
bytes
```

To stop tracing enter fg to bring tracing task to foreground, and then **<CTRL-C>** to stop the trace.

```
root@VA_router:~# fg
tcpdump -i any -n proto ospf
^C
33 packets captured
33 packets received by filter
0 packets dropped by kernel
```

17.8 Quagga/Zebra console

Quagga is the routing protocol suite embedded in the router firmware. Quagga is split into different daemons for implementation of each routing protocol. Zebra is a core daemon for Quagga, providing the communication layer to the underlying Linux kernel, and routing updates to the client daemons.

Quagga has a console interface to Zebra for advanced debugging of the routing protocols.

To access, enter:

```
root@VA_router:~# telnet localhost zebra


Entering character mode
Escape character is '^]'.




Hello, this is Quagga (version 0.99.21).
Copyright 1996-2005 Kunihiro Ishiguro, et al.




User Access Verification


Password:
```

To see OSPF routing from Zebra console, enter:

```
root@VA_router:~# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, H - HSLS, o - OLSR,
       b - BATMAN, A - Babel,
       > - selected route, * - FIB route
```

```
K>* 0.0.0.0/0 via 10.206.4.65, usb0
O   10.1.0.0/16 [110/11] via 11.11.11.1, gre-GRE, 02:35:28
C>* 10.1.0.0/16 is directly connected, eth1
C>* 10.206.4.64/30 is directly connected, usb0
O   11.11.11.0/29 [110/10] is directly connected, gre-GRE, 02:35:29
C>* 11.11.11.0/29 is directly connected, gre-GRE
K>* 89.101.154.151/32 via 10.206.4.65, usb0
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.100.0/24 is directly connected, eth0
O>* 192.168.101.1/32 [110/11] via 11.11.11.1, gre-GRE, 02:35:28
O>* 192.168.104.1/32 [110/20] via 11.11.11.4, gre-GRE, 02:30:45
O   192.168.105.1/32 [110/10] is directly connected, lo, 02:47:52
C>* 192.168.105.1/32 is directly connected, lo
```

17.8.1 OSPF debug console

When option tty_enabled (see Global settings section above) is enabled in the OSPF configuration, OSPF debug console can be accessed for advanced OSPF debugging.

To access OSPF debug console enter: `telnet localhost ospfd (password zebra)`

```
root@VA_router:~# telnet localhost ospfd

Entering character mode
Escape character is '^]'.


Hello, this is Quagga (version 0.99.21).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
```

To see OSPF routing from OSPF debug console, enter: `sh ip ospf route`

```
UUT> sh ip ospf route
============ OSPF network routing table ============
```

```
N    10.1.0.0/16           [11] area: 0.0.0.0
                           via 11.11.11.1, gre-GRE
N    11.11.11.0/29         [10] area: 0.0.0.0
                           directly attached to gre-GRE
N    192.168.101.1/32      [11] area: 0.0.0.0
                           via 11.11.11.1, gre-GRE
N    192.168.104.1/32      [20] area: 0.0.0.0
                           via 11.11.11.4, gre-GRE
N    192.168.105.1/32      [10] area: 0.0.0.0
                           directly attached to lo

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

To see OSPF neighbours from OSPF debug console, enter: `sh ip ospf neighbour`

```
root@VA_router:~# sh ip ospf neighbor


    Neighbor ID Pri State    Dead Time Address    Interface  RXmtL RqstL
DBsmL
1.1.1.1         255 Full/DR  33.961s 11.11.11.1   gre-GRE:11.11.11.5
0     0     0
```

To see OSPF interface details from OSPF debug console, enter: `sh ip ospf interface`

```

root@VA_router:~# sh ip ospf interface
base0 is up
  ifindex 8, MTU 1518 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  OSPF not enabled on this interface
eth0 is up
  ifindex 9, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  OSPF not enabled on this interface
eth1 is up
  ifindex 10, MTU 1500 bytes, BW 0 Kbit
<UP,BROADCAST,RUNNING,PROMISC,MULTICAST>
  OSPF not enabled on this interface
eth2 is down
```

```
  ifindex 11, MTU 1500 bytes, BW 0 Kbit <BROADCAST,MULTICAST>
  OSPF not enabled on this interface
eth3 is down
  ifindex 12, MTU 1500 bytes, BW 0 Kbit <BROADCAST,MULTICAST>
  OSPF not enabled on this interface
eth4 is down
  ifindex 13, MTU 1500 bytes, BW 0 Kbit <BROADCAST,MULTICAST>
  OSPF not enabled on this interface
eth5 is down
  ifindex 14, MTU 1500 bytes, BW 0 Kbit <BROADCAST,MULTICAST>
  OSPF not enabled on this interface
eth6 is down
  ifindex 15, MTU 1500 bytes, BW 0 Kbit <BROADCAST,MULTICAST>
  OSPF not enabled on this interface
eth7 is down
  ifindex 16, MTU 1500 bytes, BW 0 Kbit <BROADCAST,MULTICAST>
  OSPF not enabled on this interface
gre-GRE is up
  ifindex 19, MTU 1472 bytes, BW 0 Kbit <UP,RUNNING,MULTICAST>
  Internet Address 11.11.11.5/29, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 192.168.105.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 1.1.1.1, Interface Address 11.11.11.1
  Backup Designated Router (ID) 192.168.105.1, Interface Address 11.11.11.5
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 3.334s
  Neighbor Count is 1, Adjacent neighbor count is 1
gre0 is down
  ifindex 6, MTU 1476 bytes, BW 0 Kbit <NOARP>
  OSPF not enabled on this interface
ifb0 is down
  ifindex 2, MTU 1500 bytes, BW 0 Kbit <BROADCAST,NOARP>
  OSPF not enabled on this interface
ifb1 is down
  ifindex 3, MTU 1500 bytes, BW 0 Kbit <BROADCAST,NOARP>
```

```
  OSPF not enabled on this interface
lo is up
  ifindex 1, MTU 16436 bytes, BW 0 Kbit <UP,LOOPBACK,RUNNING>
  Internet Address 192.168.105.1/32, Broadcast 192.168.105.1, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 192.168.105.1, Network Type LOOPBACK, Cost: 10
  Transmit Delay is 1 sec, State Loopback, Priority 1
  No designated router on this network
  No backup designated router on this network
  Multicast group memberships: <None>
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in inactive
  Neighbor Count is 0, Adjacent neighbor count is 0
sit0 is down
  ifindex 7, MTU 1480 bytes, BW 0 Kbit <NOARP>
  OSPF not enabled on this interface
teql0 is down
  ifindex 4, MTU 1500 bytes, BW 0 Kbit <NOARP>
  OSPF not enabled on this interface
tunl0 is down
  ifindex 5, MTU 1480 bytes, BW 0 Kbit <NOARP>
  OSPF not enabled on this interface
usb0 is up
  ifindex 17, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  OSPF not enabled on this interface
```

To see OSPF database details from OSPF debug console, enter: `sh ip ospf database`

```
root@VA_router:~# sh ip ospf database


       OSPF Router with ID (192.168.105.1)


                Router Link States (Area 0.0.0.0)


Link ID         ADV Router      Age  Seq#       CkSum  Link count
1.1.1.1         1.1.1.1          873 0x80006236 0xd591 3
192.168.104.1   192.168.104.1    596 0x8000000a 0x3a2d 2
```

```
192.168.105.1   192.168.105.1    879 0x8000000b 0x4919 2

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
11.11.11.1      1.1.1.1          595 0x80000004 0x5712
```

18 Configuring VRRP

18.1 Overview

Virtual Router Redundancy Protocol (VRRP) is a networking protocol designed to eliminate the single point of failure inherent in the static default routed environment.

VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router controlling the IP address(es) associated with a virtual router is called the Master, and forwards packets sent to these IP addresses. The election process provides dynamic failover in the forwarding responsibility from the Master to a backup router should the Master become unavailable. This process allows the virtual router IP address(es) on the LAN to be used as the default first hop router by end hosts. The advantage gained from using VRRP is a higher availability default path without requiring configuration of dynamic routing or router discovery protocols on every end host.

Two or more routers forming the redundancy cluster are configured with the same Router ID and Virtual IP address. A VRRP router group operates within the scope of the single LAN. Additionally, the VRRP routers are configured with its initial role (Master or Backup) and the router priority, which is a factor in the master router election process. You can also configure a password authentication to protect VRRP protocol messages against spoofing.

The VRRP protocol is implemented according to internet standard RFC2338.

18.2 Configuration package used

Package	Sections
vrrp	main vrrp_group

18.3 Configuring VRRP using the web interface

To configure VRRP through the web interface, in the top menu, select **Network -> VRRP**. The VRRP page appears.

There are two sections in the VRRP page:

Section	Description
Global Settings	Enables VRRP
VRRP Group Configuration	Configures the VRRP group settings.

18.3.1 Global settings

The Global Settings section configures vrrp package main section.

To access configuration settings, click **ADD**.

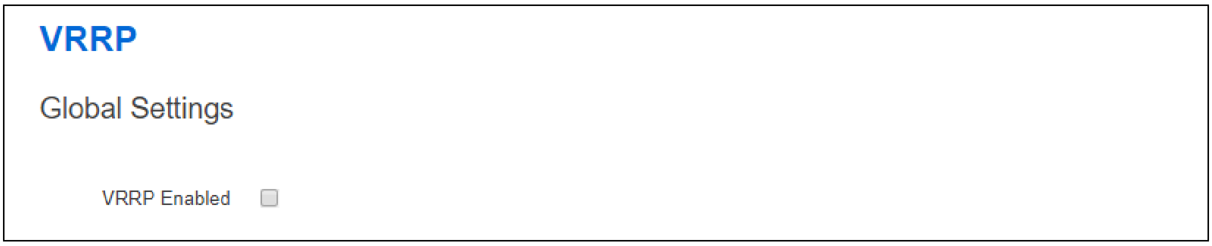


Figure 86: The VRRP global settings configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: VRRP Enabled
UCI: vrrp.main.enabled
Opt: Enabled</td><td>Globally enables VRRP on the router.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
</table>

18.3.2 VRRP group configuration settings

The VRRP Group Configuration section configures vrrp package vrrp_group section.

To access configuration settings, click **ADD**.

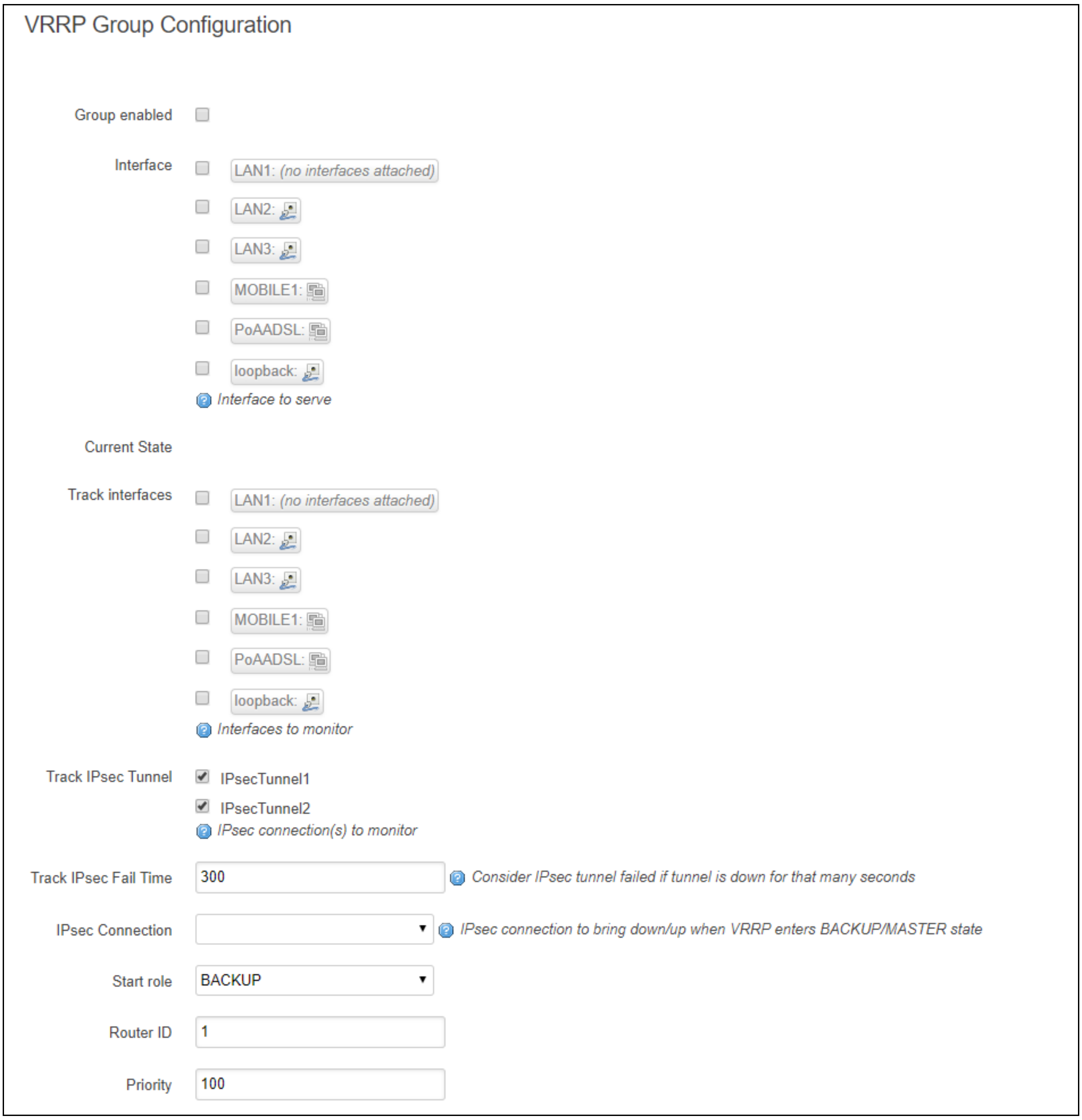


Figure 87: The VRRP group configuration page

Web Field / UCI / Package Option	Description
Web: Group Enabled UCI: vrrp.@vrrp_group[X].enabled Opt: Enabled	Enables a VRRP group on the router. 0 – Disabled. 1 – Enabled.
Web: Interface UCI: vrrp.@vrrp_group[X].interface Opt: interface	Sets the local LAN interface name in which the VRRP cluster is to operate. For example, ‘lan’. The interface name is taken from the network package and all configured interfaces will be displayed.. lan Range

<table>
<tr><td>Web: Track Interfaces
UCI: vrrp.@vrrp_group[X].track_iface
Opt: list track_iface</td><td>Defines one or more WAN interfaces that VRRP should monitor. If a monitored interface goes down on the Master VRRP router, it goes into ‘Fault’ state and the Backup VRRP router becomes the Master.
Multiple interfaces are entered using <code>uci set</code> and <code>uci add_list</code> commands. Example:
<code>uci set vrrp.@vrrp_group[0].track_iface=wan1</code>
<code>uci add_list vrrp.@vrrp_group[0].track_iface=wan2</code>
or using a list of options via package options
<code>list track_iface ‘wan1’</code>
<code>list track_iface ‘wan2’</code><table><tr><td>wan</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Track IPsec Tunnel
UCI: vrrp.@vrrp_group[X].track_ipsec
Opt: list track_ipsec</td><td>Defines one or more IPsec tunnels that VRRP should monitor. If a monitored tunnel goes down on the Master VRRP router for the configured Track IPsec Fail Time, it goes into ‘Fault’ state and the Backup VRRP router becomes the Master.
Multiple IPsec connections are entered using <code>uci set</code> and <code>uci add_list</code> commands. Example:
<code>uci set vrrp.@vrrp_group[0].track_ipsec=Tunnel1</code>
<code>uci add_list vrrp.@vrrp_group[0].track_ipsec=Tunnel2</code>
or using a list of options via package options
<code>list track_ipsec ‘Tunnel1’</code>
<code>list track_ipsec ‘Tunnel2’</code><table><tr><td>Blank</td><td>No IPSec connection to track.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Track IPsec Fail Time
UCI: vrrp.@vrrp_group[X].
track_ipsec_fail_sec
Opt: track_ipsec_fail_sec</td><td>Defines duration in seconds to determine IPsec tunnel failure<table><tr><td>300</td><td>300 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: IPSec connection
UCI:
vrrp.@vrrp_group[X].ipsec_connection
Opt: ipsec_connection</td><td>Sets which IPSec connection to bring up or down when VRRP enters ‘Backup/Master’ state.
Multiple IPSec connections are entered via the package option using a space separator. Example:
<code>option ipsec_connection ‘IPSecTunnel1 IPSecTunnel2’</code><table><tr><td>Blank</td><td>No IPSec connection to toggle.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Start role
UCI: vrrp.@vrrp_group[X].init_state
Opt: init_state</td><td>Sets the initial role in which a VRRP router starts up. In a cluster of VRRP routes, set one as a Master and the others as Backup.<table><tr><td>BACKUP</td><td></td></tr><tr><td>MASTER</td><td></td></tr></table></td></tr>
<tr><td>Web: Router ID
UCI: vrrp.@vrrp_group[X].router_id
Opt: router_id</td><td>Sets the VRRP router ID (1 to 255). All co-operating VRRP routers serving the same LAN must be configured with the same router ID.<table><tr><td>1</td><td></td></tr><tr><td>Range</td><td>1-255</td></tr></table></td></tr>
<tr><td>Web: Priority
UCI: vrrp.@vrrp_group[X].priority
Opt: priority</td><td>Sets the VRRP router’s priority. Higher values equal higher priority. The VRRP routers must use priority values between 1-254. The Master router uses a higher priority.<table><tr><td>100</td><td></td></tr><tr><td>Range</td><td>0-255</td></tr></table></td></tr>
<tr><td>Web: Advert intvl
UCI: vrrp.@vrrp_group[X].advert_int_sec
Opt: advert_int_sec</td><td>Sets the VRRP hello value in seconds. This value must match the value set on a peer.<table><tr><td>120</td><td>120 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web: Password UCI: vrrp.@vrrp_group[X].password Opt: password	Sets the password to use in the VRRP authentication (simple password authentication method). This field may be left blank if no authentication is required.
Web: Virtual IP UCI: vrrp.@vrrp_group[X].virtual_ipaddr Opt: virtual_ipaddr	Sets the virtual IP address and mask in prefix format. For example, '11.1.1.99/24'. All co-operating VRRP routers serving the same LAN must be configured with the same virtual IP address.
Web: GARP delay UCI: vrrp.@vrrp_group[X].garp_delay_sec Opt: garp_delay_sec	Sets the Gratuitous ARP message sending delay in seconds. 5 — 5 seconds Range —

Web: n/a
UCI: vrrp.@vrrp_group[X].track_ipsec
Opt: list track_ipsec
	Sets one or more IPSec connection that VRRP should monitor. If a monitored IPSec connection goes down on the Master VRRP router, it goes into 'Fault' state and the Backup VRRP router becomes the Master.
	Multiple IPsec connections are entered using `uci set` and `uci add_list` commands. Example:
	`uci set vrrp.@vrrp_group[0].track_ipsec=Tunnel1`
	`uci add_list vrrp.@vrrp_group[0].track_ipsec=Tunnel2`
	or using a list of options via package options
	`list track_ipsec 'Tunnel1'`
	`list track_ipsec 'Tunnel2'`
	Blank — No IPSec connection to track.
	Range —

Table 47: Information table for VRRP group settings

18.4 Configuring VRRP using command line

The configuration file is stored on **/etc/config/vrrp**.

There are two config sections – **main** and **vrrp_group**.

Multiple VRRP groups can be configured. By default, all VRRP group instances are named 'vrrp_group'. It is identified by @vrrp_group then the vrrp_group position in the package as a number. For example, for the first vrrp_group in the package using UCI:

```
vrrp.@vrrp_group[0]=vrrp_group
vrrp.@vrrp_group[0].enabled=1
```

Or using package options:

```
config vrrp_group
        option enabled '1'
```

However, to better identify, it is recommended to give the vrrp_group instance a name. For example, to define a vrrp_group instance named 'g1' using UCI, enter:

```
vrrp.g1.vrrp_group
vrrp.g1.enabled=1
```

To define a named keepalive instance using package options, enter:

```
config vrrp_group 'g1'
        option enabled '1'
```

18.4.1 VRRP using UCI

To view the configuration in UCI format, enter:

```
root@VA_router:~# uci show vrrp
vrrp.main=vrrp
vrrp.main.enabled=yes
vrrp.g1=vrrp_group
vrrp.g1.enabled=yes
vrrp.g1.interface=lan
vrrp.g1.track_iface=WAN MOBILE
vrrp.g1.init_state=BACKUP
vrrp.g1.router_id=1
vrrp.g1.priority=100
vrrp.g1.advert_int_sec=120
vrrp.g1.password=secret
vrrp.g1.virtual_ipaddr=10.1.10.150/16
vrrp.g1.garp_delay_sec=5
vrrp.g1.ipsec_connection=Test
vrrp.g1.track_ipsec=conn1 conn2
```

18.4.2 VRRP using package options

To view the configuration in package option format, enter:

```
root@VA_router:~# uci export vrrp
package vrrp

config vrrp 'main'
        option enabled 'yes'

config vrrp_group 'g1'
        option enabled 'yes'
        option interface 'lan'
        list track_iface 'WAN'
        list track_iface 'MOBILE'
```

```
        option init_state 'BACKUP'
        option router_id '1'
        option priority '100'
        option advert_int_sec '120'
        option password 'secret'
        option virtual_ipaddr '10.1.10.150/16'
        option garp_delay_sec '5'
        option ipsec_connection 'Test'
        list track_ipsec 'conn1'
        list track_ipsec 'conn2'
```

19 Configuring Routing Information Protocol (RIP)

19.1 Introduction

RIP is a dynamic routing algorithm used on IP-based internet networks.

A distance-vector routing algorithm is used by RIP to assist in maintaining network convergence. It uses a metric or ‘hop’ count as the only routing criteria. Each route is advertised with the number of hops a datagram would take to reach the destination network. The maximum metric for RIP is 15. This limits the size of the network that RIP can support. Smaller metrics are more efficient-based on the cost associated with each metric.

RIP protocol is most useful as an Interior Gateway Protocol (IGP). An IGP refers to the routing protocol used within a single autonomous system. There may be a number of autonomous systems, using different routing protocols, combined together to form a large network.

In most networking environments, RIP is not the preferred choice for routing as its time to converge and scalability are poor compared to EIGRP or OSPF.

19.1.1 RIP characteristics

RIP is a standardised distance vector protocol, designed for use on smaller networks. RIP was one of the first true distance vector routing protocols, and is supported on a wide variety of systems.

RIP adheres to the following distance vector characteristics:

- RIP sends out periodic routing updates, every 30 seconds
- RIP sends out the full routing table every periodic update
- RIP uses a form of distance as its metric, in this case, hopcount
- RIP uses the Bellman-Ford distance vector algorithm to determine the best path to a particular destination

Other characteristics of RIP include:

- RIP supports IP and IPX routing
- RIP utilizes UDP port 520
- RIP routes have an administrative distance of 120
- RIP has a maximum hopcount of 15 hops. Any network that is 16 hops away or more is considered unreachable to RIP, thus the maximum diameter of the network is 15 hops. A metric of 16 hops in RIP is considered a poison route or infinity metric.

If multiple paths exist to a particular destination, RIP will load balance between those paths, by default, up to 4, only if the metric (hopcount) is equal. RIP uses a round-robin system of load-balancing between equal metric routes, which can lead to pinhole congestion.

For example, two paths might exist to a particular destination, one going through a 9600 baud link, the other via a T1. If the metric (hopcount) is equal, RIP will load-balance, sending an equal amount of traffic down the 9600 baud link and the T1. This will cause the slower link to become congested.

19.1.2 RIP versions

RIP has two versions, Version 1 (RIPv1) and Version2 (RIPv2).

RIPv1 (RFC 1058) is classful, and therefore does not include the subnet mask with its routing table updates. Because of this, RIPv1 does not support Variable Length Subnet Masks (VLSMs). When using RIPv1, networks must be contiguous, and subnets of a major network must be configured with identical subnet masks. Otherwise, route table inconsistencies or worse will occur.

RIPv1 sends updates as broadcasts to address 255.255.255.255.

RIPv2 (RFC 2453) is classless, and therefore does include the subnet mask with its routing table updates. RIPv2 fully supports VLSMs, allowing discontinuous networks and varying subnet masks to exist.

Other enhancements offered by RIPv2 include:

- Routing updates are sent via multicast, using address 224.0.0.9
- Encrypted authentication can be configured between RIPv2 routers
- Route tagging is supported

RIPv2 can interoperate with RIPv1. By default:

- RIPv1 routers will sent only Version 1 packets
- RIPv1 routers will receive both Version 1 and 2 updates
- RIPv2 routers will both send and receive only Version 2 updates

Virtual Access **ripd** package supports RIP version 2 as described in RFC2453 and RIP version 1 as described in RFC1058. It is part of Quagga suite of applications for routing.

19.2 Configuration package used

Package	Sections
ripd	routing interface key_chain offset

19.3 Configuring RIP using the web interface

To configure RIP using the web interface, select **Network->RIP**. The RIP page appears.

There are four sections in the RIP page.

Section	Description
Global Settings	Enables RIP and configures the RIP routing section containing global configuration parameters. The web automatically names the routing section `ripd`
Interfaces Configuration	Configures the `interface` sections. Defines interface configuration for RIP and interface specific parameters.
Offset Configuration	Configures the `offset` sections for metric manipulation.
MD5 Authentication Key Chains	Configures the `key_chain` sections. Defines MD5 authentication settings.

19.3.1 Global settings

The web browser automatically names the routing section 'ripd'.

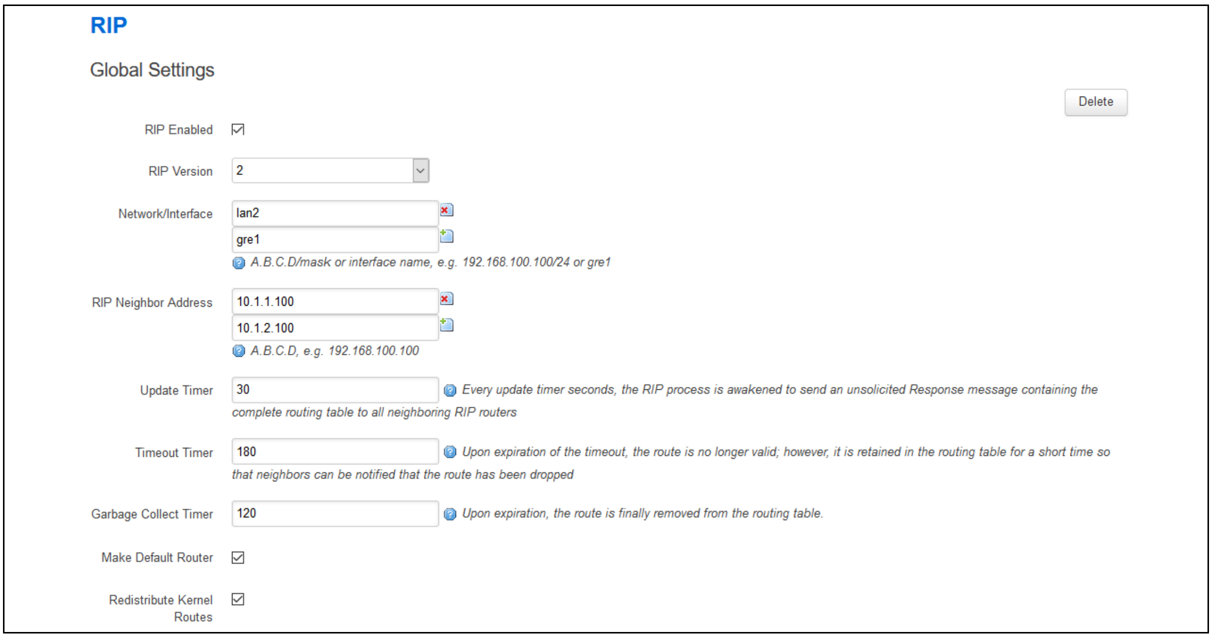


Figure 88: The RIP global settings configuration page

Web Field / UCI / Package Option	Description
Web: RIP Enabled UCI: ripd.ripd.enabled Opt: enabled	Enables RIP advertisements on router. 0 — Disabled. 1 — Enabled.
Web: RIP Version UCI: ripd.ripd.version Opt: version	Specifies the RIP version that will be used. Version 2 is recommended. 1 — RIP version 1 2 — RIP version 2

Web: Network/Interface
UCI: ripd.ripd.network
Opt: list network
	Defines the list of the interfaces that will be used to advertise RIP packets.
	Format: A.B.C.D/mask or interface name
	Multiple RIP interfaces are entered using `uci set` and `uci add_list` commands. Example:
	`uci set ripd.ripd.network=lan1`
	`uci add_list ripd.ripd.network=lan2`
	or using a list of options via package options
	`list network ‘lan1’`
	`list network ‘lan2’`

Web: RIP Neighbor Address
UCI: ripd.ripd.neighbor
Opt: list neighbor
	Specifies the list of RIP neighbours. When a neighbour doesn’t understand multicast, this command is used to specify neighbours. In some cases, not all routers will be able to understand multicasting, where packets are sent to a network or a group of addresses. In a situation where a neighbour cannot process multicast packets, it is necessary to establish a direct link between routers. The neighbour command allows the network administrator to specify a router as a RIP neighbour.
	Multiple RIP neighbours are entered using `uci set` and `uci add_list` commands. Example:
	`uci set ripd.ripd.neighbor=1.1.1.1`
	`uci add_list ripd.ripd.neighbor=2.2.2.2`
	or using a list of options via package options
	`list neighbor ‘1.1.1.1’`
	`list neighbor ‘2.2.2.2’`

Web: Update Timer UCI: ripd.ripd.tb_update_sec Opt: tb_update_sec	Every update timer seconds, the RIP process is awakened to send an unsolicited response message containing the complete routing table to all neighbouring RIP routers. 30 Range
Web: Timeout Timer UCI: ripd.ripd.tb_timeout_sec Opt:tb_timeout_sec	Defines timeout in seconds. Upon expiration of the timeout, the route is no longer valid; however, it is retained in the routing table for a short time so that neighbours can be notified that the route has been dropped. 180 Range
Web: Garbage Collect Timer UCI: ripd.ripd.tb_garbage_sec Opt: tb_garbage_sec	Upon expiration of the *Garbage-Collection* timer, the route is finally removed from the routing table. This timer starts when *Timeout* timer expires or when route is advertised as “unreachable”. The reason for using this two-stage removal method (marking-deleting) is to give the router that declared the route no longer reachable a chance to propagate this information to other routers. When the timer expires the route is deleted. If during the garbage collection period a new *RIP Response* for the route is received, then the deletion process is aborted: the *Garbage-Collection* timer is cleared, the route is marked as valid again, and a new *Timeout* timer starts. 120 Range

Web: Make Default Router UCI: ripd.ripd.default_info_originate Opt: default_info_originate	Advertising a default route via RIP. 0 | Disable. 1 | Enable.
Web: Redistribute Kernel Routes UCI: ripd.ripd.redistribute_kernel_routes Opt: redistribute_kernel_routes	Redistributes routing information from kernel route entries into the RIP tables. 0 | Disable. 1 | Enable.
Web: n/a UCI: ripd.ripd.vty_enabled Opt: vty_enabled	Enable vty for RIPd (telnet to localhost:2602).

Table 48: Information table for RIP global settings

19.3.2 Offset configuration

This section is used for RIP metric manipulation. RIP metric is a value for distance in the network. Usually, ripd package increments the metric when the network information is received. Redistributed routes' metric is set to 1.

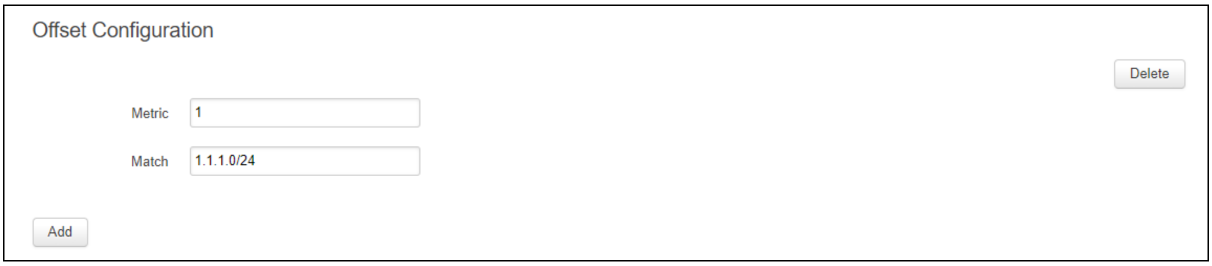


Figure 89: The RIP global settings configuration page

Web Field / UCI / Package Option	Description
Web: Metric UCI: ripd.@offset[0].metric Opt: metric	Defines the metric offset value. This modifies the default metric value for redistributed and connected routes. 1 Range
Web: Match UCI: ripd.@offset[0].match_network Opt: match_network	Defines the prefixes to match. Format: A.B.C.D/mask

Table 49: Information table for RIP offset commands

19.3.3 Interfaces configuration

Figure 90: The RIP interfaces configuration page

Web Field / UCI / Package Option	Description
Web: Interface UCI: ripd.@interface[0].rip_interface Opt: rip_interface	Specifies the interface name.
Web: Split Horizon UCI: ripd.@interface[0].split_horizon Opt: split_horizon	Prohibits the router from advertising a route back onto the interface from which it was learned. 0: Disable. 1: Enable.
Web: Poison Reverse UCI: ripd.@interface[0].poison_reverse Opt: poison_reverse	Router tells its neighbour gateways that one of the gateways is no longer connected. Notifies the gateway, setting the hop count to the unconnected gateway to 16 which would mean "infinite". 0: Disable. 1: Enable.
Web: Passive UCI: ripd.@interface[0].passive Opt: passive	Sets the specified interface to passive mode. On passive mode interface, all receiving packets are processed as normal and ripd does not send either multicast or unicast RIP packets except to RIP neighbour specified with a neighbour command. 0: Disable 1: Enable
Web: Authentication UCI: ripd.@interface[0].auth_mode Opt: auth_mode	RIPv2 (only) allows packets to be authenticated via either an insecure plain text password, included with the packet, or via a more secure MD5 based HMAC (keyed-Hashing for Message AuthentiCation). Enabling authentication prevents routes being updated by unauthenticated remote routers, but still can allow routes, that is, the entire RIP routing table, to be queried remotely, potentially by anyone on the internet, via RIPv1. no: Default value. No authentication. md5: Sets the interface with RIPv2 MD5 authentication. text: Sets the interface with RIPv2 simple password authentication.
Web: Text Auth. Key UCI: ripd.@interface[0].auth_key Opt: auth_key	This command sets the authentication string for text authentication. The string must be shorter than 16 characters.
Web: MD5 Key Chain Name UCI: ripd.@interface[0].key_chain Opt: key_chain	Specifiy Keyed MD5 chain.

Table 50: Information table for RIP interface configuration

19.3.4 MD5 authentication key chains

RIPv2 (only) allows packets to be authenticated using either an insecure plain text password, included with the packet, or by a more secure MD5 based HMAC (keyed-Hashing for Message AuthentiCation). Enabling authentication prevents routes being updated by unauthenticated remote routers, but still can allow routes, that is, the entire RIP routing table, to be queried remotely, potentially by anyone on the internet, using RIPv1.

This section defines key_chains to be used for MD5 authentication.

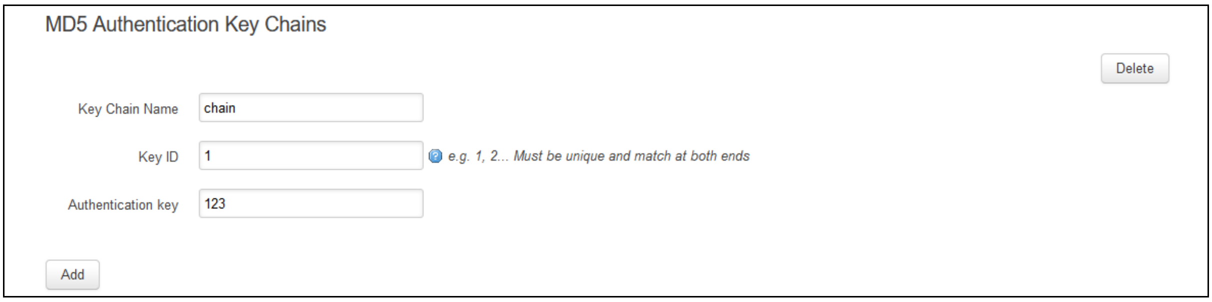


Figure 91: The MD5 authentication key chains configuration section

Web Field / UCI / Package Option	Description
Web: Key Chain Name UCI: ripd.@key_chain[0].key_chain_name Opt: key_chain_name	Specifies chain name
Web: Key ID UCI: ripd.@key_chain[0].key_id Opt: key_id	Specifies key ID. Must be unique and match at both ends.
Web: Authentication key UCI: ripd.@key_chain[0].auth_key Opt: auth_key	Specifiy Keyed MD5 chain.

Table 51: Information table for MD5 authentication key chains commands

19.4 Configuring RIP using command line

RIP is configured under the ripd package **/etc/config/ripd.**

There are four config sections ripd, interface, key_chain and offset.

You can configure multiple interface, key_chain and offset sections.

By default, all RIP interface instances are named interface, it is identified by `@interface` then the interface position in the package as a number. For example, for the first interface in the package using UCI:

```
ripd.@interface[0]=interface
ripd.@interface[0].rip_interface=lan
```

Or using package options:

```
config interface
      option rip_interface 'lan'
```

By default, all RIP key_chain instances are named key_chain, it is identified by `@key_chain` then the key_chain position in the package as a number. For example, for the first key_chain in the package using UCI:

```
ripd.@key_chain[0]=key_chain
ripd.@key_chain[0].key_chain_name=Keychain1
```

Or using package options:

```
config key_chain
      option key_chain_name 'Keychain1'
```

By default, all RIP offset instances are named offset, it is identified by `@offset` then the offset position in the package as a number. For example, for the first offset in the package using UCI:

```
ripd.@offset[0]=offset
ripd.@offset[0].metric=1
```

Or using package options:

```
config offset
      option metric '1'
```

19.4.1 RIP using UCI

```
root@VA_router:~# uci show ripd
ripd.ripd=routing
ripd.ripd.version=2
ripd.ripd.enabled=yes
ripd.ripd.network=lan2 gre1
ripd.ripd.neighbor=10.1.1.100 10.1.2.100
ripd.ripd.tb_update_sec=30
ripd.ripd.tb_timeout_sec=180
ripd.ripd.tb_garbage_sec=120
ripd.ripd.default_info_originate=yes
ripd.ripd.redistribute_kernel_routes=yes
ripd.@interface[0]=interface
ripd.@interface[0].rip_interface=lan
ripd.@interface[0].auth_mode=no
ripd.@interface[0].split_horizon=1
ripd.@interface[0].poison_reverse=0
ripd.@interface[0].passive=0
ripd.@interface[1]=interface
ripd.@interface[1].rip_interface=lan2
ripd.@interface[1].split_horizon=1
ripd.@interface[1].poison_reverse=0
ripd.@interface[1].passive=0
```

```
ripd.@interface[1].auth_mode=text
ripd.@interface[1].auth_key=secret
ripd.@interface[2]=interface
ripd.@interface[2].rip_interface=lan3
ripd.@interface[2].split_horizon=1
ripd.@interface[2].poison_reverse=0
ripd.@interface[2].passive=0
ripd.@interface[2].auth_mode=md5
ripd.@interface[2].key_chain=Keychain1
ripd.@key_chain[0]=key_chain
ripd.@key_chain[0].key_chain_name=Keychain1
ripd.@key_chain[0].key_id=1
ripd.@key_chain[0].auth_key=123
ripd.@offset[0]=offset
ripd.@offset[0].metric=1
ripd.@offset[0].match_network=10.1.1.1/24
```

19.4.2 RIP using package options

```
root@VA_router:~# uci export ripd
package ripd

config routing 'ripd'
        option version '2'
        option enabled 'yes'
        list network 'lan2'
        list network 'gre1'
        list neighbor '10.1.1.100'
        list neighbor '10.1.2.100'
        option tb_update_sec '30'
        option tb_timeout_sec '180'
        option tb_garbage_sec '120'
        option default_info_originate 'yes'
        option redistribute_kernel_routes 'yes'

config interface
        option rip_interface 'lan'
```

```
        option auth_mode 'no'
        option split_horizon '1'
        option poison_reverse '0'
        option passive '0'

config interface
        option rip_interface 'lan2'
        option split_horizon '1'
        option poison_reverse '0'
        option passive '0'
        option auth_mode 'text'
        option auth_key 'textsecret'

config interface
        option rip_interface 'lan3'
        option split_horizon '1'
        option poison_reverse '0'
        option passive '0'
        option auth_mode 'md5'
        option key_chain 'keychain1'

config key_chain
        option key_chain_name 'Keychain1'
        option key_id '1'
        option auth_key '123'

config offset
      option metric '1'
      option match_network '10.1.1.1/24'
```

19.5 RIP diagnostics

19.5.1 Route status

To show the current routing status, enter `route -n`:

```
root@VA_router:~#
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
0.0.0.0         10.205.154.65   0.0.0.0         UG    1      0      0 usb0
10.1.0.0        0.0.0.0         255.255.0.0     U     0      0      0 eth1
10.205.154.64   0.0.0.0         255.255.255.252 U     0      0      0 usb0
11.11.11.0      0.0.0.0         255.255.255.248 U     0      0      0 gre-
GRE
89.101.154.151  10.205.154.65   255.255.255.255 UGH   0      0      0 usb0
192.168.100.0   0.0.0.0         255.255.255.0   U     0      0      0 eth0
192.168.104.1   11.11.11.4      255.255.255.255 UGH   3      0      0 gre-
GRE
192.168.154.154 11.11.11.1      255.255.255.255 UGH   2      0      0 gre-
GRE
```

Note: a route will only be displayed in the routing table when the interface is up.

19.5.2 Tracing RIP packets

RIP uses UDP port 520. To trace RIP packets on any interface on the router, enter:
`tcpdump -i any -n -p port 520 &`

```
root@VA_router:~# tcpdump -i any -n -p port 520 &
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535
bytes
```

To stop tracing enter `fg` to bring tracing task to foregraound, and then **<CTRL-C>** to stop the trace.

```
root@VA_router:~# fg
tcpdump -i any -n -p port 67
^C
33 packets captured
33 packets received by filter
```

```
0 packets dropped by kernel
```

19.5.3 Quagga / zebra console

Quagga is the routing protocol suite embedded in the router firmware. Quagga is split into different daemons for implementation of each routing protocol. Zebra is a core daemon for Quagga, providing the communication layer to the underlying Linux kernel, and routing updates to the client daemons.

Quagga has a console interface to Zebra for advanced debugging of the routing protocols.

To access, enter: `telnet localhost zebra` (password: zebra)

```
root@VA_router:~# telnet localhost zebra


Entering character mode
Escape character is '^]'.



Hello, this is Quagga (version 0.99.21).
Copyright 1996-2005 Kunihiro Ishiguro, et al.



User Access Verification


Password:
```

To see RIP routing information from Zebra console, enter: `sh ip route`

```
root@VA_router:~# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, H - HSLS, o - OLSR,
       b - BATMAN, A - Babel,
       > - selected route, * - FIB route

K>* 0.0.0.0/0 via 10.205.154.65, usb0
C>* 10.1.0.0/16 is directly connected, eth1
C>* 10.205.154.64/30 is directly connected, usb0
C>* 11.11.11.0/29 is directly connected, gre-GRE
K>* 89.101.154.151/32 via 10.205.154.65, usb0
C>* 127.0.0.0/8 is directly connected, lo
```

```
C>* 192.168.100.0/24 is directly connected, eth0
R>* 192.168.104.1/32 [120/3] via 11.11.11.4, gre-GRE, 15:54:47
C>* 192.168.105.1/32 is directly connected, lo
R>* 192.168.154.154/32 [120/2] via 11.11.11.1, gre-GRE, 16:09:51
```

19.5.4 RIP debug console

When option **tty_enabled** (see Global settings section above) is enabled in the RIP configuration, RIP debug console can be accessed for advanced RIP debugging.

To access RIP debug console enter: `telnet localhost ripd` (password zebra)

```
root@VA_router:~# telnet localhost ripd


Entering character mode
Escape character is '^]'.



Hello, this is Quagga (version 0.99.21).
Copyright 1996-2005 Kunihiro Ishiguro, et al.



User Access Verification


Password:
```

To see RIP status from RIP debug console, enter: `sh ip rip`

```
root@VA_router:~# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Tag Time
C(i) 11.11.11.0/29      0.0.0.0               1 self              0
R(n) 192.168.104.1/32   11.11.11.4            3 11.11.11.1        0 02:48
C(i) 192.168.105.1/32   0.0.0.0               1 self              0
R(n) 192.168.154.154/32 11.11.11.1            2 11.11.11.1        0 02:48
```

To see RIP status from RIP debug console, enter: `sh ip rip status`

```
root@VA_router:~# sh ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 17 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:
  Default version control: send version 2, receive version 2
    Interface        Send  Recv   Key-chain
    gre-GRE          2     2
    lo               2     2
  Routing for Networks:
    11.0.0.0/8
    192.168.105.1/32
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
    11.11.11.1               0         0       120   00:00:20
  Distance: (default is 120)
```

20 Configuring Multi-WAN

Multi-WAN is used for managing WAN interfaces on the router, for example, 3G interfaces to ensure high-availability. You can customise Multi-WAN for various needs, but its main use is to ensure WAN connectivity and provide a failover system in the event of failure or poor coverage.

Multi-WAN periodically does a health check on the interface. A health check comprises of a configurable combination of the following:

- interface state
- pings to an ICMP target
- signal level checks using signal threshold, RSCP threshold and ECIO threshold option values

A fail for any of the above health checks, results in a fail. After a configurable number of health check failures, Multi-WAN will move to the next highest priority interface. Multi-WAN will optionally stop the failed interface and start the new interface, if required.

In some circumstances, particulary in mobile environments, it is desirable for a primary interface to be used whenever possible. In this instance Multi-WAN will perform a health check on the primary interface after a configurable period. If the health checks pass for the configured number of recovery health checks then the primary will be used.

20.1 Configuration package used

Package	Sections
multiwan	config wan

20.2 Configuring Multi-WAN using the web interface

In the top menu, select **Network -> Multi-Wan**. The Multi-WAN page appears.

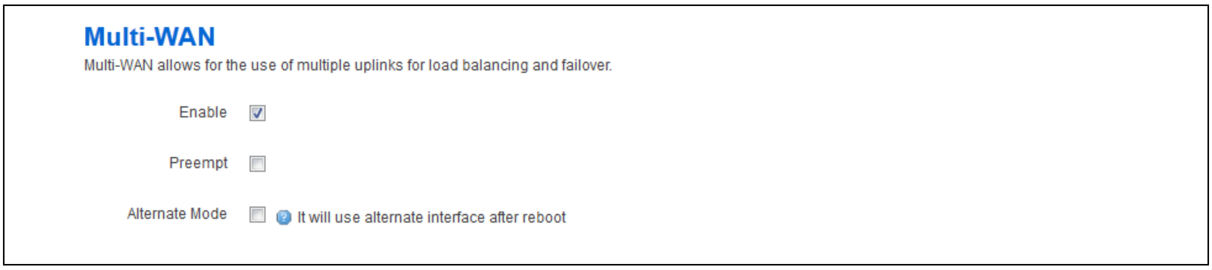


Figure 92: The multi-WAN page

Web Field / UCI / Package Option	Description
Web: Enable UCI: multiwan.config.enabled Opt: enabled	Enables or disables Multi-WAN. 0 — Disabled. 1 — Enabled.
Web: Preempt UCI: multiwan.config.preempt Opt: preempt	Enables or disables pre-emption for Multi-WAN. If enabled the router will keep trying to connect to a higher priority interface depending on timer set by ifup_retry_sec 0 — Disabled. 1 — Enabled.
Web: Alternate Mode UCI: multiwan.config.alt_mode Opt: alt_mode	Enables or disables alternate mode for Multi-WAN. If enabled the router will use an alternate interface after reboot. 0 — Disabled. 1 — Enabled.

Table 52: Information table for multi-WAN page

When you have enabled Multi-WAN, you can add the interfaces that will be managed by Multi-WAN, for example 3G interfaces.

The name used for Multi-WAN must be identical, including upper and lowercases, to the actual interface name defined in your network configuration. To check the names and settings are correct, select **Network - > Interfaces** and view the Interfaces Overview page.

In the WAN interfaces section, enter the name of the WAN interface to configure, and then click **Add**. The new section for configuring specific parameters appears.

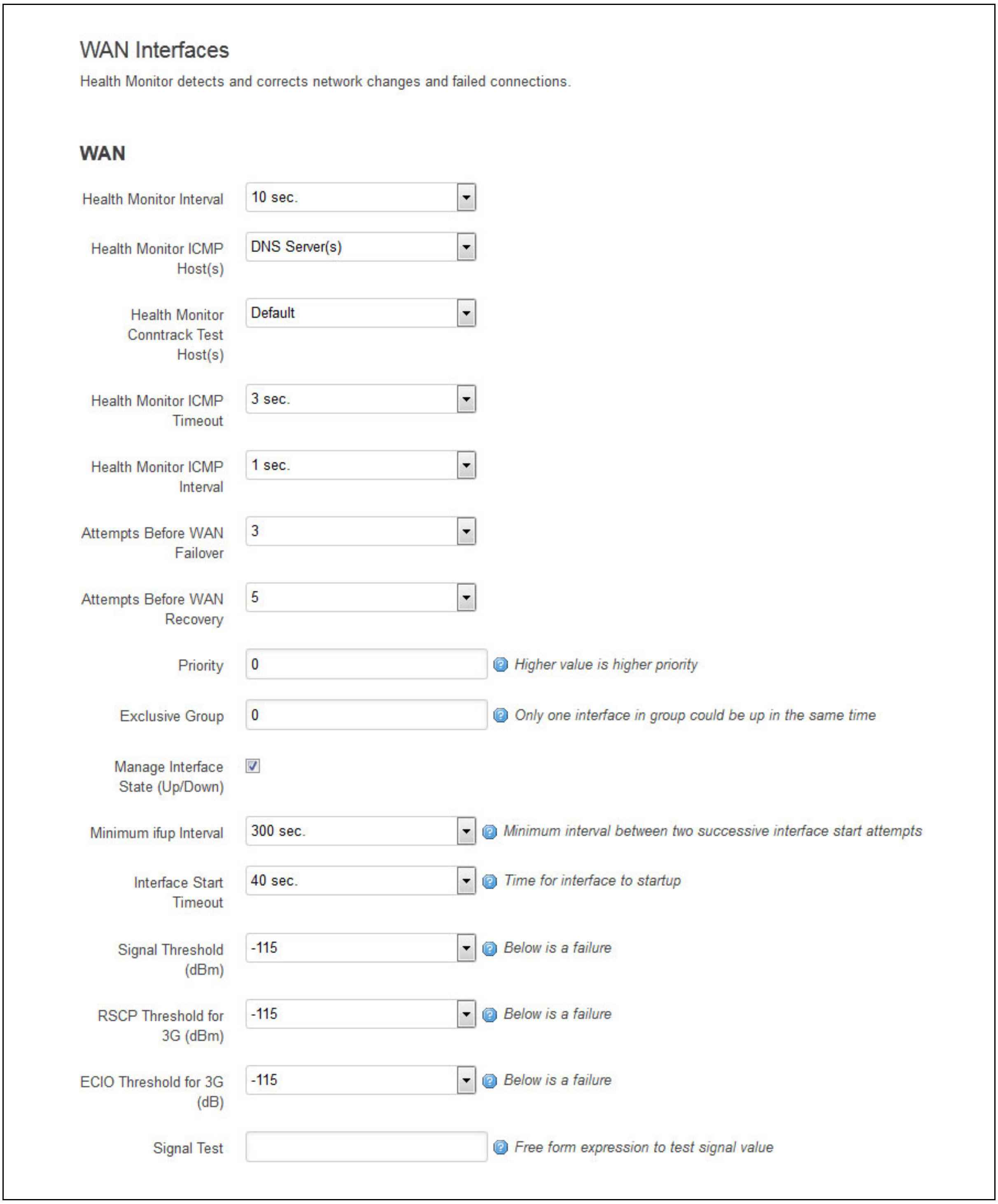


Figure 93: Example interface showing failover traffic destination as the added multi-WAN interface

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Health Monitor Interval
UCI: multiwan.wan.health_interval
Opt: health_interval</td><td>Sets the period to check the health status of the interface. The Health Monitor interval will be used for:
• Interface state checks
• Ping interval
• Signal strength checks</td></tr>
<tr><td>Web: Health Monitor ICMP Host(s)
UCI: multiwan.wan.icmp_hosts
Opt: icmp_hosts</td><td>Sends health ICMPs to configured value DNS servers by default. Configure to any address.<table>
<tr><td>Disable</td><td>Disables the option.</td></tr>
<tr><td>DNS servers</td><td>DNS IP addresses will be used.</td></tr>
<tr><td>WAN Gateway</td><td>Gateway IP address will be used.</td></tr>
<tr><td>Custom</td><td>Ability to provide IP address. Multiple pings targets can be entered, comma separated. Pings to both must fail for health check to fail. Example:
option icmp_hosts ‘1.1.1.1,2.2.2.2’</td></tr>
</table></td></tr>
<tr><td>Web: Health Monitor Conntrack Test Host(s)
UCI: multiwan.wan.conntrack_hosts
Opt: conntrack_hosts</td><td>Conntrack is the feature used to track if there is any traffic to and from an IP destination within the health interval.
The Conntrack_hosts option defines the IP for conntrack to track, usually the icmp_host IP is used.
If traffic to the conntrack_hosts IP is detected then multiwan does not send a ping health check to the icmp_host; otherwise a ping is sent as normal to the icmp_host.
By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated.
Conntrack is generally used to limit the traffic sent on a GSM network.<table>
<tr><td>Default</td><td>Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.</td></tr>
<tr><td>Disable</td><td>Conntrack disabled.</td></tr>
<tr><td>Custom</td><td>Specifies an IP other than the icmp_host for conntrack to track.</td></tr>
</table></td></tr>
<tr><td>Web: Health Monitor ICMP Timeout
UCI: multiwan.wan.timeout
Opt: timeout</td><td>Sets Ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.<table>
<tr><td>3</td><td>Wait 3 seconds for ping reply.</td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Health Monitor ICMP Interval
UCI: multiwan.wan.icmp_interval
Opt: icmp_interval</td><td>Defines the interval between multiple pings sent at each health check<table>
<tr><td>1</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Health Monitor ICMP Count
UCI: multiwan.wan.icmp_count
Opt: icmp_count</td><td>Defines the number of pings to send at each health check.<table>
<tr><td>1</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Attempts Before WAN Failover
UCI: multiwan.wan.health_fail_retries
Opt: health_fail_retries</td><td>Sets the amount of health monitor retries before interface is considered a failure.<table>
<tr><td>3</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Attempts Before WAN Recovery
UCI: multiwan.wan.health_recovery_retries
Opt: health_recovery_retries</td><td>Sets the number of health monitor checks before the interface is considered healthy. Only relevent if pre-empt mode is enbled.<table>
<tr><td>5</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Priority
UCI: multiwan.wan.priority
Opt: priority</td><td>Specifies the priority of the interface. The higher the value, the higher the priority.<table>
<tr><td>0</td><td></td></tr>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Web: Manage Interface State (Up/Down)
UCI: multiwan.wan.manage_state
Opt: manage_state</td><td>Defines whether multi-wan will start and stop the interface.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
<tr><td>Web: Exclusive Group
UCI: multiwan.wan.exclusive_group
Opt: exclusive_group</td><td>Defines the group to which the interface belongs; only one interface can be active.<table><tr><td>0</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Minimum ifup Interval
UCI: multiwan.wan.ifup_retry_sec
Opt: ifup_retry_sec</td><td>Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled.<table><tr><td>300</td><td>Retry primary interface every 300 seconds.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Interface Start Timeout
UCI: multiwan.wan.ifup_timeout
Opt: ifup_timeout</td><td>Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail.<table><tr><td>40</td><td>40 seconds.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Signal Threshold (dBm)
UCI: multiwan.wan.signal_threshold
Opt: signal_threshold</td><td>Specifies the minimum signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for sig_dbm in mobile diagnostics.-115.<table><tr><td></td><td>Disabled</td></tr><tr><td>Range</td><td>-46 to -115 dBm</td></tr></table></td></tr>
<tr><td>Web: RSCP Threshold (dBm)
UCI: multiwan.wan.rscp_threshold
Opt: rscp_threshold</td><td>Specifies the minimum RSCP signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for rscp_dbm in mobile diagnostics.<table><tr><td>-115</td><td>Disabled</td></tr><tr><td>Range</td><td>-46 to -115 dBm</td></tr></table></td></tr>
<tr><td>Web: ECIO Threshold (dB)
UCI: multiwan.wan.ecio_threshold
Opt: ecio_threshold</td><td>Specifies the minimum ECIO signal strength in dB before considering if the interface fails signal health check. Uses the value stored for ecio_db in mobile diagnostics.<table><tr><td>-115</td><td>Disabled</td></tr><tr><td>Range</td><td>-46 to -115 dB</td></tr></table></td></tr>
<tr><td>Web: Signal Test
UCI: multiwan.wan.signal_test
Opt: signal_test</td><td>Defines a script to test various signal characteristics in multiwan signal test. For example:
option signal_test '(tech == 0) then (sig_dbm > -70) else (rscp_dbm > -105 and ecio_db > -15)'
This states that when technology is GSM, a health fail is determined when signal strength is less than -70dBm. When technology is not GSM a health fail occurs when either rscp_dbm falls below -105dBm or ecio_db falls below -15dB
Tech values are:<table><tr><td>0</td><td>GSM</td></tr><tr><td>1</td><td>GSM Compact</td></tr><tr><td>2</td><td>UTRAN</td></tr><tr><td>3</td><td>GSM w/EGPRS</td></tr><tr><td>4</td><td>UTRAN w/HSPDA</td></tr><tr><td>5</td><td>UTRAN w/HSUPA</td></tr><tr><td>6</td><td>UTRAN w/HSUPA and HSDPA</td></tr><tr><td>7</td><td>E-UTRAN</td></tr></table></td></tr>
</table>

Table 53: Information table for multi-WAN interface page

20.3 Configuring Multi-WAN using UCI

Multi-WAN UCI configuration settings are stored on **/etc/config/multiwan**

Run UCI export or show commands to see multiwan UCI configuration settings. A sample is shown below.

```
root@VA_router:~# uci export multiwan

package multiwan

config multiwan 'config'
        option preempt 'yes'
        option alt_mode 'no'
        option enabled 'yes'
config interface 'wan'
        option disabled '0'
        option health_interval '10'        option health_fail_retries '3'
        option health_recovery_retries '5'
        option priority '2'
        option manage_state 'yes'
        option exclusive_group '0'
        option ifup_retry_sec '40'
        option icmp_hosts 'disable'
        option icmp_interval ‘1’
        option timeout ‘3’
        option icmp_count ‘1’
        option conntrack_hosts ‘disable’        option signal_threshold '-
111'
        option rscp_threshold '-90'
        option ecio_threshold '-15'
        option ifup_timeout_sec '120'

root@VA_router:~# uci show multiwan
multiwan.config=multiwan
multiwan.config.preempt=yes
multiwan.config.alt_mode=no
multiwan.config.enabled=yes
multiwan.wan=interface
multiwan.wan.disabled=0
multiwan.wan.health_interval=10multiwan.wan.health_fail_retries=3
multiwan.wan.health_recovery_retries=5
multiwan.wan.priority=2
multiwan.wan.manage_state=yes
```

```
multiwan.wan.exclusive_group=0
multiwan.wan.ifup_retry_sec=36000
multiwan.wan.icmp_hosts=disable
multiwan.wan.timeout=3
multiwan.wan.icmp_interval ‘1’
multiwan.wan.timeout ‘3’
multiwan.wan.icmp_count ‘1’
multiwan.wan.conntrack_hosts ‘disable’
multiwan.wan.signal_threshold=-111
multiwan.wan.rscp_threshold=-90
multiwan.wan.ecio_threshold=-15
```

20.4 Multi-WAN diagnostics

The multi-WAN package is linked to the network interfaces within /etc/config/network.

Note: multi-WAN will not work if the WAN connections are on the same subnet and share the same default gateway.

To view the multi-WAN package, enter:

```
root@VA_router:~# uci export multiwan
package multiwan

config multiwan 'config'
        option enabled 'yes'
        option preempt 'yes'
        option alt_mode 'no'

config interface 'ADSL'
        option health_interval '10'
        option icmp_hosts 'dns'
        option timeout '3'
        option health_fail_retries '3'
        option health_recovery_retries '5'
        option priority '1'
        option manage_state 'yes'
        option exclusive_group '0'
        option ifup_retry_sec '300'
        option ifup_timeout_sec '40'
```

```
config interface 'Ethernet'
        option health_interval '10'
        option icmp_hosts 'dns'
        option timeout '3'
        option health_fail_retries '3'
        option health_recovery_retries '5'
        option priority '2'
        option manage_state 'yes'
        option exclusive_group '0'
        option ifup_retry_sec '300'
        option ifup_timeout_sec '40'
```

The following output shows the multi-WAN standard stop/start commands for troubleshooting.

```
root@VA_router:~# /etc/init.d/multiwan
Syntax: /etc/init.d/multiwan [command]
```

Available commands:

```
        start   Start the service
        stop    Stop the service
        restart Restart the service
        reload  Reload configuration files (or restart if that fails)
        enable  Enable service autostart
        disable Disable service autostart
```

When troubleshooting, make sure that the routing table is correct using `route –n`.

Ensure all parameters in the multi-WAN package are correct. The name used for multi-WAN interfaces must be identical, including upper and lowercases, to the interface name defined in the network configuration.

To check the names and settings are correct, browse to **Network - > interfaces** (or alternatively, run: cat/etc/config/network through CLI).

Enter the name of the WAN interface to configure, and then click **Add**. The new section for configuring specific parameters will appear.

21 Automatic operator selection

This section describes how to configure and operate the Automatic Operator Selection feature of a Virtual Access router.

When the roaming SIM is connected, the radio module has the ability to scan available networks. The router, using mobile and multi-WAN packages, finds available networks to create and sort interfaces according to their signal strength. These interfaces are used for failover purposes.

21.1 Configuration package used

Package	Sections
Multiwan	General, interfaces
Mobile	Main, Template interface
Network	2G/3G/4G interface

21.2 Configuring automatic operator selection via the web interface

While the router boots up it checks for mobile networks. Based on available networks, the router creates interfaces and the multiwan package is used to run failover between interfaces. Typically these auto-generated interfaces are sorted by signal strength.

Details for these interfaces are provided in the mobile package. When you have created the interfaces, Multi-WAN manages the operation of primary (predefined) and failover (auto created) interfaces.

Multi-WAN periodically does a health check on the active interface. A health check comprises of a configurable combination of the following:

- interface state
- pings to an ICMP target
- signal level checks using signal threshold, RSCP threshold and ECIO threshold option values

A fail for any of the above health checks results in an overall fail. After a configurable number of health check failures, multiwan will move to the next highest priority interface. Multi-WAN will optionally stop the failed interface and start the new interface, if required.

In some circumstances, particulary in mobile environments, it is desirable for a primary interface to be used whenever possible. In this instance, if the active interface is a not the primary interface, multiwan will perform a health check on the primary interface after a configurable period. If the health checks pass for the configured number of recovery health checks then the primary interface will be used.

There are typcailly three scenarios:

- Primary Mobile Provider (PMP) + roaming: pre-empt enabled
- PMP + roaming: pre-empt disabled
- No PMP + roaming

21.2.1 Scenario 1: PMP + roaming: pre-empt enabled

21.2.1.1 Overview

In this scenario, the PMP interface is used whenever possible.

The PMP interface is attempted first. When the health checks fail on the PMP interface, and Multi-WAN moves to an autogenerated interface, a timer is started `multiwan option ifup_retry_sec`. On expiration of this timer, multiwan will disconnect the current interface and retry the PMP interface.

The PMP interface will then be used if the configurable number of health checks pass the checks.

21.2.1.2 Software operation

1. Multiwan first attemts to bring up the PMP interface. If the PMP interface connects within the time set by multiwan option `ifup_timeout` continue to step 2. Otherwise go to step 4.
2. A health check is periodically done on the PMP interface as determined by the multiwan option `health_interval`. If the health check fails for the number of retries (multiwan option `health_fail_retries`), disconnect the PMP interface.
3. Connect the first auto-generated interface.
4. If the interface connects within the time set by multiwan option `ifup_timeout` continue to step 5, otherwise multiwan moves to the next auto-generated interface.
5. Wait until the health check fails on the auto-generated interface, or until the PMP interface is available to connect after it was disconnected in step 2. (multiwan option `ifup_retry_sec`).
6. Disconnect auto-generated interface.
7. If the interface was disconnected due to health check failure then connect the next auto-generated interface and repeat step 4. If the interface was disconnected because `ifup_retry_sec` of PMP interface timed out, then go back to step 1 and repeat the process.

The PMP predefined interface is defined in the network package. Ensure the interface name matches the interface name defined in the multiwan package.

21.2.1.3 Create a primary predefined interface

In the web interface top menu, go to **Network ->Interfaces**. The Interfaces page appears.

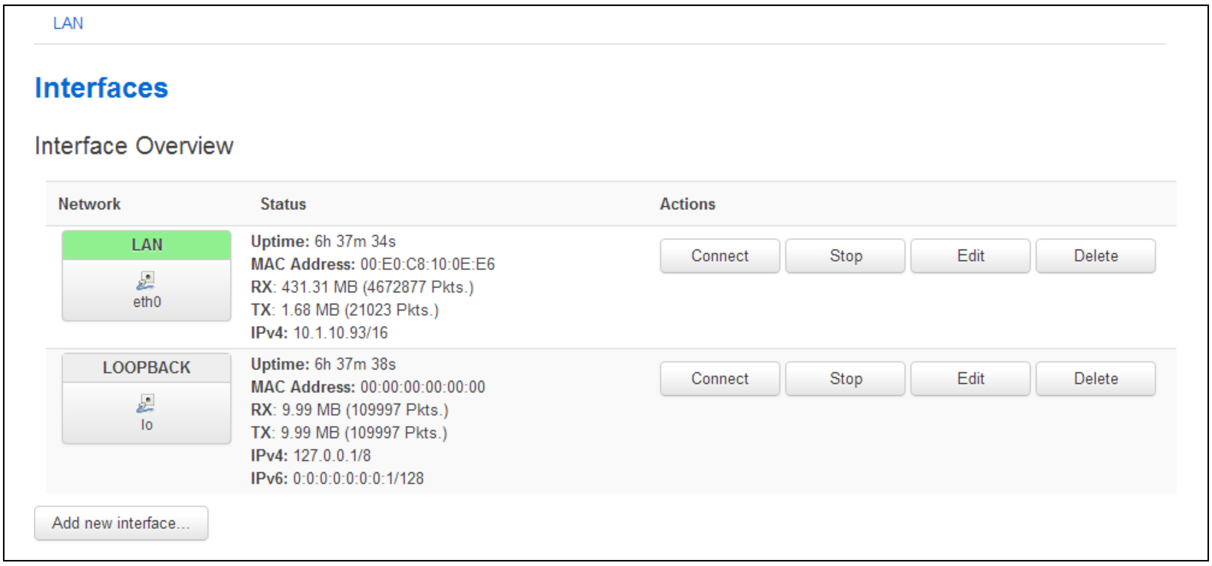


Figure 94: The interface overview page

Click **Add new interface…** The Create Interface page appears.

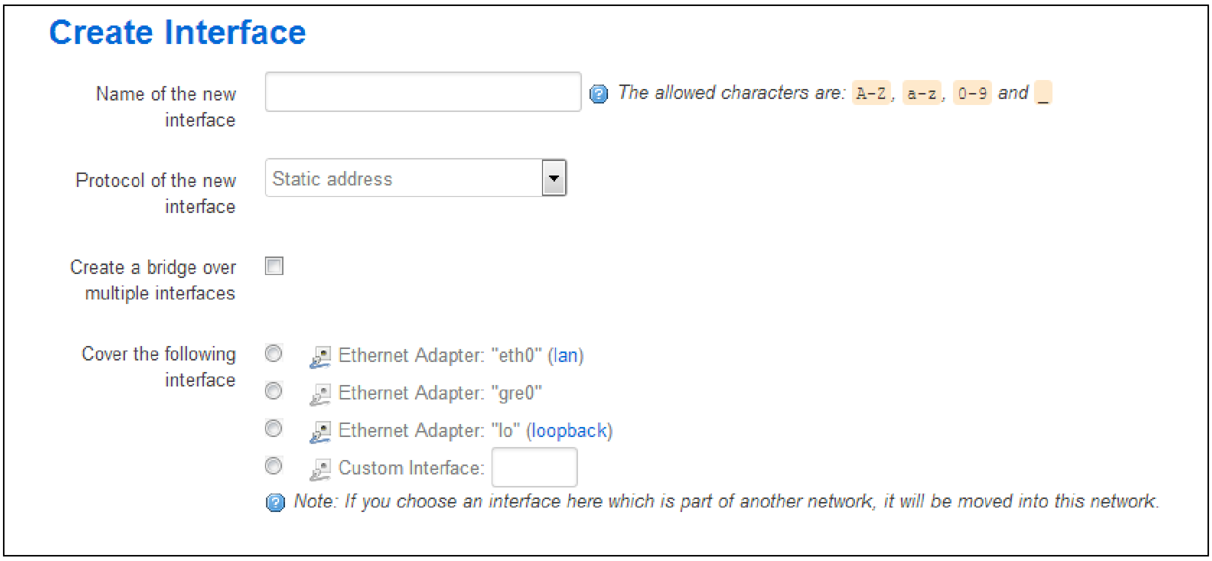


Figure 95: The create interface page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface
UCI: network.3g_s<sim-number>_<short-operator-name>.
Opt: 3g_s<sim-number>_<short-operator-name>.</td><td>Type the name of the new interface.
Type the interface name in following format:
3g_s<sim-number>_<short-operator-name>. Where <sim-number> is number of roaming SIM (1 or 2) and <short-operator-name> is first four alphanumeric characters of operator name (as reported by 'AT+COPS=?' command).
Type the short operator name in lower case, for example:
<table>
<tr><th>Operator name</th><th>First four alphanumeric numbers</th></tr>
<tr><td>Vodafone UK</td><td>voda</td></tr>
<tr><td>O2 – UK</td><td>o2uk</td></tr>
<tr><td>Orange</td><td>oran</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td>Web: Protocol of the new interface
UCI: network.[..x..].proto
Opt: proto</td>
<td>Protocol type. Select LTE/UMTS/GPRS/EV-DO.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>IPv4 tunnels that carry IPv6.</td></tr>
<tr><td>IPv6 over IPv4</td><td>IPv6 over IPv4 tunnel.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation.</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol.</td></tr>
<tr><td>PPP</td><td>Point to Point Protocol.</td></tr>
<tr><td>PPPoE</td><td>Point to Point Protocol over Ethernet.</td></tr>
<tr><td>PPPoATM</td><td>Point to Point Protocol over ATM.</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table>
</td>
</tr>
<tr>
<td>Web: Create a bridge over multiple interfaces
UCI: network.[..x..].typeOpt: type</td>
<td>Enables bridge between two interfaces.
<table>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
</table>
</td>
</tr>
<tr>
<td>Web: Cover the following interface
UCI: network.[..x..].ifname
Opt: ifname</td>
<td>Selects interfaces for bridge connection.</td>
</tr>
</table>

Table 54: Information table for the create interface page

Click **Submit**. The Common Configuration page appears.

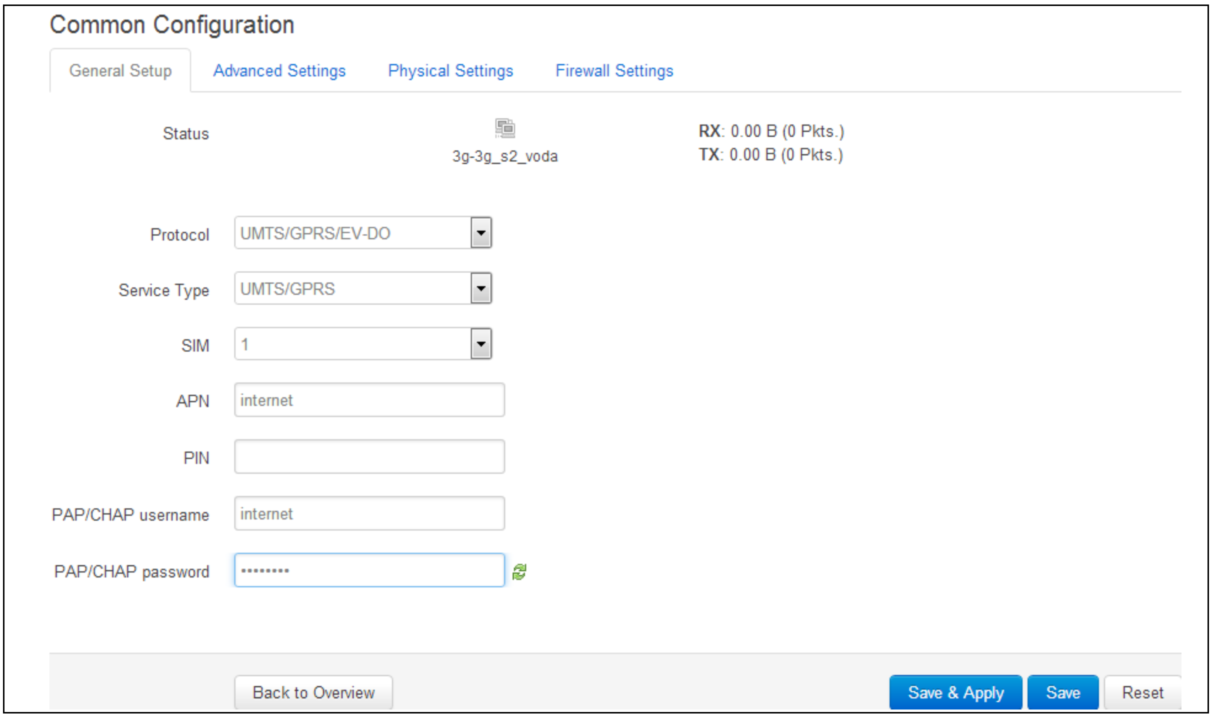


Figure 96: The common configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Protocol
UCI: network.[..x..].proto
Opt: proto</td><td>Protocol type. Select LTE / UMTS / GPRS / EV-DO.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>IPv4 tunnels that carry IPv6.</td></tr>
<tr><td>IPv6 over IPv4</td><td>IPv6 over IPv4 tunnel.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation.</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol.</td></tr>
<tr><td>PPP</td><td>Point to Point Protocol.</td></tr>
<tr><td>PPPoE</td><td>Point to Point Protocol over Ethernet.</td></tr>
<tr><td>PPPoATM</td><td>Point to Point Protocol over ATM.</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: Service Type
UCI: network.[..x..].service
Opt: service</td><td>Service type that will be used to connect to the network.
<table>
<tr><td>gprs_only</td><td>Allows GSM module to only connect to GPRS network.</td></tr>
<tr><td>lte_only</td><td>Allows GSM module to only connect to LTE network.</td></tr>
<tr><td>cdma</td><td>Allows GSM module to only connect to CDMA network.</td></tr>
<tr><td>auto</td><td>GSM module will automatically detect the best available technology code.</td></tr>
</table></td></tr>
<tr><td>Web: SIM
UCI: network.[..x..].sim
Opt: sim</td><td>Select SIM 1 or SIM 2.
<table>
<tr><td>auto</td><td>Automatically detects which SIM slot is used.</td></tr>
<tr><td>SIM 1</td><td>Selects Sim from slot 1.</td></tr>
<tr><td>SIM 2</td><td>Selects Sim from slot 2.</td></tr>
</table></td></tr>
<tr><td>Web: APN
UCI: network.[..x..].apn
Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: APN username
UCI: network.[..x..].username
Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: APN password
UCI: network.[..x..].password
Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: Modem Configuration
UCI: N/A
Opt: N/A</td><td>Click the link if you need to configure additional options from Mobile Manager.</td></tr>
</table>

Table 55: Information table for the general set up section

Click **Save & Apply**.

21.2.1.4 Set multi-WAN options for primary predefined interface

On the web interface go to **Network ->Multi-Wan**. The Multi-WAN page appears.

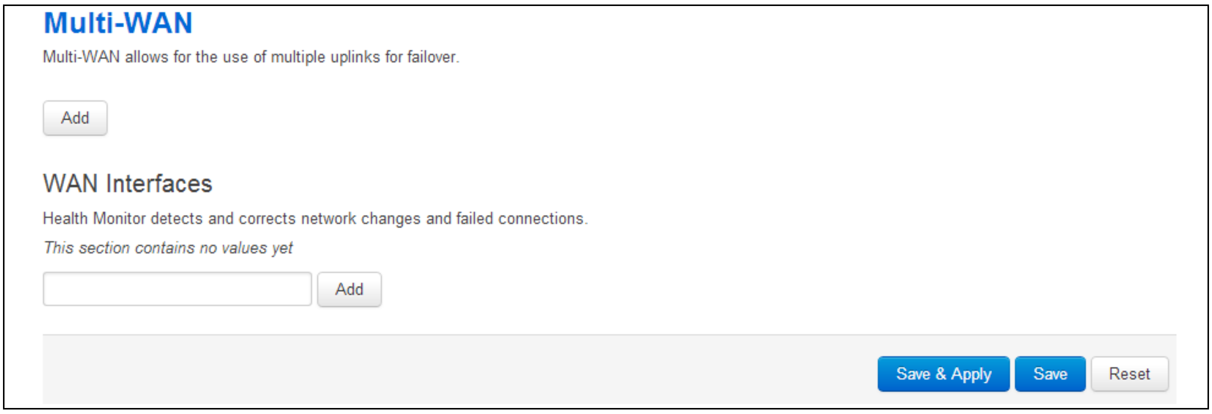


Figure 97: The multi-WAN page

In the WAN Interfaces section, type in the name of the Multi-WAN interface.

Click **Add**. The Multi-WAN page appears.

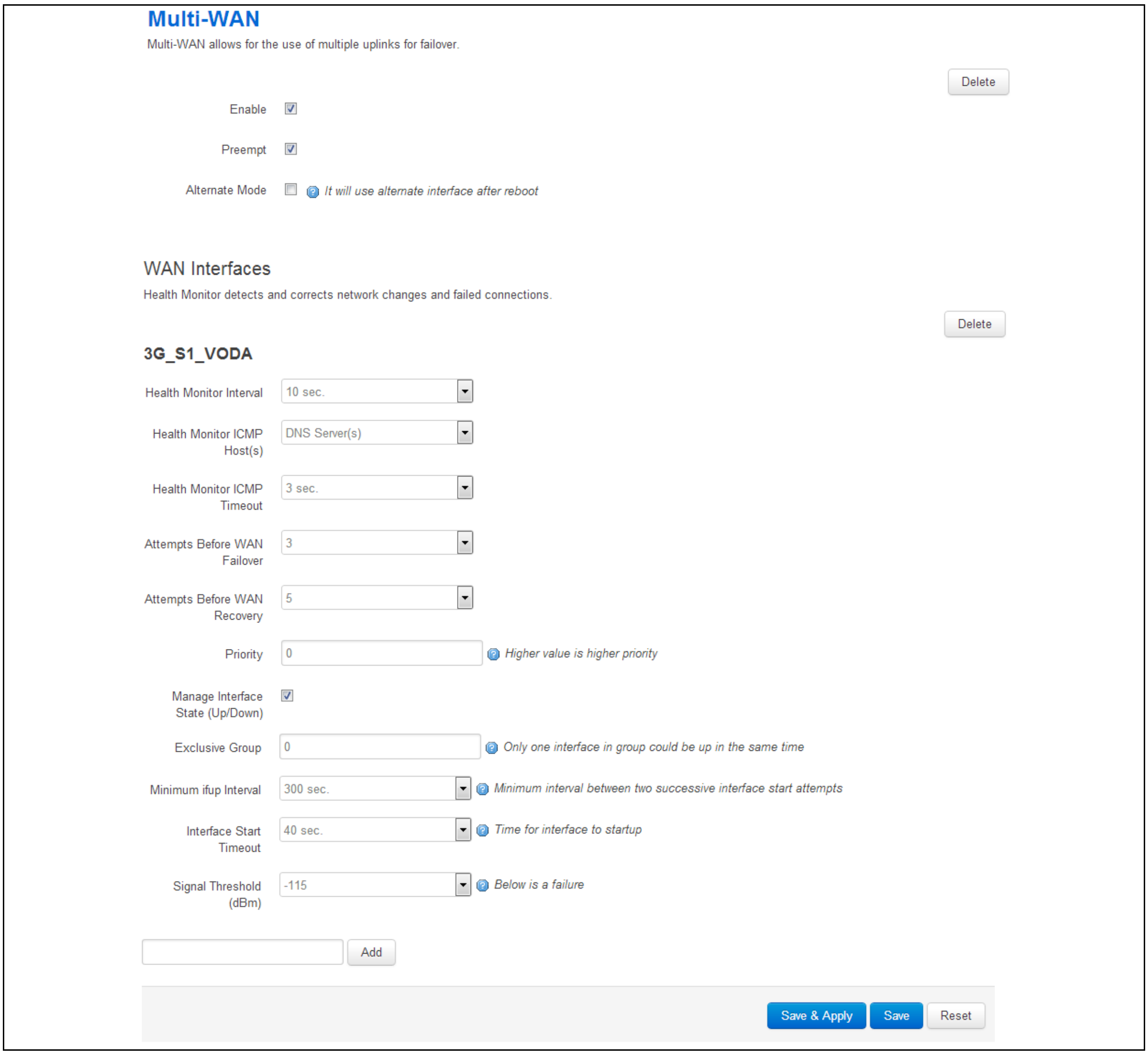


Figure 98: The multi-WAN page

Web Field / UCI / Package Option	Description
Web: Enable UCI: multiwan.config.enabled Opt: enabled	Enables multiwan. 0 Disabled. 1 Enabled.
Web: Preempt UCI: multiwan.config.preempt Opt: preempt	Enables or disables pre-emption for multiwan. If enabled the router will keep trying to connect to a higher priority interface depending on timer set. 0 Disabled. 1 Enabled.
Web: Alternate Mode UCI: multiwan.config.alt Opt: alt	Enables or disables alternate mode for multiwan. If enabled the router will use an alternate interface after reboot. 0 Disabled. 1 Enabled.
Web: WAN Interfaces UCI: multiwan.3g_s<sim-number>_<short-operator-name> Opt: 3g_s<sim-number>_<short-operator-name>	Provide the same interface name as chosen in multiwan section below and click **Add**.

Web: Health Monitor Interval UCI: multiwan.[..x..].health_interval Opt: health_interval	Sets the period to check the health status of the interface. The Health Monitor interval will be used for: • Interface state checks • Ping interval • Signal strength checks
Web: Health Monitor ICMP Host(s) UCI: multiwan.[..x..].icmp_hosts Opt: icmp_hosts	Specifies the target IP address for ICMP packets. Disable: Disables the option. DNS servers: DNS IP addresses will be used. WAN Gateway: Gateway IP address will be used. custom: Ability to provide IP address.
Web: Health Monitor Conntrack Test Host(s) UCI: multiwan.wan.conntrack_hosts Opt: conntrack_hosts	Conntrack is the feature used to track if there is any traffic to and from an IP destination within the health interval. Conntrack_hosts option defines the IP for conntrack to track – usually the icmp_host IP is used. If traffic to the conntrack_hosts IP is detected then multiwan does not send a ping health check to the icmp_host otherwise a ping is sent as normal to the icmp_host. By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated. Conntrack is generally used to limit the traffic sent on a GSM network. Default: Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes. Disable: Conntrack disabled. Custom: Specifies an IP other than the icmp_host for conntrack to track.
Web: Health Monitor ICMP Timeout UCI: multiwan.[..x..].timeout Opt: timeout	Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at. 3: Wait 3 seconds for ping reply. Range
Web: Health Monitor ICMP Interval UCI: multiwan.wan.icmp_interval Opt: icmp_interval	Defines the interval between multiple pings sent at each health check. 1 Range
Web: Health Monitor ICMP Count UCI: multiwan.wan.icmp_count Opt: icmp_count	Defines the number of pings to send at each health check. 1 Range
Web: Attempts Before WAN Failover UCI: multiwan. [..x..].health_fail_retries Opt: health_fail_retries	Sets the amount of health monitor retries before interface is considered a failure. 3 Range
Web: Attempts Before WAN Recovery UCI: multiwan. [..x..].health_recovery_retries Opt: health_recovery_retries	Sets the number of health monitor checks before the interface is considered healthy. Only relevent if pre-empt mode is enbled. 5 Range
Web: Priority UCI: multiwan.[..x..].priority Opt: priority	Specifies the priority of the interface. The higher the value, the higher the priority. This multiwan interface priority must be higher than the one specified in the priority field in the ‘Roaming Interface Template’ page described in the following section. 0 Range
Web: Exclusive Group UCI: multiwan.[..x..].exclusive_group Opt: exclusive_group	Defines the group to which the interface belongs; only one interface can be active. 0 Range

Web: Manage Interface State (Up/Down) UCI: multiwan.[..x..].manage_state Opt: manage_state	Defines whether multi-WAN will start and stop the interface. Select **Enabled**. 0 | Disabled. 1 | Enabled.
Web: Minimum ifup Interval UCI: multiwan.[..x..].ifup_retry_sec Opt: ifup_retry_sec	Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled.
Web: Interface Start Timeout UCI: multiwan.[..x..].ifup_timeout Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail. Choose timer greater than 120 seconds. 40 | 40 seconds Range |
Web: Signal Threshold (dBm) UCI: multiwan.[..x..].signal_threshold Opt: signal_threshold	Specifies the minimum signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for sig_dbm in mobile diagnostics. -115 | Disabled. Range | -46 to -115 dBm
Web: RSCP Threshold (dBm) UCI: multiwan.[..x..].rscp_threshold Opt: rscp_threshold	Specifies the minimum RSCP signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for rscp_dbm in mobile diagnostics. -115 | Disabled. Range | -46 to -115 dBm
Web: ECIO Threshold (dB) UCI: multiwan.[..x..].ecio_threshold Opt: ecio_threshold	Specifies the minimum ECIO signal strength in dB before considering if the interface fails signal health check. Uses the value stored for ecio_db in mobile diagnostics. -115 | Disabled. Range | -46 to -115 dB
Web: Signal Test UCI: multiwan.[..x..].signal_test Opt: signal_test	Defines script to test various signal characteristics in multiwan signal test. For example: option signal_test '(tech == 0) then (sig_dbm > -70) else (rscp_dbm > -105 and ecio_db > -15)' This states that when technology is GSM a health fail is determined when signal strength is less than -70dBm. When technology is not GSM a health fail occurs when either rscp_dbm falls below -105dBm or ecio_db falls below -15dB. Tech values are: 0 | GSM 1 | GSM Compact 2 | UTRAN 3 | GSM w/EGPRS 4 | UTRAN w/HSPDA 5 | UTRAN w/HSUPA 6 | UTRAN w/HSUPA and HSDPA 7 | E-UTRAN

Table 56: Information table for Multi-WAN page

Click **Save**.

21.2.2 Set options for automatically created interfaces (failover)

From the top menu on the web interface page, select **Services ->Mobile Manager**. The Mobile Manager page appears.

There are four sections in the mobile manager page:

Section	Description
Basic settings	Enable SMS, configure SIM pin code, select roaming SIM, collect ICCCIDs and set IMSI.
CDMA*	CDMA configuration
Callers	Configure callers that can use SMS.
Roaming Interface Template	Configure Preferred Roaming List options
*Option available only for Telit CE910-SL module.	

21.2.3 Mobile manager: basic settings

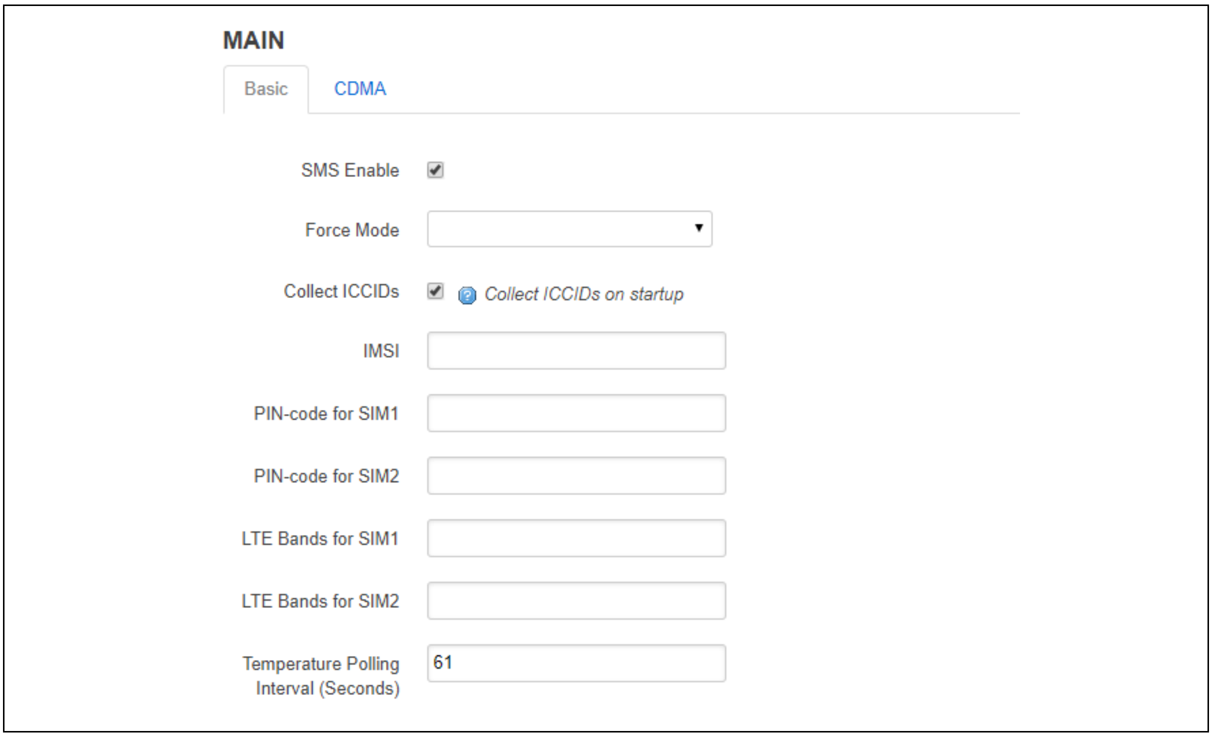


Figure 99: The mobile manager basic page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: SMS Enable
UCI: mobile.main.sms
Opt: sms</td><td>Enables or disables SMS functionality.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Force Mode
UCI: mobile.main.force_mode
Opt: force_mode</td><td>Defines whether to operate mobile modem in TTY or Ethernet mode. The mode will be dependent on the service provided by the mobile provider. In general, this is Ethernet mode (default).<table><tr><td>Blank</td><td>Ethernet mode (option not present).</td></tr><tr><td>tty</td><td>Enable TTY mode.</td></tr></table></td></tr>
<tr><td>Web: Collect ICCIDs
UCI: mobile.main.init_get_iccids
Opt: init_get_iccids</td><td>Enables or disables integrated circuit card identifier ICCID’s collection functionality. If enabled then both SIM 1 and SIM 2 ICCIDs will be collected otherwise it will default to SIM 1. This will be displayed under mobile stats.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: IMSI
UCI: mobile.main.imsi
Opt: imsi</td><td>Allows the IMSI (International Mobile Subscriber Identity) to be changed.<table><tr><td>Default</td><td>Programmed in module.</td></tr><tr><td>Digits</td><td>Up to 15 digits.</td></tr></table></td></tr>
<tr><td>Web: PIN code for SIM1
UCI: mobile.main.sim1pin
Opt: sim1pin</td><td>Depending on the SIM card specify the pin code for SIM 1.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td>Depends on the SIM provider.</td></tr></table></td></tr>
<tr><td>Web: PIN code for SIM2
UCI: mobile.main.sim2pin
Opt: sim2pin</td><td>Depending on the SIM card specify the pin code for SIM 2.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td>Depends on the SIM provider.</td></tr></table></td></tr>
<tr><td>Web: LTE bands for SIM1
UCI: mobile.main.sim1_lte_bands
Opt: sim1_lte_bands</td><td>Depending on the SIM card specify the LTE bands for SIM 1. Comma delimiter. Example:
option sim1_lte_bands ‘3,20’
Limits LTE bands to 3 and 20.
Note: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200 and Asiatel.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td>LTE bands range from 1 to 70.</td></tr></table></td></tr>
<tr><td>Web: LTE bands for SIM2
UCI: mobile.main.sim2_lte_bands
Opt:sim2_lte_bands</td><td>Depending on the SIM card specify the LTE bands for SIM 2. Comma delimiter. Example:
option sim1_lte_bands ‘3,20’
Limits LTE bands to 3 and 20.
Note: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200 and Asiatel.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td>LTE bands range from 1 to 70.</td></tr></table></td></tr>
<tr><td>Web: Temperature Polling Interval
UCI: mobile.main.temp_poll_interval_sec
Opt: temp_poll_interval_sec</td><td>Defines the time in seconds to poll the mobile module for temperature. Set to 0 to disable.<table><tr><td>61</td><td>61 seconds.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a
UCI: mobile.main.disable_time
Opt: disable_time</td><td>Defines whether to use time obtained from the mobile carrier to update the system clock when NTP is enabled.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
</table>

<table>
<tr>
<td>Web: n/a
UCI: mobile.main.service_order
Opt: service_order</td>
<td>Defines a space separated list of services, in preferred order. Valid options are gprs, umts, lte, auto.
If no valid_service order is defined, then the configured Service Type is used. Example:
mobile.main.service_order=”gprs umts lte auto”
<table>
<tr><td>Blank</td><td>Use configured service type.</td></tr>
<tr><td>Range</td><td>gprs umts lte auto.</td></tr>
</table>
</td>
</tr>
</table>

Table 57: Information table for mobile manager basic settings

21.2.4 Mobile manager: CDMA settings

This configuration page is only supported for the Telit CE910-SL CDMA module.

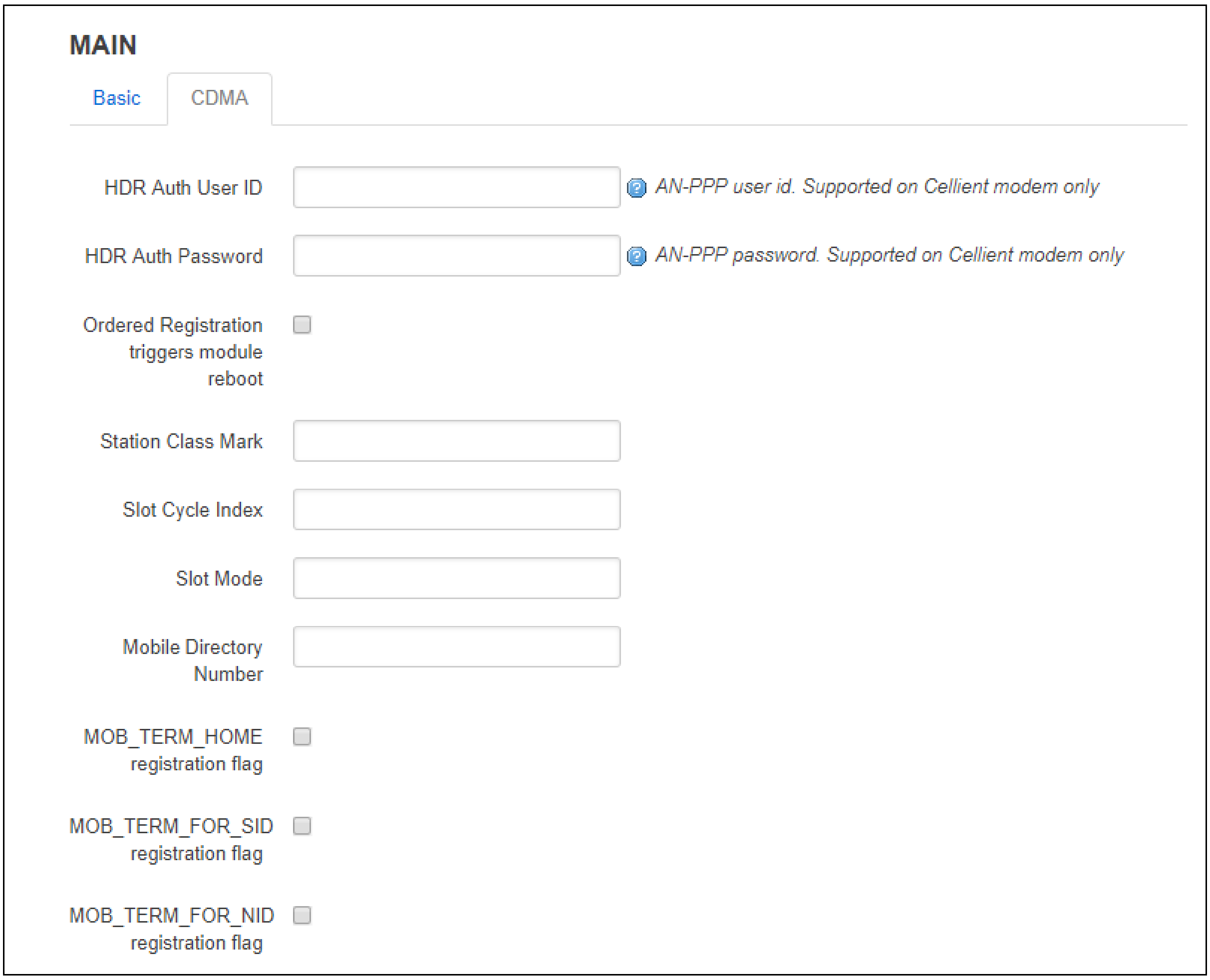


Figure 100: The mobile manager CDMA page

Web Field / UCI / Package Option	Description	
Web: HDR Auth User ID UCI: mobile.main.hdr_userid Opt: hdr_userid	AN-PPP user ID. Supported on Cellient (CDMA) modem only.	
	Blank	
	Range	Depends on the CDMA provider.
Web: HDR Auth User Password UCI: mobile.main.hdr_password Opt: hdr_password	AN-PPP password. Supported on Cellient (CDMA) modem only.	
	Blank	
	Range	Depends on the CDMA provider.
Web: Ordered Registration triggers module reboot UCI: mobile.main. mobile.main.cdma_ordered_registration_reboot_enabled Opt: cdma_ordered_registration_reboot_enabled	Enables or disables rebooting the module after Order Registration command is received from a network.	
	0	Disabled.
	1	Enabled.
Web: Station Class Mark UCI: mobile.main.cdma_station_class_mark Opt: cdma_station_class_mark	Allows the station class mark for the MS to be changed.	
	58	
	0-255	
Web: Slot Cycle Index UCI: mobile.main.cdma_slot_cycle_index Opt: cdma_slot_cycle_index	The desired slot cycle index if different from the default.	
	2	
	0-7	
Web: Slot Mode UCI: mobile.main.cdma_slot_mode Opt: cdma_slot_mode	Specifies the slot mode.	
	0	
Web: Mobile Directory Number UCI: mobile.main.cdma_mobile_directory_number Opt: cdma_mobile_directory_number	Allows the mobile directory number (MDN) to be changed	
	Default	Programmed in module.
	Digits	Up to 15 digits.
Web: MOB_TERM_HOME registration flag UCI: mobile.main. cdma_mob_term_home_registration_flag Opt: cdma_mob_term_home_registration_flag	The MOB_TERM_HOME registration flag	
	0	Disabled.
	1	Enabled.
Web: MOB_TERM_FOR_SID registration flag UCI: mobile.main. cdma_mob_term_for_sid_registration_flag Opt: cdma_mob_term_for_sid_registration_flag	The MOB_TERM_FOR_SID registration flag	
	0	Disabled.
	1	Enabled.
Web: MOB_TERM_FOR_NID registration flag UCI: mobile.main. cdma_mob_term_for_nid_registration_flag Opt: cdma_mob_term_for_nid_registration_flag	The MOB_TERM_FOR_NID registration flag	
	0	Disabled.
	1	Enabled.
Web: Access Overload Control UCI: mobile.main.cdma_access_overload_control Opt: cdma_access_overload_control	Allows the access overload class to be changed	
	Default	Programmed into module as part of IMSI
	Range	0-7
Web: Preferred Serving System UCI: mobile.main.cdma_preferred_serving_system Opt: cdma_preferred_serving_system	The CDMA Preferred Serving System(A/B)	
	5	
Web: Digital Analog Mode Preference UCI: cdma_digital_analog_mode_preference Opt: cdma_digital_analog_mode_preference	Digital/Analog Mode Preference.	
	4	

Web: Primary Channel A UCI: mobile.main.cdma_primary_channel_a Opt: cdma_primary_channel_a.	Allows the primary channel (A) to be changed. 283 1-2016 Any band class 5 channel number.
Web: Primary Channel B UCI: mobile.main.cdma_primary_channel_b Opt: cdma_primary_channel_b	Allows the primary channel (B) to be changed. 384 1-2016 Any band class 5 channel number
Web: Secondary Channel A UCI: mobile.main.cdma_secondary_channel_a Opt: cdma_secondary_channel_a	Allows the secondary channel (A) to be changed. 691 1-2016 Any band class 5 channel number.
Web: Secondary Channel B UCI: mobile.main.cdma_secondary_channel_b Opt: cdma_secondary_channel_b	Allows the secondary channel (B) to be changed. 777 1-2016 Any band class 5 channel number.
Web: Preferred Forward & Reverse RC UCI: mobile.main.cdma_preferred_forward_and_reverse_rc Opt:cdma_preferred_forward_and_reverse_rc	The Preferred Forward & Reverse RC value, this takes the form "forward_rc,reverse_rc" Format: forward radio channel, reverse radio channel Default: 0,0
Web: SID-NID pairs UCI: mobile.main.cdma_sid_nid_pairs Opt:cdma_sid_nid_pairs	Allows specification of SID:NID pairs, this takes the form "SID1,NID1,SID2,NID2, … Format: SID1 (0-65535),NID (0-65535) Default: 0,65535

Table 58: Information table for mobile manager CDMA settings

21.2.5 Mobile manager: callers

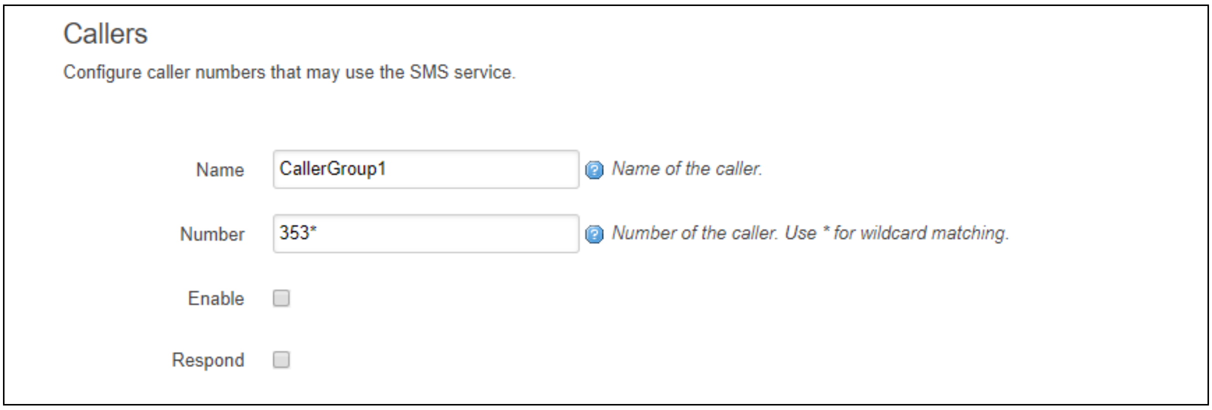


Figure 101: The mobile manager CDMA page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name
UCI: mobile.@caller[0].name
Opt:name</td><td>Name assigned to the caller.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td>No limit</td></tr></table></td></tr>
<tr><td>Web: Number
UCI: mobile.@caller[0].number
Opt:number</td><td>Number of the caller allowed to SMS the router. Add in specific caller numbers, or use the * wildcard symbol.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td>No limit</td></tr><tr><td>Characters</td><td>Global value (*) is accepted.
International value (+) is accepted.</td></tr></table></td></tr>
<tr><td>Web: Enable
UCI: mobile.@caller[0].enabled
Opt:enabled</td><td>Enables or disables incoming caller ID.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Respond
UCI: mobile.@caller[0].respond
Opt: respond</td><td>If checked, the router will return an SMS. Select Respond if you want the router to reply.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
</table>

Table 59: Information table for mobile manager callers settings

21.2.6 Roaming interface template

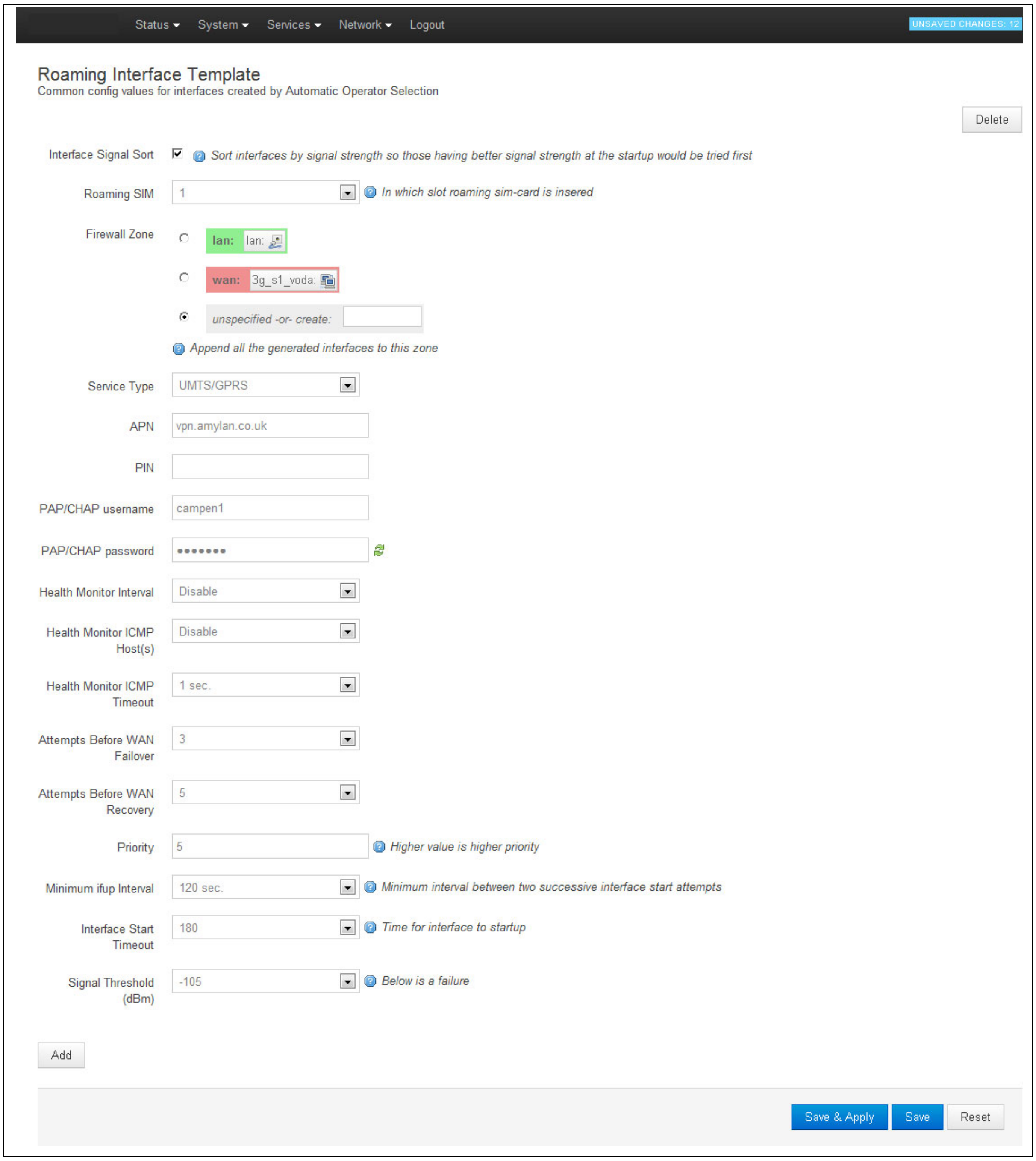


Figure 102: The roaming interface template page

Web Field / UCI / Package Option	Description
Web: Interface Signal Sort UCI: mobile.@roaming_template[0].sort_sig_strength Opt: sort_sig_strength	Sorts interfaces by signal strength priority so those that have a better signal strength will be tried first. 0: Disabled. 1: Enabled.
Web: Roaming SIM UCI: mobile.main.roaming_sim Opt: roaming_sim	Sets in which slot to insert roaming SIM card. 1: SIM slot 1. 2: SIM slot 2.
Web: Firewall Zone UCI: mobile.@roaming_template[0].firewall_zone Opt: firewall_zone	Adds all generated interfaces to this zone. Select existing zone or click **unspecified** or **create** to create new zone.
Web: Service Type UCI: mobile.@roaming_template[0].service Opt: service	Specifies the service type that will be used to connect to the network. Auto: GSM module will automatically detect the best available technology code. lte_only: Allows GSM module to only connect to LTE network. umts_only: Allows GSM module to only connect to 3G network. gprs_only: Allows GSM module to only connect to GPRS network. cdma: Allows GSM module to only connect to cdma network.
Web: APN UCI: mobile.@roaming_template[0].apn Opt: apn	APN name of Mobile Network Operator.
Web: PIN UCI: mobile.@roaming_template[0].pincode Opt: pincode	SIM card's PIN number.
Web: PAP/CHAP username UCI: mobile.@roaming_template[0].username Opt: username	Username used to connect to APN.
Web: PAP/CHAP password UCI: mobile.@roaming_template[0].password Opt: password	Password used to connect to APN.
Web: Health Monitor Interval UCI: mobile.@roaming_template[0].health_interval Opt: health_interval	Sets the period to check the health status of the interface. The Health Monitor interval will be used for: • Interface state checks • Ping interval • Signal strength checks
Web: Health Monitor ICMP Host(s) UCI: mobile.@roaming_template[0].icmp_hosts Opt: icmp_hosts	Specifies target IP address for ICMP packets. Disable: Disables the option. DNS servers: DNS IP addresses will be used. WAN gateway: Gateway IP address will be used. custom: Ability to provide IP address.
Web: Health Monitor ICMP Timeout UCI: mobile.@roaming_template[0].timeout Opt: timeout	Specifies the time in seconds that Health Monitor ICMP will timeout at. Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at. 3: Wait 3 seconds for ping reply. Range

Web: Attempts Before WAN Failover UCI: mobile.@roaming_template[1].health_fail_retries Opt: health_fail_retries	Defines the number of health check failures before interface is disconnected. 3 | Range |
Web: Attempts Before WAN Recovery UCI: mobile.@roaming_template[0].health_recovery_retries Opt: health_recovery_retries	Sets the number of health check passes before the interface is considered healthy. This field is not used for a roaming template. 5 | Range |
Web: Priority UCI: mobile.@roaming_template[0].priority Opt: priority	Type the priority number. The higher the value, the higher the priority. This multi-WAN interface priority must be lower than the one specified in the priority field for the PMP interface. 0 | Range |
Web: Minimum ifup interval UCI: multiwan.wan.ifup_retry_sec Opt: ifup_retry_sec	Not used for a roaming interface. 300 | Retry primary interface every 300 seconds Range |
Web: Interface Start Timeout UCI: mobile.@roaming_template[0].ifup_timeout_sec Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail. 40 | 40 seconds Range |
Web: Signal Threshold (dBm) UCI: mobile.@roaming_template[0].signal_threshold Opt: signal_threshold	Specifies the minimum RSCP signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for rscp_dbm in mobile diagnostics. Range | -46 to -115 dBm -115dBm |
Web: n/a UCI: mobile.main.service_order Opt: service_order	Defines a space separated list of services, in preferred order. Valid options are `gprs, umts, lte, auto`. If no valid_service order is defined, then the configured Service Type is used. Example: `mobile.@roaming_template[0].service_order="gprs umts lte auto"` Blank | Use configured service type. Range | gprs umts lte auto

Table 60: Information table for roaming interface template

When you have configured your settings, click **Save & Apply**.

In the top menu, select **System -> Reboot**. The System page appears.

Figure 103: The reboot page

Check the **Reboot now** check box and then click **Reboot**.

21.2.7 Scenario 2: PMP + roaming: pre-empt disabled

As in the previous section, multi-WAN connects the PMP interface and uses auto created interfaces for failover.

However, in this scenario, the auto-created interface will not be disconnected as soon as the `ifup_retry_sec` expires for the PMP interface. The primary interface will be reconnected when the current auto-created interface fails multiwan health checks after expiration of the `ifup_retry_sec` timer.

Follow the instructions in the section above for creation of the PMP interface, multi-WAN and Mobile Manager roaming interfaces. The only change in configuration compared to the PMP + roaming: pre-empt enabled scenario is that you must disable the pre-empt option in the multi-WAN package.

21.2.7.1 Set multi-WAN options for pre-empt disabled

To disable PMP + roaming pre-empt, in the top menu, select **Network -> Multi-Wan**.

In the Multi-WAN page, ensure Preempt is not selected.

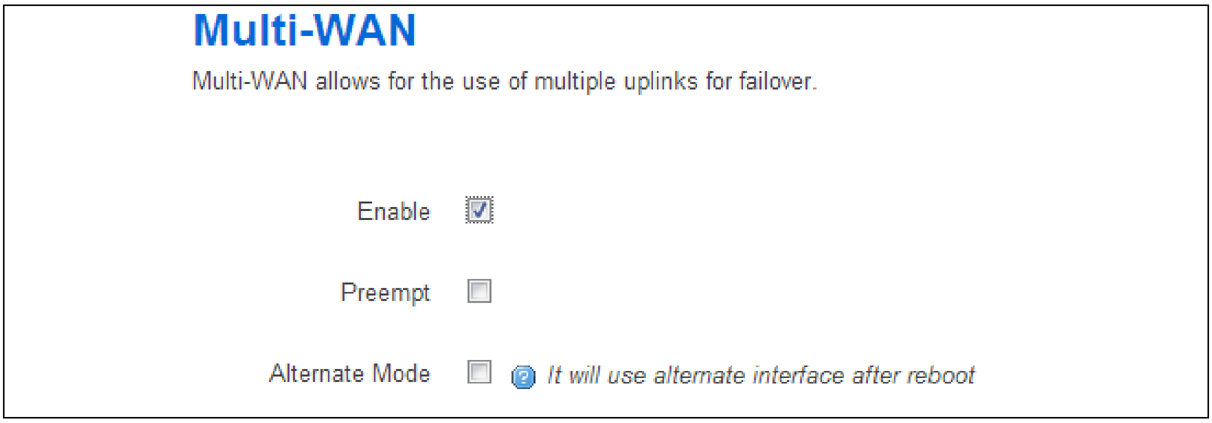


Figure 104: The multi-wan page, pre-empt not selected

Click **Save & Apply**.

In the top menu, select **System -> Reboot**. The System Reboot page appears.

Figure 105: The system reboot page

Check the **Reboot now** check box and then click **Reboot**.

21.2.8 Scenario 3: No PMP + roaming

In this scenario there is no PMP interface that can be used for a connection. The router scans the available mobile networks at boot and sorts the networks according to signal strength.

The network that offers the best signal strength will be the first to connect. Multi-WAN then controls the failover between the available networks.

Multiwan periodically does a health check on the interface. A health check comprises of a configurable combination of the following:

- Interface state
- Pings to an ICMP target
- Signal level checks using signal threshold, RSCP threshold and ECIO threshold option values

A fail for any of the above health checks results in a fail. After a configurable number of health check failures, Multi-WAN will disconnect the failed interface and attempt to connect to the next best roaming interface.

21.2.9 Set options for automatically created interfaces (failover)

In the top menu on the web interface page, select **Services ->Mobile Manager**. The Mobile Manager page appears.

There are three sections:

Basic settings	Configure SMS, select roaming SIM and collect ICCCIDs
Callers	Configure callers that can use SMS.
Roaming Interface Template	Configure common values for interface created by Automatic Operator Selection.

21.2.9.1 Basic settings

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: SMS Enable
UCI: mobile.main.sms
Opt: sms</td><td>Enables SMS.<table><tr><td>no</td><td>Disabled.</td></tr><tr><td>yes</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Collect ICCIDs
UCI: mobile.main.init_get_iccids
Opt: init_get_iccids</td><td>Enables or disables integrated circuit card identifier ICCID's collection functionality. If enabled then both SIM 1 and SIM 2 ICCID's will be collected otherwise it will default to SIM 1. This will be display under mobile stats.<table><tr><td>no</td><td>Disabled.</td></tr><tr><td>yes</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: PIN code for SIM1
UCI: mobile.main.sim2pin
Opt: sim2pin</td><td>Depending on the SIM card specify the pin code for SIM 1.<table><tr><td>Blank</td><td></td></tr><tr><td>range</td><td></td></tr></table></td></tr>
<tr><td>Web: PIN code for SIM2
UCI: mobile.main.sim2pin
Opt: sim2pin</td><td>Depending on the SIM card specify the pin code for SIM 2.<table><tr><td>Blank</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web: HDR Auto User ID UCI: mobile.main.hdr_userid Opt: hdr_userid	AN-PPP user ID. Supported on Cellient (CDMA) modem only. Blank: Range:

Table 61: Information table for mobile manager basic settings

21.2.9.2 Caller settings

Web Field / UCI / Package Option	Description
Web: Name UCI: mobile.@caller[0].name Opt: name	Name assigned to the caller. Blank: Range:
Web: Number UCI: mobile.@caller[0].number Opt: number	Number of the caller allowed to SMS the router. Add in specific caller numbers, or use the wildcard symbol. Blank: Range:
Web: Enable UCI: mobile.@caller[0].enabled Opt: enabled	Enables or disables incoming caller ID. no: Disabled. yes: Enabled.
Web: Respond UCI: mobile.@caller[0].respond Opt: respond	If checked, the router will return an SMS. Select Respond if you want the router to reply. 0: Disabled. 1: Enabled.

Table 62: Information table for mobile manager caller settings

21.2.10 Roaming interface template

Figure 106: The roaming interface template page

Web Field / UCI / Package Option	Description
Web: Interface Signal Sort UCI: mobile.@roaming_template[0].sort_sig_strength Opt: sort_sig_strength	Sorts interfaces by signal strength priority so those that have a better signal strength will be tried first.
Web: Roaming SIM UCI: mobile.main.roaming_sim Opt: roaming_sim	Sets which slot to insert roaming SIM card. 1 — SIM slot 1. 2 — SIM slot 2.
Web: Firewall Zone UCI: mobile.@roaming_template[0].firewall_zone Opt: firewall_zone	Adds all generated interfaces to this zone. Select existing zone or click unspecified or create to create a new zone.

<table>
<tr><td>Web: Service Type
UCI: mobile.@roaming_template[0].service
Opt: service</td><td>Specifies the service type that will be used to connect to the network.<table><tr><td>UMTS/GPRS</td><td>GSM module will automatically detect the best available technology code.</td></tr><tr><td>Umts_only</td><td>Allows GSM module to only connect to 3G network.</td></tr><tr><td>GPRS_only</td><td>Allows GSM module to only connect to GPRS network.</td></tr><tr><td>cdma</td><td>Allows GSM module to only connect to cdma network.</td></tr></table></td></tr>
<tr><td>Web: APN
UCI: mobile.@roaming_template[0].apn
Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: PIN
UCI: mobile.@roaming_template[0].pincode
Opt: pincode</td><td>SIM card's PIN number.</td></tr>
<tr><td>Web: PAP/CHAP username
UCI: mobile.@roaming_template[0].username
Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: PAP/CHAP password
UCI: mobile.@roaming_template[0].password
Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: Health Monitor Interval
UCI: mobile.@roaming_template[0].health_interval
Opt: health_interval</td><td>Sets the period to check the health status of the interface. The Health Monitor interval will be used for:
Interface state checks
Ping interval
Signal strength checks</td></tr>
<tr><td>Web: Health Monitor ICMP Host(s)
UCI: mobile.@roaming_template[0].icmp_hosts
Opt: icmp_hosts</td><td>Specifies target IP address for ICMP packets.<table><tr><td>Disable</td><td>Disables the option</td></tr><tr><td>DNS servers</td><td>DNS IP addresses will be used.</td></tr><tr><td>WAN gateway</td><td>Gateway IP address will be used.</td></tr><tr><td>custom</td><td>Ability to provide IP address.</td></tr></table></td></tr>
<tr><td>Web: Health Monitor ICMP Timeout
UCI: mobile.@roaming_template[0].timeout
Opt: timeout</td><td>Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.<table><tr><td>3</td><td>Wait 3 seconds for ping reply</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Attempts Before WAN Failover
UCI: mobile.@roaming_template[1].health_fail_retries
Opt: health_fail_retries</td><td>Defines the number of health check failures before interface is disconnected.<table><tr><td>3</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Attempts Before WAN Recovery
UCI: mobile.@roaming_template[0].health_recovery_retries
Opt: health_recovery_retries</td><td>Sets the number of health check passes before the interface is considered healthy. This field is not used for a roaming template.</td></tr>
<tr><td>Web: Priority
UCI: mobile.@roaming_template[0].priority
Opt: priority</td><td>Type the priority number. The higher the value, the higher the priority.<table><tr><td>0</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web: Minimum ifup interval UCI: mobile.@roaming_template[0].ifup_retry_sec Opt: ifup_retry_sec	Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled. 300 – Retry primary interface every 300 seconds. Range
Web: Interface Start Timeout UCI: mobile.@roaming_template[0].ifup_timeout_sec Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail It is recommended to configure a value greater than 120 seconds. 40 Range
Web: Signal Threshold (dBm) UCI: mobile.@roaming_template[0].signal_threshold Opt: signal_threshold	Specifies the minimum signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for sig_dbm in mobile diagnostics.-115 dBm. Disabled Range – -46 to -115 dBm

Table 63: Information table for roaming interface template

When you have configured your settings, click **Save & Apply**.

21.2.10.1 Set multi-WAN operation

From the top menu, select **Network -> Multi-Wan**. The Multi-WAN page appears.

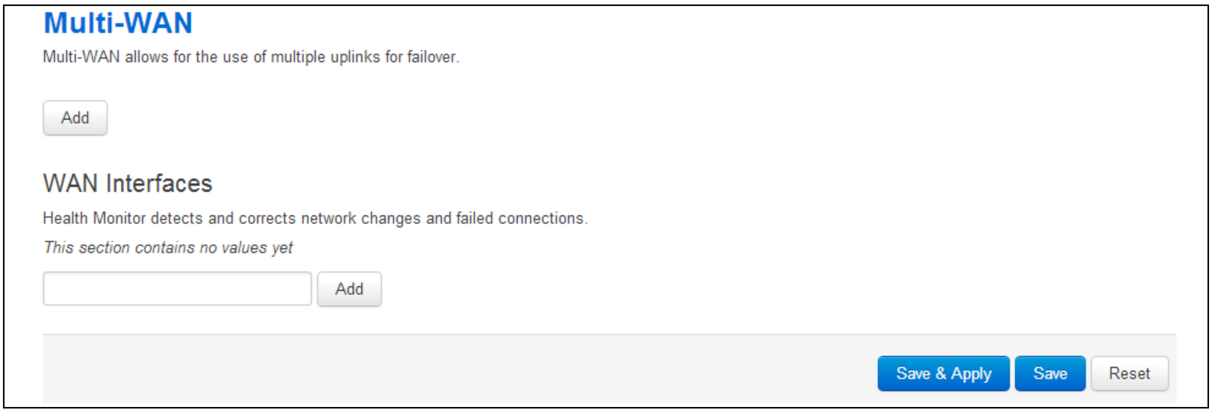


Figure 107: The multi-WAN page

In the Multi-WAN section click **Add**.

Web Field / UCI / Package Option	Description
Web: Enable UCI: multiwan.config.enabled Opt: enabled	Enables multiwan. Select this option. 0 – Disabled. 1 – Enabled.
Web: Preempt UCI: multiwan.config.preempt Opt: pre-empt	Enables or disables pre-emption for multiwan. If enabled the router will keep trying to connect to a higher priority interface depending on timer set by ifup_retry_sec. Leave this option unselected. 0 – Disabled. 1 – Enabled.
Web: Alternate Mode UCI: multiwan.config.alt Opt: alt	Enables or disables alternate mode for multiwan. If enabled the router will use an alternate interface after reboot. Leave this option unselected. 0 – Disabled. 1 – Enabled.

Table 64: Information table for multi-WAN operation

21.3 Configuring via UCI

21.3.1 PMP + roaming: pre-empt enabled & disabled via UCI

21.3.1.1 PMP interface configuration

The PMP interface is configured in the network package /etc/config/network. To view the network configuration file, enter:

```
root@VA_router:~# uci export network
package network

config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config interface 'lan'
        option ifname 'eth0'
        option proto 'static'
        option ipaddr '192.168.100.1'
        option netmask '255.255.255.0'

config interface '3g_s1_voda'
        option auto '0'
        option proto '3g'
        option service 'umts'
        option apn 'testIE'
        option username 'test'
        option password 'test'
        option sim '1'        option operator 'vodafone IE'
```

To view uci commands, enter:

```
root@VA_router:~# uci show network
network.loopback=interface
network.loopback.ifname=lo
network.loopback.proto=static
network.loopback.ipaddr=127.0.0.1
network.loopback.netmask=255.0.0.0
```

```
network.lan=interface
network.lan.ifname=eth0
network.lan.proto=static
network.lan.ipaddr=192.168.100.1
network.lan.netmask=255.255.255.0
network.3g_s1_voda=interface
network. 3g_s1_voda.auto=0
network. 3g_s1_voda.proto=3g
network. 3g_s1_voda.service=umts
network. 3g_s1_voda.apn=test IE
network. 3g_s1_voda.username=test
network. 3g_s1_voda.password=test
network. 3g_s1_voda.sim=1
network. 3g_s1_voda.operator=vodafone IE
```

21.3.1.2 Roaming interface configuration

The roaming interface configurations are stored in the mobile package /etc/config/mobile.

To view the mobile configuration file, enter: `root@VA_router:~# uci export mobile`

```
config mobile 'main'
        option sms 'yes'
        option roaming_sim '1'
        option init_get_iccids 'no'
config caller
        option name 'Test'
        option number '*'
        option enabled 'yes'
        option respond 'yes'
config roaming_template
        option roaming_sim '1'
        option firewall_zone 'wan'
        option apn 'test IE'
        option username 'test'
        option password 'test'
        option service 'umts'
        option health_interval '4'
        option icmp_hosts 'disable'
```

```
        option timeout 'disable'
        option health_fail_retries '3'
        option signal_threshold '-95'
        option priority '5'
        option ifup_retry_sec '120'
        option ifup_timeout_sec '180'
        option defaultroute 'yes'
        option sort_sig_strength 'yes'
```

To view the uci command of package mobile, enter:

```
root@VA_router:~#uci show mobile
mobile.main=mobile
mobile.main.sms=yes
mobile.main.roaming_sim=1
mobile.main.init_get_iccids=no
mobile.@caller[0]=caller
mobile.@caller[0].name=Test
mobile.@caller[0].number=*
mobile.@caller[0].enabled=yes
mobile.@caller[0].respond=yes
mobile.@roaming_template[0]=roaming_template
mobile.@roaming_template[0].roaming_sim=1
mobile.@roaming_template[0].firewall_zone=wan
mobile.@roaming_template[0].apn=test IE
mobile.@roaming_template[0].username=test
mobile.@roaming_template[0].password=test
mobile.@roaming_template[0].service=umts
mobile.@roaming_template[0].health_interval=4
mobile.@roaming_template[0].icmp_hosts=disable
mobile.@roaming_template[0].timeout=disable
mobile.@roaming_template[0].health_fail_retries=3
mobile.@roaming_template[0].signal_threshold=-95
mobile.@roaming_template[0].priority=5
mobile.@roaming_template[0].ifup_retry_sec=120
mobile.@roaming_template[0].ifup_timeout_sec=180
mobile.@roaming_template[0].defaultroute=yes
mobile.@roaming_template[0].sort_sig_strength=yes
```

21.3.1.3 Multi-WAN configuration using UCI

The configuration file for package multiwan is stored on **/etc/config/multiwan**

To see configuration file of mobile package, enter:

```
root@VA_router:~# cat /etc/config/multiwan
config multiwan 'config'
        option enabled '1'
        option preempt '1'

config interface '3g_s1_voda'
        option health_fail_retries '3'
        option health_interval '3'
        option timeout '1'
        option icmp_hosts 'disable'
        option priority '10'
        option exclusive_group '3g'
        option signal_threshold '-95'
        option ifup_retry_sec '350'
        option ifup_timeout_sec '180'
        option manage_state '1'
```

To view the uci command of package multiwan, enter:

```
root@VA_router:~# uci show multiwan
multiwan.config=multiwan
multiwan.config.enabled=1
multiwan.config.preempt=1
multiwan.main_voda=interface
multiwan.main_voda.health_fail_retries=3
multiwan.main_voda.health_interval=3
multiwan.3g_s1_voda.timeout=1
multiwan.3g_s1_voda.icmp_hosts=disable
multiwan.3g_s1 main _voda.priority=10
multiwan.3g_s1_voda.exclusive_group=3g
multiwan.3g_s1_voda.signal_threshold=-95
multiwan.3g_s1_voda.ifup_retry_sec=350
multiwan.3g_s1_voda.ifup_timeout_sec=180
multiwan.3g_s1_voda.manage_state=1
```

The difference between PMP + roaming: pre-empt enabled and disabled is setting one option parameter. To disable pre-empt, enter:

```
uci set multiwan.config.preempt=0
uci commit
```

Note: available values are:

0	Disabled
1	Enabled

21.4 Configuring no PMP + roaming using UCI

The roaming interface configuration file is stored in the mobile package **/etc/config/mobile**. To view the mobile package, enter:

```
root@VA_router:~# uci export mobile

package mobile
config mobile 'main'
        option sms 'yes'
        option roaming_sim '1'
        option debug '1'

config caller
        option name 'Eval'
        option number '*'
        option enabled 'yes'
        option respond 'yes'

config roaming_template
        option roaming_sim '1'
        option firewall_zone 'wan'
        option apn 'test IE'
        option username 'test'
        option password 'test'
        option service 'umts'
        option health_fail_retries '2'
        option signal_threshold '-100'
        option priority '5'
        option ifup_timeout_sec '180'
        option defaultroute 'yes'
```

```
        option sort_sig_strength 'yes'
        option ifup_retry_sec '200'
        option health_interval '120'
        option icmp_hosts '172.31.4.129'
        option timeout '3'
        option health_recovery_retries '3'
```

To view the mobile package via uci commands, enter:

```
root@VA_router:~# uci show mobile
mobile.main=mobile
mobile.main.sms=yes
mobile.main.roaming_sim=1
mobile.main.debug=1
mobile.@caller[0]=caller
mobile.@caller[0].name=Eval
mobile.@caller[0].number=*
mobile.@caller[0].enabled=yes
mobile.@caller[0].respond=yes
mobile.@roaming_template[0]=roaming_template
mobile.@roaming_template[0].roaming_sim=1
mobile.@roaming_template[0].firewall_zone=wan
mobile.@roaming_template[0].apn=stream.co.uk
mobile.@roaming_template[0].username=default
mobile.@roaming_template[0].password=void
mobile.@roaming_template[0].service=umts
mobile.@roaming_template[0].health_fail_retries=2
mobile.@roaming_template[0].signal_threshold=-100
mobile.@roaming_template[0].priority=5
mobile.@roaming_template[0].ifup_timeout_sec=180
mobile.@roaming_template[0].defaultroute=yes
mobile.@roaming_template[0].sort_sig_strength=yes
mobile.@roaming_template[0].ifup_retry_sec=200
mobile.@roaming_template[0].health_interval=120
mobile.@roaming_template[0].icmp_hosts=172.31.4.129
mobile.@roaming_template[0].timeout=3
mobile.@roaming_template[0].health_recovery_retries=3
```

The multiwan package is stored on **/etc/config/multiwan**. To view the multiwan package, enter:

```
root@VA_router:~# uci export multiwan
package multiwan

config multiwan 'config'
        option enabled 'yes'
        option preempt 'no'
        option alt_mode 'no'
To see multiwan package via uci, enter:
root@VA_router:~# uci show multiwan
multiwan.config=multiwan
multiwan.config.enabled=yes
multiwan.config.preempt=no
multiwan.config.alt_mode=no
```

21.5 Automatic operator selection diagnostics via the web interface

21.5.1 Checking the status of the Multi-WAN package

When interfaces are auto-created they are presented in the network and in the Multi-WAN package.

To check interfaces created in the Multi-WAN package, from the top menu, select **Network -> Multi-WAN**.

To check interfaces that have been created in the network package, from the top menu, select **Network -> Interfaces**.

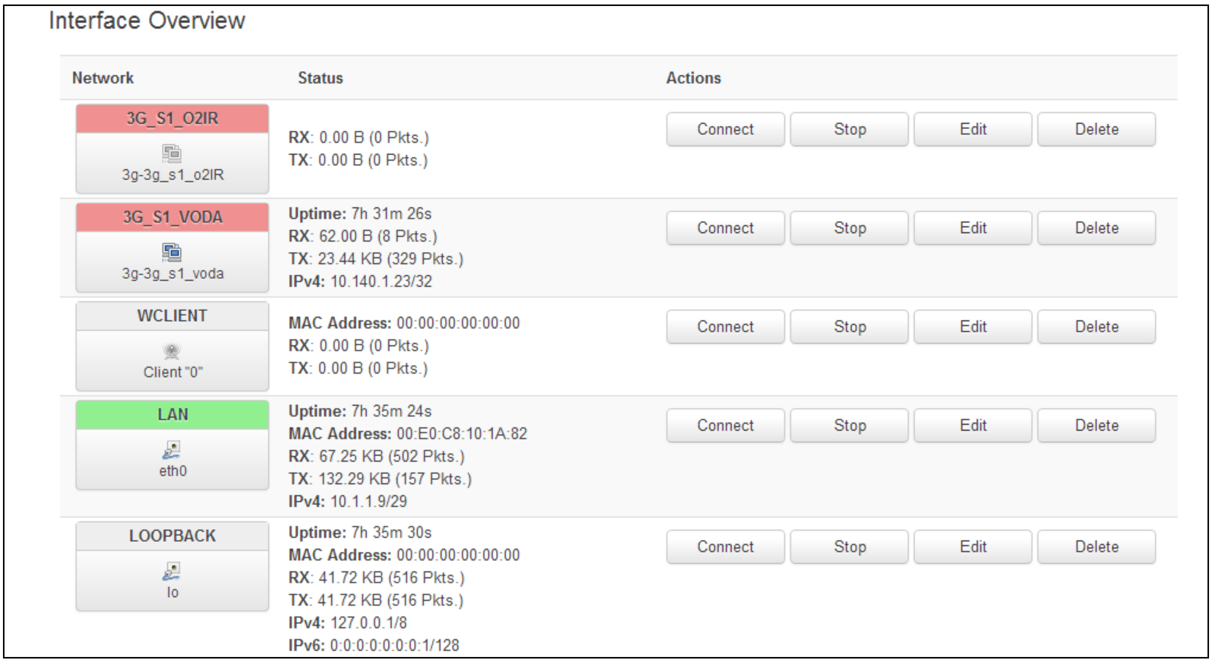


Figure 108: The interface overview page

To check the status of the interface you are currently using, in the top menu, click **Status**. The Interface Status page appears.

Scroll down to the bottom of the page to view Multi-WAN Stats.

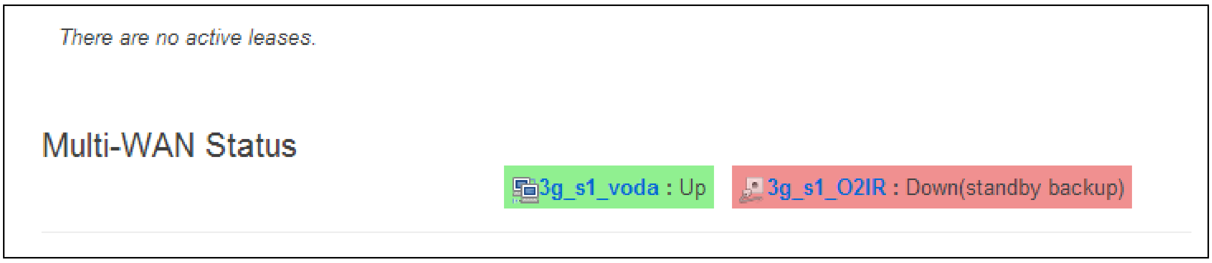
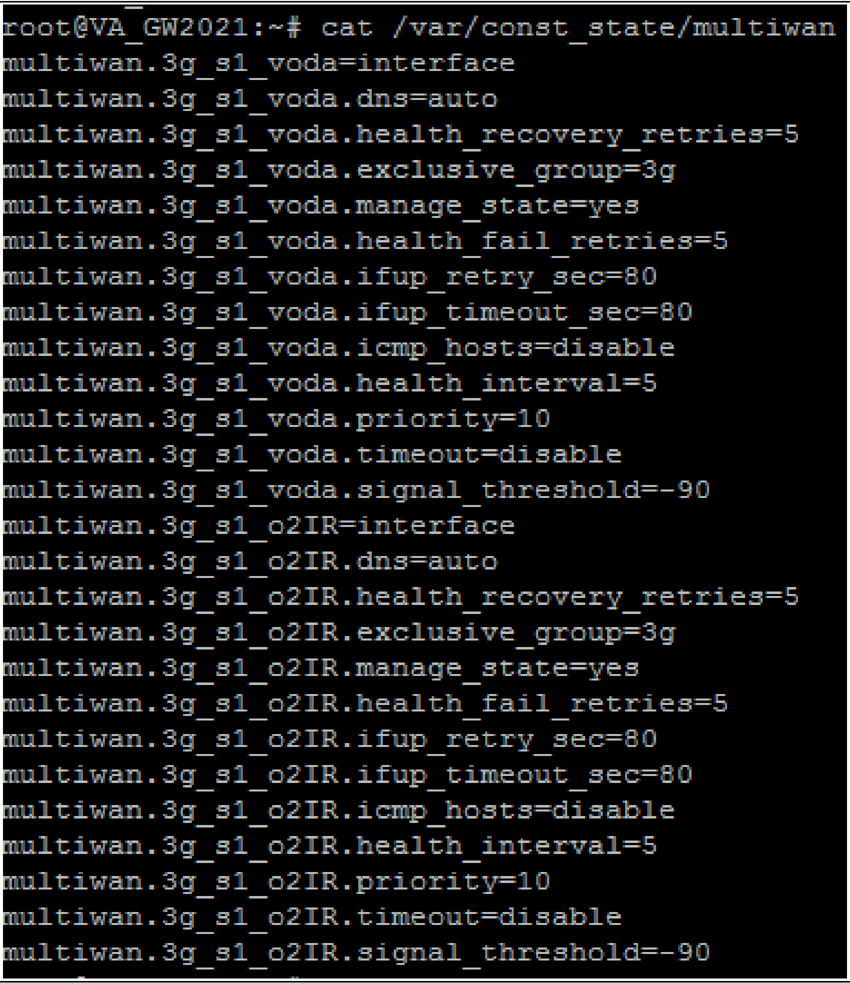


Figure 109: The status page: multi-WAN status section page

21.6 Automatic operator selection diagnostics via UCI

To check interfaces created in the multi-WAN package, enter:

```
root@VA_router:~# cat /var/const_state/multiwan
```



```
root@VA_GW2021:~# cat /var/const_state/multiwan
multiwan.3g_s1_voda=interface
multiwan.3g_s1_voda.dns=auto
multiwan.3g_s1_voda.health_recovery_retries=5
multiwan.3g_s1_voda.exclusive_group=3g
multiwan.3g_s1_voda.manage_state=yes
multiwan.3g_s1_voda.health_fail_retries=5
multiwan.3g_s1_voda.ifup_retry_sec=80
multiwan.3g_s1_voda.ifup_timeout_sec=80
multiwan.3g_s1_voda.icmp_hosts=disable
multiwan.3g_s1_voda.health_interval=5
multiwan.3g_s1_voda.priority=10
multiwan.3g_s1_voda.timeout=disable
multiwan.3g_s1_voda.signal_threshold=-90
multiwan.3g_s1_o2IR=interface
multiwan.3g_s1_o2IR.dns=auto
multiwan.3g_s1_o2IR.health_recovery_retries=5
multiwan.3g_s1_o2IR.exclusive_group=3g
multiwan.3g_s1_o2IR.manage_state=yes
multiwan.3g_s1_o2IR.health_fail_retries=5
multiwan.3g_s1_o2IR.ifup_retry_sec=80
multiwan.3g_s1_o2IR.ifup_timeout_sec=80
multiwan.3g_s1_o2IR.icmp_hosts=disable
multiwan.3g_s1_o2IR.health_interval=5
multiwan.3g_s1_o2IR.priority=10
multiwan.3g_s1_o2IR.timeout=disable
multiwan.3g_s1_o2IR.signal_threshold=-90
```

Figure 110: Example of output from the command: cat /var/const_stat/multiwan

To check interfaces created in the network package, enter:

```
root@VA_router:~# cat /var/const_state/network
```

```
root@VA_GW2021:~# cat /var/const_state/network
network.3g_s1_voda=interface
network.3g_s1_voda.auto=no
network.3g_s1_voda.service=umts
network.3g_s1_voda.roaming_sim=1
network.3g_s1_voda.defaultroute=no
network.3g_s1_voda.username=internet
network.3g_s1_voda.apn=hs.vodafone.ie
network.3g_s1_voda.operator=vodafone IE
network.3g_s1_voda.proto=3g
network.3g_s1_voda.sim=1
network.3g_s1_voda.password=internet
network.3g_s1_o2IR=interface
network.3g_s1_o2IR.auto=no
network.3g_s1_o2IR.service=umts
network.3g_s1_o2IR.roaming_sim=1
network.3g_s1_o2IR.defaultroute=no
network.3g_s1_o2IR.username=internet
network.3g_s1_o2IR.apn=hs.vodafone.ie
network.3g_s1_o2IR.operator=o2 IRL
network.3g_s1_o2IR.proto=3g
network.3g_s1_o2IR.sim=1
network.3g_s1_o2IR.password=internet
root@VA_GW2021:~#
```

Figure 111: Example of output from the command cat /var/const_state/network

To check the status of the interface you are currently using, enter:

```
root@VA_router:~# cat /var/const_state_/mobile
```

```
root@VA_GW2021:~# cat /var/const_state/mobile
mobile.3g_0=status
mobile.3g_0.sim1_iccid=89314404000039480265
root@VA_GW2021:~#
root@VA_GW2021:~#
root@VA_GW2021:~# cat /var/state/mobile
mobile.3g_0=status
mobile.3g_0.sim_slot=1
mobile.3g_0.sim_in=yes
mobile.3g_0.registered=5, Roaming
mobile.3g_0.reg_code=5
mobile.3g_0.imei=357784040034322
mobile.3g_0.imsi=204043726270034
mobile.3g_0.registered_pkt=5, Roaming
mobile.3g_0.reg_code_pkt=5
mobile.3g_0.area=BCC
mobile.3g_0.tech=2
mobile.3g_0.technology=UTRAN
mobile.3g_0.operator=1,0,"vodafone IE",2
mobile.3g_0.cell=AA787
mobile.3g_0.sig_dbm=-113
root@VA_GW2021:~#
```

Figure 112: Example of output from the command cat /vat/const_state_/mobile

22 Configuring Connection Watch (cwatch)

Connection Watch is a recovery feature to enable dynamic recovery of an interface. You can configure multiple instances of Connection Watch.

Connection Watch consists of the following configurable instances:

- Interface(s) to be monitored
- Failure periods
- Recovery actions

If no data is received over the monitored interface during the configured duration, then the recovery action is performed. If more than one interface is specified under a single Connection Watch, the recovery action will be performed only if no data is received on **both** of the interfaces for the defined period.

Currently three configurable periods and associated recovery actions can be defined.

22.1 Configuration package used

Package	Sections
cwatch	watch

22.2 Configuring Connection Watch using the web interface

To configure Connection Watch using the web interface, select **Services->Connection Watch**. The Connection Watch page appears.

If no Connection Watch configuration exists in the configuration file, first enter a name for the Connection Watch instance and select **Add**.

Figure 113: The add connection watch configuration page

Figure 114: The connection watch configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: cwatch.@watch[0].enabled
Opt: enabled</td><td>Enables a cwatch instance.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Interfaces
UCI: cwatch.@watch[0].test_ifaces
Opt: test_ifaces</td><td>Defines the interface name(s) to monitor. Multiple interfaces are delimited by space separator. Example:
<code>option test_ifaces ‘WANADSL WANMOBILE`</code>
If multiple interfaces are defined the failure action will only be triggered if no traffic is received on all interfaces for the defined period.</td></tr>
<tr><td>Web: Failure Time for Action 1
UCI: cwatch.@watch[0].failure_time_1
Opt: failure_time_1</td><td>Defines a duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days.<table><tr><td>1h</td><td></td></tr><tr><td>Range</td><td>s; m; h; d;</td></tr></table></td></tr>
<tr><td>Web: Failure Action 1
UCI: cwatch.@watch[0].failure_action_1
Opt: failure_action_1</td><td>Defines the failure action associated with failure_time_1. Example to force up interface:
<code>option failure_action_1 ‘ifup wan’</code><table><tr><td>blank</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web: Failure Time for Action 2 UCI: cwatch.@watch[0].failure_time_2 Opt: failure_time_2	Defines a second duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days. 10h | Range | s; m; h; d;
Web: Failure Action 2 UCI: cwatch.@watch[0].failure_action_2 Opt: failure_action_2	Defines the failure action associated with failure_time_2. Example to reset usb: `option failure_action_1 '/etc/init.d/usb_startup restart'` blank | Range |
Web: Failure Time for Action 3 UCI: cwatch.@watch[0].failure_time_3 Opt: failure_time_3	Defines a third duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days. 24h | Range | s; m; h; d;
Web: Failure Action 3 UCI: cwatch.@watch[0].failure_action_3 Opt: failure_action_3	Defines the failure action associated with failure_time_3. Example to reset usb: `option failure_action_3 'reboot'` blank | Range |

Table 65: Information table for cwatch section

22.3 Configuring cwatch using command line

By default, all cwatch instances are named 'watch', the cwatch instance is identified by @watch then the watch position in the package as a number. For example, for the first route in the package using UCI:

```
cwatch.@watch[0]=watch
cwatch.@watch[0].enabled=1
```

Or using package options:

```
config watch
        option enabled '1'
```

However, to better identify it, we recommend giving the cwatch instance a name. For example, a watch named 'WATCH_MOBILE' will be cwatch.WATCH_MOBILE.

To define a named cwatch instance using UCI, enter:

```
cwatch.WATCH_MOBILE=watch
cwatch.WATCH_MOBILE.enabled=1
```

To define a named cwatch instance using package options, enter:

```
config watch 'WATCH_MOBILE'
        option 'enabled' '1'
```

22.3.1 cwatch using UCI

```
root@VA_router:~# uci show cwatch
cwatch.WATCH_MOBILE=watch
cwatch.WATCH_MOBILE.enabled=1
cwatch.WATCH_MOBILE.test_ifaces=wan
cwatch.WATCH_MOBILE.failure_time_1=1h
cwatch.WATCH_MOBILE.failure_action_1=ifup wan
cwatch.WATCH_MOBILE.failure_time_2=10h
cwatch.WATCH_MOBILE.failure_action_2=/etc/init.d/usb_startup restart
cwatch.WATCH_MOBILE.failure_time_3=24h
cwatch.WATCH_MOBILE.failure_action_3=reboot
```

22.3.2 cwatch using package options

```
root@VA_router:~# uci export cwatch
package cwatch

config watch 'WATCH_MOBILE'
        option enabled '1'
        option test_ifaces wan
        option failure_time_1 '1h'
        option failure_action_1 'ifup wan
        option failure_time_2 '10h'
        option failure_action_2 '/etc/init.d/usb_startup restart'
        option failure_time_3 '24h'
        option failure_action_3 'reboot'
```

22.4 cwatch diagnostics

22.4.1 Syslog

A syslog message will be generated when cwatch starts:

```
cwatch[x]: cwatch configuration OK. Entering main loop...
```

Syslog messages will be generated when the failure action is triggered:

```
cwatch[x]: Watch WATCH_MOBILE executed action 1
cwatch[x]: Watch WATCH_MOBILE executed action 2
cwatch[x]: Watch WATCH_MOBILE executed action 3
```

23 Configuring DHCP server and DNS (Dnsmasq)

Dynamic Host Configuration Protocol (DHCP) server is responsible for assigning IP addresses to hosts. IP addresses can be given out on different interfaces and different subnets. You can manually configure lease time as well as setting static IP to host mappings.

Domain Name Server (DNS) is responsible for resolution of IP addresses to domain names on the internet.

Dnsmasq is the application which controls DHCP and DNS services. Dnsmasq has two sections; one to specify general DHCP and DNS settings and one or more DHCP pools to define DHCP operation on the desired network interface.

23.1 Configuration package used

Package	**Sections**
dhcp	dnsmasq
	dhcp
	host

23.2 Configuring DHCP and DNS using the web interface

In the top menu, select **Network -> DHCP and DNS**. The DHCP and DNS page appears. There are three sections: Server Settings, Active Leases, and Static Leases.

Figure 115: The DHCP and DNS page

23.2.1 Dnsmasq: general settings

Web Field/UCI/Package Option	Description
Web: Domain required UCI: dhcp.@dnsmasq[0].domainneeded Opt: domainneeded	Defines whether to forward DNS requests without a DNS name. Dnsmasq will never forward queries for plain names, without dots or domain parts, to upstream nameservers. If the name is not known from /etc/hosts or DHCP then a "not found" answer is returned. 1 — Enabled. 0 — Disabled.
Web: Authoritative UCI: dhcp.@dnsmasq[0]. authoritative Opt: authoritative	Forces authoritative mode, this speeds up DHCP leasing. Used if this is the only server in the network. 1 — Enabled. 0 — Disabled.
Web: Interfaces UCI: dhcp.@dnsmasq[0].interface Opt: list interface	Defines the list of interfaces to be served by dnsmasq. If you do not select a specific interface, dnsmasq will serve on all interfaces. Configured interfaces are shown via the web GUI. Lan — Serve only on LAN interface. Range
Web: Local Server UCI: dhcp.@dnsmasq[0].local Opt: local	Specifies the local domain. Names matching this domain are never forwarded and are resolved from DHCP or host files only. /lan/ Range
Web: Local Domain UCI: dhcp.@dnsmasq[0].domain Opt: domain	Specifies local domain suffix appended to DHCP names and hosts file entries. lan Range
Web: Log Queries UCI: dhcp.@dnsmasq[0].logqueries Opt: logqueries	Writes received DNS requests to syslog. 0 — Disabled. 1 — Enabled.
Web: DNS Forwardings UCI: dhcp.@dnsmasq[0].server Opt: list server	List of DNS server to forward requests to. To forward specific domain requests only, use // syntax. When using UCI, enter multiple servers with a space between them. (blank) — No DNS server configured. Range
Web: Rebind Protection UCI: dhcp.@dnsmasq[0].rebind_protection Opt: rebind_protection	Enables DNS rebind attack protection by discarding upstream RFC1918 responses. 0 — Disabled. 1 — Enabled.
Web: Allow Localhost UCI: dhcp.@dnsmasq[0].rebind_localhost Opt: rebind_localhost	Defines whether to allow upstream responses in the 127.0.0.0/8 range. This is required for DNS based blacklist services. Only takes effect if rebind protection is enabled. 0 — Disabled. 1 — Enabled.
Web: Domain Whitelist UCI: dhcp.@dnsmasq[0].rebind_domain Opt: list rebind_domain	Defines the list of domains to allow RFC1918 responses to. Only takes effect if rebind protection is enabled. When using UCI multiple servers should be entered with a space between them. (blank) — No list configured. Range

Table 66: Information table for general server settings

23.2.2 Dnsmasq: resolv and host files

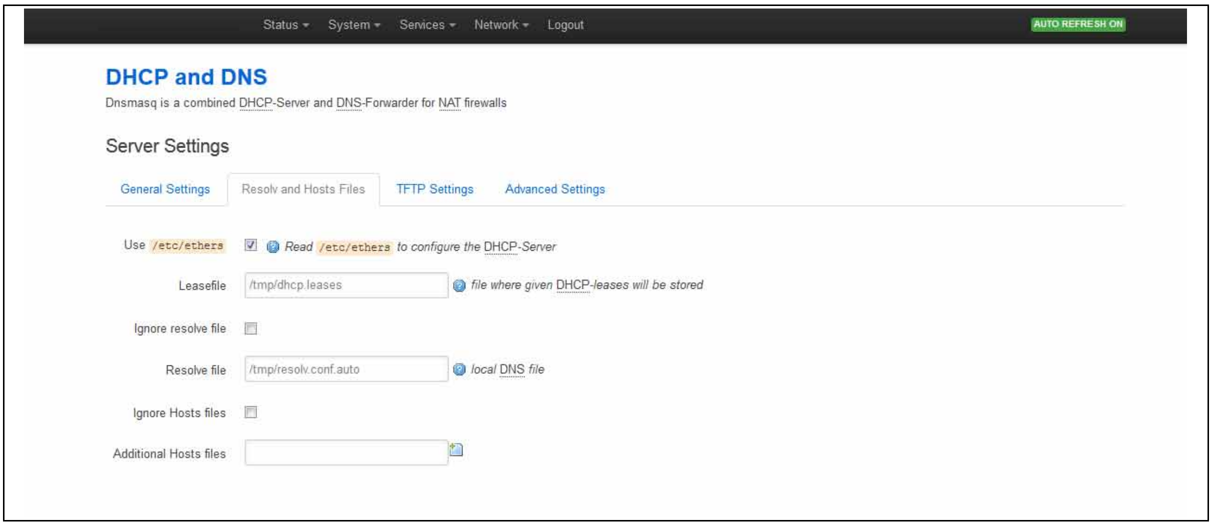


Figure 116: The resolv and host files section

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Use /etc/ethers
UCI: dhcp.@dnsmasq[0].readethers
Opt: readethers</td><td>Defines whether static lease entries are read from /etc/ethers.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
<tr><td>Web: Leasefile
UCI: dhcp.@dnsmasq[0].leasefile
Opt: leasefile</td><td>Defines the file where given DHCP leases will be stored. The DHCP lease file allows leases to be picked up again if dnsmasq is restarted.<table><tr><td>/tmp/dhcp.leases</td><td>Store DHCP leases in this file.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Ignore resolve file
UCI: dhcp.@dnsmasq[0].noresolv
Opt: noresolv</td><td>Defines whether to use the local DNS file for resolving DNS.<table><tr><td>0</td><td>Use local DNS file.</td></tr><tr><td>1</td><td>Ignore local DNS file.</td></tr></table></td></tr>
<tr><td>Web: Resolve file
UCI: dhcp.@dnsmasq[0].resolvfile
Opt: resolvfile</td><td>Defines the local DNS file. Default is /tmp/resolv.conf.auto</td></tr>
<tr><td>Web: Ignore Hosts files
UCI: dhcp.@dnsmasq[0].nohosts
Opt: nohosts</td><td>Defines whether to use local host's files for resolving DNS.<table><tr><td>0</td><td>Use local hosts file.</td></tr><tr><td>1</td><td>Ignore local hosts file.</td></tr></table></td></tr>
<tr><td>Web: Additional Hosts files
UCI: dhcp.@dnsmasq[0].addnhosts
Opt: list addnhosts</td><td>Defines local host's files. When using UCI multiple servers should be entered with a space between them.</td></tr>
</table>

Table 67: Information table for resolv and host files section

23.2.3 Dnsmasq: TFTP settings

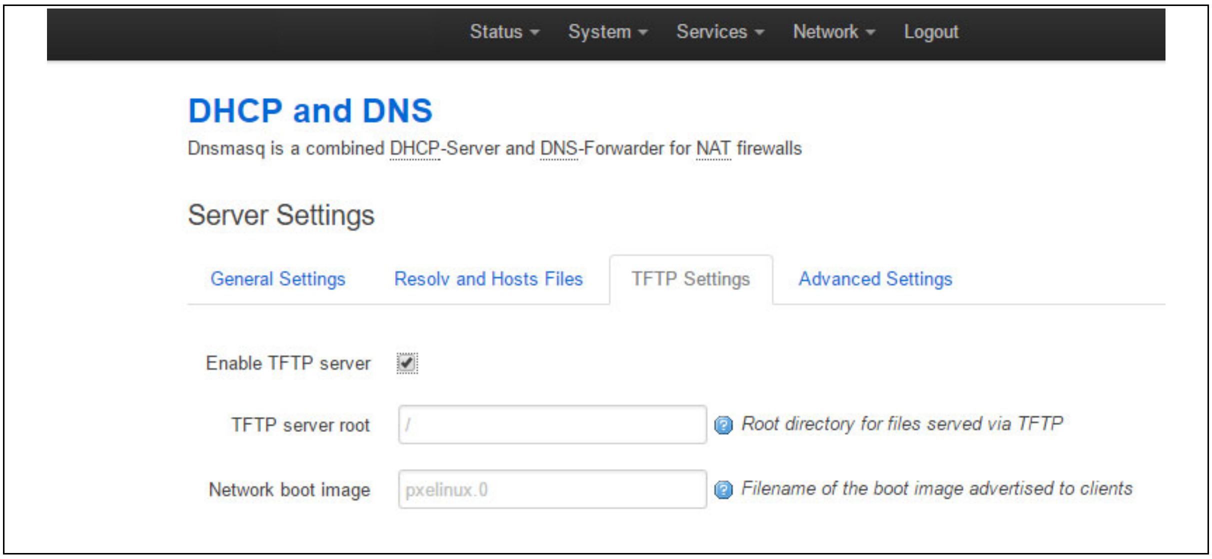


Figure 117: The TFTP settings section

Web Field / UCI / Package Option	Description
Web: Enable TFTP Server UCI: dhcp.@dnsmasq[0].enable_tftp Opt: enable_tftp	Enables the TFTP server. 0 Disabled. 1 Enabled.
Web: Enable TFTP Server UCI: dhcp.@dnsmasq[0].tftp_root Opt: tftp_root	Defines root directory for file served by TFTP.
Web: Enable TFTP Server UCI: dhcp.@dnsmasq[0].dhcp_boot Opt: dhcp_boot	Defines the filename of the boot image advertised to clients. This specifies BOOTP options, in most cases just the file name.

Table 68: Information table for TFTP settings

23.2.4 Dnsmasq: advanced settings

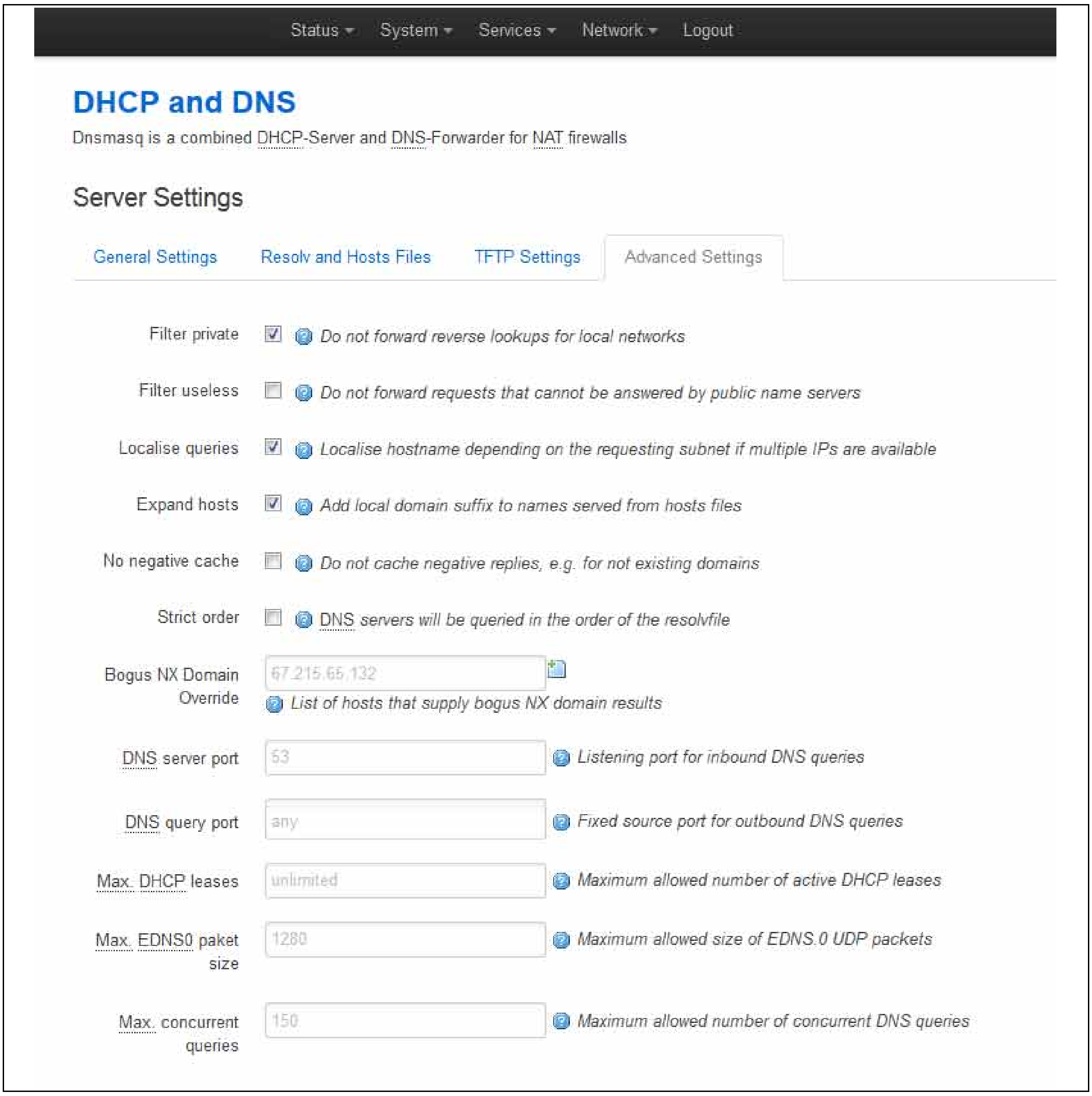


Figure 118: The advanced settings page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Filter private
UCI: dhcp.@dnsmasq[0].
Opt: boguspriv</td><td>Enables disallow option for forwarding reverse lookups for local networks. This rejects reverse lookups to private IP ranges where no corresponding entry exists in /etc/hosts.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
<tr><td>Web: Filter useless
UCI: dhcp.@dnsmasq[0].filterwin2k
Opt: filterwin2k</td><td>Enables disallow option for forwarding requests that cannot be answered by public name servers. Normally enabled for dial on demand interfaces.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
</table>

Web: Localise queries UCI: dhcp.@dnsmasq[0].localise_queries Opt: localise_queries	Defines whether to uses IP address to match the incoming interface if multiple addresses are assigned to a host name in /etc/hosts. 1 – Enabled. 0 – Disabled.
Web: Expand hosts UCI: dhcp.@dnsmasq[0].expandhosts Opt: expandhosts	Adds a local domain suffix to names served from host files. 1 – Enabled. 0 – Disabled.
Web: No negative cache UCI: dhcp.@dnsmasq[0].nonegcache Opt: nonegcache	Enable this to stop caching of negative replies. For example, non-existing domains. 1 – Enabled. 0 – Disabled.
Web: Strict order UCI: dhcp.@dnsmasq[0].strictorder Opt: strictorder	Enable this to query DNS servers in the order of the resolve file. 1 – Enabled. 0 – Disabled.
Web: Bogus NX Domain override UCI: dhcp.@dnsmasq[0].bogusnxdomain Opt: list bogusnxdomain	A list of hosts that supply bogus NX domain results. When using UCI multiple servers should be entered with a space between them. Empty list Range
Web: DNS server port UCI: dhcp.@dnsmasq[0].port Opt: port	Listening port for inbound DNS queries. 53 – Set to 0 to disable DNS functionality. Range – 0 - 65535
Web: DNS query port UCI: dhcp.@dnsmasq[0].queryport Opt: queryport	Defines fixed source port for outbound DNS queries. any Range – any; 0 - 65535
Web: Max DHCP leases UCI: dhcp.@dnsmasq[0].dhcpleasemax Opt:dhcpleasemax	Defines the maximum allowed number of active DHCP leases. unlimited Range
Web: Max EDNS0 packet size UCI: dhcp.@dnsmasq[0].ednspacket_max Opt: ednspacket_max	Defines the maximum allowed size of EDNS.0 UDP packets in bytes. 1280 – 1280 bytes Range
Web: Max concurrent queries UCI: dhcp.@dnsmasq[0].dnsforwardmax Opt: dnsforwardmax	Maximum allowed number of concurrent DNS queries. 150 – 1280 bytes Range

Table 69: Information table for advanced settings

23.2.5 Active leases

This section displays all currently active leases.

Figure 119: The active leases section

Web Field / UCI / Package Option	Description
Web: Hostname UCI: dhcp.@host[0].name Opt: name	Displays the hostname of the client.
Web: IPv4 Address UCI: dhcp.@host[0].ip Opt: ip	Displays the IP address of the client.
Web: MAC Address UCI: dhcp.@host[0].mac Opt: mac	Displays the MAC address of the client.
Web: Lease time remaining UCI: n/a Opt: n/a	Displays the remaining lease time.

Table 70: Information table for active leases section

23.2.6 Static leases

Use static leases to assign fixed IP addresses and symbolic hostnames to DHCP clients. Static leases are also required for non-dynamic interface configurations where only hosts with a corresponding lease are served. Click **Add** to add a new lease entry.

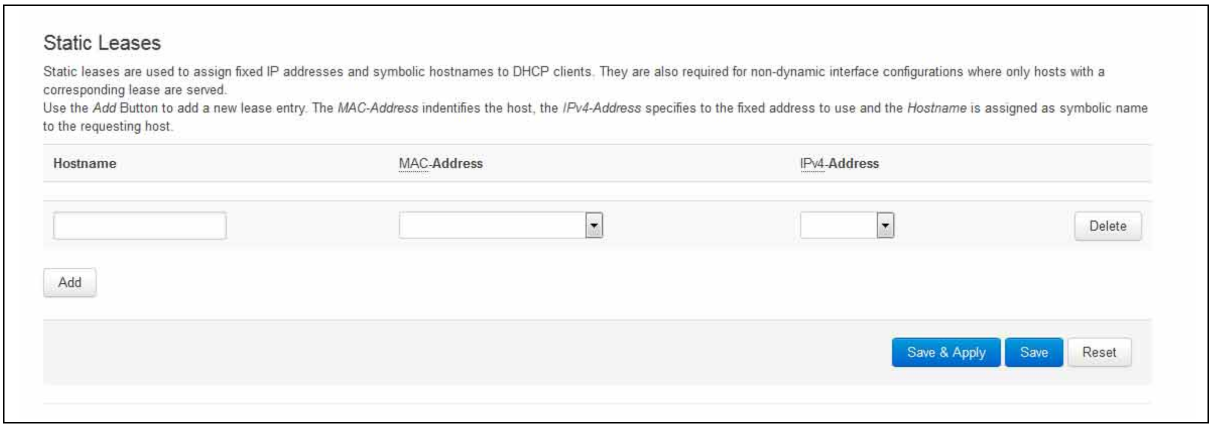


Figure 120: The static leases section

Web Field / UCI / Package Option	Description
Web: Hostname UCI: dhcp.@host[0].name Opt: name	Defines the optional symbolic name to assign to this static DHCP entry. 1 Enabled. 0 Disabled.
Web: MAC Address UCI: dhcp.@host[0].mac Opt: mac	Defines the hardware address that identifies the host.
Web: IPv4 Address UCI: dhcp.@host[0].ip Opt: ip	The IPv4 address specifies the fixed address to use for this host.

Table 71: Information table for static leases

23.3 Configuring DHCP and DNS using UCI

23.3.1 Common options section

Possible section types of the DHCP configuration file are defined below. Not all types may appear in the file and most of them are only needed for special configurations. Common configurations are Common Options, DHCP Pools and Static Leases.

The configuration section type dnsmasq determines values and options relevant to the overall operation of dnsmasq and the DHCP options on all interfaces served. The following table lists all available options, their default value, as well as the corresponding dnsmasq command line option.

These are the default settings for the common options:

```
root@VA_router:~# uci show dhcp
dhcp.@dnsmasq[0]=dnsmasq
dhcp.@dnsmasq[0].domainneeded=1
dhcp.@dnsmasq[0].boguspriv=1
dhcp.@dnsmasq[0].filterwin2k=0
dhcp.@dnsmasq[0].localise_queries=1
dhcp.@dnsmasq[0].logqueries=1
dhcp.@dnsmasq[0].rebind_protection=1
dhcp.@dnsmasq[0].rebind_localhost=1
dhcp.@dnsmasq[0].local=/lan/
dhcp.@dnsmasq[0].domain=lan
dhcp.@dnsmasq[0].expandhosts=1
dhcp.@dnsmasq[0].nonegcache=0
dhcp.@dnsmasq[0].authoritative=1
dhcp.@dnsmasq[0].readethers=1
dhcp.@dnsmasq[0].leasefile=/tmp/dhcp.leases
dhcp.@dnsmasq[0].noresolve=0
dhcp.@dnsmasq[0].resolvfile=/tmp/resolv.conf.auto
dhcp.@dnsmasq[0].nohosts=0
dhcp.@dnsmasq[0].addnhosts=hostfile1 hostfile2
dhcp.@dnsmasq[0].interface=lan
dhcp.@dnsmasq[0].server=1.1.1.1 2.2.2.2
dhcp.@dnsmasq[0].rebind domain=tes.domain
dhcp.@dnsmasq[0].enable_tftp=0
dhcp.@dnsmasq[0].tftp_root=/tmp/tftp
dhcp.@dnsmasq[0].dhcp_boot=boot.image
dhcp.@dnsmasq[0].nonegcache=0
```

```
dhcp.@dnsmasq[0].strictorder=0
dhcp.@dnsmasq[0].bogusnxdomain=1.1.1.1  2.2.2.2
dhcp.@dnsmasq[0].port=53
dhcp.@dnsmasq[0].dhcpleasemax=150
dhcp.@dnsmasq[0].ednspacket_max=1280
dhcp.@dnsmasq[0].dnsforwardmax=150
root@VA_router:~# uci show dhcp
config 'dnsmasq'
      option domainneeded '1'
        option rebind_protection '1'
        option rebind_localhost '1'
        option local '/lan/'
        option domain 'lan'
        option authoritative '1'
        option readethers '1'
        option leasefile '/tmp/dhcp.leases'
        list interface 'lan'
        list server '1.2.3.4'
        list server '4.5.6.7'
        list rebind_domain 'test1.domain'
        list rebind_domain 'tes2.domain'
        option logqueries '1'
        option resolvfile '/tmp/resolv1.conf.auto'
        list addnhosts 'hosts1'
        list addnhosts 'hosts2'
        option enable_tftp '1'
        option tftp_root '/tmp/tftp'
        option dhcp_boot 'boot.image'
        option filterwin2k '1'
        option nonegcache '1'
        option strictorder '1'
        list bogusnxdomain '1.1.1.1 '
        list bogusnxdomain '2.2.2.2'
        option port '53'
        option dhcpleasemax '150'
        option ednspacket_max '1280'
        option dnsforwardmax '150'
```

Options `local` and `domain` enable dnsmasq to serve entries in /etc/hosts as well as the DHCP client's names as if they were entered into the LAN DNS domain.

For options `domainneeded`, `boguspriv`, `localise_queries`, and `expandhosts` make sure that requests for these local host names (and the reverse lookup) never get forwarded to the upstream DNS servers.

23.4 Configuring DHCP pools using UCI

Sections of the type dhcp specify per interface lease pools and settings. Typically there is at least one section of this type present in the /etc/config/dhcp file to cover the LAN interface.

You can disable a lease pool for a specific interface by specifying the ignore option in the corresponding section.

A minimal example of a dhcp section is shown below.

```
root@VA_router:~# uci show dhcp.lan
dhcp.lan=dhcp
dhcp.lan.interface=lan
dhcp.lan.start=100
dhcp.lan.limit=150
dhcp.lan.leasetime=12h
dhcp.lan.ignore=0
root@VA_router:~# uci export dhcp
config 'dhcp' 'lan'
      option 'interface'   'lan'
      option 'start'       '100'
      option 'limit'        '150'
      option 'leasetime'   '12h'
      option ignore   0
```

<table>
<tr><th>UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: n/a
UCI: dhcp.<pool_name>.interface
Opt: interface</td><td>Defines the interface that is served by this DHCP pool. This must be one of the configured interfaces.<table><tr><td>lan</td><td>Enabled.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a
UCI: dhcp.<pool_name>.start
Opt: start</td><td>Defines the offset from the network address for the start of the DHCP pool. It may be greater than 255 to span subnets.<table><tr><td>100</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a
UCI: dhcp.<pool_name>.limit
Opt: limit</td><td>Defines the offset from the network address for the end of the DHCP pool.<table><tr><td>150</td><td></td></tr><tr><td>Range</td><td>0 - 255</td></tr></table></td></tr>
</table>

Web: n/a UCI: dhcp.<pool_name>.leasetime Opt: leasetime	Defines the lease time of addresses handed out to clients, for example 12h or 30m. 12h | 12 hours Range |
Web: n/a UCI: dhcp.<pool_name>.ignore Opt: ignore	Defines whether this DHCP pool is enabled. 0 | DHCP pool enabled. 1 | DHCP pool disabled.
Web: n/a UCI: dhcp.<pool_name>.force Opt: force	Forces DHCP serving on the specified interface even if another DHCP server is detected on the same network segment. 0 | Disabled. 1 | Enabled.
Web: n/a UCI: dhcp.<pool_name>.dhcp_option Opt: list dhcp_option	Defines additional options to be added for this dhcp pool. For example with 'list dhcp_option 26,1470' or 'list dhcp_option mtu, 1470' you can assign a specific MTU per DHCP pool. Your client must accept the MTU option for this to work. | No options defined. Syntax | Option_number, option_value.
Web: n/a UCI: dhcp.<pool_name>.dynamicdhcp Opt: dynamicdhcp	Defines whether to allocate DHCP leases. 1 | Dynamically allocate leases. 0 | Use /etc/ethers file for serving DHCP leases.
Web: n/a UCI: dhcp.<pool_name>.dynamicdhcp Opt: networkid	Assigns a network-id to all clients that obtain an IP address from this pool.

Table 72: Information table for DHCP pool UCI and package options

23.5 Configuring static leases using UCI

You can assign fixed IP addresses to hosts on your network, based on their MAC (hardware) address.

```
root@VA_router:~# uci show dhcp.mypc
dhcp.mypc=host
root@VA_router:~# uci show dhcp.mypc
dhcp.mypc.ip=192.168.1.2
dhcp.mypc.mac=00:11:22:33:44:55
dhcp.mypc.name=mypc
root@VA_router:~# uci export dhcp
config host 'mypc'
        option ip       '192.168.1.2'
        option mac      '00:11:22:33:44:55'
        option name     'mypc'
```

This adds the fixed IP address 192.168.1.2 and the name "mypc" for a machine with the (Ethernet) hardware address 00:11:22:33:44:55.

24 Configuring DHCP client

This section describes how to configure an interface as a DHCP client. This section will only detail the configuration for DHCP client. For information on how to configure other interface options such as firewall zone, mapping of switch ports, etc, refer to standard interface configuration document.

24.1 Configuration packages used

Package	Sections
network	interface

24.2 Configuring DHCP client using the web interface

DHCP client is configured under the interface configuration by setting the interface protocol to DHCP Client. To create and edit interfaces via the web interface, in the top menu, click **Network -> Interfaces**. The Interfaces overview page appears.

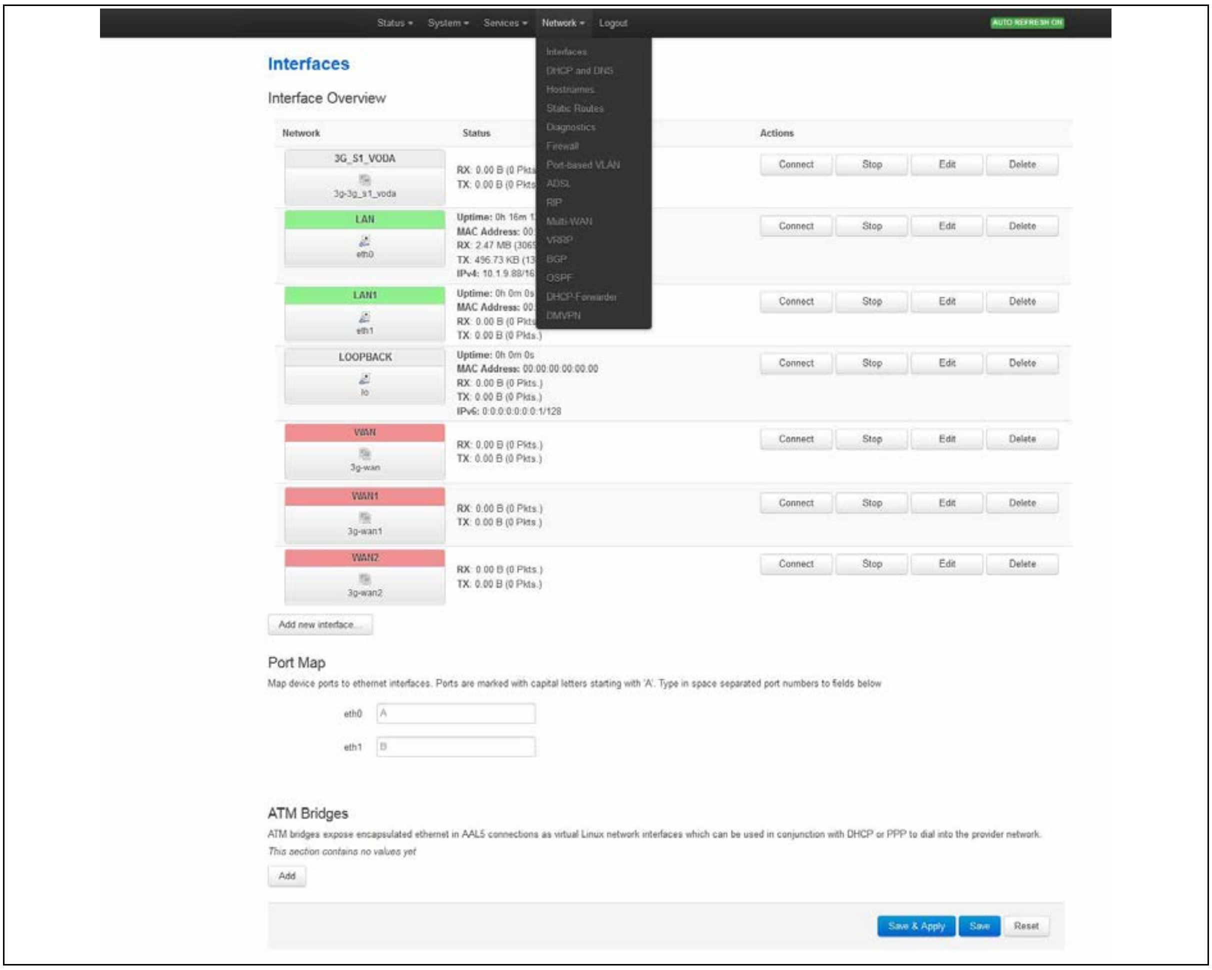


Figure 121: The interfaces overview page

There are three sections in the Interfaces page.

Section	Description
Interface Overview	Shows existing interfaces and their status. You can create new, and edit existing interfaces here.
Port Map	In this section you can map device ports to Ethernet interfaces. Ports are marked with capital letters starting with 'A'. Type in space-separated port character in the port map fields.
ATM Bridges	ATM bridges expose encapsulated Ethernet in AAL5 connections as virtual Linux network interfaces, which can be used in conjunction with DHCP or PPP to dial into the provider network.

24.2.1 Editing an existing interface for DHCP client

To edit an existing interface, from the interface tabs at the top of the page, select the interface you wish to configure. Alternatively, click **Edit** in the interface's row.

24.2.2 Creating a new interface for DHCP client

To create a new interface, in the Interface Overview section, click **Add new interface**. The Create Interface page appears.

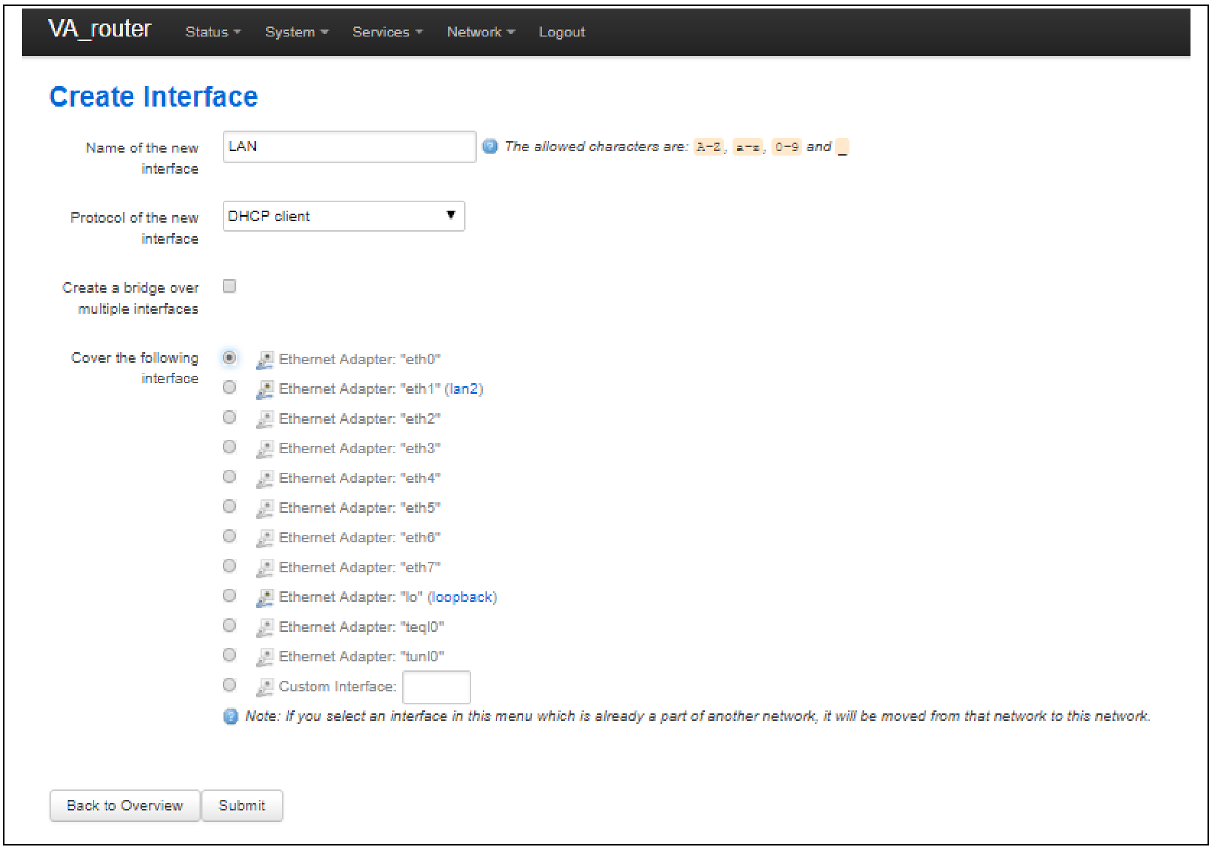


Figure 122: The create interface page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface
UCI: network.<if name>
Opt: config interface</td><td>Assigns a logical name to the interface. The network interface section will assign this name (<if name>).
Type the name of the new interface.
Allowed characters are A-Z, a-z, 0-9 and _</td></tr>
<tr><td>Web: Protocol of the new interface
UCI: network.<if name>.proto
Opt: proto</td><td>Specifies what protocol the interface will operate on. Select DHCP Client.
<table>
<tr><th>Option</th><th>Description</th><th>UCI</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td><td>Static</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td><td>dhcp</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td><td>Empty</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td><td></td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport.</td><td></td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td><td></td></tr>
<tr><td>IOT</td><td></td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td><td></td></tr>
<tr><td>PPP</td><td>Point to Point Protocol</td><td></td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td><td></td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td><td></td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Create a bridge over multiple interfaces
UCI: network.<if name>.type
Opt: type</td><td>If you select this option, then the new logical interface created will act as a bridging interface between the chosen existing physical interfaces.
<table>
<tr><td>Empty</td><td></td></tr>
<tr><td>Bridge</td><td>Configures a bridge over multiple interfaces.</td></tr>
</table></td></tr>
<tr><td>Web: Cover the following interface
UCI: network.<if name>.ifname
Opt: ifname</td><td>Physical interface name to assign to this logical interface. If creating a bridge over multiple interfaces select two interfaces to bridge. When using UCI, the interface names should be separated by a space e.g. option ifname ‘eth2 eth3’.</td></tr>
</table>

Table 73: Information table for the create new interface page

Click **Submit**. The Interface configuration page appears. There are three sections:

Section	Description
Common Configuration	Configure the interface settings such as protocol, IP address, gateway, netmask, custom DNS servers, MTU and firewall configuration.
IP-Aliases	Assign multiple IP addresses to the interface.
DHCP Server	Configure DHCP server settings for this interface.

24.2.3 Common configuration

The common configuration section has four sub sections.

Section	Description
General Setup	Configure the basic interface settings such as protocol, IP address, gateway, netmask, custom DNS servers.
Advanced Settings	'Bring up on boot', 'Monitor interface state', Override MAC address, Override MTU and 'Use gateway metric'.
Physical Settings	Bridge interfaces, VLAN PCP to SKB priority mapping.
Firewall settings	Assign a firewall zone to the interface.

Only **General setup** and **Advanced Settings** have DHCP client option configuration options

24.2.3.1 Common configuration – general setup

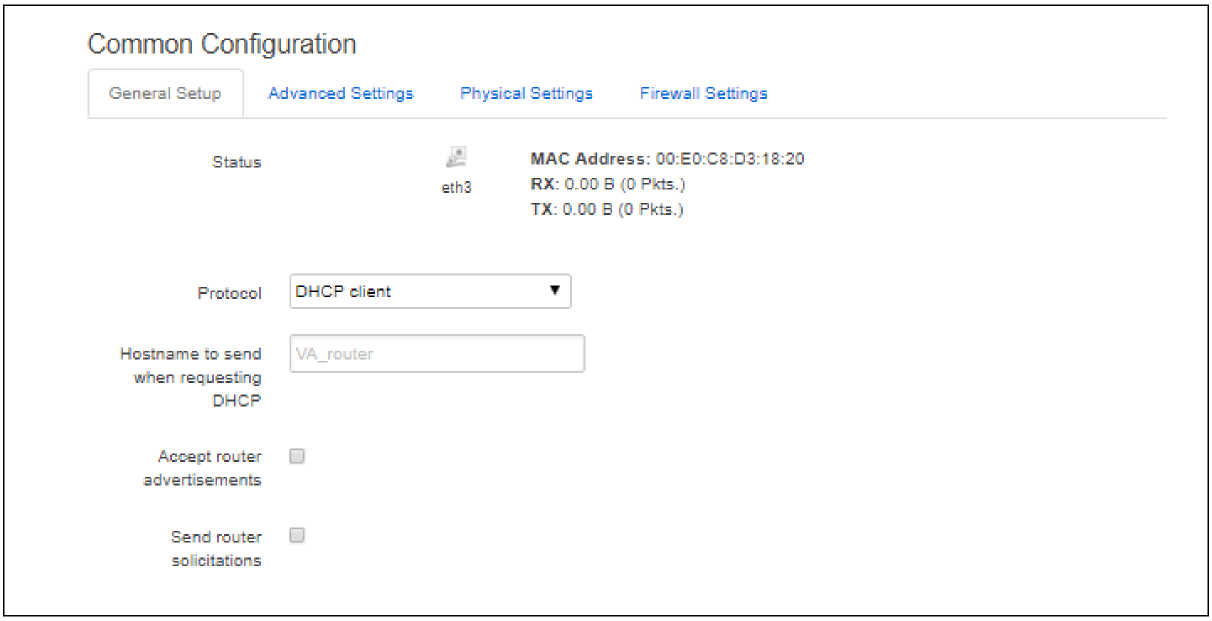


Figure 123: The interface general setup configuration page for DHCP client protocol

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Status</td><td>Shows the current status of the interface.</td></tr>
<tr><td>Web: Protocol
UCI: network.<if name>.proto
Opt: proto</td><td>Protocol type. The interface protocol may be one of the options shown below. The protocol selected in the previous step will be displayed as default but can be changed if required.
Select DHCP Client.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport.</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol.</td></tr>
<tr><td>PPP</td><td>Point-to-Point protocol</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td></tr>
<tr><td>LTE/UMTS/ GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem.</td></tr>
</table></td></tr>
<tr><td>Web: Hostname to send when requesting DHCP
UCI: network.<if name>.hostname
Opt: hostname</td><td>Defines the hostname to include in DHCP requests</td></tr>
<tr><td>Web: Accept router advertisements
UCI: network.<if name>.accept_ra
Opt: accept_ra</td><td>Specifies whether to accept IPv6 Router Advertisements on this interface (optional).
Note: default is 1 if protocol is set to DHCP, otherwise the setting defaults to 0.
<table>
<tr><td>0</td><td>Does not accept IPv6 router advertisements.</td></tr>
<tr><td>1</td><td>Accept IPv6 router advertisements.</td></tr>
</table></td></tr>
<tr><td>Web: Send router solicitations
UCI: network.<if name>.send_rs
Opt: send_rs</td><td>Specifies whether to send router Solicitations on this interface (optional).
Note: defaults to 1 for Static protocol, otherwise the setting defaults to 0.
<table>
<tr><td>0</td><td>Do not send router solicitations</td></tr>
<tr><td>1</td><td>Send router solicitations</td></tr>
</table></td></tr>
</table>

Table 74: Information table for general setup configuration settings for DHCP client protocol

24.2.3.2 Common configuration: advanced settings

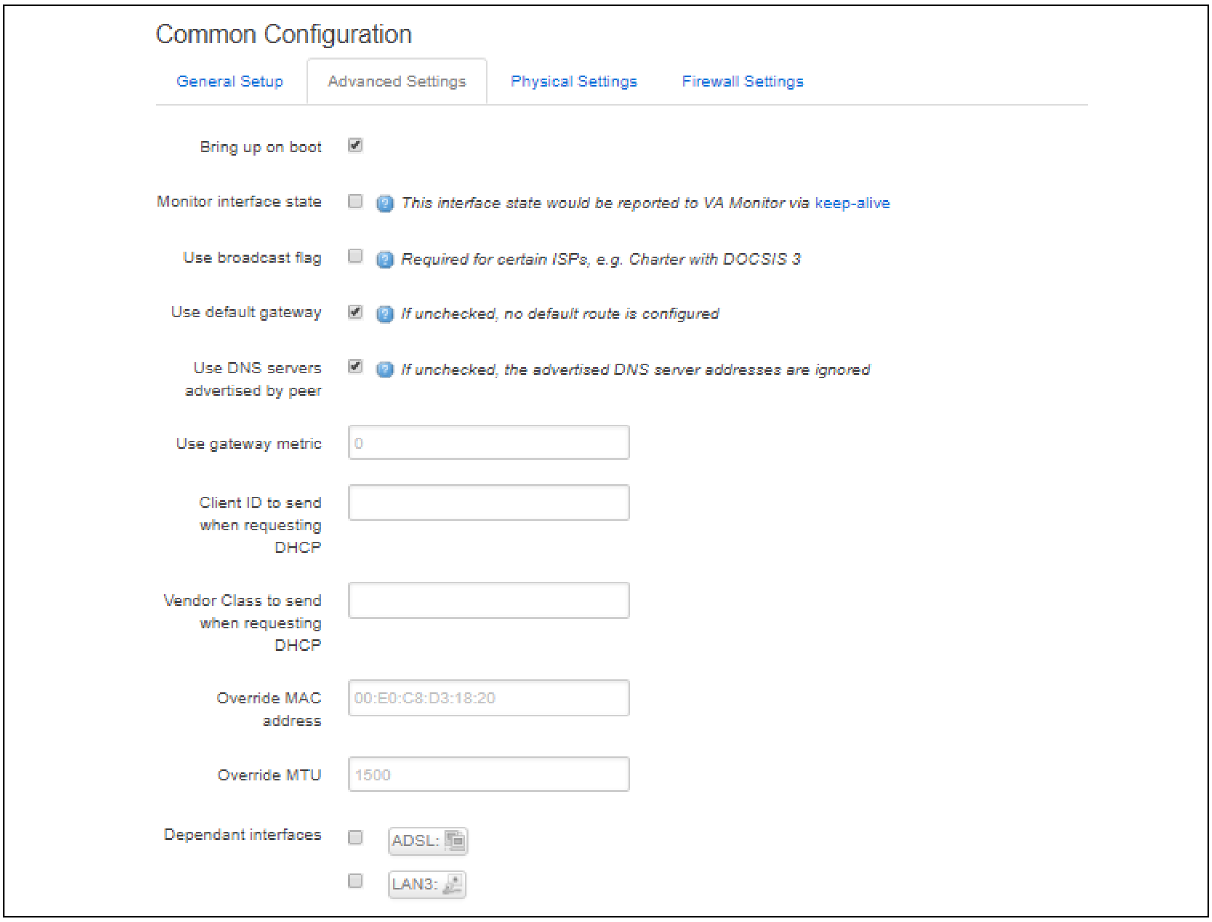


Figure 124: The interface advanced settings page for DHCP client protocol

Web Field / UCI / Package Option	Description
Web: Bring up on boot UCI: network.<if name>.auto Opt: auto	Enables the interface to connect automatically on boot up. 0 — Disabled. 1 — Enabled.
Web: Monitor interface state UCI: network.<if name>.monitored Opt: monitored	Enabled if status of interface is presented on Monitoring platform. 0 — Disabled. 1 — Enabled.
Web: Use broadcast flag UCI: network.<if name>.broadcast Opt: broadcast	Enables the broadcast flag in DHCP requests (required for certain ISPs). 0 — Disabled. 1 — Enabled.
Web: Use default gateway UCI: network.<if name>.gateway Opt: gateway	Defines whether to suppress the DHCP assigned default gateway. When disabled via web option gateway is set to 0.0.0.0. 0 — Disabled (option gateway set to 0.0.0.0) 1 — Enabled.
Web: Use DNS servers advertised by peer UCI: n/a Opt: n/a	Defines whether to override DHCP assigned DNS servers with configured list of DNS servers. When unchecked allows configuration of custom DNS servers via web. There is no uci option set when checking or unchecking this option.

<table>
<tr><td>Web: Use custom DNS servers
UCI: network.<if name>.dns
Opt: dns</td><td>Defines whether to override DHCP assigned DNS servers with configured list of DNS servers.
Multiple DNS Servers are separated by a space if using UCI. Example: <code>option dns '1.1.1.1 2.2.2.2'</code><table><tr><td>0</td><td>Disabled (option gateway set to 0.0.0.0)</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Use gateway metric
UCI: network.<if name>.metric
Opt: metric</td><td>Specifies the default route metric to use for this interface.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Client ID to send when requesting DHCP
UCI: network.<if name>. clientid
Opt: clientid</td><td>Defines whether to override the client identifier in DHCP requests.<table><tr><td>Blank</td><td>Do not override.</td></tr><tr><td>Range</td><td>Override.</td></tr></table></td></tr>
<tr><td>Web: Vendor Class to send when requesting DHCP
UCI: network.<if name>.vendorid
Opt: vendorid</td><td>Defines whether to override the vendor class in DHCP requests.<table><tr><td>Blank</td><td>Do not override.</td></tr><tr><td>Range</td><td>Override.</td></tr></table></td></tr>
<tr><td>Web: Override MAC address
UCI: network.<if name>.macaddr
Opt: macaddr</td><td>Override the MAC address assigned to this interface. Must be in the form: hh:hh:hh:hh:hh:hh, where h is a hexadecimal number.</td></tr>
<tr><td>Web: Override MTU
UCI: network.<if name>.mtu
Opt: mtu</td><td>Defines the value to override the default MTU on this interface.<table><tr><td>1500</td><td>1500 bytes</td></tr></table></td></tr>
<tr><td>Web: Dependant Interfaces
UCI: network.[if_name].dependants
Opt: dependants</td><td>Lists interfaces that are dependent on this parent interface. Dependant interfaces will go down when parent interface is down and will start or restart when parent interface starts.
Separate multiple interfaces by a space when using UCI.
Example: <code>option dependants 'PPPADSL MOBILE'</code>
This replaces the following previous options in child interfaces.<table><tr><td>gre</td><td>option local_interface</td></tr><tr><td>lt2p</td><td>option src_ipaddr</td></tr><tr><td>iot</td><td>option wan1 wan2</td></tr><tr><td>6in4</td><td>option ipaddr</td></tr><tr><td>6to4</td><td>option ipaddr</td></tr></table></td></tr>
<tr><td>Web: SNMP Alias ifIndex
UCI: network.@interface[X].snmp_alias_ifindex
Opt: snmp_alias_ifindex</td><td>Defines a static SNMP interface alias index for this interface, that can be polled using via the SNMP interface index (snmp_alias_ifindex+1000)<table><tr><td>Blank</td><td>No SNMP interface alias index</td></tr><tr><td>Range</td><td>0 - 4294966295</td></tr></table></td></tr>
</table>

Table 75: Information table for advanced settings for DHCP client protocol

24.3 Configuring DHCP client using command line

The configuration files for DHCP client are stored on **/etc/config/network**

24.3.1 DHCP client using UCI

```
root@VA_router:~# uci show network
      …..
network.DHCPCLIENTLAN=interface
network.DHCPCLIENTLAN.proto=dhcp
```

```
network.DHCPCLIENTLAN.ifname=eth3
network.DHCPCLIENTLAN.monitored=0
network.DHCPCLIENTLAN.broadcast=0
network.DHCPCLIENTLAN.accept_ra=1
network.DHCPCLIENTLAN.send_rs=0
network.DHCPCLIENTLAN.metric=1
```

24.3.2 DHCP client using package options

```
root@VA_router:~# uci export network
package network
      ……
config interface 'DHCPCLIENTLAN'
        option proto 'dhcp'
        option ifname 'eth3'
        option monitored '0'
        option broadcast '0'
        option accept_ra '1'
        option send_rs '0'
        option metric '1'
```

24.4 DHCP client diagnostics

24.4.1 Interface status

To see IP address of DHCP client interface, enter **ifconfig**:

```
root@VA_router:~# ifconfig
3g-CDMA  Link encap:Point-to-Point Protocol
         inet addr:10.33.152.100  P-t-P:178.72.0.237  Mask:255.255.255.255
         UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1400  Metric:1
         RX packets:6 errors:0 dropped:0 overruns:0 frame:0
         TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:3
         RX bytes:428 (428.0 B)  TX bytes:2986 (2.9 KiB)

eth0      Link encap:Ethernet  HWaddr 00:E0:C8:12:12:15
          inet addr:192.168.100.1  Bcast:192.168.100.255
Mask:255.255.255.0
          inet6 addr: fe80::2e0:c8ff:fe12:1215/64 Scope:Link
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6645 errors:0 dropped:0 overruns:0 frame:0
          TX packets:523 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:569453 (556.1 KiB)  TX bytes:77306 (75.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:385585 errors:0 dropped:0 overruns:0 frame:0
          TX packets:385585 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:43205140 (41.2 MiB)  TX bytes:43205140 (41.2 MiB)
```

To display a specific interface, enter:

```
root@VA_router:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:E0:C8:12:12:15
          inet addr:192.168.100.1  Bcast:192.168.100.255
Mask:255.255.255.0
          inet6 addr: fe80::2e0:c8ff:fe12:1215/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7710 errors:0 dropped:0 overruns:0 frame:0
          TX packets:535 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:647933 (632.7 KiB)  TX bytes:80978 (79.0 KiB)
```

24.4.2 ARP table status

To show the current ARP table of the router, enter:

```
root@GW7314:~# arp
? (10.67.253.141) at 30:30:41:30:43:36 [ether]  on eth8
? (10.47.48.1) at 0a:44:b2:06 [ether]  on gre-gre1
```

24.4.3 Route status

To show the current routing status, enter:

```
root@VA_router:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.100.0   *               255.255.255.0   U     0      0        0 eth0
```

Note: a route will only be displayed in the routing table when the interface is up.

25 Configuring DHCP forwarding

This section describes how to configure the router to forward DHCP requests from an interface to a network DHCP server.

25.1 Configuration packages used

Package	Sections
dhcp_fwd	dhcpfwd

25.2 Configuring DHCP forwarding using the web interface

To configure DHCP forwarding using the web interface, in the top menu, click **Network -> DHCP-Forwarder**.

The DHCP forwarder page appears. The web GUI creates a dhcpfwd section called main so this will be used in the uci examples below.

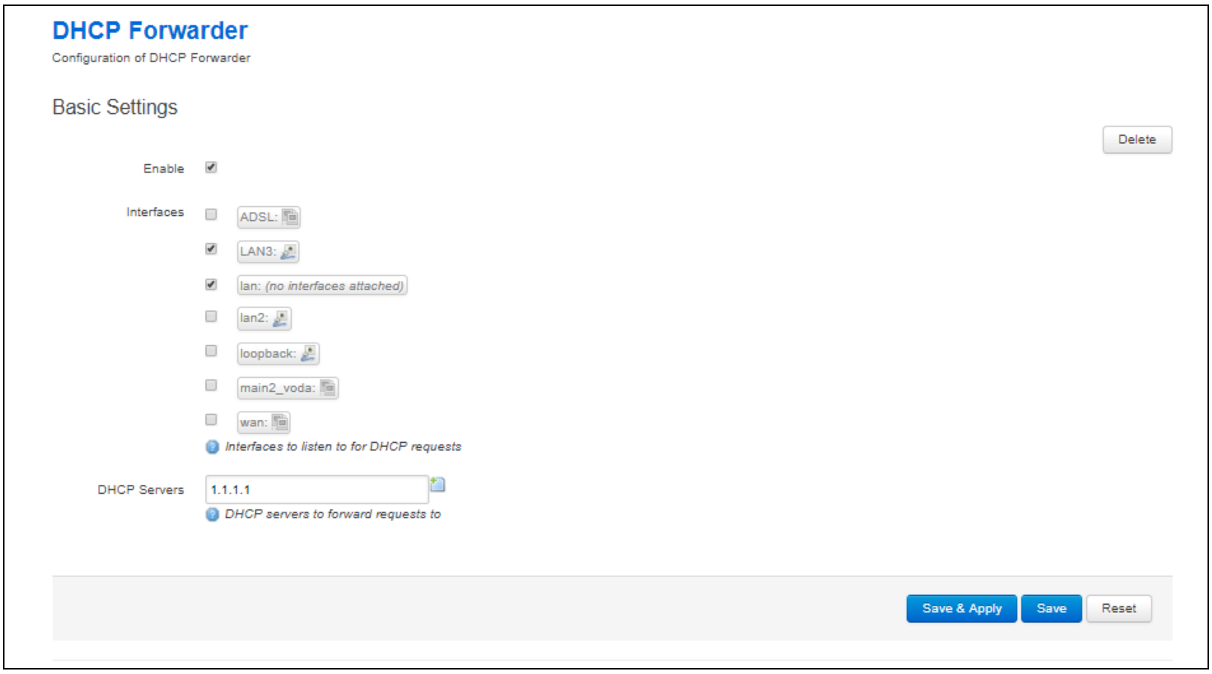


Figure 125: The DHCP forwarder configuration page

Web Field/UCI/Package Option	Description
Web: Enabled UCI: dhcp_fwd.main.enabled Opt: enabled	Defines whether DHCP forwarding is enabled or disabled. 0 — Do not send router solicitations. 1 — Send router solicitations.

Web: Interfaces
UCI: dhcp_fwd.main.listen_interface
Opt: list listen_interface
	Defines a list of the source interface name(s) to forward DHCP messages from. Multiple interface_name(s) are entered using `uci set` and `uci add_list` commands. Example:
	`uci set dhcp_fwd.main.listen_interface=LAN1`
	`uci add_list dhcp_fwd.main.listen_interface=LAN2`
	or using a list of options via package options
	`list listen_interface ‘LAN1’`
	`list listen_interface ‘LAN2’`

Web: DHCP Servers
UCI: dhcp_fwd.main.server
Opt: list server
	Defines a list of the network DHCP servers to forward DHCP messages to. Multiple interface_name(s) are entered using `uci set` and `uci add_list` commands. Example:
	`uci set dhcp_fwd.main.server=1.1.1.1`
	`uci add_list dhcp_fwd.main.main.server=2.2.2.2`
	or using a list of options via package options
	`list server ‘1.1.1.1’`
	`list server ‘2.2.2.2’`

Table 76: Information table for the DHCP forwarder section

25.3 Configuring DHCP forwarding using command line

The configuration files for DHCP client are stored in **/etc/config/dhcp_fwd**

25.3.1 DHCP forwarding using UCI

```
root@VA_router:~# uci show dhcp_fwd
dhcp_fwd.main=dhcpfwd
dhcp_fwd.main.enabled=1
dhcp_fwd.main.listen_interface=LAN3 lan2
dhcp_fwd.main.server=1.1.1.1
```

25.3.2 DHCP forwarding using package options

```
root@VA_router:~# uci export dhcp_fwd
package dhcp_fwd

config dhcpfwd 'main'
        option enabled '1'
        list listen_interface 'LAN3'
        list listen_interface 'lan2'
        list server '1.1.1.1'
```

25.4 DHCP forwarding over IPSec

DHCP messages are forwarded over the WAN interface using the IP address of the WAN interface as the source IP for the transmitted packet. This means that when forwarding over an IPSec tunnel a source NAT firewall rule is required to change the source IP to match an IPSec connection rule.

25.4.1 Configuration packages used

Package	Sections
firewall	redirect

25.4.2 Configuring source NAT for DHCP forwarding over IPsec

To enter a source NAT rule, browse to **Network -> Firewall.** Select **Traffic Rules** tab. The Firewall - Traffic Rules page appears. Configure a source NAT rule that changes the source IP for UDP destination port 67 from the required LAN.

For more information on configuring a source NAT rule, read the ‘Configuring Firewall’ section of the User Manual.

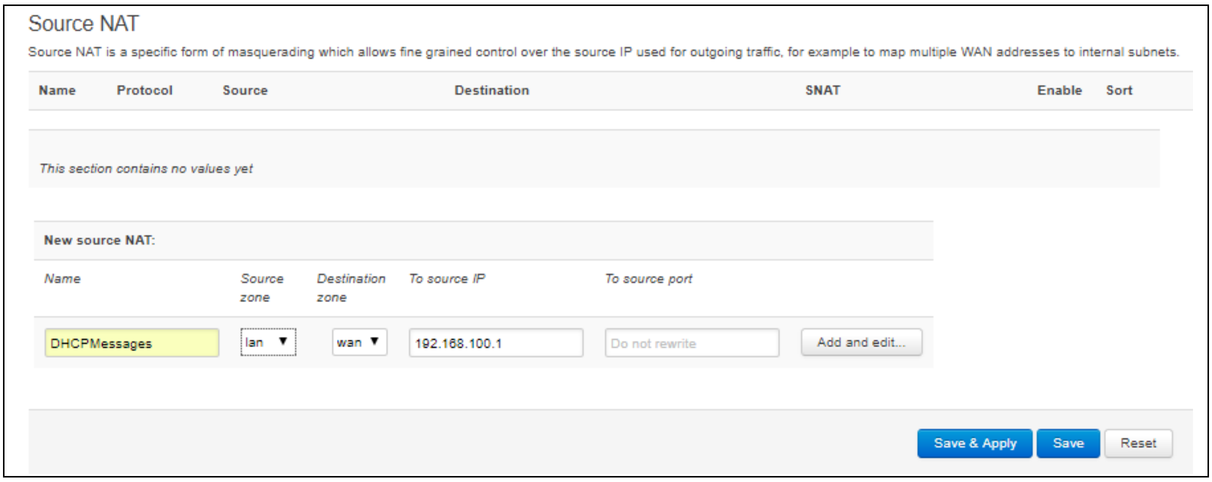


Figure 126: The firewall – traffic rules configuration page

Web Field / UCI / Package Option	Description
Web: Name UCI: firewall.@redirect[X].name Opt: name	Defines a name for the source NAT rule.
Web: Source Zone UCI: firewall.@redirect[X].src Opt: src	Defines the source interface for the source NAT rule. Select **the interface where the DHCP requests are originating**.
Web: Destination Zone UCI: firewall.@redirect[X].dest Opt: dest	Defines destination interface for the source NAT rule. Select **the interface where the DHCP requests are intended to be transmitted**.
Web: To source IP UCI: firewall.@redirect[X].src_dip Opt: src_dip	Defines the IP address to rewrite matched traffic souce IP. Select **the source IP address to match the required IPSec rule**.
Web: To source port UCI: firewall.@redirect[X].src_dport Opt: src_dport	Defines the port number to rewrite matched traffic souce port number. **Leave empty**.

Table 77: Information table for the souce NAT configuration

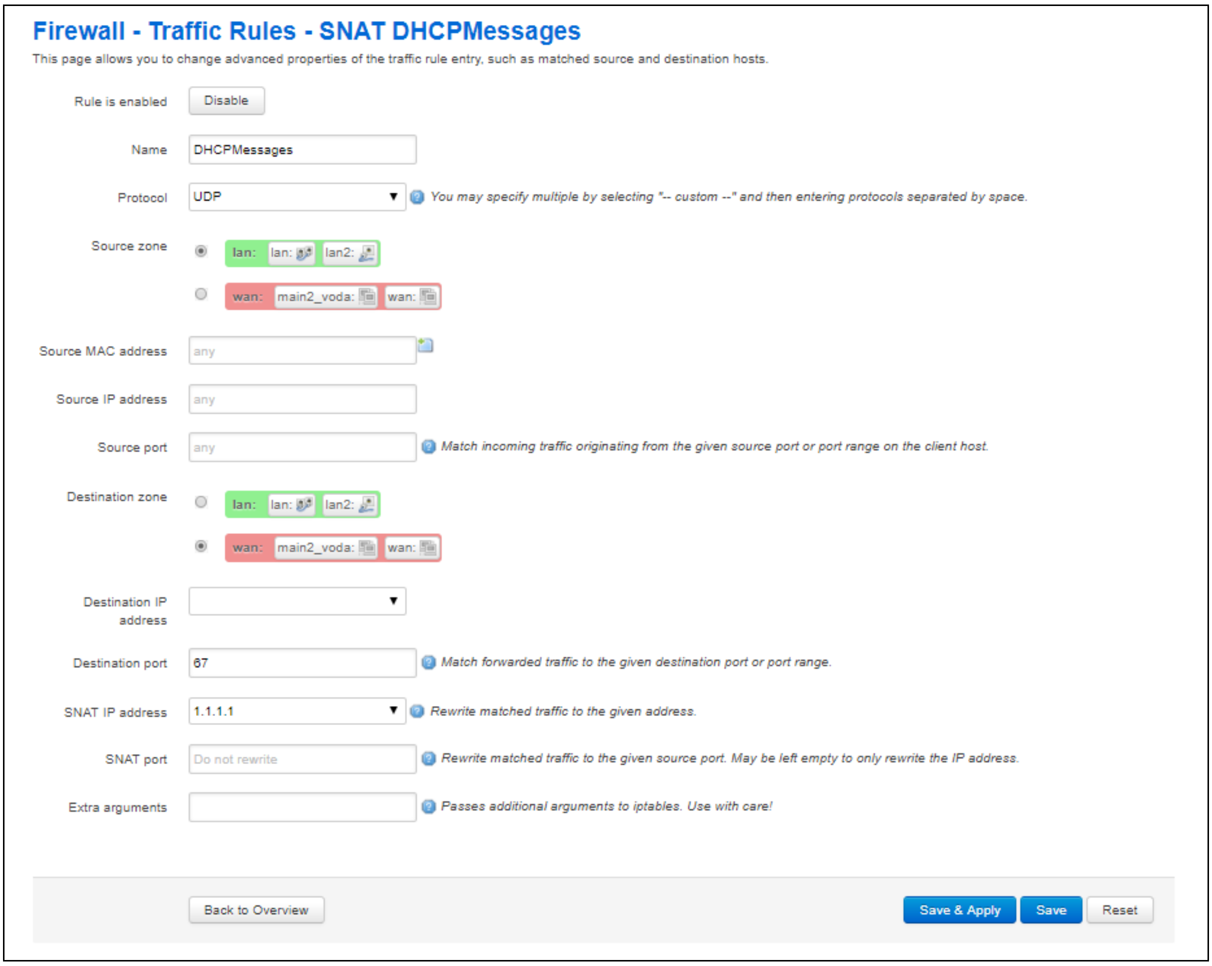


Figure 127: The firewall – traffic rules – SNAT configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled
UCI: firewall.@redirect[X].enabled
Opt: enabled</td><td>Defines whether source NAT rule is enabled.<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Name
UCI: firewall.@redirect[X].name
Opt: name</td><td>Defines a name for the source NAT rule.</td></tr>
<tr><td>Web: Protocol
UCI: firewall.@redirect[X].proto
Opt: proto</td><td>Defines the protocol for the souce NAT rule to match.
Select UDP.<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>All protocols</td><td>Match all protocols</td><td>all</td></tr><tr><td>TCP+UDP</td><td>Match TCP and UDP protocols</td><td>tcp upd</td></tr><tr><td>TCP</td><td>Match TCP protocol</td><td>tcp</td></tr><tr><td>UDP</td><td>Match UDP protocol</td><td>udp</td></tr><tr><td>ICMP</td><td>Match ICMP protocol</td><td>icmp</td></tr><tr><td>Custom</td><td>Enter custom protocol</td><td></td></tr></table></td></tr>
<tr><td>Web: Source Zone
UCI: firewall.@redirect[X].src
Opt: src</td><td>Defines the source interface for the source NAT rule.
Select the interface where the DHCP requests are originating.</td></tr>
<tr><td>Web: Destination Zone
UCI: firewall.@redirect[X].dest
Opt: dest</td><td>Defines destination interface for the source NAT rule.
Select the interface where the DHCP requests are intended to be transmitted.</td></tr>
<tr><td>Web: Destination port
UCI: firewall.@redirect[X].port
Opt: port</td><td>Defines the destination port number to match.
Select 67.</td></tr>
<tr><td>Web: SNAT IP address
UCI: firewall.@redirect[X].src_dip
Opt: src_dip</td><td>Defines the IP address to rewrite matched traffic.
Select the source IP address to match the required IPSec rule.</td></tr>
</table>

Table 78: Information table for the advanced source NAT configuration

25.4.3 Configuring source NAT for DHCP forwarding over IPSec using command line

25.4.3.1 Source NAT for DHCP forwarding over IPSec using UCI

```
root@VA_router:~# uci show firewall
……
firewall.@redirect[0]=redirect
firewall.@redirect[0].target=SNAT
firewall.@redirect[0].src=lan
firewall.@redirect[0].dest=wan
firewall.@redirect[0].src_dip=192.168.100.1
firewall.@redirect[0].name=DHCPMessages
firewall.@redirect[0].proto=udp
firewall.@redirect[0].dest_port=67
```

25.4.3.2 Source NAT for DHCP forwarding over IPSec using package options

```
root@VA_router:~# uci export firewall
package firewall
……
config redirect
        option target 'SNAT'
        option src 'lan'
        option dest 'wan'
        option src_dip '192.168.100.1'
        option name 'DHCPMessages'
        option proto 'udp'
        option dest_port '67'
```

25.5 DHCP forwarding diagnostics

25.5.1 Tracing DHCP packets

To trace DHCP packets on any interface on the router, enter **tcpdump -i any -n -p port 67 &**

```
root@VA_router:~# tcpdump -i any -n -p port 67 &
root@VA_router:~# tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535 bytes
16:39:20.666070 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:e0:c8:13:02:3d, length 360
16:39:20.666166 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:e0:c8:13:02:3d, length 360
```

To trace stop tracing enter **fg** (to bring tracing task to foregraound), and then **<CTRL-C>** to stop the trace.

```
root@VA_router:~# fg
tcpdump -i any -n -p port 67
^C
33 packets captured
33 packets received by filter
0 packets dropped by kernel
```

```
16:39:20.666166 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request
from 00:e0:c8:13:02:3d, length 360
```

25.5.2 ARP table status

To show the current ARP table of the router, enter **arp**

```
root@VA_router:~# arp
? (10.67.253.141) at 30:30:41:30:43:36 [ether]  on eth8
? (10.47.48.1) at 0a:44:b2:06 [ether]  on gre-gre1
```

26 Configuring Dynamic DNS

26.1 Overview

Dynamic DNS (DDNS) functionality on a Virtual Access router will dynamically perform DDNS updates to a server so it can associate an IP address with a correctly associated DNS name. Users can then contact a machine, router, device and so on with a DNS name rather than a dynamic IP address.

An account is required with the provider, and one or more domain names are associated with that account. A dynamic DNS client on the router monitors the public IP address associated with an interface and whenever the IP address changes, the client notifies the DNS provider to update the corresponding domain name.

When the DNS provider responds to queries for the domain name, it sets a low lifetime, typically a minute or two at most, on the response so that it is not cached. Updates to the domain name are thus visible throughout the whole Internet with little delay.

Note: most providers impose restrictions on how updates are handled: updating when no change of address occurred is considered abusive and may result in an account being blocked. Sometimes, addresses must be refreshed periodically, for example, once a month, to show that they are still in active use.

26.2 Configuration packages used

Package	Sections
ddns	service

26.3 Configuring Dynamic DNS using the web interface

In the top menu, select **Services -> Dynamic DNS**. The Dynamic DNS Configuration page appears.

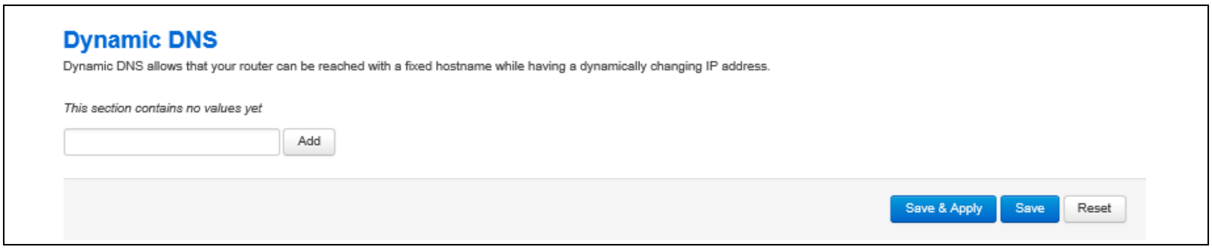


Figure 128: The Dynamic DNS configuration page

Enter a text name that will be used for the dynamic DNS section in the configuration. Select **Add**. The Dynamic DNS configuration options appear.

26.3.1 Dynamic DNS settings

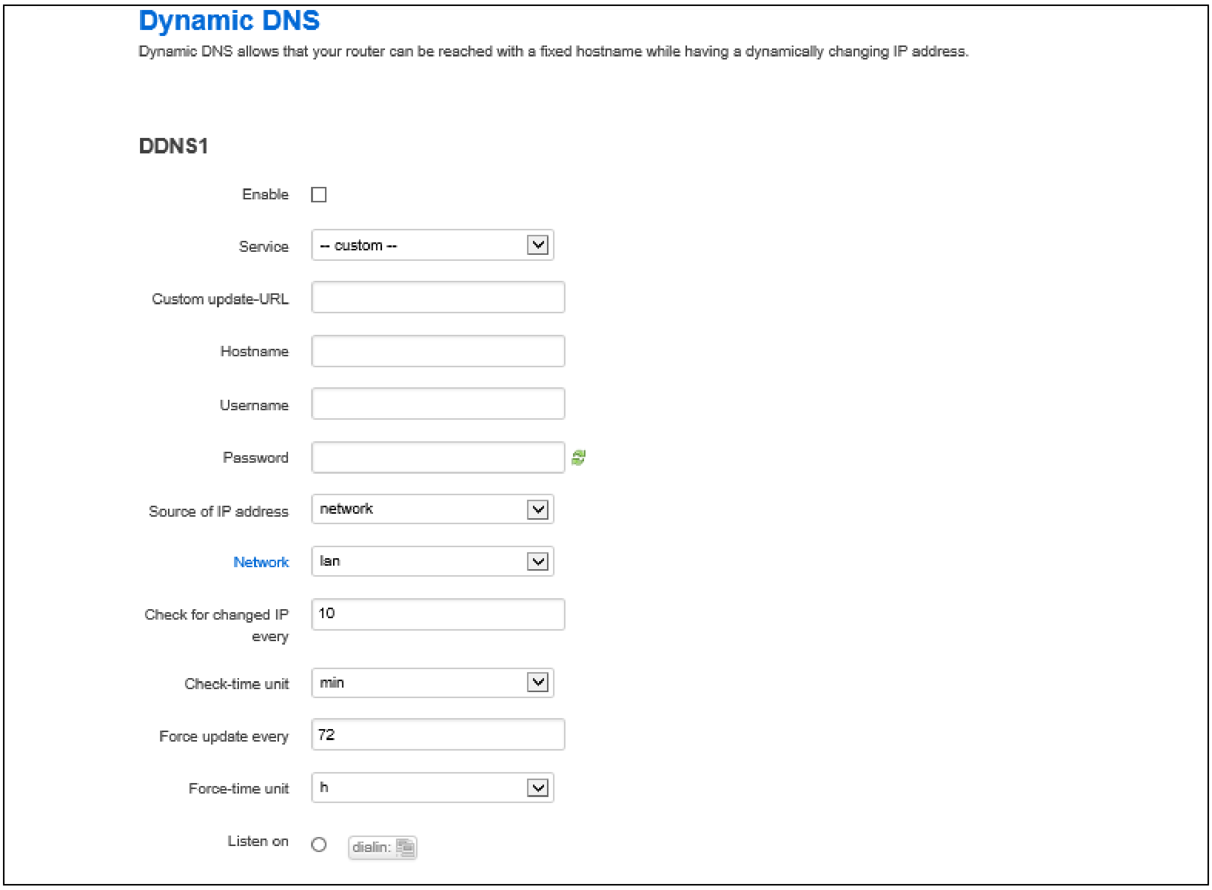


Figure 129: The dynamic DNS main settings page

Web Field / UCI / Package Option	Description
Web: Enable UCI: ddns.<name>.enabled Opt: enabled	Enables a Dynamic DNS entry on the router. 0: Disabled. 1: Enabled
Web: Service UCI: ddns.<name>.service_name Opt: service_name	Defines the Dynamic DNS provider.
Web: Customer update-URL UCI: ddns.<name>.update_url Opt: update_url	Defines the customer DNS provider. Displayed when the service is set to custom in the web interface.
Web: Hostname UCI: ddns.<name>.domain Opt: domain	Defines the fully qualified domain name associated with this entry. This is the name to update with the new IP address as needed.
Web: Username UCI: ddns.<name>.username Opt: username	Defines the user name to use for authenticating domain updates with the selected provider.
Web: Password UCI: ddns.<name>.password Opt: password	Defines the password to use for authenticating domain name updates with the selected provider.
Web: Source of IP address UCI: ddns.<name>.ip_source Opt: ip_source	Defines the type of interface whose IP needs to be updated. network: IP is a associated with a network configuration. interface: IP is associated with an interface. web: IP is associated with a URL.

Web: Network UCI: ddns.<name>.ip_network Opt: ip_network	Defines the network whose IP needs to be updated. Displayed when the Source of IP address option is set to network. All the configured network interfaces will be shown.
Web: Inteface UCI: ddns.<name>.ip_interface Opt: ip_interface	Defines the interface whose IP needs to be updated. Displayed when the Source of IP address option is set to interface. All the configured interfaces will be shown.
Web: URL UCI: ddns.<name>.ip_url Opt: ip_url	Defines the URL where the IP downloaded from. Displayed when the Source of IP address option is set to URL.
Web: Check for changed IP every UCI: ddns.<name>.check_interval Opt: check_interval	Defines how often to check for an IP change. Used in conjunction with check_unit. 10 Range
Web: Check-time unit UCI: ddns.<name>.check_unit Opt: check_unit	Defines the time unit to use for check for an IP change. Used in conjunction with check_interval. Minutes hours
Web: Force update every UCI: ddns.<name>.force_interval Opt: force_interval	Defines how often to force an IP update to the provider. Used in conjunction with force_unit. 72 — Disabled. Range — Enabled
Web: Force-time unit UCI: ddns.<name>.force_unit Opt: force_unit	Defines the time unit to use for check for an IP change. Used in conjunction with force_interval. Minutes Hours
Web: Listen on UCI: ddns.<name>.interface Opt: interface	Defines the interface for ddns monitoring. Typically this will be the same as the interface whose IP is being updated – as defined ip_network or ip_interface. All configured interfaces will be displayed.

Table 79: Information table for dynamic DNS settings

26.4 Dynamic DNS using UCI

Dynamic DNS uses the ddns package **/etc/config/ddns**

26.4.1 UCI commands for DDNS

```
root@VA_router:~# uci show ddns
ddns.ddns1=service
ddns.ddns1.enabled=1
ddns.ddns1.service_name=dyndns.org
ddns.ddns1.domain=fqdn_of_interface
ddns.ddns1.username=testusername
ddns.ddns1.password=testpassword
ddns.ddns1.ip_source=network
ddns.ddns1.ip_network=dsl0
ddns.ddns1.check_interval=10
```

```
ddns.ddns1.check_unit=minutes
ddns.ddns1.force_interval=72
ddns.ddns1.force_unit=hours
ddns.ddns1.interface=dsl0
Package options for DDNS
root@VA_router:~# uci export ddns
package ddns

config service 'ddns1'
        option enabled '1'
        option service_name 'dyndns.org'
        option domain 'fqdn_of_interface'
        option username 'test'
        option password 'test'
        option ip_source 'network'
        option ip_network 'dsl0'
        option check_interval '10'
        option check_unit 'minutes'
        option force_interval '72'
        option force_unit 'hours'
        option interface 'dsl0'
```

27 Configuring hostnames

27.1 Overview

Hostnames are human-readable names that identify a device connected to a network. There are several different ways in which hostnames can be configured and used on the router.

- Local host file records
- PTR records
- Static DHCP leases

27.2 Local host file records

The hosts file is an operating system file that maps hostnames to IP addresses. It is used preferentially to other name resolution methods such as DNS.

The hosts file contains lines of text consisting of an IP address in the first text field followed by one or more host names. Each field is separated by white space – tabs are often preferred for historical reasons, but spaces are also used. Comment lines may be included; they are indicated by an octothorpe (#) in the first position of such lines. Entirely blank lines in the file are ignored.

By default, the routers local host file contains:

```
127.0.0.1 localhost
::1 ip6-localhost ip6-loopback
```

The local host file is stored at **/etc/hosts**

27.2.1 Configuration packages used

Package	Sections
network	host

27.2.2 Configuring local host files entries using the web interface

In the top menu, select **Network -> Interfaces**. The Interfaces configuration page appears.

Browse to **Host Records** section at the bottom of the page.

Figure 130: The host records add page

Select **Add**. Enter a hostname and IP address and select **Save & Apply**.

Figure 131: The host records configuration page

Web Field/UCI/Package Option	Description
Web: Hostname UCI: network.host.hostname Opt: hostname	Defines the hostname.
Web: IP-Address UCI: network.host.addr Opt: addr	Defines the IP address associated with the hostname.

Table 80: Information table for host records settings

27.2.3 Local host records using command line

Local host records are configured in the host section of the network package **/etc/config/network.**

Multiple hosts can be configured.

By default, all host instances are named host and are identified by @host then the host position in the package as a number. For example, for the first host in the package using UCI:

```
network.@host[0]=host
network.@host[0].hostname=Device1
```

Or using package options:

```
config host
        option hostname 'Device1'
```

27.2.3.1 Local host records using uci

```
root@VA_router:~# uci show network
……
network.@host[0]=host
network.@host[0].hostname=Device1
network.@host[0].addr=1.1.1.1
```

27.2.3.2 Local host records using package option

```
root@VA_router:~# uci export network
package network
……
config host
        option hostname 'Device1'
        option addr '1.1.1.1'
```

27.2.4 Local host records diagnostics

27.2.4.1 Hosts file

Local host records are written to the local hosts file stored at **/etc/hosts**. To view the local hosts file, enter:

```
root@VA_router:~# cat /etc/hosts
127.0.0.1 localhost
::1 ip6-localhost ip6-loopback
1.1.1.1 Device1
```

27.3 PTR records

PTR records are used for reverse DNS.

The primary purpose for DNS is to map domains to IP addresses. A pointer record works in the opposite way; it associates an IP address with a domain name.

27.3.1 Configuration packages used

Package	Sections
dhcp	domain

27.3.2 Configuring PTR records using the web interface

In the top menu, select **Network -> Hostnames**. The Hostnames configuration page appears.

Figure 132: The hostnames add page

Select **Add**. Enter a hostname and IP address for the PTR record and select **Save & Apply**.

Figure 133: The hostnames configuration page

Web Field/UCI/Package Option	Description
Web: Hostname UCI: dhcp.domain.name Opt: name	Defines the domain name for the PTR record.
Web: IP-Address UCI: dhcp.domain.ip Opt: ip	Defines the IP address associated with the domain name.

Table 81: Information table for hostnames settings

27.3.3 PTR records using command line

PTR records are configured in the **domain** section of the dhcp package. **/etc/config/dhcp.**

Multiple **domains** can be configured.

By default, all domain instances are named domain and are identified by @domain then the domain position in the package as a number. For example, for the first domain in the package using UCI:

```
dhcp.@domain[0]=domain
dhcp.@domain[0].name=Domain1
```

Or using package options:

```
config domain
        option name 'Domain1'
```

27.3.3.1 PTR records using uci

```
root@VA_router:~# uci show dhcp
……
dhcp.@domain[0]=domain
dhcp.@domain[0].name=Domain1
dhcp.@domain[0].ip=2.2.2.2
```

27.3.3.2 PTR records using package option

```
root@VA_router:~# uci export dhcp
package dhcp
……
config domain
        option name 'Domain1'
        option ip '2.2.2.2'
```

27.3.4 PTR records diagnostics

27.3.4.1 PTR records table

To view PTR records, enter:

```
root@VA_router:~# pgrep -fl dnsmasq
4724 /usr/sbin/dnsmasq -K -D -y -Z -b -E -s lan -S /lan/ -l
/tmp/dhcp.leases -r /tmp/resolv.conf.auto --stop-dns-rebind --rebind-
localhost-ok -A /Device1.lan/1.1.1.1 --ptr-record=1.1.1.1.in-
addr.arpa,Device1.lan -A /Device2.lan/2.2.2.2 --ptr-record=2.2.2.2.in-
addr.arpa,Device2.lan
```

27.4 Static leases

Static leases are used to assign fixed IP addresses and symbolic hostnames to DHCP clients based on their MAC (hardware) address.

They are also required for non-dynamic interface configurations where only hosts with a corresponding lease are served.

27.4.1 Configuration packages used

Package	Sections
dhcp	host

27.4.2 Configuring static leases using the web interface

In the top menu, select **Network -> DHCP and DNS**. The DHCP and DNS configuration page appears.

Browse to **Static leases** section.

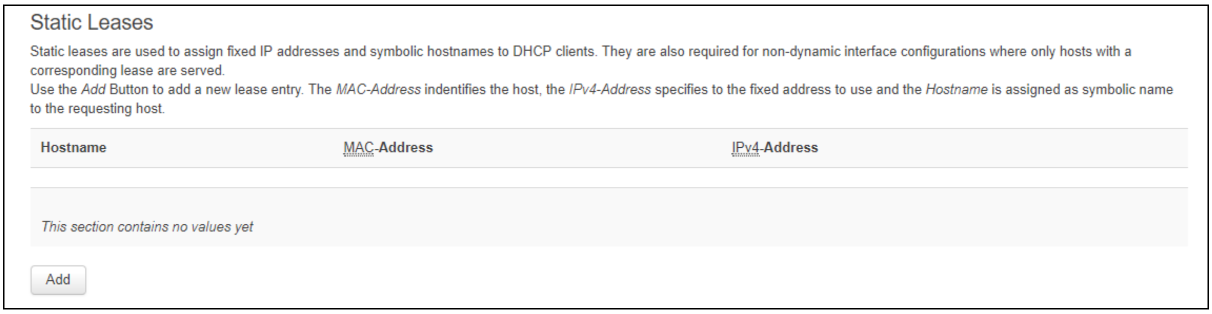


Figure 134: The static leases add page

Select **Add**. Enter a hostname, MAC address and IP address for the static lease. Select **Save & Apply**.

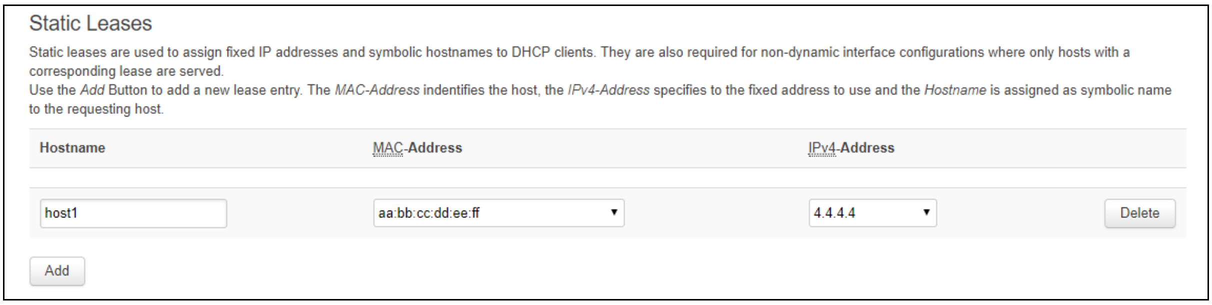


Figure 135: The static leases configuration page

Web Field/UCI/Package Option	Description
Web: Hostname UCI: dhcp.host.name Opt: name	Defines the symbolic hostname to assign.
Web: MAC-Address UCI: dhcp.host.mac Opt: mac	Defines the MAC address for this host. MAC addresses should be entered in the format `aa:bb:cc:dd:ee:ff`
Web: IPv4-Address UCI: dhcp.host.ip Opt: ip	Defines the IP address to be used for this host.

Table 82: Information table for static leases settings

27.4.3 Static leases using command line

Static leases are configured in the **host** section of the dhcp package **/etc/config/dhcp.**

Multiple **hosts** can be configured.

By default, all dhcp host instances are named host. It is identified by `@host` then the host position in the package as a number. For example, for the first host in the package using UCI:

```
dhcp.@host[0]=host
dhcp.@host[0].name=Host1
```

Or using package options:

```
config host
        option name 'Host1'
```

27.4.3.1 Static leases using uci

```
root@VA_router:~# uci show dhcp
……
dhcp.@host[0]=host
dhcp.@host[0].name=Host1
dhcp.@host[0].mac=aa:bb:cc:dd:ee:ff
dhcp.@host[0].ip=4.4.4.4
```

27.4.3.2 Static leases using package option

```
root@VA_router:~# uci export dhcp
package dhcp
……
config host
        option name 'Host1'
        option mac 'aa:bb:cc:dd:ee:ff'
        option ip '4.4.4.4'
```

28 Configuring firewall

The firewall itself is not required. It is a set of scripts which configure Netfilter. If preferred, you can use Netfilter directly to achieve the desired firewall behaviour.

Note: the UCI firewall exists to simplify the configuration of Netfilter for many scenarios, without requiring the knowledge to deal with the complexity of Netfilter.

The firewall configuration consists of several zones covering one or more interfaces. Permitted traffic flow between the zones is controlled by forwardings. Each zone can include multiple rules and redirects (port forwarding rules).

The Netfilter system is a chained processing filter where packets pass through various rules. The first rule that matches is executed often leading to another rule-chain until a packet hits either ACCEPT or DROP/REJECT.

Accepted packets pass through the firewall. Dropped packets are prohibited from passing. Rejected packets are also prohibited but an ICMP message is returned to the source host.

A minimal firewall configuration for a router usually consists of one 'defaults' section, at least two 'zones' (LAN and WAN) and one forwarding to allow traffic from LAN to WAN. Other sections that exist are 'redirects', 'rules' and 'includes'.

28.1 Configuration package used

Package	Sections
firewall	

28.2 Configuring firewall using the web interface

In the top menu, select **Network -> Firewall**. The Firewall page appears. It is divided into four sections:

Section	Description
General Zone Settings	Defines the firewall zones, both global and specific.
Port Forwards	Port Forwards are also known as Redirects. This section creates the redirects using DNAT (Destination Network Address Translation) with Netfilter.
Traffic Rules	Defines rules to allow or restrict access to specific ports, hosts or protocols.

28.2.1 Firewall: zone settings

The Zone settings section is divided into two:

Section	Description
General Settings	Defines the global firewall settings that do not belong to any specific zones.
Zones	The zones section groups one or more interfaces and serves as a source or destination for forwardings, rules and redirects. Masquerading (NAT) of outgoing traffic is controlled on a per-zone basis.

28.2.1.1 Firewall general settings

The General Settings page, or defaults section declares global firewall settings that do not belong to any specific zones. These default rules take effect last and more specific rules take effect first.

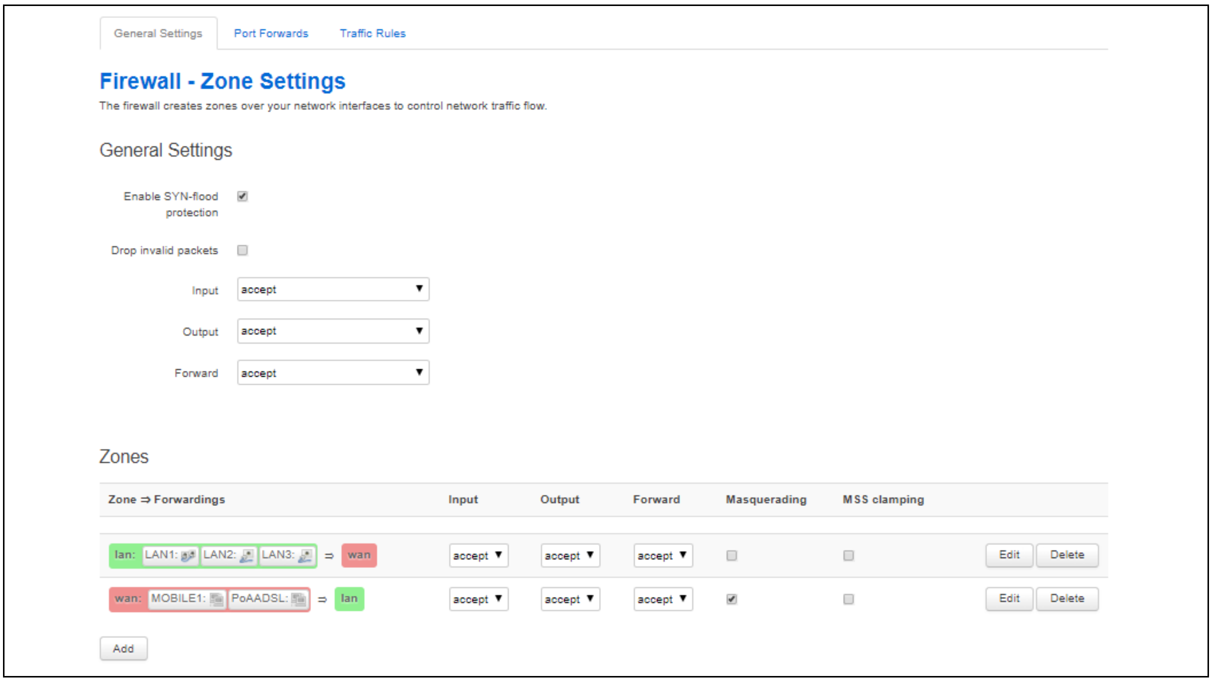


Figure 136: The firewall zone general settings page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enable SYN-flood protection
UCI: firewall.defaults.syn_flood
Opt: syn_flood</td><td>Enables SYN flood protection.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Drop invalid packets
UCI: firewall.defaults.drop_invalid
Opt: drop_invalid</td><td>Drops packets not matching any active connection.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Input
UCI: firewall.defaults.input
Opt: input</td><td>Default policy for the Input chain.<table><tr><td>Accept</td><td>Accepted packets pass through the firewall.</td></tr><tr><td>Reject</td><td>Rejected packets are blocked by the firewall and ICMP message is returned to the source host.</td></tr><tr><td>Drop</td><td>Dropped packets are blocked by the firewall.</td></tr></table></td></tr>
<tr><td>Web: Output
UCI: firewall.defaults.output
Opt: output</td><td>Default policy for the Output chain.<table><tr><td>Accept</td><td>Accepted packets pass through the firewall.</td></tr><tr><td>Reject</td><td>Rejected packets are blocked by the firewall and ICMP message is returned to the source host.</td></tr><tr><td>Drop</td><td>Dropped packets are blocked by the firewall.</td></tr></table></td></tr>
<tr><td>Web: Forward
UCI: firewall.defaults.forward
Opt: forward</td><td>Default policy for the Forward chain.<table><tr><td>Accept</td><td>Accepted packets pass through the firewall.</td></tr><tr><td>Reject</td><td>Rejected packets are blocked by the firewall and ICMP message is returned to the source host.</td></tr><tr><td>Drop</td><td>Dropped packets are blocked by the firewall.</td></tr></table></td></tr>
</table>

Table 83: Information table for general zone general settings page

28.2.1.2 Firewall zones

The Zones section groups one or more interfaces and serves as a source or destination for forwardings, rules and redirects. Masquerading (NAT) of outgoing traffic is controlled on a per-zone basis. To view a zone's settings, click **Edit**.

The number of concurrent dynamic/static NAT entries of any kind (NAT/PAT/DNAT/SNAT) is not limited in any way by software; the only hardware limitation is the amount of RAM installed on the device.

28.2.1.3 Firewall zone: general settings

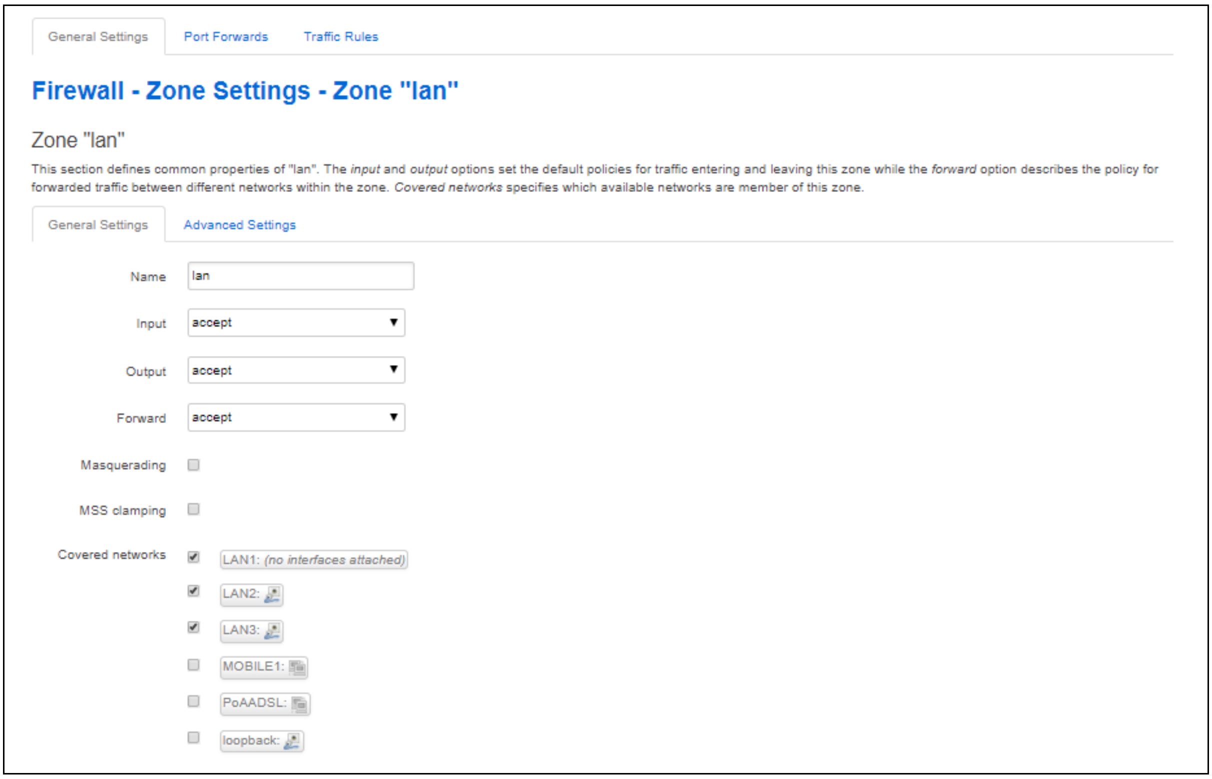


Figure 137: The firewall zone general settings

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: name
UCI: firewall.<zone label>.name
Opt: name</td><td>Sets the unique zone name. Maximum of 11 characters allowed.
Note: the zone label is obtained by using the 'uci show firewall' command and is of the format '@zone[x]' where x is an integer starting at 0.</td></tr>
<tr><td>Web: Input
UCI: firewall.<zone label>.input
Opt: input</td><td>Default policy for incoming zone traffic. Incoming traffic is traffic entering the router through an interface selected in the 'Covered Networks' option for this zone.<table><tr><td>Accept</td><td>Accepted packets pass through the firewall.</td></tr><tr><td>Reject</td><td>Rejected packets are blocked by the firewall and ICMP message is returned to the source host.</td></tr><tr><td>Drop</td><td>Dropped packets are blocked by the firewall.</td></tr></table></td></tr>
</table>

<table>
<tr><td>Web: Output
UCI: firewall.<zone label>.output
Opt: output</td><td>Default policy for outgoing zone traffic. Outgoing traffic is traffic leaving the router through an interface selected in the 'Covered Networks' option for this zone.<table><tr><td>Accept</td><td>Accepted packets pass through the firewall.</td></tr><tr><td>Reject</td><td>Rejected packets are blocked by the firewall and ICMP message is returned to the source host.</td></tr><tr><td>Drop</td><td>Dropped packets are blocked by the firewall.</td></tr></table></td></tr>
<tr><td>Web: Forward
UCI: firewall.<zone label>.forward
Opt: forward</td><td>Default policy for internal zone traffic between interfaces. Forward rules for a zone describe what happens to traffic passing between different interfaces within that zone.<table><tr><td>Accept</td><td>Accepted packets pass through the firewall.</td></tr><tr><td>Reject</td><td>Rejected packets are blocked by the firewall and ICMP message is returned to the source host.</td></tr><tr><td>Drop</td><td>Dropped packets are blocked by the firewall.</td></tr></table></td></tr>
<tr><td>Web: Masquerading
UCI: firewall.<zone label>.masq
Opt: masq</td><td>Specifies whether outgoing zone traffic should be masqueraded (NATTED). This is typically enabled on the wan zone.</td></tr>
<tr><td>Web: MSS Clamping
UCI: firewall.<zone label>.mtu_fix
Opt: mtu_fix</td><td>Enables MSS clamping for outgoing zone traffic. Subnets are allowed.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Covered networks
UCI: firewall.<zone label>.network
Opt: network</td><td>Defines a list of interfaces attached to this zone, if omitted, the value of name is used by default.
Note: use the uci list syntax to edit this setting through UCI.</td></tr>
</table>

Table 84: Information table for firewall zone general settings

28.2.1.4 Firewall zone: advanced settings

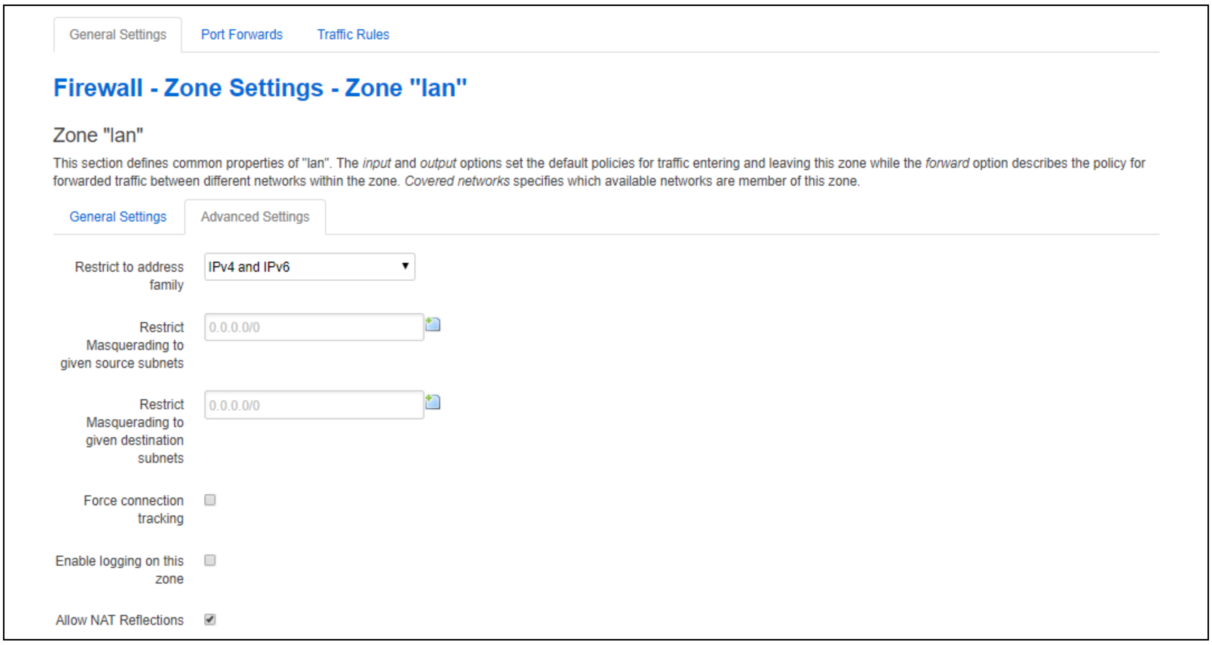


Figure 138: Firewall zone advanced settings

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Restrict to address family
UCI: firewall.<zone label>.family
Opt: family</td><td>Restricts zone to IPv4, IPv6 or both IPv4 and IPv6.<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>IPv4 and IPv6</td><td>Any address family</td><td>any</td></tr><tr><td>IPv4 only</td><td>IPv4 only</td><td>ipv4</td></tr><tr><td>IPv6 only</td><td>IPv6 only</td><td>Ipv6</td></tr></table></td></tr>
<tr><td>Web: Restrict Masquerading to given source subnets.
UCI: firewall.<zone label>.masq_src
Opt: masq_src</td><td>Limits masquerading to the given source subnets. Negation is possible by prefixing the subnet with ‘!’. Multiple subnets are allowed.</td></tr>
<tr><td>Web: Restrict Masquerading to given destination subnets.
UCI: firewall.<zone label>.masq_dest
Opt: masq_dest</td><td>Limits masquerading to the given destination subnets. Negation is possible by prefixing the subnet with ‘!’. Multiple subnets are allowed. Multiple IP addresses/subnets should be separated by a space, for example: option masq_dest ‘1.1.1.1 2.2.2.0/24’.</td></tr>
<tr><td>Web: Force connection tracking
UCI: firewall.<zone label>.conntrack
Opt: conntrack</td><td>Forces connection tracking for this zone.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>If masquerading is used. Otherwise, default is 0.</td></tr></table></td></tr>
<tr><td>Web: Enable logging on this zone
UCI: firewall.<zone label>.log
Opt: log</td><td>Creates log rules for rejected and dropped traffic in this zone.</td></tr>
<tr><td>Web: Allow NAT reflections
UCI: firewall.<zone label>.reflection
Opt: reflection</td><td>Enable/disable all NAT reflections for this zone.
Note: For configs with a large number of firewall rules, disabling NAT reflection will speed up load of firewall rules on interface start.<table><tr><td>0</td><td>Disable reflection.</td></tr><tr><td>1</td><td>Enable reflection.</td></tr></table></td></tr>
</table>

Web: n/a UCI: firewall.<zone label>.log_limit Opt: log_limit	Limits the amount of log messages per interval.

Table 85: Information table for firewall zone advanced settings

28.2.1.5 Inter-zone forwarding

This section controls the traffic flow between zones. Selecting a source or destination zone generates a Forwarding rule. Only one direction is covered by any forwarding rule. Hence for bidirectional traffic flow between two zones then two rules are required, with source and destination alternated.

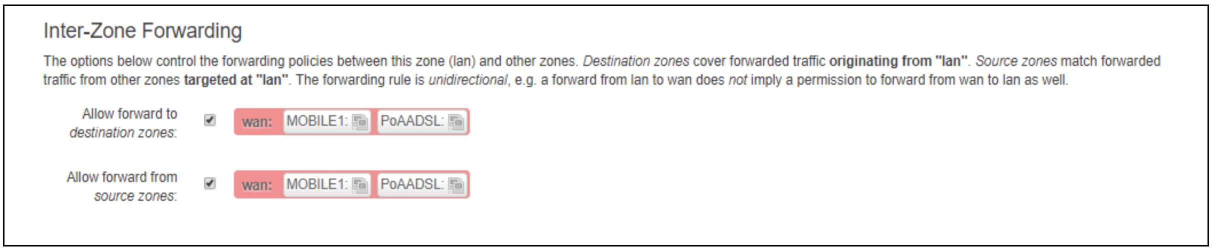


Figure 139: The inter-zone forwarding section

Web Field / UCI / Package Option	Description
Web: Allow forward to destination zones UCI: firewall.<forwarding label>.dest Opt: dest	Allows forward to other zones. Enter the current zone as the source. Enabling this option puts two entries into the firewall file: destination and source.
UCI firewall.<forwarding label>.src Opt: src	
Web: Allow forward from source zones UCI: firewall.<forwarding label>.dest Opt: dest	Allows forward from other zones. Enter the current zone as the destination. Enabling this option puts two entries into the firewall file: destination and source.
UCI: firewall.<forwarding label>.src Opt: src	

Table 86: Information table for inter-zone forwarding settings

Note: the rules generated for forwarding traffic between zones relay connection tracking to be enabled on at least one of the source or destination zones. This can be enabled through the conntrack option or through masq.

28.2.2 Firewall port forwards

Port Forwards are also known as Redirects. This section creates the redirects using DNAT (Destination Network Address Translation) with Netfilter. The redirects are from the firewall zone labelled as wan to the firewall zone labelled as lan. These zones can refer to multiple external and internal interfaces as defined in the Firewall Zone settings.

To edit an existing port forward select **edit**.

To add a new port forward select **add**.

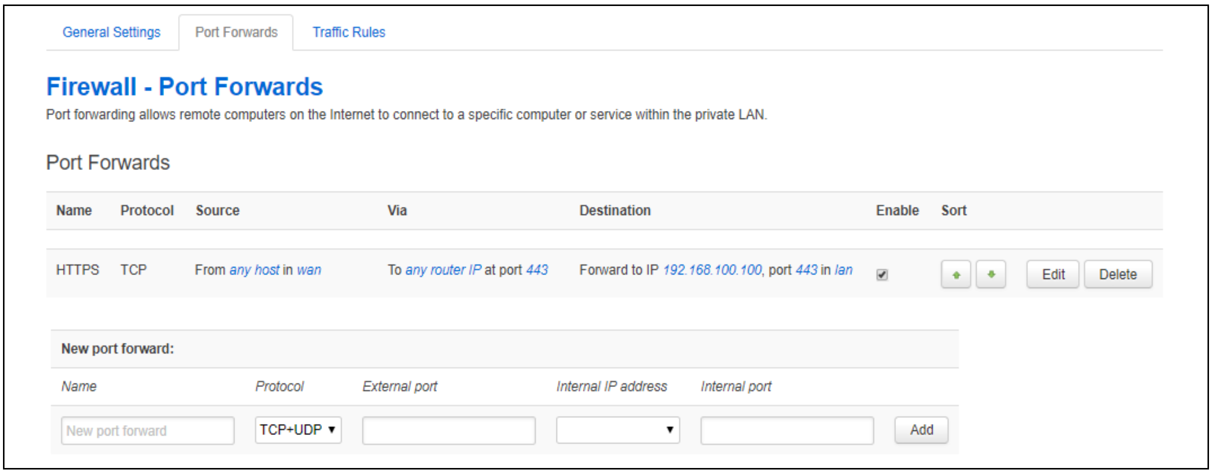


Figure 140: The firewall port forward page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: name
UCI: firewall.<redirect label>.name
Opt: name</td><td>Sets the port forwarding name. For Web UI generated redirects the <redirect label> takes the form of @redirect[x], where x is an integer starting from 0.</td></tr>
<tr><td>Web: Protocol
UCI: firewall.<redirect label>.proto
Opt: proto</td><td>Defines layer 4 protocol to match incoming traffic.
<table>
<tr><th>Option</th><th>Description</th><th>UCI</th></tr>
<tr><td>tcp+udp</td><td>Match either TCP or UDP packets.</td><td>tcp udp</td></tr>
<tr><td>tcp</td><td>Match TCP packets only.</td><td>tcp</td></tr>
<tr><td>udp</td><td>Match UDP packets only.</td><td>udp</td></tr>
</table></td></tr>
<tr><td>Web: External port
UCI: firewall.<redirect label>.src_dport
Opt: src_dport</td><td>Specifies the incoming TCP/UDP port or port range to match. This is the incoming destination port specified by the external host. Port ranges specified as start:stop, for example, 2001:2020.
<table>
<tr><td>Blank</td><td>Match traffic to any port.</td></tr>
<tr><td>Range</td><td>1 - 65535</td></tr>
</table></td></tr>
<tr><td>Web: Internal IP address
UCI: firewall.<redirect label>.dest_ip
Opt: dest_ip</td><td>Specifies the internal (LAN) IP address for the traffic to be redirected to.</td></tr>
<tr><td>Web: Internal port
UCI: firewall.<redirect label>.dest_port
Opt: dest_port</td><td>Specifies the destination tcp/udp port for the redirect traffic.</td></tr>
</table>

Table 87: Information table for firewall port forward settings

The defined redirects can be sorted into a specific order to be applied. More specific rules should be placed first.

After the redirect is created and saved, to make changes, click **Edit**. This will provide further options to change the source/destination zones; specify source mac addresses and enable NAT loopback (reflection).

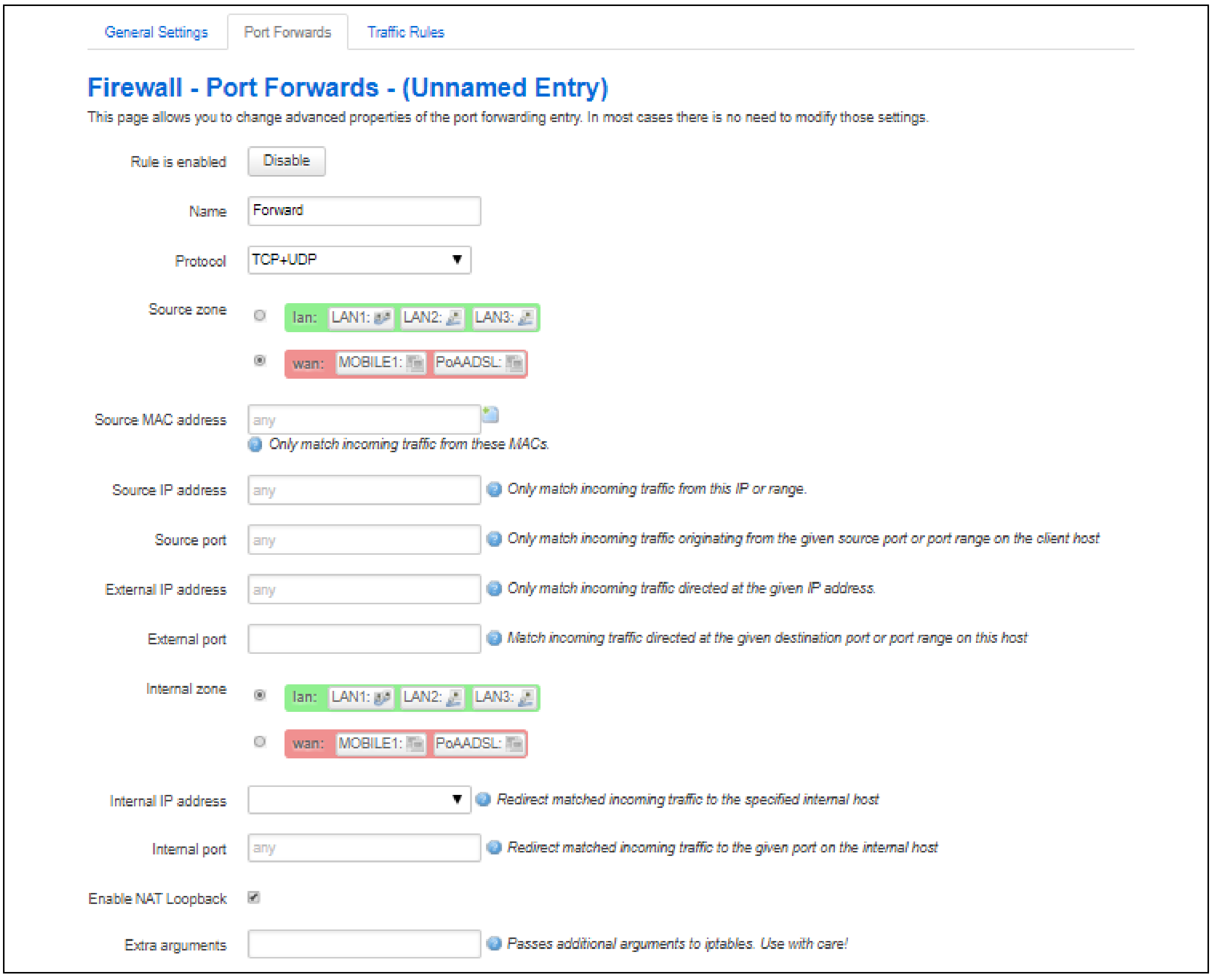


Figure 141: The firewall port forwards edits page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled
UCI: firewall.<redirect label>.enabled
Opt: enabled</td><td>Specifies if this redirect should be enabled or disabled.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: name
UCI: firewall.<redirect label>.name
Opt: name</td><td>Sets the port forwarding name. For Web UI generated redirects the <redirect label> takes the form of @redirect[x], where x is an integer starting from 0.</td></tr>
<tr><td>Web: Protocol
UCI: firewall.<redirect label>.proto
Opt: proto</td><td>Defines layer 4 protocol to match incoming traffic.<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>tcp+udp</td><td>Match either TCP or UDP packets.</td><td>tcp udp</td></tr><tr><td>tcp</td><td>Match TCP packets only.</td><td>tcp</td></tr><tr><td>udp</td><td>Match UDP packets only.</td><td>udp</td></tr></table></td></tr>
<tr><td>Web: Source zone
UCI: firewall.<redirect label>.src
Opt: src</td><td>Specifies the traffic source zone. It must refer to one of the defined zone names. When using the web interface, this is set to WAN initially.</td></tr>
</table>

Web: Source MAC address
UCI: firewall.<redirect label>.src_mac
Opt: list src_mac
	Defines the list of source MAC addresses that this redirect will match.
	Format: aa:bb:cc:dd:ee:ff
	Multiple RIP interfaces are entered using `uci set` and `uci add_list` commands. Example:
	`uci set firewall.@redirect[0].src_mac=aa:bb:cc:dd:ee:ff`
	`uci add_list firewall.@redirect[0].src_mac=12:34:56:78:90:12`
	or using a list of options via package options
	`list network ‘aa:bb:cc:dd:ee:ff’`
	`list network ‘12:34:56:78:90:12’`

Web: Source IP address UCI: firewall.<redirect label>.src_ip Opt: src_ip	Defines a source IP address that this redirect will match. Blank — Match traffic from any source IP. Range — A.B.C.D/mask.
Web: Source port UCI: firewall.<redirect label>.src_port Opt: src_port	Defines a source IP port that this redirect will match. You can enter multiple ports, using a space separator. For example: option src_port ’22 23’ *see note below on use with options src_dport and dest_port Blank — Match traffic from any source port. Range — 1 - 65535
Web: External port UCI: firewall.<redirect label>.src_dport Opt: src_dport	Specifies the incoming TCP/UDP port or port range to match. This is the incoming destination port specified by the external host. Port ranges specified in format start:stop, for example, 2001:2020. You can enter multiple ports, using a space separator. For example: option src_dport ’22 23’ *see note below on use with options src_port and dest_port Blank — Match traffic to any port. Range — 1 – 65535
Web: Internal zone UCI: firewall.<redirect label>.dest Opt: dest	Specifies the traffic destination zone, must refer to one of the defined zone names.
Web: Internal IP address UCI: firewall.<redirect label>.dest_ip Opt: dest_ip	Specifies the internal (LAN) IP address for the traffic to be redirected to.
Web: Internal port UCI: firewall.<redirect label>.dest_port Opt: dest_port	Specifies the destination tcp/udp port for the redirect traffic. You can enter multiple ports, using a space separator. *For example: option dest_port ’22 23’ *See note below table on use with options src_port and src_dport.
Web: Enable NAT Loopback UCI: firewall.<redirect label>.reflection Opt: reflection	Enable or disable NAT reflection for this redirect. 0 — Reflection disabled. 1 — Reflection enabled.
Web: Extra arguments UCI: firewall.<redirect label>.extra Opt: extra	Passes extra arguments to IP tables. This is useful to specify additional match options, like -m policy --dir in for IPSec. The arguments are entered as text strings.

Table 88: Information table for port forward edits fields

***Note**: redirect rule options src_port and src_dport/dest_port accept space-separated lists of ports. If src_port is a list, then src_dport/dst_port cannot be, to avoid ambiguity.

If src_dport/dest_port are lists of different lengths, then the missing values of the shorter list default to the corresponding port in the other list. For example, if configuration file is:

```
option src_dport '21 22 23'
option dest_port '21 22 23 24'
```

then the firmware will interpret the values as:

```
option src_dport '21 22 23 24'
option dest_port '21 22 23 24'
```

28.2.3 Firewall traffic rules

Rules can be defined to allow or restrict access to specific ports, hosts or protocols.

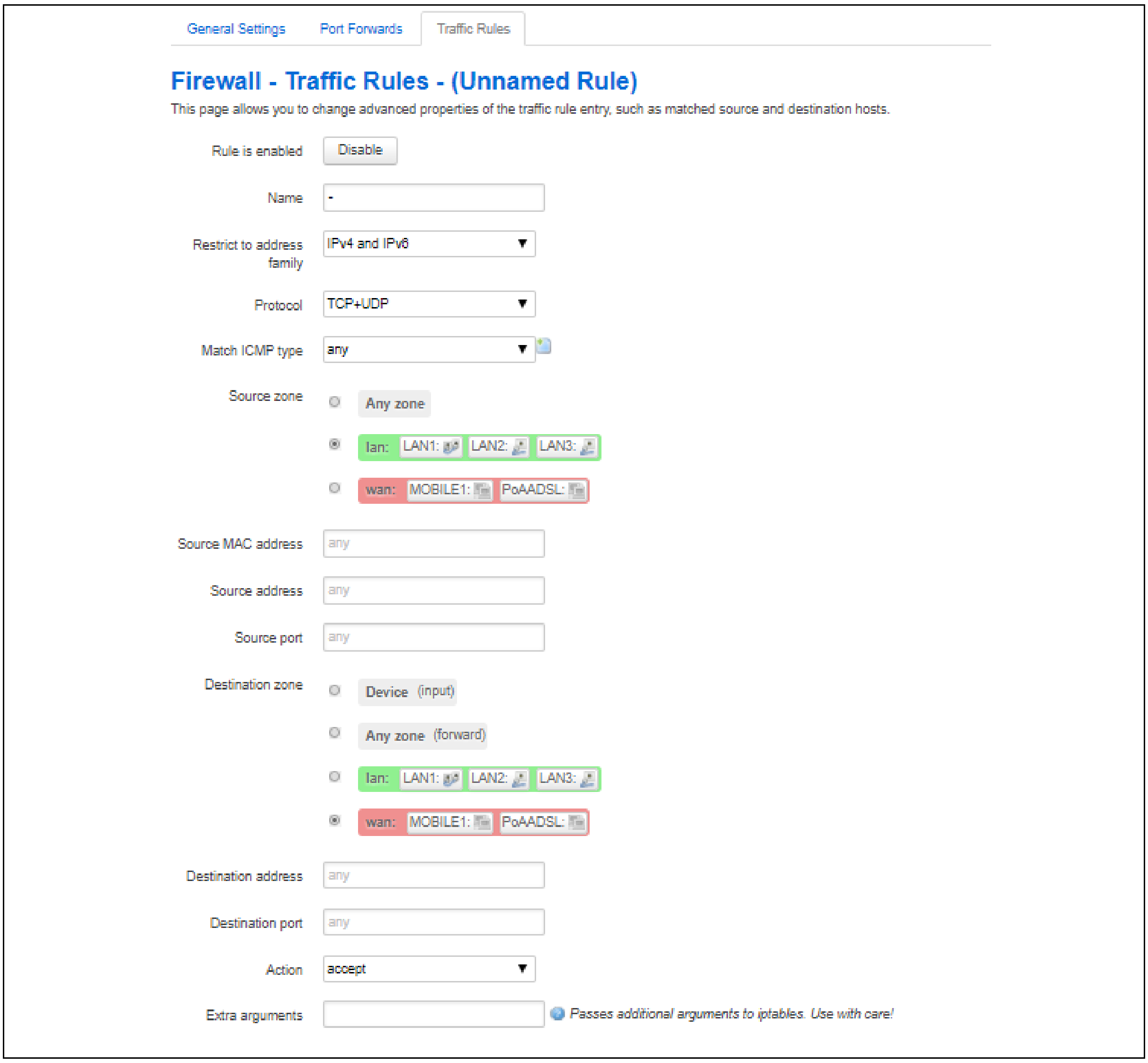


Figure 142: The firewall traffic rules page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled
UCI: firewall.<rule label>.enabled
Opt: enabled</td><td>Enables or disables traffic rule.<table><tr><td>0</td><td>Rule is disabled.</td></tr><tr><td>1</td><td>Rule is enabled.</td></tr></table></td></tr>
<tr><td>Web: Name
UCI: firewall.<rule label>.name
Opt: name</td><td>Select a descriptive name limited to less than 11 characters. No spaces are allowed in the naming convention.</td></tr>
<tr><td>Web: Restrict to address family
UCI: firewall.<rule label>.family
Opt: family</td><td>Restrict to protocol family.<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>IPv4 and IPv6</td><td>Traffic rule applies to any address family</td><td>any</td></tr><tr><td>IPv4 only</td><td>IPv4 only</td><td>ipv4</td></tr><tr><td>IPv6 only</td><td>IPv6 only</td><td>Ipv6</td></tr></table></td></tr>
<tr><td>Web: Protocol
UCI: firewall.<rule label>.proto
Opt: proto</td><td>Matches incoming traffic using the given protocol.<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>TCP+UDP</td><td>Applies rule to TCP and UDP only</td><td>tcp udp</td></tr><tr><td>TCP</td><td>Applies rule to TCP only</td><td>tcp</td></tr><tr><td>UDP</td><td>Applies rule to UDP only</td><td>udp</td></tr><tr><td>ICMP</td><td>Applies rule to ICMP only</td><td>icmp</td></tr><tr><td>custom</td><td>Specify protocol from /etc/protocols</td><td></td></tr></table></td></tr>
<tr><td>Web: Match ICMP type
UCI: firewall.<rule label>.icmp_type
Opt: icmp_type</td><td>Match specific icmp types.
This option is only valid when ICMP is selected as the protocol. ICMP types can be listed as either type names or type numbers.
Note: for a full list of valid ICMP type names, see the ICMP Options table below.</td></tr>
<tr><td>Web: Source zone
UCI: firewall.<rule label>.src
Opt: src</td><td>Specifies the traffic source zone, must refer to one of the defined zone names. For typical port forwards, this is usually WAN.</td></tr>
<tr><td>Web: Source MAC address
UCI: firewall.<rule label>.src_mac
Opt: src_mac</td><td>Matches incoming traffic from the specified MAC address.
The MAC address must be entered in the following format:
<code>aa:bb:cc:dd:ee:ff:</code>
To only match the first portion of the MAC address append <code>/prefix</code> to the option value, where <code>prefix</code> defines the bits from the start of the MAC to match on.
Example:
<code>option src_mac 00:E0:C8:12:34:56/24</code>
will match on all packets with prefix 00:E0:C8.</td></tr>
<tr><td>Web: Source address
UCI: firewall.<rule label>.src_ip
Opt: src_ip</td><td>Matches incoming traffic from the specified source IP address.</td></tr>
<tr><td>Web: Source port
UCI: firewall.<rule label>.src_port
Opt: src_port</td><td>Matches incoming traffic originating from the given source port or port range on the client host.</td></tr>
<tr><td>Web: Destination zone
UCI: firewall.<rule label>.dest
Opt: dest</td><td>Specifies the traffic destination zone. Must refer to one of the defined zone names.</td></tr>
<tr><td>Web: Destination address
UCI: firewall.<rule label>.dest_ip
Opt: dest_ip</td><td>For DNAT, redirects matched incoming traffic to the specified internal host.
For SNAT, matches traffic directed at the given address.</td></tr>
</table>

<table>
<tr><td>Web: Destination port
UCI: firewall.<rule label>.dest_port
Opt: dest_port</td><td>For DNAT, redirects matched incoming traffic to the given port on the internal host.
For SNAT, matches traffic directed at the given ports.</td></tr>
<tr><td>Web: Action
UCI: firewall.<rule label>.target
Opt: target</td><td>Action to take when rule is matched.<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>drop</td><td>Drop matching traffic</td><td>DROP</td></tr><tr><td>accept</td><td>Allow matching traffic</td><td>ACCEPT</td></tr><tr><td>reject</td><td>Reject matching traffic</td><td>REJECT</td></tr><tr><td>don’t track</td><td>Disable connection tracking for the rule. See the ‘Connection tracking’ section below for more information.</td><td>NOTRACK</td></tr></table></td></tr>
<tr><td>Web: Extra arguments
UCI: firewall.<rule label>.extra
Opt: extra</td><td>Passes extra arguments to IP tables. This is useful to specify additional match options, like -m policy --dir in for IPSec.</td></tr>
<tr><td>Web: n/a
UCI: firewall.<rule label>.reflection
Opt: reflection</td><td>Disables NAT reflection for this redirect if set to 0. Applicable to DNAT targets.</td></tr>
<tr><td>Web: n/a
UCI: firewall.<rule label>.limit
Opt: limit</td><td>Sets maximum average matching rate; specified as a number, with an optional /second, /minute, /hour or /day suffix. Example: 3/hour.</td></tr>
<tr><td>Web: n/a
UCI: firewall.<rule label>.limit_burst
Opt: limit_burst</td><td>Sets maximum initial number of packets to match. This number gets recharged by one every time the limit specified above is not reached, up to this number.</td></tr>
<tr><td>Web: n/a
UCI: firewall.<rule label>.recent
Opt: recent</td><td>Sets number of allowed connections within specified time. This command takes two values e.g. recent=2 120 will allow 2 connections within 120 seconds.</td></tr>
</table>

Table 89: Information table for firewall traffic rules

ICMP Options	ICMP Options	ICMP Options	ICMP Options
address-mask-reply	host-redirect	pong	time-exceeded
address-mask-request	host-unknown	port-unreachable	timestamp-reply
any	host-unreachable	precedence-cutoff	timestamp-request
communication-prohibited	ip-header-bad	protocol-unreachable	TOS-host-redirect
destination-unreachable	network-prohibited	redirect	TOS-host-unreachable
echo-reply	network-redirect	required-option-missing	TOS-network-redirect
echo-request	network-unknown	router-advertisement	TOS-network-unreachable
fragmentation-needed	network-unreachable	router-solicitation	ttl-exceeded
host-precedence-violation	parameter-problem	source-quench	ttl-zero-during-reassembly
host-prohibited	ping	source-route-failed	ttl-zero-during-transit

Table 90: Information table for match ICMP type drop-down menu

28.3 Configuring firewall using UCI

Firewall is configured under the firewall package /etc/config/firewall.

There are three config sections: defaults, zone, forwarding, redirect, rule and include.

You can configure multiple zone, forwarding and redirect sections.

28.3.1 Firewall general settings

To set general (default) settings, enter:

```
uci add firewall defaults
uci set firewall.@defaults[0].syn_flood=1
uci set firewall.@defaults[0].drop_invalid=1
uci set firewall.@defaults[0].input=ACCEPT
uci set firewall.@defaults[0].output=ACCEPT
uci set firewall.@defaults[0].forward=ACCEPT
```

Note: this command is only required if there is no defaults section.

28.3.2 Firewall zone settings

By default, all firewall zone instances are named zone, instances are identified by @zone then the zone position in the package as a number. For example, for the first zone in the package using UCI:

```
firewall.@zone[0]=zone
firewall.@zone[0].name=lan
```

Or using package options:

```
config zone
      option name 'lan'
```

To set up a firewall zone, enter:

```
uci add firewall zone
uci set firewall.@zone[1].name=lan
uci set firewall.@zone[1].input=ACCEPT
uci set firewall.@zone[1].output=ACCEPT
uci set firewall.@zone[1].forward=ACCEPT
uci set firewall.@zone[1].network=lan1 wifi_client
uci set firewall.@zone[1].family=any
uci set firewall.@zone[1].masq_src=10.0.0.0/24
uci set firewall.@zone[1].masq_dest=20.0.0.0/24
uci set firewall.@zone[1].conntrack=1
```

```
uci set firewall.@zone[1].masq=1
uci set firewall.@zone[1].mtu_fix=1
uci set firewall.@zone[1].log=1
uci set firewall.@zone[1].log_limit=5
```

28.3.3 Inter-zone forwarding

By default, all inter-zone instances are named ‘forwarding’; instances are identified by `@forwarding` then the forwarding position in the package as a number. For example, for the first forwarding in the package using UCI:

```
firewall.@forwarding[0]=forwarding
firewall.@forwarding[0].src=lan
```

Or using package options:

```
config forwarding
      option src 'lan'
```

To enable forwarding of traffic from WAN to LAN, enter:

```
uci add firewall forwarding
uci set firewall.@forwarding[1].dest=wan
uci set firewall.@forwarding[1].src=lan
```

28.3.4 Firewall port forwards

By default, all port forward instances are named ‘redirect’; instances are identified by `@redirect` then the redirect position in the package as a number. For example, for the first redirect in the package using UCI:

```
firewall.@redirect[0]=redirect
firewall.@redirect[0].name=Forward
```

Or using package options:

```
config redirect
      option name 'Forward'
```

To set port forwarding rules, enter:

```
uci add firewall redirect
uci set firewall.@redirect[1].name=Forward
uci set firewall.@redirect[1].proto=tcp
uci set firewall.@redirect[1].src=wan   #  <- zone names
```

```
uci set firewall.@redirect[1].dest=lan   # <- zone names
uci set firewall.@redirect[1].src_dport=2001
uci set firewall.@redirect[1].dest_ip=192.168.0.100
uci set firewall.@redirect[1].dest_port=2005
uci set firewall.@redirect[1].enabled=1
```

28.3.5 Firewall traffic rules

By default, all traffic rule instances are named rule, instances are identified by @rule then the rule position in the package as a number. For example, for the first rule in the package using UCI:

```
firewall.@rule[0]=rule
firewall.@rule[0].enabled=1
```

Or using package options:

```
config rule
      option enabled '1'
```

To set traffic rules, enter:

```
uci add firewall rule
uci set firewall.@rule[1].enabled=1
uci set firewall.@rule[1].name=Allow_ICMP
uci set firewall.@rule[1].family=any
uci set firewall.@rule[1].proto=ICMP
uci set firewall.@rule[1].icmp_type=any
uci set firewall.@rule[1].src=wan
uci set firewall.@rule[1].src_mac=ff:ff:ff:ff:ff:ff
uci set firewall.@rule[1].src_port=
uci set firewall.@rule[1].dest=lan
uci set firewall.@rule[1].dest_port=
uci set firewall.@rule[1].dest_ip=192.168.100.1
uci set firewall.@rule[1].target=ACCEPT
uci set firewall.@rule[1].extra=
uci set firewall.@rule[1].src_ip=8.8.8.8
uci set firewall.@rule[1].src_dip=9.9.9.9
uci set firewall.@rule[1].src_dport=68
uci set firewall.@rule[1].reflection=1
uci set firewall.@rule[1].limit=3/second
uci set firewall.@rule[1].limit_burst=30
```

28.3.5.1 Custom firewall scripts: includes

It is possible to include custom firewall scripts by specifying one or more include sections in the firewall configuration.

There is only one possible parameter for includes:

Parameter	Description
path	Specifies a shell script to execute on boot or firewall restarts.

Custom scripts are executed as shell scripts and are expected to contain iptables commands.

28.4 IPv6 notes

As described above, the option family is used for distinguishing between IPv4, IPv6 and both protocols. However, the family is inferred automatically if a specific IP address family is used. For example; if IPv6 addresses are used then the rule is automatically treated as IPv6 only rule.

```
config rule
        option src wan
        option src_ip fdca:f00:ba3::/64
        option target ACCEPT
```

Similarly, the following rule is automatically treated as IPv4 only.

```
config rule
        option src wan
        option dest_ip 88.77.66.55
        option target REJECT
```

Rules without IP addresses are automatically added to iptables and ip6tables, unless overridden by the family option. Redirect rules (port forwards) are always IPv4 since there is no IPv6 DNAT support at present.

28.5 Implications of DROP vs. REJECT

The decision whether to drop or to reject traffic should be done on a case-by-case basis. Many people see dropping traffic as a security advantage over rejecting it because it exposes less information to a hypothetical attacker. While dropping slightly increases security, it can also complicate the debugging of network issues or cause unwanted side-effects on client programs.

If traffic is rejected, the router will respond with an icmp error message ("destination port unreachable") causing the connection attempt to fail immediately. This also means that for each connection attempt a certain amount of response traffic is generated. This can actually harm if the firewall is attacked with many simultaneous connection attempts, the resulting backfire of icmp responses can clog up all available upload and make the connection unusable (DoS).

When connection attempts are dropped the client is not aware of the blocking and will continue to re-transmit its packets until the connection eventually times out. Depending on the way the client software is implemented, this could result in frozen or hanging programs that need to wait until a timeout occurs before they're able to continue.

DROP

- less information is exposed
- less attack surface
- client software may not cope well with it (hangs until connection times out)
- may complicate network debugging (where was traffic dropped and why)

REJECT

- may expose information (like the IP at which traffic was actually blocked)
- client software can recover faster from rejected connection attempts
- network debugging easier (routing and firewall issues clearly distinguishable)

28.6 Connection tracking

By default, the firewall will disable connection tracking for a zone if no masquerading is enabled. This is achieved by generating NOTRACK firewall rules matching all traffic passing via interfaces referenced by the firewall zone. The purpose of NOTRACK is to speed up routing and save memory by circumventing resource intensive connection tracking in cases where it is not needed. You can check if connection tracking is disabled by issuing iptables -t raw -S, it will list all rules, check for NOTRACK target.

NOTRACK will render certain iptables extensions unusable, for example the MASQUERADE target or the state match will not work.

If connection tracking is required, for example by custom rules in /etc/firewall.user, the conntrack option must be enabled in the corresponding zone to disable NOTRACK. It should appear as option 'conntrack' '1' in the right zone in /etc/config/firewall.

28.7 Firewall examples

28.7.1 Opening ports

The default configuration accepts all LAN traffic, but blocks all incoming WAN traffic on ports not currently used for connections or NAT. To open a port for a service, add a rule section:

```
config rule
        option src              wan
        option dest_port        22
        option target           ACCEPT
        option proto            tcp
```

This example enables machines on the internet to use SSH to access your router.

28.7.2 Forwarding ports (destination NAT/DNAT)

This example forwards http, but not HTTPS, traffic to the web server running on 192.168.1.10:

```
config redirect
        option src       wan
        option src_dport 80
        option proto     tcp
        option dest_ip   192.168.1.10
```

The next example forwards one arbitrary port that you define to a box running SSH behind the firewall in a more secure manner because it is not using default port 22.

```
config 'redirect'
        option 'name' 'ssh'
        option 'src' 'wan'
        option 'proto' 'tcpudp'
        option 'src_dport' '5555'
        option 'dest_ip' '192.168.1.100'
        option 'dest_port' '22'
        option 'target' 'DNAT'
        option 'dest' 'lan'
```

28.7.3 Source NAT (SNAT)

Source NAT changes an outgoing packet destined for the system so that is looks as though the system is the source of the packet.

Define source NAT for UDP and TCP traffic directed to port 123 originating from the host with the IP address 10.55.34.85. The source address is rewritten to 63.240.161.99.

```
config redirect
        option src              lan
        option dest             wan
        option src_ip           10.55.34.85
        option src_dip          63.240.161.99
        option dest_port        123
        option target           SNAT
```

When used alone, Source NAT is used to restrict a computer's access to the internet, but allows it to access a few services by manually forwarding what appear to be a few local services; for example, NTP to the Internet. While DNAT hides the local network from the Internet, SNAT hides the Internet from the local network.

Source NAT and destination NAT are combined and used dynamically in IP masquerading to make computers with private (192.168.x.x, etc.) IP addresses appear on the internet with the system's public WAN IP address.

28.7.4 True destination port forwarding

This usage is similar to SNAT, but as the destination IP address is not changed, machines on the destination network need to be aware that they'll receive and answer requests from a public IP address that is not necessarily theirs. Port forwarding in this fashion is typically used for load balancing.

```
config redirect
        option src              wan
        option src_dport        80
        option dest             lan
        option dest_port        80
        option proto            tcp
```

28.7.5 Block access to a specific host

The following rule blocks all connection attempts to the specified host address.

```
config rule
        option src              lan
        option dest             wan
        option dest_ip          123.45.67.89
        option target           REJECT
```

28.7.6 Block access to the internet using MAC

The following rule blocks all connection attempts from the client to the internet.

```
config rule
        option src              lan
        option dest             wan
        option src_mac          00:00:00:00:00:00
        option target           REJECT
```

28.7.7 Block access to the internet for specific IP on certain times

The following rule blocks all connection attempts to the internet from 192.168.1.27 on weekdays between 21:00pm and 09:00am.

```
config rule
        option src          lan
        option dest         wan
        option src_ip       192.168.1.27
        option extra        '-m time --weekdays Mon,Tue,Wed,Thu,Fri --
timestart 21:00 --timestop 09:00'
        option target       REJECT
```

28.7.8 Restricted forwarding rule

The example below creates a forward rule rejecting traffic from LAN to WAN on the ports 1000-1100.

```
config rule
        option src              lan
        option dest             wan
        option dest_port        1000-1100
        option proto            tcpudp
        option target           REJECT
```

28.7.9 Denial of service protection rule

The example below shows a sample configuration of SSH DoS attack where if more than two SSH connections are attempted within 120 seconds, every further connection will be dropped. You can configure this for any port number.

```
config rule 'sshattack'
        option src 'lan'
        option dest_port '22'
        option proto 'tcp'
        option recent '2 120'
        option target 'DROP'
```

28.7.10 IP spoofing prevention mechnism

Configure IP spoofing protection on a per interface basis in the /etc/config/network configuration file. The example below shows the ipv4_rp_filter option enabled on the Vlan12 interface in the network file. When reverse path filtering mechanism is enabled, the router will check whether a receiving packet source address is routable.

If it is routable through the interface from which it came, then the machine will accept the packet

If it is not routable through the interface from which it came, then the machine will drop that packet.

```
config interface 'Vlan12'
        option type 'bridge'
        option proto 'static'
        option monitored '0'
        option ipaddr '10.1.28.122'
        option netmask '255.255.0.0'
        option ifname 'eth1 eth3.12'
        option ipv4_rp_filter '1'
```

28.7.11 Simple DMZ rule

The following rule redirects all WAN ports for all protocols to the internal host 192.168.1.2.

```
config redirect
      option src              wan
      option proto            all
      option dest_ip          192.168.1.2
```

28.7.12 Transparent proxy rule (external)

The following rule redirects all outgoing HTTP traffic from LAN through an external proxy at 192.168.1.100 listening on port 3128. It assumes the router LAN address to be 192.168.1.1 - this is needed to masquerade redirected traffic towards the proxy.

```
config redirect
        option src              lan
        option proto            tcp
        option src_ip           !192.168.1.100
        option src_dport        80
        option dest_ip          192.168.1.100
        option dest_port        3128
        option target           DNAT

config redirect
        option dest             lan
        option proto            tcp
```

```
        option src_dip          192.168.1.1
        option dest_ip          192.168.1.100
        option dest_port        3128
        option target           SNAT
```

28.7.13 Transparent proxy rule (same host)

The rule below redirects all outgoing HTTP traffic from LAN through a proxy server listening at port 3128 on the router itself.

```
config redirect
      option src              lan
      option proto            tcp
      option src_dport        80
      option dest_port        3128
```

28.7.14 IPSec passthrough

This example enables proper forwarding of IPSec traffic through the WAN.

```
# AH protocol
config rule
        option src              wan
        option dest             lan
        option proto            ah
        option target           ACCEPT
# ESP protocol
config rule
        option src              wan
        option dest             lan
        option proto            esp
        option target           ACCEPT
```

For some configurations you also have to open port 500/UDP.

```
# ISAKMP protocol
config rule
        option src              wan
        option dest             lan
        option proto            udp
        option src_port         500
```

```
        option dest_port        500
        option target           ACCEPT
```

28.7.15 Manual iptables rules

You can specify traditional iptables rules, in the standard iptables UNIX command form, in an external file and included in the firewall config file. It is possible to use this process to include multiple files.

```
config include
       option path /etc/firewall.user

config include
       option path /etc/firewall.vpn
```

The syntax for the includes is Linux standard and therefore different from UCIs.

28.7.16 Firewall management

After a configuration change, to rebuild firewall rules, enter:

```
root@VA_router:/# /etc/init.d/firewall restart
```

Executing the following command will flush all rules and set the policies to ACCEPT on all standard chains:

```
root@VA_router:/# /etc/init.d/firewall stop
```

To manually start the firewall, enter:

```
root@VA_router:/# /etc/init.d/firewall start
```

To permanently disable the firewall, enter:

```
root@VA_router:/# /etc/init.d/firewall disable
```

Note: disable does not flush the rules, so you might be required to issue a stop before.

To enable the firewall again, enter:

```
root@VA_router:/# /etc/init.d/firewall enable
```

28.7.17 Debug generated rule set

It is possible to observe the iptables commands generated by the firewall programme. This is useful to track down iptables errors during firewall restarts or to verify the outcome of certain UCI rules.

To see the rules as they are executed, run the `fw` command with the FW_TRACE environment variable set to **1**:

```
root@VA_router:/# FW_TRACE=1 fw reload
```

To direct the output to a file for later inspection, enter:

```
root@VA_router:/# FW_TRACE=1 fw reload 2>/tmp/iptables.lo
```

29 Configuring IPSec

Internet Protocol Security (IPSec) is a protocol suite used to secure communications at IP level. Use IPSec to secure communications between two hosts or between two networks. Virtual Access routers implement IPSec using strongSwan software.

If you need to create an IPSec template for DMVPN, read the chapter ‘Dynamic Multipoint Virtual Private Network (DMVPN)’.

The number of IPSec tunnels supported by Virtual Access’ routers is not limited in any way by software; the only hardware limitation is the amount of RAM installed on the device.

29.1 Configuration package used

Package	Sections
strongswan	general connection secret

29.2 Configuring IPSec using the web interface

To configure IPSec using the web interface, in the top menu, select **Services -> IPSec**. The strongSwan IPSec VPN page appears. There are three sections:

Common Settings	Control the overall behaviour of strongSwan. This behaviour is common across all tunnels.
Connection Settings	Together, these sections define the required parameters for a two-way IKEv1 tunnel.
Secret Settings	

29.2.1 Configure common settings

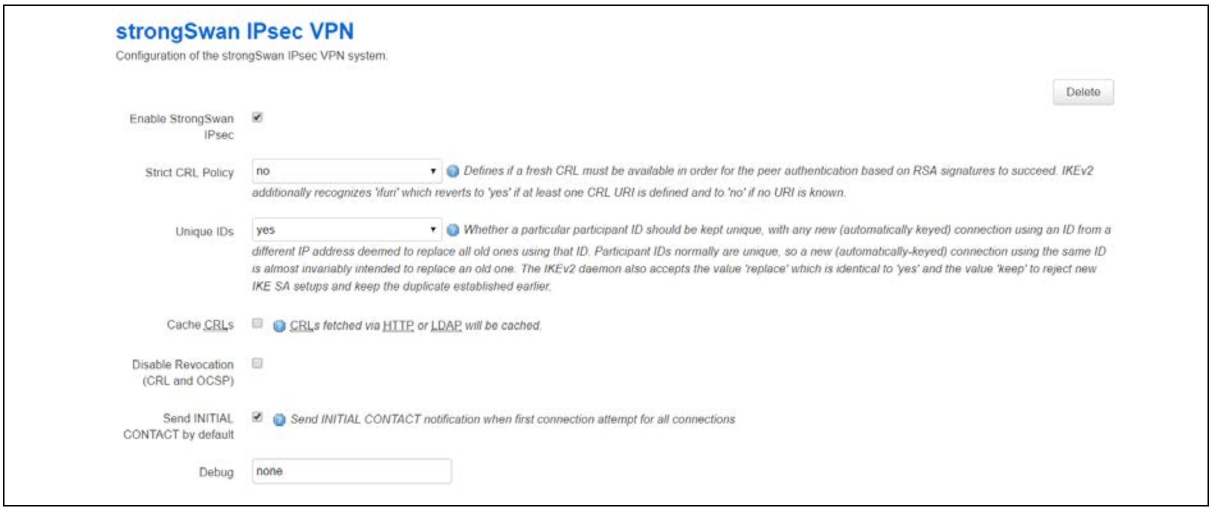


Figure 143: The common settings section

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enable strongswan
UCI: strongswan.general.enable
Opt: enabled</td><td>Enables or disables IPSec.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Strict CRL Policy
UCI: strongswan.general.strictcrlpolicy
Opt: strictcrlpolicy</td><td>Defines if a fresh CRL must be available for the peer authentication based on RSA signatures to succeed.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr><tr><td>ifuri</td><td>The IKEv2 application additionally recognizes the “ifuri” option which reverts to ‘yes’ if at least one CRL URI is defined and to ‘no’ if no URI is known.</td></tr></table></td></tr>
<tr><td>Web: Unique IDs
UCI: strongswan.general.uniqueids
Opt: uniqueids</td><td>Defines whether a particular participant ID should be kept unique, with any new (automatically keyed) connection using an ID from a different IP address deemed to replace all old ones using that ID.
Participant IDs normally are unique, so a new (automatically-keyed) connection using the same ID is almost invariably intended to replace an old one.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr><tr><td>replace</td><td>Identical to Yes.</td></tr><tr><td>keep</td><td>Rejects new IKE SA and keep the duplicate established earlier</td></tr></table></td></tr>
<tr><td>Web: Cache CRLs
UCI: strongswan.general.cachecrls
Opt: cachecrls</td><td>Certificate Revocation Lists (CRLs) fetched via HTTP or LDAP will be cached in /etc/ipsec.d/crls/ under a unique file name derived from the certification authority's public key.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Disable Revocation
UCI: strongswan.general.revocation_disabled
Opt: revocation_disabled</td><td>Defines whether disable CRL and OCSP checking for revoked certificates.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Send INITIAL CONTACT by default
UCI: strongswan.general.initial_contact
Opt: initial_contact</td><td>Defines whether the first attempt to contact a remote peer by this strongswan instance sets the initial_contact flag, which should cause compliant peers to automatically bring down any previous sessions. This can also be enabled/disabled per connection.<table><tr><td>0</td><td>Does not set initial contact flag.</td></tr><tr><td>1</td><td>Sets initial contact flag on first attempt.</td></tr></table></td></tr>
<tr><td>Web: Debug
UCI: strongswan.general.debug
Opt: debug</td><td>Enables debugging. This option is used for trouble shooting issues. It is not suitable for a production environment.<table><tr><td>None</td><td>Debug disabled.</td></tr><tr><td>Control</td><td>Debug enabled. Shows generic control flow with errors and very basic auditing logs.</td></tr><tr><td>All</td><td>Debug enabled. Most verbose logging also includes sensitive information such as keys.</td></tr></table></td></tr>
</table>

Table 91: Information table for IPSec common settings

29.2.2 Common settings: configure connection

Figure 144: The configuring IPSec settings

Web Field / UCI / Package Option	Description
Web: Enabled UCI: strongswan.@connection[X].enabled Opt: enable	Enables or disables IPSec connection. 0: Disabled. 1: Enabled.
Web: Aggressive UCI: strongswan.@connection[X].aggressive Opt: aggressive	Enables or disables IKE aggressive mode. **Note**: using aggressive mode along with PSK authentication is less secure method than main mode and should be avoided. 0: Disabled. 1: Enabled.
Web: Name UCI: strongswan.@connection[X].name Opt: name	Specifies a name for the tunnel.
Web: Autostart Action UCI: strongswan.@connection[X].auto Opt: auto	Specifies when the tunnel is initiated. start: On start up. route: When traffic routes this way. add: Loads a connection without starting it. ignore: Ignores the connection. always: Actively retries to establish the tunnel if it went down.
Web: Connection Type UCI: strongswan.@connection[X].type Opt: type	Defines the type of IPSec connection. tunnel: Connection uses tunnel mode. transport: Connection uses transport mode. pass: Connection does not perform any IPSec processing. drop: Connection drops all the packets.

Table 92: Information table for connection settings

29.2.3 Common settings: IP addressing

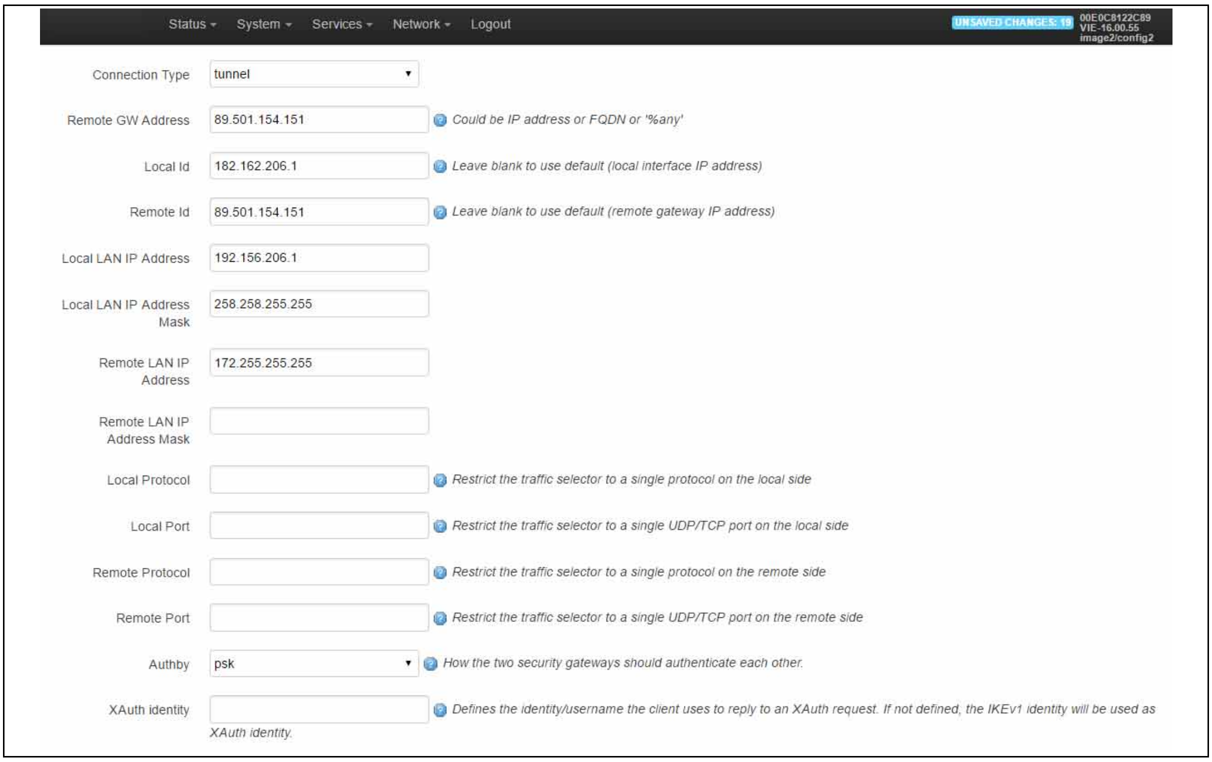


Figure 145: The IP addressing settings

Web Field / UCI / Package Option	Description
Web: Remote GW Address UCI: strongswan.@connection[X].remoteaddress Opt: remoteaddress	Sets the public IP address of the remote peer.
Web: Local ID UCI: strongswan.@connection[X].localid Opt: localid	Defines the local peer identifier.
Web: Remote ID UCI: strongswan.@connection[X].remoteid Opt:remoteid	Defines the remote peer identifier.
Web: Local LAN IP Address UCI: strongswan.@connection[X]. locallan Opt: locallan	Defines the local IP of LAN.
Web: Local LAN IP Address Mask UCI: strongswan.@connection[X].locallanmask Opt: locallanmask	Defines the subnet of local LAN.
Web: Remote LAN IP Address UCI: strongswan.@connection[X]. remotelan Opt:remotelan	Defines the IP address of LAN serviced by remote peer.
Web: Remote LAN IP Address Mask UCI: strongswan.@connection[X].remotelanmask Opt:remotelanmask	Defines the Subnet of remote LAN.

<table>
<tr><td>Web: Local Protocol
UCI: strongswan.@connection[X].localproto
Opt: localproto</td><td>Restricts the connection to a single protocol on the local side.</td></tr>
<tr><td>Web: Local Port
UCI: strongswan.@connection[X].localport
Opt: localport</td><td>Restricts the connection to a single port on the local side.</td></tr>
<tr><td>Web: Remote Protocol
UCI: strongswan.@connection[X].remoteproto
Opt:remoteproto</td><td>Restricts the connection to a single protocol on the remote side.</td></tr>
<tr><td>Web: Remote Port
UCI: strongswan.@connection[X].remoteport
Opt: remoteport</td><td>Restricts the connection to a single port on the remote side.</td></tr>
<tr><td>Web: Authby
UCI: strongswan.@connection[X].authby
Opt: authby</td><td>Defines how the two secure gateways should authenticate.
Note: using aggressive mode along with PSK authentication is unsecure and should be avoided.
<table>
<tr><td>Pubkey</td><td>For public key signatures.</td></tr>
<tr><td>Rsasig</td><td>For RSA digital signatures.</td></tr>
<tr><td>ecdsasig</td><td>For Elliptic Curve DSA signatures.</td></tr>
<tr><td>Psk</td><td>Using a preshared key.</td></tr>
<tr><td>xauthrsasig</td><td>Enables eXtended Authentication (XAuth) with addition to RSA signatures.</td></tr>
<tr><td>xauthpsk</td><td>Using extended authentication and preshared key.</td></tr>
<tr><td>never</td><td>Can be used if negotiation is never to be attempted or accepted (shunt connections).</td></tr>
</table></td></tr>
</table>

Table 93: Information table for IP addressing settings

29.2.4 Common settings: IPSec settings

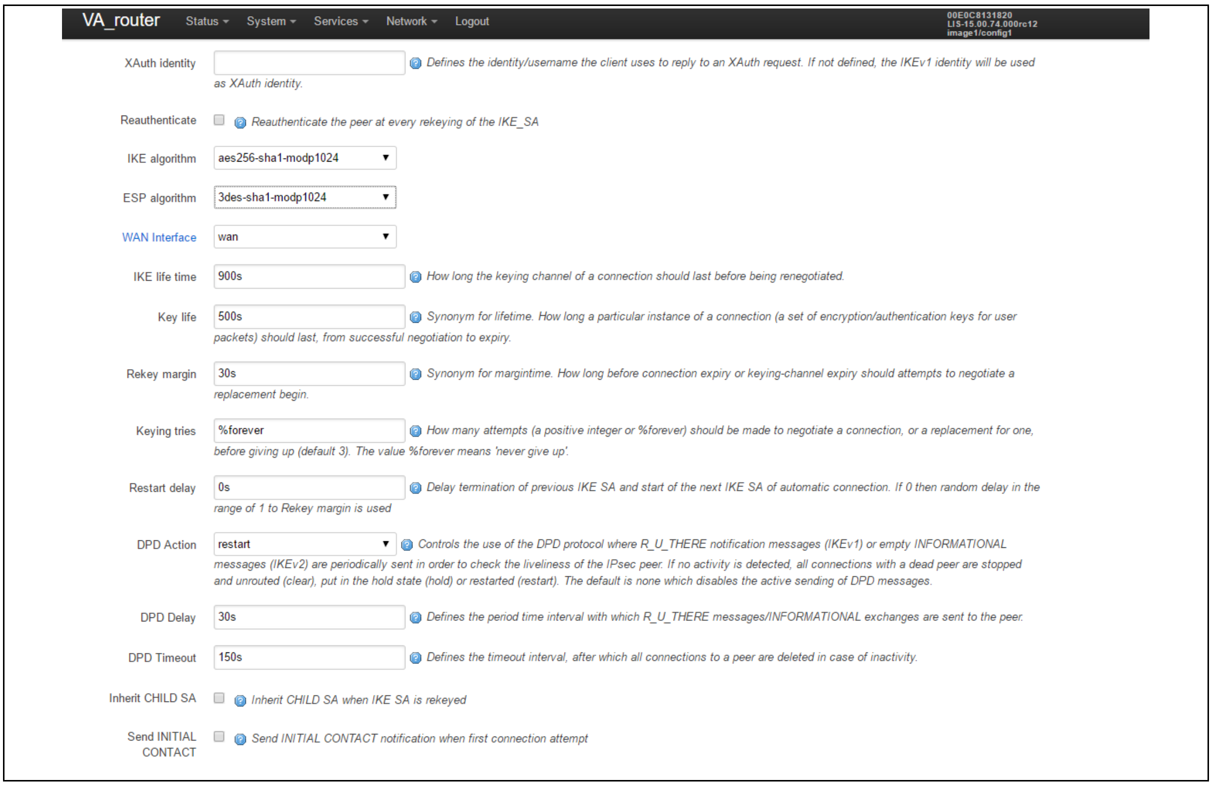


Figure 146: The IPSec connections settings

Web Field/UCI/Package Option	Description
Web: XAuth Identity UCI: strongswan.@connection[X].xauth_identity Opt: xauth_identity	Defines Xauth ID.
Web: IKE Algorithm UCI: strongswan.@connection[X].ike Opt: ike	Specifies the IKE algorithm to use. The format is: encAlgo | authAlgo | DHGroup encAlgo: 3des aes128 aes256 serpent twofish blowfish authAlgo: md5 sha sha2 DHGroup: modp1024 modp1536 modp2048 modp3072 modp4096 modp6144 modp8192 For example, a valid IKE algorithm is aes128-sha-modp1536.

Web: ESP algorithm UCI: strongswan.@connection[X].esp Opt: esp	Specifies the esp algorithm to use. The format is: encAlgo | authAlgo | DHGroup encAlgo: 3des aes128 aes256 serpent twofish blowfish authAlgo: md5 sha sha2 DHGroup: modp1024 modp1536 modp2048 modp3072 modp4096 modp6144 modp8192 For example, a valid encryption algorithm is: aes128-sha-modp1536. If no DH group is defined then PFS is disabled.
Web: WAN Interface UCI: strongswan.@connection[X].waniface Opt: waniface	This is a space-separated list of the WAN interfaces the router will use to establish a tunnel with the secure gateway. On the web, a list of the interface names is automatically generated. If you want to specify more than one interface use the "custom" value. Example: if you have a 3G WAN interface called 'wan and a WAN ADSL interface called 'dsl' and wanted to use one of these interfaces for this IPSec connection, you would use: 'wan adsl'.
Web: IKE Life Time UCI: strongswan.@connection[X].ikelifetime Opt:ikelifetime	Specifies how long the keyring channel of a connection (ISAKMP or IKE SA) should last before being renegotiated. 3h | Timespec | 1d, 3h, 25m, 10s.
Web: Key Life UCI: strongswan.@connection[X].keylife Opt: keylife	Specifies how long a particular instance of a connection (a set of encryption/authentication keys for user packets) should last, from successful negotiation to expiry. Normally, the connection is renegotiated (via the keying channel) before it expires (see rekeymargin). 1h | Timespec | 1d, 1h, 25m, 10s.
Web: Rekey Margin UCI: strongswan.@connection[X].rekeymargin Opt: rekeymargin	Specifies how long before connection expiry or keying-channel expiry should attempt to negotiate a replacement begin. Relevant only locally, other end need not agree on it. 9m | Timespec | 1d, 2h, 9m, 10s.

Web: Restart Delay UCI: strongswan.@connection[X].restartdelay Opt: restartdelay	Defines specific delay when re-establishing a connection. Previously if `close_action=restart`, then new option `restartdelay` controls how many seconds it waits before attempting to re-establish the tunnel (to allow head-end some time to tidy up). If not set, it defaults to zero, which means that the previous behaviour of choosing a random time interval in the range 0..`RekeyMargin` seconds takes effect. Relevant only locally, other end need not agree on it. 0 Timespec: 1d, 2h, 9m, 10s.
Web: Keying Tries UCI: strongswan.@connection[X].keyringtries Opt: keyringtries	Specifies how many attempts (a positive integer or %forever) should be made to negotiate a connection, or a replacement for one, before giving up. The value %forever means 'never give up'. Relevant only locally, other end need not agree on it.
Web: DPD Action UCI: strongswan.@connection[X].dpdaction Opt: dpdaction	Defines DPD (Dead Peer Detection) action. None: Disables DPD. Clear: Clear down the tunnel if peer does not respond. Reconnect when traffic brings the tunnel up. Hold: Clear down the tunnel and bring up as soon as the peer is available. Restart: Restarts DPD when no activity is detected.
Web: DPD Delay UCI: strongswan.@connection[X].dpddelay Opt: dpddelay	Defines the period time interval with which R_U_THERE messages and INFORMATIONAL exchanges are sent to the peer. These are only sent if no other traffic is received. 30s Timespec: 1d, 2h, 25m, 10s.
Web: DPD Timeout UCI: strongswan.@connection[X].dpdtimeout Opt: dpdtimeout	Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity. 150s Timespec: 1d, 2h, 25m, 10s.
Web: Inherit CHILD SA UCI: strongswan.@connection[X].inherit_child Opt: inherit_child	Defines whether the existing phase two IPSEC SA is maintained through IKE rekey for this tunnel. This is normally set to match the behaviour on the IPSEC headend. 0: Delete the existing IPSEC SA on IKE rekey 1: Maintain the existing IPSEC SA on IKE rekey
Web: Send INITIAL CONTACT UCI: strongswan.@connection[X].initial_contact Opt: initial_contact	Defines whether the first attempt to contact a remote peer by this strongswan instance sets the initial_contact flag which should cause compliant peers to automatically bring down any previous sessions. 0: Do not set initial contact flag 1: Set initial contact flag on first attempt

Table 94: Information table for IPSec connections settings

29.2.5 Configure secret settings

Each tunnel requires settings to configure how the local end point of the tunnel proves its identity to the remote end point.

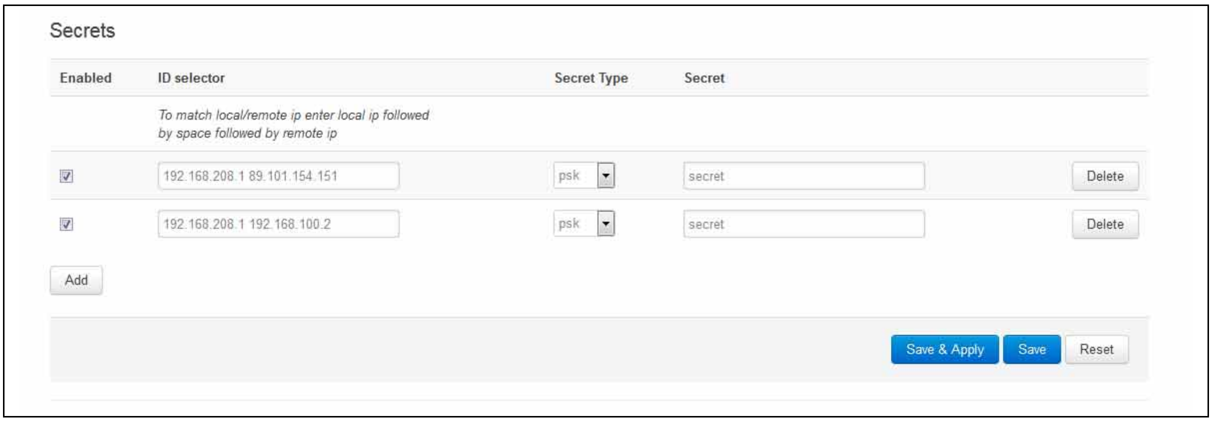


Figure 147: IPSec secrets settings

Web Field / UCI / Package Option	Description
Web: Enabled UCI: strongswan.@secret[X].enabled Opt: enabled	Defines whether this set of credentials is to be used or not. 0 – Disabled. 1 – Enabled.
Web: ID selector UCI: strongswan.@secret[X].idtype Opt: idtype	Defines whether IP address or userfqdn is used.
Web: ID selector UCI: strongswan.@secret[X].localaddress Opt: localaddress	Defines the local address this secret applies to.
Web: ID selector UCI: strongswan.@secret[X]. remoteaddress Opt: remoteaddress	Defines the remote address this secret applies to.
Web: N/A UCI: strongswan.@secret[X].userfqnd Opt: userfqnd	FQDN or Xauth name used of Extended Authentication. This must match xauth_identity from the configuration connection section.
Web: Secret Type UCI: strongswan.@secret[X].secrettype Opt: secrettype	Specifies the authentication mechanism to be used by the two peers. Psk – Preshared secret Pubkey – Public key signatures Rsasig – RSA digital signatures Ecdsasig – Elliptic Curve DSA signatures Xauth – Extended authentication
Web: Secret UCI: strongswan.@secret[X].secret Opt: secret	Defines the secret.

Table 95: Information table for IPSec secrets settings

29.3 Configuring IPSec using UCI

29.3.1 Common settings

```
# Commands
touch /etc/config/strongswan
uci set strongswan.general=general
uci set strongswan.general.enabled=yes
uci set strongswan.general.strictcrlpolicy=no
uci set strongswan.general.uniqueids=yes
uci set strongswan.general.cachecrls=no
uci set strongswan.general.debug=none
uci set strongswan.general.initial_contact=0
uci commit
```

This will create the following output:

```
config general 'general'
        option enabled 'yes'
        option strictcrlpolicy 'no'
        option uniqueids 'yes'
        option cachecrls 'no'
        option debug 'none'
        option initial_contact ‘0’
```

29.3.2 Connection settings

```
touch /etc/config/strongswan
uci add strongswan connection
uci set strongswan.@connection[0].ikelifetime=3h
uci set strongswan.@connection[0].keylife=1h
uci set strongswan.@connection[0].rekeymargin=9m
uci set strongswan.@connection[0].keyingtries=3
uci set strongswan.@connection[0].restartdelay=0
uci set strongswan.@connection[0].dpdaction=none
uci set strongswan.@connection[0].dpddelay=30s
uci set strongswan.@connection[0].dpdtimeout=150s
uci set strongswan.@connection[0].enabled=yes
uci set strongswan.@connection[0].name=3G_Backup
```

```
uci set strongswan.@connection[0].auto=start
uci set strongswan.@connection[0].type=tunnel
uci set strongswan.@connection[0].remoteaddress=100.100.100.100
uci set strongswan.@connection[0].localid=192.168.209.1
uci set strongswan.@connection[0].remoteid=100.100.100.100
uci set strongswan.@connection[0].locallan=192.168.209.1
uci set strongswan.@connection[0].locallanmask=255.255.255.255
uci set strongswan.@connection[0].remotelan=172.19.101.3
uci set strongswan.@connection[0].remotelanmask=255.255.255.255
uci set strongswan.@connection[0].authby=xauthpsk
uci set strongswan.@connection[0].xauth_identity=testxauth
uci set strongswan.@connection[0].ike=3des-md5-modp1024
uci set strongswan.@connection[0].esp=3des-md5
uci set strongswan.@connection[0].waniface=wan
uci set strongswan.@connection[0].inherit_child=0
uci set strongswan.@connection[0].initial_contact=0
uci commit
```

This will create the following output:

```
config connection
        option ikelifetime '3h'
        option keylife '1h'
        option rekeymargin '9m'
        option keyingtries '3'
        option restartdelay ‘0’
        option dpdaction ‘none’
        option dpddelay '30s'
        option dpdtimeout '150s'
        option enabled 'yes'
        option name '3G_Backup'
        option auto 'start'
        option type 'tunnel'
        option remoteaddress '100.100.100.100 '
        option localid '192.168.209.1'
        option remoteid '100.100.100.100 '
        option locallan '192.168.209.1'
        option locallanmask '255.255.255.255'
```

```
        option remotelan '172.19.101.3'
        option remotelanmask '255.255.255.255'
        option authby 'xauthpsk'
        option xauth_identity 'testxauth'
        option ike '3des-md5-modp1024'
        option esp '3des-md5'
        option waniface 'wan'
        option inherit_child ‘0’
        option initial_contact ‘0’
```

29.3.3 Shunt connection

If the remote LAN network is 0.0.0.0/0 then all traffic generated on the local LAN will be sent via the IPSec tunnel. This includes the traffic destined to the router’s IP address. To avoid this situation you must include an additional config connection section.

```
# Commands
touch /etc/config/strongswan
uci add strongswan connection
uci set strongswan.@connection[1].name=local
uci set strongswan.@connection[1].enabled=yes
uci set strongswan.@connection[1].locallan=10.1.1.1
uci set strongswan.@connection[1].locallanmask=255.255.255.255
uci set strongswan.@connection[1].remotelan=10.1.1.0
uci set strongswan.@connection[1].remotelanmask=255.255.255.0
uci set strongswan.@connection[1].type=pass
uci set strongswan.@connection[1].auto=route
uci commit
```

This will create the following output:

```
config connection
        option name 'local'
        option enabled 'yes'
        option locallan '10.1.1.1'
        option locallanmask '255.255.255.255'
        option remotelan '10.1.1.0'
        option remotelanmask '255.255.255.0'
        option type 'pass'
        option auto 'route'
```

Traffic originated on `remotelan` and destined to `locallan` address is excluded from VPN IPSec policy.

29.3.4 Secret settings

Each tunnel also requires settings for how the local end point of the tunnel proves its identity to the remote end point.

A sample secret section, which could be used with the connection section in 'Connection Settings', is shown below.

```
# Commands to add a secret for psk auth
touch /etc/config/strongswan
uci add strongswan secret
uci set strongswan.@secret[0].enabled=yes
uci set strongswan.@secret[0].localaddress=192.168.209.1
uci set strongswan.@secret[0].remoteaddress= 100.100.100.100
uci set strongswan.@secret[0].secrettype=psk
uci set strongswan.@secret[0].secret=secret
uci commit
```

This will create the following output:

```
config secret
        option enabled 'yes'
        option localaddress '192.168.209.1'
        option remoteaddress '100.100.100.100 '
        option secrettype 'psk'
        option secret 'secret'
```

If xauth is defined as the authentication method then you must include an additional config secret section, as shown in the example below.

```
# Commands to add a secret for xauth auth
touch /etc/config/strongswan
uci add strongswan secret
uci set strongswan.@secret[1].enabled=yes
uci set strongswan.@secret[1].idtype=userfqdn
uci set strongswan.@secret[1].userfqdn=testxauth
uci set strongswan.@secret[1].remoteaddress=100.100.100.100
uci set strongswan.@secret[1].secret=xauth
```

```
uci set strongswan.@secret[1].secrettype=XAUTH
uci commit
```

This will create the following output:

```
config secret
        option enabled 'yes'
        option idtype 'userfqdn'
        option userfqdn 'testxauth'
        option remoteaddress '100.100.100.100'
        option secret 'xauth'
        option secrettype 'XAUTH'
```

29.4 Configuring an IPSec template for DMVPN via the web interface

To configure IPSec using the web interface, in the top menu, select **Services -> IPSec**. The strongSwan IPSec VPN page appears. There are three sections:

Common Settings	Control the overall behaviour of strongSwan. This behaviour is common across all tunnels.
Connection Settings	Together, these sections define the required parameters for a two-way IKEv1 tunnel.
Secret Settings	

29.4.1 Configure common settings

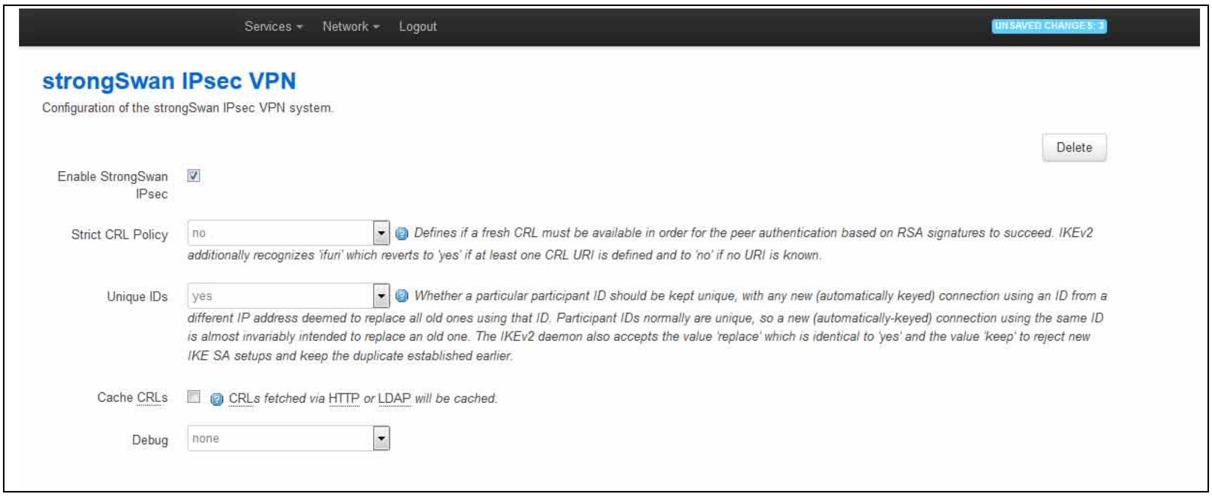


Figure 148: The common settings section

<table>
<tr><th>Web Field / UCI / Package Option</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">Web: Enable strongswan
UCI: strongswan.general.enable
Opt: enabled</td><td colspan="2">Enables or disables IPSec.</td></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
<tr><td rowspan="4">Web: Strict CRL Policy
UCI: strongswan.general.strictcrlpolicy
Opt: strictcrlpolicy</td><td colspan="2">Defines if a fresh CRL must be available for the peer authentication based on RSA signatures to succeed.</td></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
<tr><td>ifuri</td><td>The IKEv2 application additionally recognizes the “ifuri” option which reverts to ‘yes’ if at least one CRL URI is defined and to ‘no’ if no URI is known.</td></tr>
<tr><td rowspan="5">Web: Unique IDs
UCI: strongswan.general.uniqueids
Opt: uniqueids</td><td colspan="2">Defines whether a particular participant ID should be kept unique, with any new (automatically keyed) connection using an ID from a different IP address deemed to replace all old ones using that ID.
Participant IDs normally are unique, so a new (automatically-keyed) connection using the same ID is almost invariably intended to replace an old one.</td></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
<tr><td>replace</td><td>Identical to Yes</td></tr>
<tr><td>keep</td><td>Rejects new IKE SA and keep the duplicate established earlier</td></tr>
<tr><td rowspan="3">Web: Cache CRLs
UCI: strongswan.general.cachecrls
Opt: cachecrls</td><td colspan="2">Certificate Revocation Lists (CRLs) fetched via HTTP or LDAP will be cached in /etc/ipsec.d/crls/ under a unique file name derived from the certification authority's public key.</td></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
<tr><td rowspan="4">Web: Debug
UCI: strongswan.general.debug
Opt: debug</td><td colspan="2">Enable debugging. This option is used for trouble shooting issues. It is not suitable for a production environment.</td></tr>
<tr><td>None</td><td>Debug disabled.</td></tr>
<tr><td>Control</td><td>Debug enabled. Shows generic control flow with errors and very basic auditing logs.</td></tr>
<tr><td>All</td><td>Debug enabled. Most verbose logging also includes sensitive information such as keys.</td></tr>
</table>

Table 96: Information table for IPSec common settings

29.4.2 Configure connection settings

Scroll down to view the connection settings section.

If you want to create a DMVPN, you do not need to configure all settings as the DMVPN will automatically create them using the template. Leave the following sections blank:

- Remote GW Address
- Local ID
- Remote Id
- Local LAN IP Address
- Local LAN IP Address Mask
- Remote LAN IP Address
- Remote LAN IP Address Mask

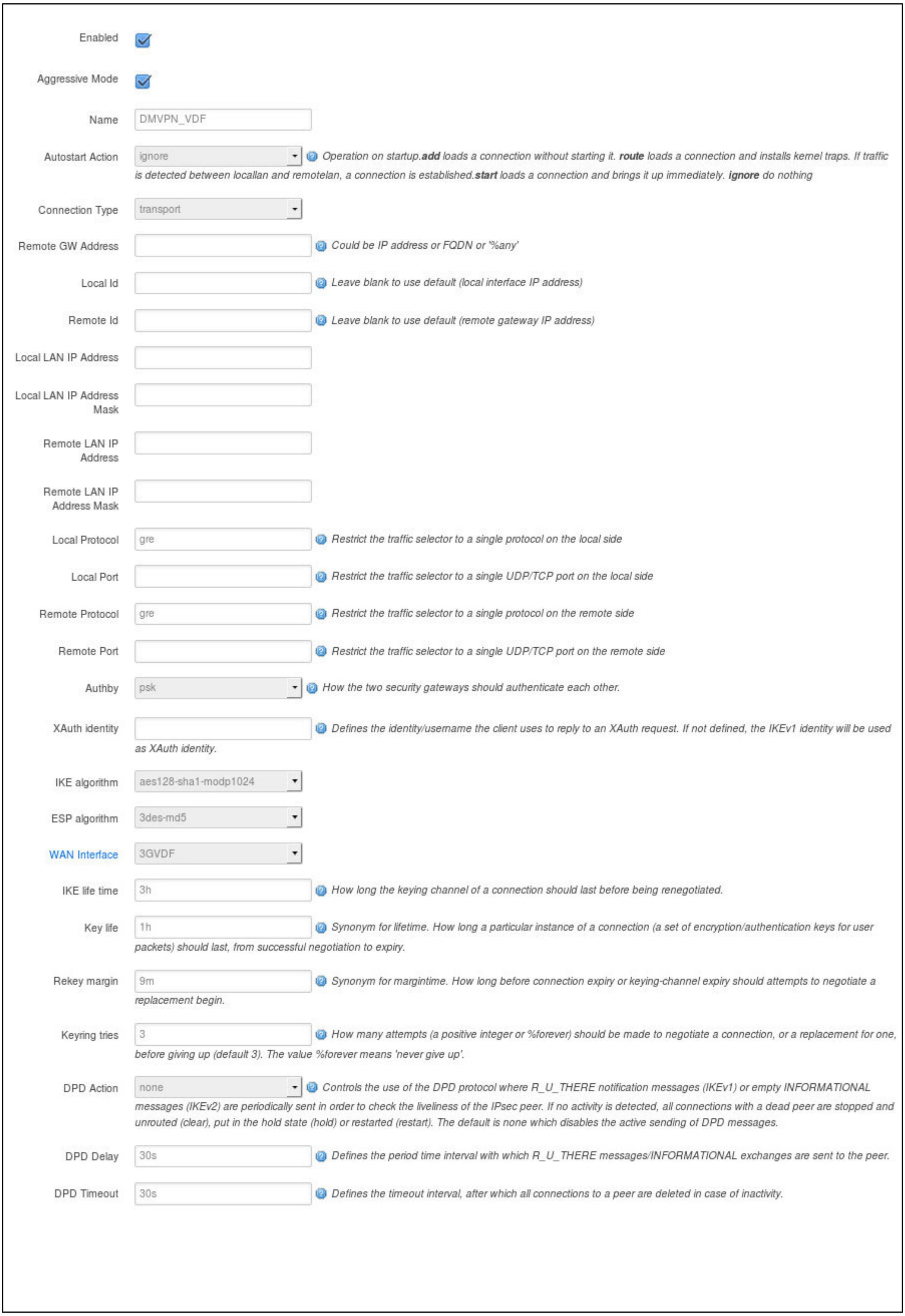


Figure 149: The connections settings section

Web Field / UCI / Package Option	Description
Web: Enabled UCI: strongswan.@connection[X].enabled Opt: enable	Enables or disables IPSec connection. 0 — Disabled. 1 — Enabled.
Web: Aggressive UCI: strongswan.@connection[X].aggressive Opt: aggressive	Enables or disables IKE aggressive mode. **Note**: using aggressive mode along with PSK authentication is less secure method than main mode and should be avoided. 0 — Disabled. 1 — Enabled.
Web: Name UCI: strongswan.@connection[X].name Opt: name	Specifies a name for the tunnel.
Web: Autostart Action UCI: strongswan.@connection[X].auto Opt: auto	Specifies when the tunnel is initiated. start — On start up. route — When traffic routes this way. add — Loads a connection without starting it. ignore — Ignores the connection. always — Actively retries to establish the tunnel if it went down.
Web: Connection Type UCI: strongswan.@connection[X].type Opt: type	Defines the type of IPSec connection. tunnel — Connection uses tunnel mode. transport — Connection uses transport mode. pass — Connection does not perform any IPSec processing. drop — Connection drops all the packets.
Web: Remote GW Address UCI: strongswan.@connection[X].remoteaddress Opt: remoteaddress	Sets the public IP address of the remote peer. Leave blank for DMVPN.
Web: Local ID UCI: strongswan.@connection[X].localid Opt: localid	Defines the local peer identifier. Leave blank for DMVPN.
Web: Remote ID UCI: strongswan.@connection[X].remoteid Opt:remoteid	Defines the remote peer identifier. Leave blank for DMVPN.
Web: Local LAN IP Address UCI: strongswan.@connection[X]. locallan Opt: locallan	Defines the local IP of LAN. Leave blank for DMVPN.
Web: Local LAN IP Address Mask UCI: strongswan.@connection[X].locallanmask Opt: locallanmask	Defines the subnet of local LAN. Leave blank for DMVPN.
Web: Remote LAN IP Address UCI: strongswan.@connection[X].remotelan Opt:remotelan	Defines the IP address of LAN serviced by remote peer. Leave blank for DMVPN.
Web: Remote LAN IP Address Mask UCI: strongswan.@connection[X].remotelanmask Opt:remotelanmask	Defines the Subnet of remote LAN. Leave blank for DMVPN.
Web: Local Protocol UCI: strongswan.@connection[X].localproto Opt: localproto	Restricts the connection to a single protocol on the local side.

Web: Local Port UCI: strongswan.@connection[X].localport Opt: localport	Restricts the connection to a single port on the local side.
Web: Remote Protocol UCI: strongswan.@connection[X].remoteproto Opt:remoteproto	Restricts the connection to a single protocol on the remote side.
Web: Remote Port UCI: strongswan.@connection[X].remoteport Opt: remoteport	Restricts the connection to a single port on the remote side.
Web: Authby UCI: strongswan.@connection[X].authby Opt: authby	Defines how the two secure gateways should authenticate. **Note**: using aggressive mode along with PSK authentication is unsecure and should be avoided. Pubkey: For public key signatures. Rsasig: For RSA digital signatures. ecdsasig: For Elliptic Curve DSA signatures. Psk: Using a preshared key. xauthrsasig: Enables eXtended Authentication (XAuth) with addition to RSA signatures. xauthpsk: Using extended authentication and preshared key. never: Can be used if negotiation is never to be attempted or accepted (shunt connections).
Web: XAuth Identity UCI: strongswan.@connection[X].xauth_identity Opt: xauth_identity	Defines Xauth ID.
Web: IKE Algorithm UCI: strongswan.@connection[X].ike Opt: ike	Specifies the IKE algorithm to use. The format is: encAlgo | authAlgo | DHGroup: encAlgo: 3des aes128 aes256 serpent twofish blowfish authAlgo: md5 sha sha2 DHGroup: modp1024 modp1536 modp2048 modp3072 modp4096 modp6144 modp8192 For example, a valid IKE algorithm is: aes128-sha-modp1536.

<table>
<tr><td>Web: ESP algorithm
UCI: strongswan.@connection[X].esp
Opt: esp</td><td>Specifies the esp algorithm to use.
The format is: encAlgo | authAlgo | DHGroup
encAlgo:
3des
aes128
aes256
serpent
twofish
blowfish
authAlgo:
md5
sha
sha2
DHGroup:
modp1024
modp1536
modp2048
modp3072
modp4096
modp6144
modp8192
For example, a valid encryption algorithm is:
aes128-sha-modp1536.
If no DH group is defined then PFS is disabled.</td></tr>
<tr><td>Web: WAN Interface
UCI: strongswan.@connection[X].waniface
Opt: waniface</td><td>This is a space separated list of the WAN interfaces the router will use to establish a tunnel with the secure gateway.
On the web, a list of the interface names is automatically generated. If you want to specify more than one interface use the “custom” value.
Example: If you have a 3G WAN interface called ‘wan and a WAN ADSL interface called ‘dsl’ and wanted to use one of these interfaces for this IPSec connection, you would use: ‘wan adsl’.</td></tr>
<tr><td>Web: IKE Life Time
UCI: strongswan.@connection[X].ikelifetime
Opt:ikelifetime</td><td>Specifies how long the keyring channel of a connection (ISAKMP or IKE SA) should last before being renegotiated.<table><tr><td>3h</td><td></td></tr><tr><td>Timespec</td><td>1d, 3h, 25m, 10s.</td></tr></table></td></tr>
<tr><td>Web: Key Life
UCI: strongswan.@connection[X].keylife
Opt: keylife</td><td>Specifies how long a particular instance of a connection (a set of encryption/authentication keys for user packets) should last, from successful negotiation to expiry.
Normally, the connection is renegotiated (via the keying channel) before it expires (see rekeymargin).<table><tr><td>1h</td><td></td></tr><tr><td>Timespec</td><td>1d, 1h, 25m, 10s.</td></tr></table></td></tr>
<tr><td>Web: Rekey Margin
UCI: strongswan.@connection[X].rekeymargin
Opt: rekeymargin</td><td>Specifies how long before connection expiry or keying-channel expiry should attempt to negotiate a replacement begin.
Relevant only locally, other end need not agree on it.<table><tr><td>9m</td><td></td></tr><tr><td>Timespec</td><td>1d, 2h, 9m, 10s.</td></tr></table></td></tr>
<tr><td>Web: Keyring Tries
UCI: strongswan.@connection[X].keyringtries
Opt: keyringtries</td><td>Specifies how many attempts (a positive integer or %forever) should be made to negotiate a connection, or a replacement for one, before giving up. The value %forever means 'never give up'. Relevant only locally, other end need not agree on it.</td></tr>
</table>

Web: DPD Action UCI: strongswan.@connection[X].dpdaction Opt: dpdaction	Defines DPD (Dead Peer Detection) action. None — Disables DPD. Clear — Clear down the tunnel if peer does not respond. Reconnect when traffic brings the tunnel up. Hold — Clear down the tunnel and bring up as soon as the peer is available. Restart — Restarts DPD when no activity is detected.
Web: DPD Delay UCI: strongswan.@connection[X].dpddelay Opt: dpddelay	Defines the period time interval with which R_U_THERE messages and INFORMATIONAL exchanges are sent to the peer. These are only sent if no other traffic is received. 30s Timespec — 1d, 2h, 25m, 10s.
Web: DPD Timeout UCI: strongswan.@connection[X].dpdtimeout Opt: dpdtimeout	Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity. 150s Timespec — 1d, 2h, 25m, 10s.

Table 97: Information table for IPSec connections settings

29.4.3 Configure secrect settings

Each tunnel requires settings to configure how the local end point of the tunnel proves its identity to the remote end point.

Figure 150: IPSec secrets settings

Web Field / UCI / Package Option	Description
Web: Enabled UCI: strongswan.@secret[X].enabled Opt: enabled	Defines whether this set of credentials is to be used or not. 0 — Disabled. 1 — Enabled.
Web: ID selector UCI: strongswan.@secret[X].idtype Opt: idtype	Defines whether IP address or userfqdn is used.
Web: ID selector UCI: strongswan.@secret[X].localaddress Opt: localaddress	Defines the local address this secret applies to.
Web: ID selector UCI: strongswan.@secret[X].remoteaddress Opt: remoteaddress	Defines the remote address this secret applies to.

<table>
<tr><td>Web: N/A
UCI: strongswan.@secret[X].userfqnd
Opt: userfqnd</td><td>FQDN or Xauth name used of Extended Authentication. This must match xauth_identity from the configuration connection section.</td></tr>
<tr><td>Web: Secret Type
UCI: strongswan.@secret[X].secrettype
Opt: secrettype</td><td>Specifies the authentication mechanism to be used by the two peers.<table><tr><td>Psk</td><td>Preshared secret</td></tr><tr><td>Pubkey</td><td>Public key signatures</td></tr><tr><td>Rsasig</td><td>RSA digital signatures</td></tr><tr><td>Ecdsasig</td><td>Elliptic Curve DSA signatures</td></tr><tr><td>Xauth</td><td>Extended authentication</td></tr></table></td></tr>
<tr><td>Web: Secret
UCI: strongswan.@secret[X].secret
Opt: secret</td><td>Defines the secret.</td></tr>
</table>

Table 98: Information table for IPSec secret settings

29.5 Configuring an IPSec template to use with DMVPN

The following example shows how to configure an IPSec connection template to use with DMVPN.

```
# Commands
touch /etc/config/strongswan
uci set strongswan.general=general
uci set strongswan.general.enabled=yes
uci set strongswan.general.strictcrlpolicy=no
uci set strongswan.general.uniqueids=yes
uci set strongswan.general.cachecrls=yes
uci set strongswan.general.nattraversal=yes
uci add strongswan connection
uci set strongswan.@connection[0].enabled=yes
uci set strongswan.@connection[0].name=dmvpn
uci set strongswan.@connection[0].type=transport
uci set strongswan.@connection[0].localproto=gre
uci set strongswan.@connection[0].remoteproto=gre
uci set strongswan.@connection[0].ike=aes-sha1-modp1024
uci set strongswan.@connection[0].esp=aes128-sha1
uci set strongswan.@connection[0].waniface=lan4
uci set strongswan.@connection[0].auto=ignore
uci set strongswan.@connection[0].ikelifetime=28800s
uci set strongswan.@connection[0].keylife=300s
uci set strongswan.@connection[0].rekeymargin=30s
uci set strongswan.@connection[0].keyingtries=%forever
uci set strongswan.@connection[0].dpdaction=hold
```

```
uci set strongswan.@connection[0].dpddelay=30s
uci set strongswan.@connection[0].dpdtimeout=150s
uci add strongswan secret
uci set strongswan.@secret[0].enabled=yes
uci set strongswan.@secret[0].secrettype=psk
uci set strongswan.@secret[0].secret=secret
```

This will create package strongswan.

```
config general 'general'
option enabled 'yes'
option strictcrlpolicy 'no'
option uniqueids 'yes'
option cachecrls 'yes'
option nattraversal 'yes'
 config connection
option enabled 'yes'
option name 'dmvpn'
option type 'transport'
option localproto 'gre'
option remoteproto 'gre'
option ike 'aes-sha1-modp1024'
option esp 'aes128-sha1'
option waniface 'lan4'
option auto 'ignore'
option ikelifetime '28800s'
option keylife '300s'
option rekeymargin '30s'
option keyingtries '%forever'
option dpdaction 'hold'
option dpddelay '30s'
option dpdtimeout '150s'
config secret
option enabled 'yes'
option secrettype 'psk'
option secret 'secret'
```

29.6 IPSec diagnostics using the web interface

29.6.1 IPSec status

In the top menu, click **Status -> IPSec**. The IPSec Connections page appears.

IPsec Connections

Name	IKE					SA			
	Status	Remote	Established	Encryption	Integrity	Status	Policy	Data In/Out	Rekey in
dmvpn_213_233_148_2	ESTABLISHED	213.233.148.2	2 hours ago	3DES_CBC	HMAC_MD5_96	INSTALLED			
dmvpn_89_101_154_151	ESTABLISHED	89.101.154.151	2 hours ago	3DES_CBC	HMAC_MD5_96	INSTALLED			

Figure 151: The IPSec connections page

In the Name column, the syntax contains the IPSec Name defined in package dmvpn and the remote IP address of the hub, or the spoke separated by an underscore; for example, dmvpn_213.233.148.2.

29.7 IPSec diagnostics using UCI

29.7.1 IPSec configuration

To view IPSec configuration via UCI, enter:

```
root@VA_router:~# uci export strongswan
```

To restart strongSwan, enter:

```
root@VA_router:~# etc/init.d/strongswan restart
```

29.7.2 IPSec status

29.7.3 To view IPSec status, enter:

```
root@VA_router:~# ipsec statusall
Security Associations (1 up, 0 connecting):
dmvpn_89_101_154_151[1]: ESTABLISHED 2 hours ago,
10.68.234.133[10.68.234.133]...89.101.154.151[89.101.154.151]
dmvpn_89_101_154_151{1}:  REKEYING, TRANSPORT, expires in 55 seconds
dmvpn_89_101_154_151{1}:   10.68.234.133/32[gre] === 192.168./32[gre]
dmvpn_89_101_154_151{1}:  INSTALLED, TRANSPORT, ESP in UDP SPIs: cca7b970_i
d874dc90_o
dmvpn_89_101_154_151{1}:   10.68.234.133/32[gre] === 89.101.154.151/32[gre]
```

To view a list of IPSec commands, enter:

```
root@VA_router:~# ipsec –help
```

30 Dynamic Multipoint Virtual Private Network (DMVPN)

Dynamic Multipoint Virtual Private Network (DMVPN) is a scalable method of creating VPN IPSec Networks. DMVPN is a suite of three protocols: NHRP, GRE and IPSec, used to dynamically create VPN tunnels between different endpoints in the network without having to pre-configure each device with VPN details of the rest of endpoints in the network.

30.1 Prerequisites for configuring DMVPN

Before configuring DMVPN, you must first configure:

- A GRE interface; the previous chapter,' Configuring GRE interfaces'
- An IPSec connection to use as a template; read the chapter, 'Configuring IPSec'.

30.2 Advantages of using DMVPN

Using DMVPN eliminates the need of IPSec configuration to the physical interface. This reduces the number of lines of configuration required for a VPN development. For example, for a 1000-site deployment, DMVPN reduces the configuration effort at the hub from 3900 lines to 13.

- Adding new peers (spokes) to the VPN requires no changes at the hub.
- Better scalability of the network.
- Dynamic IP addresses can be used at the peers' site.
- Spokes can be connected in private or public network.
- NHRP NAT extension allows spoke-to-spoke tunnels to be built, even if one or more spokes is behind a Network Address Translation (NAT) device.
- New hubs can be added to the network to improve the performances and reliability.
- Ability to carry multicast and main routing protocols traffic (RIP, OSPF, BGP).
- DMVPN can be deployed using Activator, the Virtual Access automated provisioning system.
- Simplifies branch communications by enabling direct branch to branch connectivity.
- Simplifies configuration on the spoke routers. The same IPSec template configuration is used to create spoke-to-hub and spoke-to-spoke VPN IPSec tunnel.
- Improves business resiliency by preventing disruption of business-critical applications and services by incorporating routing with standards-based IPsec technology.

30.3 DMVPN scenarios

30.3.1 Scenario 1

Spoke1, spoke2 and a hub are in the same public or private network.

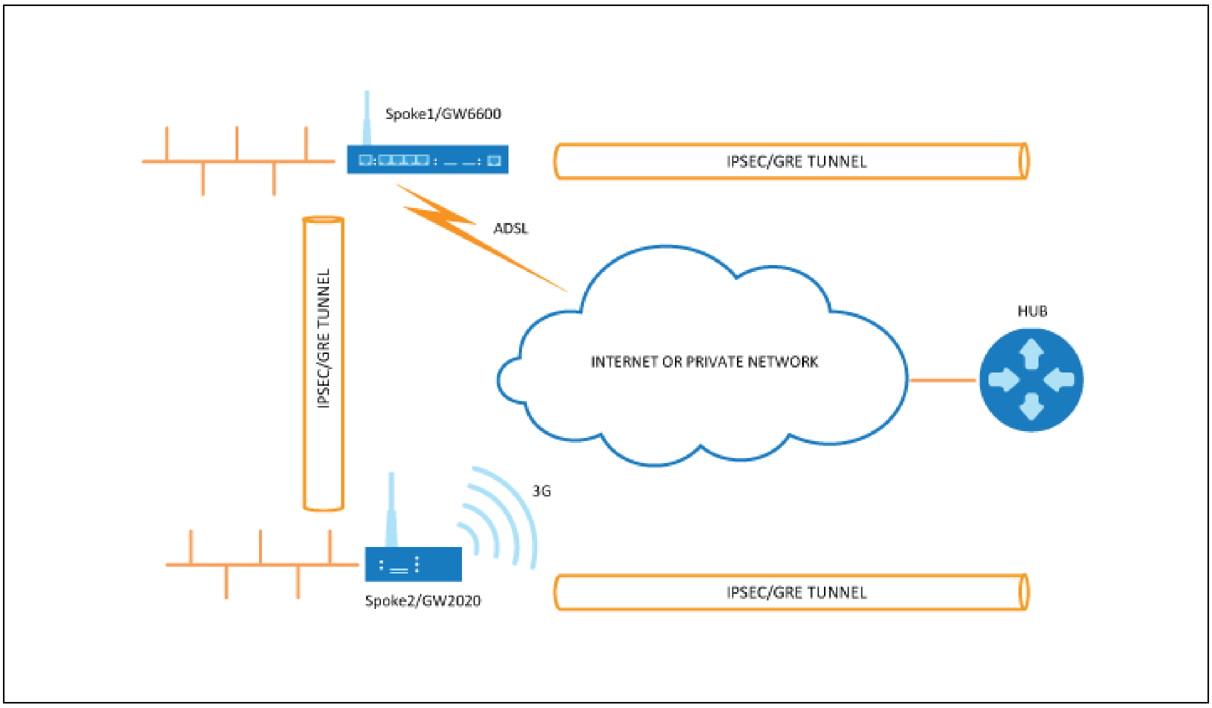


Figure 152: Network diagram for DMVPN spoke to spoke

- Spoke1 and spoke2 connect on their WAN interface: ADSL, 3G and initiate main mode IPSec in transport mode to the hub.
- After an IPSec tunnel is established, spokes register their NHRP membership with the hub.
- GRE tunnels come up.
- Hub caches the GRE tunnel and real IP addresses of each spoke.
- When spoke1 wants to talk to spoke2, it sends an NHRP resolution request to the hub.
- The hub checks its cache table and forwards that request to spoke2.
- Spoke2 caches spoke1's GRE and real IP address and sends an NHRP resolution reply via the hub.
- Spoke1 receives an NHRP resolution reply and updates its NHRP table with spoke2 information. Then it initiates VPN IPSec connection to spoke2.
- When an IPSec tunnel is established, spoke1 and spoke2 can send traffic directly to each other.

30.3.2 Scenario 2

Spoke1 is in a private (NAT-ed) network, spoke2 and hub are in public network.

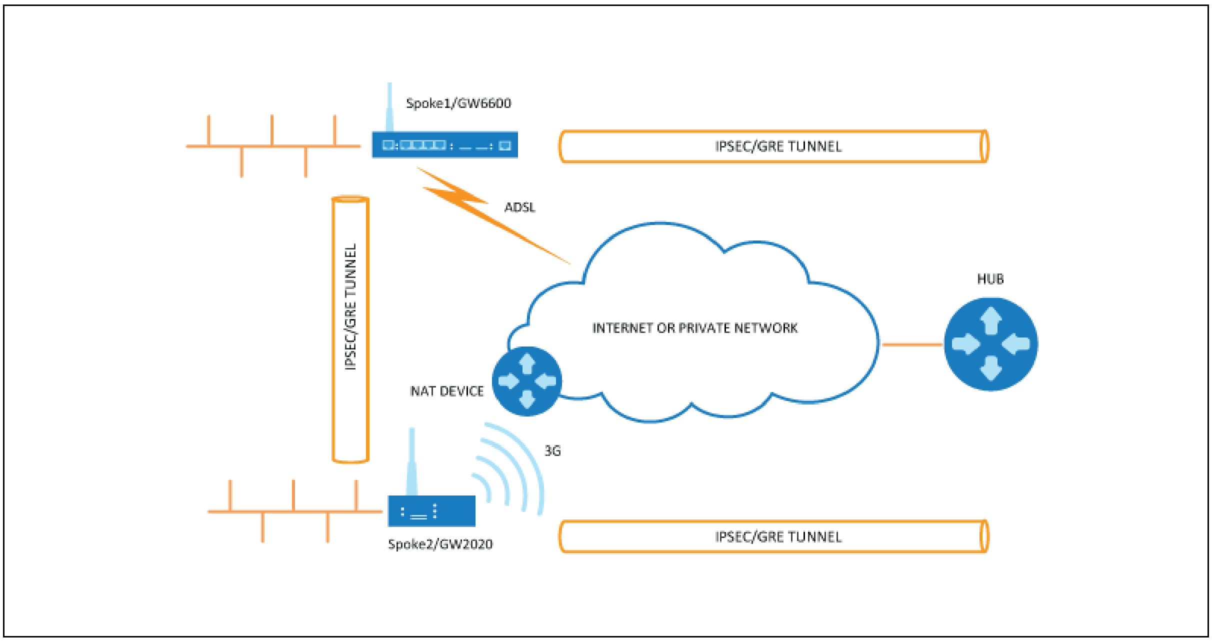


Figure 153: Network diagram for DMVPN spoke behind NAT

- Spoke1 sends an NHRP registration request to the hub.
- Hub receives this request and compares the source tunnel address of the spoke with the source of the packet.
- Hub sends an NHRP registration reply with a NAT extension to spoke1.
- The NAT extension informs spoke1 that it is behind the NAT-ed device.
- Spoke1 registers its pre- and post-NAT address.
- When spoke1 wants to talk to spoke2, it sends an NHRP resolution request to the hub.
- Hub checks its cache table and forwards that request to spoke2.
- Spoke2 caches spoke1's GRE pre- and post-NAT IP address and sends an NHRP resolution reply via the hub.
- Spoke1 receives the NHRP resolution reply and updates its NHRP table with spoke2 information. It initiates a VPN IPSec connection to spoke2.
- When the IPSec tunnel is established, spoke1 and spoke2 can send traffic directly to each other.

Note: if an IPSec tunnel fails to be established between the spokes then packets between the spokes are sent via the hub.

30.4 Configuration packages used

Package	Sections
network	For configuring the GRE tunnels.
strongswan	For enabling and configuring the IPSec connection template
dmvpn	

30.5 Configuring DMVPN using the web interface

The DMVPN section contains fields required to configure the parameters relative to the DMVPN Hub. These are used for DMVPN tunnels, such as GRE tunnels, GRE tunnel remote IP, DMVPN Hub IP and password.

30.5.1 DMVPN general settings

In the top menu, select **Network -> DMVPN**. The DMVPN page appears. There are two sections: General and DMVPN Hub Settings.

Figure 154: The DMVPN general section

Web Field / UCI / Package Option	Description
Web: Enable DMVPN UCI: dmvpn.common.enabled Opt: enable	Enables DMVPN. 0 — Disabled. 1 — Enabled.
Web: IPSec template connection UCI: dmvpn.common.ipsec_template_name Opt: ipsec_template_name	Selects the IPSec connection, defined in strongSwan, to be used as a template.

Table 99: Information table for DMVPN general settings

30.5.2 DMVPN hub settings

Figure 155: The DMVPN hub settings

Web Field / UCI / Package Option	Description
Web: GRE Interface UCI: dmvpn.@interface[X].gre_interface Opt: gre_interface	Specifies which GRE interface will be used with this DMVPN configuration.
Web: GRE Remote Endpoint IP Address UCI: dmvpn.@interface[X].gre_endpoint_ip Opt: gre_endpoint_ip	Configures the GRE IP address of the hub.
Web: GRE Remote Endpoint Mask Length UCI: dmvpn.@interface[X].gre_endpoint_mask_length Opt: gre_endpoint_mask_length	Configures the length of the mask of the GRE interface on the hub. For example if the mask is 255.255.0.0 the length will be 16.
Web: DMVPN Hub IP Address UCI: dmvpn.@interface[X].nhs_ip Opt: nhs_ip	Configures the physical IP address for the DMVPN hub.
Web: NHRP Authentication UCI: dmvpn.@interface[X].cisco_auth Opt: cisco_auth	Enables authentication on NHRP. The password will be applied in plaintext to the outgoing NHRP packets. Maximum length is 8 characters.
Web: NHRP Holding Time UCI: dmvpn.@interface[X].holding_time Opt: holding_time	Timeout for cached NHRP requests.
Web: Use As Default Route UCI : dmvpn.@interface[X].defaultroute Opt: defaultroute	Adds a default route into tunnel interface. 0 — Disabled. 1 — Enabled.
Web: Default Route Metric UCI: dmvpn.@interface[X].defaultroutemetric Opt: defaultroutemetric	Metric to use for the default route.
Web: LED state indication UCI: dmvpn.@interface[X].led Opt: led	LED to use for indicating if the VPN is up.

Table 100: Information table for DMVPN hub settings

30.5.3 Configuring an IPSec template for DMVPN using the web interface

Configuring an IPSec template is covered in the chapter 'Configuring IPSec'.

30.6 DMVPN diagnostics

In the top menu, click **Status -> IPSec**. The IPSec Connections page appears.

IPsec Connections

Name	IKE					SA			
	Status	Remote	Established	Encryption	Integrity	Status	Policy	Data In/Out	Rekey in
dmvpn_213_233_148_2	ESTABLISHED	213.233.148.2	2 hours ago	3DES_CBC	HMAC_MD5_96	INSTALLED			
dmvpn_89_101_154_151	ESTABLISHED	89.101.154.151	2 hours ago	3DES_CBC	HMAC_MD5_96	INSTALLED			

Figure 156: The IPSec connections page

In the Name column, the syntax contains the IPSec name defined in package dmvpn and the remote IP address of the hub, or the spoke separated by an underscore; for example, dmvpn_213.233.148.2.

To check the status of DMVPN, in the top menu, click **Status -> DMVPN**.

NBMA peers

NBMA Address	Interface	Address	Type
213.233.148.2	GRE	11.11.11.3/32	spoke
89.101.154.151	GRE	11.11.11.1/29	hub

Powered by LuCI Trunk (trunk+svn8382) VIE-16.00.28 image1 config2

Figure 157: The NBMA peers page

To check DMVPN status, enter:

```
:~# opennhrpctl show
Status: ok
Interface: gre-GRE
Type: local
Protocol-Address: 11.11.11.7/32
Alias-Address: 11.11.11.3
Flags: up
Interface: gre-GRE
Type: local
Protocol-Address: 11.11.11.3/32
Flags: up
Interface: gre-GRE
Type: cached
Protocol-Address: 11.11.11.2/32
NBMA-Address: 178.237.115.129
NBMA-NAT-OA-Address: 172.20.38.129
```

```
Flags: used up
Expires-In: 0:18

Interface: gre-GRE
Type: static
Protocol-Address: 11.11.11.1/29
NBMA-Address: 89.101.154.151
Flags: up
```

Interface	Description
Type	incomplete: Resolution request sent. negative: Negative cached. cached: Received/relayed resolution reply. shortcut_route: Received/relayed resolution for route. dynamic: NHC resolution. dynamic_nhs: Dynamic NHS from dns-map. static: Static mapping from config file. dynamic_map: Static dns-map from config file. local_route: Non-local destination, with local route. local_addr: Local destination (IP or off-NBMA subnet).
Protocol Address	Tunnel IP address
NBMA-Address	Pre-NAT IP address if NBMA-NAT-OA-Address is present or real address if NAT is not present.
NBMA-NAT-OA-Address	Post NAT IP address. This field is present when Address is translated in the network.
Flags	up: Can send all packets (registration ok). unique: Peer is unique. used: Peer is kernel ARP table. lower-up: openhrp script executed successfully.
Expires-In	Expiration time.

Table 101: Information table for DMVPN status

You can check IPSec status using UCI commands.

```
root@VA-router:~# ipsec status
Security Associations (1 up, 0 connecting):
dmvpn_89_101_154_151[1]: ESTABLISHED 2 hours ago,
10.68.234.133[10.68.234.133]...89.101.154.151[89.101.154.151]
dmvpn_89_101_154_151{1}:  REKEYING, TRANSPORT, expires in 55 seconds
dmvpn_89_101_154_151{1}:  10.68.234.133/32[gre] === 192.168./32[gre]
dmvpn_89_101_154_151{1}:  INSTALLED, TRANSPORT, ESP in UDP SPIs: cca7b970_i
d874dc90_o
dmvpn_89_101_154_151{1}:  10.68.234.133/32[gre] === 89.101.154.151/32[gre]
```

You can check DMVPN status using UCI commands.

```
:~# opennhrpctl show
Status: ok

Interface: gre-GRE
Type: local
Protocol-Address: 11.11.11.7/32
Alias-Address: 11.11.11.3
Flags: up

Interface: gre-GRE
Type: local
Protocol-Address: 11.11.11.3/32
Flags: up
Interface: gre-GRE
Type: cached
Protocol-Address: 11.11.11.2/32
NBMA-Address: 178.237.115.129
NBMA-NAT-OA-Address: 172.20.38.129
Flags: used up
Expires-In: 0:18
Interface: gre-GRE
Type: static
Protocol-Address: 11.11.11.1/29

NBMA-Address: 89.101.154.151
Flags: up
```

31 Configuring multicasting using PIM and IGMP interfaces

31.1 Overview

IP multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to potentially thousands of corporate recipients. Applications that take advantage of multicast include video conferencing and corporate communications.

IP multicast delivers application source traffic to multiple receivers without burdening the source or the receivers while using a minimum of network bandwidth.

PIM (Protocol Independent Multicast) and IGMP (Internet Group Management Protocol) are protocols used to create multicasting networks within a regular IP network.

A multicast group is an arbitrary group of receivers that expresses an interest in receiving a particular data stream. The receivers (the designated multicast group) are interested in receiving a data stream from the source. They indicate this by sending an Internet Group Management Protocol (IGMP) host report to their closest router in the network. The routers are then responsible for delivering the data from the source to the receivers. The routers use Protocol Independent Multicast (PIM) between themselves to dynamically create a multicast distribution tree. The data stream will then be delivered only to the network segments that are in the path between the source and the receivers.

To summarize: PIM is used between routers while IGMP is used between a receiver and its router only. As a result, PIM must be enabled on all the interfaces on the route from the multicast source to the multicast client while IGMP must be enabled on the interface to the multicast client only.

31.2 Configuration package used

Package	Sections
pimd	pimd interface

31.3 Configuring PIM and IGMP using the web interface

To configure PIM through the web interface, in the top menu, select **Network -> PIM**. The PIM page appears. To access the Global settings, click **Add**.

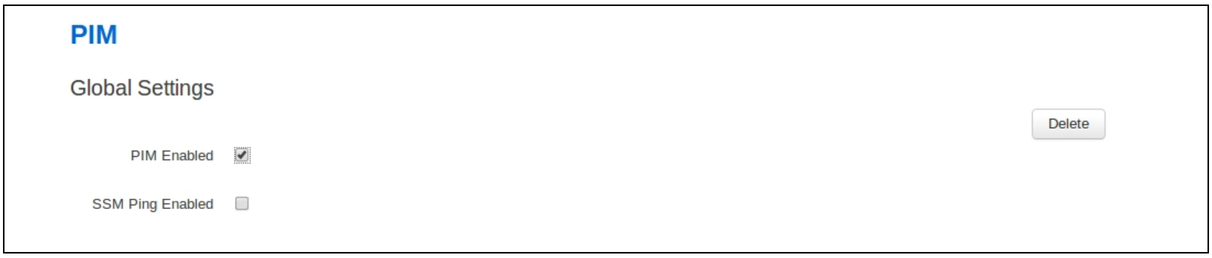


Figure 158: The global settings interface

31.3.1 Global settings

Web Field / UCI / Package Option	Description
Web: PIM Enabled UCI: pimd.pimd.enabled Opt: enabled	Globally enables PIM on the router. 0 — Disabled. 1 — Enabled.
Web: SSM Ping Enabled UCI: pimd.pimd.ssmpingd Opt: ssmpingd	Enables answers to SSM pings. 0 — Disabled. 1 — Enabled.

Table 102: Information table for PIM global settings

31.3.2 Interfaces configuration

Figure 159: The interfaces configuration section

Web Field / UCI / Package Option	Description
Web: Enabled UCI: pimd.interface[x].enabled Opt: enabled	Enables multicast management of the given interface by the PIM application. 0 — Disabled. 1 — Enabled.
Web: Interface UCI: pimd.interface[x].interface Opt: interface	Selects the interface to apply PIM settings to.
Web: Enable IGMP UCI: pimd.interface[x].igmp Opt: igmp	Enable IGMP on given interface. 0 — Disabled. 1 — Enabled. **Note**: you must enable PIM SSM and/or IGMP depending on your requirements. ICMP must be enabled on the interface to the multicast client only.
Web: Enable SSM UCI: pimd.interface[x].ssm Opt: ssm	Enable SSM on given interface. 0 — Disabled. 1 — Enabled.

Table 103: Information table for interface settings

To save your configuration updates, click **Save & Apply**.

31.4 Configuring PIM and IGMP using UCI

You can configure PIM and IGMP through CLI using UCI.

The configuration file is stored on **/etc/config/pimd**

To view the configuration file, enter:

```
uci export pimd
root@VA_router:/etc/config1# uci export pimd
package pimd
config routing 'pimd'
        option enabled 'yes'

config interface
        option enabled 'yes'
        option interface 'lan'
        option ssm 'yes'
        option igmp 'yes'

config interface
        option enabled 'yes'
        option interface 'wan'
        option ssm 'yes'
        option igmp 'no'

Alternatively, enter:
uci show pimd
root@VA_router:/etc/config1# uci show pimd
pimd.pimd=routing
pimd.pimd.enabled=yes
pimd.@interface[0]=interface
pimd.@interface[0].enabled=yes
pimd.@interface[0].interface=lan
pimd.@interface[0].ssm=yes
pimd.@interface[0].igmp=yes
pimd.@interface[1]=interface

pimd.@interface[1].enabled=yes
pimd.@interface[1].interface=wan
```

```
pimd.@interface[1].ssm=yes
pimd.@interface[1].igmp=no
```

To change any of the above values use `uci set` command.

32 QoS: VLAN 802.1Q PCP tagging

32.1 Configuring VLAN PCP tagging

Virtual Access routers have the capability to respect and set PCP priority values inside 802.1Q VLAN tagged frames. The following partial export of network configuration shows how to configure VLAN priorities for specific interfaces (VLANs).

```
root@VA_router:~# uci export network package network
config va_switch
        option eth0 'A E'
        option eth1 'B F'
        option eth2 'C G'
        option eth3 'D'
        option eth4 'H'

config interface 'VLAN_1'
        option type 'bridge'
        option proto 'static'
        option ipaddr '10.1.28.99'
        option netmask '255.255.0.0'
        option ifname 'eth0 eth4'

config interface 'VLAN_2'
        option type 'bridge'
        option proto 'static'
        option ipaddr '192.168.2.1'
        option netmask '255.255.255.0'
        option ifname 'eth1 eth4.2'
        option vlan_qos_map_ingress '1:1'
        option vlan_qos_map_egress '0:1'

config interface 'VLAN_3'
        option ifname 'eth2 eth4.3'
        option type 'bridge'
        option proto 'static'
        option ipaddr '192.168.3.1'
        option netmask '255.255.255.0'
```

```
        option vlan_qos_map_ingress '3:3'
        option vlan_qos_map_egress '0:3'

config interface 'VLAN_4'
        option ifname 'eth3 eth4.4'
        option type 'bridge'
        option proto 'static'
        option ipaddr '192.168.3.1'
        option netmask '255.255.255.0'
        option vlan_qos_map_ingress '5:5'
        option vlan_qos_map_egress '0:5'
```

UCI / Package Option	Description
UCI: network.<if name>.vlan_qos_map_ingress Opt: list vlan_qos_map_ingress	VLAN priority code point to socket buffer mapping. Example: network.<if name>. vlan_qos_map_ingress =1:1
UCI: network.<if name>.vlan_qos_map_egress Opt: list vlan_qos_map_egress	Socket buffer to VLAN priority code point mapping. Example: network.<if name>. vlan_qos_map_egress =0:1

The above sample configuration specifies that any frames on VLAN2, VLAN3 and VLAN4 will be processed or have their PCP value adjusted according to QoS values set.

VLAN1

- VLAN1 is an untagged VLAN so there are no 802.1Q tags on the frames.

VLAN2

- Any frames received on VLAN2 destined to VLAN2 with PCP priority of 1 will be forwarded without altering the priority; it will be still set to 1.
- Any frames received on VLAN2 destined to VLAN2 with a PCP priority set to 0 will have a priority of 1 set as they leave the router on VLAN2.

VLAN3

- Any frames received on VLAN3 destined to VLAN3 with a PCP priority of 3 will be forwarded without altering the priority; it will be still set to 3.
- Any frames received on VLAN3 destined to VLAN2 with PCP priority set to 0 will have a priority of 3 set as they leave the router on VLAN3.

VLAN4

- Any frames received on VLAN4 destined to VLAN2 with PCP priority of 5 will be forwarded without altering the priority; it will be still set to 5.
- Any frames received on VLAN4 destined to VLAN2 with PCP priority set to 0 will have a priority of 5 set as they leave the router on VLAN4.

Four queues are supported and are structured as follows:

- Queue 1: PCP values 0 and 1 - Default
- Queue 2: PCP values 2 and 3 - Normal
- Queue 3: PCP values 4 and 5 - High
- Queue 4: PCP values 6 and 7 - Express

Value 7 is the highest priority and 0 is the lowest. These queues prioritise 802.1Q tagged frames as they are received on the port, these are hardware defined.

When 802.1Q frames are received on the port they are processed according to the above queues on arrival (even if not defined in the configuration). Then if value 'vlan_qos_map_ingress' is configured you can modify the PCP priority for egress if the frame was to be forwarded on another tagged interface.

When frames are received on an untagged VLAN interface configured with 'vlan_qos_map_egress' and are destined to tagged interface, 802.1Q tag will be created with a default priority of 0 and then the priority will be set according to the PCP value specified as the frames leave port.

33 QoS: type of service

Virtual Access routers are capable of implementing quality of service configurations on a per interface basis, which allows traffic prioritisation based on type of service criteria parameters.

33.1 QoS configuration overview

A minimal QoS configuration usually consists of:

- One interface section
- Some rules allocating packets to at least two buckets
- Configuration of the buckets

33.2 Configuration packages used

Package	Sections
qos	interface
	classgroup
	class
	classify

33.3 Configuring QoS using the web interface

Browse to the router's IP address and login.

Select **Network tab -> QoS**. The QoS page appears. From this page you can configure interfaces that QoS is applied to as well as classification rules.

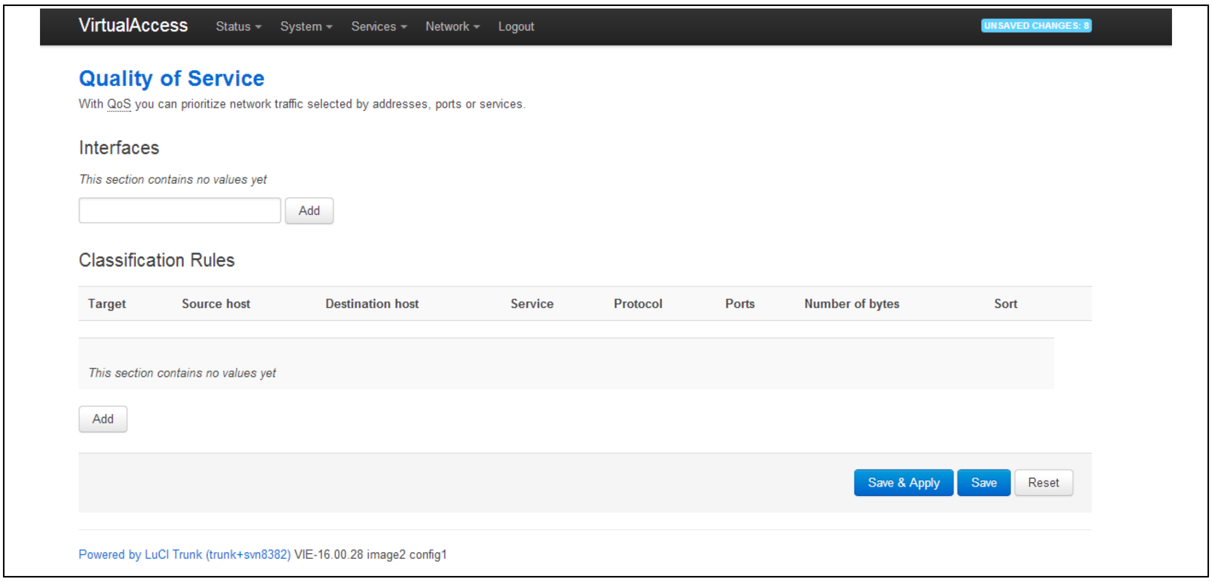


Figure 160: The quality of service page

To configure an interface, enter a relevant interface name and click **Add**. The Quality of Service page for that interface appears.

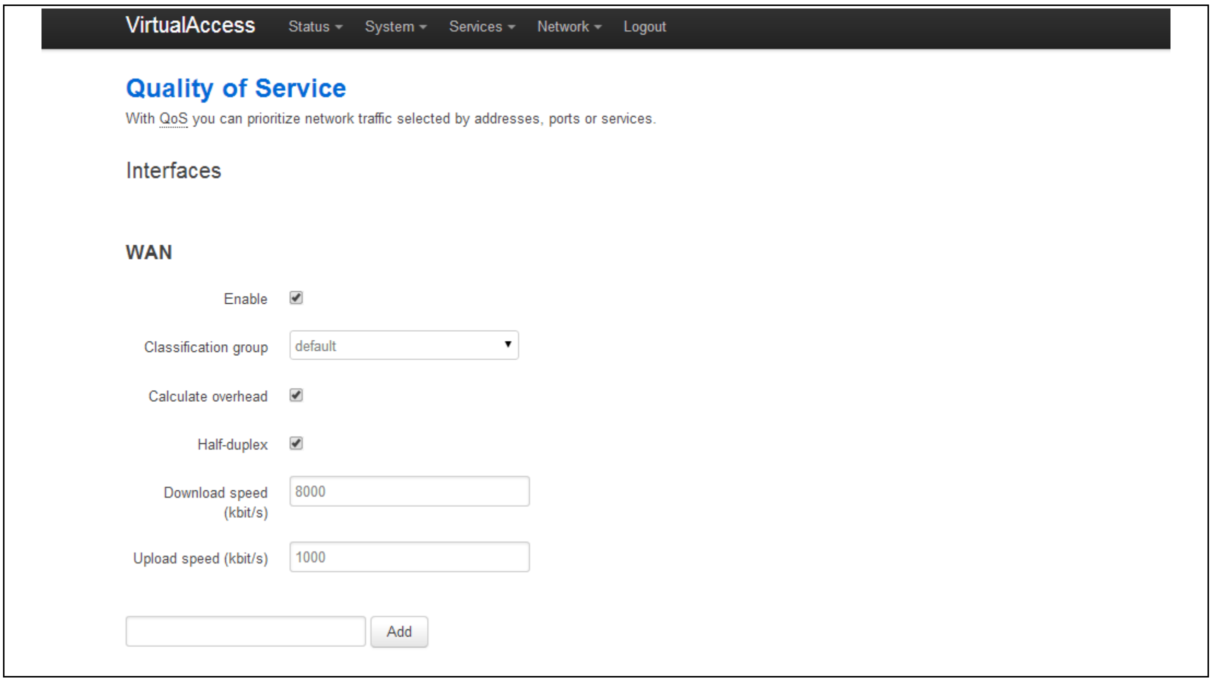


Figure 161: The quality of service page for WAN interface

The following parameters can be configured for theinterface you have chosen. The name of the interfaces should match with the logical name given to the interface in the network configuration.

Web Field / UCI / Package Option	Description
Web: Enabled UCI: qos.[interface].enabled Opt: enabled	Enables or disables QoS interface. 1 — Enabled. 0 — Disabled.
Web: Classification group UCI: qos. [interface].classgroup Opt: classgroup	Creates a mapping before previously created classgroup and interface to which it should be assigned to.
Web: Calculate overhead UCI: qos. [interface].overhead Opt: overhead	Decreases upload and download ratio to prevent link saturation.
Web: Half-duplex UCI: qos [interface].halfduplex Opt: halfduplex	Enables or disables half-duplex operation. 1 — Enabled. 0 — Disabled.
Web: Download speed UCI: qos.[interface].download Opt: download	Download speed limit in kbits/sec.
Web: Upload speed UCI: qos.[interface].upload=2000 Opt:upload	Upload speed limit in kbits/sec.

Table 104: Information table for QoS page

To add classification rules, click **Add**. TheClassification Rules section appears.

Configure each classification rule with the following parameters.

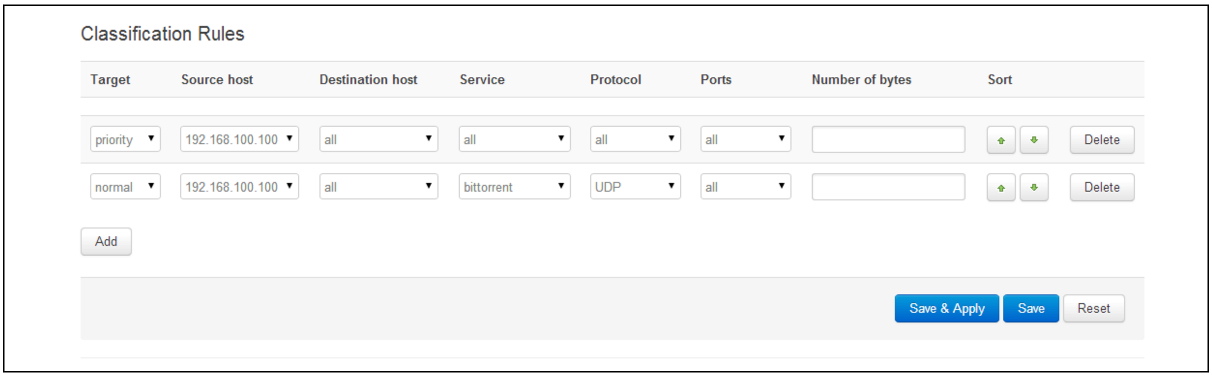


Figure 162: Parameters for classification rules

Web Field/UCI/Package Option	Description
Web: Target UCI: Opt:	Creates and configures selected target bucket. Normal Priority Low Express
Web: Source host UCI: Opt:	Source host.
Web: Destination host UCI: Opt:	Destination host.
Web: Service UCI: Opt:	Selectable service.
Web: Protocol UCI: Opt:	Protocol to classify.
Web: Ports UCI: Opt:	Upload speed kbits/sec.
Web: Number of bytes UCI: Opt:	Number of bytes for bucket.

Table 105: Information table for classification rules

33.4 Configuring QoS using UCI

You can also configure QoS using UCI. The configuration file is stored on:

/etc/config/qos

33.4.1 Interface

Defines the interface on which configured QoS settings will take place.

Each interface can have its own buffer. The interface section declares global characteristics of the connection on which the specified interface is communicating. The following options are defined within this section:

```
config interface 'ADSL'
        option classgroup 'Default'
        option enabled '1'
        option overhead '1'
        option halfduplex '0'
        option download '900'
        option upload '245'
```

Web Field / UCI / Package Option	Description
Web: Enabled UCI: qos.[interface].enabled Opt: enabled	Enables or disables QoS interface. 1: Enabled. 0: Disabled.
Web: Classification group UCI: qos. [interface].classgroup Opt: classgroup	Creates a mapping before previously created classgroup and interface to which it should be assigned to.
Web: Calculate overhead UCI: qos. [interface].overhead Opt: overhead	Decrease upload and download ratio to prevent link saturation.
Web: Half-duplex UCI: qos [interface].halfduplex Opt: halfduplex	Enables or disables half-duplex operation. 1: Enabled. 0: Disabled.
Web: Download speed UCI: qos.[interface].download Opt: download	Download speed limit in kbits/sec.
Web: Upload speed UCI: qos.[interface].upload=2000 Opt:upload	Upload speed limit in kbits/sec.

33.4.2 Classgroup

As there is more than one interface you can have more than one classgroup.

```
config classgroup 'Default'
     option classes 'Express Normal'
     option default 'Normal'
```

UCI / Package Option	Description
UCI: qos.Default=classgroup Opt: Default	Specifies name of classgroup.
UCI: qos.Default.classes=Express Normal Opt: classes	Specifies the list of names of classes which should be part of classgroup.
qos.Default.default=Normal Opt: default	Defines which class is considered default.

33.4.3 Classes

Each bucket has its own configuration.

```
config class 'Normal'
     option packetsize '1500'
     option avgrate '30'
     option priority '5'

config class 'Express'
     option packetsize '1000'
     option maxsize '800'
     option avgrate '50'
     option priority '10'
     option limitrate '10'
```

UCI / Package Option	Description
UCI: qos.Normal=class Opt: Normal	Specifies class name.
UCI: qos.Normal.packetsize=1500 Opt: packetsize	Specifies packet size for the class in bytes.
UCI: qos.Normal.avgrate=30 Opt: avgrate	Average rate for this class, value in % of bandwidth in %.
UCI: qos.Normal.priority=5 Opt: priority	Specifies priority for the class in %.
UCI: qos.Express=class Opt: Express	Specifies class name.
UCI: qos.Express.packetsize=1000 Opt: packetsize	Specifies packet size for the class in bytes.
UCI: qos.Express.maxsize=800 Opt: maxsize	Specify max packet size in bytes.
UCI: qos.Express.avgrate=50 Opt: avgrate	Average rate for this class, value in % of bandwidth in %.
UCI: qos.Express.priority=10 Opt: priority	Specifies priority for the class in %.
UCI: qos.Express.limitrate=10 Opt: limitrate	Defines to how many % of the available bandwidth this class is capped to.

33.4.4 Classify

Classifiers match the traffic for desired class.

```
config classify
       option target 'Express'
       option proto 'udp'
```

UCI / Package Option	Description
UCI: qos.@classify[0]=classify Opt: classify	Part of classify rule.
UCI: qos.@classify[0].target=Express Opt: target	Specifies target class.
UCI: qos.@classify[0].proto=udp Opt: proto	Specifies protocol.

33.5 Example QoS configurations

```
config interface 'ADSL'
      option classgroup 'Default'
      option enabled '1'
      option overhead '1'
      option download '900'
      option upload '245'

config classgroup 'Default'
     option classes 'Express Normal'
     option default 'Normal'

config class 'Normal'
     option packetsize '1500'
     option avgrate '30'
     option priority '5'

config class 'Express'
     option packetsize '1000'
     option maxsize '800'
     option avgrate '50'
     option priority '10'
     option limitrate '10'

config classify
     option target 'Express'
     option proto 'udp'
```

34 Management configuration settings

This chapter contains the configuration sections and parameters required to manage and monitor your device using Activator and Monitor.

34.1 Activator

Activator is a Virtual Access proprietary provisioning system, where specific router configurations and firmware can be stored to allow central management and provisioning. Activator has two distinct roles in provisioning firmware and configuration files to a router.

- Autoload activation of firmware and configuration files on router boot up:
    - Autoload is generally used for router installation. In this scenario the router will initiate the request for firmware and configuration files when it boots up. The router is installed with a factory config that will allow it to contact Activator. The autoload feature controls the behaviour of the router in requesting firmware and configuration files; this includes when to start the Activation process and the specific files requested. The HTTP Client (uhttpd) contains information about the Activator server and the protocol used for activation.
- Deployment of firmware to routers after installation:
    - In this scenario, Activator initiates the process. This process, known as Active Updates, allows for central automatic deployment of firmware and configuration files. It is used when configuration or firmware changes need to be pushed to live routers.

34.2 Monitor

Monitor is a Virtual Access proprietary tool, based on SNMP protocol, to monitor wide networks of deployed routers. The router will be configured to send information to Monitor, which is then stored and viewed centrally via the Monitor application. This includes features such as traffic light availability status, syslog and SLA monitoring.

34.3 Configuration packages used

Package	Sections
autoload	main
httpclient	default
management_users	user

34.4 Autoload: boot up activation

Autoload configurations specify how the device should behave with respect to activation when it boots up. Autoload entries contain information about the specific files to be downloaded and the destination for the downloaded file. Standard autoload entry configurations to download are:

- A firmware file ($$.img)
- A configuration file ($$.ini)
- A .vas file ($$.vas). This file signals the end of the autolaod sequence to Activator

Activator identifies the device using the serial number of the router. $$ syntax is used to denote the serial number of the router when requesting a file. The requested files are written to the alternate image or config segment.

You can change the settings either directly in the configuration file or via appropriate UCI set commands. It is normal procedure for autoload to be enabled in the router's factory settings and disabled in running configurations (config 1 and 2).

Autoload may already have been set at factory config level. If you wish to enable autoload services, proceed through the following steps.

34.5 Autoload packages

Package	Sections
autoload	main

34.5.1 Create a configuration file

In the top menu, select **Services ->Autoload**. The Autoload page has two sections: Basic Settings and Entries. Click **Add** to access configuration settings for each section.

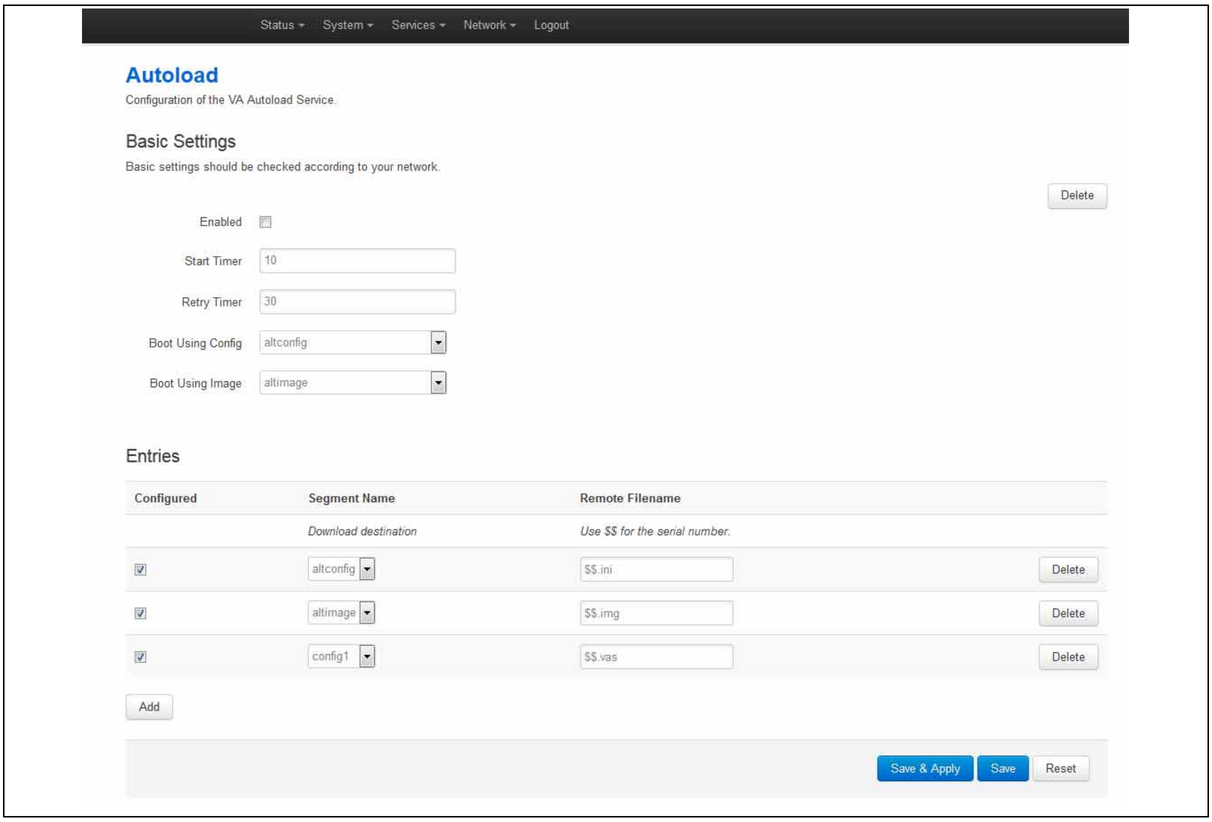


Figure 163: The autoload settings page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td colspan="2">Basic settings</td></tr>
<tr><td>Web: Enabled
UCI: autoload.main.enabled
Opt: Enabled</td><td>Enables activation at system boot.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
<tr><td>Web: Start Timer
UCI: autoload.main.StartTimer
Opt: StartTimer</td><td>Defines how long to wait after the boot up completes before starting activation.<table><tr><td>10</td><td></td></tr><tr><td>Range</td><td>0-300 secs</td></tr></table></td></tr>
<tr><td>Web: Retry Timer
UCI: autoload.main.RetryTimer
Opt: RetryTimer</td><td>Defines how many seconds to wait between retries if a download of a particular autoload entry fails.<table><tr><td>30</td><td></td></tr><tr><td>Range</td><td>0-300 secs</td></tr></table></td></tr>
<tr><td>Web: N/A
UCI: autoload.main.NumberOfRetries
Opt: Numberofretries</td><td>Defines how many retries to attempt before failing the overall activation sequence, backing off and trying the whole activation sequence again.<table><tr><td>5</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: N/A
UCI: autoload.main.BackoffTimer
Opt: Backofftimer</td><td>Defines how many minutes to back off for if a download and all retires fail. After the backoff period, the entire autoload sequence will start again.<table><tr><td>15</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web: Boot Using Config UCI: autoload.main.BootUsingConfig Opt: BootUsingConfig	Specifies which configuration to boot up with after the activation sequence. Altconfig — Alternative configuration Config1 — Configuration 1 Config2 — Configuration 2 Factconf — Factory configuration
Web: Boot Using Image UCI: autoload.main.BootUsingImage Opt: BootUsingImage	Specifies which image to boot up with after the activation sequence completes successfully. Altimage — Alternative image Image 1 — image 1 Image 2 — image 2
Entries	
Web: Configured UCI: autoload.@entry[x].Configured Opt: Configured	Enables the autoload sequence to process this entry. 1 — Enabled. 0 — Disabled.
Web: Segment Name UCI: autoload.@entry[x].SegmentName Opt: SegmentName	Defines where the downloaded file should be stored: (config1 | config2 | altconfig | image1 | image2 | altimage). Typically only altconfig and altimage are used.
Web: RemoteFilename UCI: autoload.@entry[x].RemoteFilename Opt: RemoteFilename	Defines the name of the file to be downloaded from Activator. $$.vas — Notifies activator sequence is complete. $$ ini — Request configuration $$ img — Request firmware Note: $$.vas should always be requested last.

Table 106: Information table for autoload

34.6 Autoload using UCI

```
root@VA_router:/# uci show autoload
autoload.main=core
autoload.main.Enabled=yes
autoload.main.StartTimer=10
autoload.main.RetryTimer=30
autoload.main.NumberOfRetries=5
autoload.main.BackoffTimer=15
autoload.main.BootUsingConfig=altconfig
autoload.main.BootUsingImage=altimage
autoload.@entry[0]=entry
autoload.@entry[0].Configured=yes
autoload.@entry[0].SegmentName=altconfig
autoload.@entry[0].RemoteFilename=$$.ini
autoload.@entry[1]=entry
autoload.@entry[1].Configured=yes
autoload.@entry[1].SegmentName=altimage
autoload.@entry[1].RemoteFilename=$$.img
autoload.@entry[2]=entry
autoload.@entry[2].Configured=yes
autoload.@entry[2].SegmentName=config1
autoload.@entry[2].RemoteFilename=$$.vas
Autoload using package options
root@VA_router:/# uci export autoload
package 'autoload'

config 'core' 'main'
      option 'Enabled' "yes"
      option 'StartTimer' "10"
      option 'RetryTimer' "30"
      option 'NumberOfRetries' "5"
      option 'BackoffTimer' "15"
      option 'BootUsingConfig' "altconfig"
      option 'BootUsingImage' "altimage"

config 'entry'
```

```
      option 'Configured' "yes"
      option 'SegmentName' "altconfig"
      option 'RemoteFilename' "\$\$.ini"

config 'entry'
      option 'Configured' "yes"
      option 'SegmentName' "altimage"
      option 'RemoteFilename' "\$\$.img"

config 'entry'
      option 'Configured' "yes"
      option 'SegmentName' "config1"
      option 'RemoteFilename' "\$\$.vas"
```

34.7 HTTP Client: configuring activation using the web interface

This section contains the settings for the HTTP Client used during activation and active updates of the device.

The httpclient core section configures the basic functionality of the module used for retrieving files from Activator during the activation process.

34.7.1 HTTP Client configuraton packages

Package	Sections
Httpclient	default

34.7.2 Web configuration

To configure HTTP Client for Activator, in the top menu, click **Services -> HTTP Client**. The HTTP Client page has two sections: Basic Settings and Advanced Settings.

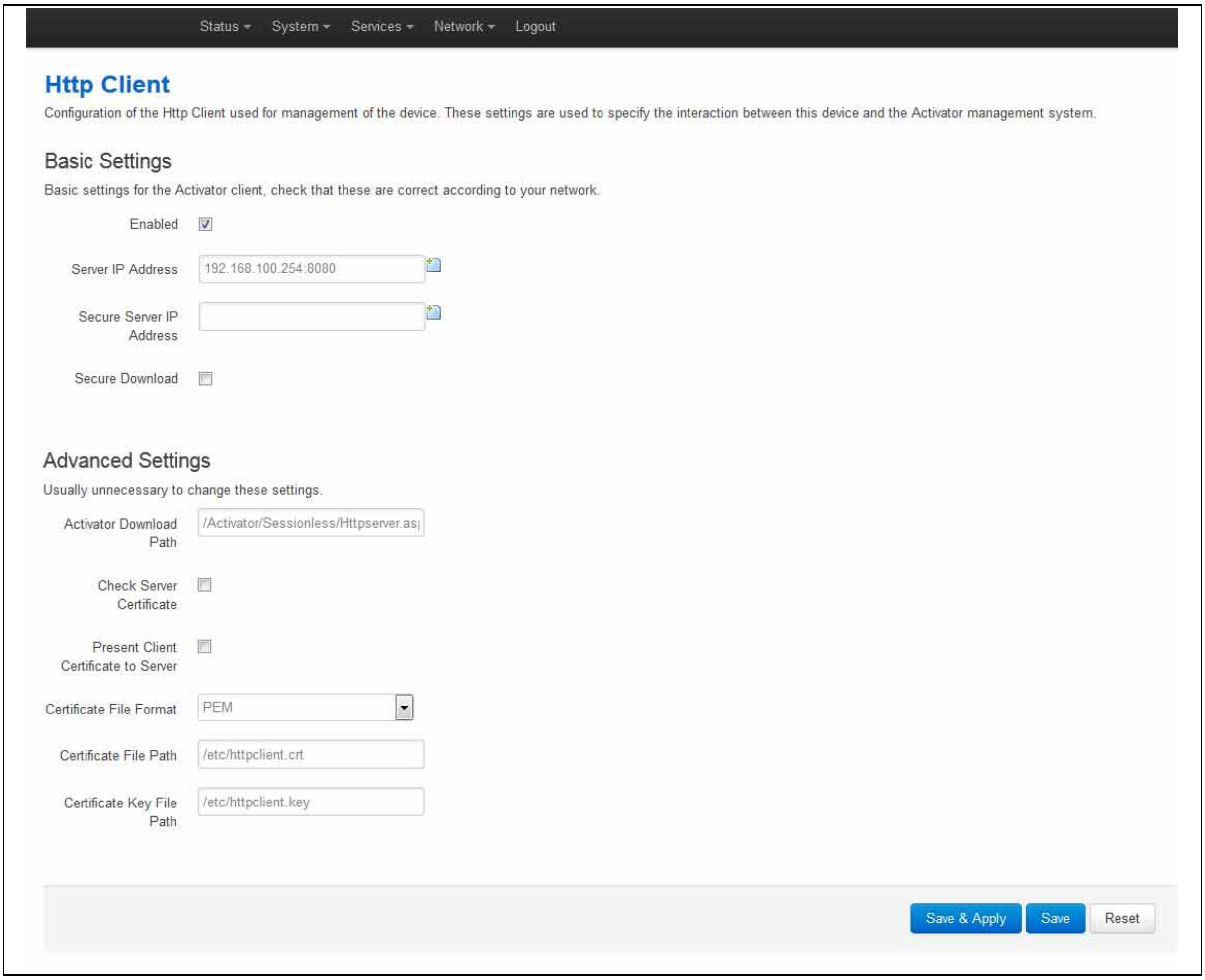


Figure 164: The HTTP client page

Web Field / UCI / Package Option	Description
Basic settings	
Web: Enabled UCI: httpclient.default.enabled Opt: Enabled	Enables the HTTP client. 1 — Enabled. 0 — Disabled.
Web: Server IP Address UCI: httpclient.default.Fileserver Opt: list Fileserver	Specifies the address of Activator that uses http port 80. This can be an IP address or FQDN. The syntax should be x.x.x.x:80 or FQDN:80. Multiple servers should be separated by a space using UCI.
Web: Secure Server IP Address UCI: httpclient.default.SecureFileServer Opt: list SecureFileServer	Specifies the address of Secure Activator that uses port 443. This can be an IP address or FQDN. The syntax should be x.x.x.x:443 or FQDN:443. Multiple servers should be separated by a space using UCI.
Web: Secure Download UCI: httpclient.default.SecureDownload Opt: SecureDownload	Enables Secure Download (port 443). 1 — Enabled. 0 — Disabled.
Advanced settings	
Web: ActivatorDownloadPath UCI: httpclient.default.ActivatorDownloadPath Opt: ActivatorDownloadPath	Specifies the URL on Activator to which the client should send requests. /Activator/Sessionless/Httpserver.asp Range

Web: Check Server Certificate UCI: httpclient.default.ValidateServerCertificateEnabled Opt: ValidateServerCertificateEnabled	Checks for the certificates presence and validity. 1 — Enabled. 0 — Disabled.
Web: Present Client Certificate to Server UCI: httpclient.default. PresentCertificateEnabled Opt: PresentCertificateEnabled	Specifies if the client presents its certificate to the server to identify itself. 1 — Enabled. 0 — Disabled.
Web: CertificateFile Format UCI: httpclient.default.CertificateFormat Opt: CertificateFormat	Specifies the value the client expects to see in the specified field in the server certificate. PEM DER
Web: Certificate File Path UCI: httpclient.default.CertificateFile Opt: CertificateFile	Defines the directory/location of the certificate. /etc/httpclient.crt Range
Web: Certificate Key File Path UCI: httpclient.default.CertificateKey Opt: CertificateKey	Specifies the directory/location of the certificate key. /etc/httpclient.key Range
Web: N/A UCI: ValidateServerCertificateFieldEnabled Opt: ValidateServerCertificate	Defines the field in the server certificate that the client should check. 1 — Enabled. 0 — Disabled.
Web: N/A UCI: httpclient.default.ActivatorChunkyDownloadPath Opt: ActivatorChunkyDownloadPath	Enables partial download activations and active updates. The default value is httpclient.default.ActivatorChunkyDownloadPath=/activator/partial/download The url (on activator) to which the client should send requests for chunky image download.
Web: N/A UCI: httpclient.default.ChunkSize Opt: ChunkSize	Specifies the size of each packet payload 100k — 100K Bytes 1-infinite — Available values
Web: N/A UCI: httpclient.default.RateLimit Opt: RateLimit	Throttle activation/active updates traffic received by device to specified limit None — By default there is no limit 1-infinite — Available values in kbps
Web: N/A UCI: httpclient.default.CAFile Opt: CAFile	Defines path to the certificate authority file stored on the router
Web: N/A UCI: httpclient.default.IgnoreServerCertificateStatus Opt: IgnoreServerCertificateStatus	Defines whether to skip the status check on the server certificate. 1 — Enabled. 0 — Disabled.

Table 107: Information table for HTTP client

34.8 Httpclient: Activator configuration using UCI

```
root@VA_router:~# uci show httpclient
httpclient.default=core
httpclient.default.Enabled=yes
httpclient.default.FileServer=10.1.83.36:80 10.1.83.37:80
httpclient.default.SecureFileServer=10.1.83.36:443 10.1.83.37:443
httpclient.default.ActivatorDownloadPath=/Activator/Sessionless/Httpserver.
asp
httpclient.default.SecureDownload=no
httpclient.default.PresentCertificateEnabled=no
httpclient.default.ValidateServerCertificateEnabled=no
httpclient.default.CertificateFile=/etc/httpclient.crt
httpclient.default.CertificateFormat=PEM
httpclient.default.CertificateKey=/etc/httpclient.key
httpclient.default.ActivatorChunkyDownloadPath=/activator/partial/download
httpclient.default.ChunkSize=100k
httpclient.default.RateLimit=2
httpclient.default.CAFile='/'
httpclient.default.IgnoreServerCertificateStatus=0
```

34.9 Httpclient: Activator configuration using package options

```
root@VA_router:~# uci export httpclient
package httpclient

config core 'default'
      option Enabled 'yes'
      list FileServer '1.1.1.1:80'
      list FileServer '1.1.1.2:80'
      listSecureFileServer '1.1.1.1:443'
      list SecureFileServer '1.1.1.2:443'
      option ActivatorDownloadPath '/Activator/Sessionless/Httpserver.asp'
      option SecureDownload 'no'
      option PresentCertificateEnabled 'no'
      option ValidateServerCertificateEnabled 'no'
      option CertificateFile '/etc/httpclient.crt'
      option CertificateFormat 'PEM'
```

```
    option CertificateKey '/etc/httpclient.key'
    option ActivatorChunkyDownloadPath '/activator/partial/download'
    option ChunkSize '100k'
    option RateLimit '2'
    option CAFile ‘\’
    option IgnoreServerCertificateStatus ‘0’
```

34.10 User management using UCI

User management is not currently available using the web interface. You can configure the feature using UCI or Activator.

34.10.1 User management packages

Package	Sections
management_users	Users

34.10.2 Configuring user management

You can create different users on the system by defining them in the user management configuration file. This gives users access to different services.

Web Field / UCI / Package Option	Description
General settings	
Web: n/a UCI: management_users.@user[x].enabled Opt: enable	Enables/creates the user. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: management_users.@user[x].username Opt: username	Specifies the user’s username.
Web: n/a UCI: management_users.@user[x].password Opt: password	Specifies the user’s password. When entering the user password enter in plain text using the password option. After reboot the password is displayed encrypted via the CLI using the hashpassword option. UCI: management_users.@user[x].hashpassword Opt: hashpassword. Note: a SRP user password will be displayed using the srphash option
Web: n/a UCI: management_users.@user[x].webuser Opt: webuser	Specifies web access permissions for the user. Note: webuser will only work if linuxuser is set to Enabled. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: management_users.@user[x].chapuser Opt: chapuser	Specifies CHAP access permissions for the PPP connection. Note: chapuser will only work if linux user is set to Enabled. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: management_users.@user[x].papuser Opt: papuser	Specifies PAP access permissions for the PPP connection. 0 — Disabled. 1 — Enabled.

Web: n/a UCI: management_users.@user[x].srpuser Opt: srpuser	Specifies SRP access permissions for the PPP connection. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: management_users.@user[x].smsuser Opt: smsuser	Specifies SMS access permissions for the user. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: linuxuser Opt: linuxuser	Specifies linuxuser access permissions for the user. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: List allowed_pages Opt: list allowed_pages	Specifies which pages the user can view. Multiple pages should be entered using a space to separate if using UCI.

Table 108: Information table for config user commands

Note:

- webuser will only work if linuxuser is set to **yes**
- chapuser will only work if linuxuser is set to **no**

When a new user is created on the system and given web access, you will no longer be able to login to the router web interface with the default root user details. The user must use their new user login details.

34.11 Configuring the management user password using UCI

The user password is displayed encrypted via the CLI using the hashpassword option.

```
root@VA_router:~# uci show management_users
management_users.@user[0].username=test
management_users.@user[0].hashpassword=$1$XVzDHHPQ$SKK4geFonctihuffMjS4U0
```

If you are changing the password via the UCI, enter the new password in plain text using the password option.

```
root@VA_router:~# uci set management_users.@user[0].username=newpassword
root@VA_router:~# uci commit
```

The new password will take effect after reboot and will now be displayed in encrypted format through the hashpassword option.

34.12 Configuring management user password using package options

The root password is displayed encrypted via CLI using the hashpassword option.

```
root@VA_router:~# uci export management_users
package management_users

config user
      option hashpassword '$1$wRYYiJOz$EeHN.GQcxXhRgNPVbqxVw
```

If you are changing the password using UCI, enter the new password in plain text using the password option.

```
package management_users

config user
      option hashpassword '$1$wRYYiJOz$EeHN.GQcxXhRgNPVbqxVw
      option password ‘newpassword’
```

The new password will take effect after reboot and will now be displayed in encrypted format via the hashpassword option.

34.13 User management using UCI

```
root@VA_router:~# uci show management_users
management_users.@user[0]=user
management_users.@user[0].enabled=1
management_users.@user[0].username=test
management_users.@user[0].hashpassword=$1$XVzDHHPQ$SKK4geFonctihuffMjS4U0
management_users.@user[0].webuser=1
management_users.@user[0].linuxuser=1
management_users.@user[0].papuser=0
management_users.@user[0].chapuser=0
management_users.@user[0].srpuser=0
management_users.@user[0].smsuser=0
```

34.14 User management using package options

```
root@VA_router:~# uci export management_users

package management_users
config user
```

```
      option enabled '1'
      option username 'test'
      option hashpassword '$1$XVzDHHPQ$SKK4geFonctihuffMjS4U0'
option webuser '1'
      option linuxuser '1'
      option papuser '0'
      option chapuser '0'
      option srpuser '0'
      options smsuser '0'
```

34.15 Configuring user access to specific web pages

To specify particular pages a user can view, add the list allowed_pages. Examples are:

```
listallowed_pages '/admin/status'
```

The user can view admin status page only.

```
listallowed_pages 'admin/system/flashops'
```

The user can view flash operation page only.

To specify monitor widgets only, enter:

```
listallowed_pages 'monitor/<widgetname>'
```

Example widget names are: dhcp, arp, 3gstats, interfaces, memory, multiwan, network, openvpn, routes, system, ipsec, dmvpn, tservd.

35 Configuring Monitor

35.1 Introduction

Virtual Access monitoring system (Monitor) is a secure portal that provides:

- Centralised monitoring of devices
- Device status
- GPS location
- Syslog reporting
- Real time diagnostics
- Email notification
- Advanced statistics
- Dashboard graph reporting

You must configure each router in the network to send the required information to Monitor. This chapter explains how to configure the different information that can be sent to Monitor, including the required router configuration for:

- Reporting device status to Monitor
- Reporting GPS location to Monitor
- Reporting syslog to Monitor
- Configuration of interface statistics collection (ISAD)

For detailed information on operating Monitor, read the Monitor User Manual.

35.2 Reporting device status to Monitor

To allow Monitor to track the IP address and ongoing presence of a device, a keepalive heartbeat SNMP trap is sent from the router. The router is capable of sending SNMP in version 1, 2c and 3.

The SNMP keepalive heartbeat sends basic information on interface status but can also be configured to contain more detailed information such as GPS location.

The basic heartbeat configuration consists of two parts:

- enabling the heartbeat keepalive
- enabling the interface(s) to be monitored

35.2.1 Configuration package used

Package	Sections
monitor	keepalive
network	interface

35.2.2 Configuring keepalive heartbeat using the web interface

Select **Services -> Monitor**. The Monitor Keepalive & ISAD page appears.

The keepalive heartbeat is configured under the **Basic Settings** section.

A single instance keepalive can be configured to multiple monitor address using the same reference, heartbeat interval and other options. Or alternatively multiple keepalive instances can be configured with unique options.

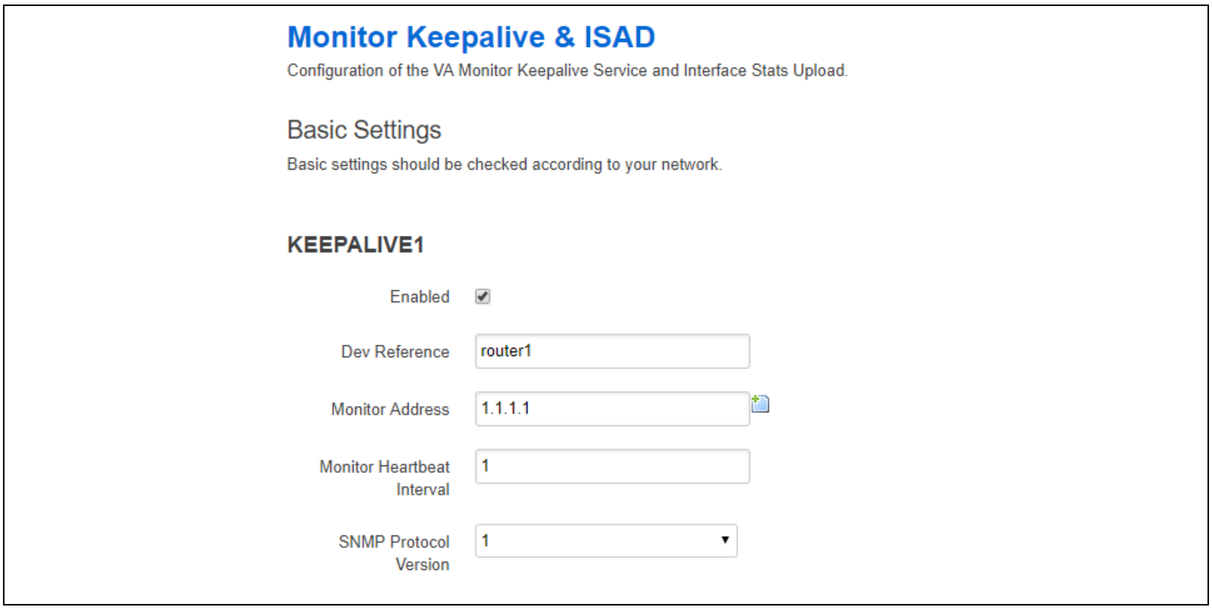


Figure 165: The Monitor & ISAD keepalive page

35.2.2.1 Basic settings

Web Field / UCI / Package Option	Description
Web: Enabled UCI: monitor.@keepalive[0].enabled Opt: Enabled	Enables Monitor to send heartbeats to the router. 0 — Disabled. 1 — Enabled.
Web: Dev Reference UCI: monitor.@keepalive[0].dev_reference Opt: dev_reference	Sets a unique identification for this device known to Monitor.
Web: Monitor Address UCI: monitor.@keepalive[0].monitor_ip Opt: list monitor_ip	Defines the IP address of Monitor. It is possible to specify multiple addresses to which SNMP heartbeat traps will be sent. To configure via UCI use a space separator. Example: `monitor.@keepalive[0].monitor_ip=1.1.1.1 2.2.2.2`
Web: Monitor Heartbeat Interval UCI: monitor.@keepalive[0].interval_min Opt: interval_min	Specifies the interval, in minutes, at which traps are sent. 1 — Trap set every 1 minute. Range —
Web: SNMP Protcol Version UCI: monitor.@keepalive[0].snmp_version Opt: snmp_version	Specifies what SNMP version is sent to remote Manager. 1 — snmp version 1 2c — SNMP version 2c 3 — SNMP version 3

Table 109: Information table for Monitor & ISAD basic configuration

The figure below shows options that are relevant only if you have selected SNMP version 3.

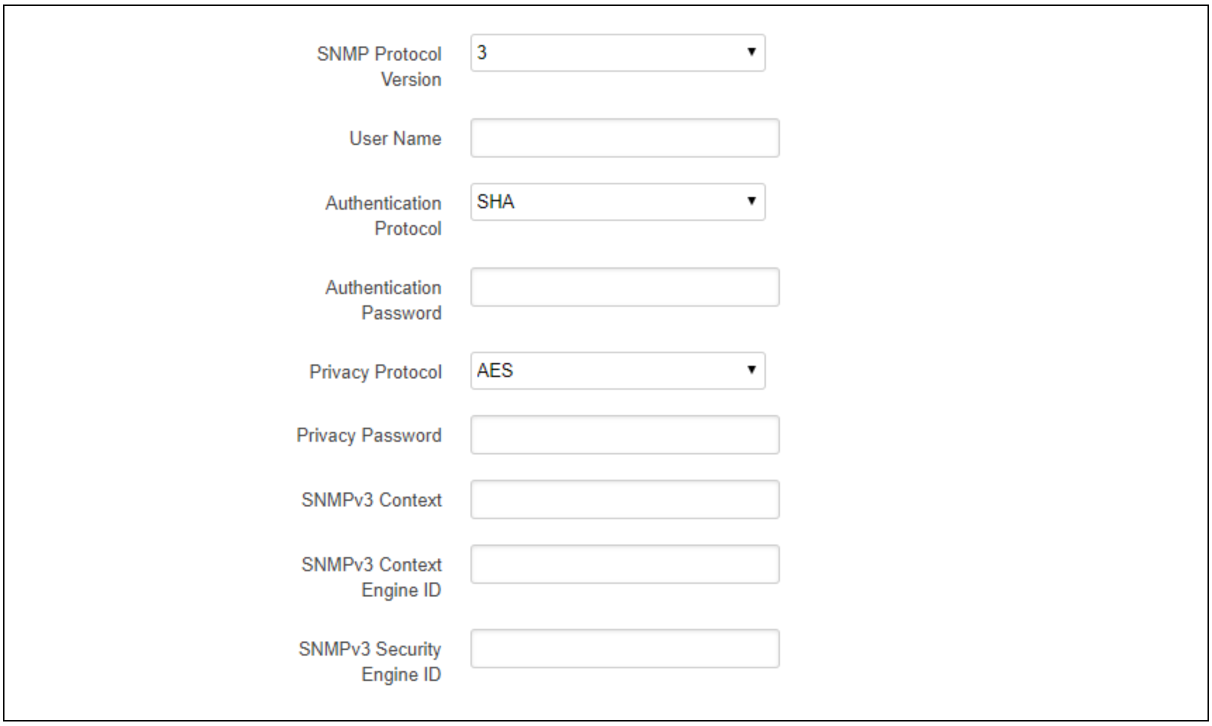


Figure 166: The Monitor & ISAD keepalive page for SNMP v3

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: User Name
UCI: monitor.@keepalive[0].snmp_uname
Opt: snmp_uname</td><td>Specifies user name.<table><tr><td>Blank</td><td>Default value</td></tr><tr><td>String</td><td></td></tr></table></td></tr>
<tr><td>Web: Authentication Password
UCI: monitor.@keepalive[0].snmp_auth_pass
Opt: snmp_auth_pass</td><td>Specifies snmpv3 authentication password.</td></tr>
<tr><td>Web: Authentication Protocol
UCI: monitor.@keepalive[0].snmp_auth_proto
Opt: snmp_auth_proto</td><td>Specifies snmpv3 authentication protocol.<table><tr><td>Blank</td><td>Default value</td></tr><tr><td>MD5</td><td>MD5 as authentication protocol</td></tr><tr><td>SHA</td><td>SHA as authentication protocol</td></tr></table></td></tr>
<tr><td>Web: Privacy Protocol
UCI: monitor.@keepalive[0].snmp_priv_proto
Opt: snmp_priv_proto</td><td>Specifies snmpv3 privacy protocol<table><tr><td>Blank</td><td>Default value</td></tr><tr><td>AES</td><td>AES as privacy protocol</td></tr><tr><td>DES</td><td>MD5 as privacy protocol</td></tr></table></td></tr>
<tr><td>Web: Privacy Password
UCI: monitor.@keepalive[0].snmp_priv_pass
Opt: snmp_priv_pass</td><td>Specifies snmpv3 privacy password.</td></tr>
<tr><td>Web: SNMPv3 Context
UCI: monitor.@keepalive[0].snmp_context
Opt: snmp_context</td><td>Specifies snmpv3 context name.</td></tr>
<tr><td>Web: SNMPv3 Context Engine ID
UCI: monitor.@keepalive[0].snmp_context_eid
Opt: snmp_context_eid</td><td>Specifies snmpv3 context engine ID.</td></tr>
</table>

Web: SNMPv3 Security Engine ID UCI: monitor.@keepalive[0].snmp_sec_eid Opt: snmp_sec_eid	Specifies snmpv3 security engine ID.

Table 110: Information table for SNMP v3 reporting device commands

35.2.3 Configuring keepalive heartbeat using command line

Keepalive is configured under the monitor package.

By default, all keepalive instances are named ‘keepalive’, it is identified by `@keepalive` then the keepalive position in the package as a number. For example, for the first keepalive in the package using UCI:

```
monitor.@keepalive[0]=keepalive
monitor.@ keepalive[0].enabled=1
```

Or using package options:

```
config keepalive
        option enabled ‘1’
```

However, to better identify, it is recommended to give the keepalive instance a name. For example, to create a keepalive instance named keepalivev1.

To define a named keepalive instance using UCI, enter:

```
monitor.keepalivev1=keepalive
monitor.keepalivev1.enable=1
```

To define a named keepalive instance using package options, enter:

```
config keepalive 'keepalivev1'
        option enabled '1'
```

35.2.4 Keepalive using UCI

```
root@VA_router:~# uci show monitor
monitor.keepalivev1=keepalive
monitor.keepalivev1enabled=1
monitor.keepalivev1.interval_min=1
monitor.keepalivev1.dev_reference=router1
monitor.keepalivev1.monitor_ip=10.1.83.36
monitor.keepalivev1.snmp_version=1
monitor.keepalivev2=keepalive
```

```
monitor.keepalivev2.enable=1
monitor.keepalivev2.interval_min=1
monitor.keepalivev2.monitor_ip=172.16.250.100
monitor.keepalivev2.dev_reference=TEST
monitor.keepalivev2.snmp_version=2c
monitor.keepalivev3=keepalive
monitor.keepalivev3.enable=1
monitor.keepalivev3.interval_min=1
monitor.keepalivev3.monitor_ip=172.16.250.101
monitor.keepalivev3.dev_reference=TEST
monitor.keepalivev3.snmp_version=3
monitor.keepalivev3.snmp_uname=TEST
monitor.keepalivev3.snmp_auth_pass=vasecret
monitor.keepalivev3.snmp_auth_proto=MD5
monitor.keepalivev3.snmp_priv_pass=vasecret
monitor.keepalivev3.snmp_priv_proto=DES
```

35.2.5 Keepalive using package options

```
root@VA_router:~# uci export monitor
package 'monitor'

config keepalive 'keepalivev1'
        option enabled '1'
        option interval_min '1'
        option dev_reference 'router1'
        option enabled 'yes'
        list monitor_ip '10.1.83.36'

config keepalive 'keepalivev2'
        option enable '1'
        option interval_min '1'
        list monitor_ip '172.16.250.100'
        option dev_reference 'TEST'
        option snmp_version '2c'

config keepalive 'keepalivev3'
```

```
        option enable '1'
        option interval_min '1'
        list monitor_ip '172.16.250.101'
        option dev_reference 'TEST'
        option snmp_version '3'
        option snmp_uname 'TEST'
        option snmp_auth_pass 'vasecret'
        option snmp_auth_proto 'MD5'
        option snmp_priv_pass 'vasecret'
        option snmp_priv_proto 'DES'
```

35.2.6 Enabling interface status in keepalive heartbeat via web interface

The keepalive heartbeat can send information on multiple interfaces. In order to send an interface status to Monitor, select **Network** -> **Interfaces**, then under the required interface select **Edit**. Under **Advanced Settings** enable the Monitor interface state option.

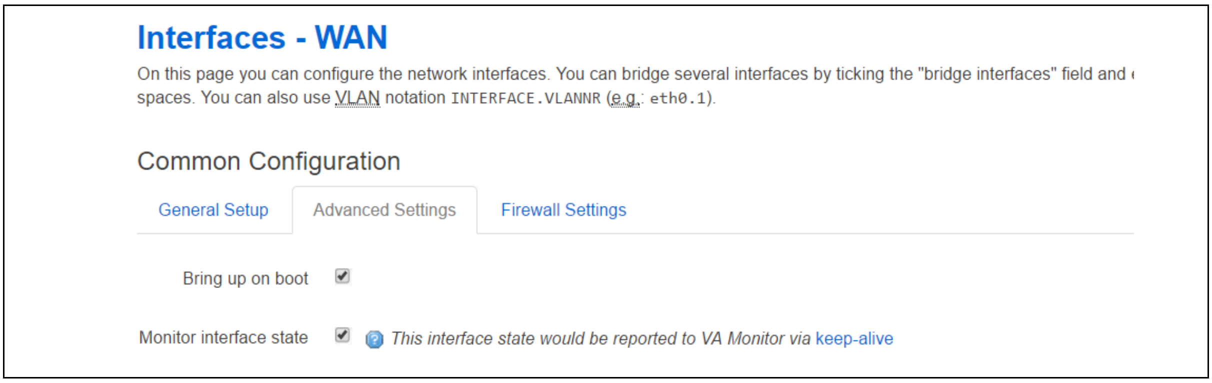


Figure 167: The interface common configuration page

Web Field / UCI / Package Option	Description
Web: Monitor interface state UCI: network.@interface[0].monitored Opt: monitored	Enables interface status to be sent in the heartbeat trap to Monitor. 0 — Disabled. 1 — Enabled.

Table 111: Information table for enabling interface status command

35.2.7 Enabling interface status using command line

Interface status is configured under the network package.

35.2.7.1 Enable interface status using UCI

```
root@VA_router:~# uci show network
network.@interface[0]=interface
……
```

```
network.@interface[0].monitored=1
……
```

35.2.7.2 Enable interface status using package option

```
root@VA_router:~# uci export network
package network
config interface ‘WAN’
      ……
      option monitored ‘1’
      ……
```

35.3 Reporting GPS location to Monitor

To allow Monitor to display a router GPS location, the GPS coordinates can be configured to be sent in the heartbeat keepalive from the router.

GPS location is only available in supported hardware models.

Ensure monitor keepalive heartbeat is correctly configured as in section 35.2 above.

35.3.1 Configuration package used

Package	Sections
gpsd	gpsd

35.3.2 Configuring GPS location via the web interface

Select **Services -> GPS**. The GPS configuration page appears.

The web interface configures a gpsd section named core.

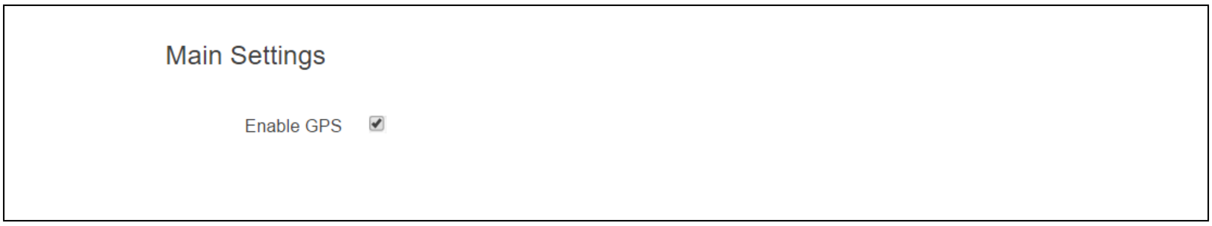


Figure 168: The GPS configuration page

Web Field / UCI / Package Option	Description
Web: Enable GPS UCI: monitor.core.enabled Opt: enabled	Enables GPS coordinates to be sent in the heartbeat keepalive to Monitor. 0: Disabled. 1: Enabled.

Table 112: Information table for reporting GPS commands

35.3.3 Configuring GPS using command line

GPS location is configured under the gpsd package.

35.3.3.1 GPS using UCI

```
root@VA_router:~# uci show gpsd
gpsd.core=gpsd
gpsd.core.enabled=1
```

35.3.3.2 GPS using package options

```
root@VA_router:~# uci export gpsd
package gpsd
config gpsd 'core'
        option enabled '1'
```

35.3.4 GPS diagnostics

To view information on GPS coordinates via the web interface, select **Status -> GPS Information**.

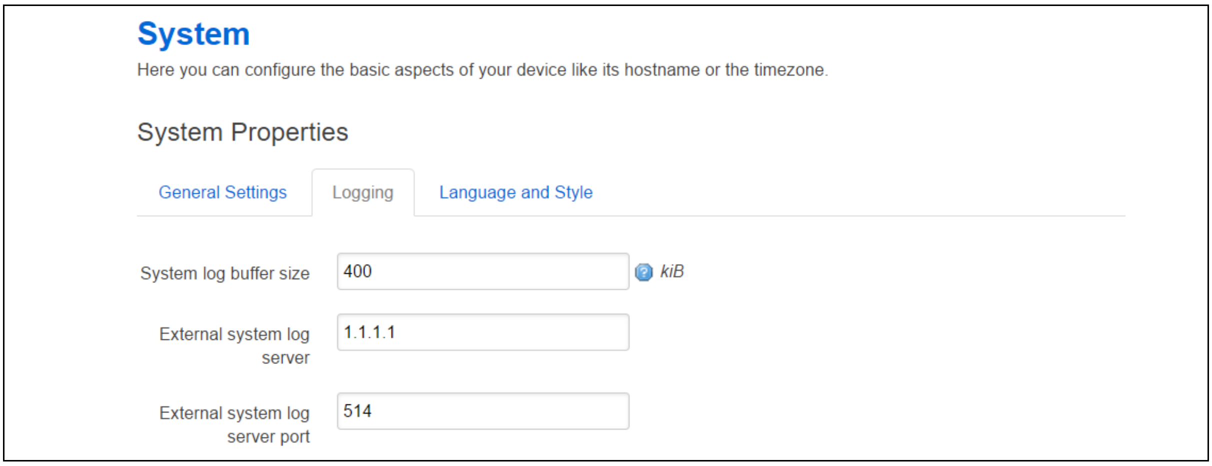


Figure 169: The GPS status page

To view GPS coordinates via command line, enter gpspeek:

```
root@VA_router:~# gpspeek
Fix: 3D,1495467700,53.342529,-
6.241236,27.700000,202.600000,0.000000,0.000000
```

35.4 Reporting syslog to Monitor

35.4.1 Configuration package used

Package	Sections
system	main

35.4.2 Configuring syslog to Monitor via the web interface

Monitor can display syslog events sent from the router. To configure the router to send syslog events, select **System -> System** -> **Logging** and set **External system log server** to the Monitor IP. You can also configure the syslog server port if required.

All syslog events are sent to the syslog server.

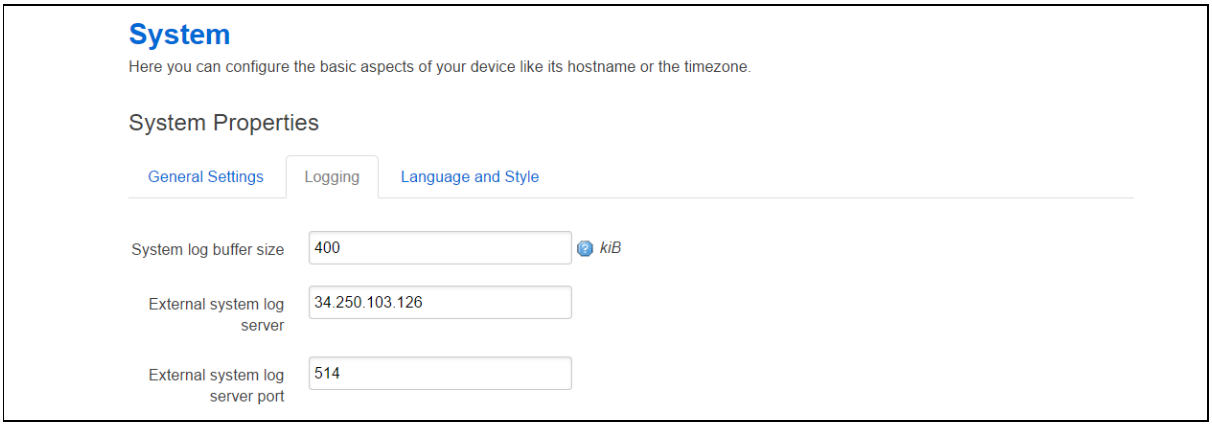


Figure 170: The system properties page

Web Field / UCI / Package Option	Description
Web: External system log server UCI: system.main.log_ip Opt: log_ip	Defines the external syslog server IP address.
Web: External system log server UCI: system.main.log_port Opt: log_port	Defines the external syslog server destination port number for syslog messages. 514 Range

Table 113: Information table for syslog properties commands

35.4.3 Configuring syslog events to Monitor using command line

Syslog is configured under the system package.

35.4.3.1 Syslog events to Monitor using UCI

```
root@VA_router:~# uci show system
system.main=system
......
system.main.log_ip=1.1.1.1
system.main.log_port=514
......
```

35.4.3.2 Syslog events to Monitor using package options

```
root@VA_router:~# uci export system
package system
```

```
config system ‘main’
……
      option log_ip ‘1.1.1.1’
      option log_port ‘514’
……
```

35.5 Configuring ISAD

ISAD is a system for collecting interface stats to be displayed on Monitor.

The following section explains how to configure interface statistics collection (iSAD). Statistical data is collected in bins with each bin containing interface transmit and receive packets/bytes/errors for a period. Signal strength and also temperature parameters are also stored in the bins. Bins are uploaded to Monitor periodically.

Note: Ensure monitor keepalive heartbeat and interface status is correctly configured as in section 30.2 above. Interfaces should have `option monitored` enabled as part of the collection.

ISAD replaces the deprecated SLA feature.

35.5.1 Configuration package used

Package	Sections
monitor	interface_stats

35.5.2 Configuring ISAD using the web interface

Select **Services -> Monitor**. The Monitor Keepalive & ISAD page appears. ISAD is configured under the **Interface Stats** section.

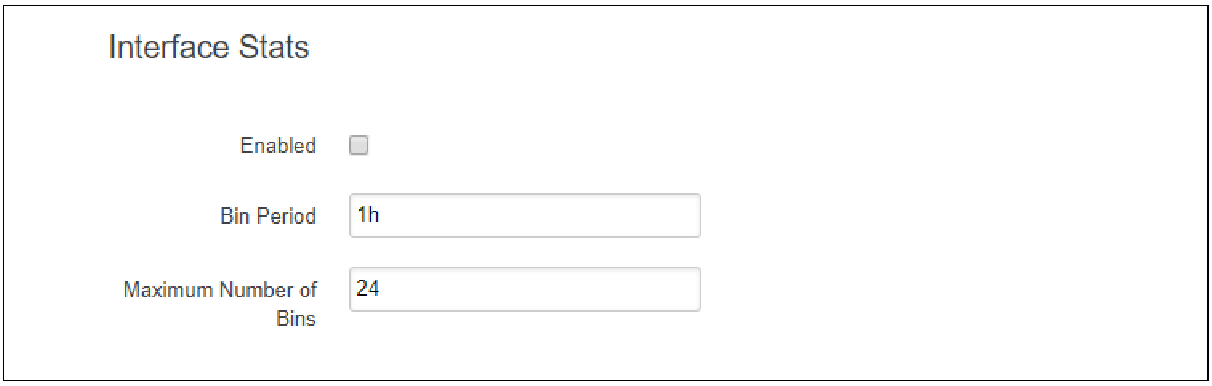


Figure 171: The Monitor Keepalive & ISAD Interface Stats page

Web Field / UCI / Package Option	Description
Web: Enabled UCI: monitor.stats.enabled=1 Opt: enabled	Enables ISAD. 0 Disabled. 1 Enabled.

Web: Bin Period UCI: monitor.stats.bin_period Opt: time	Specifies how long to collect data for one bin. Specifies the interval, in minutes, at which traps are sent. 1h: Bin collected for 1 hour Range:
Web: Maximum Number of Bins UCI: monitor.stats.bin_cache_size Opt: bin_cache_size	Specifies the maximum number of bins to store. Empty: 24 Range:

Table 114: Information table for ISAD Monitor Keepalive & ISAD Interface Stats section

35.5.3 Configuring ISAD using the command line

ISAD is configured under the Monitor package.

35.5.3.1 ISAD using UCI

```
root@VA_router:~# uci show monitor
monitor.keepalivev1=keepalive
monitor.keepalivev1enabled=1
monitor.keepalivev1.interval_min=1
monitor.keepalivev1.dev_reference=router1
monitor.keepalivev1.monitor_ip=10.1.83.36
monitor.keepalivev1.snmp_version=1
monitor.stats=interface_stats
monitor.stats.enabled=1
monitor.stats.bin_period=1h
monitor.stats.bin_cache_size=24
```

35.5.3.2 ISAD using package options

```
root@VA_router:~# uci export monitor
package monitor

config keepalive 'keepalivev1'
        option interval_min '1'
        option enabled '1'
        list monitor_ip '10.1.83.36
        option dev_reference 'router1'

config interface_stats 'stats'
        option enabled '1'
        option bin_period '1h'
        option bin_cache_size '24'
```

35.5.4 ISAD diagnostics

35.5.4.1 Checking process

To check to see if ISAD is running, enter pgrep –fl isad:

```
root@VA_router:~# pgrep -fl isad
5303 /usr/sbin/isad -b 60 -s 10 -c 200 -u /var/state /var/const_state
```

35.5.4.2 Checking bin statistics

To check if stats are being collected, enter cat /var/state/monitor:

```
root@VA_router:~# cat /var/state/monitor
monitor.bin_0=isad
monitor.bin_0.end_ts=85020
monitor.bin_0.start_ts=84960
monitor.bin_1=isad
monitor.bin_1.end_ts=85080
monitor.bin_1.start_ts=85020
monitor.bin_2=isad
monitor.bin_2.end_ts=85140
monitor.bin_2.start_ts=85080
```

35.5.5 ISAD operation

The bin statistics stored on the router must be periodically pushed statistics to Monitor.

This is normally done centrally when statistics are enabled on Monitor. Monitor contacts each router and auto-generates a script that will automatically schedule the upload of the bin statistics.

However, if Monitor cannot access the router WAN IP, you must do this manually on each router using a UDS script. An example is shown below where the bins are uploaded every hour to a Monitor server IP 89.101.154.154 using TFTP.

```
package uds


config script 'isb_upload_scr'
        option enabled '1'
        option exec_type 'periodic'
        option period '1h'
        list text '/usr/sbin/isb_upload.lua 89.101.154.154:69'
```

36 Configuring SNMP

SNMP (Simple Network Management Protocol) is an internet-standard protocol for managing devices on IP networks. SNMP exposes management data in the form of a hierarchy of variables in a MIB (Management Information Base). These variables can be queried individually, or in groups using their OIDs (Object Identifiers) defined in MIBs. In addition, information from the router can be pushed to a network management station in the form of SNMP traps.

36.1 Configuration package used

Package	Sections				
snmpd	access agent com2sec constant	exec group heartbeat informreceiver	inventory inventory_iftable monitor_disk monitor_ioerror	monitor_load monitor_memory monitor_process pass	system trapreceiver usm_user view

The SNMP application has several configuration sections:

System and Agent	Configures the SNMP agent.
Com2Sec	Maps SNMP community names into an arbitrary security name.
Group	Assigns community names and SNMP protocols to groups.
View and Access	Creates views and sub views of the whole available SNMP tree and grants specific access to those views on a group by group basis.
usm_user	Define a user for SNMPv3 USM
Trap receiver	Address of a notification receiver that should be sent SNMPv1 TRAPs and SNMPv2c TRAP2s.
Inform receiver	Address of a notification receiver that should be sent SNMPv2 INFORM notifications respectively

36.2 Configuring SMNP using the web interface

In the top menu, select **Services -> SNMP**. The SNMP Service page appears.

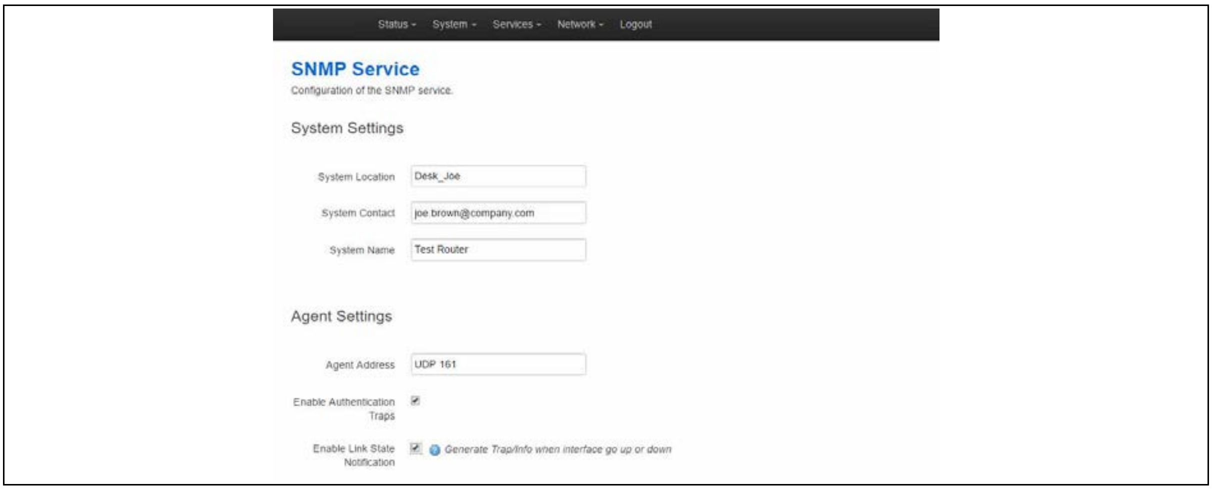


Figure 172: The SNMP service page

36.2.1 System and agent settings

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td colspan="2">System settings</td></tr>
<tr><td>Web: System Location
UCI: snmpd.system[0].sysLocation
Opt: sysLocation</td><td rowspan="3">Sets the system location, system contact or system name for the agent. This information is reported in the ‘system’ group in the mibII tree.</td></tr>
<tr><td>Web: System Contact
UCI: snmpd.system[0].sysContact
Opt: sysContact</td></tr>
<tr><td>Web: System Name
UCI: snmpd.system[0].sysName
Opt: sysName</td></tr>
<tr><td colspan="2">Agent Settings</td></tr>
<tr><td>Web: Agent Address
UCI: snmpd.agent[0].agentaddress
Opt: agentaddress</td><td>Specifies the address(es) and port(s) on which the agent should listen.
[(udp|tcp):]port[@address][,…]</td></tr>
<tr><td>Web: Enable Authentication Traps
UCI: snmpd.agent[0].authtrapenabled
Opt: authtrapenabled</td><td>Enables or disables SNMP authentication trap.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table>Note: this is the SNMP poll authentication trap to be set when there is a community mismatch.</td></tr>
<tr><td>Web: Enable Link State Notification
UCI: snmpd.agent[0].link_updown_notify
Opt: link_updown_notify</td><td>Generates trap/info when interface goes up or down. When enabled, the router sends a trap notification link up or down.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
</table>

Table 115: Information table for system and agent settings

36.2.2 Com2Sec settings

To access Com2Sec settings, scroll down the SNMP Services page.

Use the COM2Sec section to map SNMP community names into an arbitrary security name. Map community names into security names based on the community name and the source subnet. Use the first source/community combination that matches the incoming packet.

A community string is a password that is applied to a device to restrict both read-only and read-write access to the SNMP data on the device. These community strings should be chosen carefully to ensure they are not trivial. They should also be changed at regular intervals and in accordance with network security policies.

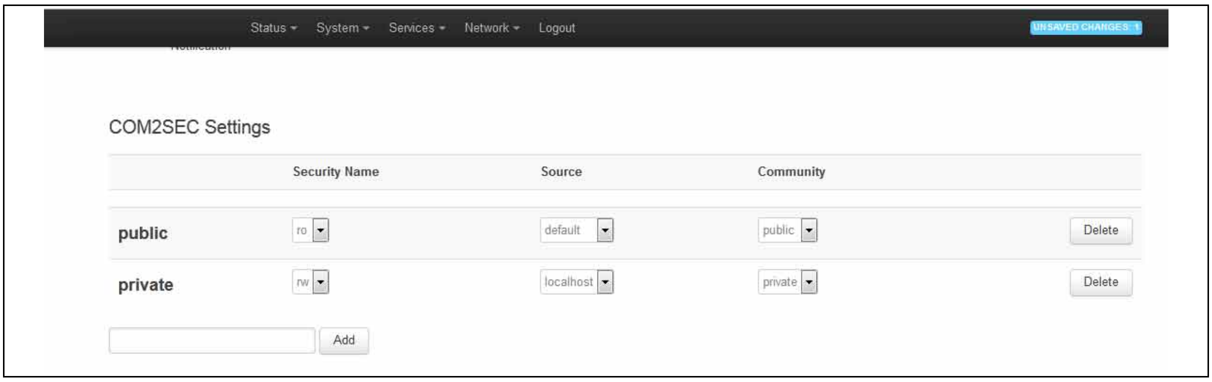


Figure 173: The COM2Sec settings section

Web Field / UCI / Package Option	Description
Web: Security Name UCI: snmpd.com2sec[x].secname Opt: secname	Specifies an arbitrary security name for the user.
Web: Source UCI: snmpd.com2sec[x].source Opt: source	A hostname, localhost or a subnet specified as a.b.c.d/mask or a.b.c.d/bits or 'default' for no restrictions.
Web: Community UCI: snmpd.com2sec[x].community Opt: community	Specifies the community string being presented in the request.

Table 116: Information table for Com2Sec settings

36.2.3 Group settings

Group settings assign community names and SNMP protocols to groups.

Figure 174: The group settings section

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Group
UCI: snmpd.group[x].group
Opt: group</td><td>Specifies an arbitrary group name.</td></tr>
<tr><td>Web: Version
UCI: snmpd.group[x].version
Opt: version</td><td>Specifies the SNMP version number being used in the request: v1, v2c and usm (User-based Security Module) are supported.
<table>
<tr><td>v1</td><td>SNMP v1</td></tr>
<tr><td>v2v</td><td>SNMP v2</td></tr>
<tr><td>usm</td><td>SNMP v3</td></tr>
<tr><td>any</td><td>Any SNMP version</td></tr>
</table></td></tr>
<tr><td>Web: Security Name
UCI: snmpd.group[x].secname
Opt: secname</td><td>An already defined security name that is being included in this group.</td></tr>
</table>

Table 117: Information table for group settings

36.2.4 View settings

View settings define a named "view", which is a subset of the overall OID tree. This is most commonly a single subtree, but several view directives can be given with the same view name, to build up a more complex collection of OIDs.

Figure 175: The view settings section

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Name
UCI: snmpd.view[x].viewname
Opt: viewname</td><td>Specifies an arbitrary view name. Typically it describes what the view shows.</td></tr>
<tr><td>Web: Type
UCI: snmpd.view[x].type
Opt: type</td><td>Specifies whether the view lists oids that are included in the view or lists oids to be excluded from the view (in which case all other oids are visible apart from those ones listed).
<table>
<tr><td>included</td><td></td></tr>
<tr><td>excluded</td><td></td></tr>
</table></td></tr>
<tr><td>Web: OID
UCI: snmpd.view[x].oid
Opt: oid</td><td>OID to be included in or excluded from the view. Only numerical representation is supported.
Example
<table>
<tr><td>1</td><td>Everything</td></tr>
<tr><td>1.3.6.1.2.1.2</td><td>Interfaces table</td></tr>
</table></td></tr>
</table>

Table 118: Information table for view settings

36.2.5 Access settings

Access settings map from a group of users/communities, in a specific context and with a particular SNMP version and minimum security level, to one of three views, depending on the request being processed.

Figure 176: The access settings section

Web Field / UCI / Package Option	Description
Web: Group UCI: snmpd.access[x].group Opt: group	Specifies the group to which access is being granted.
Web: Context UCI: snmpd.access[x].context Opt: context	SNMPv3 request context is matched against the value according to the prefix below. For SNMP v1 and SNMP v2c, the context must be **none**. none all
Web: Version UCI: snmpd.access[x].version Opt: version	Specifies the SNMP version number being used in the request: any, v1, v2c and usm are supported. v1 — SNMP v1 v2v — SNMP v2 usm — SNMP v3 any — Any SNMP version
Web: Level UCI: snmpd.access[x].level Opt: level	Specifies the security level. For SNMP v1 and SNMP v2c level must be **noauth**. noauth auth priv
Web: Prefix UCI: snmpd.access[x].prefix Opt: prefix	Prefix specifies how context (above) should be matched against the context of the incoming pdu. exact any all
Web: Read UCI: snmpd.access[x].read Opt: read	Specifies the view to be used for read access.
Web: Write UCI: snmpd.access[x].write Opt: write	Specifies the view to be used for write access.
Web: Notify UCI: snmpd.access[x].notify Opt: notify	Specifies the view to be used for notify access.

Table 119: Information table for access settings

36.2.6 Trap receiver

Trap receiver settings define a notification receiver that should be sent SNMPv1 TRAPs and SNMPv2c TRAP2.

Figure 177: The trap receiver settings page

Web Field / UCI / Package Option	Description
Web: Host UCI: snmpd.trapreceiver[x].host Opt: host	Host address. Can be either an IP address or an FQDN.
Web: Port UCI: snmpd.trapreceiver[x].port Opt: port	UDP port to be used for sending traps. Range 162
Web: Version UCI: snmpd.trapreceiver[x].version Opt: version	SNMP version. v1 V2
Web: Community UCI: snmpd.trapreceiver[x].community Opt: community	Community to use in trap messages for this host.

Table 120: Information table for trap receiver settings

36.2.7 Inform receiver

Inform receiver settings define a notification receiver that should be sent SNMPv2c INFORM notifications.

Figure 178: The inform receiver settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Host
UCI: snmpd.informreceiver[x].host
Opt: host</td><td>Host address. Can be either an IP address or an FQDN.</td></tr>
<tr><td>Web: Port
UCI: snmpd.informreceiver[x].port
Opt: port</td><td>UDP port to be used for sending traps.<table><tr><td>Range</td><td></td></tr><tr><td>162</td><td></td></tr></table></td></tr>
<tr><td>Web: Community
UCI: snmpd.informreceiver[x].community
Opt: community</td><td>Community to use in inform messages for this host.</td></tr>
</table>

Table 121: Information table for trap receiver settings

36.3 Configuring SNMP using command line

The configuration files are stored on **/etc/config/snmpd**.

36.3.1 System settings using UCI

```
root@VA_router:~# uci show snmpd
snmpd.system=system
snmpd.system.sysLocation=Office 123
snmpd.system.sysContact=Mr White
snmpd.system.sysName=Backup Access 4
snmpd.agent=agent
snmpd.agent.agentaddress=UDP:161
snmpd.agent.authtrapenabled=yes
snmpd.agent.link_updown_notify=yes
```

36.3.2 System settings using package options

```
root@VA_router:~# uci export snmpd
package snmpd
config 'system'
      option sysLocation 'Office 123'
      option sysContact 'Mr White'
      option sysName 'Backup Access 4'

config 'agent'
      option agentaddress 'UDP:161'
      option authtrapenabled '1'
       option link_updown_notify '1'
```

Another sample agent configuration shown below causes the agent to listen on UDP port 161, TCP port 161 and UDP port 9161 on only the interface associated with the localhost address.

```
config 'agent'
      option agentaddress 'UDP:161,tcp:161,9161@localhost'
```

36.3.3 com2sec settings

The following sample specifies that a request from any source using “public” as the community string will be dealt with using the security name “ro”. However, any request from the localhost itself using “private” as the community string will be dealt with using the security name “rw”.

Note: the security names of “ro” and “rw” here are simply names – the fact of a security name having read only or read-write permissions is handled in the access section and dealt with at a group granularity.

36.3.3.1 Com2sec using UCI

```
snmpd.c2s_1=com2sec
snmpd.c2s_1.source=default
snmpd.c2s_1.community=public
snmpd.c2s_1.secname=rw
snmpd.c2s_2=com2sec
snmpd.c2s_2.source=localhost
snmpd.c2s_2.community=private
snmpd.c2s_2.secname=ro
```

36.3.3.2 Com2sec using package options

```
config 'com2sec' 'public'
      option secname 'ro'
      option source 'default'
      option community 'public'

config 'com2sec' 'private'
      option secname 'rw'
      option source 'localhost'
      option community 'private'
```

36.3.4 Group settings

The following example specifies that a request from the security name “ro” using snmp v1, v2c or USM (User Based Security Model for SNM P v3) are all mapped to the “public” group. Similarly, requests from the security name “rw” in all protocols are mapped to the “private” group.

36.3.4.1 Group settings using UCI

```
snmpd.grp_1_v1=group
snmpd.grp_1_v1.version=v1
snmpd.grp_1_v1.group=public
snmpd.grp_1_v1.secname=ro
snmpd.grp_1_v2c=group
snmpd.grp_1_v2c.version=v2c
snmpd.grp_1_v2c.group=public
snmpd.grp_1_v2c.secname=ro
snmpd.grp_1_usm=group
snmpd.grp_1_usm.version=usm
snmpd.grp_1_usm.group=public
snmpd.grp_1_usm.secname=ro
snmpd.grp_1_access=access
snmpd.grp_1_access.context=none
snmpd.grp_1_access.version=any
snmpd.grp_1_access.level=noauth
snmpd.grp_1_access.prefix=exact
snmpd.grp_1_access.read=all
snmpd.grp_1_access.write=none
snmpd.grp_1_access.notify=none
snmpd.grp_1_access.group=public
snmpd.grp_2_v1=group
snmpd.grp_2_v1.version=v1
snmpd.grp_2_v1.group=public
snmpd.grp_2_v1.secname=ro
snmpd.grp_2_v2c=group
snmpd.grp_2_v2c.version=v2c
snmpd.grp_2_v2c.group=public
snmpd.grp_2_v2c.secname=ro
snmpd.grp_2_usm=group
snmpd.grp_2_usm.version=usm
snmpd.grp_2_usm.group=public
snmpd.grp_2_usm.secname=ro
snmpd.grp_2_access=access
```

```
snmpd.grp_2_access.context=none
snmpd.grp_2_access.version=any
snmpd.grp_2_access.level=noauth
snmpd.grp_2_access.prefix=exact
snmpd.grp_2_access.read=all
snmpd.grp_2_access.write=all
snmpd.grp_2_access.notify=all
snmpd.grp_2_access.group=public
```

36.3.4.2 Group settings using package options

```
config 'group' 'public_v1'
      option group 'public'
      option version 'v1'
      option secname 'ro'

config 'group' 'public_v2c'
      option group 'public'
      option version 'v2c'
      option secname 'ro'

config 'group' 'public_usm'
      option group 'public'
      option version 'usm'
      option secname 'ro'

config 'group' 'private_v1'
      option group 'private'
      option version 'v1'
      option secname 'rw'
config 'group' 'private_v2c'
      option group 'private'

      option version 'v2c'
      option secname 'rw'

config 'group' 'private_usm'
      option group 'private'
```

```
      option version 'usm'
      option secname 'rw'
```

36.3.5 View settings

The following example defines two views, one for the entire system and another for only mib2.

36.3.5.1 View settings using UCI

```
snmpd.all=view
snmpd.all.viewname=all
snmpd.all.oid=.1
snmpd.mib2=view
snmpd.mib2.viewname=mib2
snmpd.mib2.type=included
snmpd.mib2.oid=.iso.org.dod.Internet.mgmt.mib-2
```

36.3.5.2 View settings using package options

```
config 'view' 'all'
      option viewname 'all'
      option type 'included'
      option oid '.1'

config 'view' 'mib2'
      option viewname 'mib2'
      option type 'included'
      option oid '.iso.org.dod.Internet.mgmt.mib-2'
```

36.3.6 Access settings

The following example shows the “public” group being granted read access on the “all” view and the “private” group being granted read and write access on the “all” view. Although it is possible to write some settings using SNMP write permission, it is not recommended, as any changes to the configuration made through an snmpset command may conflict with the UCI configuration. In this instance the changes will be overwritten by other processes and will not persist after a reboot.

36.3.6.1 Access using package options

```
config 'access' 'public_access'
      option group 'public'
      option context 'none'
      option version 'any'
```

```
      option level 'noauth'
      option prefix 'exact'
      option read 'all'
      option write 'none'
      option notify 'none'

config 'access' 'private_access'
      option group 'private'
      option context 'none'
      option version 'any'
      option level 'noauth'
      option prefix 'exact'
      option read 'all'
      option write 'all'
      option notify 'all'
```

36.3.7 SNMP traps settings

36.3.7.1 SNMP trap using UCI

```
snmpd.@trapreceiver[0]=trapreceiver
snmpd.@trapreceiver[0].host=1.1.1.1:161
snmpd.@trapreceiver[0].version=v1
snmpd.@trapreceiver[0].community=public
```

36.3.7.2 SNMP trap using package options

```
# for SNMPv1 or v2c trap receivers
config trapreceiver
   option host 'IPADDR[:PORT]'
   option version 'v1|v2c'
   option community 'COMMUNITY STRING'
# for SNMPv2c inform request receiver

config informreceiver
   option host 'IPADDR[:PORT]'
   option community 'COMMUNITY STRING'
```

36.4 Configuring SNMP interface alias with static SNMP index

A Linux interface index changes dynamically. This is not ideal for SNMP managers that require static interface indexes to be defined.

The network package interface section allows defining a static SNMP interface alias index for this interface.

An alias entry is created in the SNMP ifEntry table at index (**snmp_alias_ifindex + 1000**). This entry is a shadow of the real underlying Linux interface corresponding to the UCI definition. You may use any numbering scheme you wish; the alias values do not need to be consecutive.

36.4.1 Configuration package used

Package	Sections
network	interface

36.4.2 Configuring SNMP interface alias

To enter and SNMP alias for an interface, select **Network->Interfaces->Edit->Common Configuration->Advanced Settings.**

Enter a small index value for **SNMP Alias ifindex** that is unique to this interface. To retrieve SNMP statistics for this interface, the SNMP manager should be configured to poll (**snmp_alias_ifindex + 1000**). For example, if an interface is configured with an **snmp_alias_ifindex** of 11, then the SNMP manager should poll **ifIndex=1011**. The ifIndex will remain fixed regardless of how many times the underlying interface is added or removed.

If the Linux interface associated with the UCI entry is active when the alias index is polled, the normal ifEntry information for that interface is reported. Otherwise, a dummy entry is created with the same ifDescr, and its ifOper field set to **DOWN**.

Note: if you are using SIM roaming, where mobile interfaces are created dynamically, you need to specify a fixed **snmp_alias_ifindex** value and a fixed **ifName** value in the roaming template. All roaming entries will then map to the same Linux interface name and underlying device.

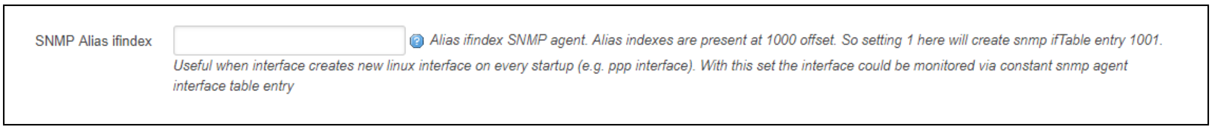


Figure 179: The interface SNMP Alias ifindex field advanced settings page

UCI / Package Option	Description
Web: SNMP Alias ifindex UCI: network.@interface[X].snmp_alias_ifindex Opt: snmp_alias_ifindex	Defines a static SNMP interface alias index for this interface, that can be polled using via the SNMP interface index (*snmp_alias_ifindex+1000*) Blank: No SNMP interface alias index Range: 0 - 4294966295
Web: n/a UCI: network.@interface[X].snmp_alias_ifdescr Opt: snmp_alias_ifdescr	Defines an alias name to be reported for the UCI name in the enterprise MIB for UCI interfaces, and in alias entries in the ifIndex table. If present, this option supercedes the default ifDescr value (usually the UCI interface name, or configured ifName) Blank: No SNMP interface alias name Range:

Table 122: Information table for static SNMP alias interface

36.4.3 Configuring SNMP interface alias using the command line

SNMP interface alias is configured under the network package **/etc/config/network**

The following examples use an interface section named MOBILE.

36.4.3.1 SNMP interface alias using UCI

```
root@VA_router:~# uci show network
network.MOBILE=interface
……
network.MOBILE.snmp_alias_ifindex=11
network.MOBILE.snmp_alias_ifdescr=primary_mobile
……
```

36.4.3.2 SNMP interface alias using package options

```
root@VA_router:~# uci show network
config interface ‘MOBILE’
……
      option snmp_alias_ifindex ‘11’
      option snmp_alias_ifdescr ‘primary_mobile’
……
```

36.4.4 SNMP interface alias MIBS

OID Name	OID
interface alias table	.1.3.6.1.2.1.2.2.1.1.
snmp_alias_ifindex	.1.3.6.1.2.1.2.2.1.1.<snmp_alias_ifindex+1000>
snmp_alias_ifdescr	1.3.6.1.4.1.2078.3.2.66.1.1.<index>.{5,6}

36.5 SNMP diagnostics

36.5.1 SNMP process

To check the SNMP process is running correctly, enter **pgrep -fl snmpd**.

```
root@VA_router:~# pgrep -fl snmpd
6970 /usr/sbin/snmpd -Lsd0-6 -p /var/run/snmpd.pid -m  -c
/var/conf/snmpd.conf
```

36.5.2 SNMP port

To check that SNMP service is listening on the configured port, enter **netstat -pantu | grep snmp**

```
root@VA_router:~# netstat -pantu | grep snmp
udp   0 0 0.0.0.0:161  0.0.0.0:*      6970/snmpd
```

36.5.3 Retrieving SNMP values

SNMP values can be queried by an snmpwalk or snmpget either locally or remotely.

36.5.3.1 snmpwalk

To do an snmpwalk locally, use **snmpwalk**. An example snmpwalk is shown below:

```
root@VA_router:~# snmpwalk -c public -v 1 localhost .1.3.6.1.2.1.1
iso.3.6.1.2.1.1.1.0 = STRING: "Virtual Access GWXXXX, SN# 00E0C812D1A0,
EDG-21.00.07.008"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.2078
iso.3.6.1.2.1.1.3.0 = Timeticks: (71816) 0:11:58.16
iso.3.6.1.2.1.1.4.0 = STRING: "info@virtualaccess.com"
iso.3.6.1.2.1.1.5.0 = STRING: "GWXXXX"
iso.3.6.1.2.1.1.6.0 = STRING: "UK"
iso.3.6.1.2.1.1.7.0 = INTEGER: 79
iso.3.6.1.2.1.1.8.0 = Timeticks: (60) 0:00:00.60
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.6.3.15.2.1.1
```

```
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.2.1.10.131
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (35) 0:00:00.35
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (38) 0:00:00.38
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (38) 0:00:00.38
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (38) 0:00:00.38
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (38) 0:00:00.38
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (60) 0:00:00.60
……
```

36.5.3.2 snmpget

To do an snmpget locally, use **snmpget.** An example snmpget is shown below.

```
root@VA_router:~# snmpget -c public -v 1 localhost .1.3.6.1.4.1.2078.3.14.2
iso.3.6.1.4.1.2078.3.14.2 = STRING: "EDG-21.00.07.008"
```

36.5.4 SNMP status

To view an overview including tx/rx packets and uptime of the SNMP process, enter **snmpstatus**.

```
root@VA_router:~# snmpstatus -c public -v 2c localhost
[UDP: [0.0.0.0]->[127.0.0.1]:161]=>[Virtual Access GWXXXX, SN#
00E0C812D1A0, EDG-21.00.07.008] Up: 0:17:05.87
Interfaces: 21, Recv/Trans packets: 47632/9130 | IP: 15045/8256
15 interfaces are down!
```

37 Event system

Virtual Access routers feature an event system. It allows you to forward Virtual Access specific router events to predefined targets for efficient control and management of devices.

This chapter explains how the event system works and how to configure it using UCI commands.

37.1 Configuration package used

Package	Section
va_eventd	main
	forwarding
	target
	conn_tester

37.2 Event system overview

37.2.1 Implementation of the event system

The event system is implemented by the **va_eventd** application.

The va_eventd application defines three types of object:

Forwardings	Rules that define what kind of events should be generated. For example, you might want an event to be created when an IPSec tunnel comes up or down.
Targets	Define the targets to send the event to. The event may be sent to a target via a syslog message, a snmp trap or email.
Connection testers	Define methods to test the target is reachable. IP connectivity to a server and link state may be checked prior to sending events.

For example, if you want to configure an SNMP trap to be sent when an IPSec tunnel comes up, you will need to:

- Define a forwarding rule for IPSec tunnel up events.
- Set an SNMP manager as the target.
- Optionally use a connection tester to ensure the SNMP manager is reachable.

37.2.2 Supported events

Events have a class, ID, name and a severity. These properties are used to fine tune which events to report.

Note: only VA events can be forwarded using the event system. A comprehensive table of events is available from the CLI by entering '**vae_cli -d**'.

37.2.3 Supported targets

The table below describes the targets currently supported.

Target	Description
Syslog	Event sent to syslog server.
Email	Event sent via email.
SNMP	Event sent via SNMP trap.
Exec	Command executed when event occurs.
SMS	Event sent via SMS.
File	Events written to a file

Table 123: Targets currently supported

The attributes of a target vary significantly depending on its type.

37.2.4 Supported connection testers

The table below describes the methods to test a connection that are currently supported.

Type	Description
link	Checks if the interface used to reach the target is up.
ping	Pings the target. And then assumes there is connectivity during a configurable amount of time.

Table 124: Event system - supported connection tester methods

37.3 Configuring the event system using the web interface

To configure the event system, select **Services->VA Event System**. The VA Event System page appears.

There are four sections in the VA Event System page.

Section	Description
Basic Settings	Configures basic global event system parameters.
Connection Tester	Configures the connection testers.
Events Destination	Configures the event targets.
Event Filters	Configures the forwarding rules.

37.3.1 Basic settings

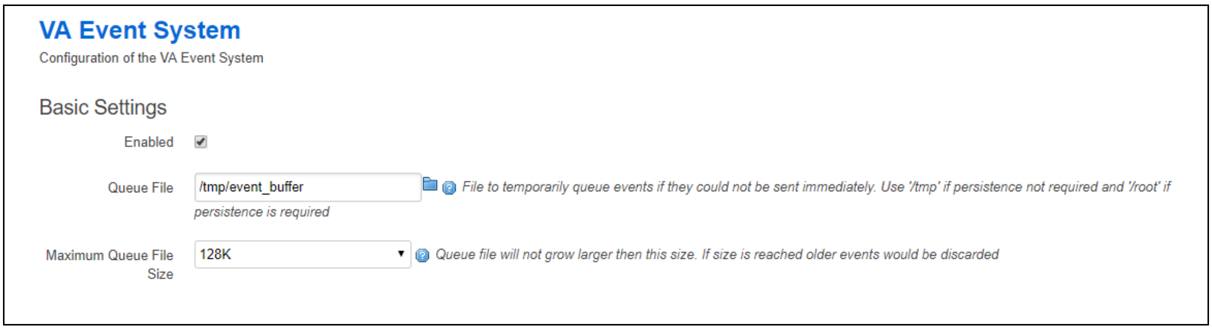


Figure 180: The VA event system basic settings configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: va_eventd.main.enabled
Opt: enabled</td><td>Enables VA Event System.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Enabled
UCI: va_eventd.main.event_queue_file
Opt: event_queue_file</td><td>Defines the file to temporarily queue events when they cannot be sent immediately.
Note: Use <code>/tmp</code> path if persistence is not required and <code>/root</code> if persistence is required.<table><tr><td>/tmp/event_buffer</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Enabled
UCI: va_eventd.main.event_queue_size
Opt: event_queue_size</td><td>Defines the file size for the temporary queue. Older events are discarded once file size is reached.<table><tr><td>128K</td><td>128 Kilobytes</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Table 125: Information table for event system basic settings

37.3.2 Connection tester

A connection tester is used to verify the event destination before forwarding the event. Connection testers configure the uci `conn_tester` section rules. Multiple connection testers can be configured. There are two types of connection tester:

Type	Description
link	Checks if the interface used to reach the target is up.
ping	Pings the target. And then assumes there is connectivity during a configurable amount of time.

Figure 181: The VA event system connection tester configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: va_eventd.@conn_tester[0].enabled
Opt: enabled</td><td>Enables a connection tester.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name
UCI: va_eventd.@conn_tester[0].name
Opt: name</td><td>Defines the connection tester name.
This is used when configuring a connection tester for an event destination.</td></tr>
<tr><td>Web: Type
UCI: va_eventd.@conn_tester[0].type
Opt: type</td><td>Defines the connection tester type.<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Ping</td><td>Verifies target by ping</td><td>ping</td></tr><tr><td>Link</td><td>Verifies target by checking routed interface is up</td><td>link</td></tr></table></td></tr>
</table>

<table>
<tr><td>Web: Ping Target
UCI: va_eventd.@conn_tester[0].ping_dest_addr
Opt: ping_dest_addr</td><td>Defines the IP address for the target ping.
Note: only displayed if connection tester type is set to 'Ping'.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Ping Source
UCI: va_eventd.@conn_tester[0].ping_source
Opt: ping_source</td><td>Defines an interface or IP address to source the pings from.
Note: only displayed if connection tester type is set to 'Ping'.<table><tr><td>eth0</td><td>Use eth0 IP for ping source</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Ping Success Duration
UCI: va_eventd.@conn_tester[0].ping_success_duration_sec
Opt: ping_success_duration_sec</td><td>Defines the duration, in seconds, for which a successful ping defines a connection tester as up.
Note: only displayed if connection tester type is set to 'Ping'.<table><tr><td>60</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Link Interface
UCI: va_eventd.@conn_tester[0].link_iface
Opt: link_iface</td><td>Defines the interface to monitor when the connection tester type is set to 'link'. Configured interfaces are listed.
Note: only displayed if connection tester type is set to 'Link'.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Table 126: Information table for event system connection tester settings

37.3.3 Event Destination

An event destination is the target for the event. Event destinations configure the uci `target` section rules. Multiple event destinations can be configured. There are currently six configurable event destinations

Target Type	Description
Syslog	Event sent to syslog server.
Email	Event sent via email.
SNMP	Event sent via SNMP trap.
Execute	Command executed when event occurs.
SMS	Event sent via SMS.
File	Event written to a file

The available configuration options differ depending on the event destination type.

37.3.3.1 Syslog target

When a syslog target receives an event, it sends it to the configured syslog server.

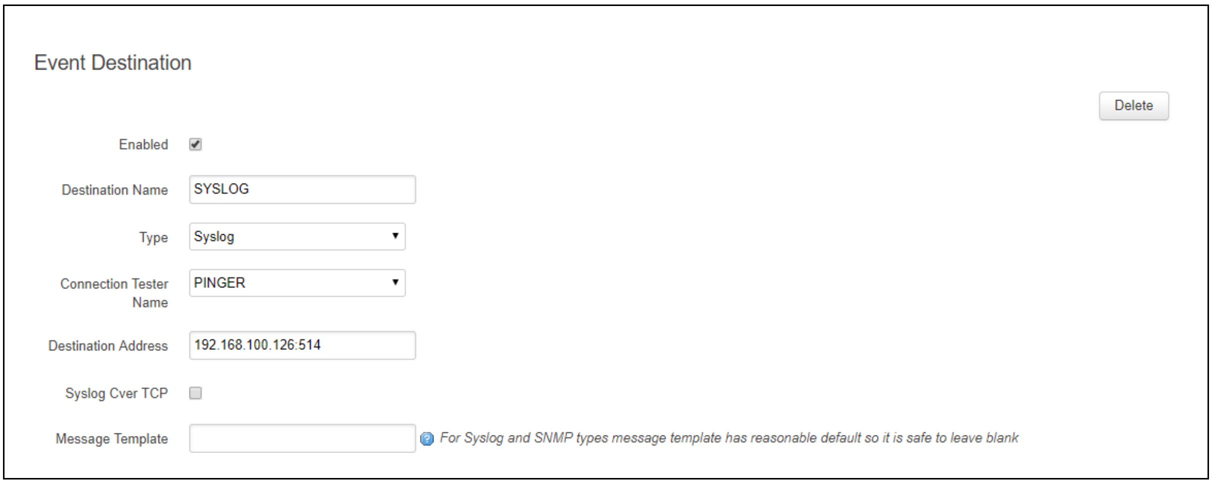


Figure 182: The VA event system syslog event destination configuration page

Web Field / UCI / Package Option	Description
Web: Enabled UCI: va_eventd.@target[0].enabled Opt: enabled	Enables an event destination. This is used in the event filters section. 0 — Disabled. 1 — Enabled.
Web: Destination name UCI: va_eventd.@target[0].name Opt: name	Defines a name for the event destination. Range
Web: Type UCI: va_eventd.@target[0].type Opt: type	Defines the event destination type. For syslog server choose **Syslog**. **Web Value** | **Description** | **UCI** Syslog | | syslog SNMP Trap | | snmptrap Email | | email Execute | | exec SMS | | sms n/a | File target | file
Web: Connection Tester Name UCI: va_eventd.@target[0]. conn_tester Opt: conn_tester	Defines the connection tester (if any) to use to verify the syslog target. None — No connection tester. UCI option not present. Range
Web: Destination Address UCI: va_eventd.@target[0]. target_addr Opt: target_addr	Defines the syslog target IP/FQDN and port. Range — a.b.c.d:port or fqdn:port
Web: Syslog Over TCP UCI: va_eventd.@target[0].tcp_syslog Opt: tcp_syslog	Defines whether to use TCP for delivery of the syslog event. 0 — Use UDP 1 — Use TCP
Web: Message Template UCI: va_eventd.@target[0].template Opt: template	Defines the message template to use for the event. In general, this should be left empty. See the section on message templates below. Range

Table 127: Information table for event system syslog event destination settings

37.3.3.2 Email target

When an email target receives an event, it sends it to the configured email address.

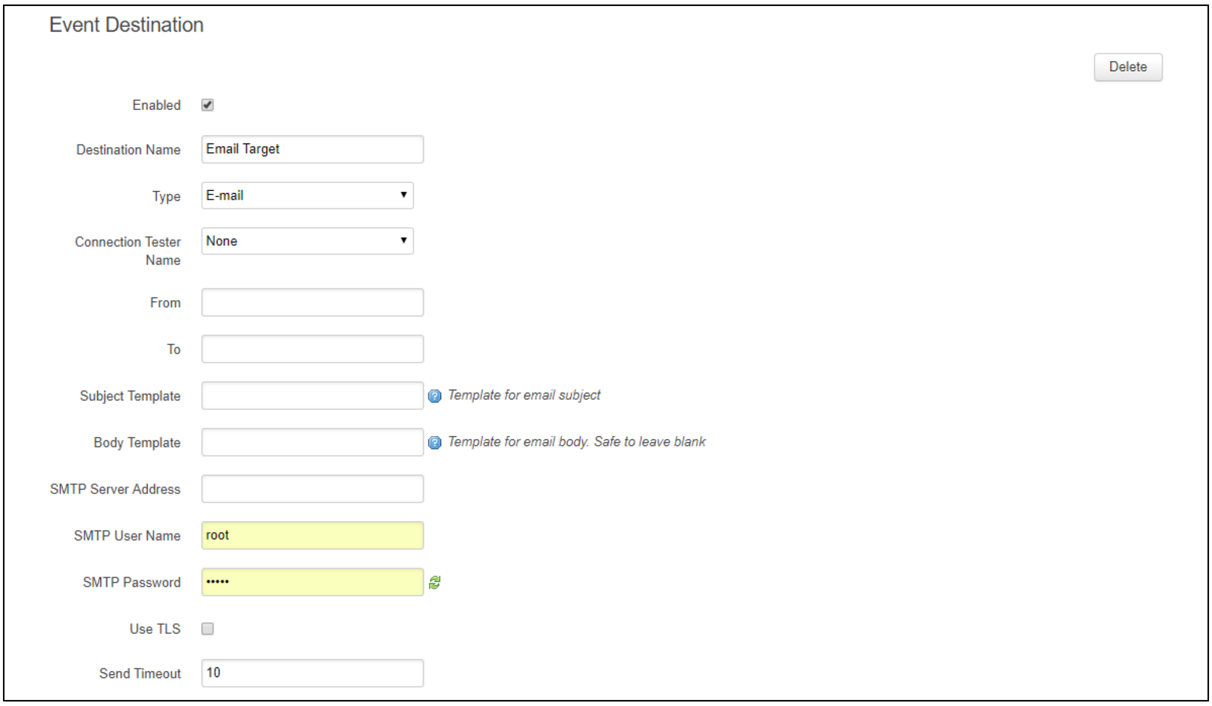


Figure 183: The VA event system email event destination configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: va_eventd.@target[0].enabled
Opt: enabled</td><td>Enables an event destination.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Destination name
UCI: va_eventd.@target[0].name
Opt: name</td><td>Defines a name for the event destination.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Type
UCI: va_eventd.@target[0].type
Opt: type</td><td>Defines the event destination type. For an email server choose Email.<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td>Syslog target</td><td>syslog</td></tr><tr><td>SNMP Trap</td><td>SNMP target</td><td>snmptrap</td></tr><tr><td>Email</td><td>Email target</td><td>email</td></tr><tr><td>Execute</td><td>Execure target</td><td>exec</td></tr><tr><td>SMS</td><td>SMS target</td><td>sms</td></tr><tr><td>n/a</td><td>File target</td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name
UCI: va_eventd.@target[0]. conn_tester
Opt: conn_tester</td><td>Defines the connection tester (if any) to use to verify the email target.<table><tr><td>None</td><td>No connection tester. UCI option not present.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: From
UCI: va_eventd.@target[0].from
Opt: from</td><td>Defines the from address for the email.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: To
UCI: va_eventd.@target[0].to
Opt: to</td><td>Defines to address for the email.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web: Subject Template UCI: va_eventd.@target[0].subject_template Opt: subject_template	Defines subject template for the email. In general, this should be left empty. Example: `va_eventd.@target[0].subject_template="%{severityName} %{eventName}!!!"` See the section on message templates below. Range
Web: Body Template UCI: va_eventd.@target[0].body_template Opt: body_template	Defines the email body template. In general, this should be left blank. Example: `va_eventd.@target[0].body_template="%{eventName} (%{class}.%{subclass}) happened!"` See the section on message templates below. Range
Web: SMTP Server Address UCI: va_eventd.@target[0].smtp_addr Opt: smtp.addr	Defines the email server address and port. Range: a.b.c.d:port or fqdn:port
Web: SMTP User Name UCI: va_eventd.@target[0].smtp_user Opt: smtp_user	Defines user name for SMTP authentication. Range: name@site.com
Web: SMTP Password UCI: va_eventd.@target[0].smtp_password Opt: smtp_password	Defines the password for SMTP authentication. Range
Web: Use TLS UCI: va_eventd.@target[0].use_tls Opt: use_tls	Enables TLS (Transport Layer Security) support. 0 1
Web: Send Timeout UCI: va_eventd.@target[0].timeout_sec Opt: timeout_sec	Defines the email send timeout in seconds. 10 Range
Web: Use StartTLS UCI: va_eventd.@target[0]. tls_starttls Opt: tls_starttls	Enables StartTLS support for TLS. (Only displayed when TLS is enabled) 0 1
Web: Force SSLv3 UCI: va_eventd.@target[0].tls_forcessl3 Opt: tls_forcessl3	Enables force SSLv3 for TLS. (Only displayed when TLS is enabled) 0 1

Table 128: Information table for event system email event destination settings

37.3.3.3 SNMP target

When a SNMP target receives an event, it sends it in a trap to the configured SNMP manager.

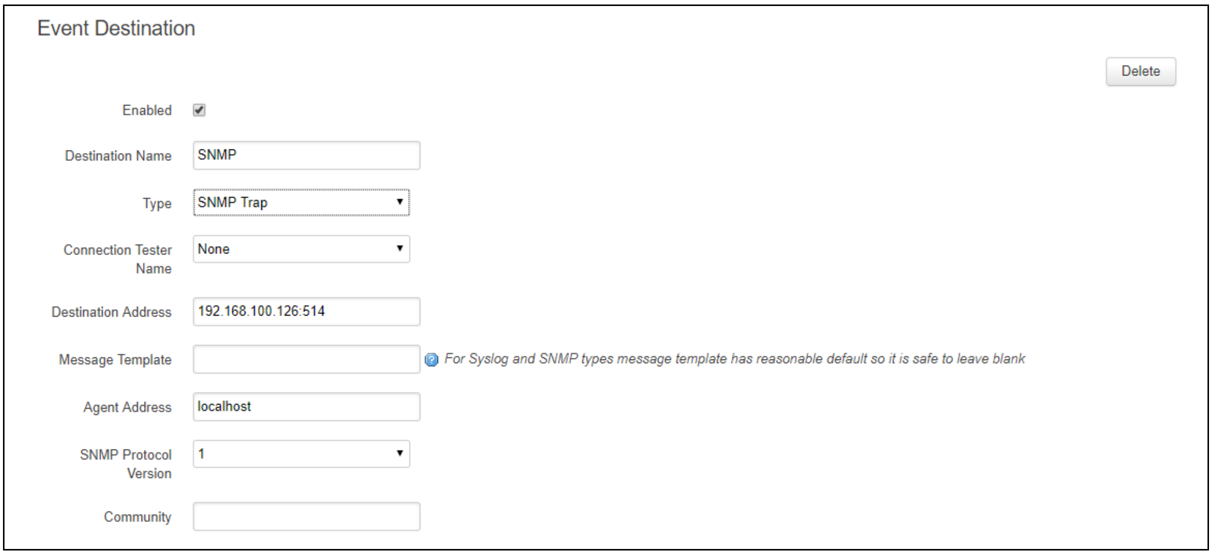


Figure 184: The VA event system SNMP event destination configuration page

Web Field / UCI / Package Option	Description
Web: Enabled UCI: va_eventd.@target[0].enabled Opt: enabled	Enables an event destination. 0 — Disabled. 1 — Enabled.
Web: Destination name UCI: va_eventd.@target[0].name Opt: name	Defines a name for the event destination. Range
Web: Type UCI: va_eventd.@target[0].type Opt: type	Defines the event destination type. For SNMP server choose **SNMP Trap**. **Web Value** / **Description** / **UCI** Syslog / Syslog target / syslog SNMP Trap / SNMP target / snmptrap Email / Email target / email Execute / Execute target / exec SMS / SMS target / sms n/a / File target / file
Web: Connection Tester Name UCI: va_eventd.@target[0]. conn_tester Opt: conn_tester	Defines the connection tester (if any) to use to verify the SNMP target. None — No connection tester. UCI option not present. Range
Web: Destination Address UCI: va_eventd.@target[0]. target_addr Opt: target_addr	Defines the SNMP target IP/FQDN and port. Range — a.b.c.d:port or fqdn:port
Web: Message Template UCI: va_eventd.@target[0].template Opt: template	Defines the message template to use for the event. In general, this should be left empty. Example: `va_eventd.@target[0].template="%{eventName} %{eventSpecificTemplate}"` See the section on message templates below. Range
Web: Agent Address UCI: va_eventd.@target[0]. agent_addr Opt: agent_addr	Defines the IP address to source the SNMP trap. (optional) localhost Range

Web: SNMP Protocol Version UCI: va_eventd.@target[0].snmp_version Opt: snmp_version	Defines the SNMP version. 1 – SNMPv1 2c – SNMPv2c 3 – SNMPv3
Web: Community UCI: va_eventd.@target[0].community Opt: community	Defines the community string for SNMPv1. Range
Web: Username UCI: va_eventd.@target[0].snmp_uname Opt: snmp_uname	Defines the username for SNMPv3. (Only displayed when SNMP protocol version is SNMPv3) Range
Web: Authentication Protocol UCI: va_eventd.@target[0].snmp_auth_proto Opt: snmp_auth_proto	Defines the SNMPv3 authentication protocol (Only displayed when SNMP protocol version is SNMPv3) MD5 SHA
Web: Authentication Password UCI: va_eventd.@target[0].snmp_auth_pass Opt: snmp_auth_pass	Defines the SNMPv3 authentication password (Only displayed when SNMPv3 authentication protocol is configured) MD5 SHA
Web: Privacy Protocol UCI: va_eventd.@target[0].snmp_priv_proto Opt: snmp_priv_proto	Defines the SNMPv3 privacy protocol. (Only displayed when SNMP authentication protocol is configured) DES AES
Web: Privacy Password UCI: va_eventd.@target[0].snmp_priv_pass Opt: snmp_priv_pass	Defines SNMPv3 privacy password. (Only displayed when SNMP privacy protocol is configured) Range
Web: SNMPv3 Context UCI: va_eventd.@target[0].snmp_context Opt: snmp_context	Defines the SNMPv3 context. (Only displayed when SNMP authentication protocol is configured) Range
Web: SNMPv3 Context Engine ID UCI: va_eventd.@target[0].snmp_context_eid Opt: snmp_context_eid	Defines the SNMPv3 context engine ID. (Only displayed when SNMP authentication protocol is configured) Range
Web: SNMPv3 Security Engine ID UCI: va_eventd.@target[0].snmp_sec_eid Opt: snmp_sec_eid	Defines the SNMPv3 security engine ID. (Only displayed when SNMP authentication protocol is configured) Range

Table 129: Information table for event system SNMP event destination settings

37.3.3.4 Exec target

When an Execute target receives an event, it executes a shell command.

Figure 185: The VA event system Exec event destination configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: va_eventd.@target[0].enabled
Opt: enabled</td><td>Enables an event destination.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Destination name
UCI: va_eventd.@target[0].name
Opt: name</td><td>Defines a name for the event destination.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Type
UCI: va_eventd.@target[0].type
Opt: type</td><td>Defines the event destination type. For shell command execution, choose Execute.<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td>Syslog target</td><td>syslog</td></tr><tr><td>SNMP Trap</td><td>SNMP target</td><td>snmptrap</td></tr><tr><td>Email</td><td>Email target</td><td>email</td></tr><tr><td>Execute</td><td>Execure target</td><td>exec</td></tr><tr><td>SMS</td><td>SMS target</td><td>sms</td></tr><tr><td>n/a</td><td>File target</td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name
UCI: va_eventd.@target[0]. conn_tester
Opt: conn_tester</td><td>Defines the connection tester (if any) to use to verify the execute target.<table><tr><td>None</td><td>No connection tester. UCI option not present.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Command Template
UCI: va_eventd.@target[0].cmd_template
Opt: cmd_template</td><td>Defines the command template to use for the event.
Example to log a syslog message:
<code>va_eventd.@target[0].cmd_template="logger -t eventer %{eventName}"</code>
See the section on message templates below.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Table 130: Information table for event system Execute event destination settings

37.3.3.5 SMS target

When an SMS target receives an event, it sends an SMS message.

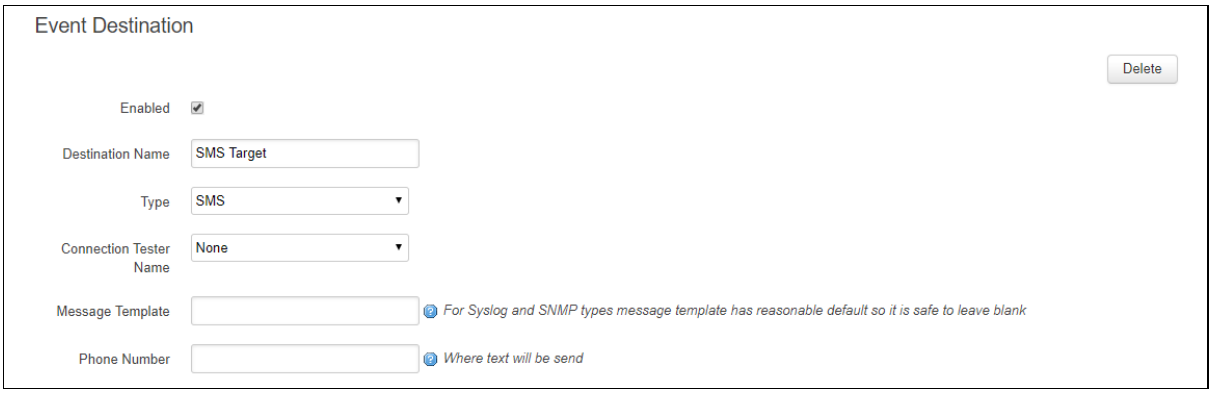


Figure 186: The VA event system SMS event destination configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: va_eventd.@target[0].enabled
Opt: enabled</td><td>Enables an event destination.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Destination name
UCI: va_eventd.@target[0].name
Opt: name</td><td>Defines a name for the event destination.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Type
UCI: va_eventd.@target[0].type
Opt: type</td><td>Defines the event destination type. For SMS destination choose SMS.<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td></td><td>syslog</td></tr><tr><td>SNMP Trap</td><td></td><td>snmptrap</td></tr><tr><td>Email</td><td></td><td>email</td></tr><tr><td>Execute</td><td></td><td>exec</td></tr><tr><td>SMS</td><td></td><td>sms</td></tr><tr><td>n/a</td><td></td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name
UCI: va_eventd.@target[0]. conn_tester
Opt: conn_tester</td><td>Defines the connection tester (if any) to use to verify the SMS target.<table><tr><td>None</td><td>No connection tester. UCI option not present.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Message Template
UCI: va_eventd.@target[0].template
Opt: template</td><td>Defines the message template to use for the event. In general, this should be left empty. Example:
<code>va_eventd.@target[0].template="%{eventName}"</code>
See the section on message templates below.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Phone Number
UCI: va_eventd.@target[0].callee
Opt: callee</td><td>Defines the phone number for sending SMS to.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Table 131: Information table for event system SMS event destination settings

37.3.3.6 File target

When file target receives an event, it logs to a file.

There is currently no web support for a file target.

Web Field / UCI / Package Option	Description
Web: n/a UCI: va_eventd.@target[0].enabled Opt: enabled	Enables an event destination. 0 – Disabled. 1 – Enabled.
Web: n/a UCI: va_eventd.@target[0].name Opt: name	Defines a name for the event destination. Range
Web: n/a UCI: va_eventd.@target[0].type Opt: type	Defines the event destination type. For file choose **Syslog**. **Web Value** – **Description** – **UCI** Syslog – – syslog SNMP Trap – – snmptrap Email – – email Execute – – exec SMS – – sms n/a – – file
Web: n/a UCI: va_eventd.@target[0].file_name Opt: file_name	Defines a file name for the event destination. (full path) Range
Web: n/a UCI: va_eventd.@target[0].max_size_kb Opt: file_name	Defines a file size in kilobits. 2048 Range
Web: n/a UCI: va_eventd.@target[0].template Opt: template	Defines the message template to use for the event. In general, this should be left empty. See the section on message templates below. Range

Table 132: Information table for event system file event destination settings

37.3.4 Event filters

Event filters are use to classify the events to be sent to the event destination. Multiple event filters can be defined. Event filters configure the uci `forwarding` section rules.

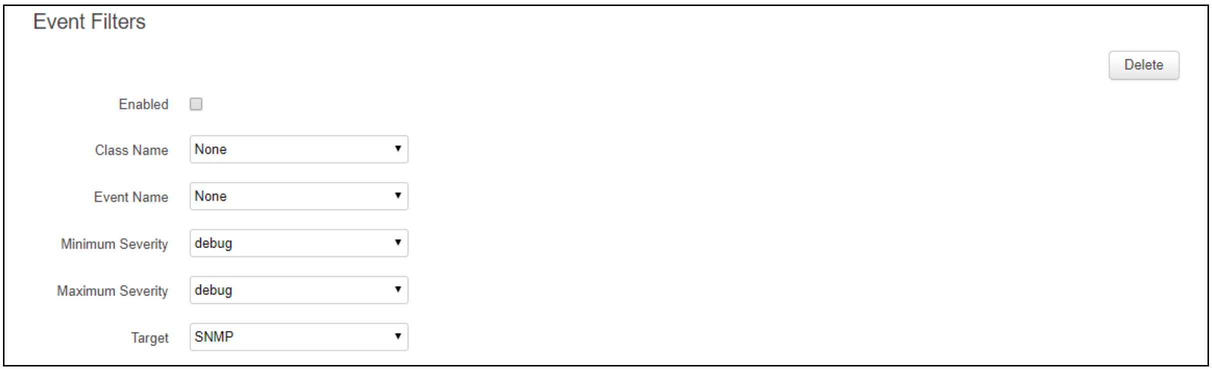


Figure 187: The VA event system event filters configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: va_eventd.@forwarding[0].enabled
Opt: enabled</td><td>Enables an event filter.<table><tr><td>1</td><td>Disabled.</td></tr><tr><td>0</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Class Name
UCI: va_eventd.@forwarding[0].className
Opt: className</td><td>Only match events with the given class name. Available class names are listed or can be viewed using the command <code>vae_cli -d</code></td></tr>
<tr><td>Web: Event Name
UCI: va_eventd.@forwarding[0].eventName
Opt: eventName</td><td>Only match events with the given event name. Available event names are listed. The event name is optional and can be omitted.</td></tr>
<tr><td>Web: Minimum Severity
UCI: va_eventd.@forwarding[0].severity
Opt: severity</td><td>Defines the minimum event severity. The minimum severity event is DEBUG. Events generated within the minimum and maximum event severity will be matched.
Minimum and maximum severity are specified in the one UCI option and entered using a dash (-) separator in the form minimum-maximum. Example:
<code>va_eventd.@forwarding[0].severity=debug-error</code><table><tr><td>debug</td><td>minimum severity</td></tr><tr><td>info</td><td></td></tr><tr><td>notice</td><td></td></tr><tr><td>warning</td><td></td></tr><tr><td>error</td><td></td></tr><tr><td>critical</td><td></td></tr><tr><td>alert</td><td></td></tr><tr><td>emergency</td><td>maximum severity</td></tr></table></td></tr>
<tr><td>Web: Maximum Severity
UCI: va_eventd.@forwarding[0].severity
Opt: severity</td><td>Defines the maximum event severity. The maximum event severity is EMERGENCY. Events generated within the minimum and maximum event severity will be matched.
The UCI command for specifying minimum and maximum severity is the same and is entered with two parameters using a dash (-) separator minimum-maximum. Example:
<code>va_eventd.@forwarding[0].severity=debug-error</code><table><tr><td>debug</td><td>minimum severity</td></tr><tr><td>info</td><td></td></tr><tr><td>notice</td><td></td></tr><tr><td>warning</td><td></td></tr><tr><td>error</td><td></td></tr><tr><td>critical</td><td></td></tr><tr><td>alert</td><td></td></tr><tr><td>emergency</td><td>maximum severity</td></tr></table></td></tr>
<tr><td>Web: Target
UCI: va_eventd.@forwarding[0].target
Opt: target</td><td>Defines the event destination to forward the event to. All configured event destinations will be displayed.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Table 133: Information table for event system event filters settings

37.4 Configuring the event system using command line

The event system configuration files are stored at **/etc/config/va_eventd**

There are four config sections main, conn_tester, target and forwarding.

You can configure multiple conn_tester, target and forwarding sections.

By default, all conn_tester instances are named conn_tester, it is identified by `@conn_tester` then the conn_tester position in the package as a number. For example, for the first conn_tester in the package using UCI:

```
va_eventd.@conn_tester[0]=conn_tester
va_eventd.@conn_tester[0].enabled=1
```

Or using package options:

```
config conn_tester
      option enabled '1'
```

By default, all target instances are named target. The target instance is identified by `@target` then the target position in the package as a number. For example, for the first target in the package using UCI:

```
va_eventd.@target[0]=target
va_eventd.@target[0].enabled=1
```

Or using package options:

```
config target
      option enabled '1'
```

By default, all forwarding instances are named forwarding. The forwarding instance is identified by `@forwarding` then the forwarding position in the package as a number. For example, for the first forwarding rule in the package using UCI:

```
va_eventd.@lforwarding[0]=forwarding
va_eventd.@forwarding[0].enabled=1
```

Or using package options:

```
config forwarding
      option enabled '1'
```

37.4.1 Event system using UCI

```
root@VA_router:~# uci show va_eventd
#Sample basic settings
va_eventd.main=va_eventd
va_eventd.main.event_queue_file=/tmp/event_buffer
va_eventd.main.event_queue_size=128K

#Sample SNMP
va_eventd.@conn_tester[0]=conn_tester
```

```
va_eventd.@conn_tester[0].type=ping
va_eventd.@conn_tester[0].ping_dest_addr=192.168.100.1
va_eventd.@conn_tester[0].ping_success_duration_sec=60
va_eventd.@conn_tester[0].name=SNMPTest
va_eventd.@conn_tester[0].ping_source=LAN1
va_eventd.@target[0]=target
va_eventd.@target[0].suppress_duplicate_forwardings=no
va_eventd.@target[0].type=snmp
va_eventd.@target[0].agent_addr=localhost
va_eventd.@target[0].name=SNMPTarget
va_eventd.@target[0].conn_tester=SNMPTest
va_eventd.@target[0].target_addr=192.168.100.126:68
va_eventd.@target[0].snmp_version=3
va_eventd.@target[0].snmp_uname=v3username
va_eventd.@target[0].snmp_auth_proto=MD5
va_eventd.@target[0].snmp_auth_pass=md5password
va_eventd.@target[0].snmp_priv_proto=AES
va_eventd.@target[0].snmp_priv_pass=aespassword
va_eventd.@target[0].snmp_context=v3context
va_eventd.@target[0].snmp_context_eid=v3contextID
va_eventd.@target[0].snmp_sec_eid=v3SecurityID
va_eventd.@forwarding[0]=forwarding
va_eventd.@forwarding[0].enabled=yes
va_eventd.@forwarding[0].className=mobile
va_eventd.@forwarding[0].target=SNMPTarget
va_eventd.@forwarding[0].eventName=LinkUp
va_eventd.@forwarding[0].severity=notice-notice

#Sample Syslog
va_eventd.@conn_tester[1]=conn_tester
va_eventd.@conn_tester[1].name=SyslogTest
va_eventd.@conn_tester[1].type=ping
va_eventd.@conn_tester[1].ping_dest_addr=192.168.100.2
va_eventd.@conn_tester[1].ping_source=LAN1
va_eventd.@conn_tester[1].ping_success_duration_sec=60
va_eventd.@target[1]=target
va_eventd.@target[1].name=SyslogTarget
```

```
va_eventd.@target[1].type=syslog
va_eventd.@target[1].conn_tester=SyslogTest
va_eventd.@target[1].target_addr=192.168.100.2:514
va_eventd.@target[1].tcp_syslog=0
va_eventd.@forwarding[1]=forwarding
va_eventd.@forwarding[1].enabled=yes
va_eventd.@forwarding[1].severity=debug-error
va_eventd.@forwarding[1].target=SyslogTarget

#Sample Email
va_eventd.@conn_tester[2]=conn_tester
va_eventd.@conn_tester[2].name=EmailTest
va_eventd.@conn_tester[2].type=link
va_eventd.@conn_tester[2].link_iface=PoAADSL
va_eventd.@target[2]=target
va_eventd.@target[2].timeout_sec=10
va_eventd.@target[2].name=EmailTarget
va_eventd.@target[2].type=email
va_eventd.@target[2].conn_tester=EmailTest
va_eventd.@target[2].from=from@example.com
va_eventd.@target[2].to=to@example.com
va_eventd.@target[2].subject_template=%{serial} %{severityName} %{eventName
}!!!
va_eventd.@target[2].body_template=%{eventName} (%{class}.%{subclass})
happened!
va_eventd.@target[2].smtp_addr=192.168.100.3:25
va_eventd.@target[2].smtp_user=root
va_eventd.@target[2].smtp_password=admin
va_eventd.@target[2].use_tls=0
va_eventd.@target[2].tls_starttls=0
va_eventd.@target[2].tls_forcessl3=0
va_eventd.@forwarding[2]=forwarding
va_eventd.@forwarding[2].enabled=yes
va_eventd.@forwarding[2].className=power
va_eventd.@forwarding[2].eventName=IgnitionOff
va_eventd.@forwarding[2].severity=notice-notice
va_eventd.@forwarding[2].target=EmailTarget
```

```
#Sample SMS
va_eventd.@target[3]=target
va_eventd.@target[3].name=SMStarget
va_eventd.@forwarding[3].target=SMStarget
va_eventd.@target[3].type=sms
va_eventd.@target[3].template=%{serial} %{severityName} %{eventName}!!!
va_eventd.@target[3].callee=0123456789
va_eventd.@forwarding[3]=forwarding
va_eventd.@forwarding[3].enabled=yes
va_eventd.@forwarding[3].target=SMStarget
va_eventd.@forwarding[3].className=auth
va_eventd.@forwarding[3].eventName=LoginSSH
va_eventd.@forwarding[3].severity=notice-notice

#Sample Execute
va_eventd.@target[4]=target
va_eventd.@target[4].name=ExecTarget
va_eventd.@target[4].type=exec
va_eventd.@target[4].cmd_template=logger -t eventer %{eventName}
va_eventd.@forwarding[4]=forwarding
va_eventd.@forwarding[4].enabled=yes
va_eventd.@forwarding[4].target=ExecTarget
va_eventd.@forwarding[4].className=ppp
va_eventd.@forwarding[4].severity=debug-error

#Sample File
va_eventd.@target[5]=target
va_eventd.@target[5].name=FileTarget
va_eventd.@target[5].type=file
va_eventd.@target[5].file_name=\tmp\eventfile
va_eventd.@target[5].max_size_kb=1028
va_eventd.@forwarding[5]=forwarding
va_eventd.@forwarding[5].enabled=yes
va_eventd.@forwarding[5].target=FileTarget
va_eventd.@forwarding[5].severity=debug-error
```

37.4.1.1 Event system using package options

```
root@VA_router:~# uci export va_eventd
package va_eventd

config va_eventd 'main'
        option event_queue_file '/tmp/event_buffer'
        option event_queue_size '128K'

# Sample SNMP
config conn_tester
        option type 'ping'
        option ping_dest_addr '192.168.100.1'
        option ping_success_duration_sec '60'
        option name 'SNMPTest'
        option ping_source 'LAN1'

config target
        option suppress_duplicate_forwardings 'no'
        option type 'snmp'
        option agent_addr 'localhost'
        option name 'SNMPTarget'
        option conn_tester 'SNMPTest'
        option target_addr '192.168.100.126:68'
        option snmp_version '3'
        option snmp_uname 'v3username'
        option snmp_auth_proto 'MD5'
        option snmp_auth_pass 'md5password'
        option snmp_priv_proto 'AES'
        option snmp_priv_pass 'aespassword'
        option snmp_context 'v3context'
        option snmp_context_eid 'v3contextID'
        option snmp_sec_eid 'v3SecurityID'

config forwarding
        option enabled 'yes'
        option className 'mobile'
        option severity 'notice-notice'
```

```
        option target 'SNMPTarget'
        option eventname 'LinkUp'

# Sample Syslog
config conn_tester
        option name 'SyslogTest'
        option type 'ping'
        option ping_dest_addr '192.168.100.2'
        option ping_source 'LAN1'
        option ping_success_duration_sec '60'

config target
        option name 'SyslogTarget'
        option type 'syslog'
        option conn_tester 'SyslogTest'
        option target_addr '192.168.100.2:514'
        option tcp_syslog '0'

config forwarding
        option enabled 'yes'
        option severity 'debug-error'
        option target 'SyslogTarget'

# Sample Email
config conn_tester
        option name 'EmailTest'
        option type 'link'
        option link_iface 'PoAADSL'

config target
        option timeout_sec '10'
        option name 'EmailTarget'
        option type 'email'
        option conn_tester 'EmailTest'
        option from 'from@example.com'
        option to 'to@example.com'
        option subject_template '%{serial} %{severityName} %{eventName}!!!'
```

```
        option body_template '%{eventName} (%{class}.%{subclass})
happened!'
        option smtp_addr '192.168.100.3:25'
        option smtp_user 'root'
        option smtp_password 'admin'
        option use_tls 'no'
        option tls_starttls 'no'
        option tls_forcessl3 'no'

config forwarding
        option enabled 'yes'
        option target 'EmailTarget'
        option className 'power'
        option eventName 'IgnitionOff'
        option severity 'notice-notice'

# Sample SMS
config target
        option name 'SMStarget'
        option type 'sms'
        option template '%{serial} %{severityName} %{eventName}!!!'
        option callee '0123456789'

config forwarding
        option enabled 'yes'
        option target 'SMSTarget'
        option className 'auth'
        option eventName 'LoginSSH'
        option severity 'notice-notice'

# Sample Execute
config target
        option name 'ExecTarget'
        option type 'exec'
        option cmd_template 'logger -t eventer %{eventName}'

config forwarding
```

```
        option enabled 'yes'
        option target 'ExecTarget'
        option className 'ppp'
        option severity 'debug-error'

# Sample File
config target
        option name 'FileTarget'
        option type 'file'
        option file_name '\tmp\eventfile'
        option max_size_kb '1028'

config forwarding
        option enabled 'yes'
        option target 'FileTarget'
        option severity 'debug-error'
```

37.5 Event system diagnostics

37.5.1 Displaying VA events

To view a list of all available class names, events and severity levels, enter:

```
root@VA_router:~# vae_cli -d
```

The following is an example of the output from this command:

```
| Class     | ID  | Name                       | Severity | Specific Template
| internal  |   1 | EventdConfigErr            | error
| %{p1} %{p2}: %{p3} has bad value..
| internal  |   2 | EventdConfigWarn           | warning
| %{p1} %{p2}: %{p3} has bad value..
| internal  |   3 | EventdConfigUnknown        | informat | %{p1} %{p2}:
field '%{p3}' is no..

| internal  |   4 | EventdSystemErr            | error
| %{p1} %{p2}: %{p3} %{p4} %{p5} %..
| internal  |   5 | EventdSystemWarn           | error
| %{p1} %{p2}: %{p3} %{p4} %{p5} %..
| internal  |   6 | EventdUpAndRunning         | informat |
| internal  |   7 | EventdStopped              | warning  | %{p1}
| mobile    |   1 | SIMin          | notice   | SIM card #%{p1}inserted
```

```
| mobile    |   2 | SIMout        | notice   | SIM card #%{p1} removed
| mobile    |   3 | LinkUp        | notice   | 3g link %{p1} up using sim
#%{p2..
| mobile    |   4 | LinkDown      | notice   | 3g link %{p1} down
| mobile    |   5 | SMSByPassword | notice   | Received SMS from %{p1} (by
pass..
| mobile    |   6 | SMSByCaller   | notice   | Received SMS from %{p1}
(%{p2}):..
| mobile    |   7 | SMSFromUnknown     | warning  | Received SMS from
unknown sender..
| mobile    |   8 | SMSSendSuccess     | informat | SMS send success: %{p1}
| mobile    |   9 | SMSSendError       | warning  | SMS send error: %{p1}
| mobile    |  10 | SMSSent            | notice   | Sent SMS
to %{p1}: %{p2}
| ethernet  |   1 | LinkUp             | notice   | Ethernet %{p1} up
| ethernet  |   2 | LinkDown           | notice   | Ethernet %{p1} down
| auth      |   2 | BadPasswordSSH     | warning  | SSH login attempt
from %{p2}: ba..
| auth      |   3 | BadUserConsole     | warning  | Console login attempt
on %{p1}: ..
| auth      |   4 | BadPasswordConsole | warning  | Console login attempt
on %{p2}: ..
| auth      |   5 | BadUserTelnet      | warning  | Telnet login attempt:
bad username
| auth      |   6 | BadPasswordTelnet  | warning  | Telnet login attempt:
bad passwo..
| auth      |   7 | BadUserLuCI        | warning  | LuCI login attempt: bad
username..
| auth      |   8 | BadPasswordLuCI    | warning  | LuCI login attempt: bad
password..
| auth      |   9 | LoginSSH           | notice   | SSH login: user %{p2}
from %{p3}
| auth      |  10 | LogoffSSH          | notice   | SSH logoff: user %{p1}
due to "%..
| auth      |  11 | LoginConsole       | notice   | Console login:
user %{p1} on %{p2}
| auth      |  12 | LogoffConsole      | notice   | Console logoff on %{p1}
| auth      |  13 | LoginTelnet        | notice   | Telnet login:
user %{p1}
| auth      |  14 | LoginLuCI          | notice   | LuCI login: user %{p1}
| auth      |  15 | ConsoleCommand     | informat | %{p1}@%{p2} %{p3}
| auth      |  16 | LuCIAction         | informat
| %{p1}@%{p2} %{p3} %{p4} %{p5}
```

```
| ipsec     |   6 | IPSecInitIKE       | informat | IPSec IKE %{p1}
established
| ipsec     |   7 | IPSecInitSA        | informat | IPSec SA %{p1}
established
| ipsec     |   8 | IPSecCloseIKE      | informat | IPSec IKE %{p1} deleted
| ipsec     |   9 | IPSecCloseSA       | informat | IPSec SA %{p1} closed
| ipsec     |  10 | IPSecDPDTimeOut    | informat | IPSec IKE %{p1} DPD
timed out
| wifi      |   1 | WiFiConnectedToAP  | notice   | WiFi %{p1} connected to
AP %{p2}
| wifi      |   1 | WiFiConnectedToAP  | notice   | WiFi %{p1} connected to
AP %{p2}
| wifi      |   2 | WiFiDisconnectedFromAP   | notice   | WiFi %{p1}
disconnected from AP
| wifi      |   2 | WiFiDisconnectedFromAP   | notice   | WiFi %{p1}
disconnected from AP
| wifi      |   3 | WiFiStationAttached      | notice   | WiFi
station %{p2} connected to ..
| wifi      |   3 | WiFiStationAttached      | notice   | WiFi
station %{p2} connected to ..
| wifi      |   4 | WiFiStationDetached      | notice   | WiFi
station %{p2} disconnected ..
| wifi      |   4 | WiFiStationDetached      | notice   | WiFi
station %{p2} disconnected ..
| wifi      |   5 | WiFiStationAttachFailed  | notice   | WiFi
station %{p2} failed to con..
| wifi      |   5 | WiFiStationAttachFailed  | notice   | WiFi
station %{p2} failed to con..
| ppp       |   1 | LinkUp                   | informat | PPP for
interface %{p2} (protoco..
| ppp       |   2 | LinkDown                 | informat | PPP for
interface %{p2} (protoco..
| ppp       |   3 | ConnEstablished          | informat | PPP connection
for interface %{p..
| adsl      |   1 | LinkUp                   | notice   | ADSL trained.
Starting interface..
| adsl      |   2 | LinkDown                 | notice   | ADSL down.
Stopping interface %{..
| adsl      |   3 | Silent                   | debug    | ADSL silent
| adsl      |   4 | Training                 | debug    | ADSL training
| adsl      |   5 | TrainingSuccess          | notice   | ADSL training
successful: data ..
| system    |   1 | BootSuccess     | informat | Success booting into %{p1}
```

```
| system    |   2 | DigitalInputChange       | notice   | Digital 
Input %{p1} changed valu..
| ntp       |   1 | InitialSync              | notice   | Initial NTP sync: 
time: %{p1}; o..
| ntp       |   2 | Adjust           | informat | NTP adjust by %{p1}
| ntp       |   3 | QueryTimeout     | warning  | NTP query to %{p1} timed 
out. Ne..
| ntp       |   4 | QueryFailed      | warning  | NTP query failed: %{p1}
```

38 Configuring data usage monitor

38.1 Introduction

Virtual Access software provides support for monitoring of data usage on mobile interfaces and to disable if the monthly limit is exceeded. This allows an element of control over data usage for SIMs with a limited data plan.

DISCLAIMER: data usage statistics calculated by Virtual Access data usage feature are best estimates and may vary from the mobile carrier statistics that are used for billing. Virtual Access cannot be held liable for any fees charged by the carrier to the customer for their data usage. We recommend that the configured data usage is lower than the allowance and that traffic percentage alerts are used.

38.2 Configuration package used

Package	Sections
procrustes	limit

38.3 Configuring data usage using the web interface

Select **Services -> Data Usage**. The Data Usage page appears.

You can monitor interfaces as a collective group, so enter a name for the group and select **Add**. The examples below show a group name configured as 'wan'.

You can configure multiple groups.

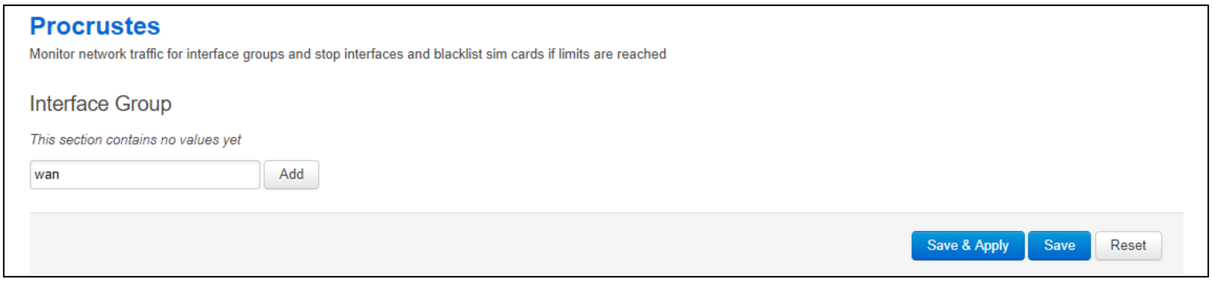


Figure 188: The data usage page

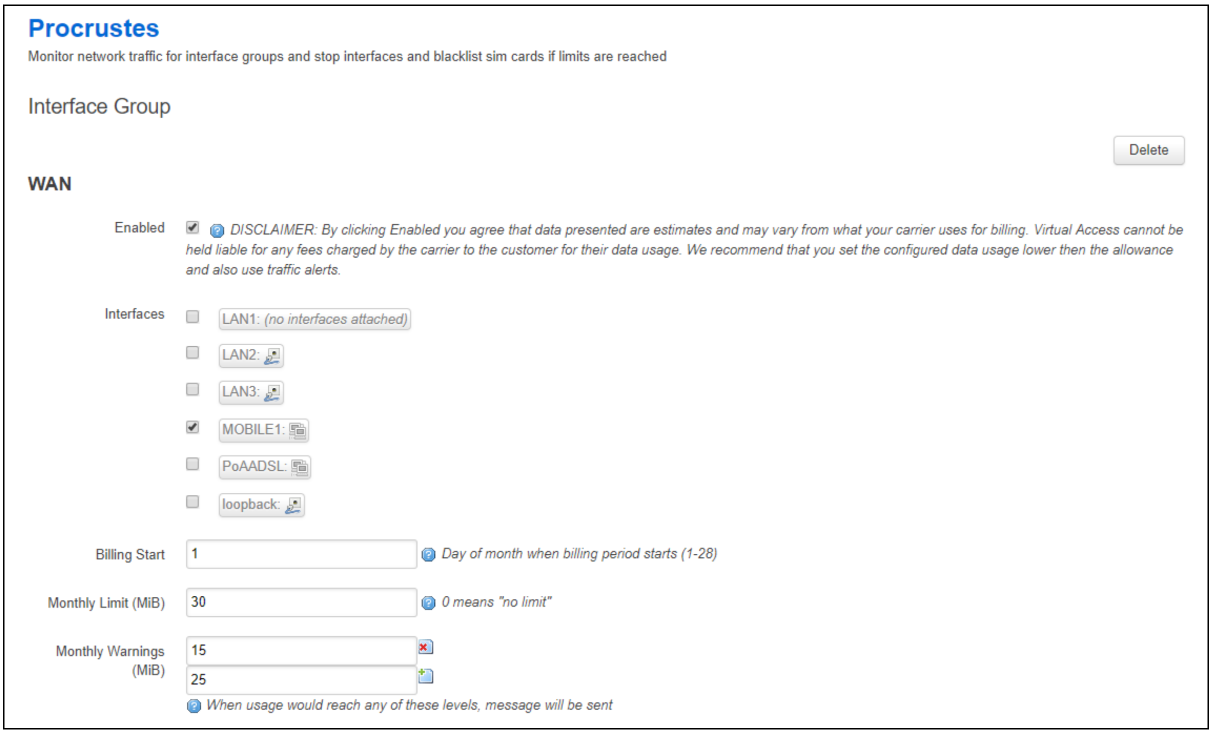


Figure 189: The data usage configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI: procrustes.@limit[0].enabled
Opt: enabled</td><td>Enable Data Usage monitor on this interface group.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Billing Start
UCI: procrustes.@limit[0].billing_period_start_day
Opt: billing_period_start_day</td><td>Day of month on which the billing period starts.<table><tr><td>1</td><td></td></tr><tr><td>Range</td><td>1 – 28</td></tr></table></td></tr>
<tr><td>Web: Interfaces
UCI: procrustes.@limit[0].interfaces
Opt: interfaces</td><td>Monitor and apply limits to these interfaces as a group. Configure multiple interfaces via UCI using a space separator. Example:
<code>uci set procrustes.@limit[0].interfaces="lan wan"</code></td></tr>
<tr><td>Web: Monthly Limit (MiB)
UCI: procrustes..@limit[0].monthly_data_limit
Opt: monthly_data_limit</td><td>Defines monthly data traffic limit in mebibytes (MiB). This is total RX and TX on the interface.<table><tr><td>0</td><td>Zero means no limit.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Monthly Warnings (MiB)
UCI: procrustes.@limit[0].monthly_warning_levels
Opt: monthly_warning_levels</td><td>Defines data usage limits for generating a log message and a VA event alert when used traffic reaches specified levels. Levels are specified in MiB.
Set multiple limits via UCI using a space separator. Example:
<code>uci set procrustes.@limit[0].monthly_warning_levels="15 25"</code><table><tr><td>0</td><td>Zero means no limit.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Table 134: Information table for data usage commands

38.3.1 Configuring data usage using command line

Data Usage is configured under the **procrustes** package **/etc/config/procrustes**.

By default, all limit instances are named 'limit', it is identified by `@limit` followed by the limit position in the package as a number. For example, for the first limit in the package using UCI:

```
procrustes.@limit[0]=limit
procrustes.@limit[0].enabled=1
```

Or using package options:

```
config limit
      option enabled '1'
```

However, to better identify, it is recommended to give the limit instance a name. For example, create a limit instance named MOBILE1.

To define a named limit instance using UCI, enter:

```
procrustes.@limit[0]=wan
procrustes.wan.enabled=1
```

To define a named limit instance using package options, enter:

```
config limit 'wan'
        option enabled '1'
```

The following examples show two limit groups wan and lan.

38.3.2 Procrustes using UCI

```
root@VA_router:~# uci show Procrustes
procrustes.lan=limit
procrustes.lan.enabled=1
procrustes.lan.interfaces=LAN1
procrustes.lan.billing_period_start_day=1
procrustes.lan.monthly_data_limit=30
procrustes.lan.monthly_warning_levels=15 25
procrustes.wan=limit
procrustes.wan.enabled=1
procrustes.wan.interfaces=MOBILE1
procrustes.wan.billing_period_start_day=1
procrustes.wan.monthly_data_limit=30
procrustes.wan.monthly_warning_levels=15 25
```

38.3.3 Procrustes using package options

```
root@VA_router:~# uci export procrustes
package procrustes

config limit 'lan'
        option enabled '1'
        option interfaces 'LAN1'
        option billing_period_start_day '1'
        option monthly_data_limit '30'
        option monthly_warning_levels '15 25'

config limit 'wan'
        option enabled '1'
        option interfaces 'MOBILE1'
        option billing_period_start_day '1'
        option monthly_data_limit '30'
        option monthly_warning_levels '15 25'
```

38.4 Data usage status

Select **System -> Overview**. The Status page appears.

To check current data usage, scroll to **Network -> Data Usage (MiB)** row.

Data usage is presented as progress bar.

Figure 190: The data usage status progress bar

38.5 Data usage diagnostics

38.5.1 Syslog events

The following events can be generated in logs by the data usage feature:

Severity	Tag	Text
NOTICE	procrustes	<if_group_name>: using counter 1404674 saved on 2017-09-30 16:26:57
NOTICE	procrustes	<if_group_name>: warning level 2097152 is reached
WARNING	procrustes	<if_group_name>: hard limit 10485760 is reached

NOTICE	procrustes	Data limit on SIM <iccid> exceeded and sim will be banned until the next month
ERROR	procrustes	Could not get iccid for <ifname>
DEBUG	procrustes	Interface <ifname> is not up
WARNING	procrustes	network.<ifname>.ifname is not defined
NOTICE	procrustes	<ifname>: reached billing start. Resetting...
DEBUG	procrustes	Saving current limit values
NOTICE	procrustes	<if_group_name>: not enabled
WARNING	procrustes	<if_group_name>: defines no interfaces
DEBUG	procrustes	<if_group_name>: sim interface <ifname>
ERROR	procrustes	Daemonization failed
ERROR	procrustes	another procrustes is running. Exiting...
NOTICE	procrustes	No limits defined. Exiting...
ERROR	mobile	SIM <iccid> is blacklisted, not establishing connection

38.5.2 Viewing data usage

The router has monitoring application named **procrustatus.lua** that can be used for viewing data usage.

This application displays data statistics used for different interface groups, percentage of time left to next billing period start and percentage of data left for use before the interface will be shut down.

To view the application, enter the command `procrustes.lua`

```
root@VA_router:~# procrustatus.lua
   name    current/       max  time left  data left
   lan:    1404674/  10485760     1.03%     86.60%
   wan:     433436/   1048576     1.03%     58.66%
```

Alternatively, total data usage can be checked via the command `cat /var/state/procrustes`

```
root@VA_router:~# cat /var/state/procrustes
procrustes.lan.total_bytes=215780
procrustes.wan.total_bytes=433436
```

38.5.3 Addditional debugging commands

Additional useful debug commands via the command line are described in the table below.

Diagnostic Command	Description
logread | grep procrustes	Shows logs related to “procrustes” only
ls /root/procrustes/sim_blacklist/	Shows list of blacklisted SIM iccids

39 Configuring Terminal Server

39.1 Overview

Terminal Server is a background application whose main task is to forward data between TCP connections or UDP streams and asynchronous or synchronous serial ports.

The Terminal Server application serves up to four sessions simultaneously, one for each serial port, depending on the device. Each Terminal Server session has an IP endpoint and an associated specific serial port.

You can configure the IP endpoint of each Terminal Server session to be a:

- TCP server: each session is listening on a unique port.
- TCP client: Terminal Server makes a TCP connection to external TCP server.
- UDP endpoint: Terminal Server forwards data between a UDP stream and a serial port.

39.2 Configuration packages used

Package	Sections
tservd	main
	port

39.3 Configuring Terminal Server using the web interface

In the top menu, select **Services -> Terminal Server**. The Terminal Server Configuration page appears. You must configure two main sections:

Main Settings to enable Terminal Server, syslog settings, and to enable log setting.

Port Settings section is for general port settings, serial settings such as port mode, port speed, parity stip bit and so on; and finally, network settings to configure the network side of the Terminal Server.

39.3.1 Configure main settings

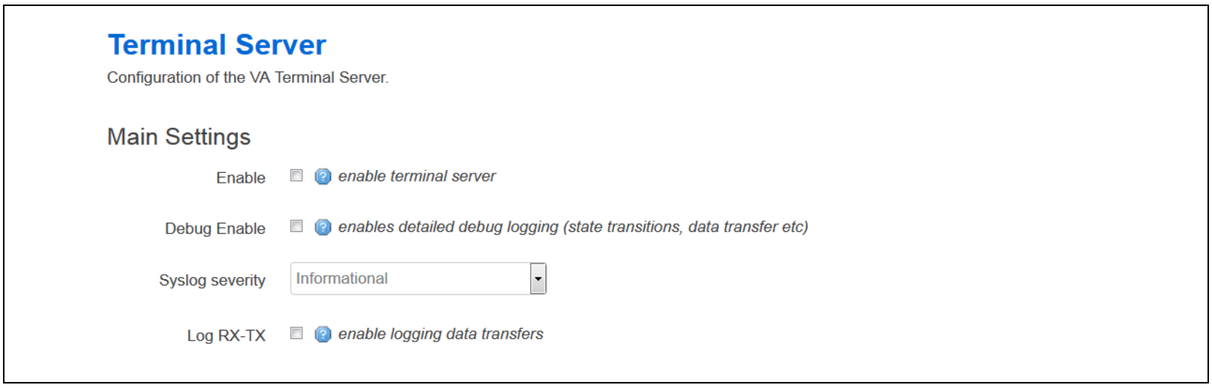


Figure 191: The terminal server main settings page

Web Field / UCI / Package Option	Description
Web: Enable UCI: tservd.main.enable Opt: enable	Enables Terminal Server on the router. 0 Disabled. 1 Enabled.
Web: Debug Enable UCI: tservd.main.debug_ev_enable Opt: debug_ev_enable	Enables detailed debug logging. 0 Disabled. 1 Enabled.
Web: Syslog severity UCI: tservd.main.log_severity Opt: log_severity	Determines the syslog level. Events up to this priority will be logged. 0 Emergency 1 Alert 2 Critical 3 Error 4 Warning 5 Notice 6 Informational 7 Debug
Web: Log RX-TX UCI: tservd.main.debug_rx_tx_enable Opt: debug_rx_tx_enable	Enables logging data transfers. 0 Disabled. 1 Enabled.

Table 135: Information table for main settings

39.3.2 Configure port settings

The Port Settings section is divided into 3 sub-sections:

- General
- Serial
- Network

39.3.2.1 Port settings: general section

In this section you can configure general port settings. The settings are usually the same for the central and the remote site.

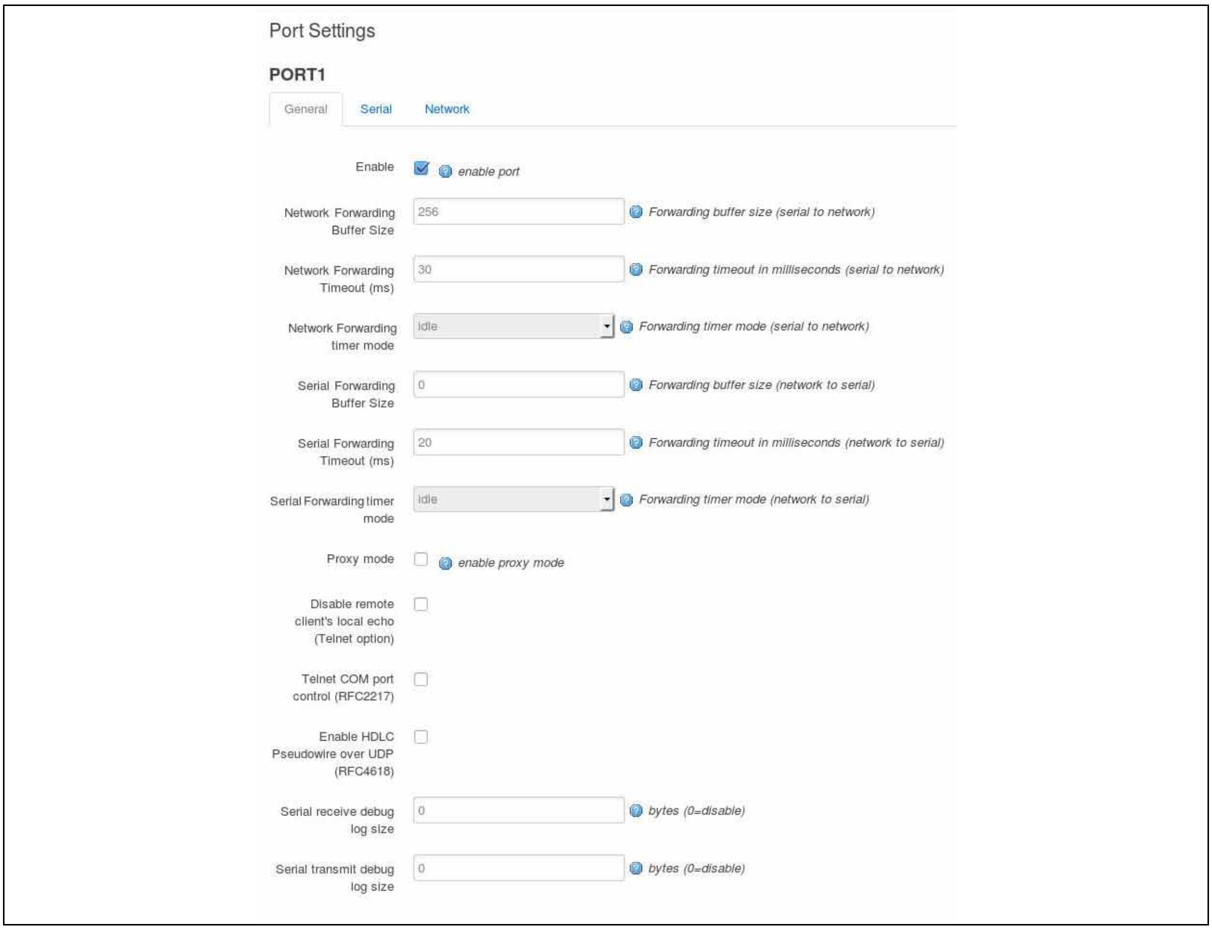


Figure 192: The general tab fields

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enable
UCI: tservd.@port[0].enable
Opt: enable</td><td>Enables Terminal Server port.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Network Forwarding Buffer Size
UCI: tservd.@port[0].fwd_buffer_size
Opt: fwd_buffer_size</td><td>Forwarding buffer size in bytes (serial to network).<table><tr><td>256</td><td>256 bytes</td></tr><tr><td>Range</td><td>0-2048</td></tr></table></td></tr>
<tr><td>Web: Network Forwarding Timeout(ms)
UCI: tservd.@port[0].fwd_timeout
Opt: fwd_timeout</td><td>Forwarding timeout in milliseconds (serial to network).<table><tr><td>30</td><td>30 ms</td></tr><tr><td>Range</td><td>0-10000</td></tr></table></td></tr>
<tr><td>Web: Network Forwarding Timer Mode
UCI: tservd.@port[0].fwd_timer_mode
Opt: fwd_timer_mode</td><td>Forwarding timer mode (serial to network).<table><tr><td>Idle</td><td>Timer is re-started on each received data.</td></tr><tr><td>Aging</td><td>Timer started on the first Rx.</td></tr></table></td></tr>
<tr><td>Web: Serial Forwarding Buffer Size
UCI: tservd.@port[0].sfwd_buffer_size
Opt: sfwd_buffer_size</td><td>Forwarding buffer size in bytes (network to serial).
Set to 0 to use maximum possible network Rx buffer size.<table><tr><td>0</td><td>2048 bytes</td></tr><tr><td>Range</td><td>0-2048</td></tr></table></td></tr>
<tr><td>Web: Serial Forwarding Timeout (ms)
UCI: tservd.@port[0].sfwd_timeout
Opt: sfwd_timeout</td><td>Forwarding timeout in milliseconds (network to serial).
Set to 0 to forward to serial immediately.<table><tr><td>20</td><td>20 ms</td></tr><tr><td>Range</td><td>0-10000</td></tr></table></td></tr>
</table>

Web: Serial Forwarding Timer Mode UCI: tservd.@port[0].sfwd_timer_mode Opt: sfwd_timer_mode	Forwarding timer mode (network to serial). Idle: Timer is re-started on each received data Aging: Timer started on the first Rx.
Web: Proxy Mode UCI: tservd.@port[0].proxy_mode Opt: proxy_mode	Defines if special proxy mode is configured to allow 'hijacking' of the terminal server. It allows a connection to be made from a remote location and redirect terminal server data temporarily for troubleshooting. When enabled, a TCP proxy server is started which listens for an incoming TCP connection from a remote peer. Once an incoming new TCP connection on the proxy server TCP port is accepted: The existing terminal server TCP client connection is disconnected. The terminal server automatically reconnects the TCP client side but this time to the local loopback address 127.0.0.1 and to the local proxies TCP port number. Once the proxy server has both local and remote TCP sessions connected it simply forwards the data between the two connections, taking into account the flow control. When either side TCP socket closes, the main terminal server client re-connects to the normal IP destination and the server proxy returns to listening for another connection from the far end. 0: Disabled. 1: Enabled.
Web: Disable Remote Client's Local Echo (Telnet option) UCI: tservd.@port[0].disable_echo Opt: disable_echo	Set to **1** to send IAC WILL ECHO Telnet option to remote client forcing it to disable local echo. For server mode only. 0: Disabled. 1: Enabled.
Web: Telnet COM Port Control UCI: tservd.@port[0].com_port_control Opt: com_port_control	Set to **1** to enable support for Telnet COM port control (RFC2217). 0: Disabled. 1: Enabled.
Web: Enable HDLC Pseudowire over UDP (RFC4618) UCI: tservd.@port[0].hdlc_pw_enabled Opt: hdlc_pw_enabled	Set to **1** to enable HDLC pseudowire over UDP support based on RFC4618. Requires Transport Mode (udpmode) to be enabled. 0: Disabled. 1: Enabled.
Web: Serial Receive Debug Log Size UCI: tservd.@port[0].serialRxLogSize Opt: serialRxLogSize	Configures serial receive log size in bytes and enables receive data logging. 0: Disabled. 1: Enabled.
Web: Serial Transmit Debug Log Size UCI: tservd.@port[0].serialTxLogSize Opt: serialTxLogSize	Configures serial transmit log size in bytes and enables transmit data logging. 0: Disabled. 1: Enabled.

Table 136: Information table for port settings section

39.3.2.2 Port settings: serial section

In this section you can configure serial interface settings, such as port mode, port speed, parity stip bit and so on.

Note:

- The displayed settings vary depending on options selected.
- DTR <--> DSR signalling is not available on GW2028 router models.

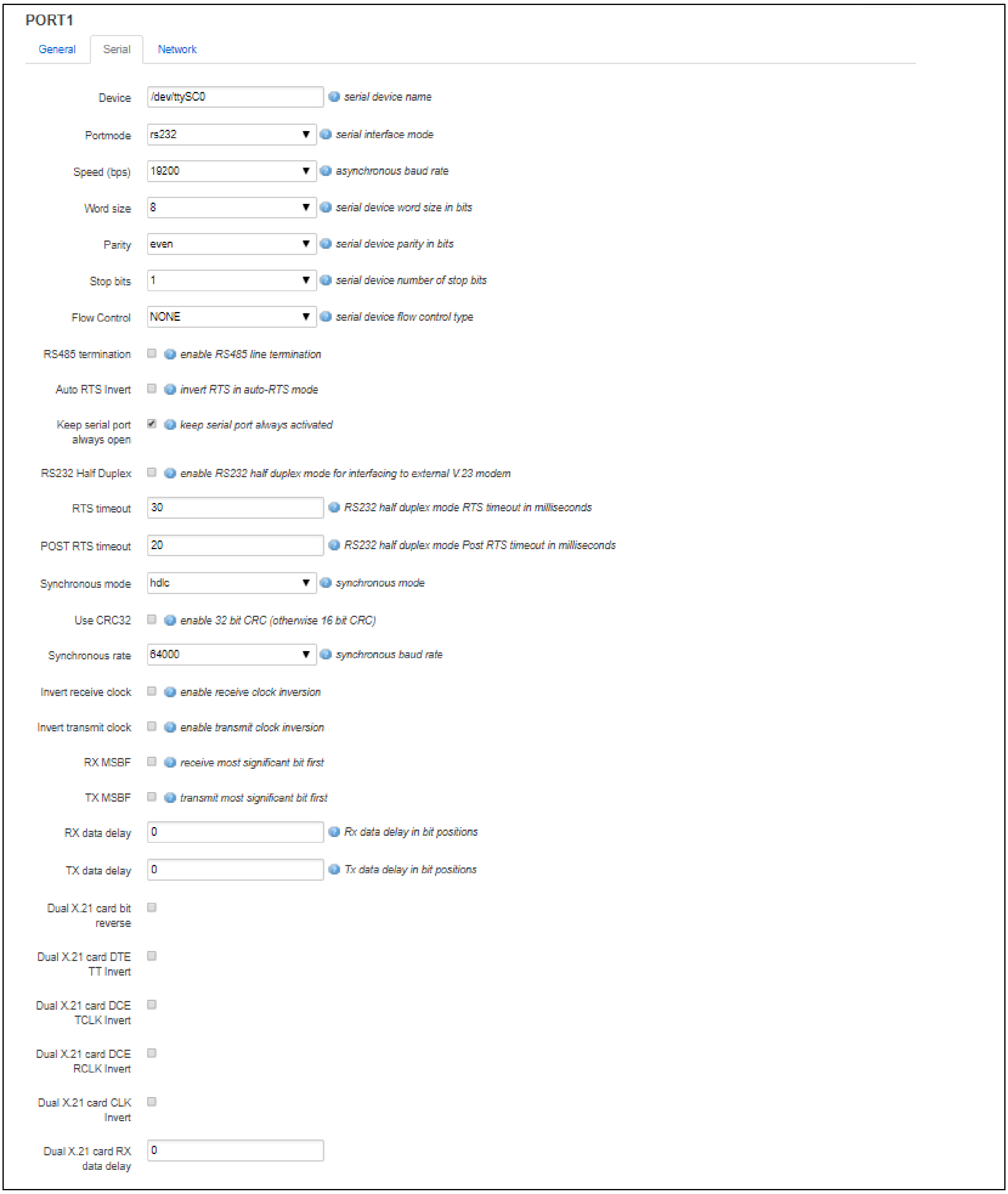


Figure 193: The serial section fields (port mode RS232)

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Device
UCI: tservd.@port[0].devName
Opt: devName</td><td>Serial device name.<table><tr><td>/dev/ttySC0</td><td>serial port 1</td></tr><tr><td>/dev/ttySC1</td><td>serial port 2</td></tr><tr><td>/dev/ttySC2</td><td>serial port 3</td></tr><tr><td>/dev/ttySC3</td><td>serial port 4</td></tr></table></td></tr>
<tr><td>Web: Port mode
UCI: tservd.@port[0].port_mode
Opt: port_mode</td><td>Sets the serial interface mode.<table><tr><td>rs232</td><td>RS232 mode.</td></tr><tr><td>rs485hdx</td><td>RS485 2 wire half duplex mode in which transmitter drives RTS.</td></tr><tr><td>rs485fdx</td><td>Rs485 4 wire full duplex mode.</td></tr><tr><td>v23</td><td>Uses V.23 leased line card driver.</td></tr><tr><td>x21</td><td>Uses USB serial card in sync mode.</td></tr></table></td></tr>
<tr><td>Web: Speed (bps)
UCI: tservd.@port[0].speed
Opt: speed</td><td>Serial device speed in baud (bps).<table><tr><td>9600</td><td></td></tr><tr><td>Range</td><td>115200; 57600; 38400; 19200; 9600 4800; 2400; 1800; 1200; 600; 300; 200; 150; 134; 110; 75; 50</td></tr></table></td></tr>
<tr><td>Web: Word size
UCI: tservd.@port[0].wsize
Opt: wsize</td><td>Serial device word size.<table><tr><td>8</td><td></td></tr><tr><td>Range</td><td>5-8</td></tr></table></td></tr>
<tr><td>Web: Parity
UCI: tservd.@port[0].parity
Opt: parity</td><td>Serial device parity.<table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Even</td></tr><tr><td>2</td><td>Odd</td></tr><tr><td>3</td><td>Space</td></tr></table></td></tr>
<tr><td>Web: Stop Bits
UCI: tservd.@port[0].stops
Opt: stops</td><td>Serial device number of stop bits.<table><tr><td>1</td><td></td></tr><tr><td>Range</td><td>1-2</td></tr></table></td></tr>
<tr><td>Web: Flow Control
UCI: tservd.@port[0].fc_mode
Opt: fc_mode</td><td>Serial flow control mode.<table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>RTS/CTS</td></tr><tr><td>2</td><td>XON/XOFF</td></tr></table></td></tr>
<tr><td>Web: RS485 Termination
UCI: tservd.@port[0].rs485_line_termination
Opt: rs485_line_termination</td><td>Enables or disable RS485 termination. Applies only if port mode is set to RS485.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Auto RTS Invert
UCI: tservd.@port[0].rtsinvert
Opt: rtsinvert</td><td>Invert RTS in auto-RTS mode, if port mode is set to RS485.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Keep Serial Port Always Open
UCI: tservd.@port[0].tty_always_open
Opt: tty_always_open</td><td>Keep serial port always open.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: RS232 Half Duplex
UCI: tservd.@port[0].hd_mode
Opt: hd_mode</td><td>Defines whether to enable special mode in the asynchronous serial driver for communication to an externally connected V.23 half-duplex modem. Note: this setting does not enable half-duplex mode in the serial hardware of the router.<table><tr><td>0</td><td>Full duplex mode.</td></tr><tr><td>1</td><td>Half duplex mode.</td></tr></table></td></tr>
<tr><td>Web: RTS Timeout
UCI: tservd.@port[0].rts_timeout
Opt: rts_timeout</td><td>In RS232 half-duplex mode, time in milliseconds between raising RTS and enabling the transmitter. For use with an externally connected V.23 modem.<table><tr><td>30</td><td>30ms</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

<table>
<tr><td>Web: POST RTS Timeout
UCI: tservd.@port[0].post_rts_timeout
Opt: post_rts_timeout</td><td>In RS232 half duplex mode, time in milliseconds between dropping RTS (transmission finished) and enabling the receiver. For use with externally connected V.23 modem.<table><tr><td>20</td><td>20 ms</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Synchronous mode
UCI: tservd.@port[0].sync mode
Opt: sync mode</td><td>Defines synchronous frame mode. This setting is only displayed if an Atmel USB serial card is enabled.<table><tr><td>hdlc</td><td>HDLC frame mode.</td></tr><tr><td>transp</td><td>Transparent mode.</td></tr></table></td></tr>
<tr><td>Web: Use CRC32
UCI: tservd.@port[0].sync_crc32
Opt: sync_crc32</td><td>Defines whether to use CRC32 or CRC16 in HDLC mode. This setting is only displayed if an Atmel USB serial card is enabled.<table><tr><td>0</td><td>Use CRC16.</td></tr><tr><td>1</td><td>Use CRC32.</td></tr></table></td></tr>
<tr><td>Web: DTR control mode
UCI: tservd.@port[0].dtr_control_mode
Opt: dtr_control_mode</td><td>Defines DTR line control modes. This setting is only displayed if an Atmel USB serial card is enabled and port mode is X21.<table><tr><td>auto</td><td>DTR set to on when port is open. Off when the port is closed.</td></tr><tr><td>on</td><td>DTR always on.</td></tr><tr><td>off</td><td>DTR always off.</td></tr><tr><td>app</td><td>DTR controlled by the application.</td></tr><tr><td>ontx</td><td>In HDLC mode DTR is on during frame transmission.</td></tr></table></td></tr>
<tr><td>Web: RTS control mode
UCI: tservd.@port[0].rts_control_mode
Opt: rts_control_mode</td><td>Defines RTS line control modes. Only displayed if Atmel USB serial card is enabled and port mode is X21.<table><tr><td>auto</td><td>RTS set to on when port is open. Off when the port is closed.</td></tr><tr><td>on</td><td>RTS always on.</td></tr><tr><td>off</td><td>RTS always off.</td></tr><tr><td>app</td><td>RTS controlled by the application.</td></tr><tr><td>ontx</td><td>In HDLC mode RTS is on during frame transmission.</td></tr></table></td></tr>
<tr><td>Web: Synchronous rate
UCI: tservd.@port[0].sync_speed
Opt: sync_speed</td><td>Defines the synchronous speed in bps. Set to 0 for external clock. If not set to 0, an internal clock is used. This setting is only displayed if an Atmel USB serial card is enabled.<table><tr><td>64000</td><td>64 kbps</td></tr><tr><td>Range</td><td>2048000; 1024000; 768000; 512000; 384000; 256000; 128000; 19200; 9600</td></tr></table></td></tr>
<tr><td>Web: Invert receive clock
UCI: tservd.@port[0].sync_invert_rxclk
Opt: sync_invert_rxclk</td><td>Defines receive clock inversion. Normal clock data is sampled on falling edge. Inverted clock data is sampled on rising edge. This setting is only displayed if an Atmel USB serial card is enabled.<table><tr><td>0</td><td>Normal.</td></tr><tr><td>1</td><td>Invert.</td></tr></table></td></tr>
<tr><td>Web: Invert transmit clock
UCI: tservd.@port[0].sync_invert_txclk
Opt: sync_invert_txclk</td><td>Defines transmit clock inversion. Normal clock data transmitted on falling edge. Inverted clock data transmitted on rising edge. Only displayed if Atmel USB serial card is enabled.<table><tr><td>0</td><td>Normal.</td></tr><tr><td>1</td><td>Invert.</td></tr></table></td></tr>
<tr><td>Web: RX MSBF
UCI: tservd.@port[0].sync_rx_msbf
Opt: sync_rx_msbf</td><td>Defines whether most significant bit is received first. This setting is only displayed if an Atmel USB serial card is enabled.<table><tr><td>0</td><td>Receive least significant bit first.</td></tr><tr><td>1</td><td>Receive most significant bit first.</td></tr></table></td></tr>
<tr><td>Web: TX MSBF
UCI: tservd.@port[0].sync_tx_msbf
Opt: sync_tx_msbf</td><td>Defines whether most significant bit is transmitted first. This setting is only displayed if an Atmel USB serial card is enabled.<table><tr><td>0</td><td>Transmit least significant bit first.</td></tr><tr><td>1</td><td>Transmit most significant bit first.</td></tr></table></td></tr>
</table>

Web: RX data delay UCI: tservd.@port[0].sync_rxdata_dly Opt: sync_rxdata_dly	Defines the number of bit positions to delay sampling data from the detecting clock edge. This setting is only displayed if an Atmel USB serial card is enabled.	0	
		Range	
Web: TX data delay UCI: tservd.@port[0].sync_txdata_dly Opt: sync_txdata_dly	Defines the number of bit positions to delay output of data from the detecting clock edge. This setting is only displayed if an Atmel USB serial card is enabled.	0	
		Range	
Web: Dual X.21 card bit reverse UCI: tservd.@port[0].bit_reverse Opt: bit_reverse	Enables bit reversal of all bits in 8 byte word during transmission.	0	Normal.
		1	Reverse.
Web: Dual X.21 card DTE TT Invert UCI: tservd.@port[0].dte_tt_inv Opt: dte_tt_inv	Enables X.21 TT clock signal inversion.	0	Normal.
		1	Invert.
Web: Dual X.21 card DCE TCLK Invert UCI: tservd.@port[0].dce_tclk_inv Opt: dce_tclk_inv	Enables X.21 DCE TCLK signal inversion.	0	Normal.
		1	Invert.
Web: Dual X.21 card DCE RCLK Invert UCI: tservd.@port[0].dce_rclk_inv Opt: dce_rclk_inv	Enables X.21 DCE RCLK signal inversion.	0	Normal.
		1	Invert.
Web: Dual X.21 card CLK Invert UCI: tservd.@port[0].x21_clk_invert Opt: x21_clk_invert	Enables X.21 DCE CLK signal inversion.	0	Normal.
		1	Invert.
Web: Dual X.21 card RX data delay UCI: tservd.@port[0] x21_data_delay Opt: x21_data_delay	Sets X.21 card RX data delay in number of bit positions.	0	
		Range	0 – 7
Web: n/a UCI: tservd.@port[0].sync_tx_idle Opt: sync_tx_idle	Defines the value of idle character (decimal) to transmit in case of tranmit underrun. In HDLC mode, this configures inter-frame fill.	0	Tranmit 0 (in HDLC mode)
		126	Transmit flags (in HDLC mode)
		255	Tranmit 1 (in HDLC mode)
		Range	0 – 255
Web: n/a UCI: tservd.@port[0].v23_inband_carrier_signalling Opt: v23_inband_carrier_signalling	Enables signalling of carrier by sending special characters.	0	Disabled.
		1	Enabled.
Web: n/a UCI: tservd.@port[0].v23_inband_carrier_on_char Opt: v23_inband_carrier_on_char	Defines the character decimal to signal remote carrier on.	255	
		Range	0 - 255
Web: n/a UCI: tservd.@port[0].v23_tx_gain Opt: v23_tx_gain	Defines the transmit gain for v23 mode.	2	Transmit samples multiplied by 2
		Range	
Web: n/a UCI: tservd.@port[0].v23_rx_loss Opt: v23_rx_loss	Defines the receive loss for v23 mode.	1	Receive samples divided by 1.
		Range	
Web: n/a UCI: tservd.@port[0].v23_rts_to_cts_delay Opt: v23_rts_to_cts_delay	Defines the v23 modem RTS to CTS delay in milliseconds.	20	
		Range	

Web: n/a UCI: tservd.@port[0].v23_is_four_wire Opt: v23_is_four_wire	Defines the V23 modem LIM operation. 0 — 2-wire 1 — 4-wire
Web: n/a UCI: tservd.@port[0].v23_tx_timeout Opt: v23_tx_timeout	Defines the V23 modem receive echo suppression timeout in milliseconds. 20 Range
Web: n/a UCI: tservd.@port[0].v23_tx_rampdown Opt: v23_tx_rampdown	Defines the time in milliseconds it takes the V23 transmitter to rampdown carrier from peak to zero. 30 Range
Web: n/a UCI: tservd.@port[0].v23_tx_maxfill Opt: v23_tx_maxfill	Defines the maximum transmit queue fill level in bytes. 127 Range — 0 - 255

Table 137: Information table for port settings serial section

39.3.2.3 Port settings: network section

In this section you can configure the network side of the Terminal Server.

Note: the displayed settings vary depending on options selected.

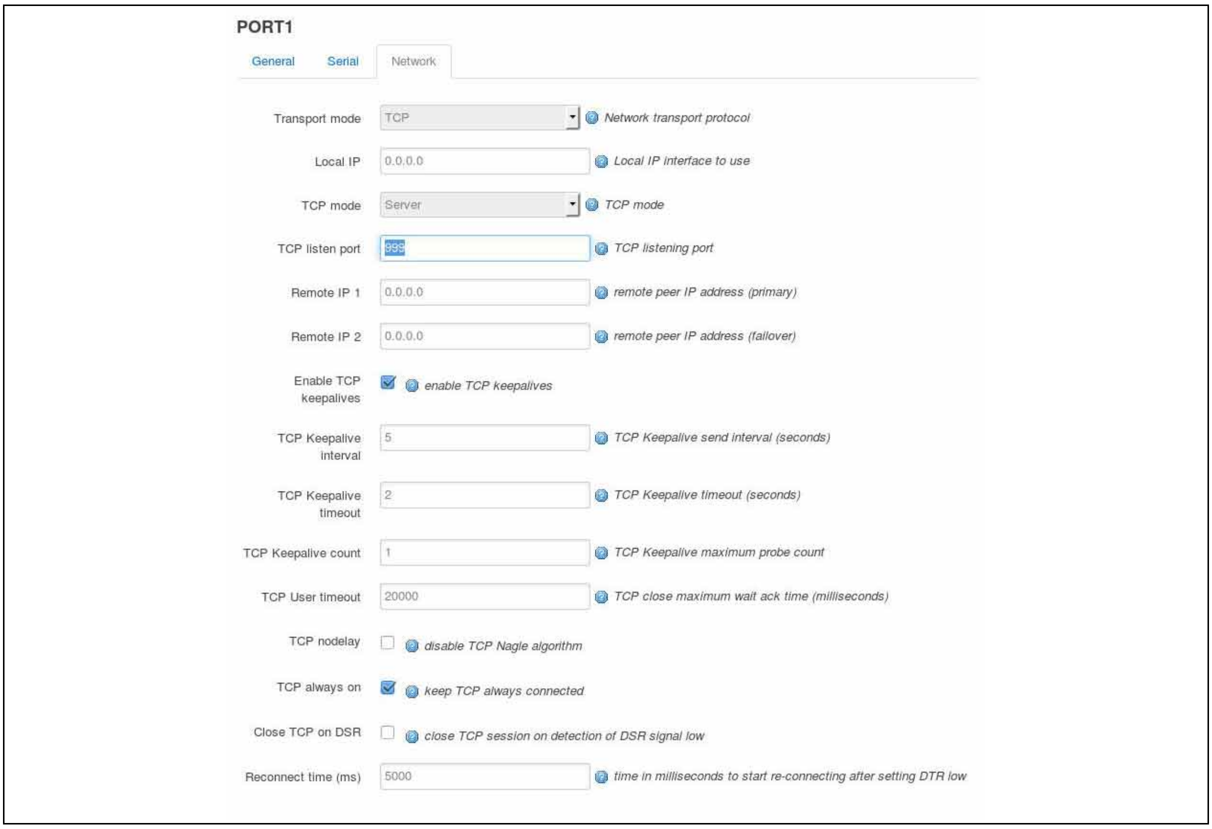


Figure 194: The port settings network fields (TCP server mode)

Web Field / UCI / Package Option	Description
Web: Transport Mode UCI: tservd.@port[0].udpMode Opt: udpMode	Selects the transport mode. 0 – TCP 1 – UDP
Web: Local IP UCI: tservd.@port[0].local_ip Opt: local_ip	Local IP address to listen on. 0.0.0.0 – Listen on any interface. Range – IPv4 address.
Web: TCP Mode UCI: tservd.@port[0].server_mode Opt: server_mode	Select between server and client modes of TCP. Only displayed if Transport Mode is TCP. 0 – Client Mode. 1 – Server Mode.
Web: TCP Listen Port UCI: tservd.@port[0].listen_port Opt: listen_port	TCP listen port for server mode. Only displayed if Transport Mode is TCP and server mode is enabled. 999 Range – 1 - 65535
Web: Remote TCP Port 1 UCI: tservd.@port[0].ip_port1 Opt: ip_port1	Destination peer port IP 1 number. Only displayed if client mode is enabled. 951 Range – 1 - 65535
Web: Remote TCP Port 2 UCI: tservd.@port[0].ip_port2 Opt: ip_port2	Destination peer port IP 2 number for failover. Only displayed if client mode is enabled. 951 Range – 1 - 65535
Web: Remote IP 1 UCI: tservd.@port[0].remote_ip1 Opt: remote_ip1	Destination peer IP 1 address. 0.0.0.0 Range – IPv4 address.
Web: Remote IP 2 UCI: tservd.@port[0].remote_ip2 Opt: remote_ip2	Destination peer IP 2 address for failover. 0.0.0.0 Range – IPv4 address.
Web: Enable TCP Keepalives UCI: tservd.@port[0].tcp_keepalives_enabled Opt: tcp_keepalives_enabled	Enable or disables TCP keepalives. Only displayed if Transport Mode is TCP. 0 – Disabled. 1 – Enabled.
Web: TCP Keepalive Interval UCI: tservd.@port[0].tcp_keepalive_interval Opt: tcp_keepalive_interval	Interval in seconds between TCP keepalive probes. Only displayed if Transport Mode is TCP. 5 – 5 seconds. Range – 0-65535
Web: TCP Keepalive Timeout UCI: tservd.@port[0].tcp_keepalive_timeout Opt: tcp_keepalive_timeout	Time in seconds to wait for response to a TCP keepalive probe. Only displayed if Transport Mode is TCP. 2 – 2 seconds. Range – 0-65535
Web: TCP Keepalive Count UCI: tservd.@port[0].tcp_keepalive_count Opt: tcp_keepalive_count	Number of TCP keepalive probes to send before connection is closed. Only displayed if Transport Mode is TCP. 1 Range – 0-65535
Web: TCP User Timeout UCI: tservd.@port[0].tcp_user_timeout Opt: tcp_user_timeout	Maximum time in milliseconds for TCP to wait for transmitted data to be ‘acked’ before closing connection in established state. Set to **0** to use kernel defaults. Only displayed if Transport Mode is TCP. 20000 – 20 seconds. Range – 0-65535

Web: TCP Nodelay UCI: tservd.@port[0].tcp_nodelay Opt: tcp_nodelay	Sets TCP to delay behaviour. Only displayed if Transport Mode is TCP. 0 — Normal operation. 1 — Disable TCP Nagle algorithm. Only displayed if Transport Mode is TCP.
Web: TCP Always on UCI: tservd.@port[0].tcp_always_on Opt: tcp_always_on	Keep TCP session always connected. Only displayed if Transport Mode is TCP and client mode is enabled. 0 — Disabled. TCP connection / UDP session is initiated on detecting high state on the DSR interface signal. 1 — Enabled. If it disconnects in the established state the TCP connection / UDP session is re-initiated.
Web: Close TCP on DSR UCI: tservd.@port[0].close_tcp_on_dsr Opt: close_tcp_on_dsr	Close TCP session on detection of DSR signal low. Only displayed if Transport Mode is TCP and client mode is enabled. 0 — Disabled. Detecting DSR down does not affect the TCP connection. 1 — Enabled. Detecting DSR down closes the established TCP connection.
Web: Reconnect Time (ms) UCI: tservd.@port[0].disc_time_ms Opt: disc_time_ms	Time in milliseconds to start reconnecting after setting DTR low. 5000 — 5 seconds. Range — 0 – 10000
Web: UDP Keepalive Interval UCI: tservd.@port[0].udpKaIntervalMs Opt: udpKaIntervalMs	Defines time in milliseconds to send UDP keepalives (empty UDP packets) when no data to send. Only displayed if transport mode is UDP. 0 — Disabled. Range — 0-65535
Web: UDP Keepalive Count UCI: tservd.@port[0].udpKaCount Opt: udpKaCount	Defines the maximum number of remote UDP keepalives not received before UDP stream is considered broken. Only displayed if transport mode is UDP. 3 Range — 0-65535
Web: local UDP Port UCI: tservd.@port[0].udpLocalPort Opt: udpLocalPort	Local UDP port used by terminal server. Only displayed if transport mode is UDP. 0 Range — 0-65535
Web: remote UDP Port UCI: tservd.@port[0].udpRemotePort Opt: udpRemotePort	Remote UDP port used by terminal server. Only displayed if transport mode is UDP. 0 Range — 0-65535

Table 138: Information table for port settings network section

39.4 Terminal Server using UCI

```
root@VA_router:~# uci show tservd
tservd.main=tservd
tservd.main.log_severity=0
tservd.main.debug_rx_tx_enable=1
tservd.main.debug_ev_enable=1
tservd.@port[0]=port
tservd.@port[0].devName=/dev/ttySC0
tservd.@port[0].remote_ip1=0.0.0.0
tservd.@port[0].remote_ip2=0.0.0.0
```

39.5 Terminal Server using package options

```
root@VA_router:~# uci export tservd
package tservd

config tservd 'main'
        option log_severity '0'
        option debug_rx_tx_enable '1'
        option debug_ev_enable '1'

config port
        option devName '/dev/ttySC0'
        option remote_ip1 '0.0.0.0'
        option remote_ip2 '0.0.0.0'
```

39.6 Terminal server DSR signal management based on network configuration

On the IP network side, the terminal server can operate in one of three modes:

- TCP Client
- TCP Server
- UDP

Based on the choosed network configuration the DSR behaviour may vary.

39.6.1 DSR signal behaviour in TCP client mode

39.6.1.1 TCP connection management

Initial TCP connection initiation or next TCP connection initiation after disconnection is affected by configuration options tcp_always_on and close_tcp_on_dsr.

When option tcp_always_on is enabled terminal server keeps the TCP session always connected. If it disconnects in the established state, the TCP session is reinitiated.

If tcp_always_on is disabled TCP connection is initiated on detection of a high state on the DSP interface signal.

When option close_tcp_on_dsr is enabled terminal server dectecting DSR down signal and closes the established TCP connection.

If option close_tcp_on_dsr is disabled then dectecting DSR down does not affect the TCP connection.

39.6.1.2 TCP connection initiation at startup

If you have set option tcp_always_on1, or DSR state is UP, the TCP connection setup is initiated immediately.

If you have set option tcp_always_on0, and DSR is DOWN, the terminal server waits for a DSR UP signal. When DSR UP is detected, the TCP connection is initiated.

39.6.1.3 TCP connection clearing

The TCP connection is cleared either by the network or by the terminal server application itself.

The TCP connection is cleared by the terminal server when it detects DSR interface signal DOWN and option close_tcp_on dsr is 1.

39.6.1.4 TCP connection reinitiation

After TCP conncetion clearing, the terminal server takes action to re-setup the TCP connection after a hand off timeout.

If you have set option tcp_always_on1, or DSR state is UP, the TCP connection setup is initiated.

If you have set option tcp_always_on0, and DSR is DOWN, the terminal server waits for a DSR UP signal and then initiates a new TCP connection.

39.6.2 DSR signal behaviour in TCP server mode

39.6.2.1 TCP connection initiation at startup

After a short startup delay, the terminal server starts listening for an incoming TCP connection from the remote peer.

39.6.2.2 TCP connection clearing

When in a TCP connection state, the TCP connection is cleared only by the network. Serial interface signals such as DSR do not cause TCP disconnection.

39.6.2.3 TCP connection re-initiation

When a TCP session goes down in the connected state, the terminal server immediately restarts listening for a new TCP connection from a remote peer.

39.6.3 DSR signal behaviour in UDP mode

39.6.3.1 UDP session setup at startup

If you have set option tcp_always_on1, or DSR state is UP, the UDP session is setup immediately on startup.

If you have set option tcp_always_on0, and DSR is DOWN, the terminal server waits for a DSR UP signal. When DSR UP is detected, the UDP session is setup.

39.6.3.2 UDP session clearing

A UDP session is normally never cleared, but if it is closed by the network sub-system, it gets re-setup after a hand off timeout.

A DSR signal DOWN event does not clear UDP session in the connected state.

39.6.3.3 UDP session reset

After UDP session clearing the terminal server takes action to resetup a UDP session after a hand off timeout.

If you have set option tcp_always_on1, or DSR state is UP, the UDP session is setup.

If you have set option tcp_always_on0, and DSR is DOWN, the terminal server waits for a DSR UP signal and then it resets up the UDP session.

39.7 Terminal Server diagnostics

The tservd process has to be running otherwise diagnostics options for terminal server will not be available.

39.7.1 Checking Terminal Server process

To check if Terminal Server is running, enter:

```
root@VA_router:~# ps | grep tservd
 1264 root      1032 S  tservd
 1769 root      1496 S  grep tservd
```

If Terminal Server is running it will be shown with its process ID.

39.7.2 Terminal Server statistics

To see Terminal Server statistics, enter:

```
root@VA_router:~# tserv show stats
TERMINAL 1, Dev: /dev/ttySC0
State:          LISTENING
```

```
Serial Bytes    Rx (0)  Tx (0)  TxErrs (0)
TCP Packets     Rx (0)  Tx (0)  TxErrs (0)      TxBlocked (0)
TCP Bytes       Rx (0)  Tx (0)
UDP Datagrams   Rx (0)  Tx (0)  TxErrs (0)
UDP Bytes       Rx (0)  Tx (0)
DSR             Up (0)  Down (0)
```

39.7.3 Terminal Server debug statistics

To see debug statistics about Terminal Server, enter:

```
root@VA_router:~# tserv show debug all


TERMINAL 1, Dev: /dev/ttySC0
State:          LISTENING
netRxBuf length=0 offset=0 hdrsz=0
ttyRxBuf length=0 offset=16 hdrsz=16
line_status_mask = 0x0 line_status = 0x0
RFC2217 negotiated=0
Tcp tx last error: 0
```

39.7.4 Terminal Server serial signals debugging

To see Terminal Server serial signals statistics, enter:

```
root@VA_router:~# tserv show serial


TERMINAL-1, Dev: /dev/ttySC1
DSR=0 DTR=1 RTS=1 CTS=0 CAR=0 CD=0 RNG=0 LE=0 RI=0 ST=0 SR=0



TERMINAL-2, Dev: /dev/ttySC0
DSR=0 DTR=1 RTS=1 CTS=0 CAR=0 CD=0 RNG=0 LE=0 RI=0 ST=0 SR=0
```

39.7.5 Terminal Server advanced debugging

To see Terminal Server advanced debug commands for the terminal server, enter:

```
root@VA_router:~# tserv
=== Termserv disgnostics. Command syntax: ===
tserv show stats - show statistics
tserv clear stats - clear statistics
```

```
tserv show serial - show serial interface status
tserv send serial0 <data>- send data to serial port 0
tserv start capture N, N=port number (0 to 3) - start capturing rx serial
data
tserv print capture N, N=port number (0 to 3) - print captured rx serial
data
tserv show serial txlog-hex <Port> [length], Port=port cfg index (0 to 3),
length=length to show
tserv show serial rxlog-hex <Port> [length], Port=port cfg index (0 to 3),
length=length to show
tserv show serial txlog-asc <Port> [length], Port=port cfg index (0 to 3),
length=length to show
tserv show serial rxlog-asc <Port> [length], Port=port cfg index (0 to 3),
length=length to show
tserv show debug - show debug info
tserv start userial rxlog - start USB serial card rx log
tserv show userial rxlog <offs> <length> - show USB serial card rx log
tserv quit - terminate termserv process
```

40 Configuring terminal package

Terminal package is used to automatically add entries for getty to inittab for extra incoming console/terminal connections.

40.1 Configuration packages used

Package	Sections
terminal	terminal

40.2 Configuring terminal package using the web interface

Terminal package is not available to configure using the web interface.

Web Field / UCI / Package Option	Description
Web: n/a UCI: terminal.console.enabled Opt: enabled	Enables Terminal on the router. 0 — Disabled. 1 — Enabled.
Web: n/a UCI: terminal.console.device Opt: device	String value point at the tty device in /dev folder. None — Default. <string> — Device name.(e.g. ttySC0 to use serial port 0)
Web: n/a UCI: terminal.console.speed Opt: speed	Set the speed of serial connection. 115200 — Default. <range> — Supported port speed.
Web: n/a UCI: terminal.console.type Opt: type	String value represents supported terminal emulation mode. vt100 — Default. <string> — Supported terminal type.
Web: n/a UCI: terminal.console.flowcontrol Opt: flowcontrol	Enables hardware flow control RTS/CTS. 0 — Disabled. 1 — Enabled.

Table 139: Information table for terminal settings

40.3 Configuring terminal package using UCI

```
root@VA_router:~# uci show terminal
terminal.ttySC0=terminal
terminal.ttySC0.enabled=1
terminal.ttySC0.device=ttySC0
terminal.ttySC0.speed=115200
terminal.ttySC0.type=vt100
terminal.ttySC0.flowcontrol=1
```

40.4 Configuring terminal using package options

```
root@VA_router:~# uci export terminal
package terminal

config terminal 'ttySC0'
        option enabled '0'
        option device 'ttySC0'
        option speed '115200'
        option type 'vt100'
        option flowcontrol '1'
```

40.5 Terminal diagnostics

40.5.1 Checking terminal entry in inittab

To check if terminal configuration is running, enter the following commands and confirm the line referring to the device name is present and looks similar to the last line below:

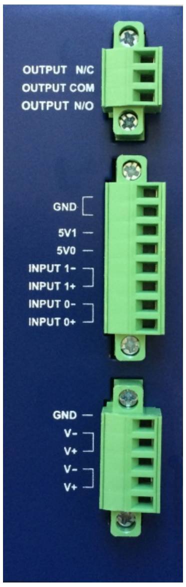
```
root@VA_router:~# cat /etc/inittab
::sysinit:/etc/init.d/rcS S boot
::shutdown:/etc/init.d/rcS K stop
ttyLTQ0::askfirst:getty -L 115200 ttyLTQ0 vt100
ttyLTQ1::askfirst:getty -L 115200 ttyLTQ1 vt100
ttySC0::respawn:getty -h -L 115200 ttySC0 vt100
```

41 Configuring GPIO

The Virtual Access GW2028 I/O interface has the following features:

- Two digital opto-isolated input ports
- One relay output port

41.1 GW2028 connectors



OUTPUT N/C	Normal Closed Isolated Relay Output
OUTPUT COM	Common (Isolated)
OUTPUT N/O	Normal Open Isolated Relay Output
GND	System Ground
5V1	5V wetting voltage for analogue input #1 (through 150R)
5V0	5V wetting voltage for analogue input #0 (through 150R)
INPUT 1	-ve analogue input #1 (optoisolator cathode)
INPUT 1+	+ve analogue input #1 (optoisolator anode though 150R)
INPUT 0	-ve analogue input #0 (optoisolator cathode)
INPUT0+	+ve analogue input #0 (optoisolator anode though 150R)
GND	System ground
V-	Power Supply Return
V+	9 – 59 V DC Power Supply Input #1
V-	Power Supply Return
V+	9 – 59 V DC Power Supply Input #2

Figure 195: Connectors on the GW2028

41.2 Digital opto-isolated input ports

You can use the digital input ports to connect to a device to monitor its status, for example an external sensor. The digital opto-isolated connectors are labelled as follows:

- INPUT 1-
- INPUT 1+
- INPUT 0-
- INPUT 0+

An event in the router's event system is raised when the status of the digital inputs changes. You can use the router's forwarding event system to forward events to a Syslog server, SNMP, email or SMS.

41.3 Configuring the event system using UCI

You can configure the va_event system via the command line interface or by using the management server, Activator.

From the command line, change to the following directory: **cd /etc/config**

An example setting is shown below. The forwarding section is configured to monitor the status of the digital I/O ports and send an alert to the configured Syslog server.

Class	ID	Name	Severity	Specific Template
system	2	DigitalInputChange	notice	Digital Input

To view a full list of events, type **vae_cli –d.**

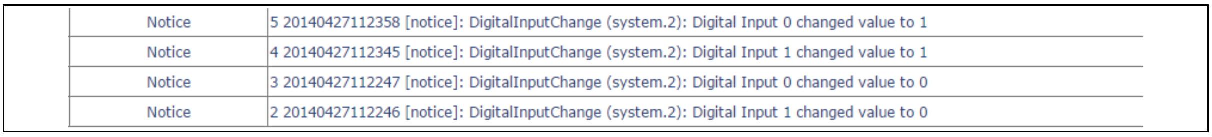
```
config va_eventd 'main'
                        option enabled 'yes'
                               option event_queue_file '/tmp/event_buffer'
                               option event_queue_size '128K'

        config target
                               option name 'syslog1'
                               option enabled 'yes'
                               option type 'syslog'
                               option target_addr '192.168.233.254:514'
                               option conn_tester 't1'

        config forwarding
                               option enabled 'yes'
                               option target 'syslog1'
                               option className 'system'
                               option eventName 'DigitalInputChange'
                               option severity 'warning-critical'
```

An example of open and closing input switches causing syslog alert messages on Monitor is shown below.

Figure 196: Syslog alert messages on Monitor



Notice	5 20140427112358 [notice]: DigitalInputChange (system.2): Digital Input 0 changed value to 1
Notice	4 20140427112345 [notice]: DigitalInputChange (system.2): Digital Input 1 changed value to 1
Notice	3 20140427112247 [notice]: DigitalInputChange (system.2): Digital Input 0 changed value to 0
Notice	2 20140427112246 [notice]: DigitalInputChange (system.2): Digital Input 1 changed value to 0

Figure 197: Severity wanting 'notice' for digital input change

INPUT	Description
Isolated Digital Input - Dry	Current supplied by external equipment
General Purpose Input 0: SW7	1-8 and 3-6 closed. Others open
General Purpose Input1: SW5	1-8 and 3-6 closed. Others open
Non-isolated Digital Input - Wet	Board supplies current to GPIO_IN+
General Purpose Input 0: SW7	1-8 and 3-6 closed. Others open
General Purpose Input 1: SW5	1-8 and 3-6 closed. Others open

Table 140: Dry and wet inputs with criteria

41.4 Relay output port

The relay will make or break a circuit depending on the state of the digital output port.

OUTPUT N/O	Normal Open
OUTPUT COM	Common
OUTPUT N/C	Normal Closed

Table 141: Connector labels

The output is controlled by command line entries.

digital_io.sh o0 1	Turns relay state on
digital_io.sh o0 0	Turns relay state off
digital_io.sh o0	Returns current state of relay

Table 142: Command line entries and their descriptions

41.4.1 Configuring the relay output port

This script is automatically installed in version LIS-15.00.52 and above, so no special configuration is required.

For versions prior to LIS15.00.73.00, configure the router's output port using a script like the one shown below. Load the script in UDS (/etc/config/uds).

```
package uds
config script 'relay'
         option enabled 'yes'
         option exec_type 'none'
         option type 'sh'
         list text 'GPIO="/sys/class/gpio/gpio103/value"'
         list text '[ ! -e "$GPIO" ] && {'
         list text '    echo "$GPIO doesn'\''t exist. Exiting..." >&2'
         list text '    exit 1'
```

```
        list text '}'
        list text 'case "$1" in'
        list text '    "") val="$(cat $GPIO)"'
        list text '        [ "$val" = "1" ] && val="off" || val="on"'
        list text '        echo "$val"'
        list text '        ;;'
        list text '    "off") echo 1 >$GPIO;;'
        list text '    "on") echo 0 >$GPIO;;'
        list text '    *) echo -e "USAGE:\n${0##*/}\t\tprint current
state\n${0##*/} on\t\tclose the relay\n${0##*/} of f\t\topen the
relay\n";;'
        list text 'esac'

config script 'link_relay'
        option enabled 'yes'
        option exec_type 'once'
        option type 'sh'
        list text 'ln -sf /var/uds/relay /usr/bin/relay_state'
        list text '    "on") echo 1 >$GPIO;;'
        list text '    *) echo -e "USAGE:\n${0##*/}\t\tprint current
state\n${0##*/} on\t\tclose the relay\n${0##*/} of f\t\topen the
relay\n";;'
        list text 'esac'
config script 'link_relay'
        option enabled 'yes'
        option exec_type 'once'
        option type 'sh'
        list text 'ln -sf /var/uds/relay /usr/bin/relay_state'
```

42 Configuring SCADA RTU (RTUD)

This chapter describes how to configure the SCADA RTU feature on a Virtual Access router. SCADA RTU is only available on routers with a digital I/O interface.

You can edit parameters using:

- the text editor 'vi' or 'nano' after logging in using SSH;
- the router's web interface; or
- Virtual Access' Activator.

42.1 Terminology

DI	Digital Input
DO	Digital Output
DNP3	Distributed Network Protocol version 3
I/O	Input/Output
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
Where a configuration parameter has the value of 1 or 0	1 = Enabled 0 = Disabled
Where a configuration parameter has the value NULL	This means blank. Specify as " "

42.2 SCADA RTU overview

The GW2027 and GW2028 routers have an integrated digital IO block consisting of 3 digital inputs (DI) and 1 digital output (DO). The digital inputs are presented on the terminal block as a series of input contact terminals. The digital output is presented on the terminal block as a relay output contact.

The SCADA RTU feature is implemented on the router by the RTUD daemon application. It allows the remote SCADA master to monitor and control the digital IOs of the Virtual Access router that acts as the RTU slave using several supported SCADA communication protocols:

- IEC 60870-5-104 (IEC104)
- DNP3 over TCP
- Modbus TCP

42.3 Configuration package used

Package	Sections
rtud	main
	iec
	dnp3
	mbtcp

42.4 Configuring SCADA RTUD using the web interface

To configure SCADA RTUD using the web interface, in the top menu browse to **Services -> SCADA RTUD**. The SCADA RTU page appears.

In Main Settings there are five tabs for SCADA RTUD options.

42.4.1 Configure general options

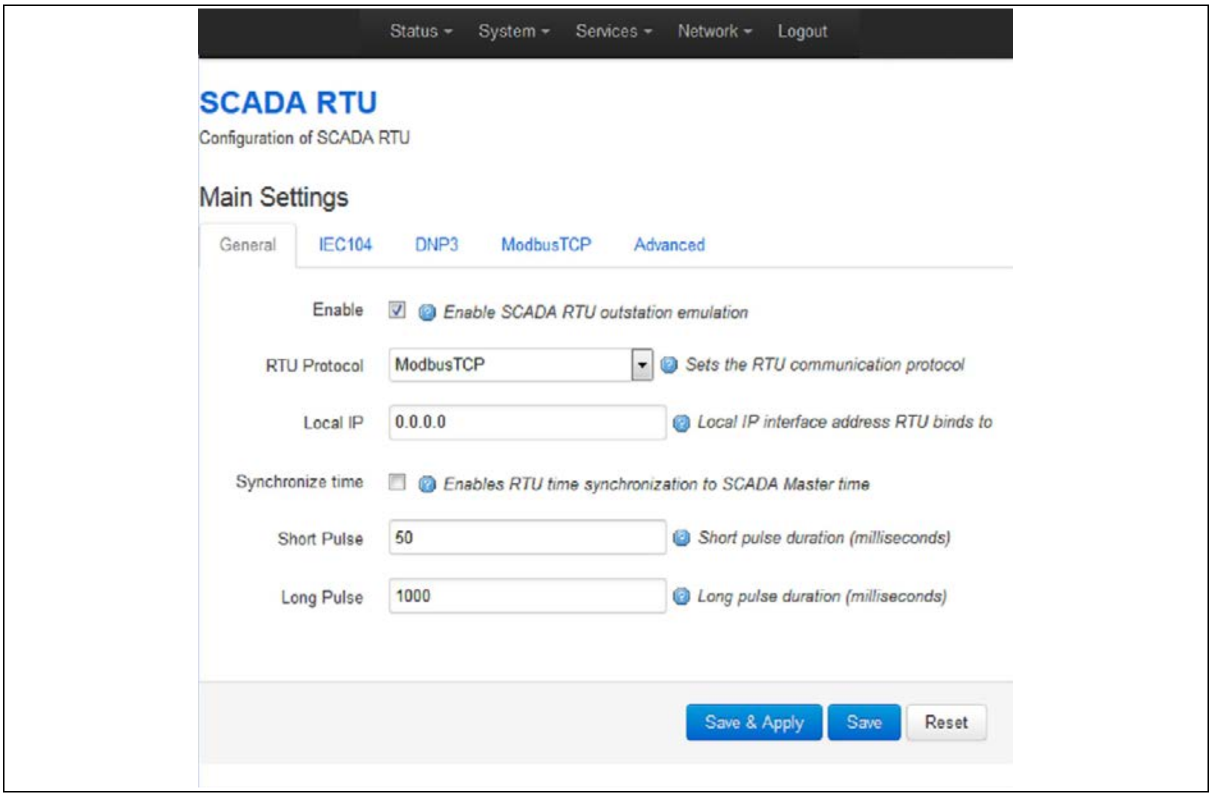


Figure 198: The SCADA RTU general options page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled
UCI:.rtud.main.enabled
Opt: enabled</td><td>Enables or disables SCADA RTU application.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
<tr><td>Web: RTU Protocol
UCI: rtud.main.protocol
Opt: protocol</td><td>Sets the RTU communication protocol.<table><tr><td>iec104</td><td>IEC 60870-5-104</td></tr><tr><td>mbtcp</td><td>Modbus TCP</td></tr><tr><td>dnp3</td><td>Distributed Network Protocol V3 (over TCP).</td></tr></table></td></tr>
<tr><td>Web: Local IP
UCI: rtud.main.local_ip
Opt: local_ip</td><td>Local IP interface address the RTU binds to.<table><tr><td>0.0.0.0</td><td></td></tr><tr><td>Range</td><td>A valid IPv4 or IPv6 address</td></tr></table></td></tr>
<tr><td>Web: Synchronize Time
UCI: rtud.main.sync_time
Opt: sync_time</td><td>Enables RTU time synchronization to master time. If enabled, the router will set its clock as the corresponding commands from the master station in each communication protocol.<table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
</table>

Web: Short Pulse UCI: rtud.main.short_pulse Opt: short_pulse	Short pulse duration in milliseconds, currently used in IEC104 protocol in processing digital output setting command, if the master specifies its use. 50 Range 10-1000
Web: Long Pulse UCI: rtud.main.long_pulse Opt:long_pulse	Long pulse duration in milliseconds, currently used in IEC104 protocol in processing digital output setting command, if the master specifies its use. 1000 Range 10-1000

Table 143: Information table for RTUD general options

42.4.2 Configure advanced options

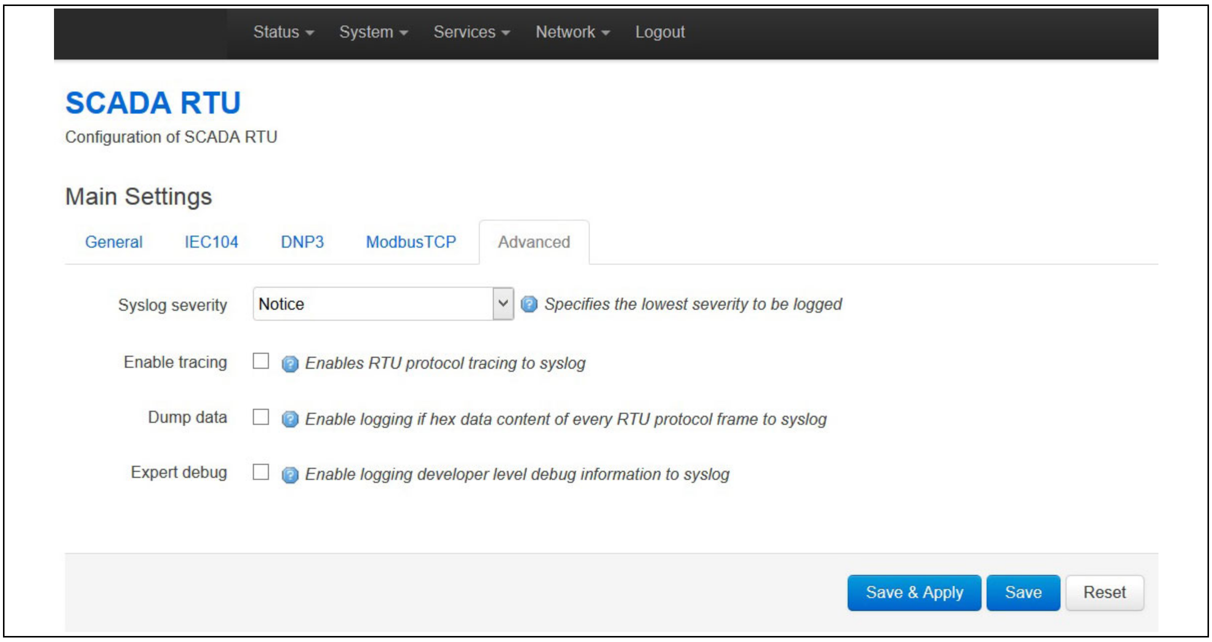


Figure 199: The SCADA RTU advanced setting page

Web Field/UCI/Package Option	Description
Web: Log level UCI: rtud.main.loglevel Opt: loglevel	Determines the syslog level. Events up to this priority will be logged. Emergency 0 Alert 1 Critical 2 Error 3 Warning 4 Notice 5 Info 6 Debug 7
Web: Trace UCI: rtud.main.trace_on Opt:trace_on	Enables protocol tracing to syslog. 1 Enabled. 0 Disabled.
Web: Dump data UCI: rtud.main.dump_data Opt: dump_data	Enables logging the context of protocol frames in ASCII hex format to syslog. 1 Enabled. 0 Disabled.

Web: Expert debug UCI: rtud.main.expert_debug Opt: expert_debug	Enables highest level of debug logging. For Virtual Access engineering use only.

Table 144: Information table for advanced options

42.4.3 Configuring IEC104 options

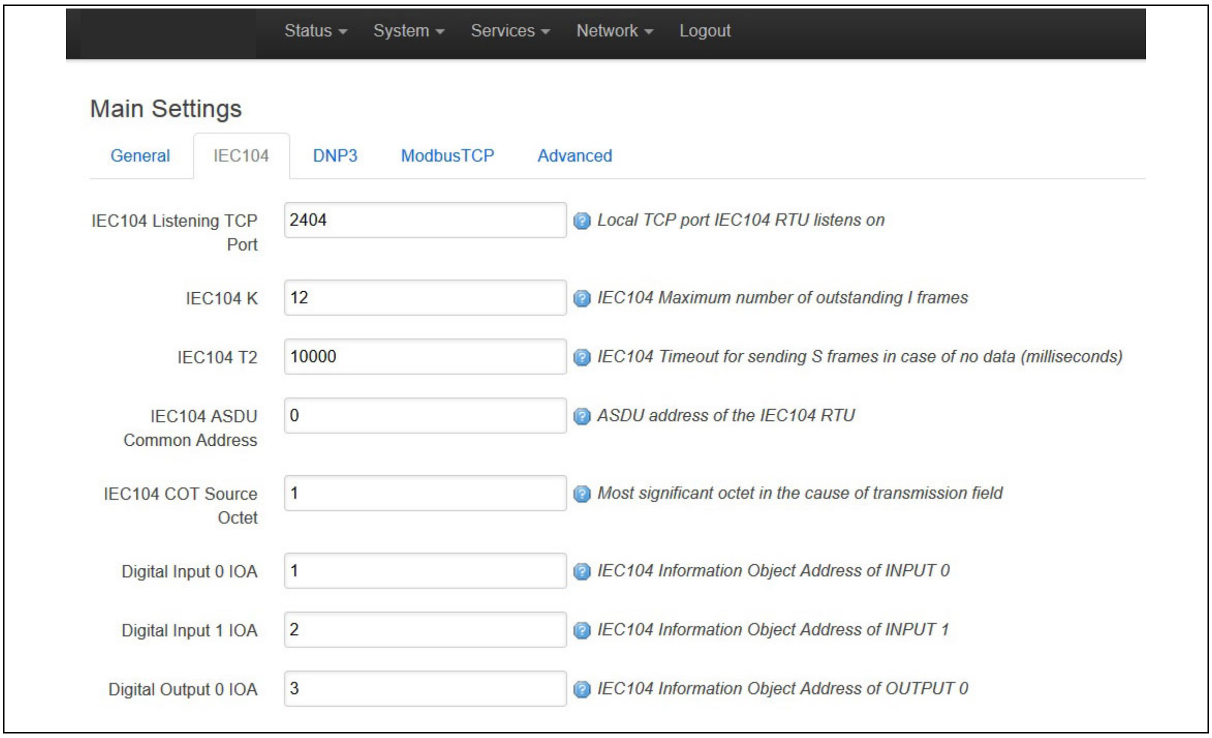


Figure 200: The SCADA RTU IEC104 settings page

Web Field / UCI / Package Option	Description
Web: IEC104 Listening TCP Port UCI: rtud.main.iec104_listen_tcpport Opt: iec104_listen_tcpport	Local TCP port IEC104 RTC listens on. Range: 1-65535 2404
Web: IEC104 K UCI: rtud.main.iec104_k Opt:iec104_k	IEC parameter K. Maximum number of outstanding frames. Range: 1-3267 12
Web: IEC104 T2 UCI: rtud.main.iec104_t2 Opt: iec104_t2	IEC104 parameter T2. Timeout for sending, in milliseconds, S frames in case of no data. Range: 1-6000 10000
Web: IEC104 ASDU Common Address UCI: rtud.main.iec104_asdu_addr Opt: iec104_asdu_addr	IEC104 parameter CA (also known as CASDU). ASDU common address of the RTU. Range: 1-65535 0
Web: IEC104 COT Source Octet UCI: rtud.main.iec104_cot_source_octet Opt: iec104_cot_source_octet	IEC104 parameter COT value. The value of the most significant octet in the ‘cause of transmission’ header field. Range: 0-255 1
Web: Digital Input 0 IOA UCI: rtud.main.dg_input0_ioaddr Opt: dg_input0_ioaddr	IEC104 Information Object Address (IOA) of Digital Input 0. Range: 1-1677712 1

Web: Digital Input 1 IOA UCI: rtud.main.dg_input1_ioaddr Opt: dg_input1_ioaddr	IEC104 Information Object Address (IOA) of Digital Input 1. Range: 1-1677712 2
Web: Digital Output 0 IOA UCI: rtud.main.dg_output0_ioaddr Opt: dg_output0_ioaddr	IEC104 Information Object Address (IOA) of Digital Output 0. Range: 1-1677712 3

Table 145: Information table for IEC104 options

42.4.4 Configure DNP3 options

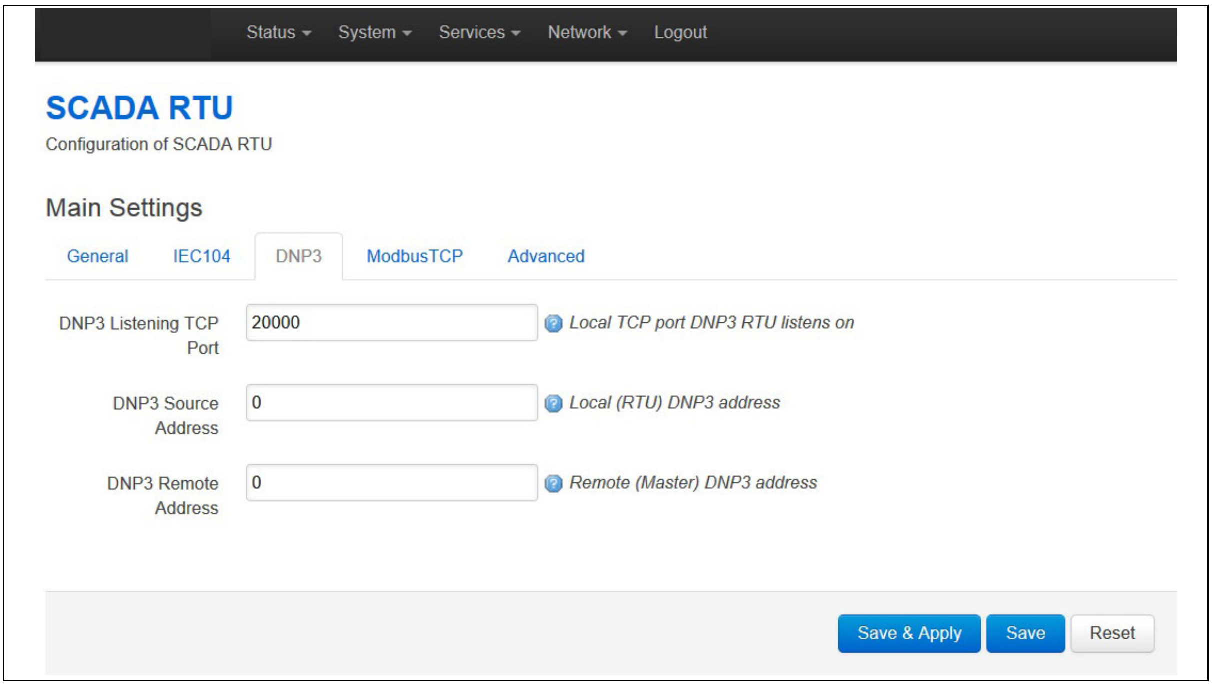


Figure 201: The SCADA RTU DNP3 settings page

Web Field / UCI / Package Option	Description
Web: DNP3 Listening TCP Port UCI: rtud.main.dnp3_listen_tcpport Opt: dnp3_listen_tcpport	Local TCP port DNP3 RTU listens on. Range: 1-65535 2000
Web: DNP3 Source Address UCI: rtud.main.dnp3_dl_srcaddr Opt: dnp3_dl_srcaddr	Local (RTU) DNP3 address. Range: 0-65535 0
Web: DNP3 Remote Address UCI: rtud.main.dnp3_dl_dstadr Opt: dnp3_dl_dstadr	Remote (Master) DNP3 address. Range: 0-255 1

Table 146: Information table for DNP3 options

42.4.5 Configure Modbus options

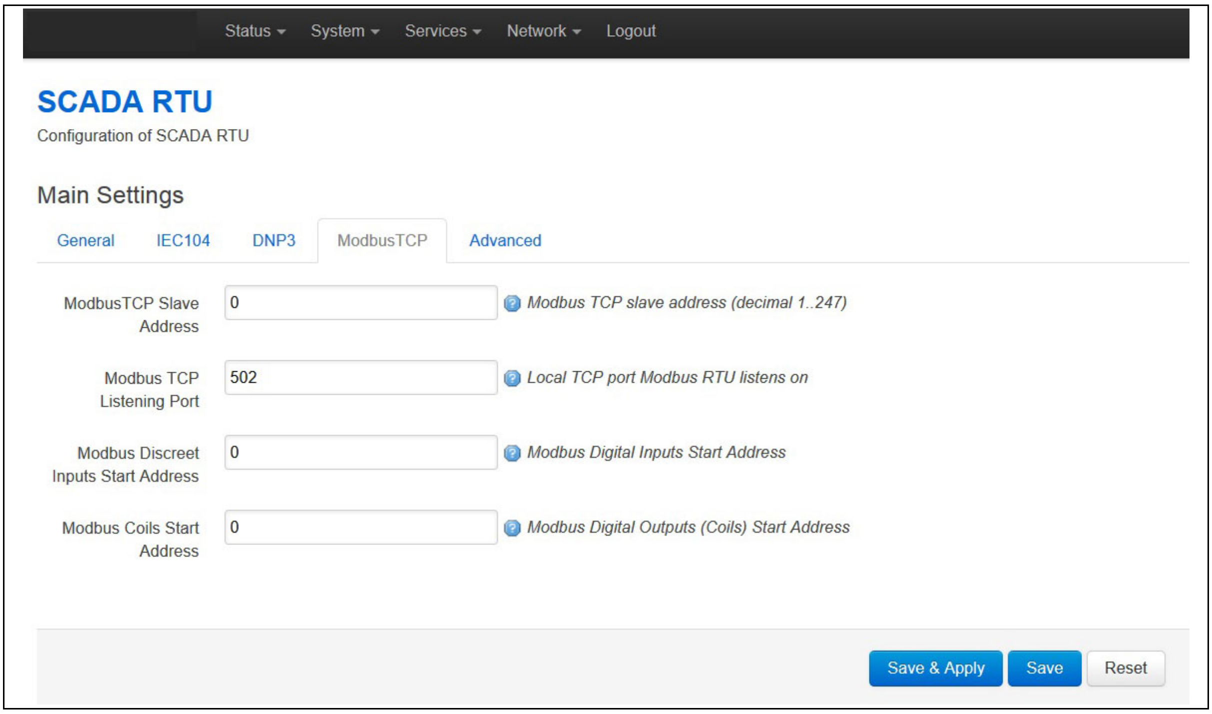


Figure 202: The SCADA RTU Modbus settings page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Modbus TCP Slave Address
UCI: rtud.main.mbtcp_devaddr
Opt: mbtcp_devaddr</td><td>Modbus slave address.<table><tr><td>Range</td><td>1-247</td></tr><tr><td>0</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus TCP Listening Port
UCI: rtud.main.mbtcp_listen_tcpport
Opt: mbtcp_listen_tcpport</td><td>Local TCP port Modbus RTU listens on.<table><tr><td>Range</td><td>1-65535</td></tr><tr><td>502</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus Discreet Inputs Start Address
UCI: rtud.main.mbtcp_di_start_addr
Opt: mbtcp_di_start_addr</td><td>Modbus Discreet Inputs start address. This is the address of the first Digital Input in the Modbus data model.
Note: address of inputs and outputs are allowed to overlap, that is, may be the same.<table><tr><td>Range</td><td>0-65535</td></tr><tr><td>0</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus Coils Start Address
UCI: rtud.main.mbtcp_co_start_addr
Opt: mbtcp_co_start_addr</td><td>Modbus Coils Start address. This is the address of the first Digital Output in the Modbus data model.
Note: address of inputs and outputs are allowed to overlap, that is, may be the same.<table><tr><td>Range</td><td>0-65535</td></tr><tr><td>0</td><td></td></tr></table></td></tr>
</table>

Table 147: Information table for Modbus options

42.5 Controlling the RTUD application manually using the web interface

When you have enabled RTUD, the application starts automatically. If necessary, you can control the application manually.

Browse to the top menu and select **System -> Startup**.

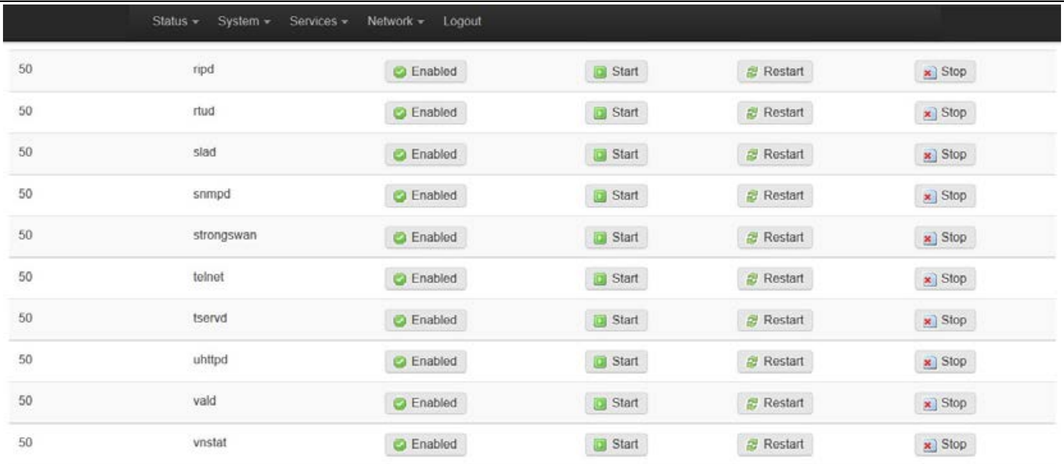


Figure 203: The startup page

Find the RTUD entry and click **Enabled**/**Disabled**, **Start**, **Restart**, or **Stop**, depending on which option you require.

To check if the application is running, select **Status -> Processes.** The Processes page appears.

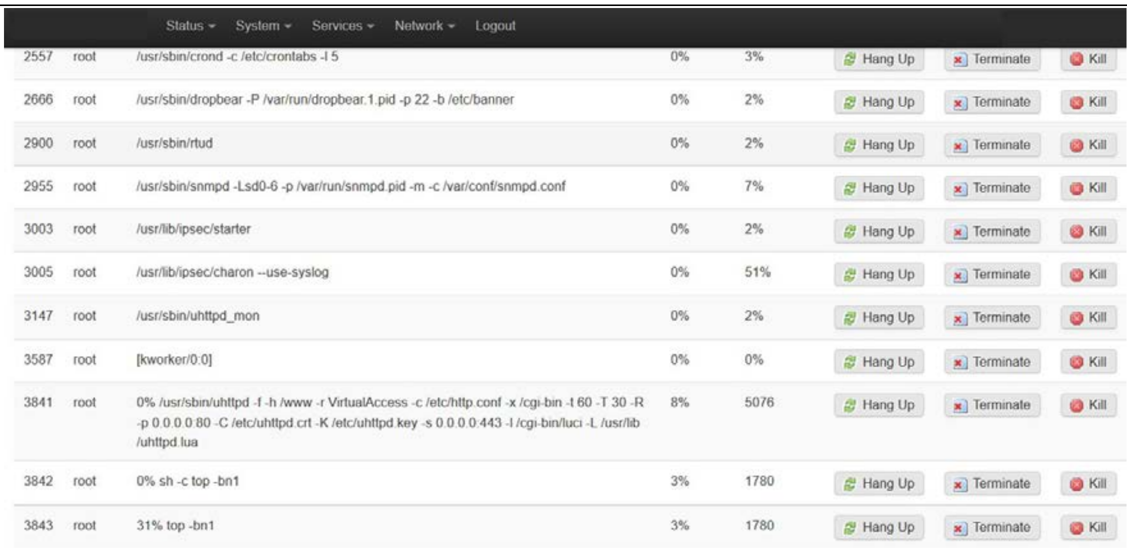


Figure 204: The status process page

42.6 Viewing RTUD statistics using the web interface

To view the SCADA RTU point list, session status and counters, from the top menu select **Status -> SCADA RTU**.

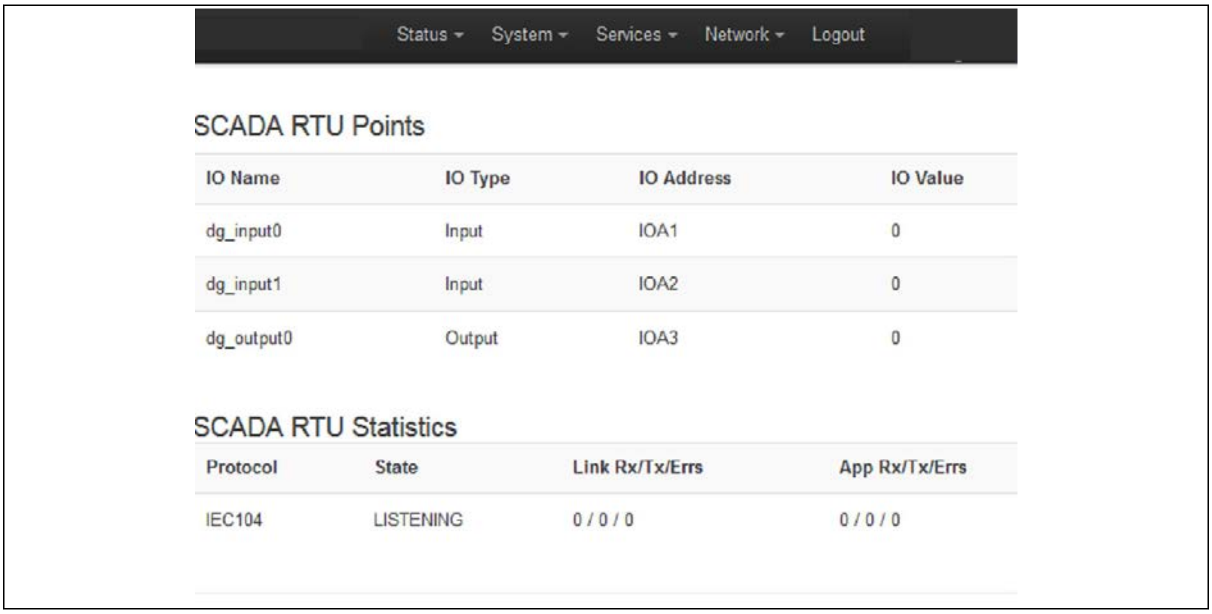


Figure 205: The SCADA RTU points screen

42.7 Configuring RTUD using command line

The RTUD configuration is stored in **/etc/config/rtud**

You must restart the RTUD application for your option changes to take effect.

The default content of the RTUD configuration file is shown below.

42.7.1 RTUD using UCI

```
root@VA_router:~# uci show rtud
rtud.main=rtud
rtud.main.enable=1
# set to 1 to enable RTUD daemon
rtud.main.protocol=iec104
rtud.main.local_ip=0.0.0.0
rtud.main.sync_time=0
rtud.main.short_pulse=50
rtud.main.long_pulse1000
rtud.main.loglevel=5
rtud.main.trace_on=0
rtud.main.dump_data=0
```

```
rtud.main.expert_debug=0

rtud.main.iec104_listen_tcpport=2404
rtud.main.iec104_k=12
rtud.main.iec104_t2=10000
rtud.main.iec104_asdu_addr=0
rtud.main.iec104_cot_source_octet=1

rtud.main.dg_input0_ioaddr=1
rtud.main.dg_input1_ioaddr=2
rtud.main.dg_output0_ioaddr=3

rtud.main.dnp3_listen_tcpport=20000
rtud.main.dnp3_dl_srcaddr=0
rtud.main.dnp3_dl_dstaddr=0

rtud.main.mbtcp_devaddr=0
rtud.main.mbtcp_listen_port=502
rtud.main.mbtcp_di_start_addr=0
rtud.main.mbtcp_co_start_addr=0
```

42.7.2 RTUD using package options

```
root@VA_router:~# uci export rtud
package rtud
config rtud main
    # set to 1 to enable RTUD daemon
    option enable 0
    option protocol 'iec104'
    option local_ip '0.0.0.0'
    option sync_time 0
    option short_pulse 50
    option long_pulse 1000
    option loglevel 5
    option trace_on 0
    option dump_data 0
    option expert_debug 0
```

```
    option iec104_listen_tcpport 2404
    option iec104_k 12
    option iec104_t2 10000
    option iec104_asdu_addr 0
    option iec104_cot_source_octet 1

    option dg_input0_ioaddr 1
    option dg_input1_ioaddr 2
    option dg_output0_ioaddr 3

    option dnp3_listen_tcpport 20000
    option dnp3_dl_srcaddr 0
    option dnp3_dl_dstaddr 0

    option mbtcp_devaddr 0
    option mbtcp_listen_port 502
    option mbtcp_di_start_addr 0
    option mbtcp_co_start_addr 0
```

42.7.3 Controlling the RTUD application manually using UCI

When you have enabled RTUD, the application starts automatically. If necessary, you can control the application manually using the router's command line.

42.7.3.1 Starting the application

```
/etc/init.d/rtud start
```

42.7.3.2 Restarting the application

```
/etc/init.d/rtud restart
```

42.7.3.3 Stopping the application

```
/etc/init.d/rtud stop
```

42.7.3.4 Checking the application is running

```
ps | grep rtud
```

This command returns the process ID if the application is running or nothing if the application is not running.

42.8 RTUD diagnostics

To view RTUD diagnostic options, enter the command **rtu**:

```
root@VA_router:~# rtu
=== RTU daemon diagnostics. Command syntax: ===

rtu set loglevel <level> (0 to 7)
rtu show config - show config
rtu show stats - show stats
rtu clear stats - clear stats
rtu show points - show RTU IO points
rtu show dnp3 - show DNP3 stats
rtu show modbus - show Modbus stats
rtu set point <IO name> <value> set output IO point value
```

43 SCADA IEC104 gateway

43.1 Overview

Supervisory control and data acquisition (SCADA) systems are used by industrial organisations and companies to control and monitor physical processes, examples of which are transmission of electricity, transportation of gas and oil in pipelines, water distribution and traffic lights. Alarm handling is usually an important part of most SCADA implementations.

SCADA systems usually consist of:

- Superviory computers
- Remote terminal units (RTUs)
- Programmable logic controllers (PLCs)

The IEC104 Gateway feature on the router is used for SCADA protocol conversion where the SCADA master is ruuning IEC104 protocol:

- IEC104 to IEC101 conversion (balanced and unbalanced)
- IEC104 to DNP3
- IEC104 to MODBUS (serial and TCP)
- IEC61850 to IEC101 unbalanced conversion

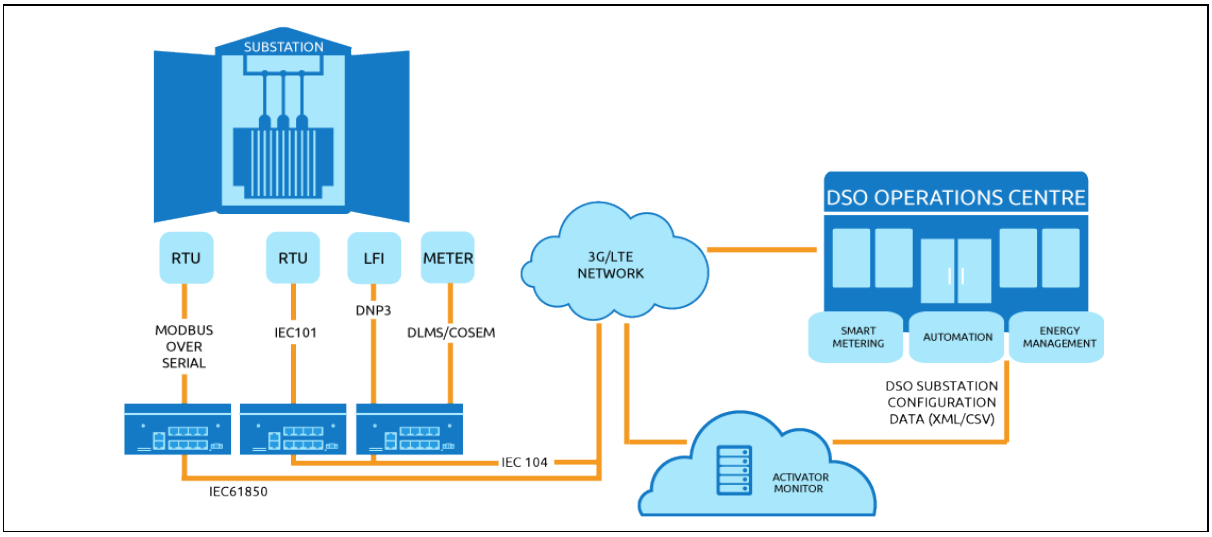


Figure 206: Example network for IEC104 to RTU protocol conversion

Configuration for the above conversions is done in two parts:

- **IEC104 Gateway** (iecd package), and
- **Terminal Server** (tservd package).

The IEC104 Gateway handles the protocol processing while the Terminal Server handles low level serial communication.

Note: The Terminal Server is not required for IEC104 to Mobus TCP.

43.2 Configuration packages used

Package	Sections
iecd	main, port, point
tservd	main, port

43.3 IEC104 gateway configuration using the web interface

In the top menu, select **Services -> IEC104 Gateway**. The IEC104 Gateway page appears.

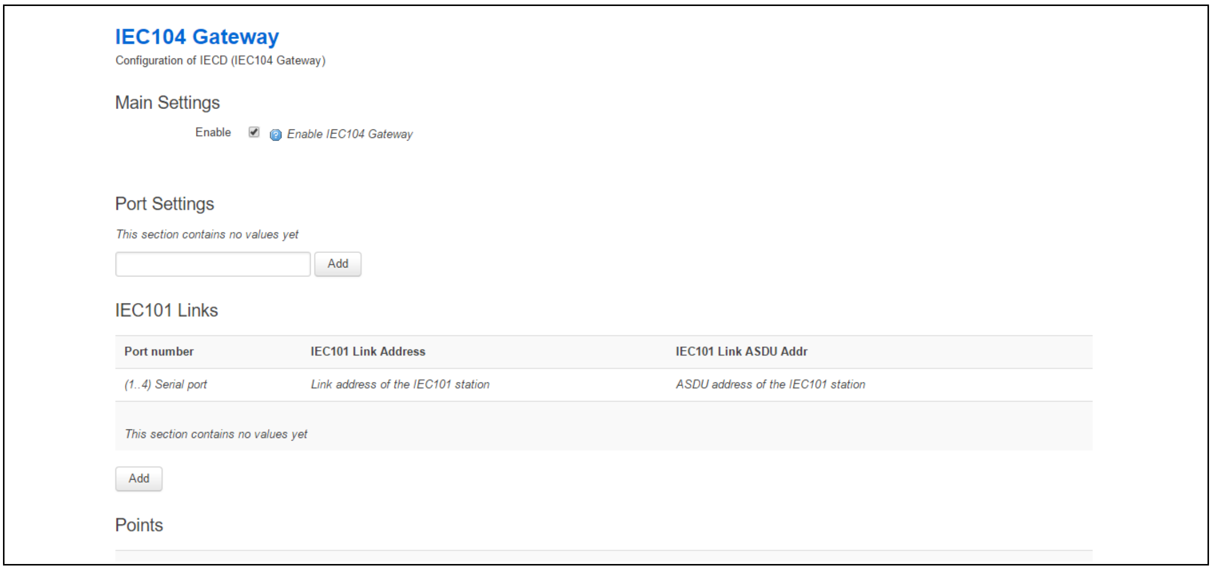


Figure 207: The IEC104 Gateway configuration page

There are four sections in the IEC104 Gateway page:

Section	Description
Main Settings	Enables the IEC104 Gateway.
Port Settings	Sets the IEC104 SCADA Master communication settings and the protocol methods used by the RTUs: • IEC101 unbalanced or balanced • DNP3 • Modbus over serial • Modbus over TCP
IEC101 Links	Defines the IEC101 slave links used in IEC101 conversion. Each link is defined by a config iec101link section block. There is a maximum of 32 links supported. In IEC101 unbalanced mode all of these links can be used. In IEC101 balanced mode only one outstation per serial port is assumed since these will be point to point links.
Points	Configures the data point mappings. **Note**: There are no data point mappings in IEC104 to IEC101 conversion.

43.3.1 Main settings

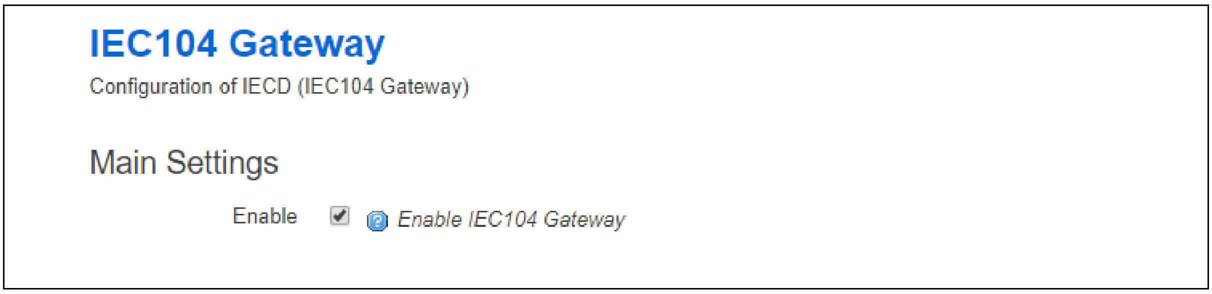


Figure 208: The IEC104 Gateway main setings configuration page

Web Field / UCI / Package Option	Description
Web: Enable UCI: iecd.main.enable Opt: enable	Enables IEC104 gateway. 0: Disabled. 1: Enabled.

Table 148: Information table for IEC104 Gateway main settings configuration

43.3.2 Port settings

The port configuration will depend on the desired protocol conversion. There are 5 sections.

Section	Description
General	Enables an IEC104 port and selects the RTU protocol method.
IEC104	Defines the IEC104 gateway configuration for communication with the SCADA Master.
IEC101	Defines the IEC104 to IEC101 conversion parameters.
DNP3	Defines the IEC104 to DNP3 conversion parameters.
Modbus	Defines the IEC104 to MODBUS conversion parameters (Modbus over serial or Modbus over TCP).
Advanced	Defines logging and TCP keepalive options for all conversion methods.

In the Port Settings section, enter a text name that will be used for the iecd port section, for example, Port1. Select **Add**. The IECD port configuration options appear.

43.3.2.1 Port settings: general

In this section you can configure general port settings. Enable the port and select the RTU Protocol from the drop-down menu.

Figure 209: The IEC104 Gateway port general configuration page

Web Field / UCI / Package Option	Description
Web: Enable UCI: iecd.<port>.enable Opt: enable	Enables an IECD port. 0 — Disabled. 1 — Enabled.
Web: RTU Protocol UCI: iecd.<port>.master_protocol Opt: master_protocol	Defines the protocol method used by the RTUs that connect to this router. iec101 — IEC101 dnp3 — DNP3 modbus — MODBUS
Web: n/a UCI: iecd.<port>.slave_protocol Opt: slave_protocol	Defines the protocol method used by the SCADA Master. iec104 — IC104 iec61850 — IEC61850
Web: n/a UCI: iecd.<port>.iec61850_local_ip Opt: iec61850_local_ip	Defines the local IP address this IEC61850 peer binds to.
Web: n/a UCI: iecd.<port>.iec61850_local_tcpport Opt: iec61850_local_tcpport	Defines the local TCP port this IEC104 peer listens on. 2404 Range — 1 - 65535
Web: n/a UCI: iecd.<port>.tcp_user_timeout Opt: tcp_user_timeout	Defines the maximum time in milliseconds to wait for a TCP ACK after data transmission before closing connection in TCP established state. Set to **0** to use kernel defaults (about 15-20 minutes). 20000 — milliseconds Range
Web: n/a UCI: iecd.<port>.pointmap_file Opt: pointmap_file	Defines the path to the points map file, for example: `/root/iecd/iecd_points1.csv`

Table 149: Information table for IEC104 Gateway port general configuration

43.3.2.2 Port settings: IEC104

In this section you can configure the IEC104 settings.

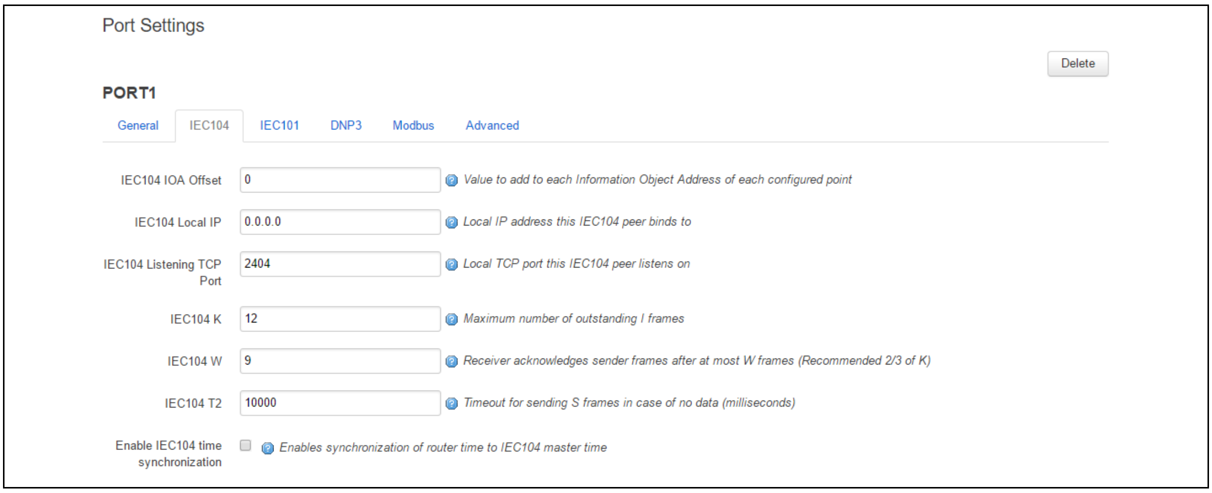


Figure 210: The IEC104 Gateway port IEC104 configuration page

Web Field / UCI / Package Option	Description
Web: IEC104 IOA Offset UCI: iecd.<port>.ioa_offset Opt: ioa_offset	Defines the value to add to each Information Object Address of each configured point. 0 Range
Web: Local IP UCI: iecd.<port>.iec104_local_ip Opt: iec104_local_ip	Defines the local IP address this IEC104 peer binds to.
Web: IEC104 Listening TCP Port UCI: iecd.<port>.iec104_local_tcpport Opt: iec104_local_tcpport	Defines the local TCP port this IEC104 peer listens on. 2404 Range 1 - 65535
Web: IEC104 K UCI: iecd.<port>.iec104_k Opt: iec104_k	Defines the maximum number of outstanding I frames. 12 Range
Web: IEC104 W UCI: iecd.<port>.iec104_w Opt: iec104_w	Defines the number of frames after which the receiver will acknowledge. It is recommended that this value be 2/3 the value of IEC104 K. 9 Range
Web: IEC104 T2 UCI: iecd.<port>.iec104_t2 Opt: iec104_t2	Defines the timeout in milliseconds for sending S frames in case of no data. 10000 milliseconds Range
Web: Enable IEC104 time synchronization UCI: iecd.<port>.iec104_sync_time Opt: iec104_sync_time	Enables synchronization of router time to IEC104 master time. 1 Enable synchronization 0 Disable synchronization
Web: n/a UCI: iecd.<port>.iec104_gi_resp_time Opt: iec104_gi_resp_time	Defines the time in milliseconds between sending successive general interrogation response messages. 200 milliseconds Range 50 - 1000
Web: n/a UCI: iecd.<port>.iec104_txq_size Opt: iec104_txq_size	Defines the the maximum size of transmit ASDU queue in the application layer (number of frames). 128 Range 2 - 256
Web: UCI: n/a iecd.<port>.iec104_time_tagged_cmds Opt: iec104_time_tagged_cmds	Enables support for IEC104 CP56TIME2A tagged commands. 0 Range
Web: n/a UCI: iecd.<port>.iec104_cmd_delay_time Opt: iec104_cmd_delay_time	Defines the maximum allowable received command age in milliseconds. If set to 0, any age is allowed. 5000 Milliseconds Range 1000 - 60000
Web: n/a UCI: iecd.<port>.iec104_fsm_debug_on Opt: iec104_fsm_debug_on	Enables log for IEC104 state transitions and events. 0 Enable. 1 Disable.
Web: n/a UCI: iecd.<port>.iec104_dump_data Opt: iec104_dump_data	Enables RX/TX Hex dump. 0 Enable 1 Disable
Web: n/a UCI: iecd.<port>.iec104_trace_on Opt: iec104_trace_on	Enables protocol tracing. 0 Enable. 1 Disable.

Table 150: Information table for IEC104 Gateway port IEC104 configuration

43.3.2.3 Port settings: IEC101

IEC104 to IEC101 conversion feature of the router allows converting commands in the control direction, and the responses and process data in the monitor direction, between the SCADA master running the IEC104 protocol and the remote RTUs running IEC101 protocol over serial interface.

IEC104 to IEC101 conversion can be configured for two modes:

IEC 101 Mode	Description
Unbalanced	In the IEC101 unbalanced mode, the router supports communication of up to 32 IEC101 slaves connected onto the same serial interface.
Balanced	The IEC101 balanced mode is used in point to point configuration. That is, the router is communicating to a single IEC101 outstation on the serial interface. Each peer, either the controlling station (Master) or controlled station (RTU) can initiate communication in balanced mode.

Figure 211: The IEC104 Gateway port IEC101 configuration page

Web Field / UCI / Package Option	Description
Web: IEC101 Startion Target IP UCI: iecd.<port>.iec101_target_ip Opt: iec101_target_ip	Defines the remote IP address of the IEC101 station to connect to. 127.0.0.1 Range
Web: IEC101 Station Target TCP Port UCI: iecd.<port>.iec101_target_tcpport Opt: iec101_target_tcpport	Defines the remote TCP port of the IEC101 station to connect to. 999 Range
Web: IEC101 Link Mode UCI: iecd.<port>.iec101_mode Opt: iec101_mode	Defines the IEC101link communication mode. unbalanced balanced
Web: IEC101 Station COT Tx Length UCI: iecd.<port>.iec101_cot_tx_length Opt: iec101_cot_tx_length	Defines the Cause of Transmission length (1 or 2 bytes). 2 bytes Range

Web: IEC101 Station COT Source Length UCI: iecd.<port>.iec101_cot_source_octet Opt: iec101_cot_source_octet	Defines the most significant octet in the Cause of Transmission field. 0 Range
Web: IEC101 Station ASDU Addr Length UCI: iecd.<port>.iec101_asdu_addrlen Opt: iec101_asdu_addrlen	Defines the length of Common Address of ASDU (1 or 2 bytes). 2 — bytes Range
Web: IEC101 Station Info Object Addr Length UCI: iecd.<port>.iec101_info_obj_addrlen Opt: iec101_info_obj_addrlen	Defines the length of the Information Object Address (1, 2 or 3 bytes). 2 — bytes Range
Web: IEC101 Station Poll Time UCI: iecd.<port>.iec101_data_polling_time Opt: iec101_data_polling_time	Defines the RTU polling interval in milliseconds if line is idle. 10000 — milliseconds Range
Web: IEC101 Link Addr Length UCI: iecd.<port>.iec101_link_addrlen Opt: iec101_link_addrlen	Defines the length of the link address field (0, 1 or 2 bytes). 1 — bytes Range
Web: n/a UCI: iecd.<port>.iec101_ack_delay Opt: iec101_ack_delay	Defines the time to wait for an IEC101 ACK in milliseconds. 0 — seconds Range
Web: n/a UCI: iecd.<port>.iec101_frame_rsp_time Opt: iec101_frame_rsp_time	Defines maximum number of milliseconds before resending an IEC101 frame. 2000 — milliseconds Range
Web: n/a UCI: iecd.<port>.iec101_max_tx_retry Opt: iec101_max_tx_retry	Defines maximum number of times to retry sending an IEC101 frame. 3 Range
Web: n/a UCI: iecd.<port>.iec101_txq_size Opt: iec101_txq_size	Defines size of transmit ASDU queue (number of frames) in the IEC101 link layer. 128 Range
Web: n/a UCI: iecd.<port>.iec101_send_spont_delay_acq Opt: iec101_send_spont_delay_acq	Defines whether to send DELAY ACQUISITION SPONTANEOUS message as part of ‘Acquisition of Transmission Delay’ procedure. Note: this option is used in the scenario where an IEC104 Master is talking to IEC101 RTU 0 — Do not send DELAY ACQUISITION SPONTANEOUS message 1 — Send DELAY ACQUISITION SPONTANEOUS message
Web: n/a UCI: iecd.<port>.iec101_fsm_debug_on Opt: iec101_fsm_debug_on	Enables logging IEC104 state transitions and events. 0 1
Web: n/a UCI: iecd.<port>.iec101_dump_data Opt: iec101_dump_data	Enables RX/TX Hex dump. 0 1
Web: n/a UCI: iecd.<port>.iec101_trace_on Opt: iec101_trace_on	Enables IEC101 protocol tracing. 0 1

Table 151: Information table for IEC104 Gateway port IEC101 configuration

43.3.2.4 Port settings: DNP3

IEC104 to DNP3 conversion feature of the router allows converting commands in the control direction, and the responses and process data in the monitor direction, between the SCADA master running the IEC104 protocol and the remote RTU running DNP3 over serial protocol.

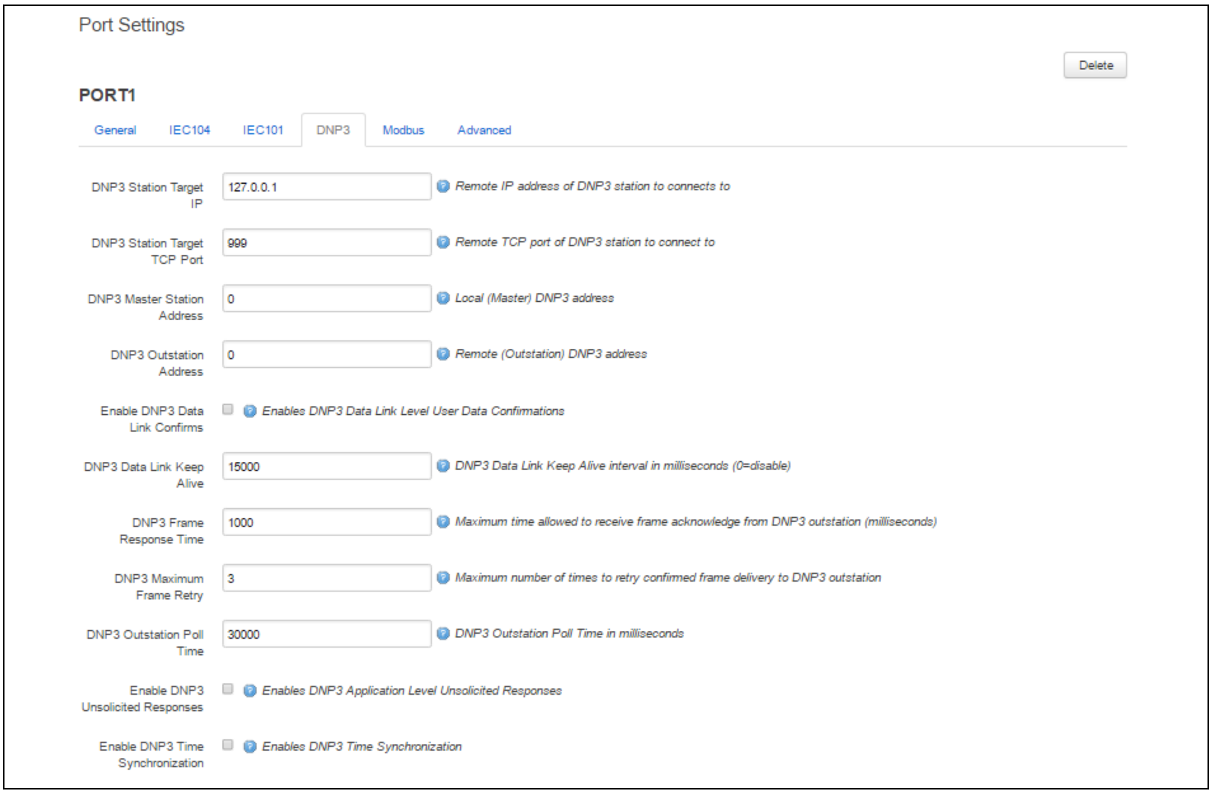


Figure 212: The IEC104 Gateway port DNP3 configuration page

Web Field / UCI / Package Option	Description
Web: DNP3 Station Target IP UCI: iecd.<port>.dnp3_target_ip Opt: dnp3_target_ip	Defines the remote IP address of the DNP3 station to connect to.
Web: DNP3 Station Target TCP Port UCI: iecd.<port>.dnp3_target_tcpport Opt: dnp3_target_tcpport	Defines the remote TCP port of the DNP3 station to connect to. 999 Range
Web: DNP3 Master Station Address UCI: iecd.<port>.dnp3_dl_srcaddr Opt: dnp3_dl_srcaddr	Defines the local (Master) DNP3 address. 0 Range
Web: DNP3 Outstation Address UCI: iecd.<port>.dnp3_dl_dstaddr Opt: dnp3_dl_dstaddr	Defines the remote (Outstation) DNP3 address. 0 Range
Web: Enable DNP3 Data Link Confirms UCI: iecd.<port>.dnp3_dl_cfrm_user_data Opt: dnp3_dl_cfrm_user_data	Enables DNP3 data link layer user data confirmations. 0 Range
Web: DNP3 Data Link Keep Alive UCI: iecd.<port>.dnp3_dl_keep_alive_int Opt: dnp3_dl_keep_alive_int	Defines the DNP3 data link keepalive interval in milliseconds (0 to disable). 15000 Milliseconds Range

Web: DNP3 Frame Response Time UCI: iecd.<port>.dnp3_dl_frame_rsp_time Opt: dnp3_dl_frame_rsp_time	Defines the maximum amount of time in milliseconds to receive a frame acknowledge from the DNP3 outstation.	1000	Milliseconds
		Range	
Web: DNP3 Maximum Frame Retry UCI: iecd.<port>.dnp3_dl_max_tx_retry Opt: dnp3_dl_max_tx_retry	Defines the maximum number of times to retry confirmed frame delivery to the DNP3 outstation.	3	
		Range	
Web: DNP3 Outstation Poll Time UCI: iecd.<port>.dnp3_app_poll_time Opt: dnp3_app_poll_time	Defines the DNP3 outsation poll time in milliseconds.	30000	Milliseconds
		Range	
Web: Enable DNP3 Unsolicited Responses UCI: iecd.<port>.dnp3_app_unsol_enable Opt: dnp3_app_unsol_enable	Enables DNP3 application level unsolicited responses.	1	Enables
		0	Disable
Web: Enable DNP3 Time Synchronizaton UCI: iecd.<port>.dnp3_app_sync_time Opt: dnp3_app_sync_time	Enables DNP3 time synchronization.	1	Enable
		0	Disable
Web: n/a UCI: iecd.<port>.dnp3_dl_utxq_size Opt: dnp3_dl_utxq_size	Defines the size of DNP3 data link transmit unconfirmed service frame queue (number of frames).	128	
		Range	2 – 256
Web: n/a UCI: iecd.<port>.dnp3_dl_ctxq_size Opt: dnp3_dl_ctxq_size	Defines size of DNP3 data link transmit confirmed service frame queue (number of frames).	128	
		Range	2 – 256
Web: n/a UCI: iecd.<port>.dnp3_app_read_attr Opt: dnp3_app_read_attr	Enables reading DNP3 device attributes at the start of the session. This feature is useful for debugging and is not recommended for production.	0	
		1	
Web: n/a UCI: iecd.<port>.dnp3_app_firstpoll_delay Opt: dnp3_app_firstpoll_delay	Defines initial timeout from start-up in milliseconds before performing first DNP3 integrity poll.	5000	milliseconds
		Range	5000 – 65535
Web: n/a UCI: iecd.<port>.dnp3_app_evpoll_time Opt: dnp3_app_evpoll_time	Defines DNP3 outstation event polling interval in milliseconds.	3000	milliseconds
		Range	1000 – 65535
Web: n/a UCI: iecd.<port>.dnp3_app_frag_rx_time Opt: dnp3_app_frag_rx_time	Defines DNP3 application level fragment response timeout.	10000	Milliseconds
		Range	1000 – 65535
Web: n/a UCI: iecd.<port>.dnp3_app_txq_size Opt:	Defines DNP3 application level transmit queue size (number of frames).	64	
		Range	2 – 256
Web: n/a UCI: iecd.<port>.dnp3_app_output_mode Opt: dnp3_app_output_mode	Defines a decimal code that controls how the router sends DNP3 binary output command to a DNP3 RTU. The most commonly used model is Select/Operate. **Note**: this command is used in scenario where the router is acting as a DNP3 master.	0	Use WRITE command.
		1	Use Select/Operate message sequence.
		2	Use Direct Operate message.
Web: n/a UCI: iecd.<port>.dnp3_app_evpoll_mode Opt: dnp3_app_evpoll_mode	Defines DNP3 outstation event polling interval in milliseconds.	3000	milliseconds
		Range	1000 – 65535

Web: n/a UCI: iecd.<port>.dnp3_fsm_debug_on Opt: dnp3_fsm_debug_on	Enables DNP3 link and application level state machine transition and event logging into syslog. 1 — Enable. 0 — Disable.
Web: n/a UCI: iecd.<port>.dnp3_object_parser_debug_on Opt: dnp3_object_parser_debug_on	Enables or disable logging low level debug information when parsing DNP3 objects in the received DNP3 slave messages 1 — Enable. 0 — Disable.
Web: n/a UCI: iecd.<port>.dnp3_dump_data Opt: dnp3_dump_data	Enables RX/TX Hex dump. 1 — Enable. 0 — Disable.
Web: n/a UCI: iecd.<port>.dnp3_trace_on Opt: dnp3_trace_on	Enables DNP3 protocol tracing. 1 — Enable. 0 — Disable.

Table 152: Information table for IEC104 Gateway port DNP3 configuration

43.3.2.5 Port settings: Modbus

The IEC104 to Modbus Conversion feature of the router allows converting commands in the control direction and the responses and process data in the monitor direction between the SCADA Master running the IEC104 protocol and the remote RTUs running Modbus protocol.

The router software supports two variations of the Modbus protocol:

- Modbus over serial: the Modbus devices are connected to the serial interface of the router
- Modbus TCP: the Modbus devices are located on the IP network reachable from the router

In the Modbus over serial variation, currently the router supports Modbus "RTU mode" frame format of the Modbus specification only.

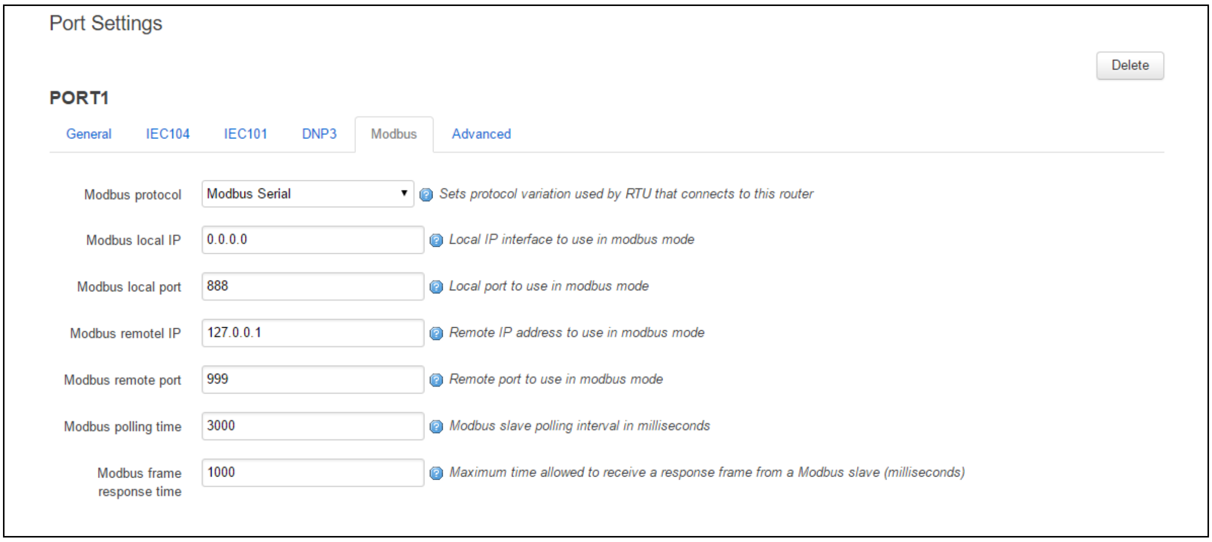


Figure 213: The IEC104 Gateway port MODBUS configuration page

Web Field / UCI / Package Option	Description
Web: Modbus Protocol UCI: iecd.<port>.modbus_protocol Opt: modbus_protocol	Defines the protocol variation used by RTU that connects to this router. **Option** | **Description** | **UCI** Modbus Serial | Modus over serial | modbus_serial Modbus TCP | Modbus over TCP | modbus_tcp
Web: Modbus local IP UCI: iecd.<port>.modbus_local_ip Opt: modbus_local_ip	Defines the local IP to use in Modbus mode. 0.0.0.0 Range
Web: Modbus local port UCI: iecd.<port>.modbus_local_port Opt: modbus_local_port	Defines the local port to use in Modbus mode. 888 Range
Web: Modbus remote IP UCI: iecd.<port>.modbus_remote_ip Opt: modbus_remote_ip	Defines the remote IP address. 127.0.0.1 Range
Web: Modbus remote port UCI: iecd.<port>.modbus_remote_port Opt: modbus_remote_port	Defines the remote port. 999 Range
Web: Modbus polling time UCI: iecd.<port>.modbus_polling_time Opt: modbus_polling_time	Defines the slave polling interval in milliseconds. 3000 | 3000 milliseconds Range
Web: Modbus frame response time UCI: iecd.<port>.modbus_resp_time Opt: modbus_resp_time	Defines the maximum time allowed to receive a response frame from a Modbus slave, in milliseconds. 1000 | 1000 milliseconds Range
Web: n/a UCI: iecd.<port>.modbus_dump_data Opt: modbus_dump_data	Enables RX/TX Hex dump. Range
Web: n/a UCI: iecd.<port>.modbus_trace_on Opt: modbus_trace_on	Enables Modbus protocol tracing Range
Web: n/a UCI: iecd.<port>.modbus_fsm_debug_on Opt: modbus_fsm_debug_on	Enables Modbus state machine debugging. Range

Table 153: Information table for IEC104 Gateway port MODBUS configuration

43.3.2.6 Port settings: advanced

In this section you can configure the advanced port settings.

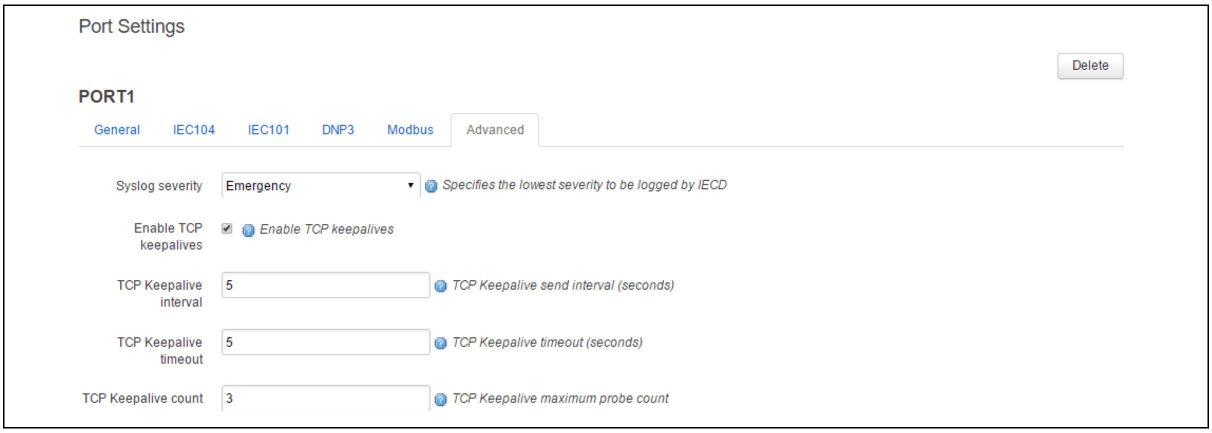


Figure 214: The IEC104 Gateway port advanced configuration page

Web Field / UCI / Package Option	Description
Web: Syslog severity UCI: iecd.<port>.loglevel Opt: loglevel	Defines the lowest severity used for logging events by iecd. 0 Emergency 1 Alert 2 Critical 3 Error 4 Warning 5 Notice 6 Informational 7 Debug
Web: Enable TCP keepalives UCI: iecd.<port>.tcp_keepalive_enabled Opt: tcp_keepalive_enabled	Defines whether to enable TCP keepalive. 1 Disabled. 0 Enabled.
Web: TCP Keepalive interval UCI: iecd.<port>.tcp_keepalive_interval Opt: tcp_keepalive_interval	Defines the TCP keepalive interval in seconds. 5 Seconds. Range
Web: TCP Keepalive timeout UCI: iecd.<port>.tcp_keepalive_timeout Opt: tcp_keepalive_timeout	Defines the TCP keepalive timeout in seconds. 5 Seconds. Range
Web: TCP Keepalive count UCI: iecd.<port>.tcp_keepalive_count Opt: tcp_keepalive_count	Defines the number of unanswered keepalives before terminating the TCP session. 3 Seconds. Range

Table 154: Information table for IEC104 Gateway port advanced configuration

43.3.3 IEC101 links

The following section defines the IEC101 slave links used in IEC101 conversion. Each link is defined by a **config iec101link** section block. There is a maximum of 32 links supported.

In IEC101 unbalanced mode all of these can be used.

However, as IEC101 balanced mode is used in a point to point scenario, it is assumed there will be only one outstation per serial port. Only the first link configured for that port will be used. Each peer - either controlling station (Master) or controlled station (RTU) can initiate communication in balanced mode.

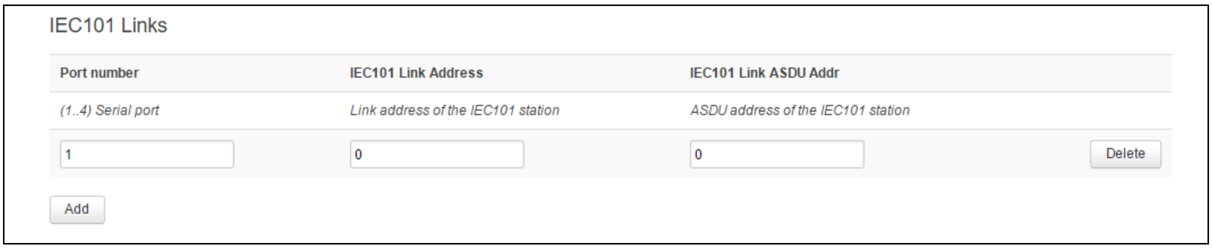


Table 155: IECD slave links configuration page

Web Field / UCI / Package Option	Description
Web: Port Number UCI: iecd.iec101link[x].portno Opt: portno	Defines the serial port number to which this point belongs. Range: 1 - 4
Web: IEC101 Link Address UCI: iecd.iec101link[x].address Opt: address	Defines the IEC101 station link address. 0 Range
Web: IEC101 Link ASDU Station UCI: iecd. iec101link[x].asduaddr Opt: asduaddr	Defines the IEC101 station common ASDU address. 0 Range

Table 156: Information table for IEC104 Gateway port IEC101 configuration

43.3.4 Points

IEC104 point mappings are used for DNP3 and Modbus conversion only.

The point mappings comprise the information necessary to perform conversion between each data variable (point) on the remote RTU and the corresponding variable in the IEC104 domain.

Modbus TCP requires a device route file (**/root/iecd/devroute.csv**) to map the point configuration to an IP address – see Modbus route file section below.

There is a maximum of 400 point mappings supported per serial port.

Figure 215: The IEC104 Gateway point mapping configuration page

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Port Number
UCI: iecd.point[x].portno
Opt: portno</td><td>Defines the port number to which this point belongs (1 to 4). This corresponds to the serial port number.<table><tr><td></td><td></td></tr><tr><td>Range</td><td>1 - 4</td></tr></table></td></tr>
<tr><td>Web: IEC104 Type ID
UCI: iecd.point[x] iec104_type_id
Opt: iec104_type_id</td><td>Defines the IEC104 type ID (data type). All types are defined in IEC-60870-5-104<table>
<tr><td>1</td><td>Single Point information.</td></tr>
<tr><td>2</td><td>Double Point information.</td></tr>
<tr><td>7</td><td>Bitstring 32 bits.</td></tr>
<tr><td>9</td><td>Measured normalized value short signed.</td></tr>
<tr><td>11</td><td>Measured scaled value short signed.</td></tr>
<tr><td>13</td><td>IEEE STD 754 = Short floating point number.</td></tr>
<tr><td>14</td><td>IEEE STD 754 = Short floating point number with time tag CP24Time2a.</td></tr>
<tr><td>15</td><td>Integrated totals, 32 bit signed integer.</td></tr>
<tr><td>20</td><td>Packed single point information with status change detection.</td></tr>
<tr><td>21</td><td>Measured normalized value short signed without quality descriptor.</td></tr>
<tr><td>30</td><td>Single-point information with time tag CP56Time2a</td></tr>
<tr><td>31</td><td>Double-point information with time tag CP56Time2a.</td></tr>
<tr><td>33</td><td>Bitstring of 32 bits with time tag CP56Time2a.</td></tr>
<tr><td>34</td><td>Measured normalized value short signed time tag CP56Time2a.</td></tr>
<tr><td>35</td><td>Measured value, scaled value with time tag CP56Time2a.</td></tr>
<tr><td>36</td><td>Measured value, short floating point number with time tag CP56Time2a.</td></tr>
<tr><td>37</td><td>Integrated totals with time tag CP56Time2a.</td></tr>
</table></td></tr>
<tr><td>Web: IEC104 IOA
UCI: iecd.point[x].iec104_ioa
Opt: iec104_ioa</td><td>Defines IEC104 information object address. This is how remote IEC104 SCADA master knows one point from another.<table><tr><td></td><td></td></tr><tr><td>Range</td><td>1 - 116777215</td></tr></table></td></tr>
<tr><td>Web: IEC101 IOA
UCI: iecd.point[x].iec101_ioa
Opt: iec101_ioa</td><td>Defines IEC101 information object address.<table><tr><td></td><td></td></tr><tr><td>Range</td><td>1 - 116777215</td></tr></table></td></tr>
<tr><td>Web: Device Addr
UCI: iecd.point[x].devaddr
Opt: devaddr</td><td>Defines the Modbus device address of the RTU (Modbus slave address). Used for identifying the point mapping to IP address in the device route file for Modbus TCP.
This is not used in DNP3 mode.<table><tr><td></td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Group
UCI: iecd.point[x].group
Opt: group</td><td>For DNP3 this defines the DNP3 group number to which this data point maps to.<table><tr><td></td><td></td></tr><tr><td>Range</td><td>0 – 255</td></tr></table>
For Modbus, this defines the Modbus data type.<table><tr><td>0</td><td>Discreet input.</td></tr><tr><td>1</td><td>Input register.</td></tr><tr><td>2</td><td>Holding register.</td></tr><tr><td>3</td><td>Coil.</td></tr></table></td></tr>
</table>

Web: Index UCI: iecd.point[x].index Opt: index	For DNP3, this defines the DNP3 point index. For Modbus, this defines the Modbus data index (point number). Range: 0 - 65535
Web: n/a UCI: iecd.point[x].dword Opt: dword	Defines the DWORD type. Relevant for Modbus data types IR (input registers) and HR (holding registers). 0: Data point is treated as 16 bit wide. 1: Data point is treated as 32 bit wide (two consecutive 16 bit registers are read from the Modbus device).

Table 157: Information table for IEC104 Gateway point mapping configuration

43.3.4.1 MODBUS device route file

If the configured MODBUS protocol variation is Modbus TCP, then the device route file at **/root/iecd/devroute.csv** is used to map the device address (`iecd.point[x].devaddr`) from the point mapping to the remote IP address of the Modbus TCP slave device.

The devroute.csv file entries will have the following format:

```
<Modbus device addr>, <IP address>
```

For example, for the point mapping file:

```
config point
        option portno 1
        option iec104_type_id 30
        option iec104_ioa  64213
        option devaddr 1
        option group 0
        option index 2
```

For the devroute.csv entry:

```
1,192.168.0.106
```

43.4 IEC104 gateway configuration using command line

IEC104 Gateway uses the iecd package /etc/config/iecd.

You can configure multiple port, iec101link and points sections.

By default, IEC104 Gateway port instances are named port, it is identified by `@port` followed by the port position in the package as a number. For example, for the first port in the package using UCI:

```
iecd.@port[0]=port
iecd.@port[0].enable=1
```

Or using package options:

```
config port
      option enable ‘1’
```

By default, all IEC104 Gateway IEC101 link instances are named iec101link, it is identified by @iec101link followed by the link position in the package as a number. For example, for the first IEC101 link in the package using UCI:

```
iecd.@iec101link[0]=iec101link
iecd.@iec101link[0].portno=1
```

Or using package options:

```
config iec101link
        option portno '1'
```

By default, all IEC104 Gateway point instances are named point, it is identified by @point followed by the point position in the package as a number. For example, for the first point in the package using UCI:

```
iecd.@point[0]=point
iecd.@point[0].portno=1
```

Or using package options:

```
config point
        option portno '1'
```

43.4.1 IEC104 to IEC101 conversion (balanced or unbalanced)

The following example shows IEC104 to IEC101 unbalanced conversion with one IEC101 link.

To configure IEC104 to IEC101 balanced conversion set option iec101_mode to **balanced**.

43.4.1.1 IEC104 to IEC101 using uci

```
root@VA_router:~# uci show iecd
iecd.main=iecd
iecd.main.enable=1
iecd.port1=port
iecd.port1.enable=1
iecd.port1.loglevel=5
iecd.port1.tcp_keepalive_enabled=1
iecd.port1.tcp_keepalive_interval=5
iecd.port1.tcp_keepalive_timeout=5
```

```
iecd.port1.tcp_keepalive_count=3
iecd.port1.tcp_user_timeout=20000
iecd.port1.master_protocol=iec101
iecd.port1.slave_protocol=iec104
iecd.port1.ioa_offset=0
iecd.port1.pointmap_file=/root/iecd/iecd_points1.csv
iecd.port1.iec104_local_ip=0.0.0.0
iecd.port1.iec104_local_tcpport=2404
iecd.port1.iec104_k=12
iecd.port1.iec104_w=9
iecd.port1.iec104_t2=10000
iecd.port1.iec104_gi_resp_time=200
iecd.port1.iec104_txq_size=128
iecd.port1.iec104_sync_time=1
iecd.port1.iec104_time_tagged_cmds=0
iecd.port1.iec104_cmd_delay_time=5000
iecd.port1.iec104_fsm_debug_on=0
iecd.port1.iec104_dump_data=0
iecd.port1.iec104_trace_on=0

#IEC101 conversion options
iecd.port1.iec101_target_ip=127.0.0.1
iecd.port1.iec101_target_tcpport=999
iecd.port1.iec101_mode=unbalanced     #balanced or unbalanced
iecd.port1.iec101_cot_tx_length=1
iecd.port1.iec101_cot_source_octet=0
iecd.port1.iec101_asdu_addrlen=1
iecd.port1.iec101_info_obj_addrlen=2
iecd.port1.iec101_data_polling_time=500
iecd.port1.iec101_ack_delay=0
iecd.port1.iec101_link_addrlen=1
iecd.port1.iec101_frame_rsp_time=2000
iecd.port1.iec101_max_tx_retry=3
iecd.port1.iec101_txq_size=128
iecd.port1.iec101_send_spont_delay_acq=1
iecd.port1.iec101_fsm_debug_on=0
iecd.port1.iec101_dump_data=0
```

```
iecd.port1.iec101_trace_on=0

# The following section defines IEC101 slave links used in IEC101
unbalanced mode on each link is defined by a config block 'config
iec101link'
# To add more links repeat the section block for each added link.
# Maximum 32 links are supported
iecd.@iec101link[0]=iec101link
iecd.@iec101link[0].portno=1
iecd.@iec101link[0].address=6
iecd.@iec101link[0].asduaddr=6

#No data point mappings for IEC104 to IEC101 conversion
```

43.4.1.2 IEC104 to IEC101 using package options

```
root@VA_router:~# uci export iecd
package iecd

config iecd 'main'
        option enable '1'

config port 'port1'
        option enable '1'
        option loglevel '5'
        option tcp_keepalive_enabled '1'
        option tcp_keepalive_interval '5'
        option tcp_keepalive_timeout '5'
        option tcp_keepalive_count '3'
        option tcp_user_timeout '20000'
        option master_protocol 'iec101'
        option slave_protocol 'iec104'
        option ioa_offset '0'
        option pointmap_file '/root/iecd/iecd_points1.csv'
        option iec104_local_ip '0.0.0.0'
        option iec104_local_tcpport '2404'
        option iec104_k '12'
        option iec104_w '9'
        option iec104_t2 '10000'
```

```
        option iec104_gi_resp_time '200'
        option iec104_txq_size '128'
        option iec104_sync_time '1'
        option iec104_time_tagged_cmds '0'
        option iec104_cmd_delay_time '5000'
        option iec104_fsm_debug_on '0'
        option iec104_dump_data '0'
        option iec104_trace_on '0'

        #IEC101 conversion options
        option iec101_target_ip '127.0.0.1'
        option iec101_target_tcpport '999'
        option iec101_mode 'unbalanced'       #balanced or unbalanced
        option iec101_cot_tx_length '1'
        option iec101_cot_source_octet '0'
        option iec101_asdu_addrlen '1'
        option iec101_info_obj_addrlen '2'
        option iec101_data_polling_time '500'
        option iec101_ack_delay '0'
        option iec101_link_addrlen '1'
        option iec101_frame_rsp_time '2000'
        option iec101_max_tx_retry '3'
        option iec101_txq_size '128'
        option iec101_send_spont_delay_acq '1'
        option iec101_fsm_debug_on '0'
        option iec101_dump_data '0'
        option iec101_trace_on '0'

# The following section defines IEC101 slave links used in IEC101
unbalanced mode on
# Each link is defined by a config block 'config iec101link'
# To add more links repeat the section block for each added link. To remove
links, s
# Maximum 32 links are supported
#
# Definition of options within the section block:
# portno - port number to which this point belongs (1 to 4)
```

```
# address - IEC101 slave link address
# asduaddr IEC101 slave common ASDU address

config iec101link
        option portno 1
        option address 6
        option asduaddr 6

#No data point mappings for IEC104 to IEC101 conversion
```

43.4.2 IEC104 to DNP3 conversion

The following example shows definition of two conversion points. The config point section should be repeated for each point to be defined.

43.4.2.1 IEC104 to DNP3 conversion using uci

```
root@VA_router:~# uci show iecd
iecd.main=iecd
iecd.main.enable=1
iecd.port1=port
iecd.port1.enable=1
iecd.port1.loglevel=5
iecd.port1.tcp_keepalive_enabled=1
iecd.port1.tcp_keepalive_interval=5
iecd.port1.tcp_keepalive_timeout=5
iecd.port1.tcp_keepalive_count=3
iecd.port1.tcp_user_timeout=20000
iecd.port1.master_protocol=dnp3
iecd.port1.slave_protocol=iec104
iecd.port1.ioa_offset=0
iecd.port1.pointmap_file=/root/iecd/iecd_points1.csv
iecd.port1.iec104_local_ip=0.0.0.0
iecd.port1.iec104_local_tcpport=2404
iecd.port1.iec104_k=12
iecd.port1.iec104_w=9
iecd.port1.iec104_t2=10000
iecd.port1.iec104_gi_resp_time=200
iecd.port1.iec104_txq_size=128
```

```
iecd.port1.iec104_sync_time=1
iecd.port1.iec104_time_tagged_cmds=0
iecd.port1.iec104_cmd_delay_time=5000
iecd.port1.iec104_fsm_debug_on=0
iecd.port1.iec104_dump_data=0
iecd.port1.iec104_trace_on=0
iecd.port1.iec101_cot_source_octet=0

#DNP3 conversion options
iecd.port1.dnp3_target_ip=127.0.0.1
iecd.port1.dnp3_target_tcpport=999
iecd.port1.dnp3_dl_srcaddr=3
iecd.port1.dnp3_dl_dstaddr=4
iecd.port1.dnp3_dl_cfrm_user_data=0
iecd.port1.dnp3_dl_keep_alive_int=15000
iecd.port1.dnp3_dl_frame_rsp_time=1500
iecd.port1.dnp3_dl_max_tx_retry=3
iecd.port1.dnp3_dl_utxq_size=128
iecd.port1.dnp3_dl_ctxq_size=128
iecd.port1.dnp3_app_read_attr=0
iecd.port1.dnp3_app_unsol_enable=0
iecd.port1.dnp3_app_poll_time=30000
iecd.port1.dnp3_app_firstpoll_delay=5000
iecd.port1.dnp3_app_evpoll_time=3000
iecd.port1.dnp3_app_frag_rx_time=10000
iecd.port1.dnp3_app_sync_time=1
iecd.port1.dnp3_app_txq_size=64
iecd.port1.dnp3_app_output_mode=0
iecd.port1.dnp3_app_evpoll_mode=0
iecd.port1.dnp3_fsm_debug_on=0
iecd.port1.dnp3_object_parser_debug_on=0
iecd.port1.dnp3_dump_data=0
iecd.port1.dnp3_trace_on=0

#DNP3 data point mappings
iecd.@point[0]=point
iecd.@point[0].portno=1
```

```
iecd.@point[0].iec104_type_id=1
iecd.@point[0].iec104_ioa=1
iecd.@point[0].devaddr=1
iecd.@point[0].group=1
iecd.@point[0].index=0
iecd.@point[1]=point
iecd.@point[1].portno=1
iecd.@point[1].iec104_type_id=1
iecd.@point[1].iec104_ioa=2
iecd.@point[1].devaddr=1
iecd.@point[1].group=1
iecd.@point[1].index=39
```

43.4.2.2 IEC104 to DNP3 conversion using package options

```
root@VA_router:~# uci export iecd
package iecd

config iecd 'main'
        option enable '1'

config port 'port1'
        option enable '1'
        option loglevel '5'
        option tcp_keepalive_enabled '1'
        option tcp_keepalive_interval '5'
        option tcp_keepalive_timeout '5'
        option tcp_keepalive_count '3'
        option tcp_user_timeout '20000'
        option master_protocol 'dnp3'
        option slave_protocol 'iec104'
        option ioa_offset '0'
        option pointmap_file '/root/iecd/iecd_points1.csv'
        option iec104_local_ip '0.0.0.0'
        option iec104_local_tcpport '2404'
        option iec104_k '12'
        option iec104_w '9'
        option iec104_t2 '10000'
```

```
        option iec104_gi_resp_time '200'
        option iec104_txq_size '128'
        option iec104_sync_time '1'
        option iec104_time_tagged_cmds '0'
        option iec104_cmd_delay_time '5000'
        option iec104_fsm_debug_on '0'
        option iec104_dump_data '0'
        option iec104_trace_on '0'
        option iec101_cot_source_octet '0'

        #DNP3 conversion options
        option dnp3_target_ip '127.0.0.1'
        option dnp3_target_tcpport '999'
        option dnp3_dl_srcaddr '3'
        option dnp3_dl_dstaddr '4'
        option dnp3_dl_cfrm_user_data '0'
        option dnp3_dl_keep_alive_int '15000'
        option dnp3_dl_frame_rsp_time '1500'
        option dnp3_dl_max_tx_retry '3'
        option dnp3_dl_utxq_size '128'
        option dnp3_dl_ctxq_size '128'
        option dnp3_app_read_attr '0'
        option dnp3_app_unsol_enable '0'
        option dnp3_app_poll_time '30000'
        option dnp3_app_firstpoll_delay '5000'
        option dnp3_app_evpoll_time '3000'
        option dnp3_app_frag_rx_time '10000'
        option dnp3_app_sync_time '1'
        option dnp3_app_txq_size '64'
        option dnp3_app_output_mode '0'
        option dnp3_app_evpoll_mode '0'
        option dnp3_fsm_debug_on '0'
        option dnp3_object_parser_debug_on '0'
        option dnp3_dump_data '0'
        option dnp3_trace_on '0'

config point
```

```
        option portno '1'
        option iec104_type_id '1'
        option iec104_ioa '1'
        option devaddr '1'
        option group '1'
        option index '0'

config point
        option portno '1'
        option iec104_type_id '1'
        option iec104_ioa '2'
        option devaddr '1'
        option group '1'
        option index '39'
```

43.4.3 IEC104 to Modbus conversion

The following example shows IEC104 to Modbus over serial.

To configure Modbus TCP, set `option modbus_protocol` to **`modbus_tcp`**.

When configuring Modbus TCP, then device route file at **`/root/iecd/devroute.csv`** must be configured to map the device address **`option devaddr`** from the point mapping to the remote IP address of the Modbus TCP slave device.

The devroute.csv file entries will have the following format:

`<Modbus device addr>, <IP address>`

For example, for the point mapping file:

```
config point
        option portno 1
        option iec104_type_id 30
        option iec104_ioa  64213
        option devaddr 1
        option group 0
        option index 2
```

For the devroute.csv entry:

```
1,192.168.0.106
```

43.4.3.1 IEC104 to modbus using uci

```
root@VA_router:~# uci show iecd
iecd.main=iecd
iecd.main.enable=1
iecd.port1=port
iecd.port1.enable=1
iecd.port1.loglevel=5
iecd.port1.tcp_keepalive_enabled=1
iecd.port1.tcp_keepalive_interval=5
iecd.port1.tcp_keepalive_timeout=5
iecd.port1.tcp_keepalive_count=3
iecd.port1.tcp_user_timeout=20000
iecd.port1.master_protocol=modbus
iecd.port1.slave_protocol=iec104
iecd.port1.ioa_offset=0
iecd.port1.pointmap_file=/root/iecd/iecd_points1.csv
iecd.port1.iec104_local_ip=0.0.0.0
iecd.port1.iec104_local_tcpport=2404
iecd.port1.iec104_k=12
iecd.port1.iec104_w=9
iecd.port1.iec104_t2=10000
iecd.port1.iec104_gi_resp_time=200
iecd.port1.iec104_txq_size=128
iecd.port1.iec104_sync_time=1
iecd.port1.iec104_time_tagged_cmds=0
iecd.port1.iec104_cmd_delay_time=5000
iecd.port1.iec104_fsm_debug_on=0
iecd.port1.iec104_dump_data=0
iecd.port1.iec104_trace_on=0
iecd.port1.iec101_cot_source_octet=0

#Modbus conversion options
iecd.port1.modbus_protocol=modbus_serial
iecd.port1.modbus_local_ip=0.0.0.0
iecd.port1.modbus_local_port=888
iecd.port1.modbus_remote_ip=127.0.0.1
iecd.port1.modbus_remote_port=999
```

```
iecd.port1.modbus_polling_time=3000
iecd.port1.modbus_resp_time=1000
iecd.port1.modbus_dump_data=0
iecd.port1.modbus_trace_on=0
iecd.port1.modbus_fsm_debug_on=0

#Modbus data point mappings
iecd.@point[0]=point
iecd.@point[0].portno=1
iecd.@point[0].iec104_type_id=36
iecd.@point[0].iec104_ioa=6620161
iecd.@point[0].iec101_ioa=0
iecd.@point[0].devaddr=11
iecd.@point[0].group=1
iecd.@point[0].index=18459
iecd.@point[0].dword=1
iecd.@point[1]=point
iecd.@point[1].portno=1
iecd.@point[1].iec104_type_id=36
iecd.@point[1].iec104_ioa=6620162
iecd.@point[1].iec101_ioa=0
iecd.@point[1].devaddr=11
iecd.@point[1].group=1
iecd.@point[1].index=18461
iecd.@point[1].dword=1
```

43.4.3.2 IEC104 to modbus using package options

```
root@VA_router:~# uci export iecd
package iecd

config iecd 'main'
        option enable '1'

config port 'port1'
        option enable '1'
        option loglevel '5'
        option tcp_keepalive_enabled '1'
```

```
        option tcp_keepalive_interval '5'
        option tcp_keepalive_timeout '5'
        option tcp_keepalive_count '3'
        option tcp_user_timeout '20000'
        option master_protocol 'modbus'
        option slave_protocol 'iec104'
        option ioa_offset '0'
        option pointmap_file '/root/iecd/iecd_points1.csv'
        option iec104_local_ip '0.0.0.0'
        option iec104_local_tcpport '2404'
        option iec104_k '12'
        option iec104_w '9'
        option iec104_t2 '10000'
        option iec104_gi_resp_time '200'
        option iec104_txq_size '128'
        option iec104_sync_time '1'
        option iec104_time_tagged_cmds '0'
        option iec104_cmd_delay_time '5000'
        option iec104_fsm_debug_on '0'
        option iec104_dump_data '0'
        option iec104_trace_on '0'
        option iec101_cot_source_octet '0'

        #Modbus conversion options
        option modbus_protocol 'modbus_serial'
        option modbus_local_ip '0.0.0.0'
        option modbus_local_port '888'
        option modbus_remote_ip '127.0.0.1'
        option modbus_remote_port '999'
        option modbus_polling_time '3000'
        option modbus_resp_time '1000'
        option modbus_dump_data '0'
        option modbus_trace_on '0'
        option modbus_fsm_debug_on '0'

config point
        option portno '1'
```

```
        option iec104_type_id '36'
        option iec104_ioa '6620161'
        option iec101_ioa '0'
        option devaddr '11'
        option group '1'
        option index '18459'
        option dword '1'

config point
        option portno '1'
        option iec104_type_id '36'
        option iec104_ioa '6620162'
        option iec101_ioa '0'
        option devaddr '11'
        option group '1'
        option index '18461'
        option dword '1'
```

43.5 Configuring the terminal server

The terminal server is used to control the data from the serial port over the IP network.

The terminal server configuration can be found at **Services -> Terminal Server.** The Terminal Server Configuration page appears. You must configure two main sections: Main Settings and Port Settings.

The terminal server for IEC104 to each of the RTU protocol conversions differ only slightly. This section shows the command line options for configuring the terminal server for IEC104 conversion.

See the terminal server user manual section for more detailed information on web configuration and option values.

43.5.1 Configuring the terminal server for IEC104 to IEC101

43.5.1.1 Configuring IEC104 to IEC101 ucing uci

```
root@VA_router:~# uci show tservd
tservd.main=tservd
tservd.main.enable=1
tservd.main.debug_ev_enable=0
tservd.main.log_severity=5
tservd.main.debug_rx_tx_enable=0
```

```
tservd.port1=port
tservd.port1.enable=1
tservd.port1.devName=/dev/ttySC0
tservd.port1.ip_port1=0
tservd.port1.ip_port2=0
tservd.port1.remote_ip1=0.0.0.0
tservd.port1.remote_ip2=0.0.0.0
tservd.port1.tcp_always_on=1
tservd.port1.close_tcp_on_dsr=0
tservd.port1.tty_always_open=1
tservd.port1.fwd_timeout=0
tservd.port1.fwd_timer_mode=idle
tservd.port1.fwd_buffer_size=1
tservd.port1.sfwd_buffer_size=0
tservd.port1.sfwd_timeout=0
tservd.port1.sfwd_timer_mode=idle
tservd.port1.speed=9600
tservd.port1.wsize=8
tservd.port1.parity=1
tservd.port1.stops=1
tservd.port1.fc_mode=0
tservd.port1.disc_time_ms=5000
tservd.port1.server_mode=1
tservd.port1.proxy_mode=0
tservd.port1.local_ip=0.0.0.0
tservd.port1.listen_port=999
tservd.port1.udpMode=0
tservd.port1.udpLocalPort=0
tservd.port1.udpRemotePort=0
tservd.port1.udpKaIntervalMs=0
tservd.port1.udpKaCount=3
tservd.port1.serial_mode_gpio_control=1
tservd.port1.tcp_nodelay=1
tservd.port1.portmode=rs232
```

43.5.1.2 Configuring IEC104 to IEC101 ucing package options

```
root@VA_router:~# uci export tservd
package tservd

config tservd main
    # set to 1 to enable terminal server
    option enable 1

    # enables detailed debug logging (state transitions, data transfer etc)
    option debug_ev_enable 0

    # sets syslog level (0 to 7), default is 6
    option log_severity 5

    option debug_rx_tx_enable 0

config port 'port1'
    # enables this port
    option enable 1

    # serial device name
    option devName '/dev/ttySC0'

    # destination peer port IP number (two number for failover)
    option ip_port1 0
    option ip_port2 0

    # destination peer ip address (two addresses for failover)
    option remote_ip1 '0.0.0.0'
    option remote_ip2 '0.0.0.0'

    # keep TCP session always connected
    option tcp_always_on 1

    # close TCP session on detection of DSR signal low
    option close_tcp_on_dsr 0
```

```
    # keep serial port always open (if option not present, default is 0)
    option tty_always_open 1

    # Forwarding timeout in milliseconds (serial to network)
    option fwd_timeout 0

    # Forwarding timer mode (serial to network), 'idle'=timer re-started on
each received data,
    # 'aging'=timer started on first rx
    option fwd_timer_mode 'idle'

    # Forwarding buffer size (serial to network)
    option fwd_buffer_size 1

    # Forwarding buffer size (network to serial), 0=use maximum possible
network rx buffer size
    option sfwd_buffer_size 0

    # Forwarding timeout in milliseconds (network to serial), 0=forward to
serial immediately
    option sfwd_timeout 0

    # Forwarding timer mode (network to serial), 'idle'=timer re-started on
each received data,
    # 'aging'=timer started on first rx
    option sfwd_timer_mode 'idle'

    # serial device speed in baud
    option speed 9600

    # serial device word size (5,6,7,8)
    option wsize 8

    # serial device parity (0=none, 1=even, 2=odd)
    option parity 1

    # serial device number of stop bits (1 or 2)
```

```
    option stops 1

    # serial flow control mode (0=none, 1=RTS CTS, 2=XONXOFF)
    option fc_mode 0

    # time in milliseconds to start re-connecting after setting DTR low
    option disc_time_ms 5000

    # TCP server mode
    option server_mode 1

    # Proxy mode (off by default)
    option proxy_mode 0

    # Local IP address to listen on (0.0.0.0=listen on any interface)
    option local_ip '0.0.0.0'

    # TCP listen port for server mode
    option listen_port 999

    # UDP mode
    option udpMode 0

    # UDP local port UDP mode
    option udpLocalPort 0

    # UDP port for UDP mode
    option udpRemotePort 0

    # If set to non zero, send empty UDP packets every this many
milliseconds to remote peer
    option udpKaIntervalMs 0

    # Max number of consecutive remote UDP keepalive missed (not received)
before UDP
    # session considered broken
    option udpKaCount 3
```

```
    option serial_mode_gpio_control 1
    option tcp_nodelay 1

    # rs232 - RS-232 mode, rs485hdx - rs485 2 wire half duplex mode in
which transmitter drives
    # RTS. rs485fdx - RS485 4 wire full duplex mode. 'v23' - using V.23
leased line card driver.
    # x21 - use USB serial card in sync mode
    option portmode 'rs232'
```

43.5.2 Configuring the terminal server for IEC104 to DNP3

The terminal server server configuration for IEC104 to DNP3 is the same as for IEC104 to IEC101 except for serial device parity which is set to **none**.

Parity setting using uci:

```
    tservd.port1.parity=1
```

Parity setting using package options:

```
    option parity 0
```

43.5.3 Configuring the terminal server for IEC104 to Modbus over serial

The terminal server server is only used for IEC104 to Modbus over serial. It is not used for Modbus over TCP.

The following options necessary for IEC104 to Modbus are listed below (for the first serial port only).

43.5.3.1 IEC104 to Modbus over serial using uci

```
root@VA_router:~# uci show tservd
tservd.main=tservd
tservd.main.enable=1
tservd.main.debug_ev_enable=0
tservd.main.log_severity=5
tservd.main.debug_rx_tx_enable=0
tservd.port1=port
tservd.port1.enable=1
tservd.port1.devName=/dev/ttySC0
tservd.port1.ip_port1=999
tservd.port1.ip_port2=999
```

```
tservd.port1.remote_ip1=127.0.0.1
tservd.port1.remote_ip2=127.0.0.1
tservd.port1.tcp_always_on=1
tservd.port1.close_tcp_on_dsr=0
tservd.port1.tty_always_open=1
tservd.port1.fwd_timeout=10
tservd.port1.fwd_timer_mode=idle
tservd.port1.fwd_buffer_size=300
tservd.port1.sfwd_buffer_size=0
tservd.port1.sfwd_timeout=0
tservd.port1.sfwd_timer_mode=idle
tservd.port1.speed=19200
tservd.port1.wsize=8
tservd.port1.parity=1
tservd.port1.stops=1
tservd.port1.fc_mode=0
tservd.port1.disc_time_ms=5000
tservd.port1.server_mode=1
tservd.port1.proxy_mode=0
tservd.port1.local_ip=0.0.0.0
tservd.port1.listen_port=999
tservd.port1.udpMode=1
tservd.port1.udpLocalPort=999
tservd.port1.udpRemotePort=888
tservd.port1.udpKaIntervalMs=0
tservd.port1.udpKaCount=3
tservd.port1.serial_mode_gpio_control=1
tservd.port1.portmode=rs232
```

43.5.3.2 IEC104 to Modbus over serial using package options

```
root@VA_router:~# uci export tservd
package tservd

config tservd main
    # set to 1 to enable terminal server
    option enable 1
```

```
    # enables detailed debug logging (state transisions, data transfer etc)
    option debug_ev_enable 0

    # sets syslog level (0 to 7), default is 6
    option log_severity 5

    option debug_rx_tx_enable 0

config port 'port1'
    # enables this port
    option enable 1

    # serial device name
    option devName '/dev/ttySC0'

    # destination peer port IP number (two number for failover)
    option ip_port1 999
    option ip_port2 999

    # destination peer ip address (two addresses for failover)
    option remote_ip1 '127.0.0.1'
    option remote_ip2 '127.0.0.1'

    # keep TCP session always connected
    option tcp_always_on 1

    # close TCP session on detection of DSR signal low
    option close_tcp_on_dsr 0

    # keep serial port always open (if option not present, default is 0)
    option tty_always_open 1

    # Forwarding timeout in milliseconds (serial to network)
    option fwd_timeout 10

    # Forwarding timer mode (serial to network), 'idle'=timer re-started on
each received data,
```

```
    # 'aging'=timer started on first rx
    option fwd_timer_mode 'idle'

    # Forwarding buffer size (serial to network)
    option fwd_buffer_size 300

    # Forwarding buffer size (network to serial), 0=use maximum possible
network rx buffer size
    option sfwd_buffer_size 0

    # Forwarding timeout in milliseconds (network to serial), 0=forward to
serial immediately
    option sfwd_timeout 0

    # Forwarding timer mode (network to serial), 'idle'=timer re-started on
each received data,
    # 'aging'=timer started on first rx
    option sfwd_timer_mode 'idle'

    # serial device speed in baud
    option speed 19200

    # serial device word size (5,6,7,8)
    option wsize 8

    # serial device parity (0=none, 1=even, 2=odd)
    option parity 1

    # serial device number of stop bits (1 or 2)
    option stops 1

    # serial flow control mode (0=none, 1=RTS CTS, 2=XONXOFF)
    option fc_mode 0

    # time in milliseconds to start re-connecting after setting DTR low
    option disc_time_ms 5000
```

```
    # TCP server mode
    option server_mode 1

    # Proxy mode (off by default)
    option proxy_mode 0

    # Local IP address to listen on (0.0.0.0=listen on any interface)
    option local_ip '0.0.0.0'

    # TCP listen port for server mode
    option listen_port 999

    # UDP mode
    option udpMode 1

    # UDP local port UDP mode
    option udpLocalPort 999

    # UDP port for UDP mode
    option udpRemotePort 888

    # If set to non zero, send empty UDP packets every this many
milliseconds to remote peer
    option udpKaIntervalMs 0

    # Max number of consecutive remote UDP keepalive missed (not received)
before UDP
    # session considered broken
    option udpKaCount 3

    option serial_mode_gpio_control 1

    # rs232 - RS-232 mode, rs485hdx - rs485 2 wire half duplex mode in
which transmitter drives
    # RTS. rs485fdx - RS485 4 wire full duplex mode. 'v23' - using V.23
leased line card driver.
    # x21 - use USB serial card in sync mode
```

```
   option portmode 'rs232'
```

43.6 Configuring IEC61850 to IEC101 conversion

The IEC61850 to IEC101-unbalanced conversion feature of the router allows converting commands in the control direction and the responses and process data in the monitor direction between the SCADA Master running the IEC61850 protocol and the remote RTUs running IEC101 protocol in unbalanced mode over serial interface.

In IEC101 unbalanced mode, the router supports communication of up to 32 IEC101 slaves connected onto the same serial interface.

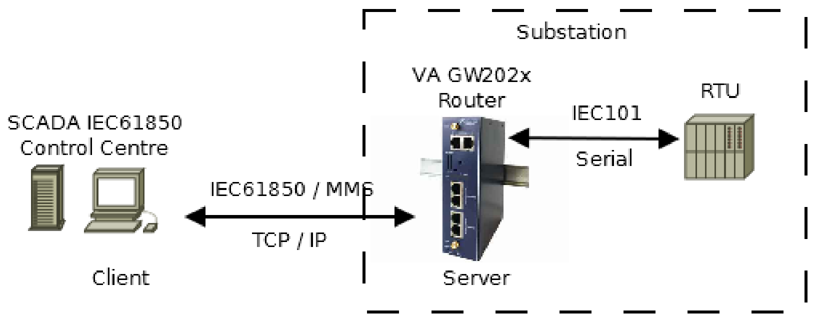


Figure 216: Example of IEC61850 to IEC101 conversion scenario

The IEC104 Gateway and terminal server are used for IEC61850 to IEC101 conversion, as in the other protocol conversions however the IEC62850 options are currently not available via the web interface. The following section details command line configuration.

Web Field / UCI / Package Option	Description
iecd port config section	
Web: n/a UCI: iecd.<port>.slave_protocol Opt: slave_protocol	Defines what protocol the SCADA control centre is using to connect to this gateway. iec104 | IC104 iec61850 | IEC61850
Web: n/a UCI: iecd.<port>.iec61850_local_ip Opt: iec61850_local_ip	Defines the local IP address this IEC61850 peer binds to.
Web: n/a UCI: iecd.<port>.iec61850_local_tcpport Opt: iec61850_local_tcpport	Defines the local TCP port this IEC104 peer listens on. 2404 | Range | 1 - 65535
iecd point config section	
Web: n/a UCI: iecd.point[x].iec61850_ld Opt: iec61850_ld	Defines the IEC61850 logical device name. For example `option iec61850_ld ‘SENSORS’` Range | 0 – 32 chars

Web: n/a UCI: iecd.point[x].iec61850_ln Opt: iec61850_ln	Defines the IEC61850 logical node name. For example `option iec61850_ln 'LLN0'` Range: 0 – 32 chars
Web: n/a UCI: iecd.point[x].iec61850_do Opt: iec61850_do	Defines the IEC61850 data object name. For example: `option iec61850_do 'SPS01'` Range: 0 – 32 chars
Web: n/a UCI: iecd.point[x] iec101_type_id Opt: iec101_type_id	Defines the IEC104 type ID (data type). For example: 1 – Single Point Information 2 – Double Point Information All types are defined in IEC-60870-5-101. Range:
Web: IEC101 IOA UCI: iecd.point[x].iec101_ioa Opt: iec101_ioa	Defines IEC101 information object address. 1: Single Point Information Range: 1 - 16777215

Table 158: Information table for IEC61850 specific configuration

43.6.1 Relation of IEC101 data types to IEC61850 data types

Supported data type combinations are listed below:

option iec101_type_id (IEC101 explanation)	option iec61850_do (IEC61850 explanation)	IEC101 point R/W	IEC61850 point R/W
'1' SPI (Single Point Information)	'SPS' Single-point status	read only	read only
'1' SPI (Single Point Information)	'SPC' Controllable single-point	read-write	read-write
'1' SPI (Single Point Information)	'SPG' Single point setting	--	write only
'3' DPI (Double Point Information)	'DPS' Double-point status	read only	read only
'3' DPI (Double Point Information)	'DPC' Controllable double-point	read-write	read-write
'11' Measured value, scaled value short signed	'INS' Integer status	read only	read only
'11' Measured value, scaled value short signed	'STV' Status value	read only	read only
'11' Measured value, scaled value short signed	'ENS' Enumerated Status	read only	read only

'11' Measured value, scaled value short signed	'ENC' Controllable enumerated status	read-write	read-write
'11' Measured value, scaled value short signed	'ENG' Enumerated status setting	--	write only
'11' Measured value, scaled value short signed	'INC' Controllable integer status	read-write	read-write
'11' Measured value, scaled value short signed	'CMD' Command	--	write only
'11' Measured value, scaled value short signed	'ING' Integer status setting	--	write only
'11' Measured value, scaled value short signed	'MV' Measured Value	read only	read only
'13' Measured value, short floating point number	'MV' Measured Value	read only	read only
'13' Measured value, short floating point number	'APC' Controllable analog set point	wead-write	read-write
'13' Measured value, short floating point number	'SPV' Set point value	--	write only
'13' Measured value, short floating point number	'ASG' Analog setting	--	write only

Table 159: IEC101 data types to IEC61850 data types

43.6.2 IEC61850 to IEC101 conversion using the command line

Two configuration packages must be configured

- **iecd** for the IEC104 Gateway; **/etc/config/iecd**
- **tservd** for the Termianl Server; **/etc/config/tservd**

The IECD point mappings comprise the information necessary to perform conversion between each data variable (point) on the remote IEC101 RTU and the corresponding variable in the IEC61850 domain.

In the IEC61850 domain, the data points are identified by unique textual names in the general form.

```
LogicalDevice/LogicalNode/DataObject, e.g. 'SENSORS/LLN0/SPSS01'
```

In the IEC101 domain, the data points are identified by type ID and information object address (IOA). For example:

```
Type ID 1 (Single Point Information), IOA 3
```

Each point is defined at the end of the **/etc/config/iecd** configuration file by a **config point** section block. A sample definition of two points is given below. The example configuration shows the points of IEC61850 domain belonging to logical device 'SENSORS' (option iec61850_ld), logical node 'LLN0' (option iec61850_ln) with data objects (option iec61850_do) 'SPS01' and 'SPS02' (single point status) mapping to IEC101 data points of type id 1 (M_SP_NA_1 – Single Point Information) and having IEC101 Information Object Addresses (option iec101_ioa) 5 and 6

To add more points repeat the section block for each added point.

To remove points, simply remove the section block.

Note: Maximum 400 points supported per serial port.

43.6.2.1 IEC61850 to IEC101 conversion using uci

```
root@VA_router:~# uci show iecd
iecd.main=iecd
iecd.main.enable=1
iecd.port1=port
iecd.port1.enable=1
iecd.port1.loglevel=5
iecd.port1.tcp_keepalive_enabled=1
iecd.port1.tcp_keepalive_interval=5
iecd.port1.tcp_keepalive_timeout=5
iecd.port1.tcp_keepalive_count=3
iecd.port1.tcp_user_timeout=20000
iecd.port1.master_protocol=iec101
iecd.port1.slave_protocol=iec61850
iecd.port1.ioa_offset=0
iecd.port1.pointmap_file=/root/iecd/iecd_points1.csv
iecd.port1.iec104_local_ip=0.0.0.0
iecd.port1.iec104_local_tcpport=2404
iecd.port1.iec104_k=12
iecd.port1.iec104_w=9
iecd.port1.iec104_t2=10000
iecd.port1.iec104_gi_resp_time=200
iecd.port1.iec104_txq_size=128
iecd.port1.iec104_sync_time=1
```

```
iecd.port1.iec104_time_tagged_cmds=0
iecd.port1.iec104_cmd_delay_time=5000
iecd.port1.iec104_fsm_debug_on=0
iecd.port1.iec104_dump_data=0
iecd.port1.iec104_trace_on=0
iecd.port1.iec61850_local_ip=0.0.0.0
iecd.port1.iec61850_local_tcpport=104
iecd.port1.iec101_target_ip=127.0.0.1
iecd.port1.iec101_target_tcpport=999
iecd.port1.iec101_mode=unbalanced
iecd.port1.iec101_cot_tx_length=1
iecd.port1.iec101_cot_source_octet=0
iecd.port1.iec101_asdu_addrlen=1
iecd.port1.iec101_info_obj_addrlen=2
iecd.port1.iec101_data_polling_time=500
iecd.port1.iec101_ack_delay=0
iecd.port1.iec101_link_addrlen=1
iecd.port1.iec101_frame_rsp_time=2000
iecd.port1.iec101_max_tx_retry=3
iecd.port1.iec101_txq_size=128
iecd.port1.iec101_send_spont_delay_acq=1
iecd.port1.iec101_fsm_debug_on=0
iecd.port1.iec101_dump_data=0
iecd.port1.iec101_trace_on=0
iecd.@iec101link[0]=iec101link
iecd.@iec101link[0].portno=1
iecd.@iec101link[0].address=6
iecd.@iec101link[0].asduaddr=6
iecd.@point[0]=point
iecd.@point[0].portno=1
iecd.@point[0].iec61850_ld=SENSORS
iecd.@point[0].iec61850_ln=LLN0
iecd.@point[0].iec61850_do=SPSS01
iecd.@point[0].iec104_type_id=1
iecd.@point[0].iec104_ioa=5
iecd.@point[0].iec101_type_id=1
iecd.@point[0].iec101_ioa=5
```

```
iecd.@point[0].devaddr=1
iecd.@point[0].group=1
iecd.@point[0].index=0
iecd.@point[0].dword=0
iecd.@point[1]=point
iecd.@point[1].portno=1
iecd.@point[1].iec61850_ld=SENSORS
iecd.@point[1].iec61850_ln=LLN0
iecd.@point[1].iec61850_do=SPSS02
iecd.@point[1].iec104_type_id=1
iecd.@point[1].iec104_ioa=6
iecd.@point[1].iec101_type_id=1
iecd.@point[1].iec101_ioa=6
iecd.@point[1].devaddr=1
iecd.@point[1].group=1
iecd.@point[1].index=0
iecd.@point[1].dword=0
```

43.6.2.2 IEC61850 to IEC101 conversion using package options

```
root@VA_router:~# uci export iecd
package iecd

config iecd 'main'
        option enable '1'

config port 'port1'
        option enable '1'
        option loglevel '5'
        option tcp_keepalive_enabled '1'
        option tcp_keepalive_interval '5'
        option tcp_keepalive_timeout '5'
        option tcp_keepalive_count '3'
        option tcp_user_timeout '20000'
        option master_protocol 'iec101'
        option slave_protocol 'iec61850'
        option ioa_offset '0'
        option pointmap_file '/root/iecd/iecd_points1.csv'
```

```
        # IEC104 related settings
        option iec104_local_ip '0.0.0.0'
        option iec104_local_tcpport '2404'
        option iec104_k '12'
        option iec104_w '9'
        option iec104_t2 '10000'
        option iec104_gi_resp_time '200'
        option iec104_txq_size '128'
        option iec104_sync_time '1'
        option iec104_time_tagged_cmds '0'
        option iec104_cmd_delay_time '5000'
        option iec104_fsm_debug_on '0'
        option iec104_dump_data '0'
        option iec104_trace_on '0'

        # IEC61850 related settings
        option iec61850_local_ip '0.0.0.0'
        option iec61850_local_tcpport '104'

        option iec101_target_ip '127.0.0.1'
        option iec101_target_tcpport '999'
        option iec101_mode 'unbalanced'
        option iec101_cot_tx_length '1'
        option iec101_cot_source_octet '0'
        option iec101_asdu_addrlen '1'
        option iec101_info_obj_addrlen '2'
        option iec101_data_polling_time '500'
        option iec101_ack_delay '0'
        option iec101_link_addrlen '1'
        option iec101_frame_rsp_time '2000'
        option iec101_max_tx_retry '3'
        option iec101_txq_size '128'
        option iec101_send_spont_delay_acq '1'
        option iec101_fsm_debug_on '0'
        option iec101_dump_data '0'
        option iec101_trace_on '0'
```

```
# The following section defines IEC101 slave links used in IEC101
unbalanced mode on
# Each link is defined by a config block 'config iec101link'
# To add more links repeat the section block for each added link. To remove
links, s
# Maximum 32 links are supported
#
# Definition of options within the section block:
# portno - port number to which this point belongs (1 to 4)
# address - IEC101 slave link address
# asduaddr IEC101 slave common ASDU address

config iec101link
        option portno 1
        option address 6
        option asduaddr 6

config point
        option portno '1'
        option iec61850_ld 'SENSORS'
        option iec61850_ln 'LLN0'
        option iec61850_do 'SPSS01'
        option iec104_type_id '1'
        option iec104_ioa '5'
        option iec101_type_id 1
        option iec101_ioa '5'
        option devaddr '1'
        option group '1'
        option index '0'
        option dword '0'

config point
        option portno '1'
        option iec61850_ld 'SENSORS'
        option iec61850_ln 'LLN0'
        option iec61850_do 'SPSS02'
```

```
        option iec104_type_id '1'
        option iec104_ioa '6'
        option iec101_type_id 1
        option iec101_ioa '6'
        option devaddr '1'
        option group '1'
        option index '0'
        option dword '0'
```

43.7 Diagnostics

43.7.1 Starting and stopping services

The iecd and tserv background services are started automatically at router power up.

These services can be manually stopped, started or restarted as follows:

iecd

```
/etc/init.d/iecd stop – stops IECD service
/etc/init.d/iecd start – starts IECD service
/etc/init.d/iecd restart – stops and starts IECD service
```

tservd

```
/etc/init.d/tservd stop – stops TSERVD service
/etc/init.d/ tservd start – starts TSERVD service
/etc/init.d/ tservd restart – stops and starts TSERVD service
```

43.7.2 Events

The diagnosing and protocol tracing on the router the following features are available:

- Viewing syslog events (error messages)
- Running and viewing protocol traces (using syslog)
- Viewing statistic counters and debug information using diagnostic commands

To see the the appropriate debug information different debug options must be enabled.

The following table summarizes various options for tracing and diagnostics of the IEC104 to IEC101 / DNP3 / Modbus conversion:

Diagnostic feature	IEC104	IEC101	DNP3	MODBUS
Protocol Tracing	option log_severity '7' option	option log_severity '7' option	option log_severity '7' option	option log_severity '7' option modbus_trace_on '1'

	iec104_trace_on '1' /etc/init.d/iecd restart logread -f	iec101_trace_on '1' /etc/init.d/iecd restart logread -f	dnp3_trace_on '1' /etc/init.d/iecd restart logread -f	/etc/init.d/iecd restart logread -f
Viewing Rx / Tx Hex dump	option log_severity '7' option iec104_dump_data '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option iec101_dump_data '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option dnp3_dump_data '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option modbus_dump_data '1' /etc/init.d/iecd restart logread -f
Viewing Statistics	iec show stats	iec show stats	iec show stats	iec show stats
Clearing Statistics	iec clear satts	iec clear satts	iec clear satts	iec clear satts
Viewing debug information	N/a	N/a	N/a	iec show modbus debug
View point loaded points	iec show points	iec show points	iec show points	iec show points

Table 160: SCADA applications debug options table

43.7.3 Viewing statistics

To view IEC104 Gateway statistics, enter:

```
root@VA_router:~/iecd# iec show stats
Modbus stats:
========================
Modbus DL Frames Rx 20 Tx 3845 TxErrs 0
Modbus DL CRCErrs 0 Bad Addr 0 LengthErrs 0 UnknownPeer 0 SessionClose 0
Modbus App PDU Rx 20 PDU Tx 3845 PDU Rx Errors 0 PDU Rx Exception 0
Modbus App PDU Rx Timeout 3825 Unknown DevAddr 0 Rx Unexpected FC 0
Modbus App PDU TxQ Overrun 0

IEC104 stats:
========================
IEC104 DL state: CLOSED
IEC104 DL uptime: 0 hrs 0 mins 0 secs
IEC104 DL PktsRx 15 PktsTx 21 TxQ Overrun 0
IEC104 App ASDU Rx 6 ASDU Tx 12 Bad ASDU 0
```

43.7.4 Viewing point mappings

To view IEC104 Gateway point mappings, enter:

```
root@VA_router:~/iecd# iec show points
==== IEC104 point map: ====
IEC 104 Types Legend:
---------------
SPI: Single point information (1 bit)
DPI: Double point information (2 bit)
MVA: Measured normalized value (16 bit signed)
MVAFP: Measured value, floating point number (32 bit signed)
SVA: Measured scaled value (16 bit signed)
BSTR32: Bitstring of 32 bits
IT: Integrated Total (Counter 32 bit)
CP24: with 3 octet time tag CP24Time2a
CP56: with 7 octet time tag CP56Time2a
NQD: Without quality descriptor
---------------
(#1) IOA=64213, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 1, Type 0 (Discreet Input (1bit))
(#2) IOA=64214, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 2, Type 0 (Discreet Input (1bit))
(#3) IOA=64215, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 9, Type 0 (Discreet Input (1bit))
(#4) IOA=64216, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 10, Type 0 (Discreet Input (1bit))
(#5) IOA=64217, Val=0x00000000, IEC104TypeId=34 (MVA-CP56) DevAddr 1 Modbus pt 2, Type 1 (Input Register (16 bit))
(#6) IOA=64218, Val=0x00000000, IEC104TypeId=34 (MVA-CP56) DevAddr 1 Modbus pt 7, Type 1 (Input Register (16 bit))
(#7) IOA=64219, Val=0x00000000, IEC104TypeId=34 (MVA-CP56) DevAddr 1 Modbus pt 1, Type 2 (Holding Register (16 bit))
```

44 DNP3 outstation application

Virtual Access routers have a feature that allows the router to operate as a DNP3 outstation application. A DNP3 SCADA master can poll the router and obtain the following information:

- Router uptime in seconds.
- The serial number of the router.
- The status of up to two router interfaces.

44.1 Configuration packages used

Package	Sections
dnposd	dnposd

44.2 Configuring using the web interface

To configure the DNP3 outstation, from the top menu select **Services -> DNP3 Outstation**.

Check the **Enable** box and fill in your unique parameters.

The router listens for inbound UDP connections from the SCADA master on the specified port.

The web automatically names the dnposd config section 'main'.

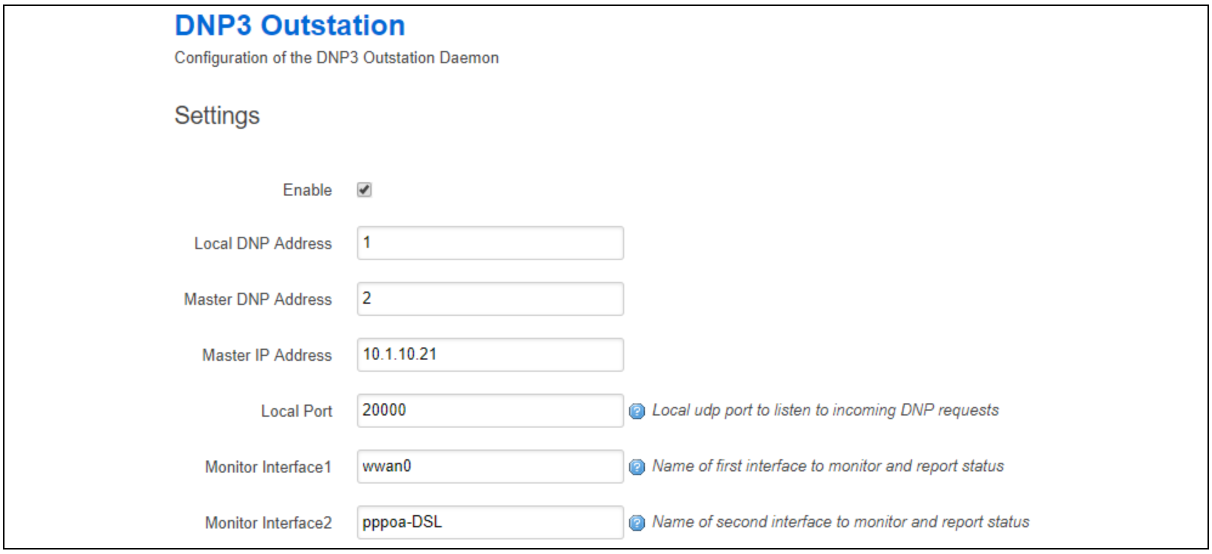


Figure 217: DNP3 outstation settings

Web Field / UCI / Package Option	Description
Web: Enable UCI: dnposd.main.enabled Opt: enabled	Enables the DNP3 outstation application on the router. 0 — Disabled. 1 — Enabled.
Web: Local DNP Address UCI: dnposd.main.local_address Opt: local_address	Defines the DNP3 address of the router. Blank Range — 0-65535
Web: Master DNP Address UCI: dnposd.main.master_address Opt: master_address	Defines the DNP3 address of the SCADA master. Blank Range — 0-65535
Web: Master IP Address UCI: dnposd.main.master_host Opt: master_host	Defines the IP Address of the SCADA master. Only requests from this host will be processed.
Web: Local Port UCI: dnposd.main.local_port Opt: local_port	Defines the UDP port for the router to listen on for incoming DNP3 messages from the SCADA master. 20000 Range — 0-65535
Web: Monitor Interface1 UCI: dnposd.main. monitor_if1 Opt: monitor_if1	Defines the first interface to monitor for status. Note: the interface names need to exactly match to the physical names. You can view the physical name by using **ifconfig** command via command line. Blank Range — 0-65535
Web: Monitor Interface2 UCI: dnposd.main. monitor_if2 Opt: monitor_if2	Defines the second interface to monitor for status. Note: the interface names need to exactly match to the physical names. You can view the physical name by using **ifconfig** command via command line. Blank Range — 0-65535

Table 161: Information table for DNP3 outstation settings

44.3 Configuring DNP3 outstation using command line

DNP3 outstation is configured under the **dnposd** package **/etc/config/dnp3osd**

44.3.1 DNP3 outstation using UCI

```
root@VA_router:~# uci show dnposd
dnposd.main=dnposd
dnposd.main.local_port=20000
dnposd.main.enabled=yes
dnposd.main.local_address=1
dnposd.main.master_address=2
dnposd.main.master_host=10.1.10.21
dnposd.main.monitor_if1=wwan0
```

```
dnposd.main.monitor_if2=pppoa-DSL
```

Modify these commands by running a `uci set <parameter>` command followed by `uci commit`.

44.3.2 DNP3 outstation using package options

```
root@VA_router:~# uci export dnposd
package dnposd

config dnposd 'main'
        option local_port '20000'
        option enabled 'yes'
        option local_address '1'
        option master_address '2'
        option master_host '10.1.10.21'
        option monitor_if1 'wwan0'
        option monitor_if2 'pppoa-DSL'
```

44.4 DNP3 outstation diagnostics

44.4.1 Restarting dnposd

To restart dnposd service, enter:

```
root@VA_router:~# /etc/init.d/dnposd restart
```

44.4.2 Tracing DNP3 packets

By default, the DNP3 outstation listens on UDP port 20000. To trace UDP packets on port 20000, enter:

```
root@VA_router:~# tcpdump -i any -n udp -p port 20000 &
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535
bytes
```

To stop tracing enter `fg` to bring tracing task to foreground, and then <CTRL-C> to stop the trace.

```
root@VA_router:~# fg

tcpdump -i any -n udp -p port 20000

^C

33 packets captured

33 packets received by filter

0 packets dropped by kernel
```

45 Serial interface

45.1 Overview

Many different applications and device drivers use the serial interface. You configure the serial interface using the relevant application; for example Terminal Server, therefore there is no standalone serial configuration page.

You can monitor the various serial interfaces using either the command line or the web interface.

45.2 Monitoring serial interfaces using the web interface

In the top menu, select **Status -> Serial Interfaces**. Depending on the number of serial interfaces present in the device, a number of tabs will appear giving access to information about each interface. The information presented will also depend on the actual type of the serial interface.

45.2.1 Serial statistics

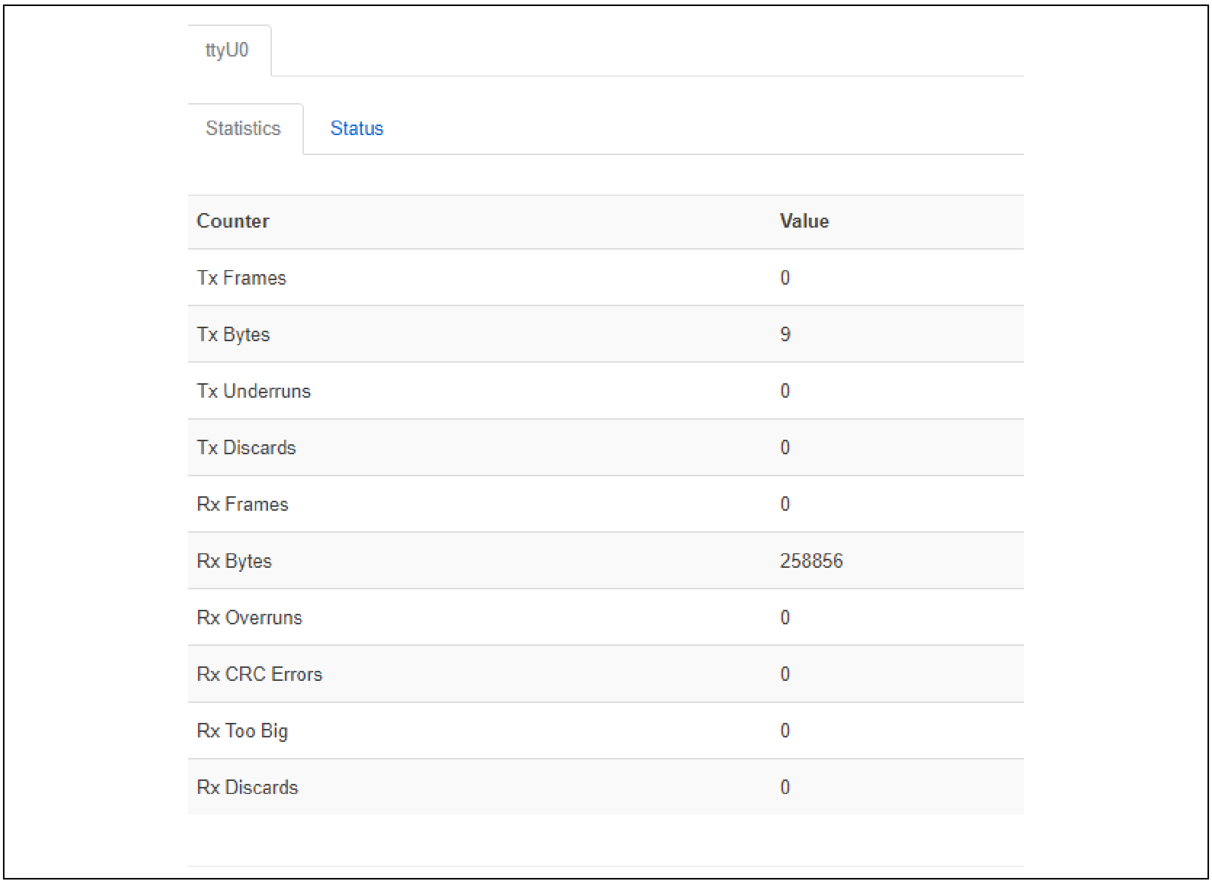


Figure 218: The serial statistics page for serial-0

45.2.2 Serial status

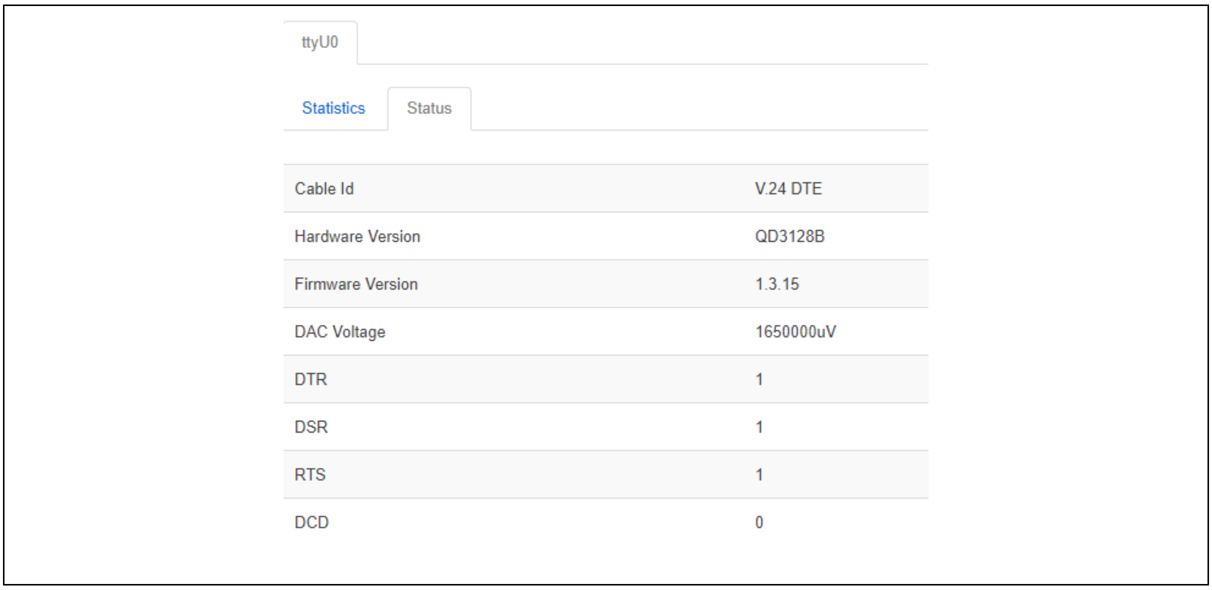


Figure 219: The serial status page for serial-0

45.3 Monitoring serial interfaces using command line

45.3.1 Serial statistics using command line

To view serial statistics, enter: `serial_stats`.

```
root@VirtualAccess:~# serial_stats
ttyU0 statistics
Tx Frames               0
Tx Bytes                9
Tx Underruns            0
Tx Discards             0
Rx Frames               0
Rx Bytes                258856
Rx Overruns             0
Rx CRC Errors           0
Rx Too Big              0
Rx Discards             0
```

45.3.2 Serial status using command line

To view serial statistics, enter: serial_status.

```
root@VirtualAccess:~# serial_status
ttyU0 status
Cable Id                 V.24 DTE
Hardware Version         QD3128B
Firmware Version         1.3.15
DAC Voltage              1650000uV
DTR                      1
DSR                      1
RTS                      1
DCD                      0
```

45.3.3 Resetting serial statistics

To reset serial statistics, enter: serial_stats_reset.

```
root@VirtualAccess:~# serial_stats_reset ttyU0
Serial interface statistics reset
```

You can reset statistics for all or individual serial interfaces.

____________________________END OF DOCUMENT______________________________