CRITICAL
APPLICATION
CONNECTIVITY

# GW2020 Series User Manual

Issue: 2.1

Date: 01 February 2018

**1 Introduction ..... 9**
1.1 Document scope ..... 9
1.2 Using this documentation ..... 9
**2 GW2020 Series router hardware specification ..... 12**
2.2 Hardware features ..... 12
2.3 Serial ports on the GW2020 Series router ..... 12
2.4 GSM and LTE technology ..... 14
2.5 Power supply ..... 14
2.6 GW2020 Series router power supply options ..... 14
2.7 GW2020 Series router environmental conditions ..... 15
2.8 GW2020 Series router dimensions ..... 15
2.9 GW2020 Series router compliance ..... 15
2.10 Operating temperature range ..... 15
2.11 Antenna ..... 15
2.12 Components ..... 15
2.13 Inserting a SIM card ..... 16
2.14 Connecting the SIM lock ..... 16
2.15 Connecting cables ..... 16
2.16 Connecting the antenna ..... 17
2.17 Powering up the GW2020 Series router ..... 17
2.18 Reset button ..... 17
2.19 Recovery mode ..... 17
**3 GW2020 Series router LED behaviour ..... 18**
3.1 Main LED behaviour ..... 18
3.2 GW2020 Ethernet port LED behaviour ..... 19
**4 Factory configuration extraction from SIM card ..... 20**
**5 Accessing the router ..... 21**
5.1 Configuration packages used ..... 21
5.2 Accessing the router over Ethernet using the web interface ..... 21
5.3 Accessing the router over Ethernet using an SSH client ..... 22
5.4 Accessing the router over Ethernet using a Telnet client ..... 23
5.5 Configuring the password ..... 23
5.6 Configuring the password using the web interface ..... 23
5.7 Configuring the password using UCI ..... 24
5.8 Configuring the password using package options ..... 24
5.9 Accessing the device using RADIUS authentication ..... 25
5.10 Accessing the device using TACACS+ authentication ..... 26
5.11 SSH ..... 29
5.12 Package dropbear using UCI ..... 31
5.13 Certs and private keys ..... 32

5.14 Configuring a router's web server ..... 33
5.15 Basic authentication (httpd conf) ..... 38
5.16 Securing uhttpd ..... 39
5.17 Displaying custom information via login screen ..... 39
**6 Router file structure ..... 41**
6.1 System information ..... 41
6.2 Identify your software version ..... 42
6.3 Image files ..... 43
6.4 Directory locations for UCI configuration files ..... 43
6.5 Viewing and changing current configuration ..... 43
6.6 Configuration file syntax ..... 44
6.7 Managing configurations ..... 44
6.8 Exporting a configuration file ..... 45
6.9 Importing a configuration file ..... 46
**7 Using the Command Line Interface ..... 50**
7.1 Overview of some common commands ..... 50
7.2 Using Unified Configuration Interface (UCI) ..... 53
7.3 Configuration files ..... 58
7.4 Configuration file syntax ..... 58
**8 Upgrading router firmware ..... 60**
8.1 Software versions ..... 60
8.2 Upgrading firmware using CLI ..... 66
**9 System settings ..... 69**
9.1 Configuration package used ..... 69
9.2 Configuring system properties ..... 69
9.3 System settings using UCI ..... 73
9.4 System diagnostics ..... 74
**10 Configuring an Ethernet interface ..... 76**
10.1 Configuration packages used ..... 76
10.2 Configuring an Ethernet interface using the web interface ..... 76
10.3 Interface configuration using UCI ..... 88
10.4 Configuring port maps ..... 91
10.5 Port map packages ..... 91
10.6 Interface diagnostics ..... 93
**11 Configuring VLAN ..... 95**
11.1 Maximum number of VLANs supported ..... 95
11.2 Configuration package used ..... 95
11.3 Configuring VLAN using the web interface ..... 95
11.4 Viewing VLAN interface settings ..... 98
11.5 Configuring VLAN using the UCI interface ..... 99

**12 Configuring a mobile connection ..... 100**
12.1 Configuration package used ..... 100
12.2 Configuring a mobile connection using the web interface ..... 100
12.3 Configuring a mobile connection using CLI ..... 106
12.4 Diagnositcs ..... 107
**13 Configuring mobile manager ..... 111**
13.1 Configuration package used ..... 111
13.2 Configuring mobile manager using the web interface ..... 111
13.3 Configuring mobile manager using command line ..... 116
13.4 Monitoring SMS ..... 117
13.5 Sending SMS from the router ..... 118
13.6 Sending SMS to the router ..... 118
**14 Configuring a GRE interface ..... 119**
14.1 Configuration packages used ..... 119
14.2 Creating a GRE connection using the web interface ..... 119
14.3 GRE configuration using command line ..... 124
14.4 GRE configuration using UCI ..... 124
14.5 GRE configuration using package options ..... 124
14.6 GRE diagnostics ..... 125
**15 Configuring static routes ..... 127**
15.1 Configuration package used ..... 127
15.2 Configuring static routes using the web interface ..... 127
15.3 Configuring IPv6 routes using the web interface ..... 128
15.4 Configuring routes using command line ..... 128
15.5 IPv4 routes using UCI ..... 129
15.6 IPv4 routes using package options ..... 130
15.7 IPv6 routes using UCI ..... 130
15.8 IPv6 routes using packages options ..... 130
15.9 Static routes diagnostics ..... 131
**16 Configuring BGP (Border Gateway Protocol) ..... 132**
16.1 Configuration package used ..... 132
16.2 Configuring BGP using the web interface ..... 132
16.3 Configuring BGP using UCI ..... 135
16.4 Configuring BGP using packages options ..... 136
16.5 View routes statistics ..... 137
**17 Configuring OSPF (Open Shortest Path First) ..... 138**
17.1 Introduction ..... 138
17.2 Configuration package used ..... 143
17.3 Configuring OSPF using the web interface ..... 144
17.4 Configuring OSPF using the command line ..... 147

17.5 OSPF using UCI ..... 148
17.6 OSPF using package options ..... 149
17.7 OSPF diagnostics ..... 150
17.8 Quagga/Zebra console ..... 151
**18 Configuring VRRP ..... 157**
18.1 Overview ..... 157
18.2 Configuration package used ..... 157
18.3 Configuring VRRP using the web interface ..... 157
18.4 Configuring VRRP using command line ..... 161
**19 Configuring Routing Information Protocol (RIP) ..... 164**
19.1 Introduction ..... 164
19.2 Configuration package used ..... 165
19.3 Configuring RIP using the web interface ..... 166
19.4 Configuring RIP using command line ..... 170
19.5 RIP diagnostics ..... 174
**20 Configuring Multi-WAN ..... 178**
20.1 Configuration package used ..... 178
20.2 Configuring Multi-WAN using the web interface ..... 178
20.3 Configuring Multi-WAN using UCI ..... 182
20.4 Multi-WAN diagnostics ..... 184
**21 Automatic operator selection ..... 186**
21.1 Configuration package used ..... 186
21.2 Configuring automatic operator selection via the web interface ..... 186
21.3 Configuring via UCI ..... 210
21.4 Configuring no PMP + roaming using UCI ..... 214
21.5 Automatic operator selection diagnostics via the web interface ..... 216
21.6 Automatic operator selection diagnostics via UCI ..... 218
**22 Configuring Connection Watch (cwatch) ..... 221**
22.1 Configuration package used ..... 221
22.2 Configuring Connection Watch using the web interface ..... 221
22.3 Configuring cwatch using command line ..... 223
22.4 cwatch diagnostics ..... 224
**23 Configuring DHCP server and DNS (Dnsmasq) ..... 225**
23.1 Configuration package used ..... 225
23.2 Configuring DHCP and DNS using the web interface ..... 225
23.3 Configuring DHCP and DNS using UCI ..... 233
23.4 Configuring DHCP pools using UCI ..... 235
23.5 Configuring static leases using UCI ..... 236
**24 Configuring DHCP client ..... 237**
24.1 Configuration packages used ..... 237

24.2 Configuring DHCP client using the web interface ..... 237
24.3 Configuring DHCP client using command line ..... 243
24.4 DHCP client diagnostics ..... 244
**25 Configuring DHCP forwarding ..... 247**
25.1 Configuration packages used ..... 247
25.2 Configuring DHCP forwarding using the web interface ..... 247
25.3 Configuring DHCP forwarding using command line ..... 248
25.4 DHCP forwarding over IPSec ..... 249
25.5 DHCP forwarding diagnostics ..... 252
**26 Configuring Dynamic DNS ..... 254**
26.1 Overview ..... 254
26.2 Configuration packages used ..... 254
26.3 Configuring Dynamic DNS using the web interface ..... 254
26.4 Dynamic DNS using UCI ..... 256
**27 Configuring hostnames ..... 258**
27.1 Overview ..... 258
27.2 Local host file records ..... 258
27.3 PTR records ..... 260
27.4 Static leases ..... 262
**28 Configuring firewall ..... 265**
28.1 Configuration package used ..... 265
28.2 Configuring firewall using the web interface ..... 265
28.3 Configuring firewall using UCI ..... 277
28.4 IPv6 notes ..... 280
28.5 Implications of DROP vs. REJECT ..... 280
28.6 Connection tracking ..... 281
28.7 Firewall examples ..... 281
**29 Configuring IPSec ..... 289**
29.1 Configuration package used ..... 289
29.2 Configuring IPSec using the web interface ..... 289
29.3 Configuring IPSec using UCI ..... 298
29.4 Configuring an IPSec template for DMVPN via the web interface ..... 302
29.5 Configuring an IPSec template to use with DMVPN ..... 309
29.6 IPSec diagnostics using the web interface ..... 311
29.7 IPSec diagnostics using UCI ..... 311
**30 Dynamic Multipoint Virtual Private Network (DMVPN) ..... 312**
30.1 Prerequisites for configuring DMVPN ..... 312
30.2 Advantages of using DMVPN ..... 312
30.3 DMVPN scenarios ..... 313
30.4 Configuration packages used ..... 315

30.5 Configuring DMVPN using the web interface ..... 315
30.6 DMVPN diagnostics ..... 317
**31 Configuring multicasting using PIM and IGMP interfaces ..... 320**
31.1 Overview ..... 320
31.2 Configuration package used ..... 320
31.3 Configuring PIM and IGMP using the web interface ..... 320
31.4 Configuring PIM and IGMP using UCI ..... 322
**32 QoS: VLAN 802.1Q PCP tagging ..... 324**
32.1 Configuring VLAN PCP tagging ..... 324
**33 QoS: type of service ..... 327**
33.1 QoS configuration overview ..... 327
33.2 Configuration packages used ..... 327
33.3 Configuring QoS using the web interface ..... 327
33.4 Configuring QoS using UCI ..... 329
33.5 Example QoS configurations ..... 332
**34 Management configuration settings ..... 333**
34.1 Activator ..... 333
34.2 Monitor ..... 333
34.3 Configuration packages used ..... 333
34.4 Autoload: boot up activation ..... 334
34.5 Autoload packages ..... 334
34.6 Autoload using UCI ..... 337
34.7 HTTP Client: configuring activation using the web interface ..... 338
34.8 Httpclient: Activator configuration using UCI ..... 341
34.9 Httpclient: Activator configuration using package options ..... 341
34.10 User management using UCI ..... 342
34.11 Configuring the management user password using UCI ..... 343
34.12 Configuring management user password using package options ..... 344
34.13 User management using UCI ..... 344
34.14 User management using package options ..... 344
34.15 Configuring user access to specific web pages ..... 345
**35 Configuring Monitor ..... 346**
35.1 Introduction ..... 346
35.2 Reporting device status to Monitor ..... 346
35.3 Reporting GPS location to Monitor ..... 352
35.4 Reporting syslog to Monitor ..... 353
35.5 Configuring ISAD ..... 355
**36 Configuring SNMP ..... 358**
36.1 Configuration package used ..... 358
36.2 Configuring SMNP using the web interface ..... 358

36.3 Configuring SNMP using command line ..... 364
36.4 Configuring SNMP interface alias with static SNMP index ..... 370
36.5 SNMP diagnostics ..... 372
**37 Event system ..... 374**
37.1 Configuration package used ..... 374
37.2 Event system overview ..... 374
37.3 Configuring the event system using the web interface ..... 375
37.4 Configuring the event system using command line ..... 386
37.5 Event system diagnostics ..... 394
**38 Configuring data usage monitor ..... 398**
38.1 Introduction ..... 398
38.2 Configuration package used ..... 398
38.3 Configuring data usage using the web interface ..... 398
38.4 Data usage status ..... 401
38.5 Data usage diagnostics ..... 401
**39 Configuring Terminal Server ..... 403**
39.1 Overview ..... 403
39.2 Configuration packages used ..... 403
39.3 Configuring Terminal Server using the web interface ..... 403
39.4 Terminal Server using UCI ..... 414
39.5 Terminal Server using package options ..... 414
39.6 Terminal server DSR signal management based on network configuration ..... 414
39.7 Terminal Server diagnostics ..... 416
**40 Configuring terminal package ..... 419**
40.1 Configuration packages used ..... 419
40.2 Configuring terminal package using the web interface ..... 419
40.3 Configuring terminal package using UCI ..... 419
40.4 Configuring terminal using package options ..... 420
40.5 Terminal diagnostics ..... 420
**41 Serial interface ..... 421**
41.1 Overview ..... 421
41.2 Monitoring serial interfaces using the web interface ..... 421
41.3 Monitoring serial interfaces using command line ..... 422

# 1 Introduction

This user manual describes the features and how to configure the range of Virtual Access routers in the GW2020 Series.

The Virtual Access GW2020 Series routers are arrange of versatile 3G/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 following models in the GW2020 Series.

| | |
|---|---|
| GW2021: | 1 x Ethernet and 3G, 4G/LTE |
| GW2022: | 2 x Ethernet and 3G, 4G/LTE |
| GW2023: | 2 x Ethernet, 3G, 4G/LTE and dual RS232 |
| GW2024: | 2 x Ethernet, 3G, 4G/LTE, single RS232 and single RS485 |

## 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<br>UCI: network.@route[0].metric<br>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.

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

**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 GW2020 Series router hardware specification

### 2.1.1 GW2020 Series router model variants

**Figure 1: The GW2020 series router**

| | |
|---|---|
| GW2021: | 1 x Ethernet and 3G, 4G/LTE, dual SIM, metal case |
| GW2022: | 2 x Ethernet and 3G, 4G/LTE, dual SIM, metal case |
| GW2023: | 2 x Ethernet, 3G, 4G/LTE and dual RS232, dual SIM, metal case |
| GW2024: | 2 x Ethernet, 3G, 4G/LTE, single RS232 and single RS485, dual SIM, metal case |

## 2.2 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

## 2.3 Serial ports on the GW2020 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.

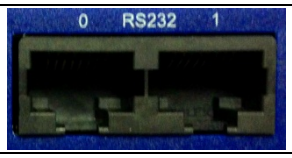



**Figure 2: Serial ports on the GW2020 series router**

#### 2.3.1.1 RS232 pinout for the GW2020 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 |

#### 2.3.1.2 RS485 pinout for the GW2020 Series router

| | Half Duplex Mode | | Full Duplex Mode | |
|---|---|---|---|---|
| Pin | Name | Direction (From GW2020 Series router) | Name | Direction (From GW2020 Series router) |
| 1 | - | - | Rx+ | In |
| 2 | - | - | Rx- | In |
| 3 | Tx/Rx+ | In/Out | Tx+ | Out |
| 4 | GND | - | GND | - |
| 5 | GND | - | GND | - |
| 6 | Tx/Rx- | In/Out | Tx- | Out |
| 7 | - | - | - | - |
| 8 | - | - | - | - |

## 2.4 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

## 2.5 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.**

### 2.5.1 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**

## 2.6 GW2020 Series router power supply options

The GW2020 Series router has three power supply options:

- 100V-240V AC PSU (standard)
- 100V-240V AC PSU with extended temperature support -20°C to +70°C
- 9V-59V DC input. DC power cable with locking connector and bare ends supplied

## 2.7 GW2020 Series router environmental conditions

The following environmental conditions apply to all GW2020 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

## 2.8 GW2020 Series router dimensions

| | |
|---|---|
| GW2020 Series unit size: | 100W 138D 34H mm |
| GW2020 Series unit weight: | 500g |

## 2.9 GW2020 Series router compliance

The GW2020 Series router is compliant and tested to the following standards:

| | |
|---|---|
| Safety | EN60950-1 |
| EMC | EN55022:1998 Class B and EN55024:1998 Class B |
| Environmental | ETSI 300 019-1-3 Sinusoidal Vibration and Shock ETSI 300 019-2-3 Random Vibration. |

## 2.10 Operating temperature range

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

| | | |
|---|---|---|
| GW202X | 0°C to 40°C | Standard AC PSU |
| GW202X-ET | -20°C to 70°C | Extended temperature AC PSU |
| GW202X-DC | -20°C to 70°C | DC power cable |

## 2.11 Antenna

The GW2020 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.

## 2.12 Components

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

The GW2020 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.

### 2.12.1 GW2020 Series router components

The GW2020 Series router comes with the following components as standard.

| | |
|---|---|
| 1 x GW2020 Series router (models vary). | |
| 1 x Ethernet cable. RJ45 connector at both ends. | |
| 1 x power supply unit. | |
| 1 x rubber right angle antenna. | |

**Table 3: GW2020 Series router standard components**

Optional components include:

| | |
|---|---|
| 1 x lockable SIM cover. | |
| 1 x extra antenna | Virtual Access supplies a wide range of antennas. Please visit our website: www.virtualaccess.com or contact Virtual Access for more information. |

**Table 4:GW2020 Series router optional components**

## 2.13 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.

## 2.14 Connecting the SIM lock

Connect the SIM lock using the Allen key provided.

## 2.15 Connecting cables

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

## 2.16 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.

## 2.17 Powering up the GW2020 Series router

Plug the power cable into an electrical socket suitable for the power supply.

The GW2020 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.

## 2.18 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 5: GW2020 Series router reset behaviour**

## 2.19 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 GW2020 Series router LED behaviour

## 3.1 Main LED behaviour

The GW2020 Series router has single colour LEDs for Power, Config, SIM1, SIM2 and signal strength. When the router is powered on, the LED is green.

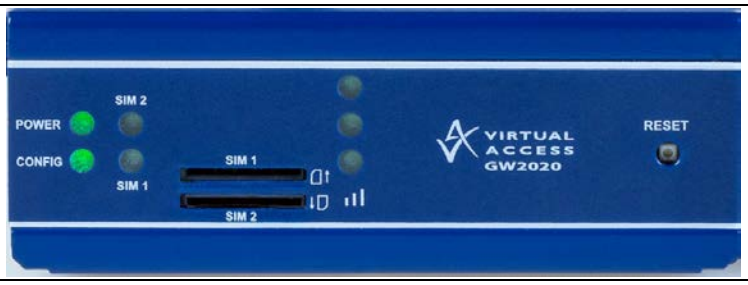



**Figure 3: Example of power and config LED activity: power and config are on**

The possible LED states are:

- Off
- Flashing slowing
- Flashing quickly
- On

The following table describes the possible LED behaviour and meanings.

| | | |
|---|---|---|
| Booting | | The GW2020 takes approximately 2 minutes to boot up. During this time, the power LED flashes.<br>Other LEDs display different diagnostic patterns during boot up.<br>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 data connection is being established. |
| Signal LEDs | None | Not connected or signal strength <= -113dBm. |
| | 1 | Connected and signal strength <= -89dBm. |
| | 2 | Connected and signal strength between -89dBm and -69dBm. |
| | 3 | Connected and signal strength >-69dBm. |

**Table 6: LED behaviour and descriptions**

**Note**: when a data connection does not exist, none of the signal LEDs will light regardless of signal strength.

## 3.2 GW2020 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 left hand side is the LINK LED, and the ACT LED is on the right hand side.

**Figure 4: 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<br>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.

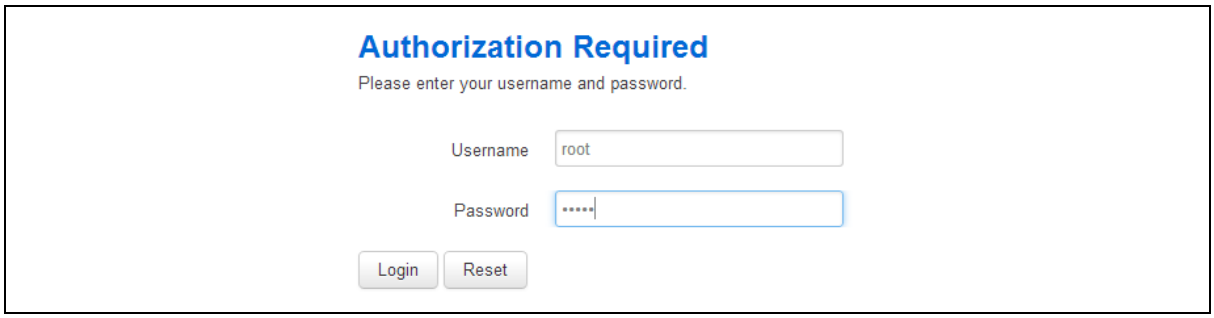



**Figure 5: The login page**

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 6: Confirming trust of the routers public key over SSH**

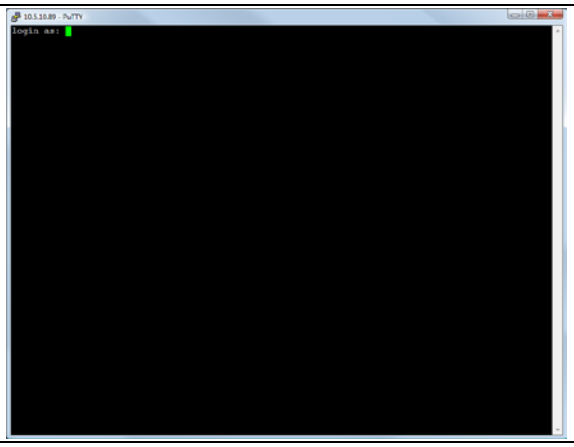



**Figure 7: SSH CLI logon screen**

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.

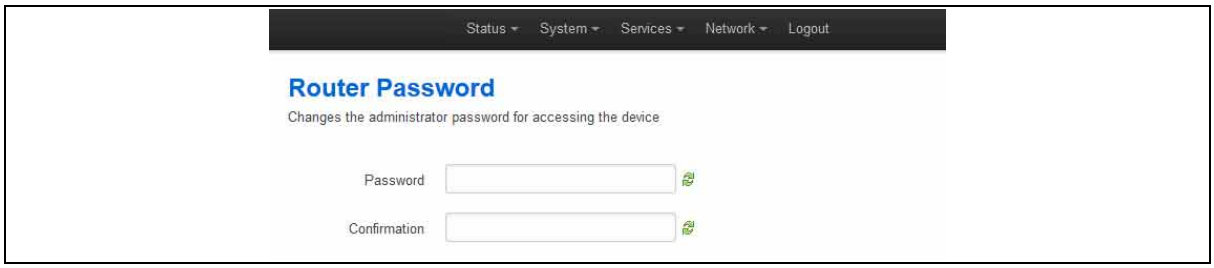



**Figure 8: The router password section**

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<br>UCI: system.main.password<br>Opt: password | Defines the root password. The password is displayed encrypted via the CLI using the 'hashpassword' option.<br>UCI: system.main.hashpassword<br>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'
```

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

**Table 7: 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'
```

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

**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.

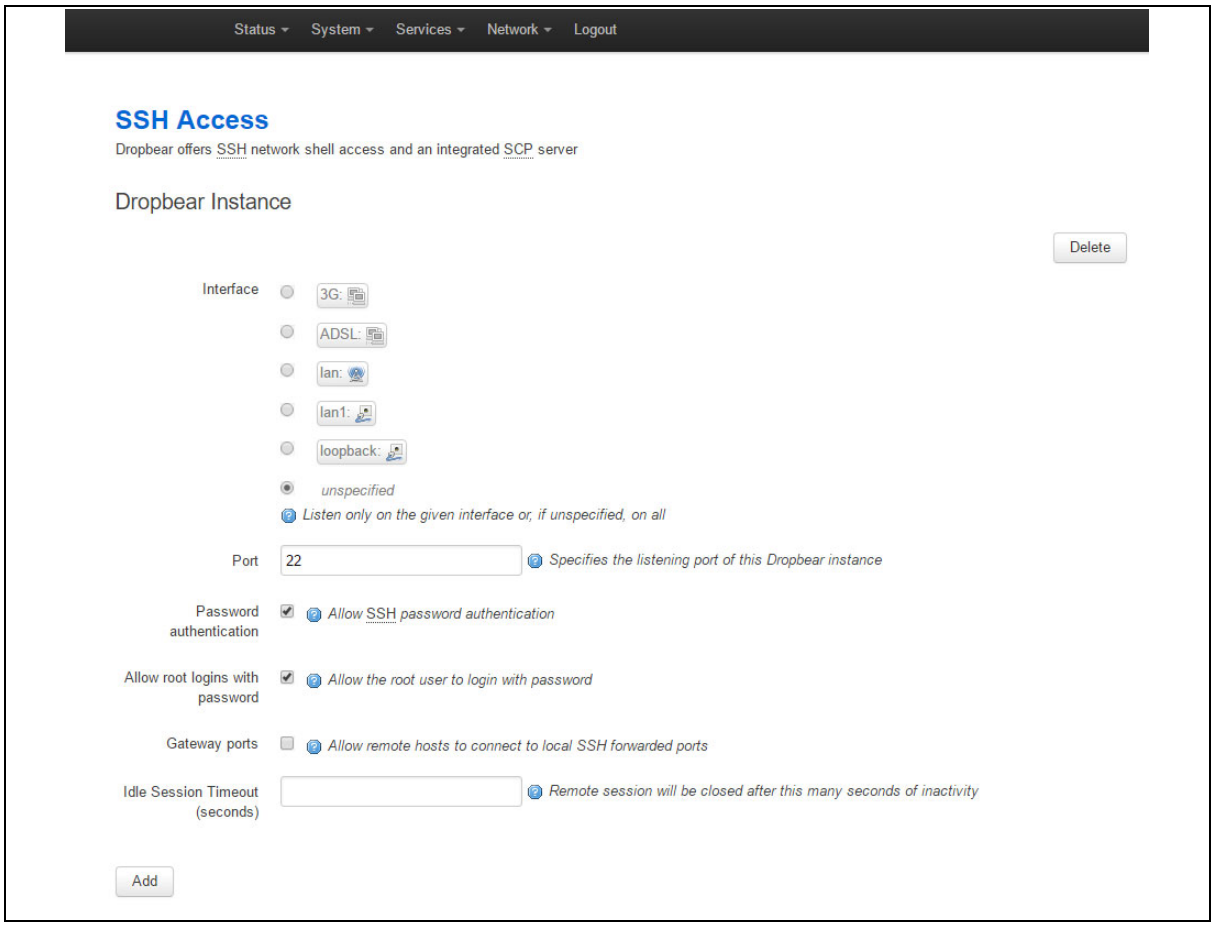



**Figure 9: The SSH access section**

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

**Table 8: 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.

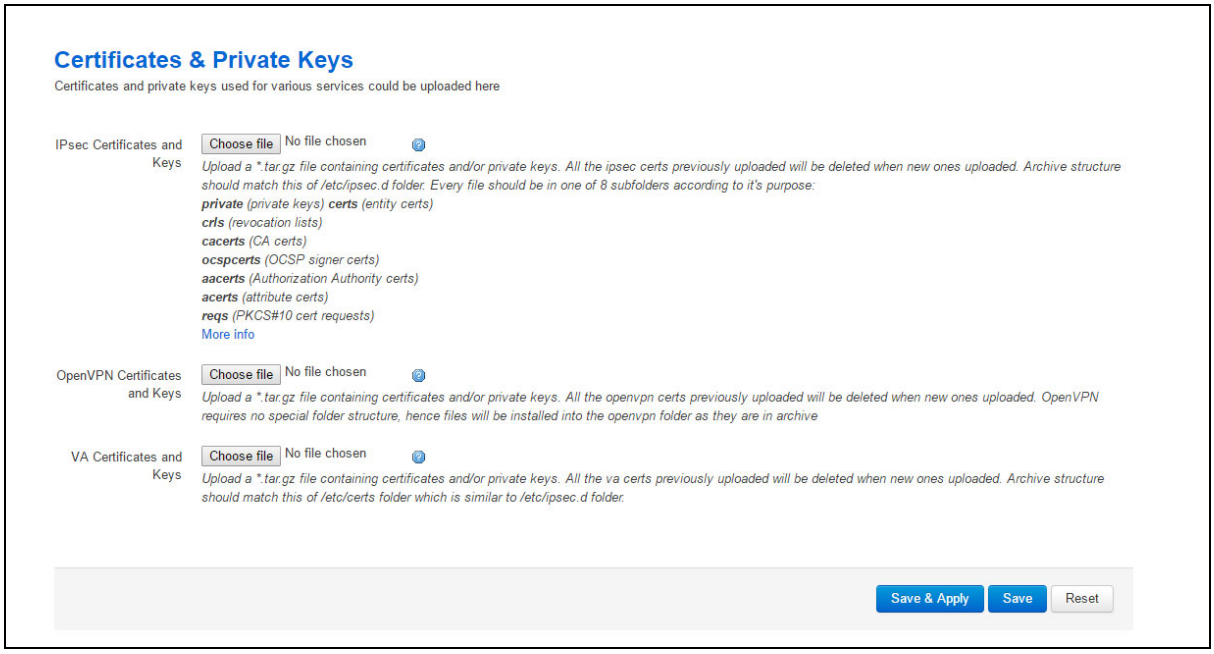



**Figure 10: The certificates & private keys section**

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 11: 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

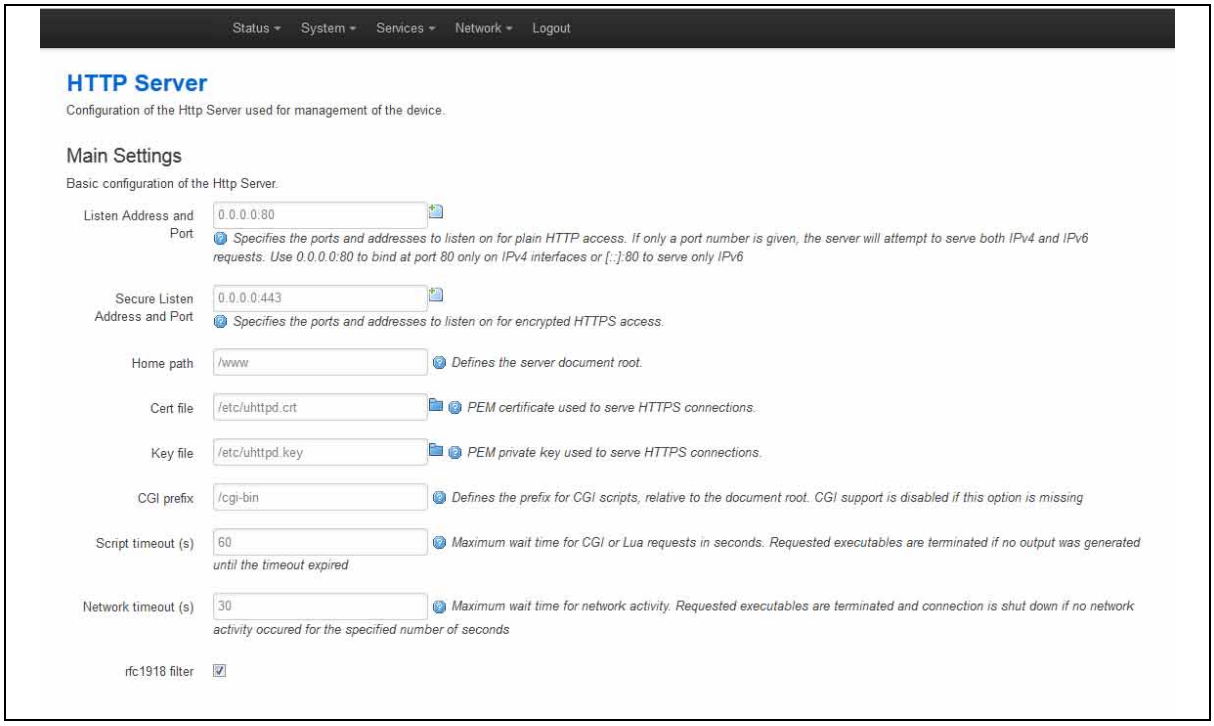



**Figure 12: HTTP server settings**

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

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

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

**Table 9: 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.

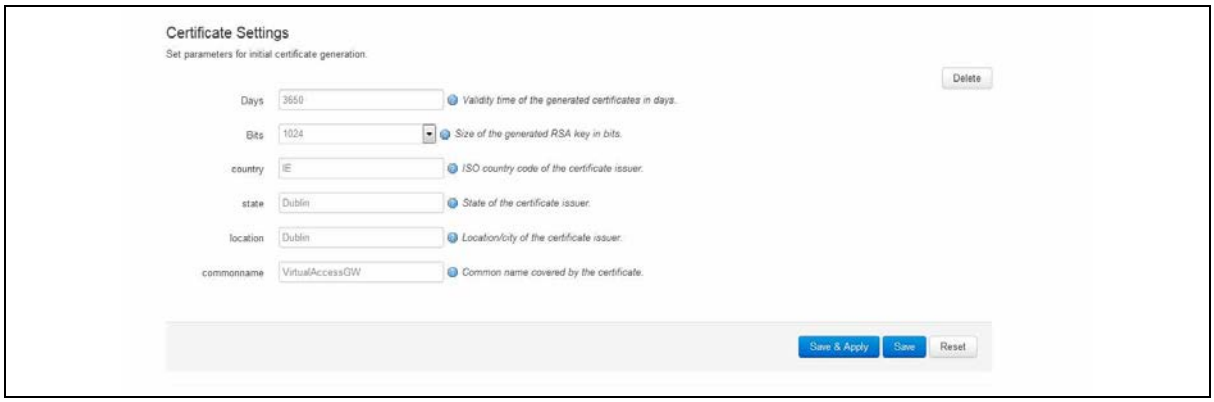



**Figure 13: HTTP server certificate settings**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Days<br>UCI: uhttpd.px5g.days<br>Opt: days | Validity time of the generated certificates in days.<br>730<br>Range |
| Web: Bits<br>UCI: uhttpd.px5g.bits<br>Opt: bits | Size of the generated RSA key in bits.<br>1024<br>Range |
| Web: Country<br>UCI: uhttpd.px5g.country<br>Opt: country | ISO code of the certificate issuer. |
| Web: State<br>UCI: uhttpd.px5g.state<br>Opt: state | State of the certificate issuer. |
| Web: Location<br>UCI: uhttpd.px5g.location<br>Opt: location | Location or city of the certificate user. |
| Web: Commonname<br>UCI: uhttpd.commonname<br>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 10: 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 14: 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.

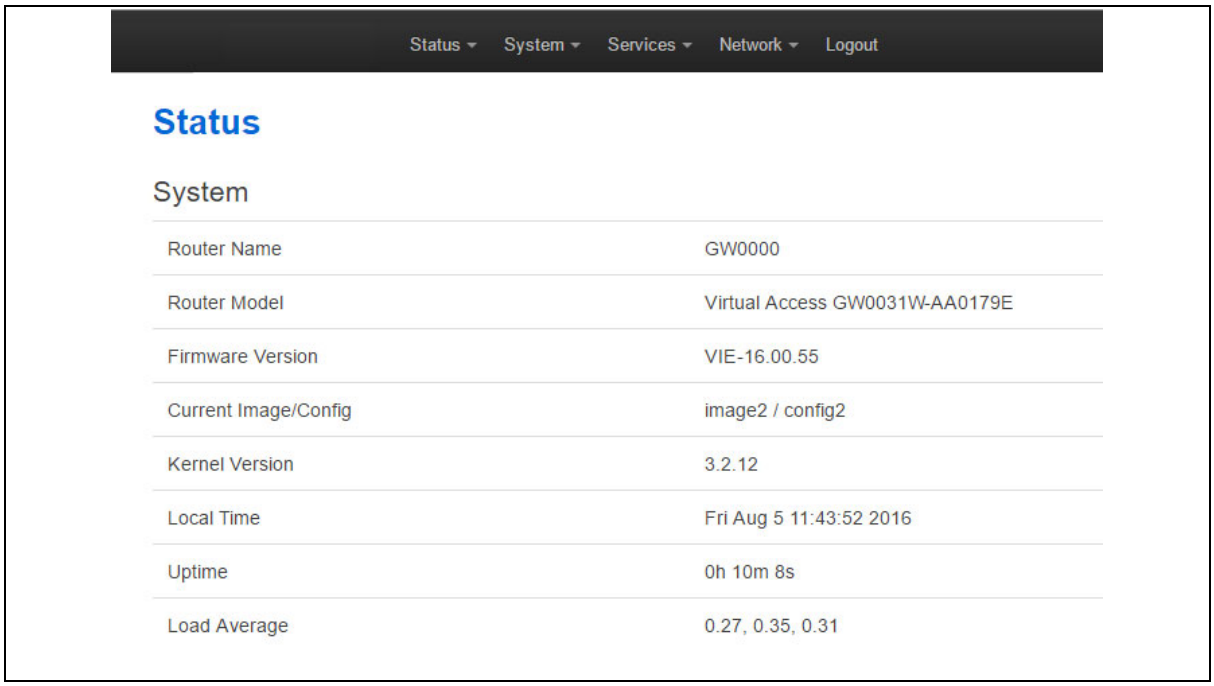



**Figure 15: Example of the status page**

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

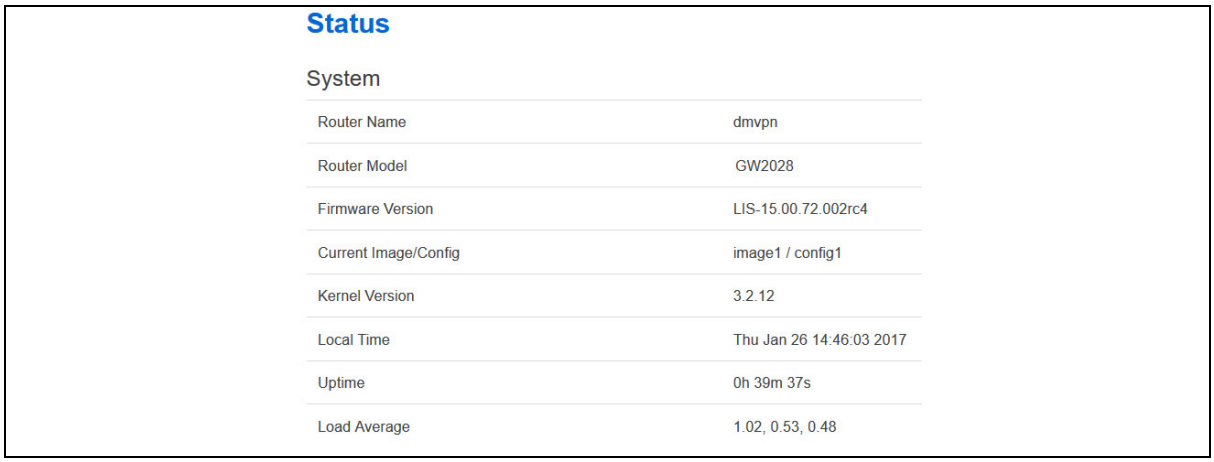
To check which software version your router is running, in the top menu, browse to **Status -> Overview**.

Status

System

| | |
|---|---|
| Router Name | GW0000 |
| Router Model | Virtual Access GW0031W-AA0179E |
| Firmware Version | VIE-16.00.55 |
| Current Image/Config | image2 / config2 |
| Kernel Version | 3.2.12 |
| Local Time | Fri Aug 5 11:43:52 2016 |
| Uptime | 0h 10m 8s |
| Load Average | 0.27, 0.35, 0.31 |

**Figure 16: The status page showing a software version prior to 72.002**



Status

System

| | |
|---|---|
| Router Name | dmvpn |
| Router Model | GW2028 |
| Firmware Version | LIS-15.00.72.002rc4 |
| Current Image/Config | image1 / config1 |
| Kernel Version | 3.2.12 |
| Local Time | Thu Jan 26 14:46:03 2017 |
| Uptime | 0h 39m 37s |
| Load Average | 1.02, 0.53, 0.48 |

**Figure 17: 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.

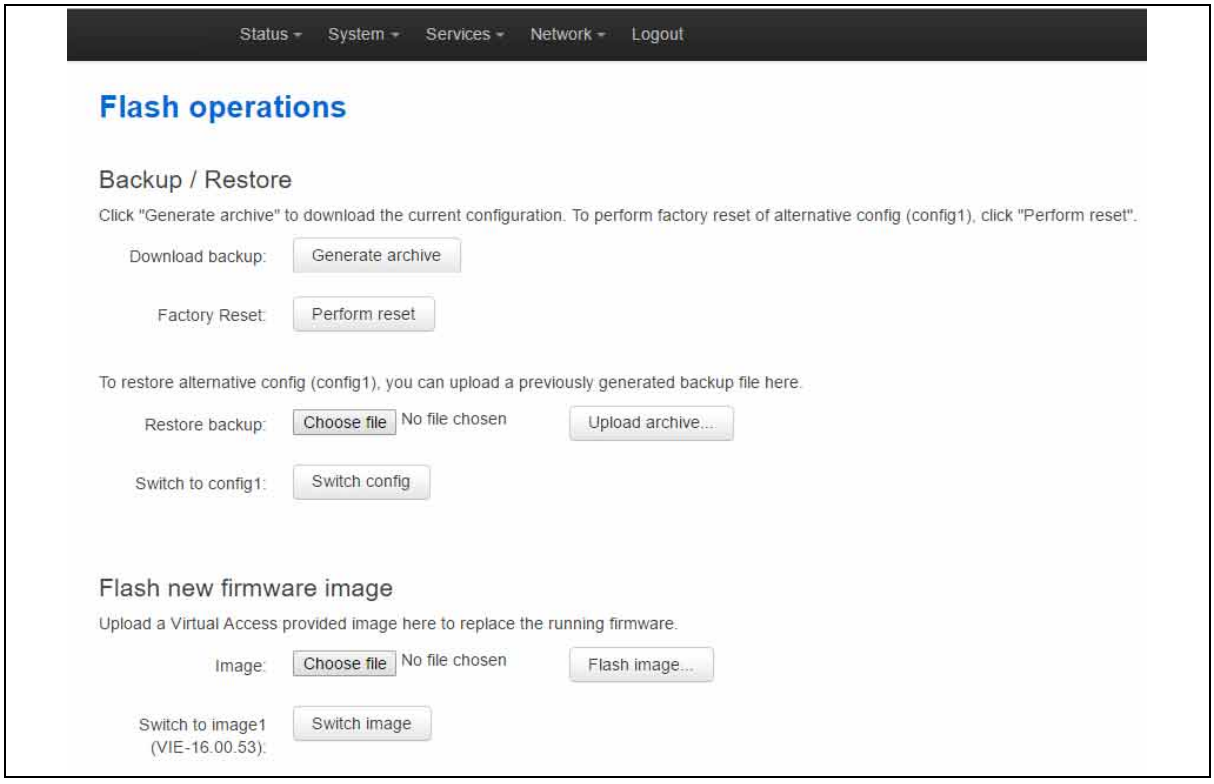



**Figure 18: The flash operations page**

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.

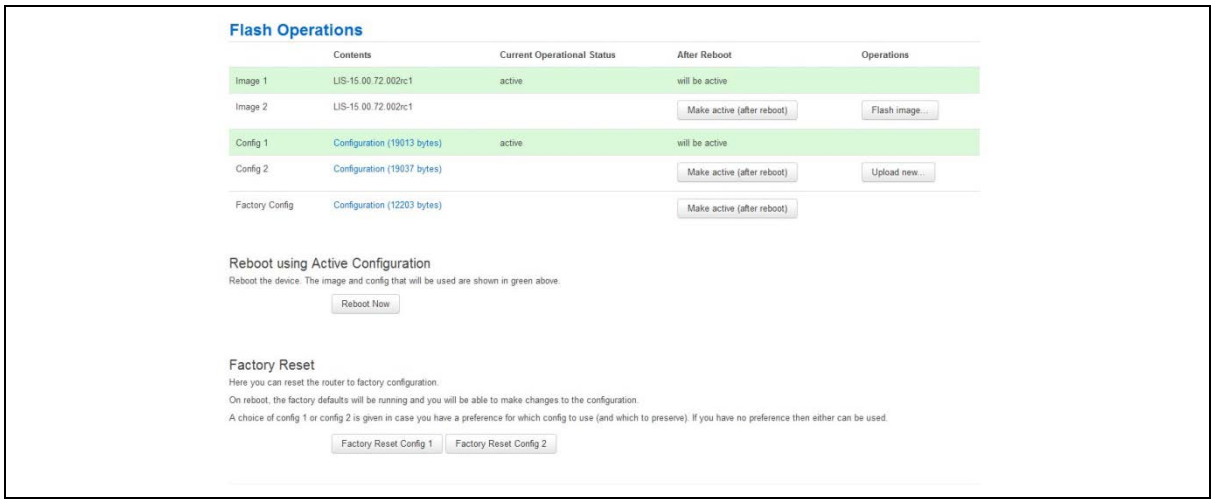



**Figure 19: The flash operations page**

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.

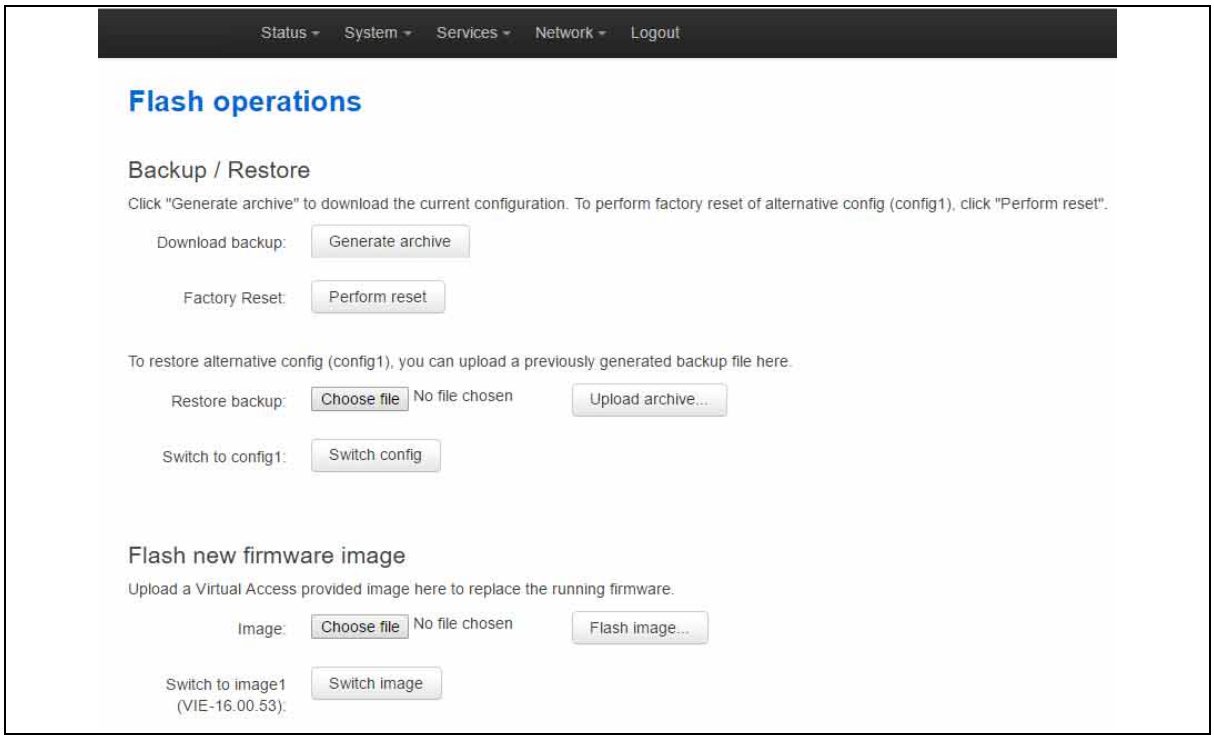



**Figure 20: The flash operations page**

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

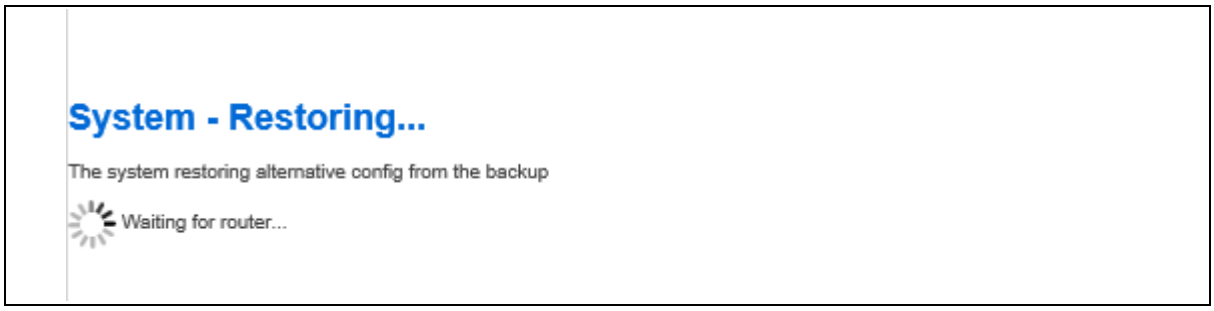



**Figure 21: The system – restoring…page**

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.

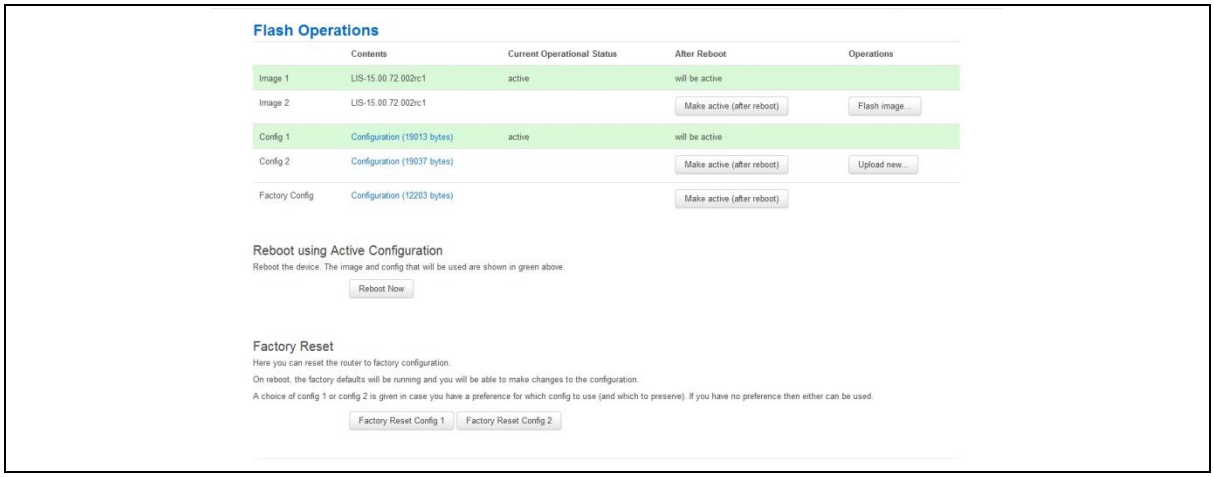



**Figure 22: The flash operations page**

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

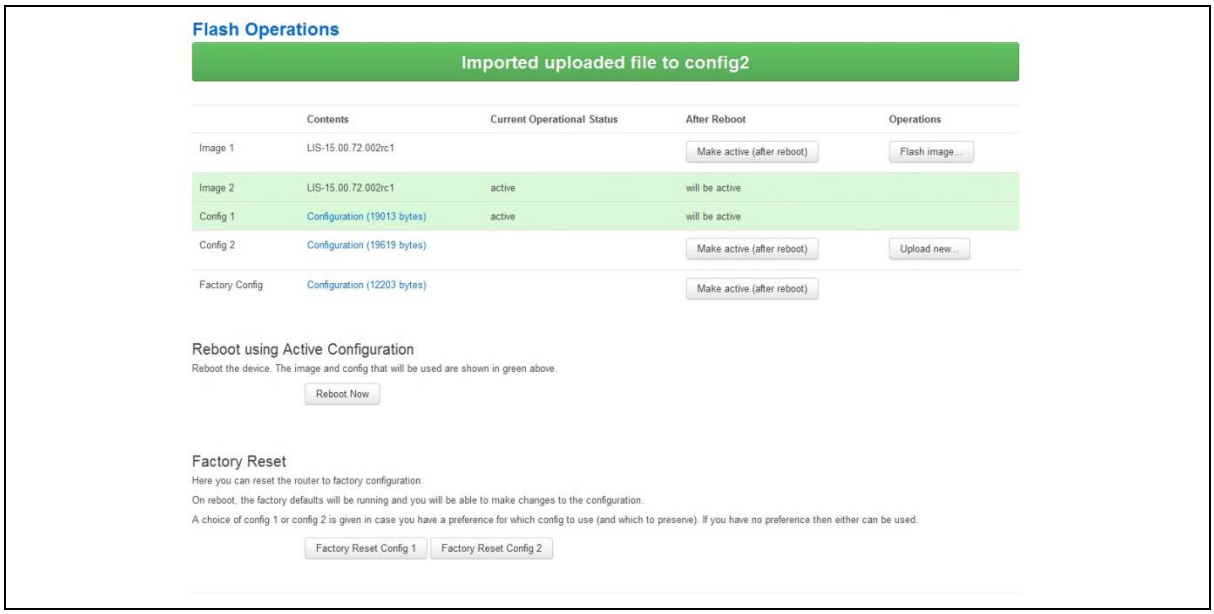



**Figure 23: The flash operations succeed upload configuration page**

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 11: 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 24: The status page showing a software version prior to 72.002**

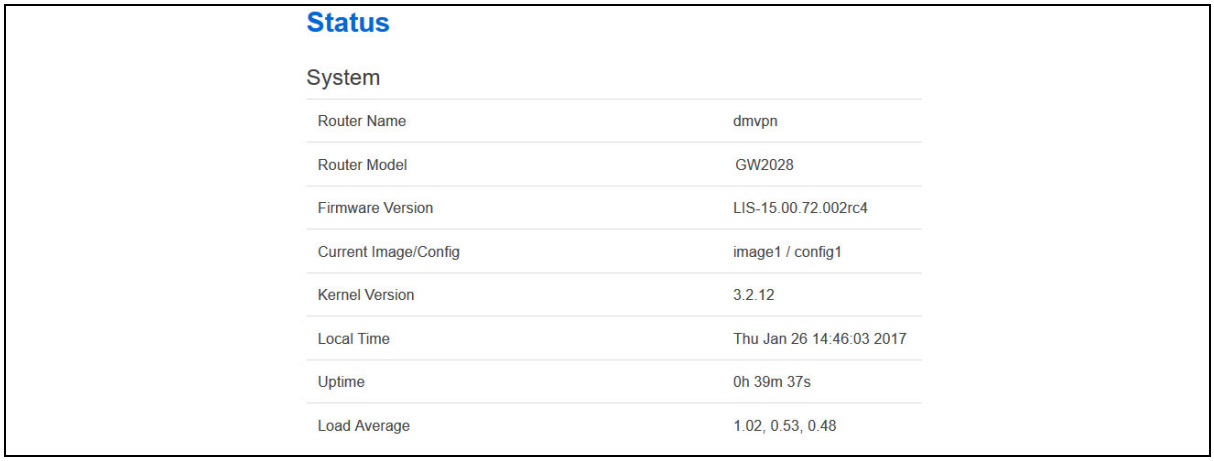



**Figure 25: The status page showing software version 72.002**

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.

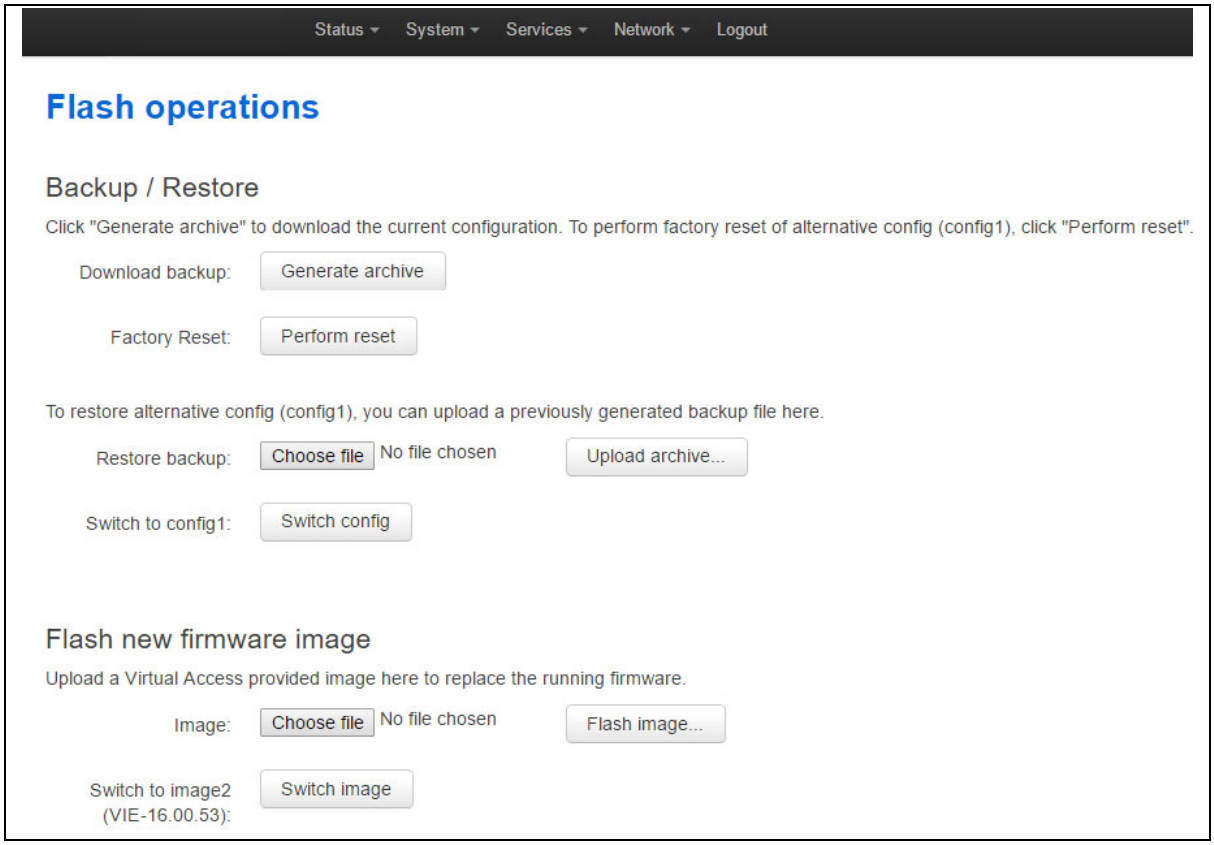



**Figure 26: The flash operations page**

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.

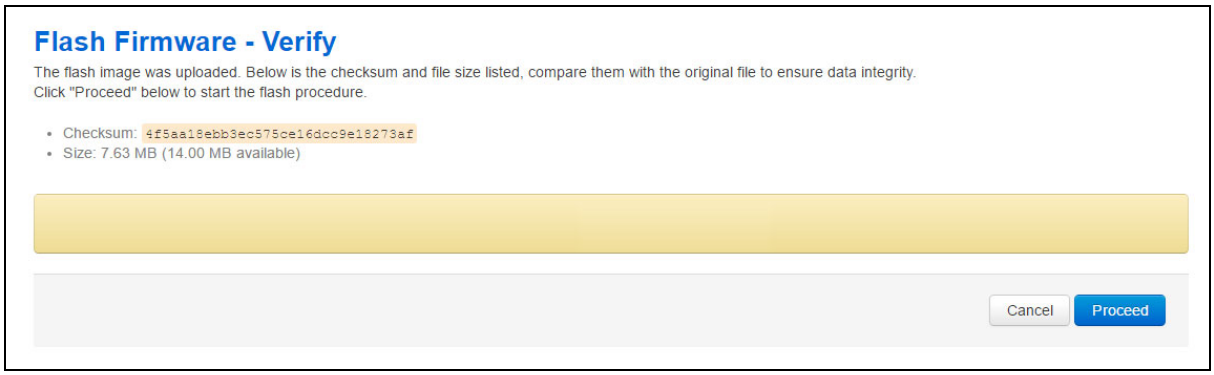



**Figure 27: The flash firmware - verify page**

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

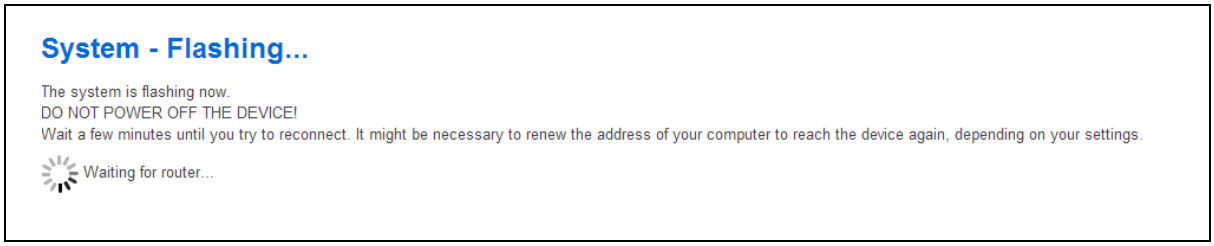



**Figure 28: The system – flashing…page**

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 29: 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.

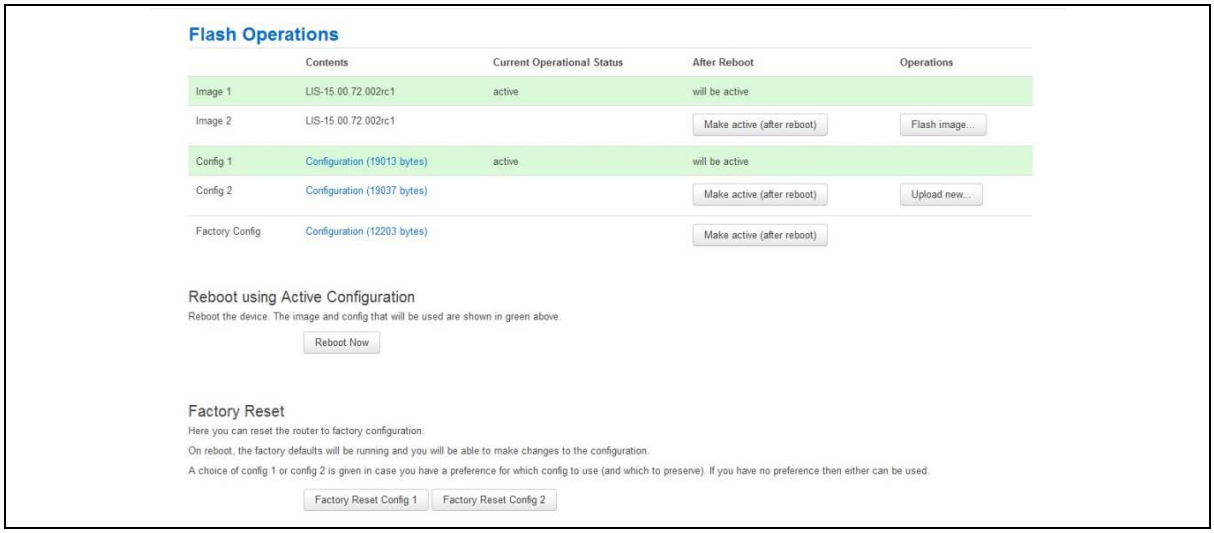



**Figure 30: The flash operations page**

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.

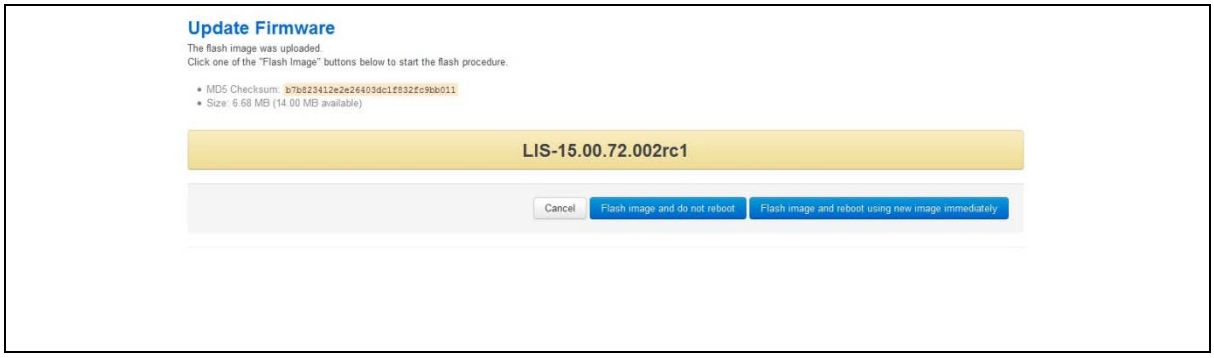



**Figure 31: The flash firmware - verify page**

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

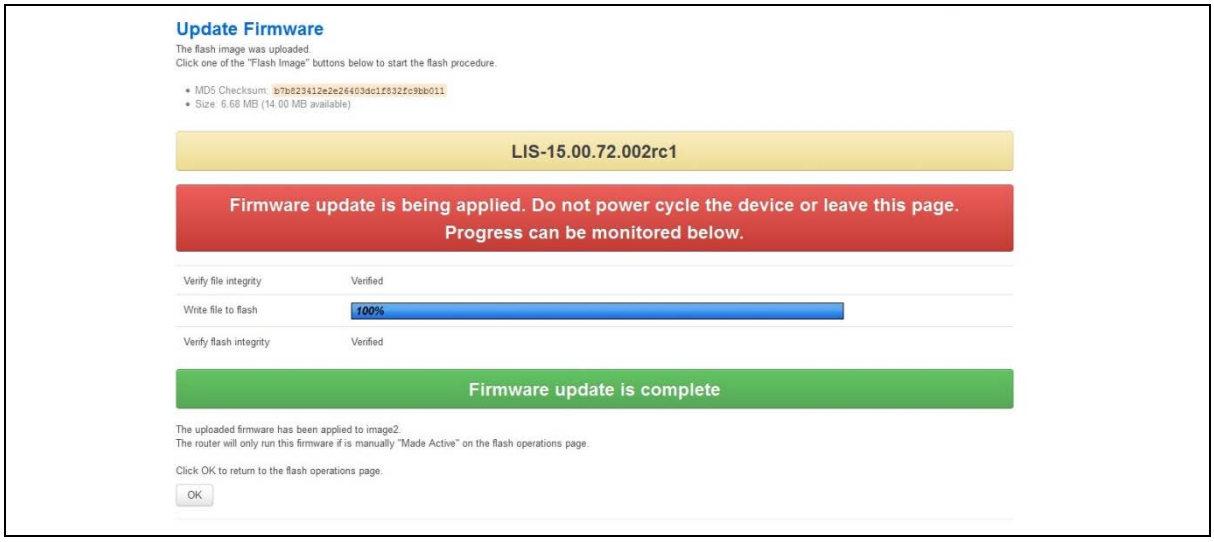



**Figure 32: The firmware update page after '…do not reboot' option selected**

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

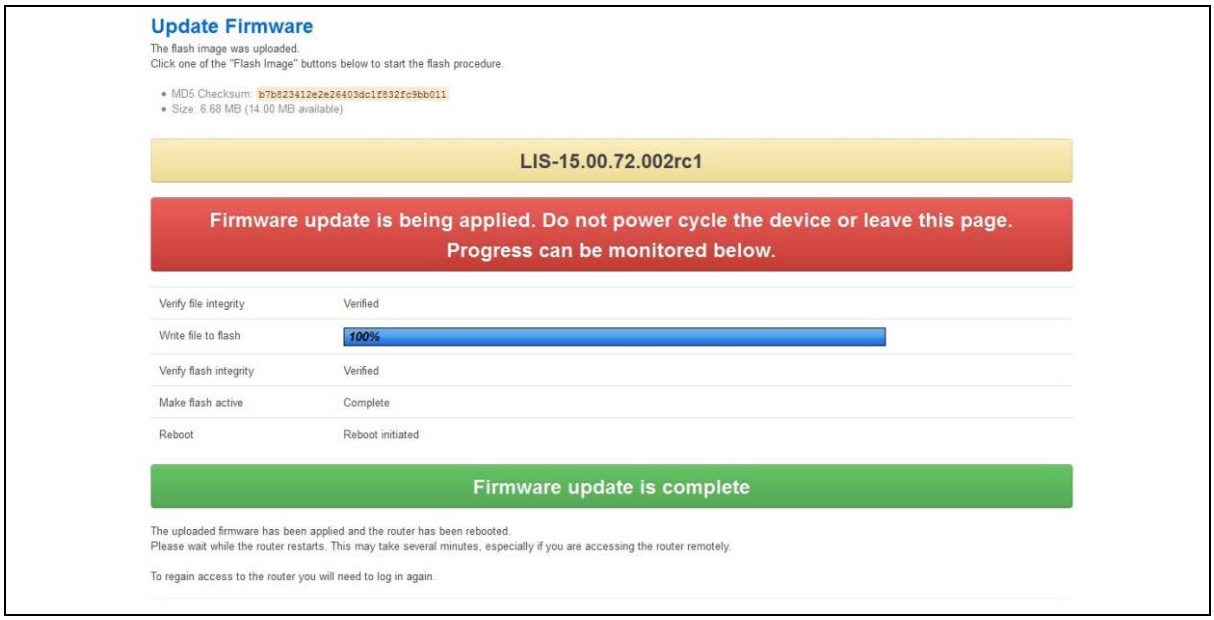



**Figure 33: The firmware update page after 'update flash image and reboot…' option selected**

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

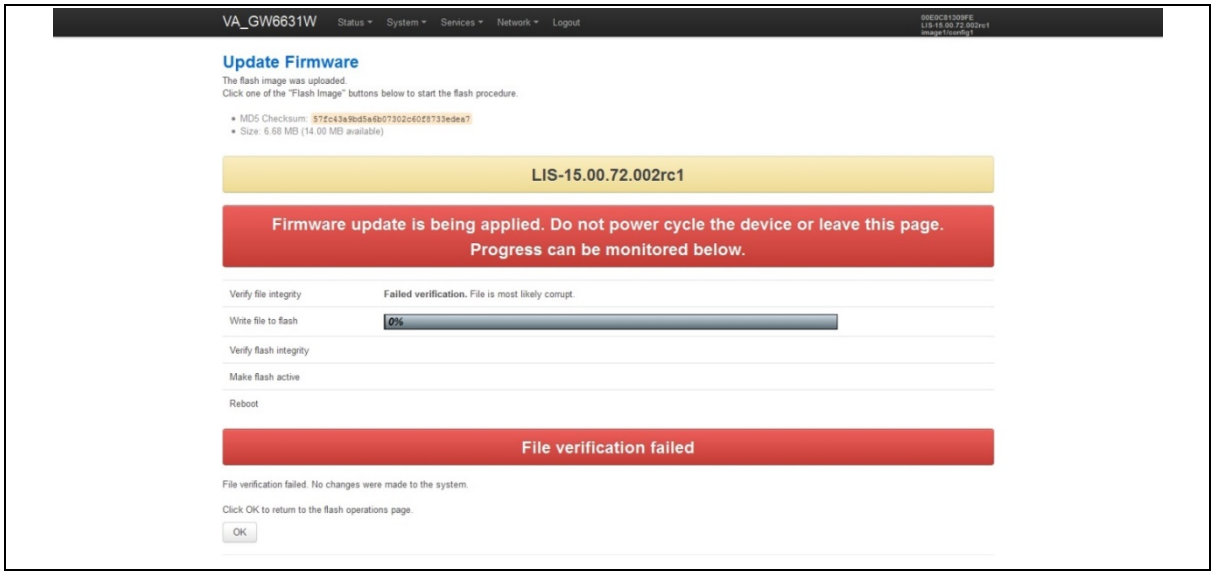



**Figure 34: The firmware update failure page**

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 35: 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

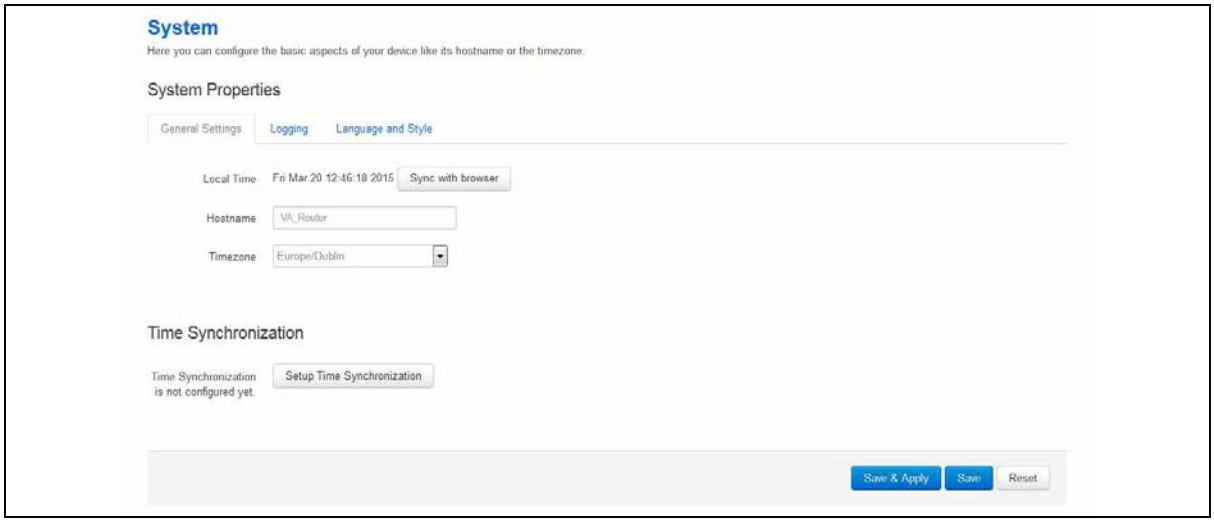



**Figure 36: General settings in system properties**

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

**Table 12: Information table for general settings section**

### 9.2.2 Logging

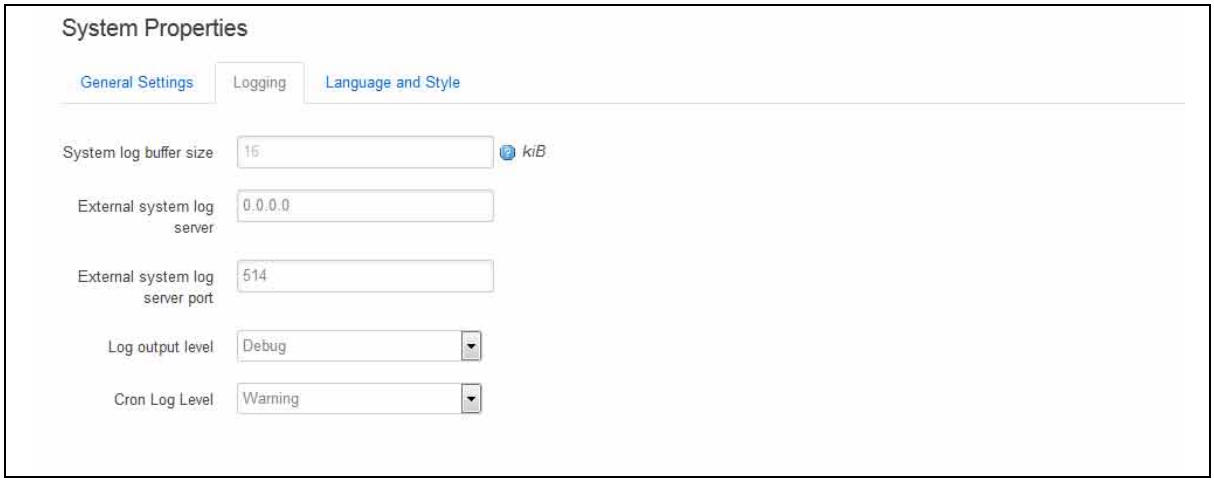



**Figure 37: The logging section in system properties**

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

<table>
<tr><td>Web: Log output level<br>UCI: system.main.conloglevel<br>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<br>UCI: system.main.cronloglevel<br>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<br>UCI: system.main.log_file<br>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<br>UCI: system.main.log_type<br>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<br>UCI: system.main.log_file_count<br>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 13: Information table for the logging section**

### 9.2.3 Language and style

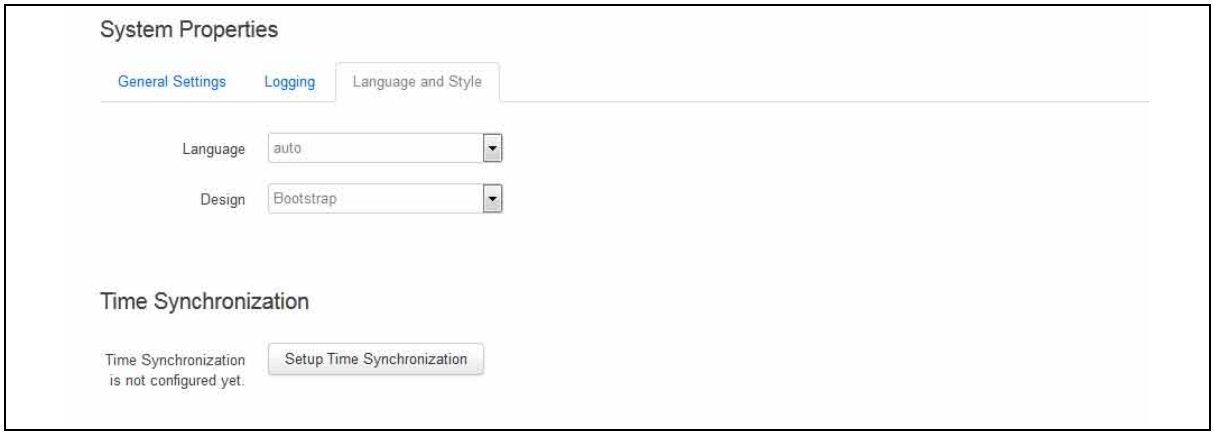



**Figure 38: The language and style section in system properties**

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

**Table 14: 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.

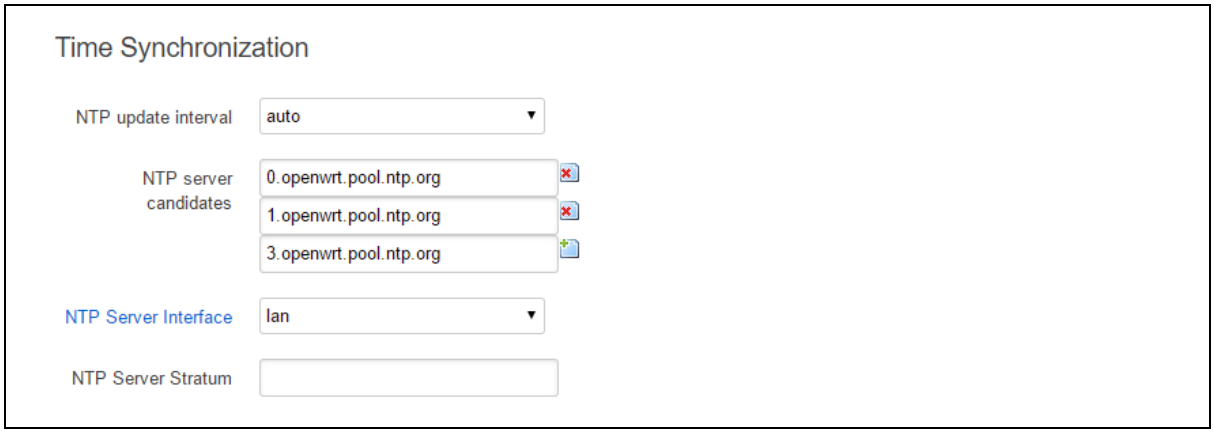



**Figure 39: The time synchronization section in system properties**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: NTP update interval<br>UCI: system.ntp.interval_hours<br>Opt: interval_hours | Specifies interval of NTP requests in hours. Default value set to auto.<br>Auto<br>Range: auto; 1-23 |
| Web: NTP server candidates<br>UCI: system.ntp.server<br>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.<br>By default all fields are set to 0.0.0.0. |
| Web: NTP Server Interface<br>UCI: system.ntp.listen<br>Opt: listen | Defines a list of interfaces that respond to NTP requests. Interfaces should be delimited using space. Example:<br>option listen 'LAN1 LAN2'<br>Blank: Do not respond to NTP requests.<br>Range |
| Web: NTP Server Stratum<br>UCI: system.ntp.stratum<br>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.<br>Blank: NTP server will derive stratum<br>Range |

**Table 15: 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.

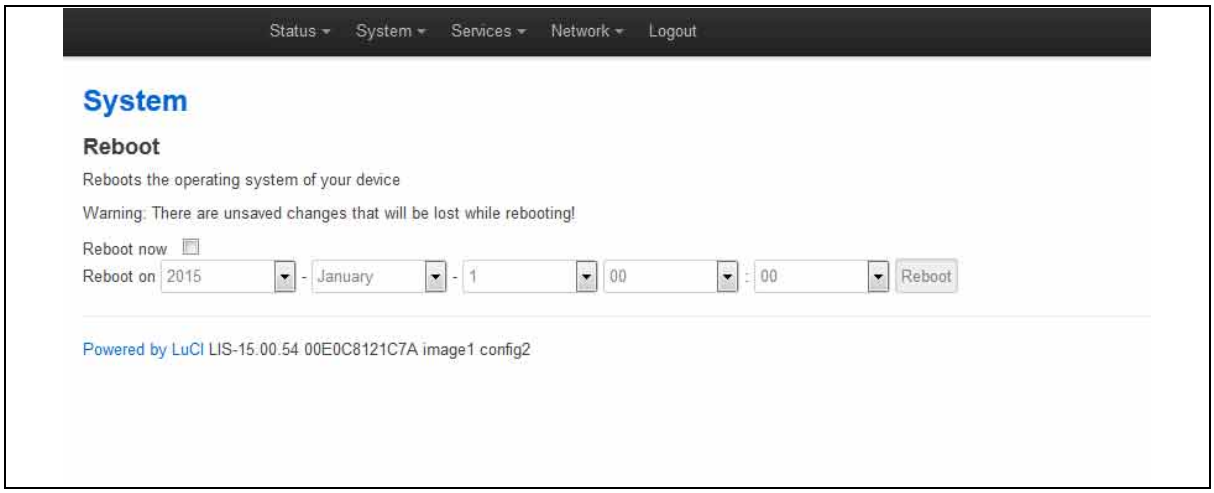



**Figure 40: The reboot page**

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.

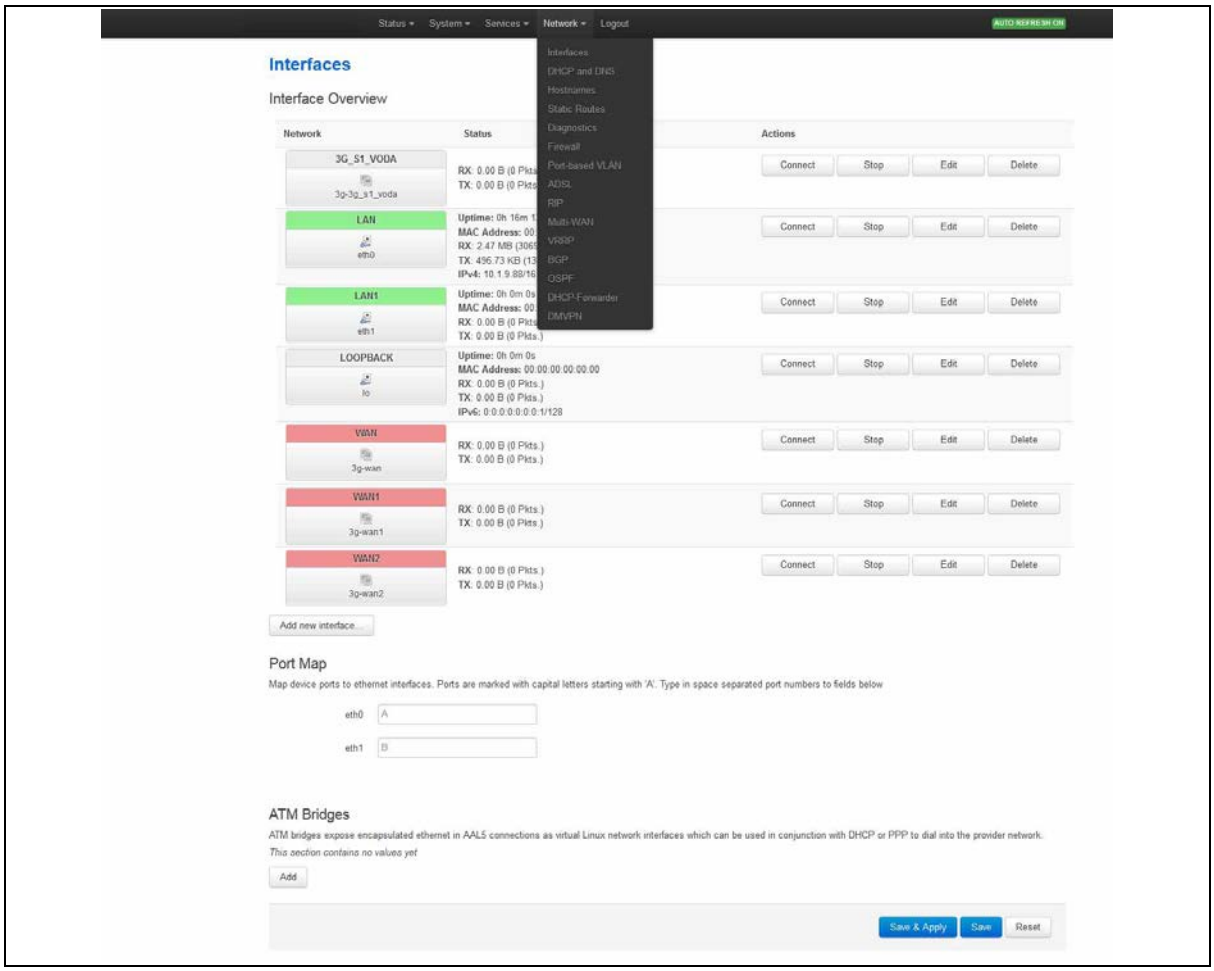



**Figure 41: The interfaces overview page**

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.

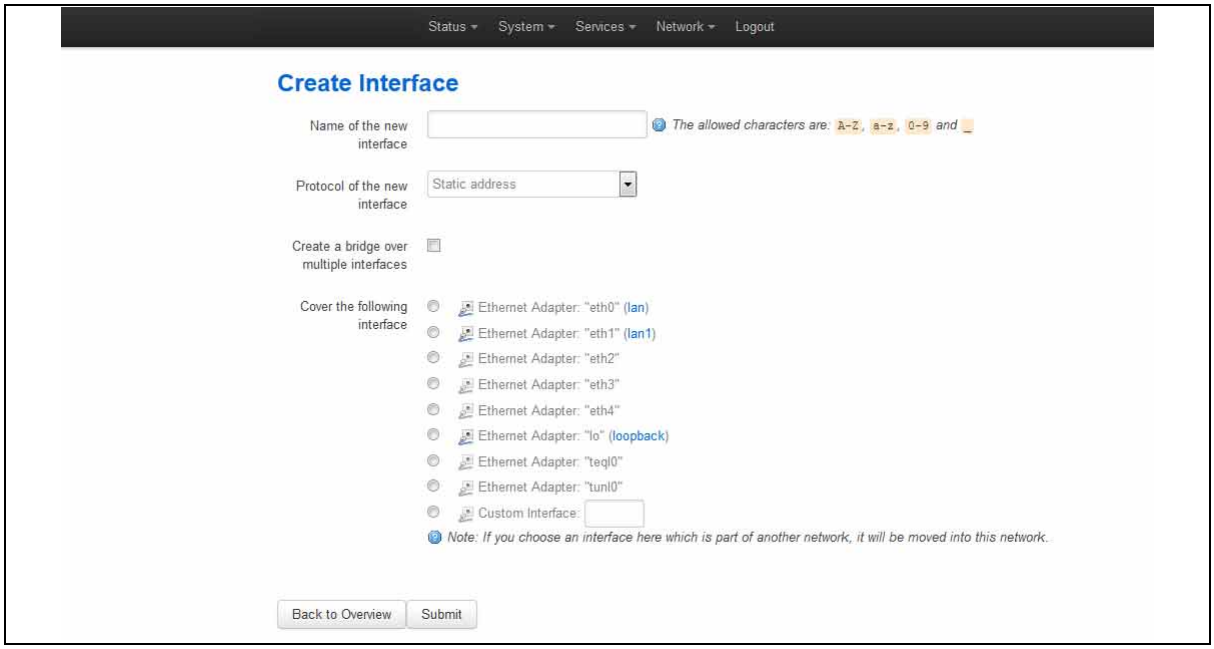



**Figure 42: The create interface page**

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

**Table 16: 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

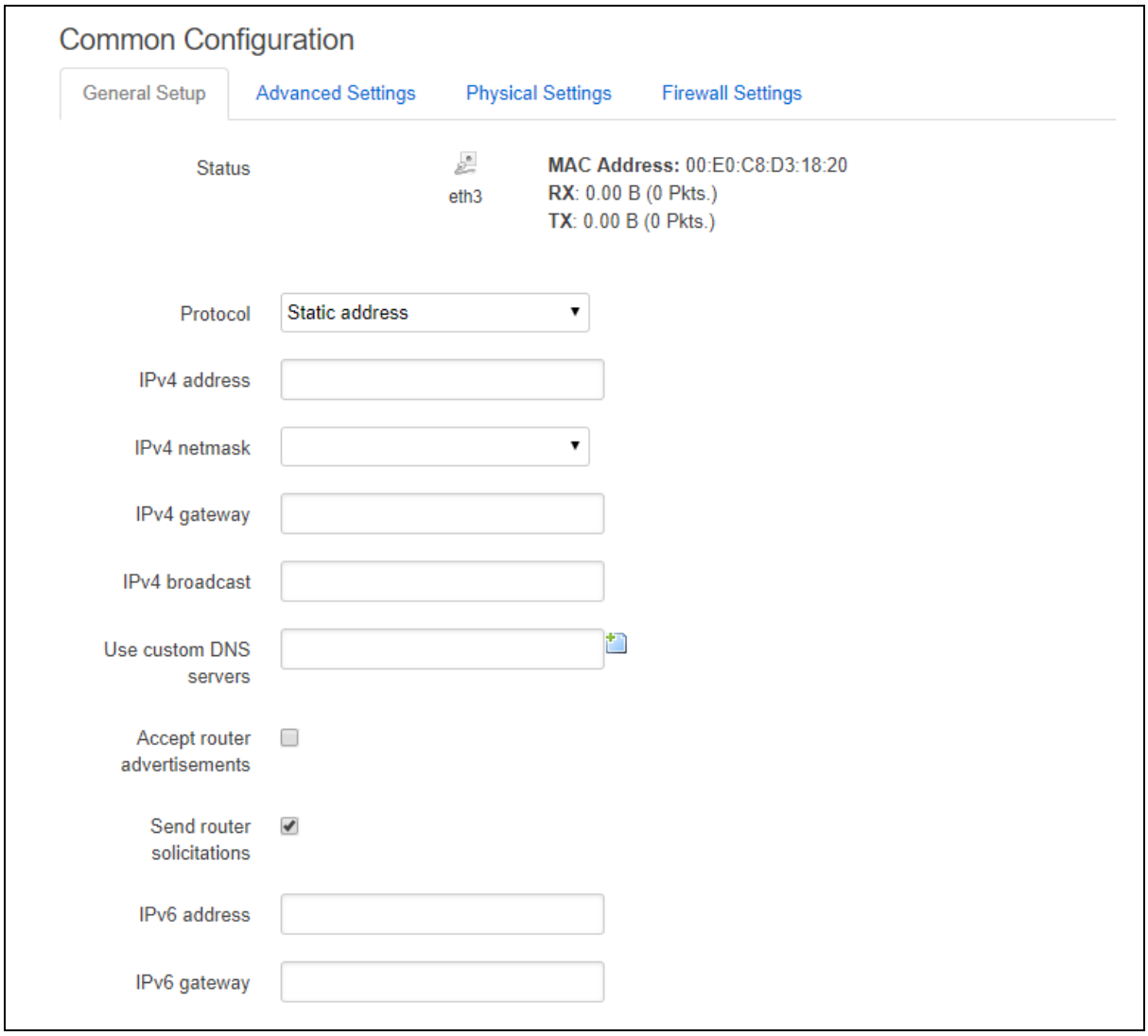



**Figure 43: The Ethernet connection common configuration settings page**

<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<br>UCI: network.<if name>.proto<br>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><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<br>UCI: network.<if name>.ipaddr<br>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<br>UCI: network.<if name>.netmask<br>Opt: netmask</td><td>Subnet mask to be applied to the IP address of this interface.</td></tr>
<tr><td>Web: IPv4 gateway<br>UCI: network.<if name>.gateway<br>Opt: gateway</td><td>IPv4 default gateway to assign to this interface (optional).</td></tr>
<tr><td>Web: IPv4 broadcast<br>UCI: network.<if name>.broadcast<br>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<br>UCI: network.<if name>.dns<br>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<br>UCI: network.<if name>.accept_ra<br>Opt: accept_ra</td><td>Specifies whether to accept IPv6 Router Advertisements on this interface (optional).<br>Note: default is 1 if protocol is set to DHCP, otherwise defaults to 0.</td></tr>
<tr><td>Web: Send router solicitations<br>UCI: network.<if name>.send_rs<br>Opt: send_rs</td><td>Specifies whether to send Router Solicitations on this interface (optional).<br>Note: defaults to 1 for Static protocol, otherwise defaults to 0.</td></tr>
<tr><td>Web: IPv6 address<br>UCI: network.<if name>.ip6addr<br>Opt: ip6addr</td><td>The IPv6 IP address of the interface. Optional if an IPv4 address is provided.<br>CIDR notation for the IPv6 address is required.</td></tr>
<tr><td>Web: IPv6 gateway<br>UCI: network.<if name>.ip6gw<br>Opt: ip6gw</td><td>Assign given IPv6 default gateway to this interface (optional).</td></tr>
</table>

**Table 17: Information table for LAN interface common configuration settings**

### 10.2.3.2 Common configuration: advanced settings

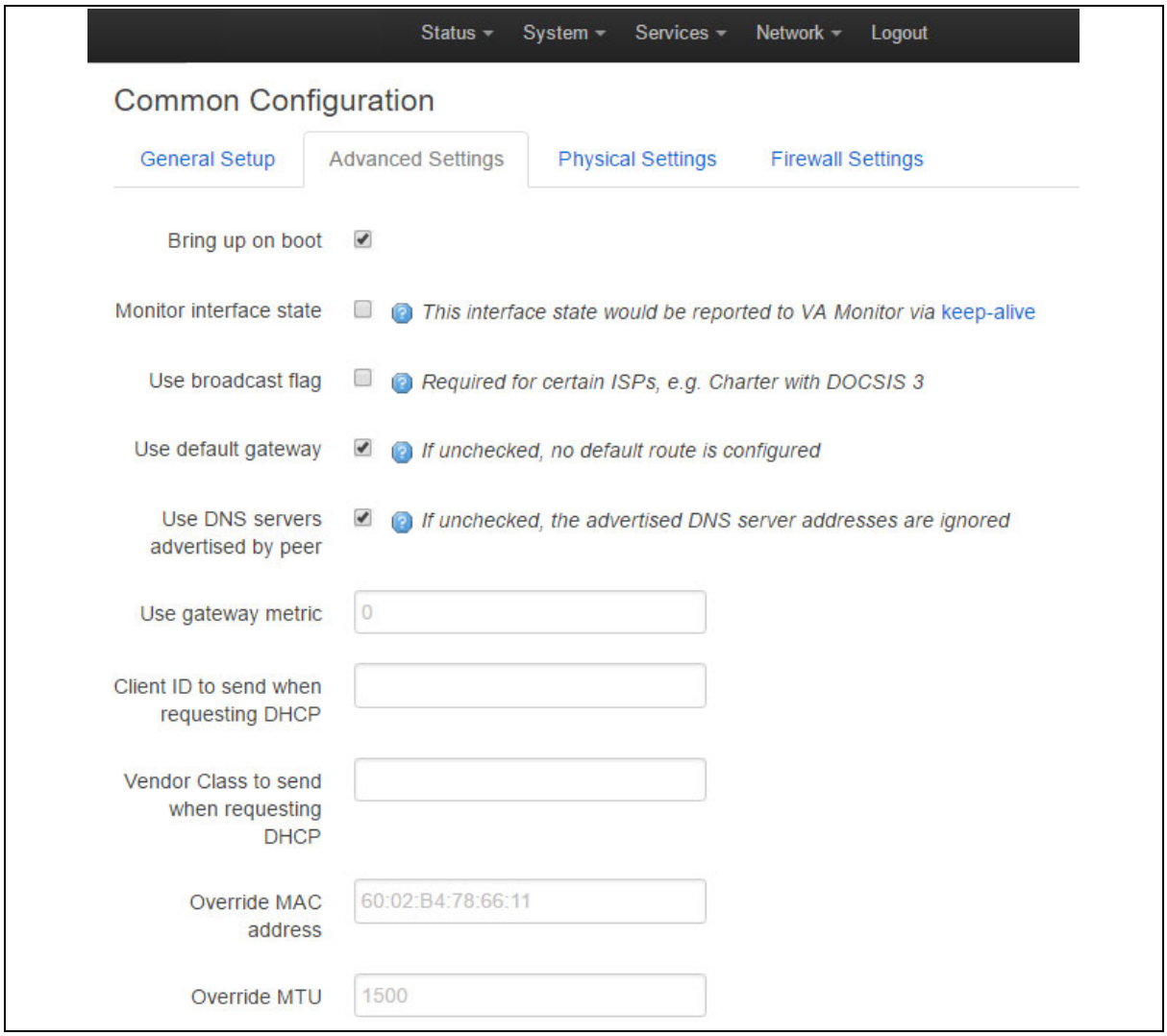



**Figure 44: The Ethernet connection advanced settings page**

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

| | |
|---|---|
| Web: Dependant Interfaces<br>UCI: network.[..x..].dependants<br>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.<br>Separate multiple interfaces by a space when using UCI.<br>Example: `option dependants 'PPPADSL MOBILE'`<br>This replaces the following previous options in child interfaces.<br>gre: option local_interface<br>lt2p: option src_ipaddr<br>iot: option wan1 wan2<br>6in4: option ipaddr<br>6to4: option ipaddr |
| Web: SNMP Alias ifindex<br>UCI: network.[..x..].snmp_alias_ifindex<br>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<br>Blank: No SNMP interface alias index<br>Range: 0 - 4294966295 |

**Table 18: Information table for common configuration advanced settings**

### 10.2.3.3 Common configuration: physical settings

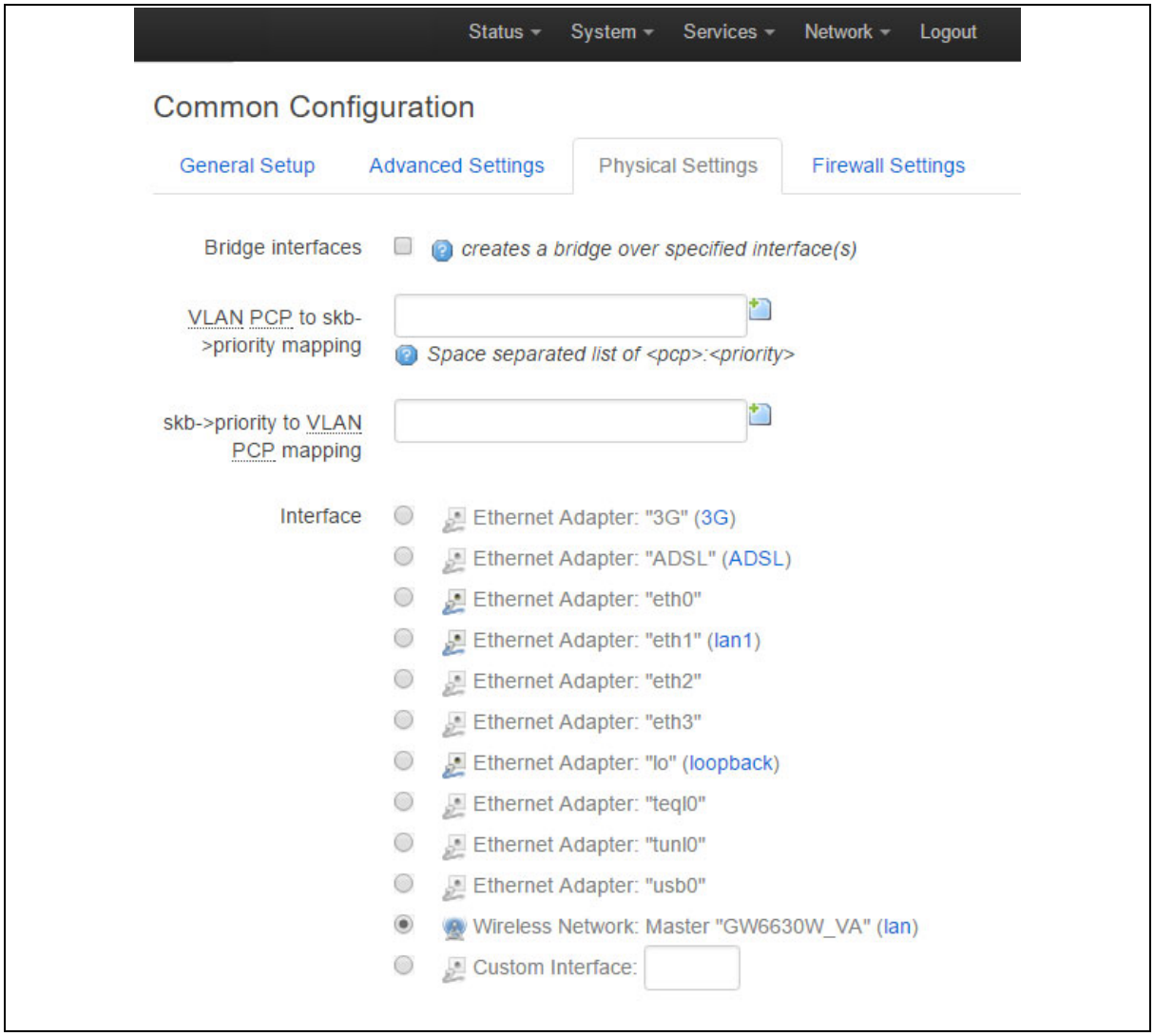



**Figure 45: The common configuration physical settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Bridge interfaces<br>UCI: network.<if name>.type<br>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.<br>Empty<br>Bridge — Configures a bridge over multiple interfaces. |
| Web: Enable STP<br>UCI: network.<if name>.stp<br>Opt: stp | Enable Spanning Tree Protocol. This option is only available when the Bridge Interfaces option is selected.<br>0 — Disabled.<br>1 — Enabled. |
| Web: VLAN PCP to skb>priority mapping<br>UCI: network.<if name>.vlan_qos_map_ingress<br>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.<br>Example: network.<if name>. vlan_qos_map_ingress =1:2 2:1 |
| Web: skb priority to >VLAN PCP mapping<br>UCI: network.<if name>.vlan_qos_map_egress<br>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.<br>Example: network.<if name>. vlan_qos_map_egress =1:2 2:1 |
| Web: Interface<br>UCI: network.<if name>.ifname<br>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.<br>Example: option ifname 'eth2 eth3' or network.<if name>.ifname=eth2 eth 3 |

**Table 19: 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.

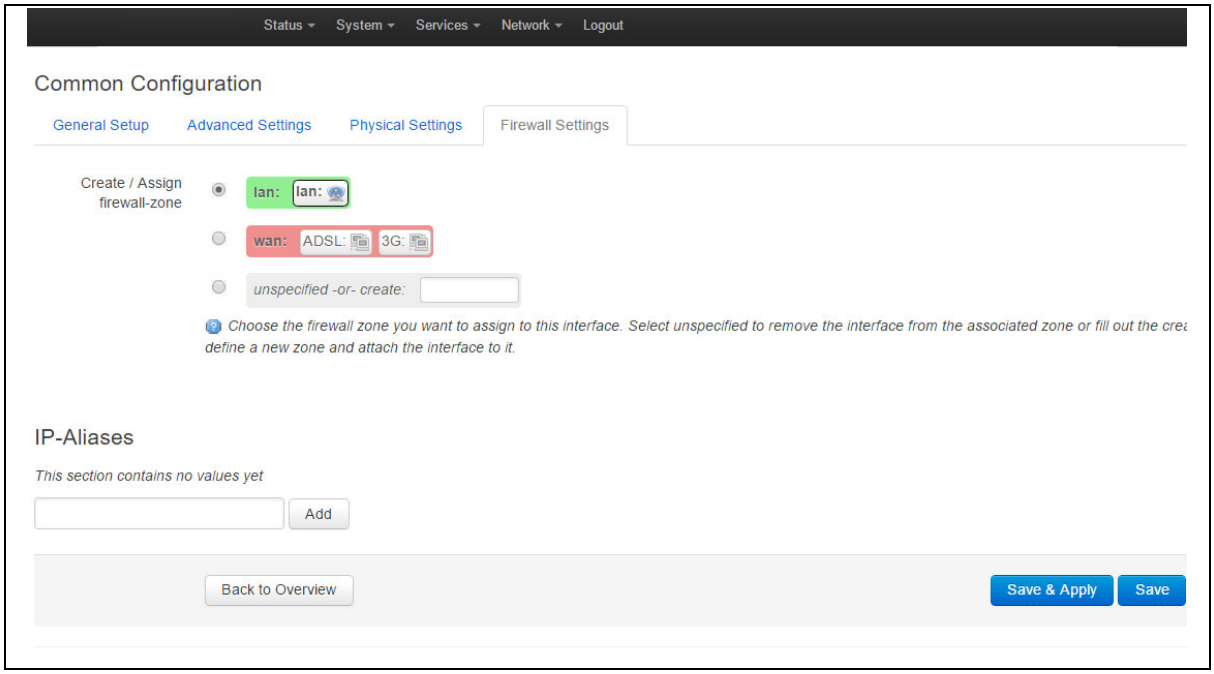



**Figure 46: GRE firewall settings**

## 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 47: The IP-Aliases section**

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

**Table 20: 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 48: The IP-Aliases general setup section**

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

**Table 21: Information table for IP-Alias general setup page**

#### 10.2.4.4 IP-aliases: advanced settings

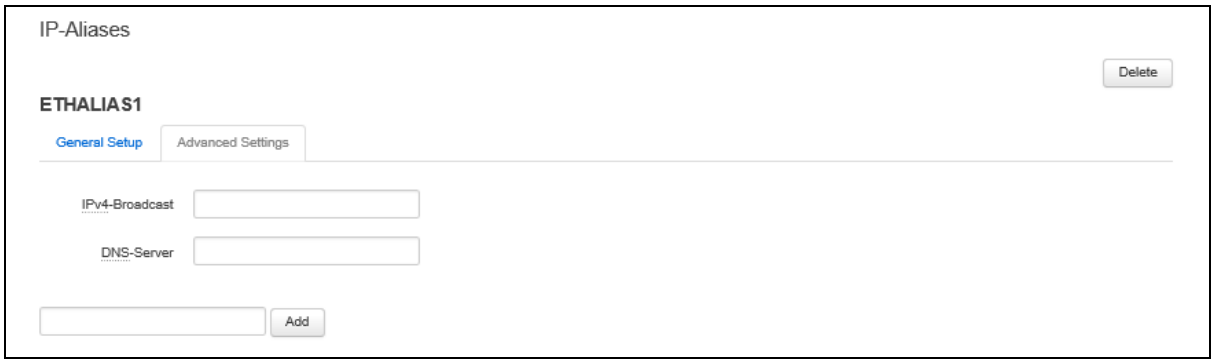



**Figure 49: The IP-Aliases advanced settings section**

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

**Table 22: 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 50: 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

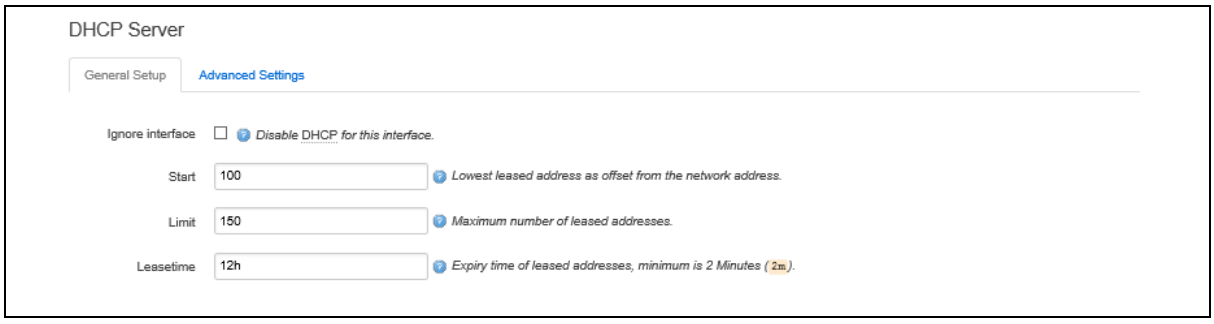



**Figure 51: The DHCP server general setup section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Ignore interface<br>UCI: dhcp.@dhcp[x].ignore<br>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.<br>0 – Disabled.<br>1 – Enabled. |
| Web: n/a<br>UCI: dhcp.@dhcp[x].start<br>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.<br>100<br>Range |
| Web: n/a<br>UCI: dhcp.@dhcp[x].limit<br>Opt: limit | Defines the offset from the network address for the end of the DHCP pool.<br>150<br>Range – 0 – 255 |
| Web: n/a<br>UCI: dhcp.@dhcp[x].leasetime<br>Opt: leasetime | Defines the lease time of addresses handed out to clients, for example 12h or 30m.<br>12h – 12 hours<br>Range |

**Table 23: Information table for DHCP server general setup page**

### 10.2.5.3 DHCP server: advanced settings

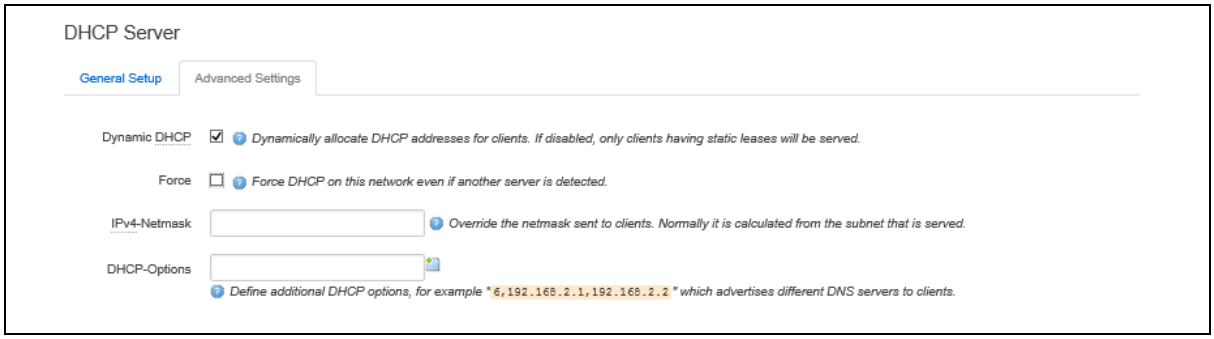



**Figure 52: The DHCP server advanced settings section**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Dynamic DHCP<br>UCI: dhcp.@dhcp[x].dynamicdhcp<br>Opt: dynamicdhcp | Defines whether to allocate DHCP leases.<br>1 — Dynamically allocate leases.<br>0 — Use /etc/ethers file for serving DHCP leases. |
| Web: Force<br>UCI: dhcp.@dhcp[x].force<br>Opt: force | Forces DHCP serving on the specified interface even if another DHCP server is detected on the same network segment.<br>0 — Disabled.<br>1 — Enabled. |
| Web: DHCP-Options<br>UCI: dhcp.@dhcp[x].dhcp_option<br>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. Options that contain multiple vales should be separated by a space.<br>Example: list dhcp_option 6,192.168.2.1 192.168.2.2<br>No options defined.<br>Syntax — Option_number, option_value |
| Web: n/a<br>UCI: dhcp.@dhcp[x].networkid<br>Opt: networked | Assigns a network-id to all clients that obtain an IP address from this pool. |

**Table 24: 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/firewall** 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**.

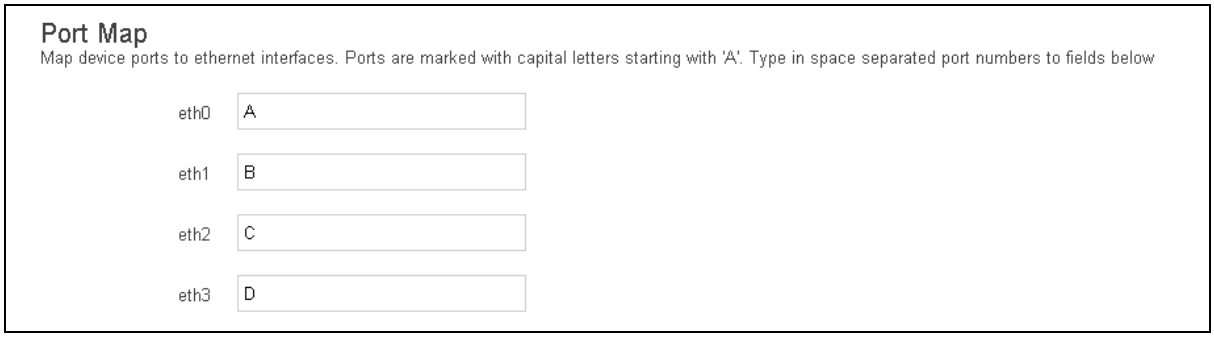



**Figure 53: The Interface port map section**

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

**Table 25: 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.

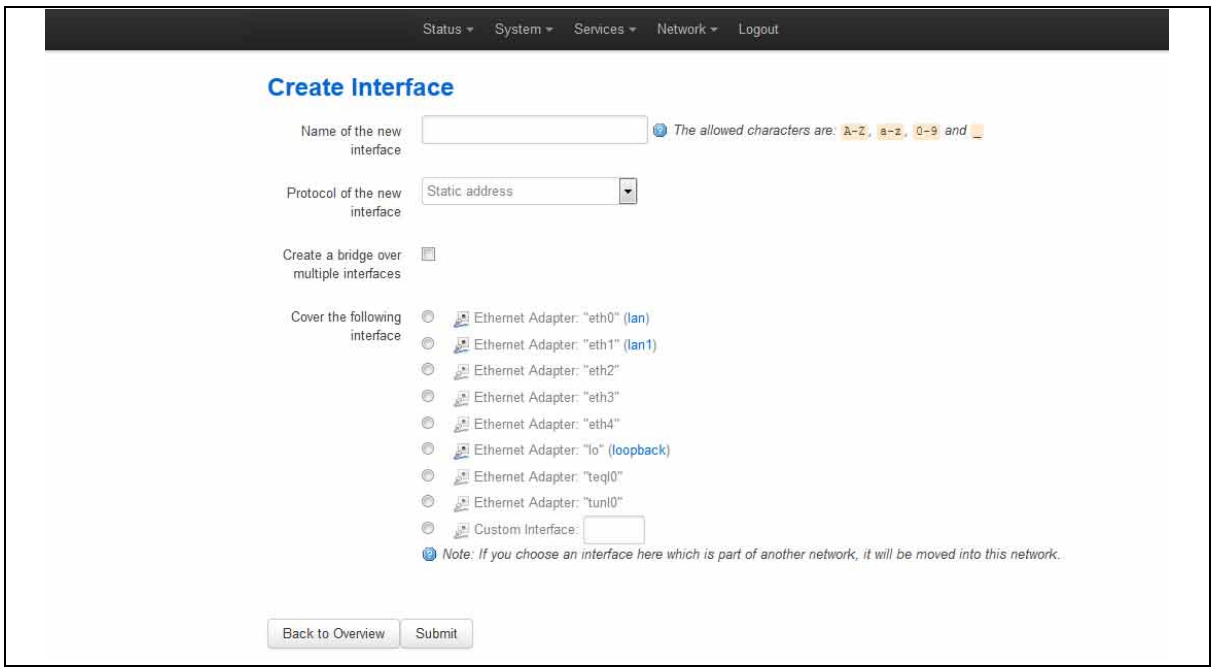



**Figure 54: The create interface page**

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface<br>UCI: network.vlan1=interface<br>Opt: interface</td><td>Type the name of the new interface. For example, VLAN1.</td></tr>
<tr><td>Web: Protocol of the new interface<br>UCI: network.vlan_test.proto<br>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<br>UCI: network.vlan1.type<br>Opt: type</td><td>Create a bridge over multiple interfaces.</td></tr>
<tr><td>Web: Cover the following interface<br>UCI: network.vlan1.ifname<br>Opt: ifname</td><td>Check the Custom Interface radio button.<br>Enter a name, for example eth0.100. This will assign VLAN 100 to the eth0 interface.</td></tr>
</table>

**Table 26: Information table for the create interface page**

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

### 11.3.2 General setup: VLAN

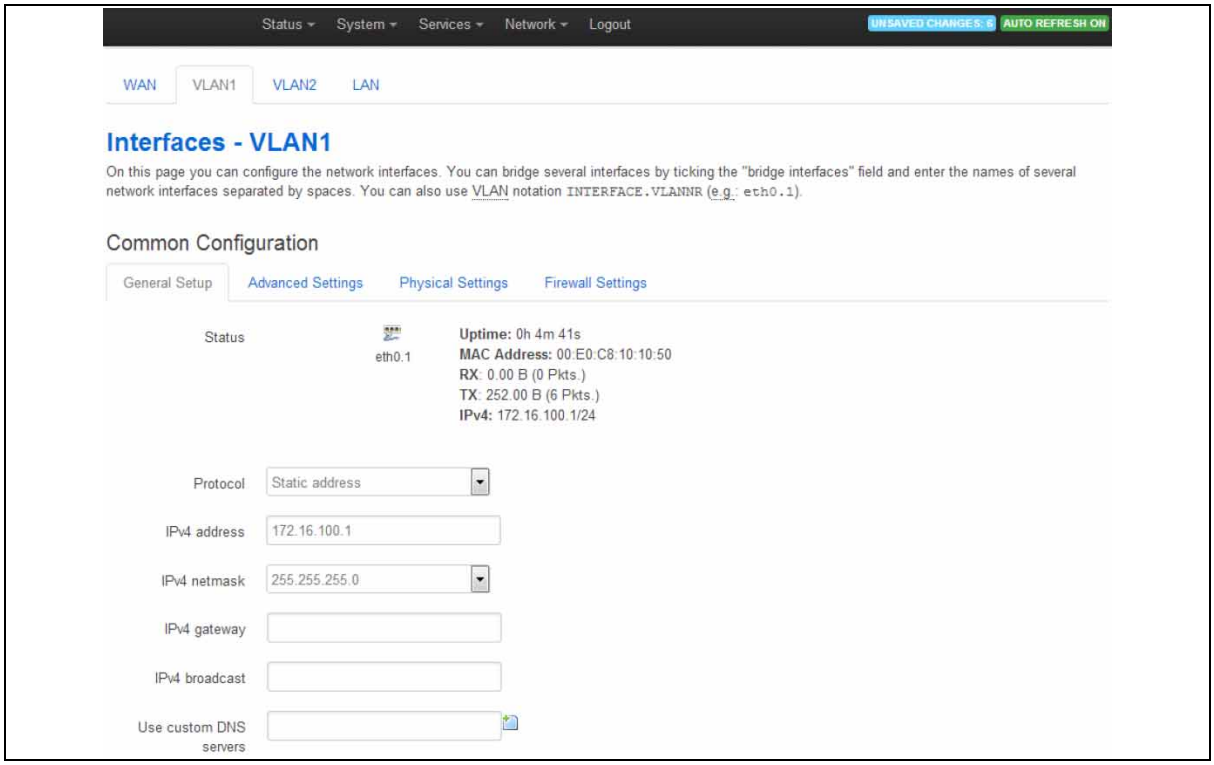



**Figure 55: The VLAN 1 interface page**

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

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

**Table 27: 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.

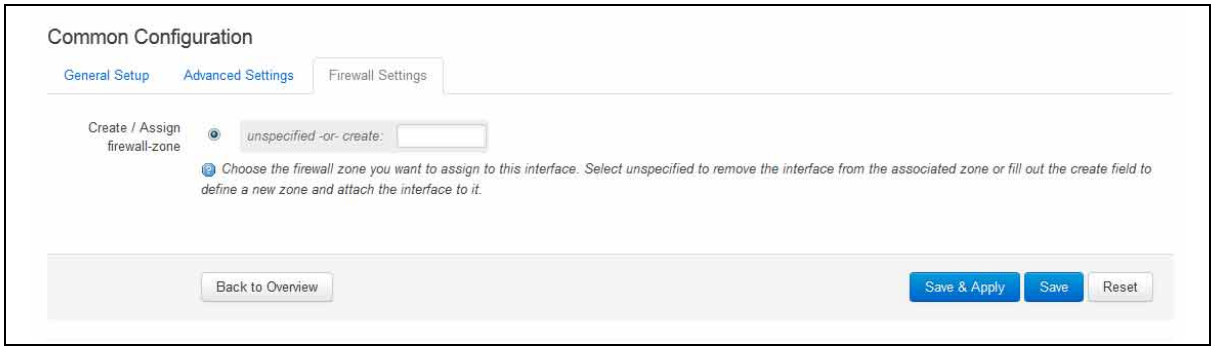



**Figure 56: Firewall settings page**

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.

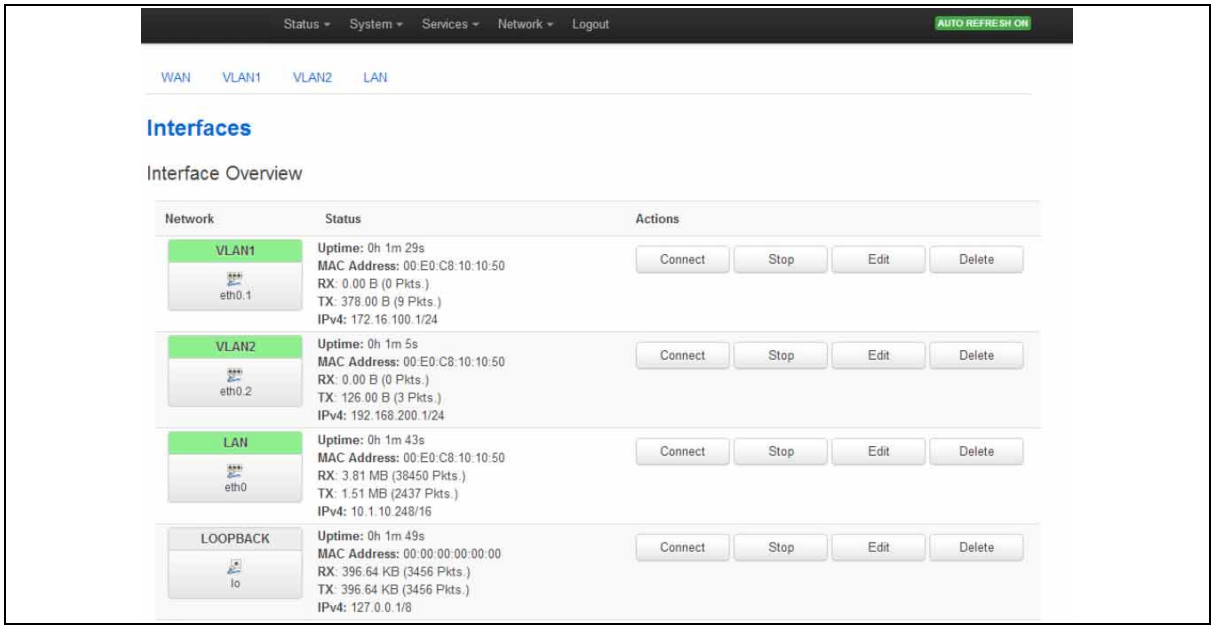



**Figure 57: The interface overview page showing two VLAN interfaces**

## 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.

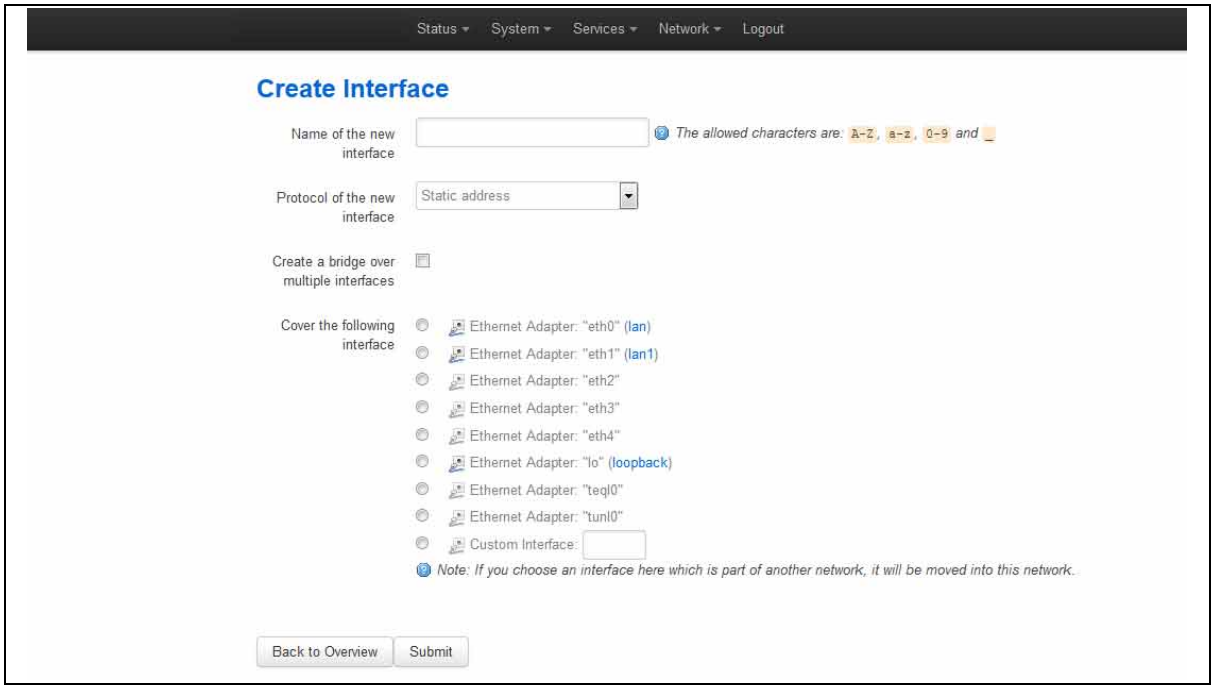



**Figure 58: The create interface page**

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

**Table 28: 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

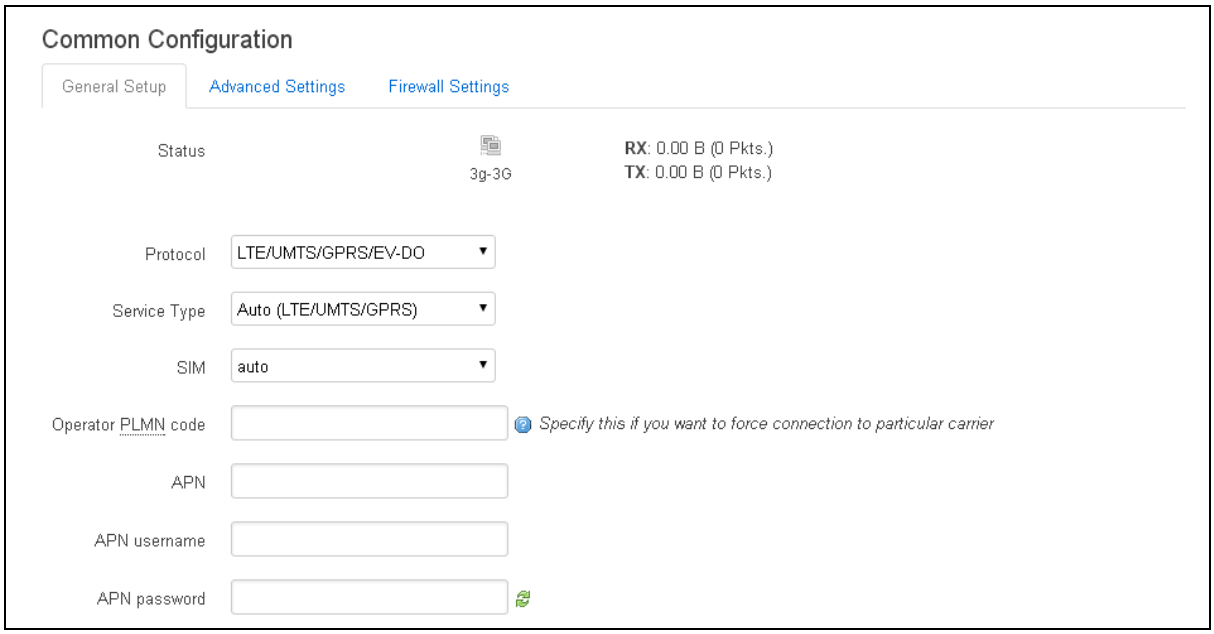



**Figure 59: The common configuration page**

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

<table>
<tr><td>Web: n/a<br>UCI: network.3G.service_order<br>Opt: service_order</td><td>Defines a space separated list of services, in preferred order. Valid options are <code>gprs, umts, lte, auto.</code><br>If no valid_service order is defined, then the configured Service Type is used. Example:<br><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<br>UCI: network.3G.operator<br>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.<br>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.<br>See below for alternative options for the operator format option.</td></tr>
<tr><td>Web: n/a<br>UCI: network.3G.opformat<br>Opt: opformat</td><td>Defines the operator format. We recommended you use PLMN code.<br>The operator is case sensitive so if using long or short character format it must match the operator exactly.<br>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<br>UCI: network.3G.sim<br>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<br>UCI: network.3G.apn<br>Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: APN username<br>UCI: network.3G.username<br>Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: APN password<br>UCI: network.3G.password<br>Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: n/a<br>UCI: network.3G.retry_interval_sec<br>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.<br>retry_interval_sec 60 180</td></tr></table></td></tr>
</table>

**Table 29: 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

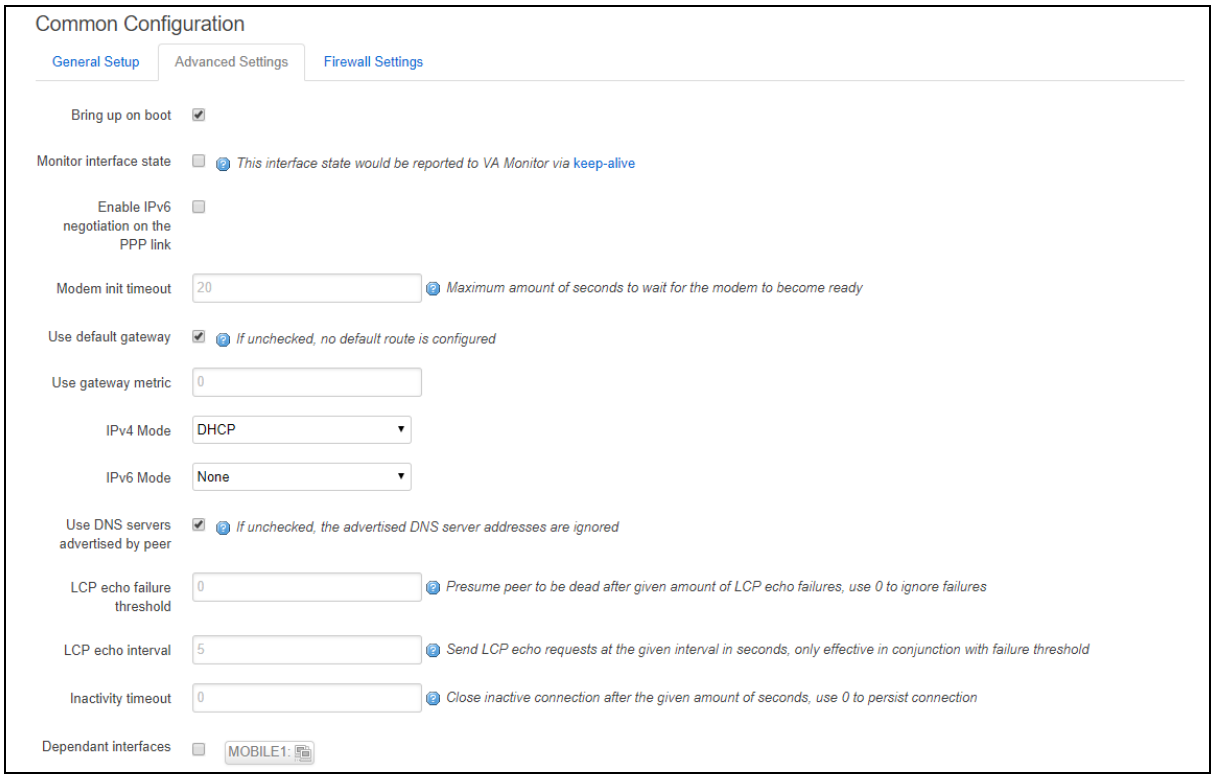



**Figure 60: The advanced settings tab**

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

<table>
<tr><td>Web: IPv4 Mode<br>UCI: network.3G.ipv4mode<br>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<br>UCI: network.3G.ipv6mode<br>Opt: ipv6mode</td><td>Defines the IPv6 address assignment approach for mobile interfaces in Ethernet Mode.<br>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<br>UCI: network.3G.peerdns<br>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<br>UCI: network.3G.dns<br>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.<br>Example:<br><code>uci set network.3G.dns='1.1.1.1 2.2.2.2'</code></td></tr>
<tr><td>Web: LCP echo failure threshold<br>UCI: network.3G.keepalive<br>Opt: keepalive</td><td>Presumes peer to be dead after a given amount of LCP echo failures, use 0 to ignore failures.<br>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><br>Example:<br><code>uci set network.3G.keepalive=15 10</code></td></tr>
<tr><td>Web: LCP echo internal<br>UCI: network.3G.keepalive<br>Opt: keepalive</td><td>Send LCP echo requests at the given interval in seconds, only effective in conjunction with failure<br>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><br>Example:<br><code>uci set network.3G.keepalive=15 10</code></td></tr>
<tr><td>Web: Inactivity timeout<br>UCI: network.3G.demand<br>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<br>UCI: network.3G.dependants<br>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.<br>Separate multiple interfaces by a space when using UCI. Example: <code>option dependants 'PPPADSL MOBILE'</code><br>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>

| | |
|---|---|
| Web: SNMP Alias ifindex<br>UCI: network.[..x..].snmp_alias_ifindex<br>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.<br>Blank: No SNMP interface alias index<br>Range: 0 - 4294966295 |

**Table 30: 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 61: 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.

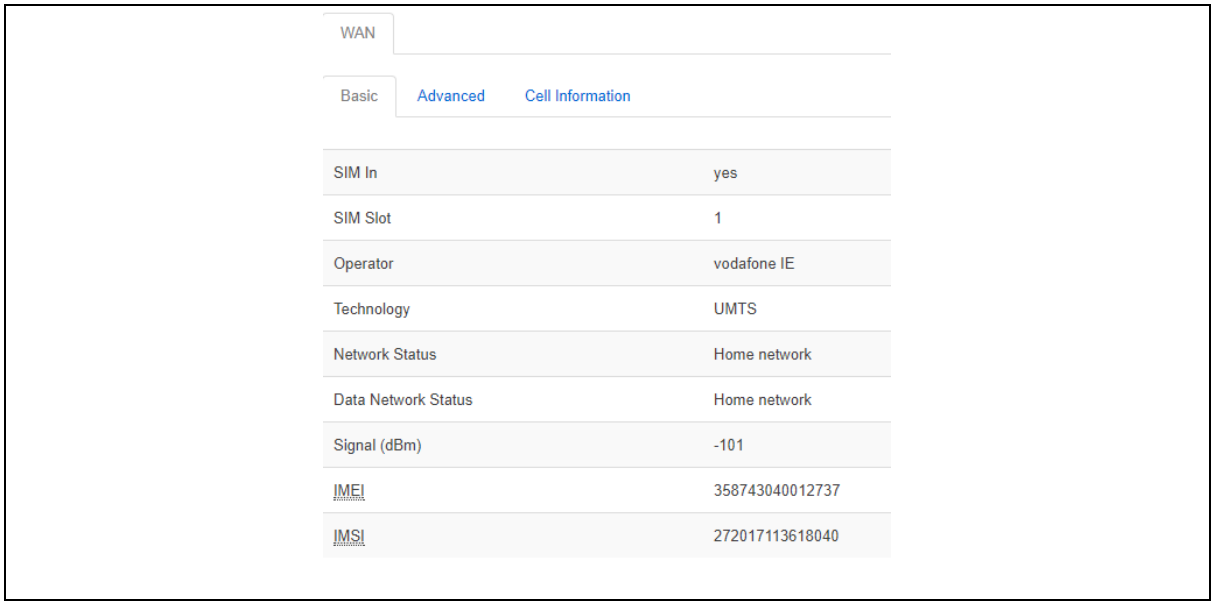



**Figure 62: The mobile information page**

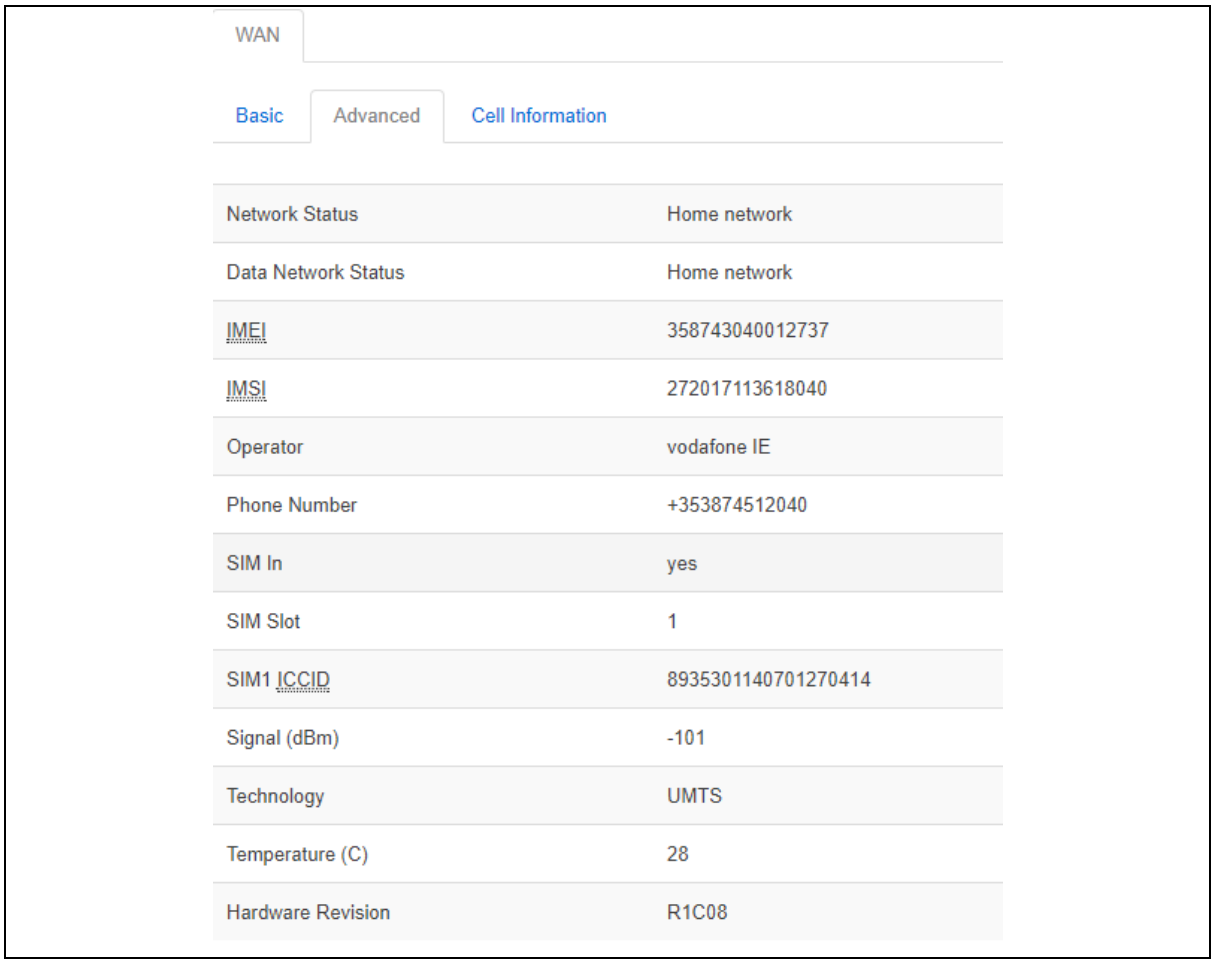



**Figure 63: The advanced information page**

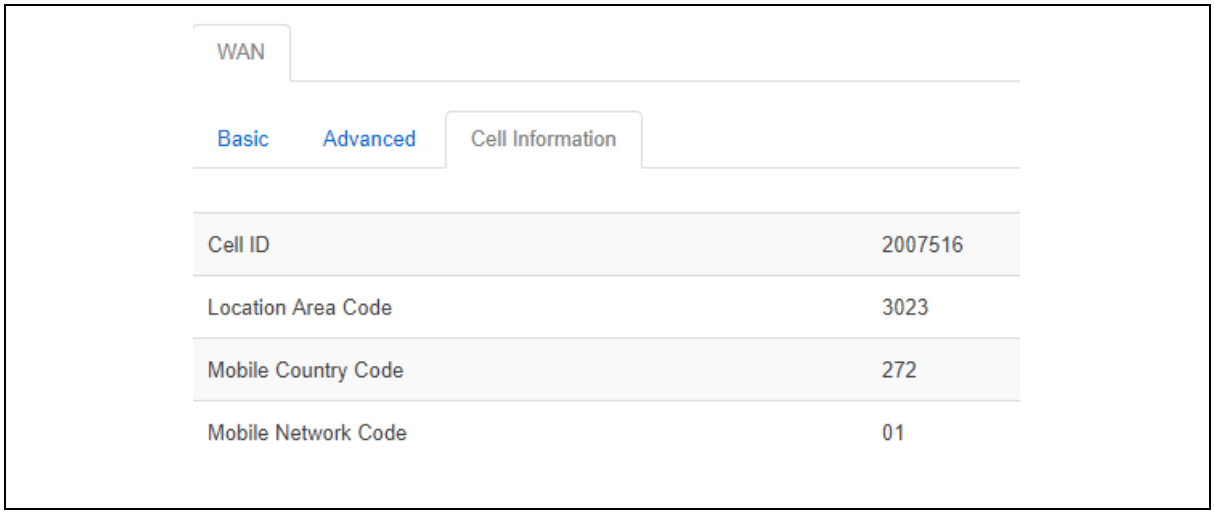



**Figure 64: The cell information page**

### 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          : 8935301140701270414
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

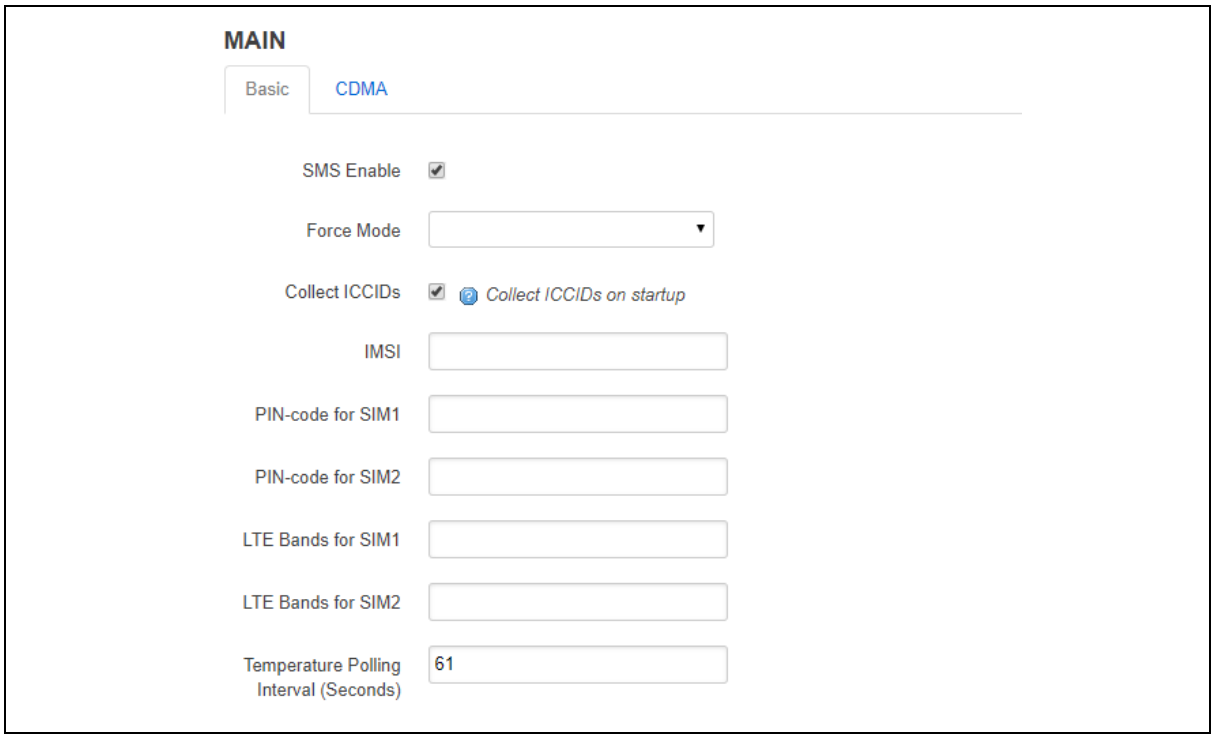



**Figure 65: The mobile manager basic page**

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

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

**Table 31: 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.

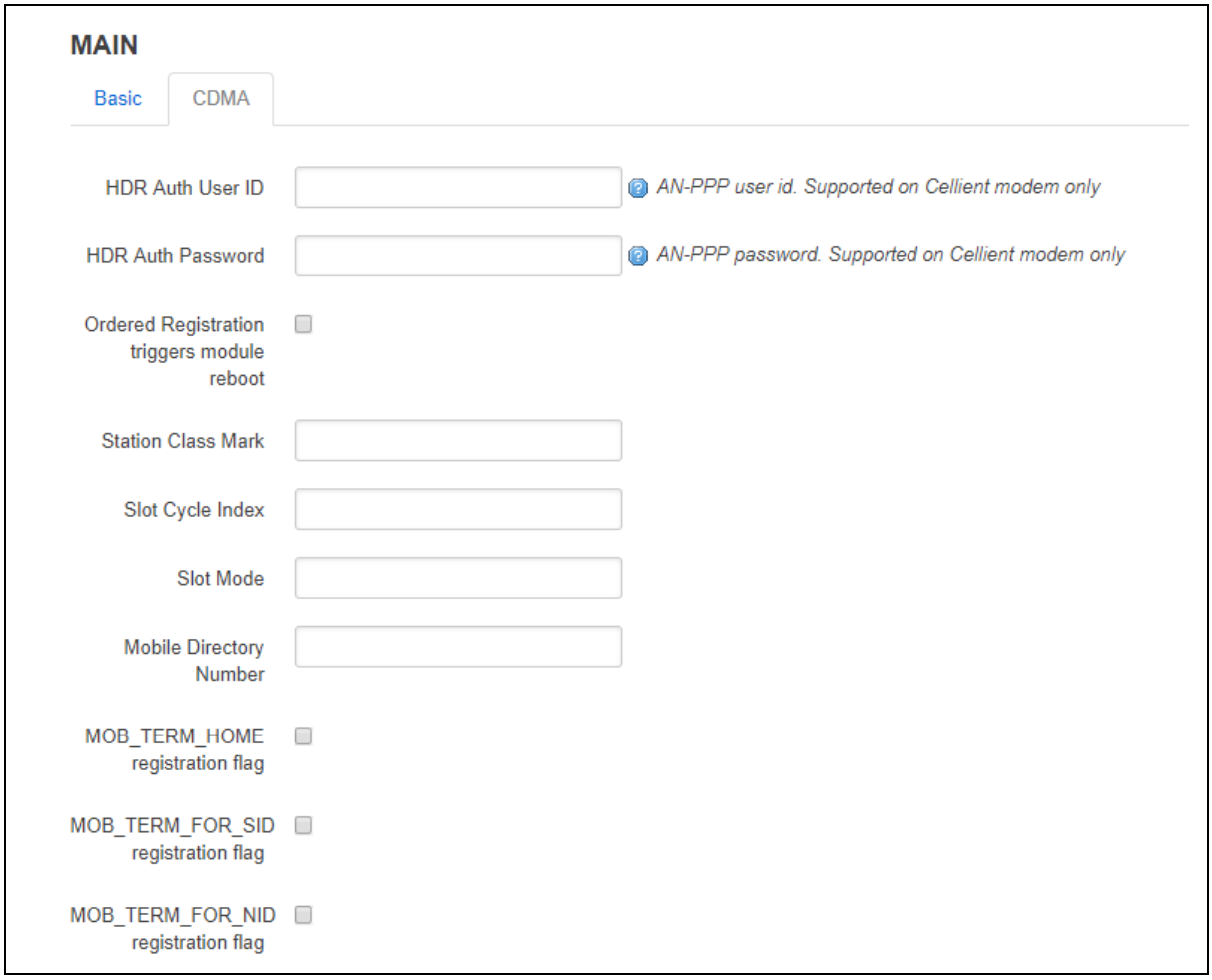



**Figure 66: The mobile manager CDMA page**

| Web Field / UCI / Package Option | Description | |
|---|---|---|
| Web: HDR Auth User ID<br>UCI: mobile.main.hdr_userid<br>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<br>UCI: mobile.main.hdr_password<br>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<br>UCI: mobile.main. mobile.main.cdma_ordered_registration_reboot_enabled<br>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<br>UCI: mobile.main.cdma_station_class_mark<br>Opt: cdma_station_class_mark | Allows the station class mark for the MS to be changed. | |
| | 58 | |
| | 0-255 | |
| Web: Slot Cycle Index<br>UCI: mobile.main.cdma_slot_cycle_index<br>Opt: cdma_slot_cycle_index | The desired slot cycle index if different from the default. | |
| | 2 | |
| | 0-7 | |
| Web: Slot Mode<br>UCI: mobile.main.cdma_slot_mode<br>Opt: cdma_slot_mode | Specifies the slot mode. | |
| | 0 | |
| | | |
| Web: Mobile Directory Number<br>UCI: mobile.main.cdma_mobile_directory_number<br>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<br>UCI: mobile.main. cdma_mob_term_home_registration_flag<br>Opt: cdma_mob_term_home_registration_flag | The MOB_TERM_HOME registration flag | |
| | 0 | Disabled. |
| | 1 | Enabled. |
| Web: MOB_TERM_FOR_SID registration flag<br>UCI: mobile.main. cdma_mob_term_for_sid_registration_flag<br>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<br>UCI: mobile.main. cdma_mob_term_for_nid_registration_flag<br>Opt: cdma_mob_term_for_nid_registration_flag | The MOB_TERM_FOR_NID registration flag | |
| | 0 | Disabled. |
| | 1 | Enabled. |
| Web: Access Overload Control<br>UCI: mobile.main.cdma_access_overload_control<br>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<br>UCI: mobile.main.cdma_preferred_serving_system<br>Opt: cdma_preferred_serving_system | The CDMA Preferred Serving System(A/B) | |
| | 5 | |
| | | |
| Web: Digital Analog Mode Preference<br>UCI: cdma_digital_analog_mode_preference<br>Opt: cdma_digital_analog_mode_preference | Digital/Analog Mode Preference. | |
| | 4 | |
| | | |

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

**Table 32: Information table for mobile manager CDMA settings**

### 13.2.3 Mobile manager: callers

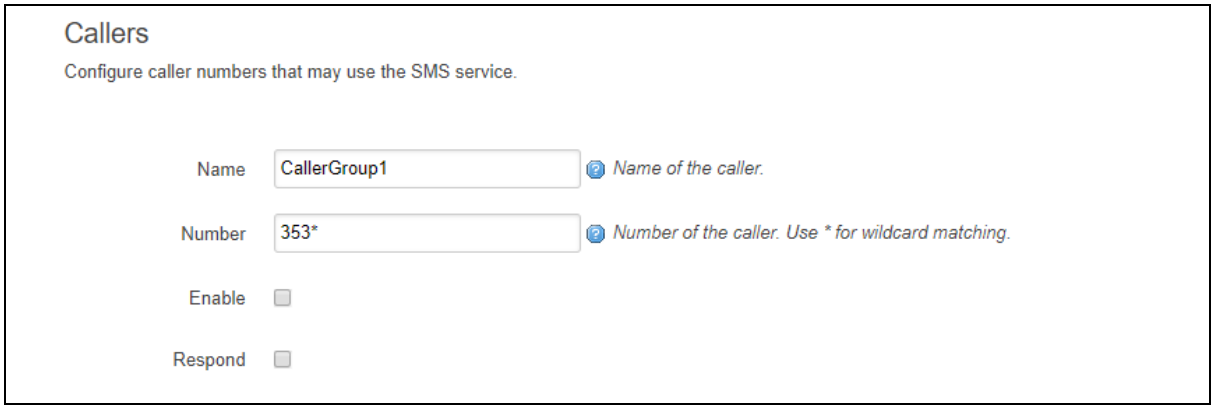



**Figure 67: The mobile manager CDMA page**

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

**Table 33: 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.

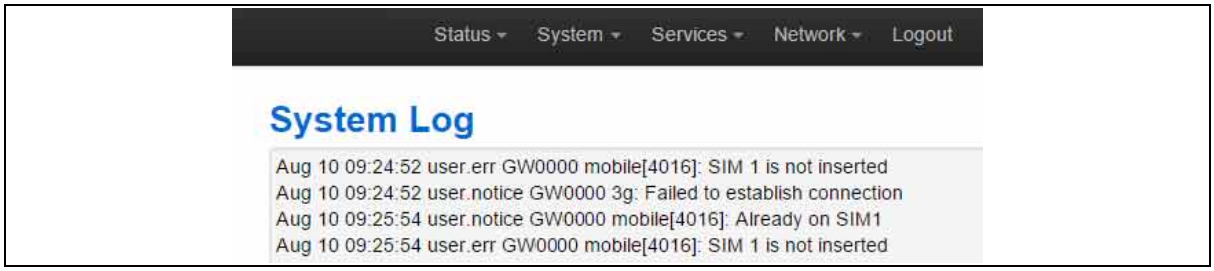



**Figure 68: Example of output from system log**

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.

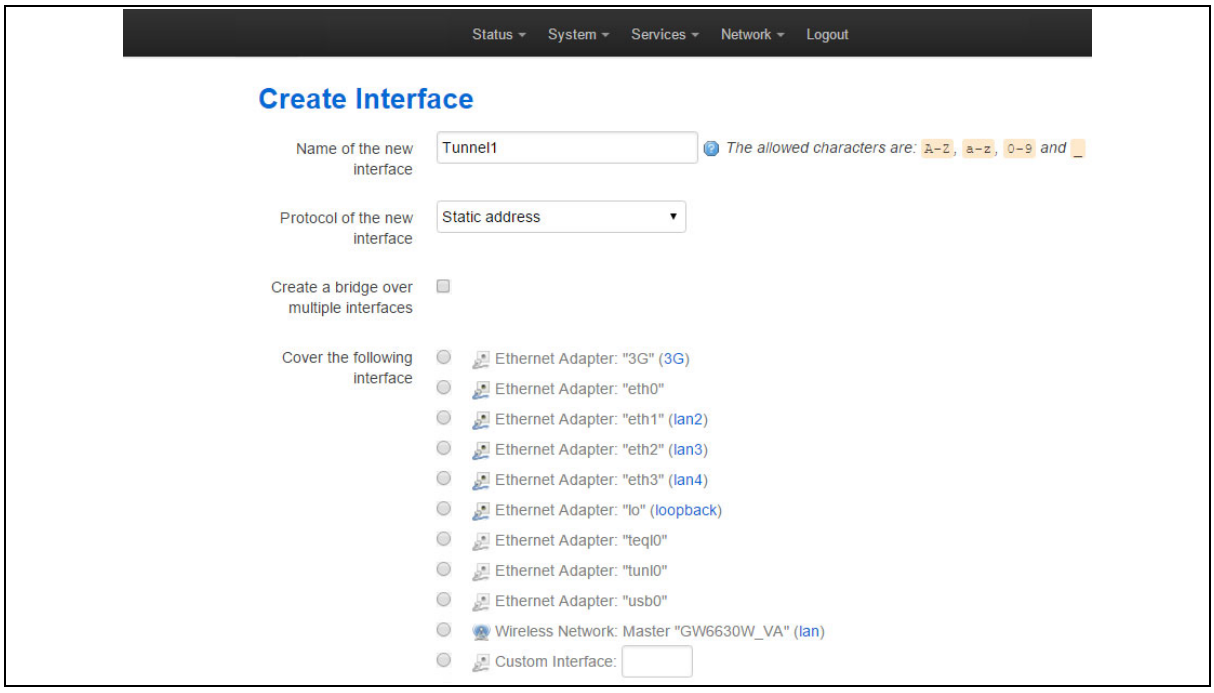



**Figure 69: The create interface page**

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

**Table 34: 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

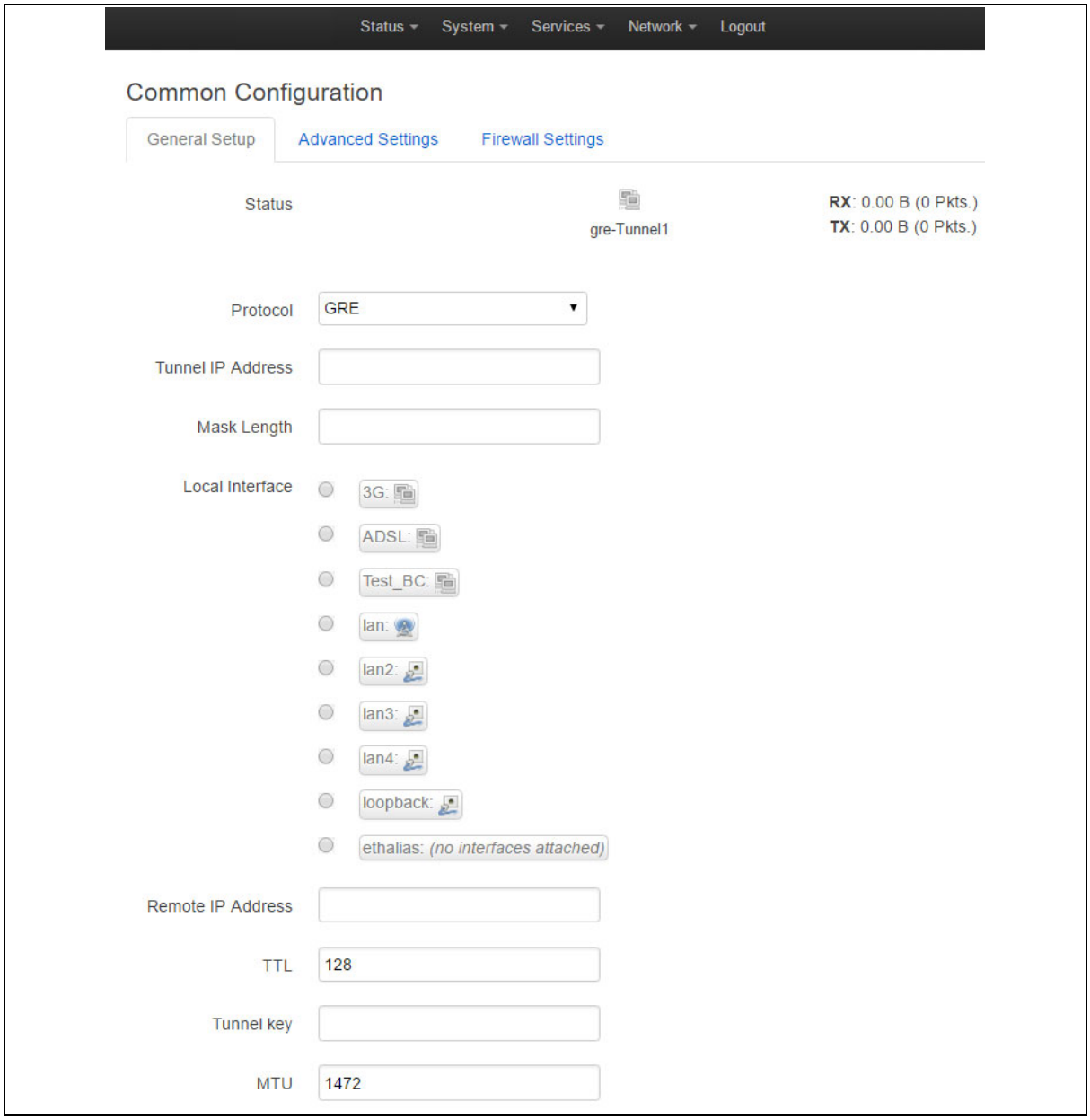



**Figure 70: The GRE common configuration page**

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

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

**Table 35: Information table for GRE**

## 14.2.2 GRE connection: common configuration-advanced settings

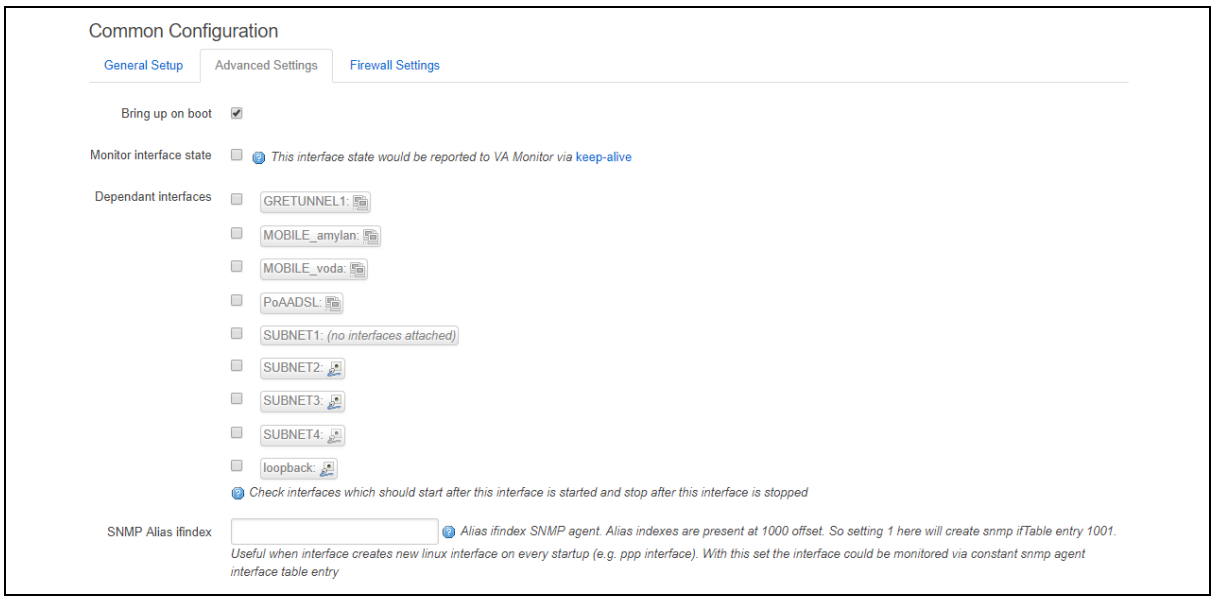



**Figure 71: GRE advanced settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Bring up on boot<br>UCI: network.<if name>.auto<br>Opt: auto | Enables the interface to connect automatically on boot up.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Monitor interface state<br>UCI: network.<if name>.monitored<br>Opt: monitored | Enabled if status of interface is presented on Monitoring platform.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Dependant Interfaces<br>UCI: network.[..x..].dependants<br>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.<br>Separate multiple interfaces by a space when using UCI.<br>Example: `option dependants 'PPPADSL MOBILE'`<br>This replaces the following previous options in child interfaces.<br>gre – option local_interface<br>lt2p – option src_ipaddr<br>iot – option wan1 wan2<br>6in4 – option ipaddr<br>6to4 – option ipaddr |
| Web: SNMP Alias ifindex<br>UCI: network.[..x..].snmp_alias_ifindex<br>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<br>Blank – No SNMP interface alias index<br>Range – 0 - 4294966295 |

**Table 36: 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.

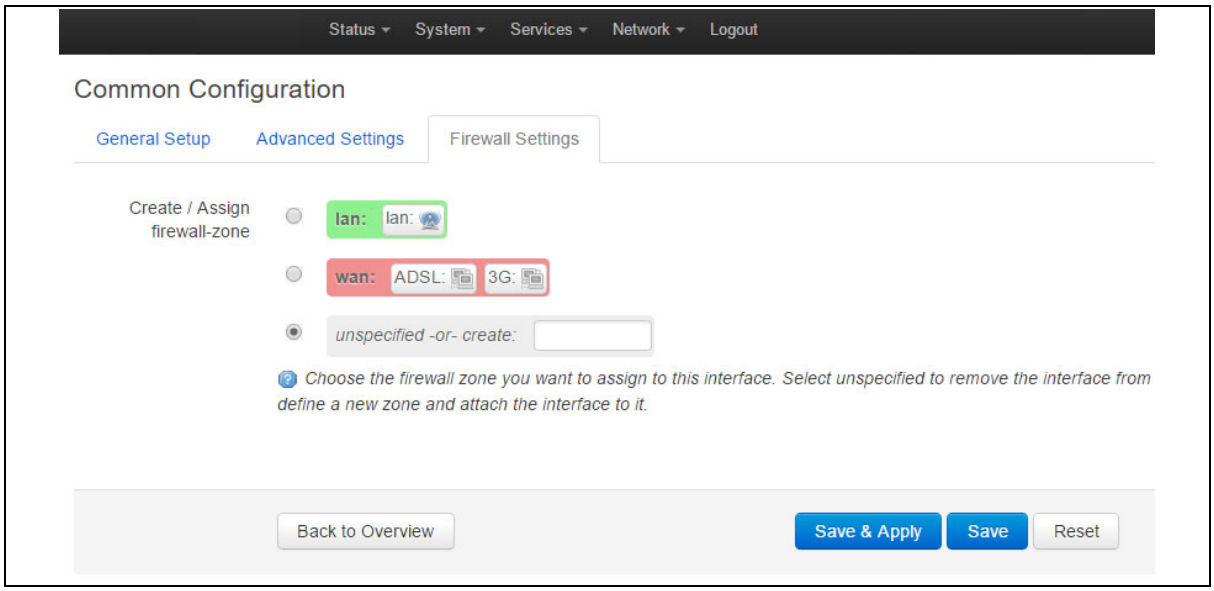



**Figure 72: GRE firewall settings**

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.

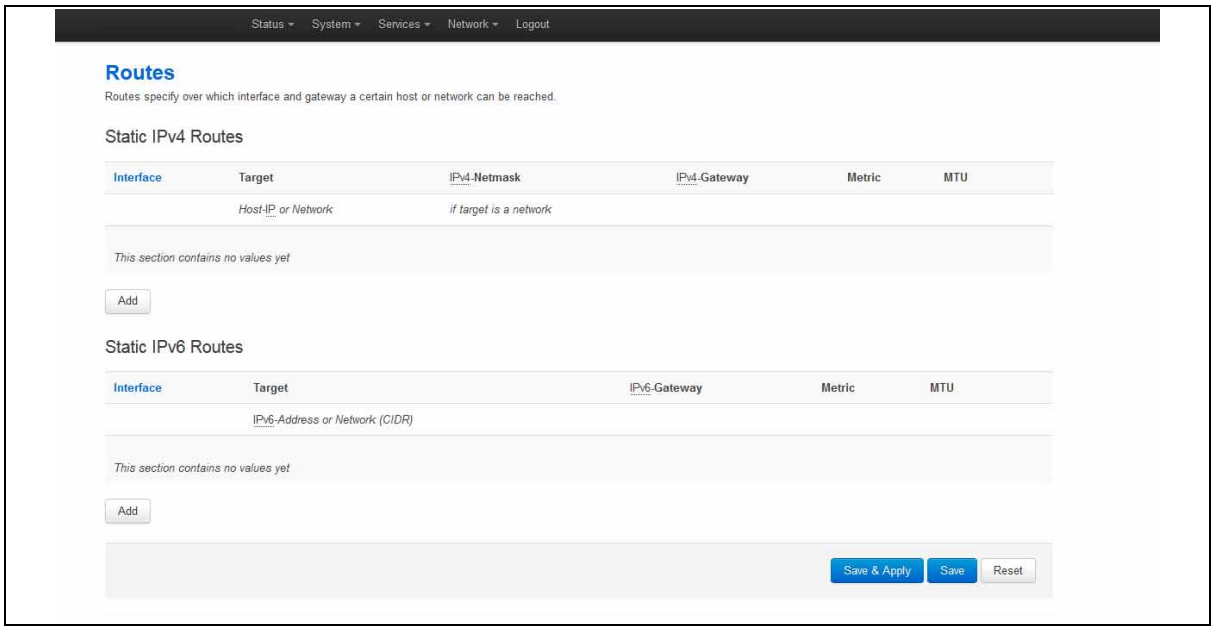



**Figure 73: The routes page**

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

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Interface<br>UCI: network.@route[0].interface<br>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<br>UCI: network.@route[0].target<br>Opt: target | Specifies the route network IP address. |
| Web: netmask<br>UCI: network.@route[0].netmask<br>Opt: netmask | Defines the route netmask. If omitted, 255.255.255.255 is assumed, which makes the target a host address. |

| | |
|---|---|
| Web: Gateway<br>UCI: network.@route[0].gateway<br>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<br>UCI: network.@route[0].metric<br>Opt: metric | Specifies the route metric to use.<br>0<br>Range |
| Web: MTU<br>UCI: network.@route[0].mtu<br>Opt:mtu | Defines a specific MTU for this route. If omitted, the MTU from the parent interface will be taken.<br>Blank<br>Range |

**Table 37: 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<br>UCI: network.@route[1].interface<br>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<br>UCI: network.@route[1].target<br>Opt: target | Specifies the route network IP address, or subnet in CIDR notation:<br>Eample: 2001:0DB8:100:F00:BA3::1/64 |
| Web: Gateway<br>UCI: network.@route[1].gateway<br>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<br>UCI: network.@route[1].metric<br>Opt: metric | Specifies the route metric to use.<br>0<br>Range |
| Web: MTU<br>UCI: network.@route[1].mtu<br>Opt:mtu | Defines a specific MTU for this route. If omitted the MTU from the parent interface will be taken.<br>Empty<br>Range |

**Table 38: 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.

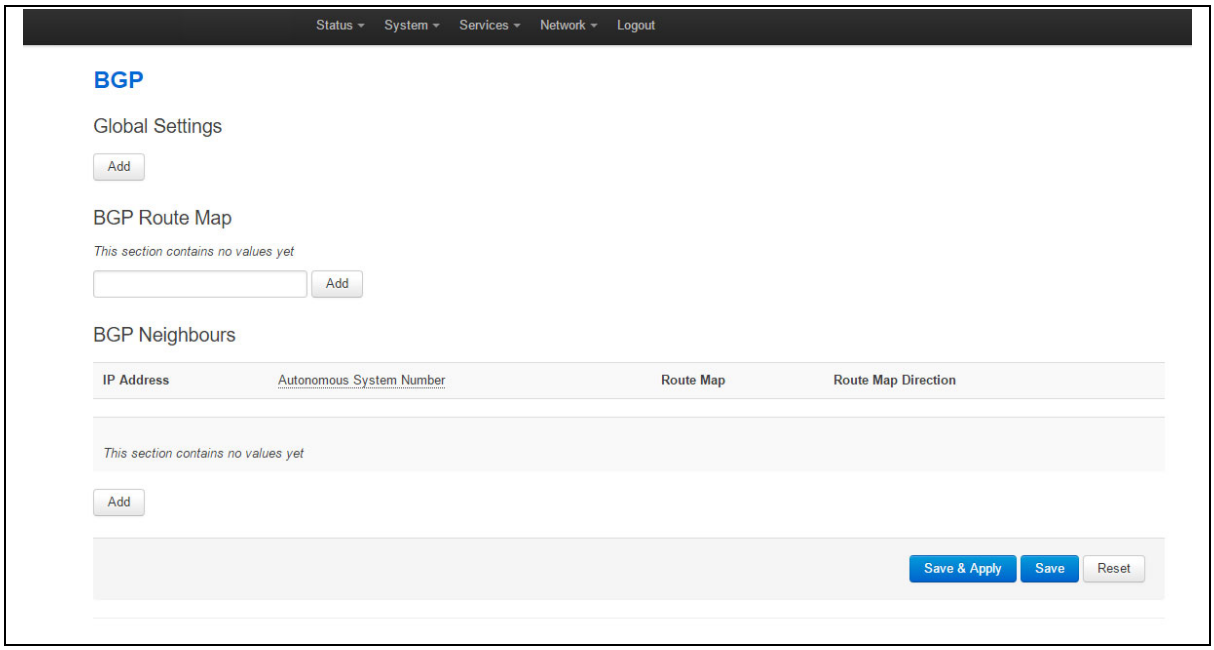



**Figure 74: The BGP page**

### 16.2.1 BGP global settings

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

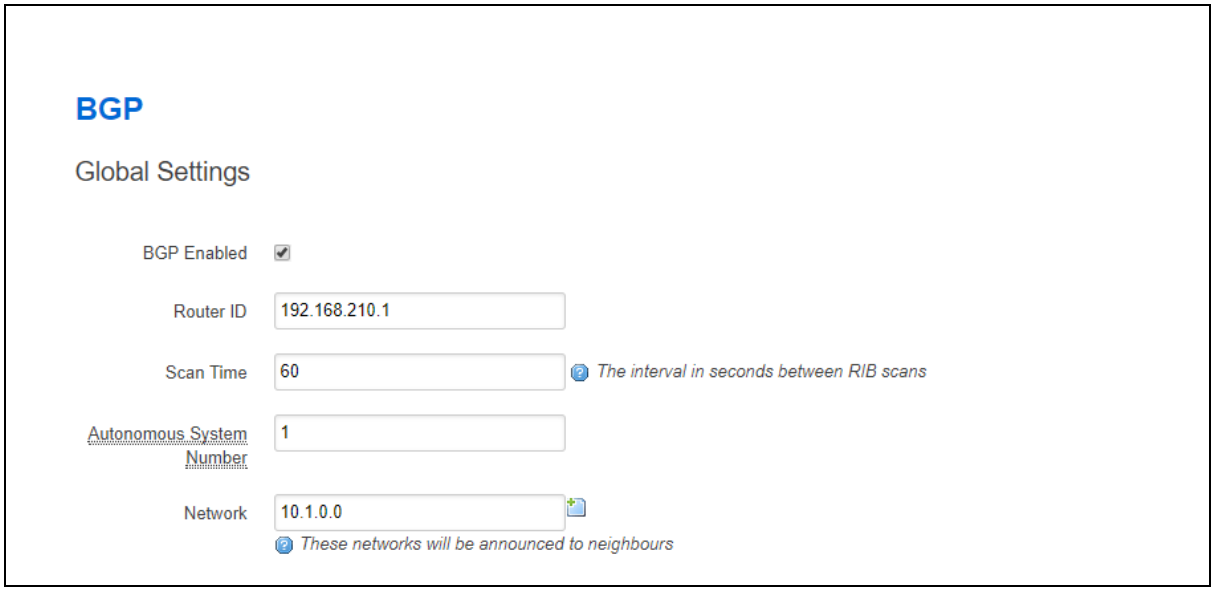



**Figure 75: The BGP global settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: BGP Enabled<br>UCI: bgpd.bgpd.enabled<br>Opt: enabled | Enables or disables BGP protocol.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Router ID<br>UCI: bgpd.bgpd.router_id<br>Opt: router_id | Sets a Unique Router ID in 4 byte format 0.0.0.0. |
| Web: Scan Time<br>UCI: bgpd.bgpd.scan_time<br>Opt: scan_time | Defines the interval in seconds between RIB scans.<br>60 — 60 seconds<br>Range |
| Web: Autonomous System Number<br>UCI: bgpd.bgpd.asn<br>Opt: asn | Defines the ASN for the local router. Type in the ASN.<br>Blank<br>Range — 1-4294967295 |
| Web: Network<br>UCI: bgpd.bgpd.network<br>Opt: list network | 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. |

**Table 39: 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.

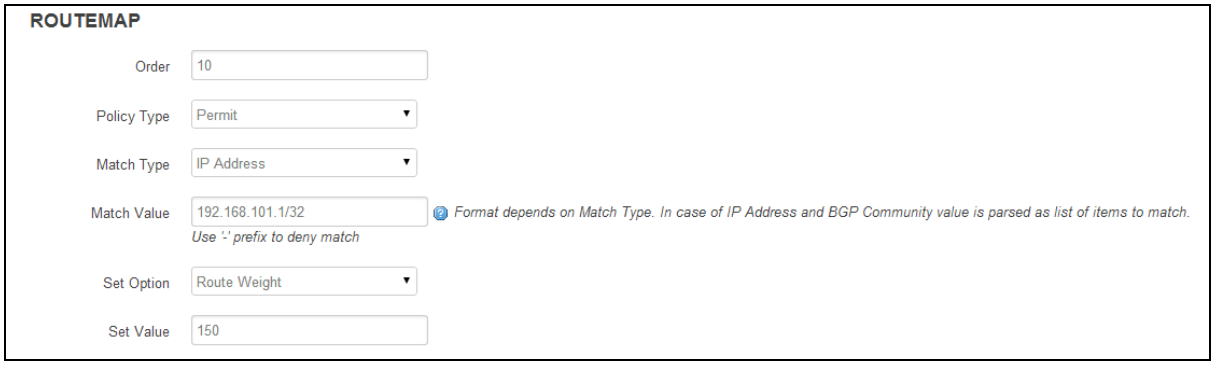



**Figure 76: The routemap section**

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

**Table 40: 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 77: The BGP neighbours section**

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

**Table 41: 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.

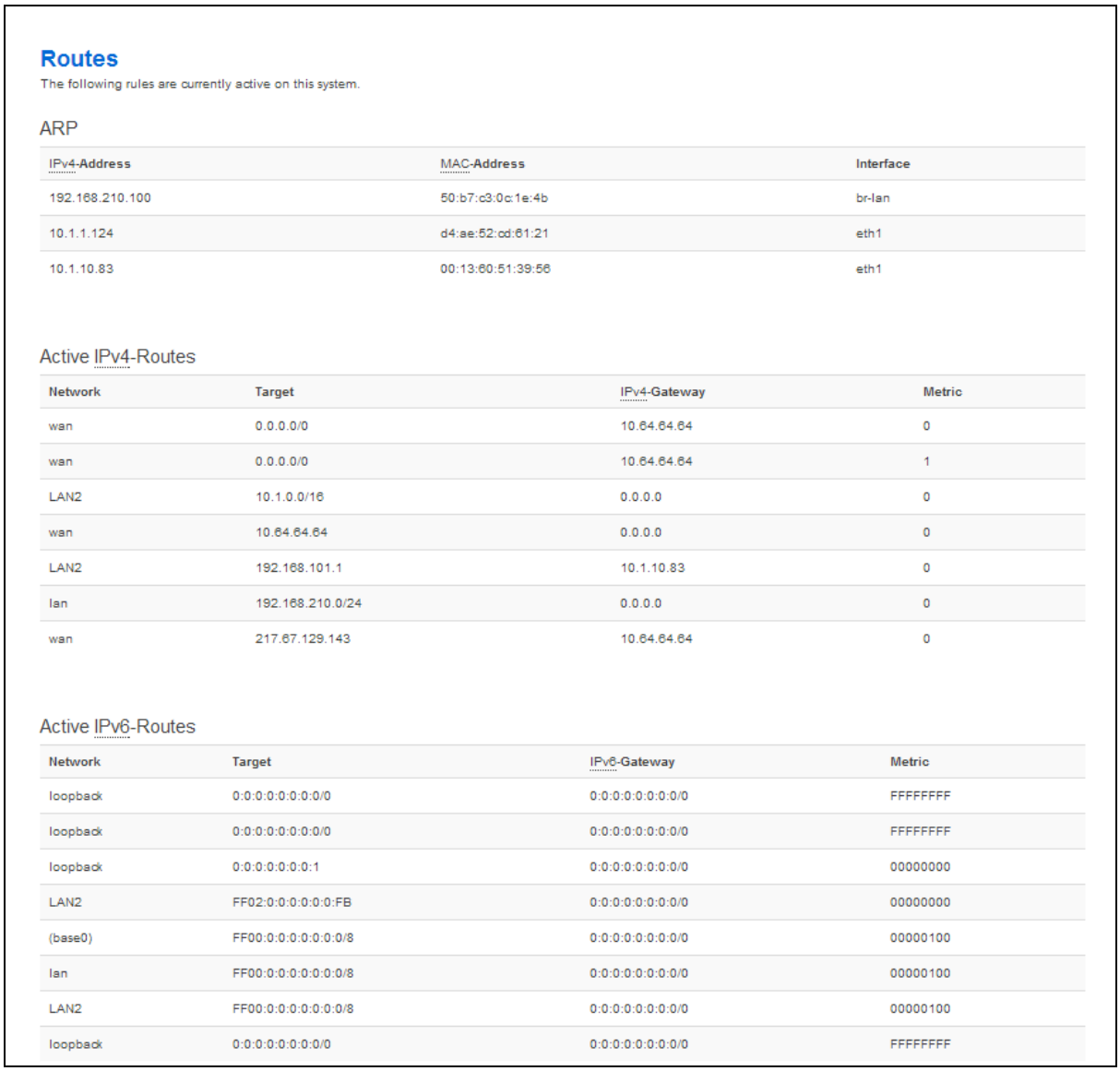
## 16.4 Configuring BGP using packages options

```
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 78: 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

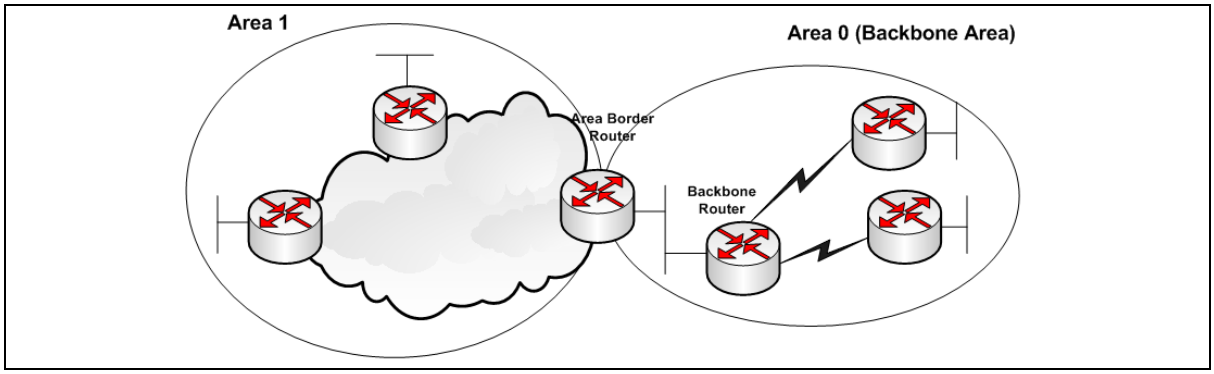



**Figure 79: OSPF areas**

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.

| State | Description |
|---|---|
| Down | Indicates that no Hellos have been heard from the neighbouring router |
| Init | Indicates a Hello packet has been heard from the neighbour, but two-way communication has not yet been initialized. |
| 2-Way | Indicates that bidirectional communication has been established.<br>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. |
| ExStart | 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. |
| Exchange | 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. |
| Loading | 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. |
| Full | 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:<br>Full/DR — Indicating that the neighbour is a Designated Router (DR)<br>Full/BDR — Indicating that the neighbour is a Backup Designated Router (BDR)<br>Full/DROther — 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. |

**Table 42: 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:<br>OSPF will elect DRs and BDRs<br>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<br>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:<br>OSPF will not elect DRs and BDRs<br>All OSPF traffic is multicast to 224.0.0.5<br>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:<br>OSPF will not elect DRs and BDRs.<br>All OSPF traffic is multicast to 224.0.0.5.<br>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:<br>OSPF will elect DRs and BDRs.<br>OSPF neighbours must be manually defined, so all OSPF traffic is unicast instead of multicast.<br>**Note**: on non-broadcast networks, neighbours must be manually specified, as multicast Hello's are not allowed |

**Table 43: 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).

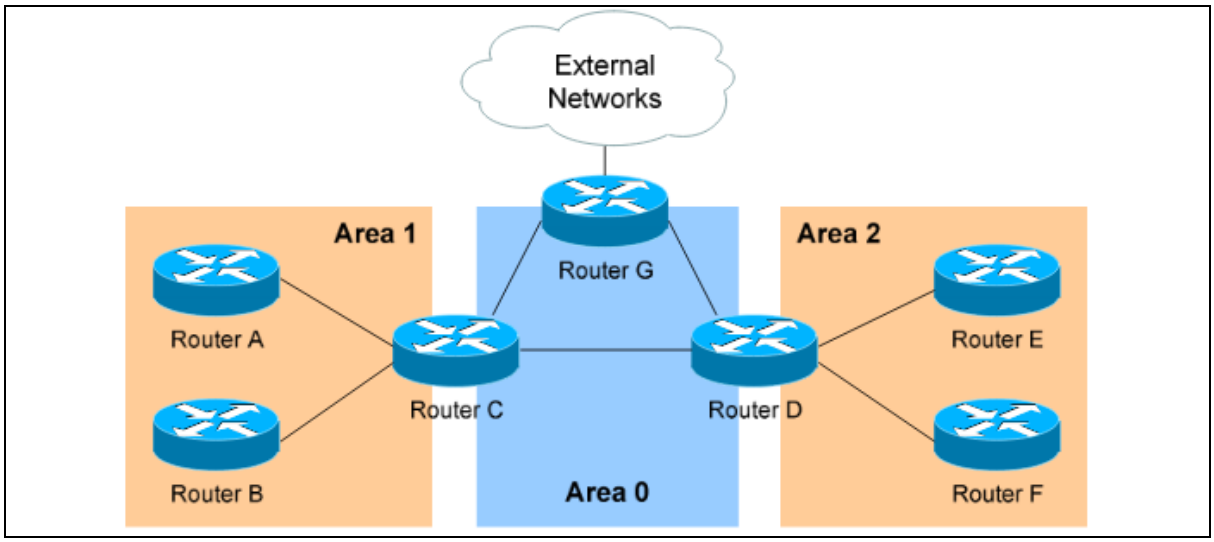



**Figure 80: OSPF hierarchy**

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 44: 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<br>network<br>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'.

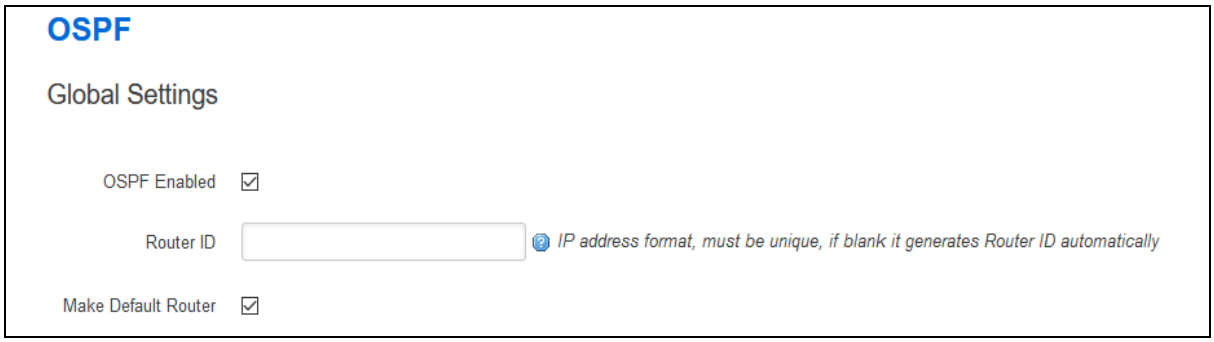



**Figure 81: The OSPF global settings configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: OSPF Enabled<br>UCI: ospfd.ospfd.enabled<br>Opt: enabled | Enables OSPF advertisements on router.<br>0 Disabled.<br>1 Enabled. |
| Web: Router ID<br>UCI: ospfd.ospfd.router_id<br>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.<br>Empty<br>Range |
| Web: Make Default Router<br>UCI: ospfd.ospfd.default_info_originate<br>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.<br>0 Disabled.<br>1 Enabled. |
| Web: n/a<br>UCI: ospfd.ospfd.vty_enabled<br>Opt: vty_enabled | Enable vty for OSPFd (telnet to localhost:2604) |

**Table 45: 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 82: The OSPF topology configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Network<br>UCI: ospfd.@network[0].ip_addr<br>Opt: ip_addr | Specify the IP address for OSPF enabled interface.<br>Format: A.B.C.D |
| Web: Mask Length<br>UCI: ospfd.@network[0].mask_length<br>Opt: mask_length | Specify the mask length for OSPF enabled interface. The mask length should be entered in CIDR notation. |
| Web: Area<br>UCI: ospfd.@network[0].area<br>Opt: area | Specify the area number for OSPF enabled interface. |
| Web: Stub Area<br>UCI: ospfd.@network[0].stub_area<br>Opt: stub_area | Only for non-backbone areas.<br>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).<br>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.<br>0 Disabled.<br>1 Enabled. |

**Table 46: 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.

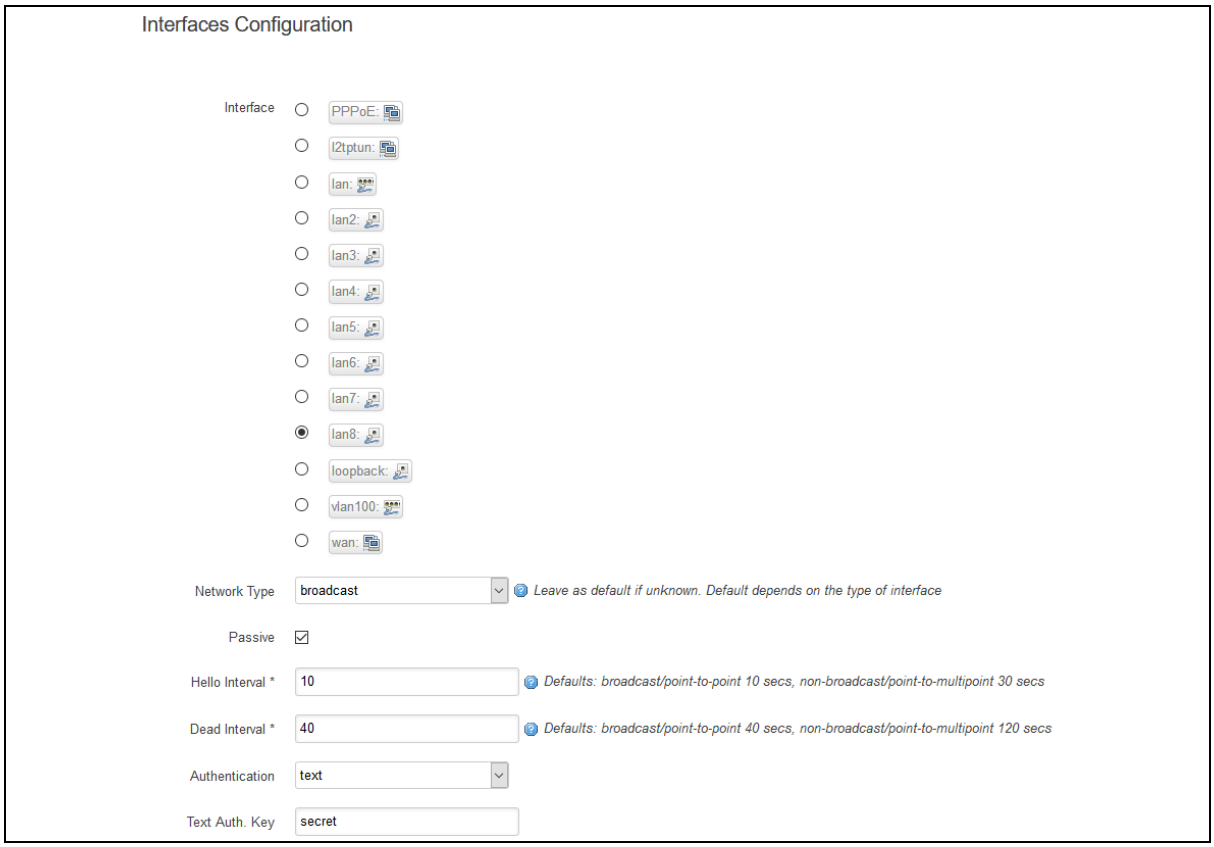



**Figure 83: The OSPF interfaces configuration section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Interface<br>UCI: ospfd.@interface[0].ospf_interface<br>Opt: ospf_interface | Defines the interface name |
| Web: Network Type<br>UCI: ospfd.@interface[0].network_type<br>Opt: network_type | Defines network type for specified interface.<br>**Default**: Autodetect: it will be broadcast. If broadcast is not supported on that interface then use point-to-point.<br>broadcast<br>non-broadcast<br>point-to-point<br>point-to-multipoint |
| Web: Passive<br>UCI: ospfd.@interface[0].passive<br>Opt: passive | 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.<br>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.<br>0: Disabled.<br>1: Enabled. |

| | |
|---|---|
| Web: Hello Interval<br>UCI: ospfd.@interface[0].hello_interval<br>Opt: hello_interval | Defines the number of seconds for the Hello Interval timer value.<br>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.<br>The default is every 10 seconds for broadcast and point-to-point interfaces, and 30 seconds for non-broadcast and point-to-multipoint interfaces.<br>10 — 10 seconds<br>Range |
| Web: Dead Interval<br>UCI: ospfd.@interface[0].dead_interval<br>Opt: dead_interval | 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.<br>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.<br>40 — 40 seconds<br>Range |
| Web: Authentication<br>UCI: ospfd.@interface[0].auth_mode<br>Opt: auth_mode | 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.<br>no — Default value. No authentication.<br>md5 — Set the interface with OSPF MD5 authentication.<br>text — Set the interface with OSPF simple password authentication. |
| Web: Text Auth. Key<br>UCI: ospfd.@interface[0].text_auth_key<br>Opt: text_auth_key | This command sets authentication string for text authentication. text_auth_key option can have length up to 8 characters.<br>Displayed only when Authentication is set to text. |
| Web: Key ID<br>UCI: ospfd.@interface[0].key_id<br>Opt: key_id | Specifies key ID. Must be unique and match at both ends.<br>Displayed only when Authentication is set to MD5. |
| Web: MD5 Auth. Key<br>UCI: ospfd.@interface[0].md5_auth_key<br>Opt: md5_auth_key | Specify Keyed MD5 chain.<br>Displayed only when Authentication is set to MD5. |

**Table 47: 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<br>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**.

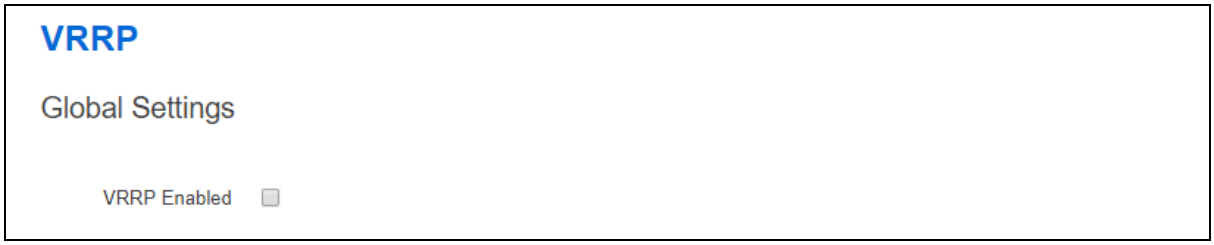



**Figure 84: The VRRP global settings configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: VRRP Enabled<br>UCI: vrrp.main.enabled<br>Opt: Enabled | Globally enables VRRP on the router.<br>0 Disabled.<br>1 Enabled. |

## 18.3.2 VRRP group configuration settings

The VRRP Group Configuration section configures vrrp package vrrp_group section.

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

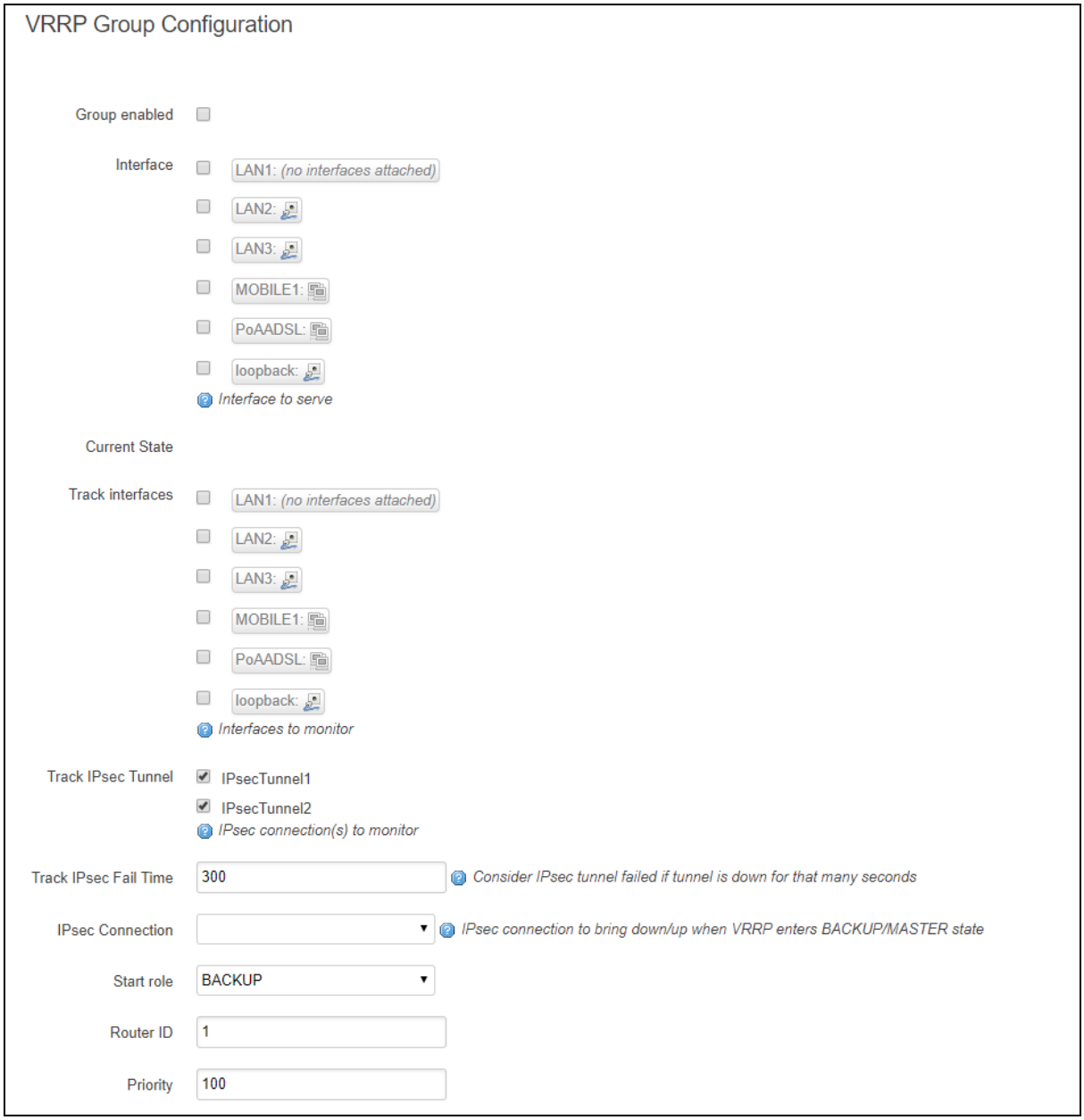



**Figure 85: The VRRP group configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Group Enabled<br>UCI: vrrp.@vrrp_group[X].enabled<br>Opt: Enabled | Enables a VRRP group on the router.<br>0 Disabled.<br>1 Enabled. |
| Web: Interface<br>UCI: vrrp.@vrrp_group[X].interface<br>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..<br>lan<br>Range |

| | |
|---|---|
| Web: Track Interfaces<br>UCI: vrrp.@vrrp_group[X].track_iface<br>Opt: list track_iface | 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.<br>Multiple interfaces are entered using `uci set` and `uci add_list` commands. Example:<br>`uci set vrrp.@vrrp_group[0].track_iface=wan1`<br>`uci add_list vrrp.@vrrp_group[0].track_iface=wan2`<br>or using a list of options via package options<br>`list track_iface 'wan1'`<br>`list track_iface 'wan2'`<br>wan: <br>Range: |
| Web: Track IPsec Tunnel<br>UCI: vrrp.@vrrp_group[X].track_ipsec<br>Opt: list track_ipsec | 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.<br>Multiple IPsec connections are entered using `uci set` and `uci add_list` commands. Example:<br>`uci set vrrp.@vrrp_group[0].track_ipsec=Tunnel1`<br>`uci add_list vrrp.@vrrp_group[0].track_ipsec=Tunnel2`<br>or using a list of options via package options<br>`list track_ipsec 'Tunnel1'`<br>`list track_ipsec 'Tunnel2'`<br>Blank: No IPSec connection to track.<br>Range: |
| Web: Track IPsec Fail Time<br>UCI: vrrp.@vrrp_group[X].<br>track_ipsec_fail_sec<br>Opt: track_ipsec_fail_sec | Defines duration in seconds to determine IPsec tunnel failure<br>300: 300 seconds<br>Range: |
| Web: IPSec connection<br>UCI:<br>vrrp.@vrrp_group[X].ipsec_connection<br>Opt: ipsec_connection | Sets which IPSec connection to bring up or down when VRRP enters 'Backup/Master' state.<br>Multiple IPSec connections are entered via the package option using a space separator. Example:<br>`option ipsec_connection 'IPSecTunnel1 IPSecTunnel2'`<br>Blank: No IPSec connection to toggle.<br>Range: |
| Web: Start role<br>UCI: vrrp.@vrrp_group[X].init_state<br>Opt: init_state | 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.<br>BACKUP: <br>MASTER: |
| Web: Router ID<br>UCI: vrrp.@vrrp_group[X].router_id<br>Opt: router_id | 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.<br>1: <br>Range: 1-255 |
| Web: Priority<br>UCI: vrrp.@vrrp_group[X].priority<br>Opt: priority | 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.<br>100: <br>Range: 0-255 |
| Web: Advert intvl<br>UCI: vrrp.@vrrp_group[X].advert_int_sec<br>Opt: advert_int_sec | Sets the VRRP hello value in seconds. This value must match the value set on a peer.<br>120: 120 seconds<br>Range: |

| | |
|---|---|
| Web: Password<br>UCI: vrrp.@vrrp_group[X].password<br>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<br>UCI: vrrp.@vrrp_group[X].virtual_ipaddr<br>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<br>UCI: vrrp.@vrrp_group[X].garp_delay_sec<br>Opt: garp_delay_sec | Sets the Gratuitous ARP message sending delay in seconds.<br>5 — 5 seconds<br>Range |
| Web: n/a<br>UCI: vrrp.@vrrp_group[X].track_ipsec<br>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.<br>Multiple IPsec connections are entered using `uci set` and `uci add_list` commands. Example:<br>`uci set vrrp.@vrrp_group[0].track_ipsec=Tunnel1`<br>`uci add_list vrrp.@vrrp_group[0].track_ipsec=Tunnel2`<br>or using a list of options via package options<br>`list track_ipsec 'Tunnel1'`<br>`list track_ipsec 'Tunnel2'`<br>Blank — No IPSec connection to track.<br>Range |

**Table 48: 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<br>interface<br>key_chain<br>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'.

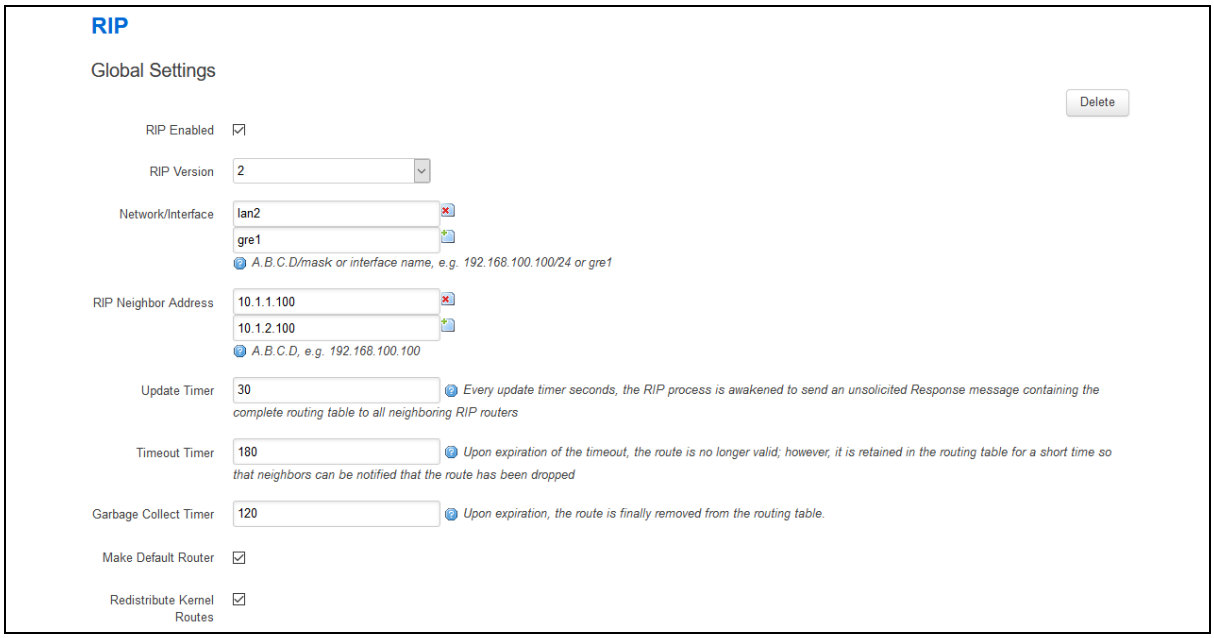



**Figure 86: The RIP global settings configuration page**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: RIP Enabled<br>UCI: ripd.ripd.enabled<br>Opt: enabled | Enables RIP advertisements on router.<br>0 Disabled.<br>1 Enabled. |
| Web: RIP Version<br>UCI: ripd.ripd.version<br>Opt: version | Specifies the RIP version that will be used. Version 2 is recommended.<br>1 RIP version 1<br>2 RIP version 2 |
| Web: Network/Interface<br>UCI: ripd.ripd.network<br>Opt: list network | Defines the list of the interfaces that will be used to advertise RIP packets.<br>Format: A.B.C.D/mask or interface name<br>Multiple RIP interfaces are entered using `uci set` and `uci add_list` commands. Example:<br>`uci set ripd.ripd.network=lan1`<br>`uci add_list ripd.ripd.network=lan2`<br>or using a list of options via package options<br>`list network 'lan1'`<br>`list network 'lan2'` |
| Web: RIP Neighbor Address<br>UCI: ripd.ripd.neighbor<br>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.<br>Multiple RIP neighbours are entered using `uci set` and `uci add_list` commands. Example:<br>`uci set ripd.ripd.neighbor=1.1.1.1`<br>`uci add_list ripd.ripd.neighbor=2.2.2.2`<br>or using a list of options via package options<br>`list neighbor '1.1.1.1'`<br>`list neighbor '2.2.2.2'` |
| Web: Update Timer<br>UCI: ripd.ripd.tb_update_sec<br>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.<br>30<br>Range |
| Web: Timeout Timer<br>UCI: ripd.ripd.tb_timeout_sec<br>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.<br>180<br>Range |
| Web: Garbage Collect Timer<br>UCI: ripd.ripd.tb_garbage_sec<br>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".<br>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.<br>120<br>Range |

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

**Table 49: 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.

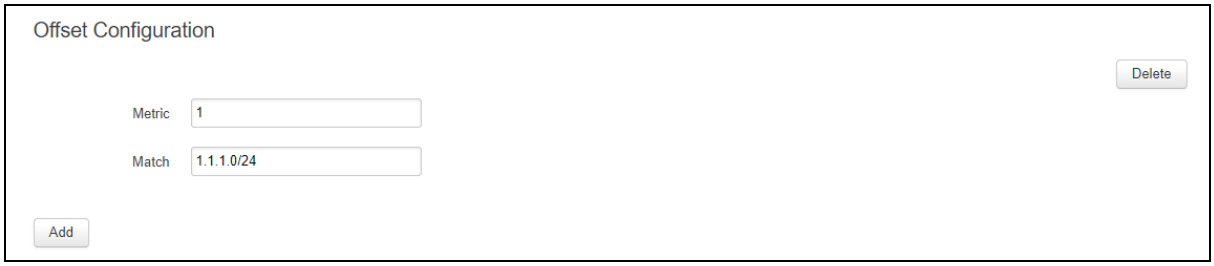



**Figure 87: The RIP global settings configuration page**

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

**Table 50: Information table for RIP offset commands**

## 19.3.3 Interfaces configuration

**Figure 88: The RIP interfaces configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Interface<br>UCI: ripd.@interface[0].rip_interface<br>Opt: rip_interface | Specifies the interface name. |
| Web: Split Horizon<br>UCI: ripd.@interface[0].split_horizon<br>Opt: split_horizon | Prohibits the router from advertising a route back onto the interface from which it was learned.<br>0 Disable.<br>1 Enable. |
| Web: Poison Reverse<br>UCI: ripd.@interface[0].poison_reverse<br>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".<br>0 Disable.<br>1 Enable. |
| Web: Passive<br>UCI: ripd.@interface[0].passive<br>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.<br>0 Disable<br>1 Enable |
| Web: Authentication<br>UCI: ripd.@interface[0].auth_mode<br>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.<br>no Default value. No authentication.<br>md5 Sets the interface with RIPv2 MD5 authentication.<br>text Sets the interface with RIPv2 simple password authentication. |
| Web: Text Auth. Key<br>UCI: ripd.@interface[0].auth_key<br>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<br>UCI: ripd.@interface[0].key_chain<br>Opt: key_chain | Specifiy Keyed MD5 chain. |

**Table 51: 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.

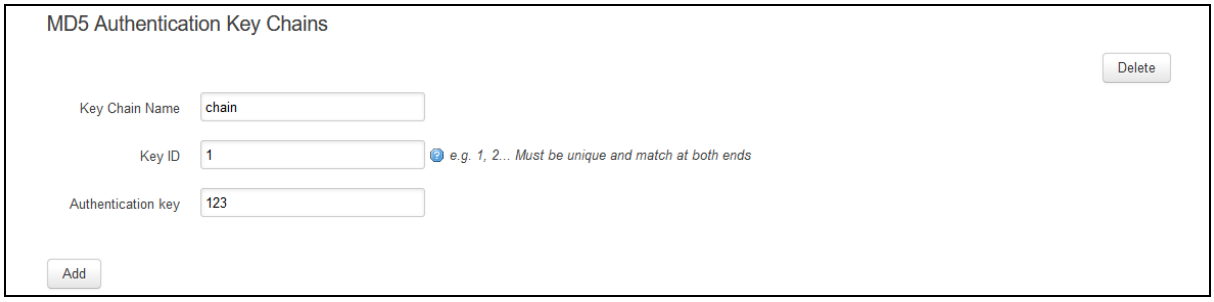



**Figure 89: The MD5 authentication key chains configuration section**

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

**Table 52: 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<br>wan |

## 20.2 Configuring Multi-WAN using the web interface

In the top menu, select **Network -> Multi-Wan**. The Multi-WAN page appears.

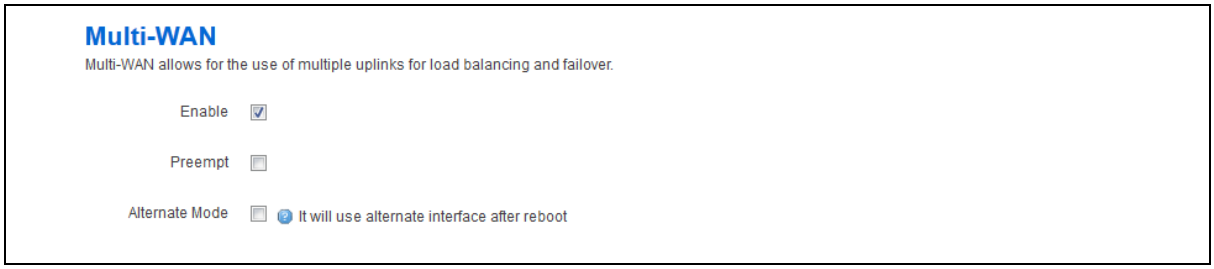



**Figure 90: The multi-WAN page**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Enable<br>UCI: multiwan.config.enabled<br>Opt: enabled | Enables or disables Multi-WAN.<br>0 Disabled.<br>1 Enabled. |
| Web: Preempt<br>UCI: multiwan.config.preempt<br>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<br>0 Disabled.<br>1 Enabled. |
| Web: Alternate Mode<br>UCI: multiwan.config.alt_mode<br>Opt: alt_mode | Enables or disables alternate mode for Multi-WAN. If enabled the router will use an alternate interface after reboot.<br>0 Disabled.<br>1 Enabled. |

**Table 53: 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.

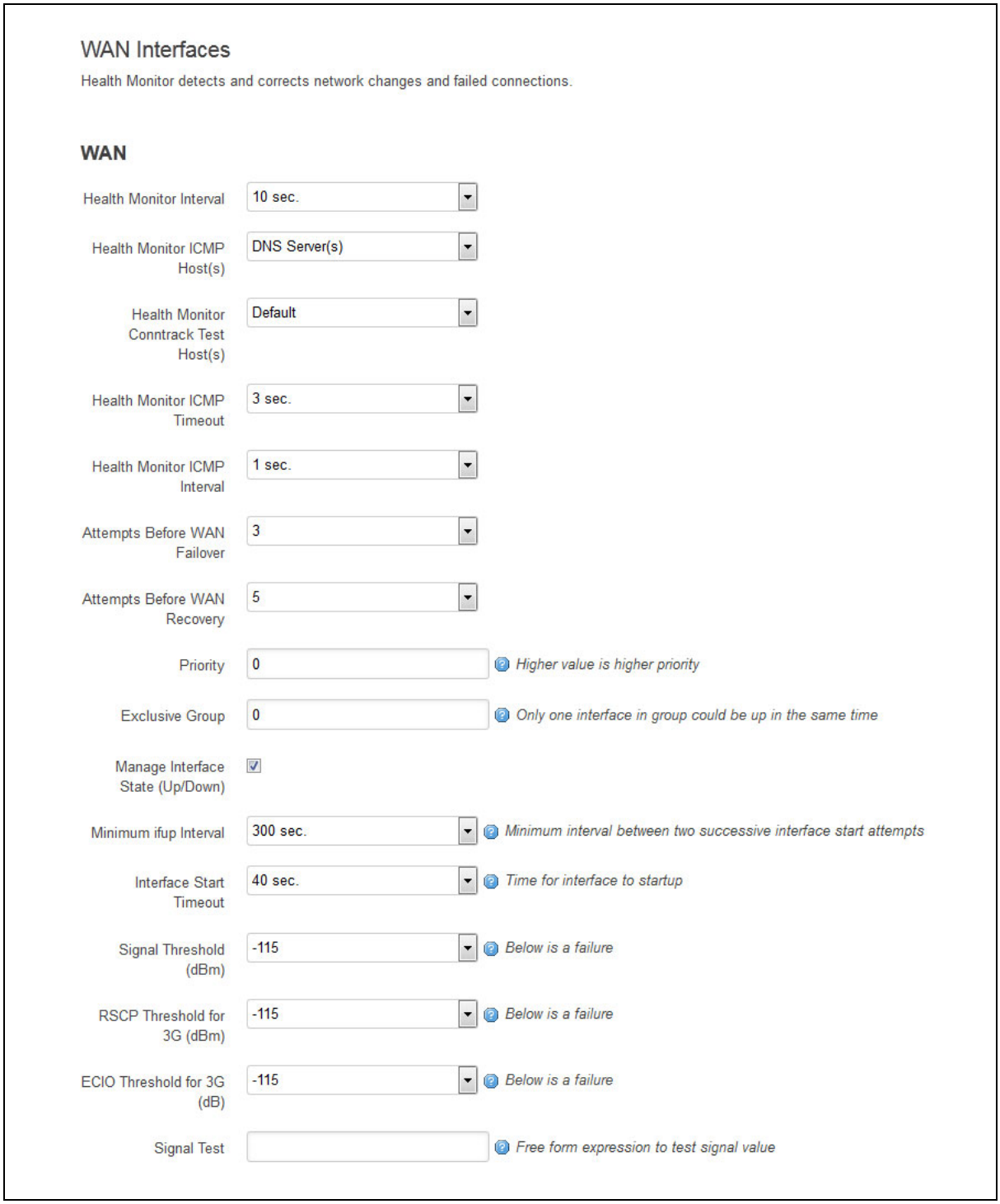



**Figure 91: Example interface showing failover traffic destination as the added multi-WAN interface**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Health Monitor Interval<br>UCI: multiwan.wan.health_interval<br>Opt: health_interval | Sets the period to check the health status of the interface. The Health Monitor interval will be used for:<br>• Interface state checks<br>• Ping interval<br>• Signal strength checks |
| Web: Health Monitor ICMP Host(s)<br>UCI: multiwan.wan.icmp_hosts<br>Opt: icmp_hosts | Sends health ICMPs to configured value DNS servers by default. Configure to any address.<br>Disable — Disables the option.<br>DNS servers — DNS IP addresses will be used.<br>WAN Gateway — Gateway IP address will be used.<br>Custom — 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' |
| Web: Health Monitor Conntrack Test Host(s)<br>UCI: multiwan.wan.conntrack_hosts<br>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.<br>The Conntrack_hosts option defines the IP for conntrack to track, usually the icmp_host IP is used.<br>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.<br>By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated.<br>Conntrack is generally used to limit the traffic sent on a GSM network.<br>Default — Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.<br>Disable — Conntrack disabled.<br>Custom — Specifies an IP other than the icmp_host for conntrack to track. |
| Web: Health Monitor ICMP Timeout<br>UCI: multiwan.wan.timeout<br>Opt: timeout | Sets Ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.<br>3 — Wait 3 seconds for ping reply.<br>Range |
| Web: Health Monitor ICMP Interval<br>UCI: multiwan.wan.icmp_interval<br>Opt: icmp_interval | Defines the interval between multiple pings sent at each health check<br>1<br>Range |
| Web: Health Monitor ICMP Count<br>UCI: multiwan.wan.icmp_count<br>Opt: icmp_count | Defines the number of pings to send at each health check.<br>1<br>Range |
| Web: Attempts Before WAN Failover<br>UCI: multiwan.wan.health_fail_retries<br>Opt: health_fail_retries | Sets the amount of health monitor retries before interface is considered a failure.<br>3<br>Range |
| Web: Attempts Before WAN Recovery<br>UCI: multiwan.wan.health_recovery_retries<br>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.<br>5<br>Range |
| Web: Priority<br>UCI: multiwan.wan.priority<br>Opt: priority | Specifies the priority of the interface. The higher the value, the higher the priority.<br>0<br>Range |

| | |
|---|---|
| Web: Manage Interface State (Up/Down)<br>UCI: multiwan.wan.manage_state<br>Opt: manage_state | Defines whether multi-wan will start and stop the interface.<br>1 Enabled.<br>0 Disabled. |
| Web: Exclusive Group<br>UCI: multiwan.wan.exclusive_group<br>Opt: exclusive_group | Defines the group to which the interface belongs; only one interface can be active.<br>0<br>Range |
| Web: Minimum ifup Interval<br>UCI: multiwan.wan.ifup_retry_sec<br>Opt: ifup_retry_sec | Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled.<br>300 Retry primary interface every 300 seconds.<br>Range |
| Web: Interface Start Timeout<br>UCI: multiwan.wan.ifup_timeout<br>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.<br>40 40 seconds.<br>Range |
| Web: Signal Threshold (dBm)<br>UCI: multiwan.wan.signal_threshold<br>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.<br>Disabled<br>Range -46 to -115 dBm |
| Web: RSCP Threshold (dBm)<br>UCI: multiwan.wan.rscp_threshold<br>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.<br>-115 Disabled<br>Range -46 to -115 dBm |
| Web: ECIO Threshold (dB)<br>UCI: multiwan.wan.ecio_threshold<br>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.<br>-115 Disabled<br>Range -46 to -115 dB |
| Web: Signal Test<br>UCI: multiwan.wan.signal_test<br>Opt: signal_test | Defines a script to test various signal characteristics in multiwan signal test. For example:<br>option signal_test '(tech == 0) then (sig_dbm > -70) else (rscp_dbm > -105 and ecio_db > -15)'<br>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<br>Tech values are:<br>0 GSM<br>1 GSM Compact<br>2 UTRAN<br>3 GSM w/EGPRS<br>4 UTRAN w/HSPDA<br>5 UTRAN w/HSUPA<br>6 UTRAN w/HSUPA and HSDPA<br>7 E-UTRAN |

**Table 54: 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.

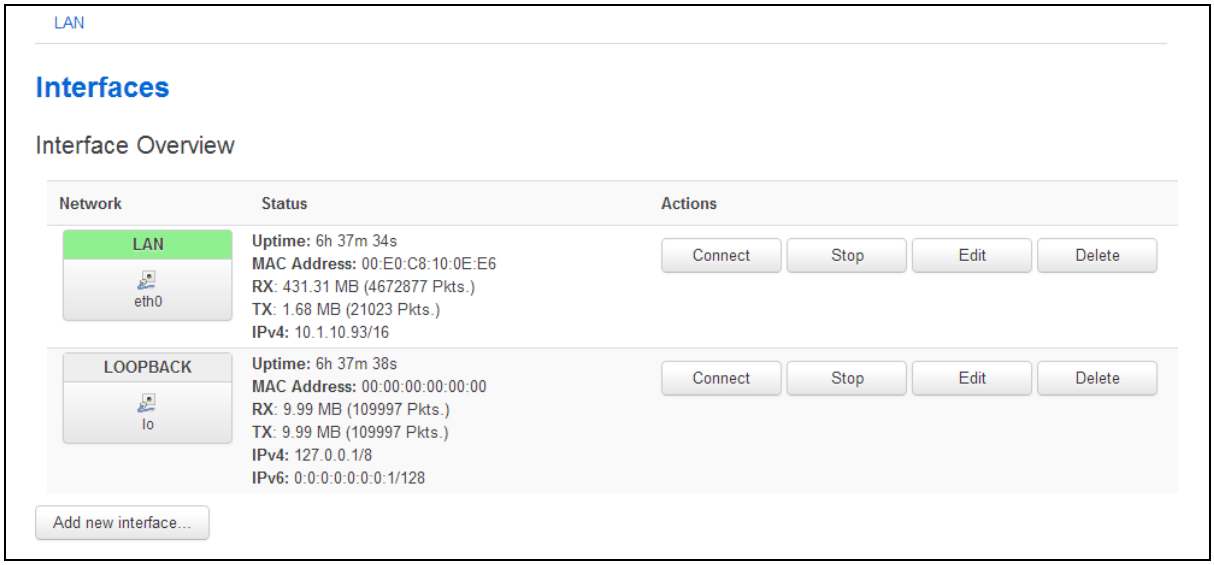



**Figure 92: The interface overview page**

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

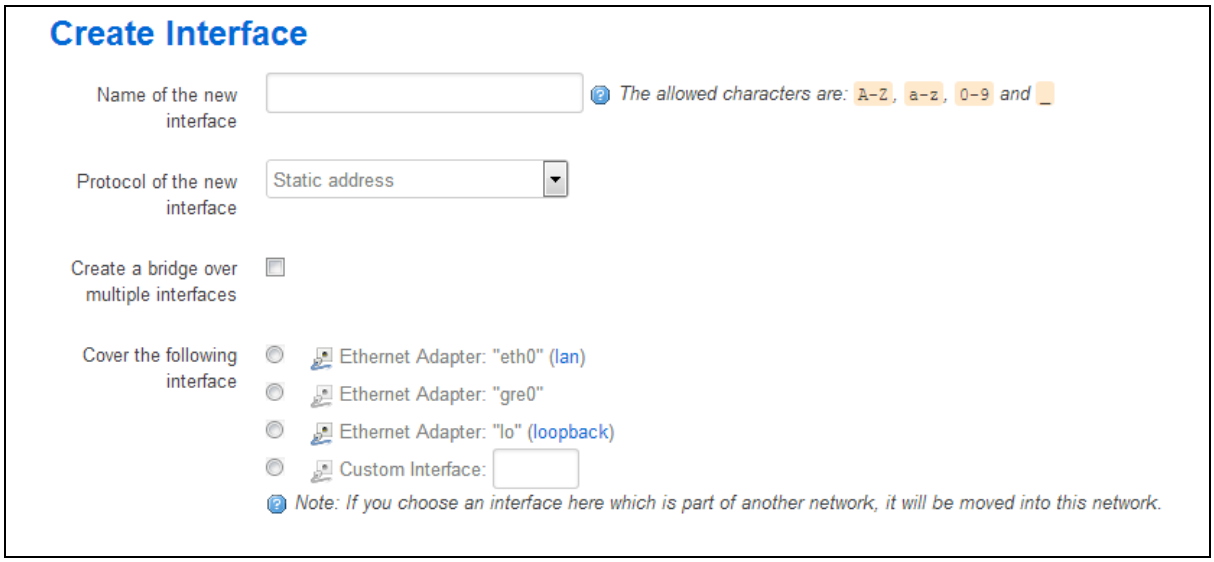



**Figure 93: The create interface page**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Name of the new interface<br>UCI: network.3g_s<sim-number>_<short-operator-name>.<br>Opt: 3g_s<sim-number>_<short-operator-name>. | Type the name of the new interface.<br>Type the interface name in following format:<br>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).<br>Type the short operator name in lower case, for example: |

| Operator name | First four alphanumeric numbers |
|---|---|
| Vodafone UK | voda |
| O2 – UK | o2uk |
| Orange | oran |

<table>
<tr><td>Web: Protocol of the new interface<br>UCI: network.[..x..].proto<br>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<br>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<br>UCI: network.[..x..].ifname<br>Opt: ifname</td><td>Selects interfaces for bridge connection.</td></tr>
</table>

**Table 55: Information table for the create interface page**

Click **Submit**. The Common Configuration page appears.

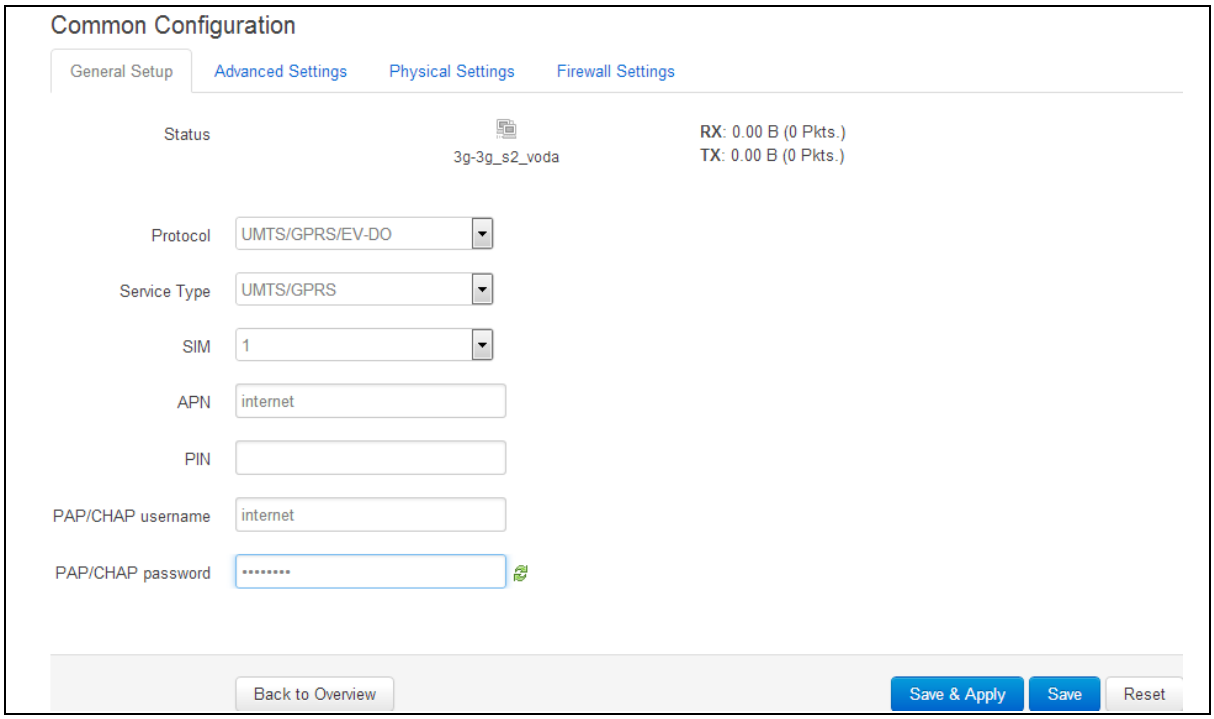



**Figure 94: The common configuration page**

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Protocol<br>UCI: network.[..x..].proto<br>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<br>UCI: network.[..x..].service<br>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<br>UCI: network.[..x..].sim<br>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<br>UCI: network.[..x..].apn<br>Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: APN username<br>UCI: network.[..x..].username<br>Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: APN password<br>UCI: network.[..x..].password<br>Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: Modem Configuration<br>UCI: N/A<br>Opt: N/A</td><td>Click the link if you need to configure additional options from Mobile Manager.</td></tr>
</table>

**Table 56: 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.

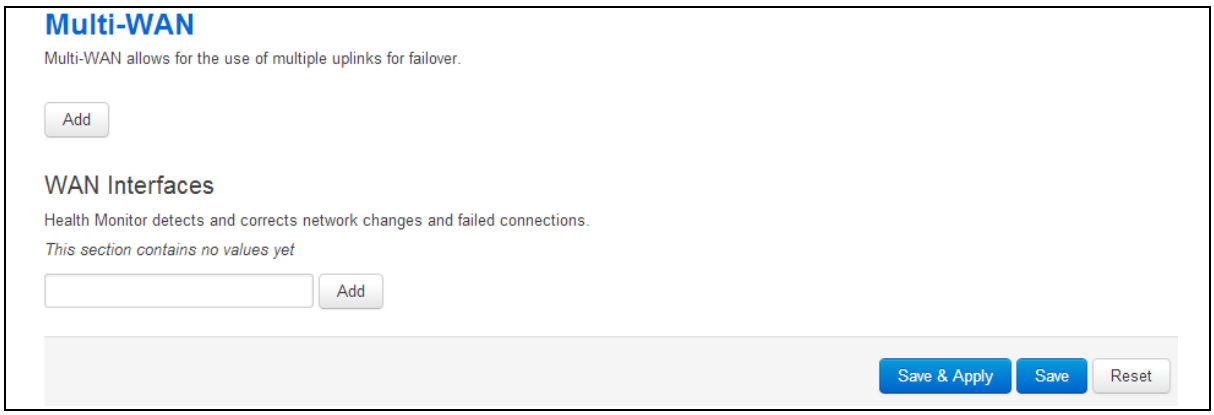



**Figure 95: The multi-WAN page**

In the WAN Interfaces section, type in the name of the Multi-WAN interface.

Click **Add**. The Multi-WAN page appears.

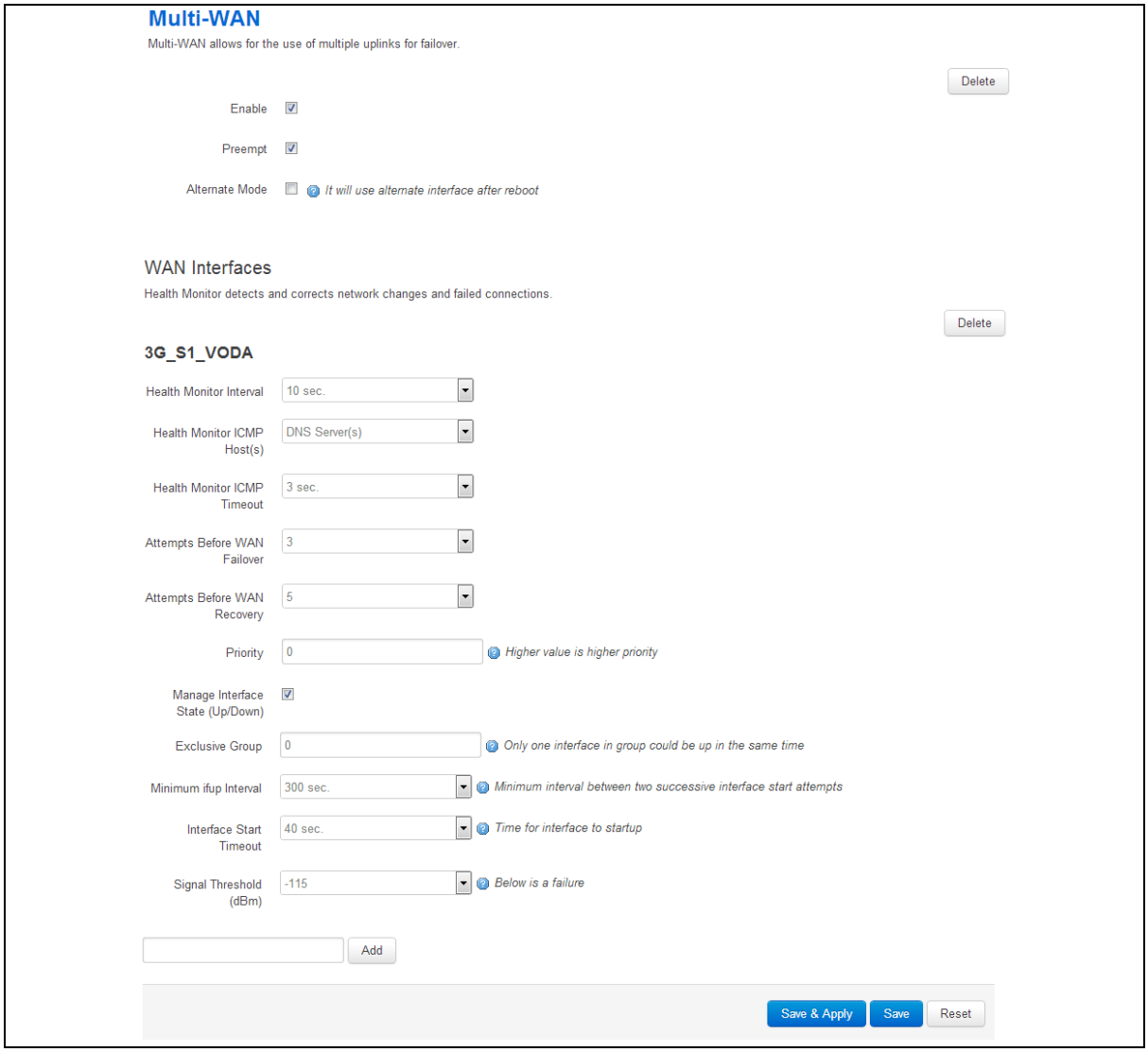



**Figure 96: The multi-WAN page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable<br>UCI: multiwan.config.enabled<br>Opt: enabled | Enables multiwan.<br>0 Disabled.<br>1 Enabled. |
| Web: Preempt<br>UCI: multiwan.config.preempt<br>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.<br>0 Disabled.<br>1 Enabled. |
| Web: Alternate Mode<br>UCI: multiwan.config.alt<br>Opt: alt | Enables or disables alternate mode for multiwan. If enabled the router will use an alternate interface after reboot.<br>0 Disabled.<br>1 Enabled. |
| Web: WAN Interfaces<br>UCI: multiwan.3g_s<sim-number>_<short-operator-name><br>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<br>UCI: multiwan.[..x..].health_interval<br>Opt: health_interval | Sets the period to check the health status of the interface. The Health Monitor interval will be used for:<br>• Interface state checks<br>• Ping interval<br>• Signal strength checks |
| Web: Health Monitor ICMP Host(s)<br>UCI: multiwan.[..x..].icmp_hosts<br>Opt: icmp_hosts | Specifies the target IP address for ICMP packets.<br>Disable — Disables the option.<br>DNS servers — DNS IP addresses will be used.<br>WAN Gateway — Gateway IP address will be used.<br>custom — Ability to provide IP address. |
| Web: Health Monitor Conntrack Test Host(s)<br>UCI: multiwan.wan.conntrack_hosts<br>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.<br>Conntrack_hosts option defines the IP for conntrack to track – usually the icmp_host IP is used.<br>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.<br>By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated.<br>Conntrack is generally used to limit the traffic sent on a GSM network.<br>Default — Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.<br>Disable — Conntrack disabled.<br>Custom — Specifies an IP other than the icmp_host for conntrack to track. |
| Web: Health Monitor ICMP Timeout<br>UCI: multiwan.[..x..].timeout<br>Opt: timeout | Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.<br>3 — Wait 3 seconds for ping reply.<br>Range |
| Web: Health Monitor ICMP Interval<br>UCI: multiwan.wan.icmp_interval<br>Opt: icmp_interval | Defines the interval between multiple pings sent at each health check.<br>1<br>Range |
| Web: Health Monitor ICMP Count<br>UCI: multiwan.wan.icmp_count<br>Opt: icmp_count | Defines the number of pings to send at each health check.<br>1<br>Range |
| Web: Attempts Before WAN Failover<br>UCI: multiwan. [..x..].health_fail_retries<br>Opt: health_fail_retries | Sets the amount of health monitor retries before interface is considered a failure.<br>3<br>Range |
| Web: Attempts Before WAN Recovery<br>UCI: multiwan. [..x..].health_recovery_retries<br>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.<br>5<br>Range |
| Web: Priority<br>UCI: multiwan.[..x..].priority<br>Opt: priority | Specifies the priority of the interface. The higher the value, the higher the priority.<br>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.<br>0<br>Range |
| Web: Exclusive Group<br>UCI: multiwan.[..x..].exclusive_group<br>Opt: exclusive_group | Defines the group to which the interface belongs; only one interface can be active.<br>0<br>Range |

| | |
|---|---|
| Web: Manage Interface State (Up/Down)<br>UCI: multiwan.[..x..].manage_state<br>Opt: manage_state | Defines whether multi-WAN will start and stop the interface. Select **Enabled**.<br>0 Disabled.<br>1 Enabled. |
| Web: Minimum ifup Interval<br>UCI: multiwan.[..x..].ifup_retry_sec<br>Opt: ifup_retry_sec | Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled. |
| Web: Interface Start Timeout<br>UCI: multiwan.[..x..].ifup_timeout<br>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.<br>Choose timer greater than 120 seconds.<br>40 40 seconds<br>Range |
| Web: Signal Threshold (dBm)<br>UCI: multiwan.[..x..].signal_threshold<br>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.<br>-115 Disabled.<br>Range -46 to -115 dBm |
| Web: RSCP Threshold (dBm)<br>UCI: multiwan.[..x..].rscp_threshold<br>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.<br>-115 Disabled.<br>Range -46 to -115 dBm |
| Web: ECIO Threshold (dB)<br>UCI: multiwan.[..x..].ecio_threshold<br>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.<br>-115 Disabled.<br>Range -46 to -115 dB |
| Web: Signal Test<br>UCI: multiwan.[..x..].signal_test<br>Opt: signal_test | Defines script to test various signal characteristics in multiwan signal test. For example:<br>option signal_test '(tech == 0) then (sig_dbm > -70) else (rscp_dbm > -105 and ecio_db > -15)'<br>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.<br>Tech values are:<br>0 GSM<br>1 GSM Compact<br>2 UTRAN<br>3 GSM w/EGPRS<br>4 UTRAN w/HSPDA<br>5 UTRAN w/HSUPA<br>6 UTRAN w/HSUPA and HSDPA<br>7 E-UTRAN |

**Table 57: 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

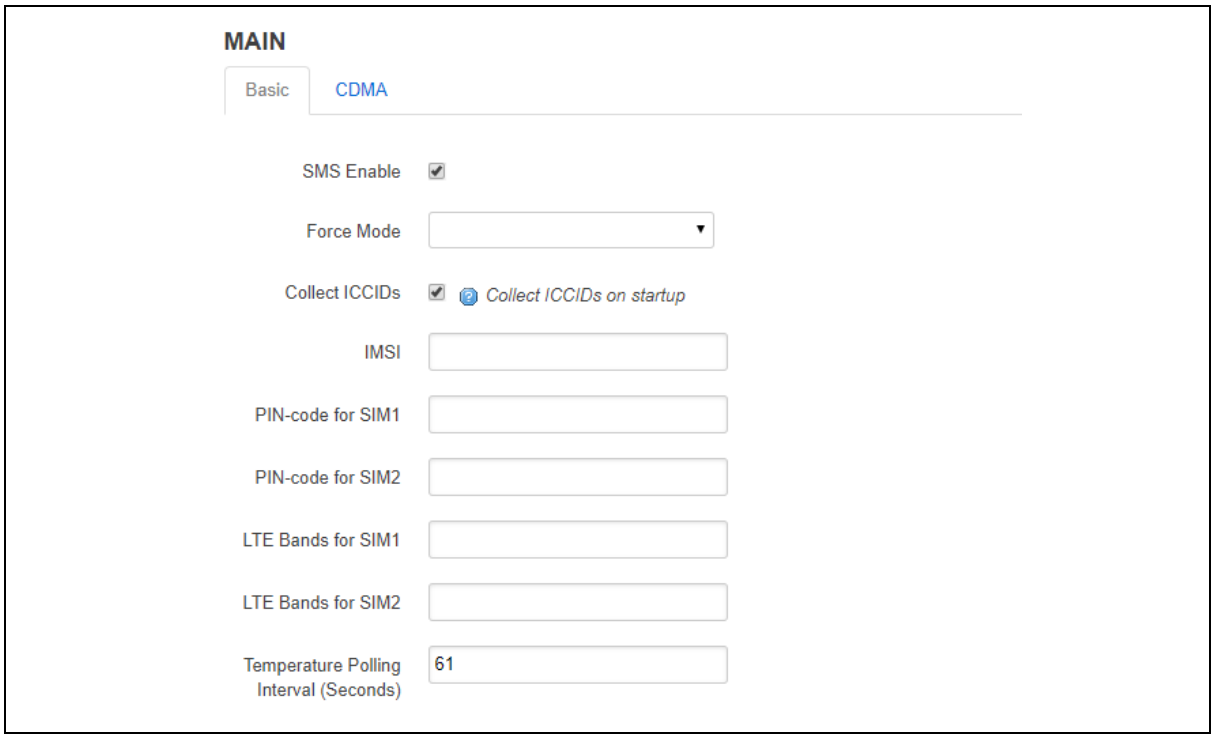



**Figure 97: The mobile manager basic page**

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

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

**Table 58: 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.

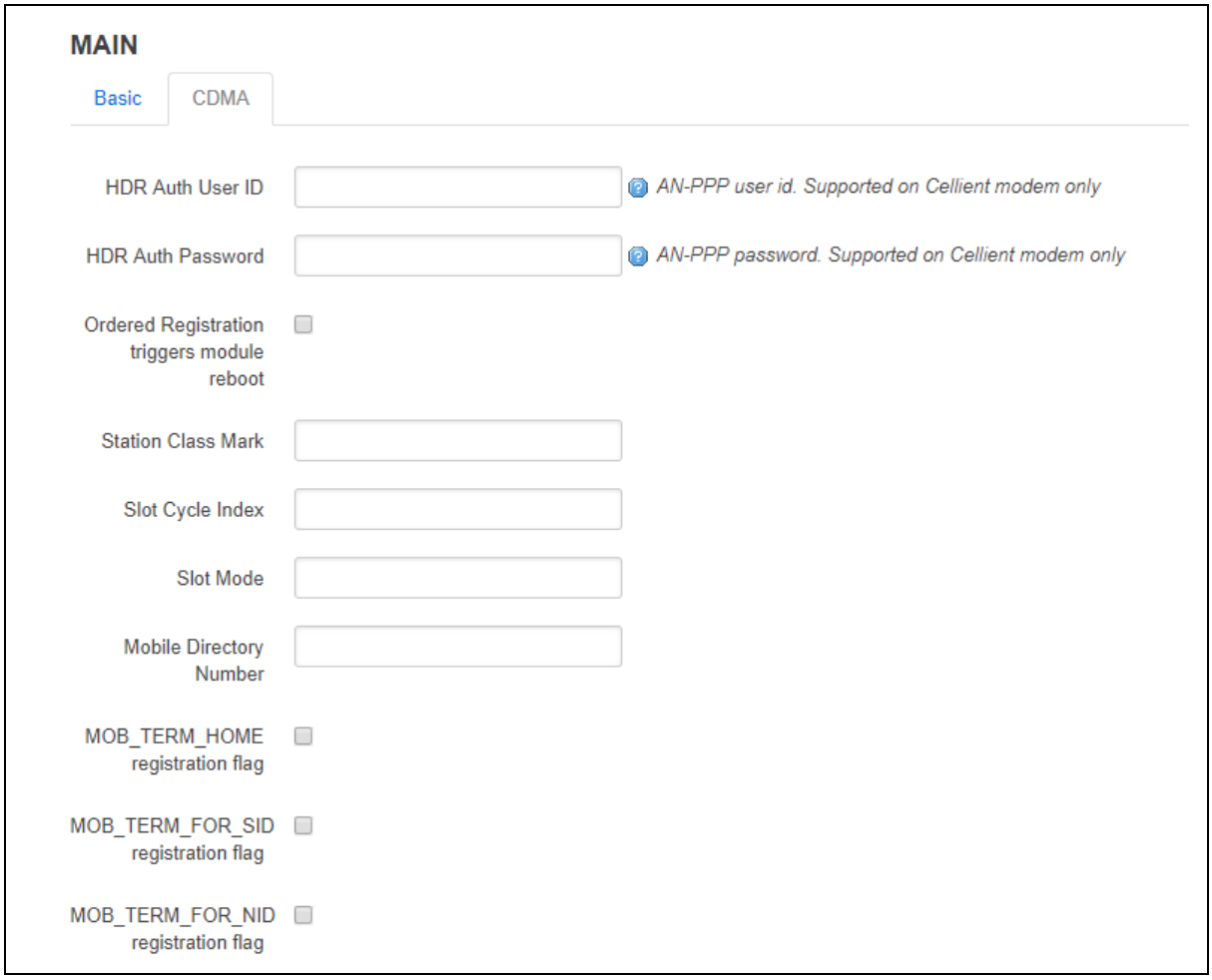



**Figure 98: The mobile manager CDMA page**

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

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

**Table 59: Information table for mobile manager CDMA settings**

### 21.2.5 Mobile manager: callers

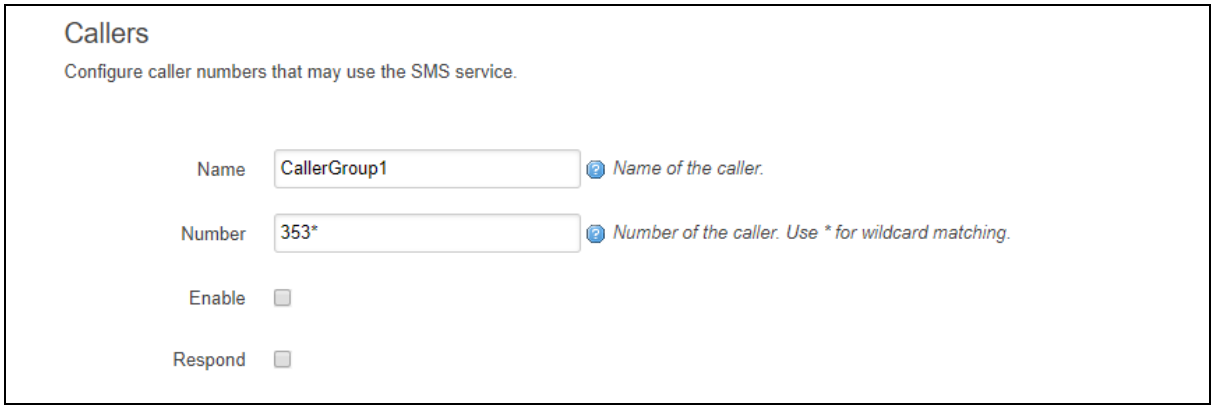



**Figure 99: The mobile manager CDMA page**

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

**Table 60: Information table for mobile manager callers settings**

### 21.2.6 Roaming interface template

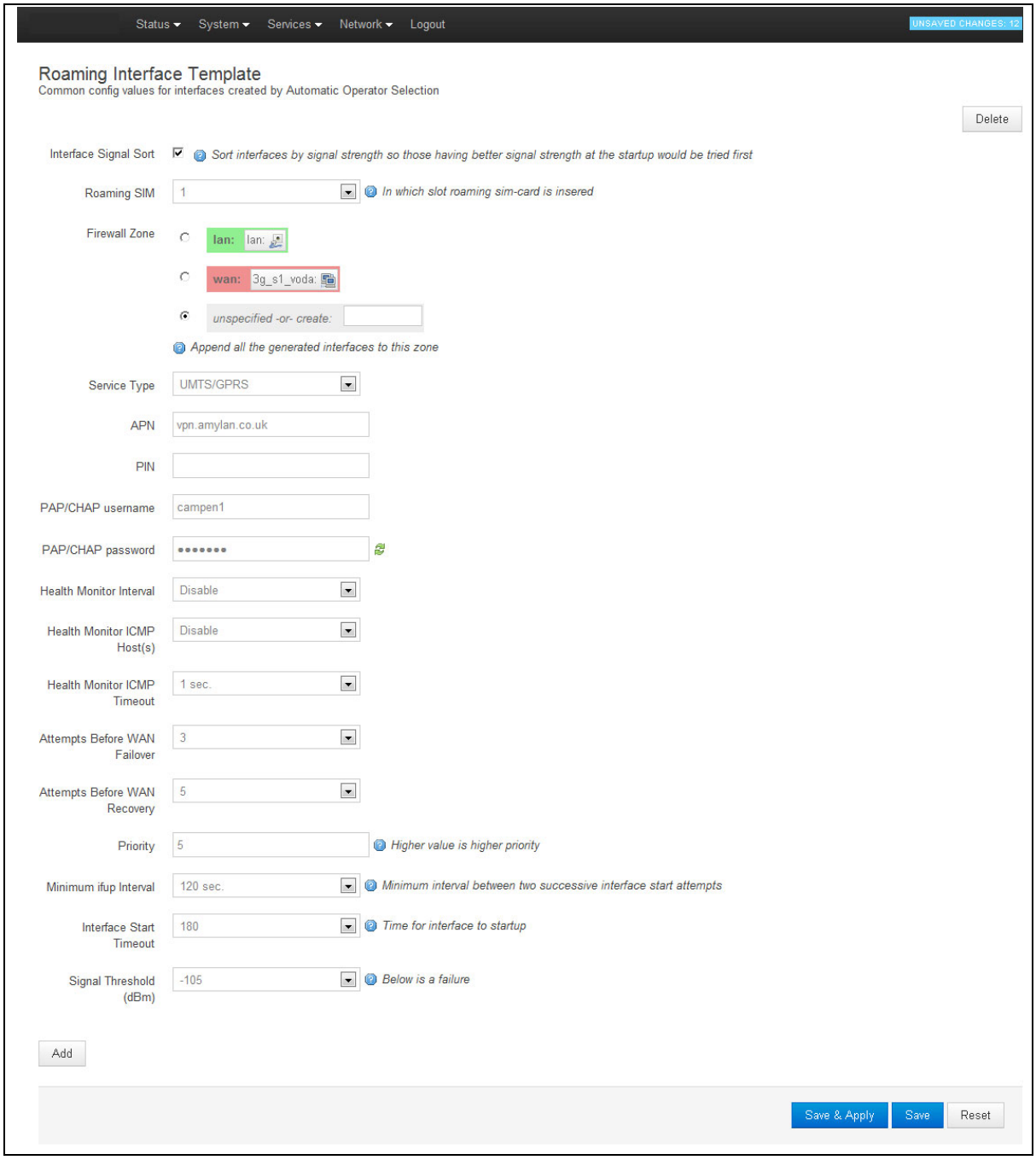



**Figure 100: The roaming interface template page**

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Interface Signal Sort<br>UCI: mobile.@roaming_template[0].sort_sig_strength<br>Opt: sort_sig_strength</td><td>Sorts interfaces by signal strength priority so those that have a better signal strength will be tried first.<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Roaming SIM<br>UCI: mobile.main.roaming_sim<br>Opt: roaming_sim</td><td>Sets in which slot to insert roaming SIM card.<table><tr><td>1</td><td>SIM slot 1.</td></tr><tr><td>2</td><td>SIM slot 2.</td></tr></table></td></tr>
<tr><td>Web: Firewall Zone<br>UCI: mobile.@roaming_template[0].firewall_zone<br>Opt: firewall_zone</td><td>Adds all generated interfaces to this zone. Select existing zone or click unspecified or create to create new zone.</td></tr>
<tr><td>Web: Service Type<br>UCI: mobile.@roaming_template[0].service<br>Opt: service</td><td>Specifies the service type that will be used to connect to the network.<table><tr><td>Auto</td><td>GSM module will automatically detect the best available technology code.</td></tr><tr><td>lte_only</td><td>Allows GSM module to only connect to LTE network.</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<br>UCI: mobile.@roaming_template[0].apn<br>Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: PIN<br>UCI: mobile.@roaming_template[0].pincode<br>Opt: pincode</td><td>SIM card’s PIN number.</td></tr>
<tr><td>Web: PAP/CHAP username<br>UCI: mobile.@roaming_template[0].username<br>Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: PAP/CHAP password<br>UCI: mobile.@roaming_template[0].password<br>Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: Health Monitor Interval<br>UCI: mobile.@roaming_template[0].health_interval<br>Opt: health_interval</td><td>Sets the period to check the health status of the interface. The Health Monitor interval will be used for:<ul><li>Interface state checks</li><li>Ping interval</li><li>Signal strength checks</li></ul></td></tr>
<tr><td>Web: Health Monitor ICMP Host(s)<br>UCI: mobile.@roaming_template[0].icmp_hosts<br>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<br>UCI: mobile.@roaming_template[0].timeout<br>Opt: timeout</td><td>Specifies the time in seconds that Health Monitor ICMP will timeout at.<br>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>
</table>

| | |
|---|---|
| Web: Attempts Before WAN Failover<br>UCI: mobile.@roaming_template[1].health_fail_retries<br>Opt: health_fail_retries | Defines the number of health check failures before interface is disconnected.<br>3 \| <br>Range \| |
| Web: Attempts Before WAN Recovery<br>UCI: mobile.@roaming_template[0].health_recovery_retries<br>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.<br>5 \| <br>Range \| |
| Web: Priority<br>UCI: mobile.@roaming_template[0].priority<br>Opt: priority | Type the priority number. The higher the value, the higher the priority.<br>This multi-WAN interface priority must be lower than the one specified in the priority field for the PMP interface.<br>0 \| <br>Range \| |
| Web: Minimum ifup interval<br>UCI: multiwan.wan.ifup_retry_sec<br>Opt: ifup_retry_sec | Not used for a roaming interface.<br>300 \| Retry primary interface every 300 seconds<br>Range \| |
| Web: Interface Start Timeout<br>UCI: mobile.@roaming_template[0].ifup_timeout_sec<br>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.<br>40 \| 40 seconds<br>Range \| |
| Web: Signal Threshold (dBm)<br>UCI: mobile.@roaming_template[0].signal_threshold<br>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.<br>Range \| -46 to -115 dBm<br>-115dBm \| |
| Web: n/a<br>UCI: mobile.main.service_order<br>Opt: service_order | Defines a space separated list of services, in preferred order. Valid options are `gprs, umts, lte, auto.`<br>If no valid_service order is defined, then the configured Service Type is used. Example:<br>`mobile.@roaming_template[0].service_order="gprs umts lte auto"`<br>Blank \| Use configured service type.<br>Range \| gprs umts lte auto |

**Table 61: 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 101: 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.

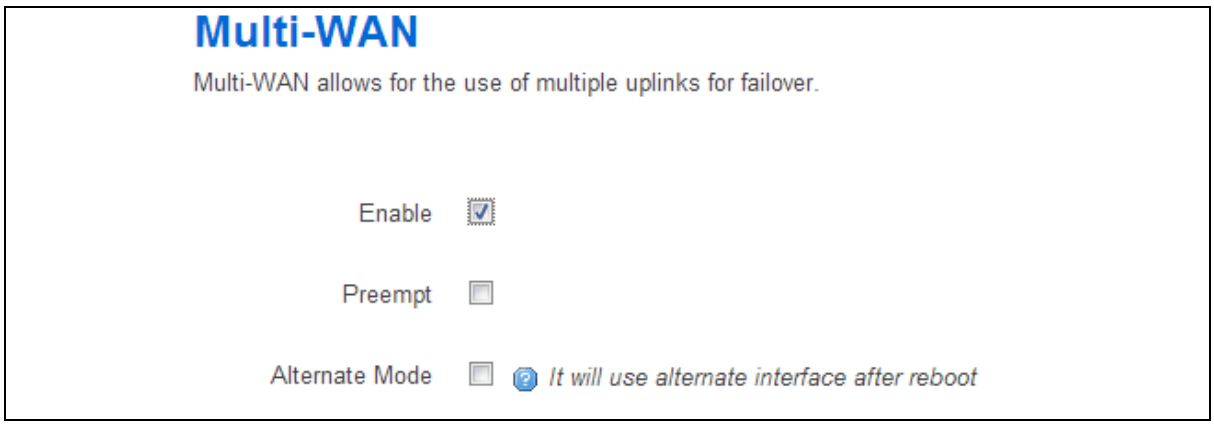



**Figure 102: The multi-wan page, pre-empt not selected**

Click **Save & Apply**.

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

**Figure 103: 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

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: SMS Enable<br>UCI: mobile.main.sms<br>Opt: sms | Enables SMS.<br>no — Disabled.<br>yes — Enabled. |
| Web: Collect ICCIDs<br>UCI: mobile.main.init_get_iccids<br>Opt: init_get_iccids | 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.<br>no — Disabled.<br>yes — Enabled. |
| Web: PIN code for SIM1<br>UCI: mobile.main.sim2pin<br>Opt: sim2pin | Depending on the SIM card specify the pin code for SIM 1.<br>Blank<br>range |
| Web: PIN code for SIM2<br>UCI: mobile.main.sim2pin<br>Opt: sim2pin | Depending on the SIM card specify the pin code for SIM 2.<br>Blank<br>Range |

| | |
|---|---|
| Web: HDR Auto User ID<br>UCI: mobile.main.hdr_userid<br>Opt: hdr_userid | AN-PPP user ID. Supported on Cellient (CDMA) modem only.<br>Blank<br>Range |

**Table 62: Information table for mobile manager basic settings**

### 21.2.9.2 Caller settings

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Name<br>UCI: mobile.@caller[0].name<br>Opt: name | Name assigned to the caller.<br>Blank<br>Range |
| Web: Number<br>UCI: mobile.@caller[0].number<br>Opt: number | Number of the caller allowed to SMS the router. Add in specific caller numbers, or use the wildcard symbol.<br>Blank<br>Range |
| Web: Enable<br>UCI: mobile.@caller[0].enabled<br>Opt: enabled | Enables or disables incoming caller ID.<br>no Disabled.<br>yes Enabled. |
| Web: Respond<br>UCI: mobile.@caller[0].respond<br>Opt: respond | If checked, the router will return an SMS. Select Respond if you want the router to reply.<br>0 Disabled.<br>1 Enabled. |

**Table 63: Information table for mobile manager caller settings**

## 21.2.10 Roaming interface template

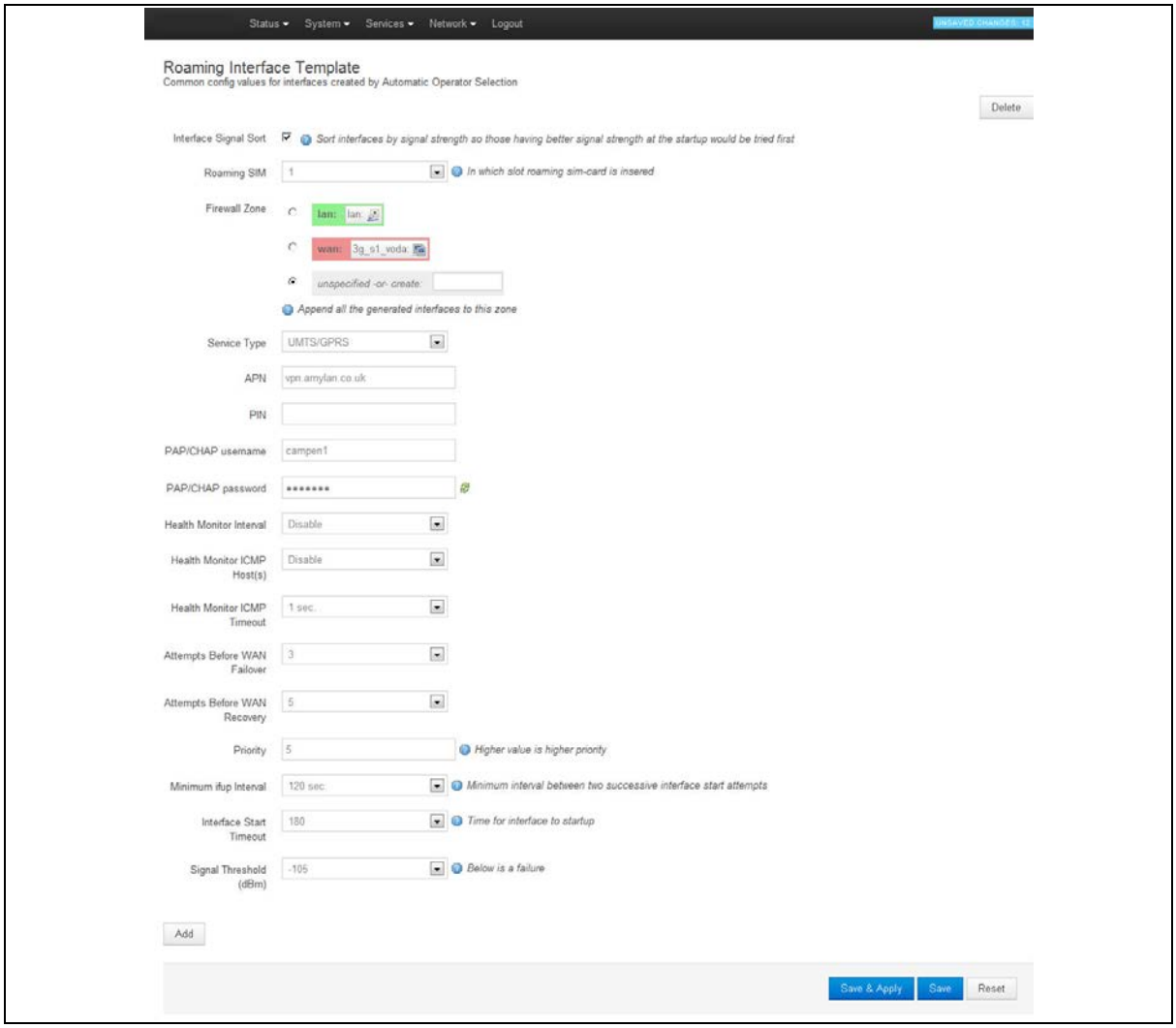



**Figure 104: The roaming interface template page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Interface Signal Sort<br>UCI: mobile.@roaming_template[0].sort_sig_strength<br>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<br>UCI: mobile.main.roaming_sim<br>Opt: roaming_sim | Sets which slot to insert roaming SIM card.<br>1 — SIM slot 1.<br>2 — SIM slot 2. |
| Web: Firewall Zone<br>UCI: mobile.@roaming_template[0].firewall_zone<br>Opt: firewall_zone | Adds all generated interfaces to this zone.<br>Select existing zone or click unspecified or create to create a new zone. |

| | |
|---|---|
| Web: Service Type<br>UCI: mobile.@roaming_template[0].service<br>Opt: service | Specifies the service type that will be used to connect to the network.<br>UMTS/GPRS — GSM module will automatically detect the best available technology code.<br>Umts_only — Allows GSM module to only connect to 3G network.<br>GPRS_only — Allows GSM module to only connect to GPRS network.<br>cdma — Allows GSM module to only connect to cdma network. |
| Web: APN<br>UCI: mobile.@roaming_template[0].apn<br>Opt: apn | APN name of Mobile Network Operator. |
| Web: PIN<br>UCI: mobile.@roaming_template[0].pincode<br>Opt: pincode | SIM card's PIN number. |
| Web: PAP/CHAP username<br>UCI: mobile.@roaming_template[0].username<br>Opt: username | Username used to connect to APN. |
| Web: PAP/CHAP password<br>UCI: mobile.@roaming_template[0].password<br>Opt: password | Password used to connect to APN. |
| Web: Health Monitor Interval<br>UCI: mobile.@roaming_template[0].health_interval<br>Opt: health_interval | Sets the period to check the health status of the interface. The Health Monitor interval will be used for:<br>• Interface state checks<br>• Ping interval<br>• Signal strength checks |
| Web: Health Monitor ICMP Host(s)<br>UCI: mobile.@roaming_template[0].icmp_hosts<br>Opt: icmp_hosts | Specifies target IP address for ICMP packets.<br>Disable — Disables the option<br>DNS servers — DNS IP addresses will be used.<br>WAN gateway — Gateway IP address will be used.<br>custom — Ability to provide IP address. |
| Web: Health Monitor ICMP Timeout<br>UCI: mobile.@roaming_template[0].timeout<br>Opt: timeout | Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.<br>3 — Wait 3 seconds for ping reply<br>Range |
| Web: Attempts Before WAN Failover<br>UCI: mobile.@roaming_template[1].health_fail_retries<br>Opt: health_fail_retries | Defines the number of health check failures before interface is disconnected.<br>3<br>Range |
| Web: Attempts Before WAN Recovery<br>UCI: mobile.@roaming_template[0].health_recovery_retries<br>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. |
| Web: Priority<br>UCI: mobile.@roaming_template[0].priority<br>Opt: priority | Type the priority number. The higher the value, the higher the priority.<br>0<br>Range |

| | |
|---|---|
| Web: Minimum ifup interval<br>UCI: mobile.@roaming_template[0].ifup_retry_sec<br>Opt: ifup_retry_sec | Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled.<br>300 — Retry primary interface every 300 seconds.<br>Range |
| Web: Interface Start Timeout<br>UCI: mobile.@roaming_template[0].ifup_timeout_sec<br>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.<br>40<br>Range |
| Web: Signal Threshold (dBm)<br>UCI: mobile.@roaming_template[0].signal_threshold<br>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.<br>Disabled<br>Range — -46 to -115 dBm |

**Table 64: 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.

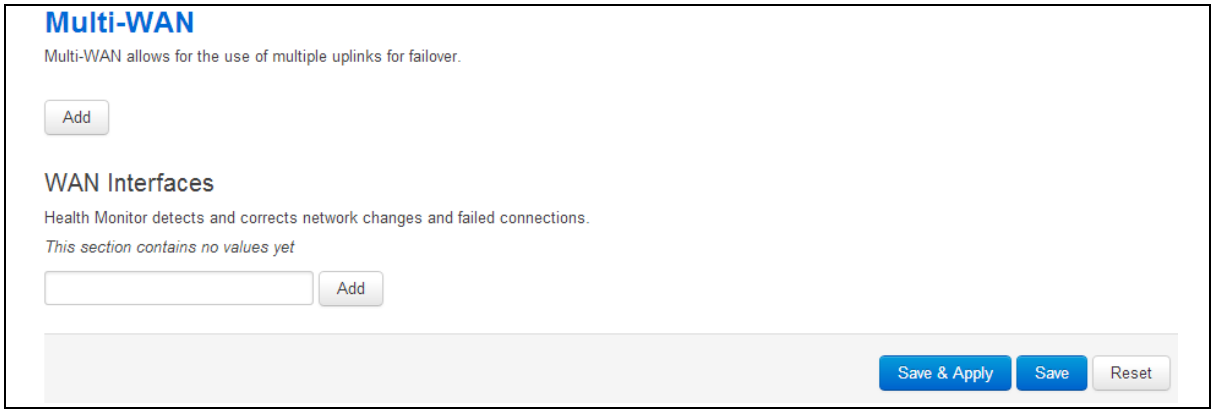



**Figure 105: The multi-WAN page**

In the Multi-WAN section click **Add**.

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Enable<br>UCI: multiwan.config.enabled<br>Opt: enabled | Enables multiwan.<br>Select this option.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Preempt<br>UCI: multiwan.config.preempt<br>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.<br>Leave this option unselected.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Alternate Mode<br>UCI: multiwan.config.alt<br>Opt: alt | Enables or disables alternate mode for multiwan. If enabled the router will use an alternate interface after reboot.<br>Leave this option unselected.<br>0 — Disabled.<br>1 — Enabled. |

**Table 65: 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**.

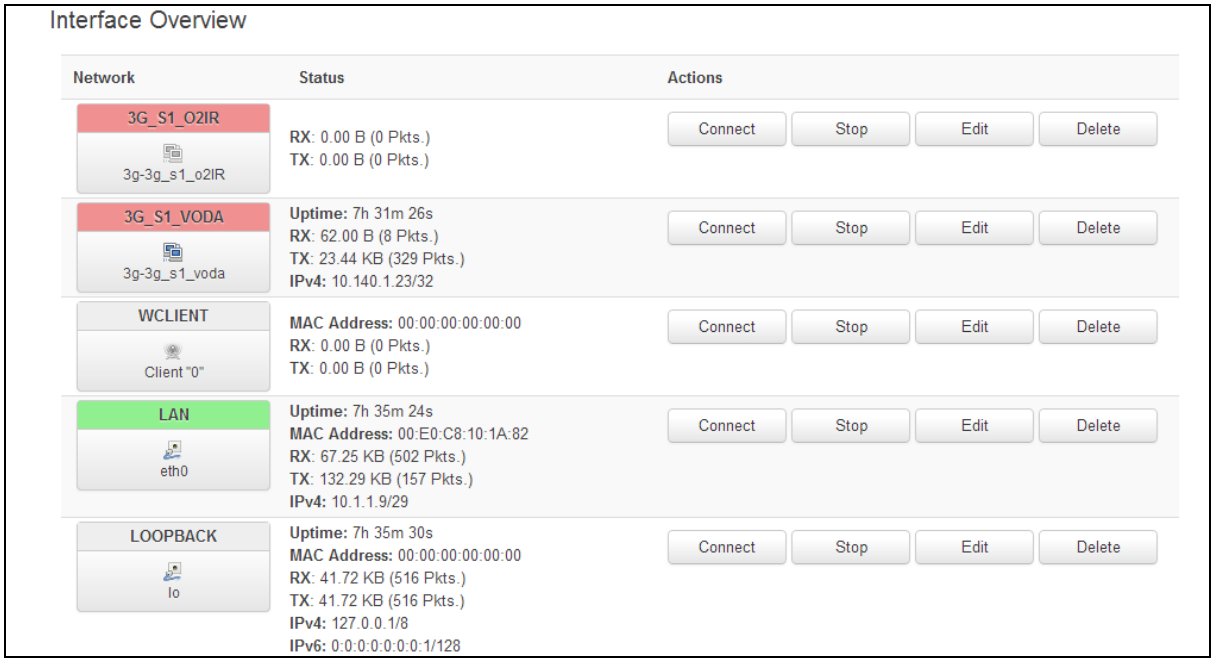



**Figure 106: The interface overview page**

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.

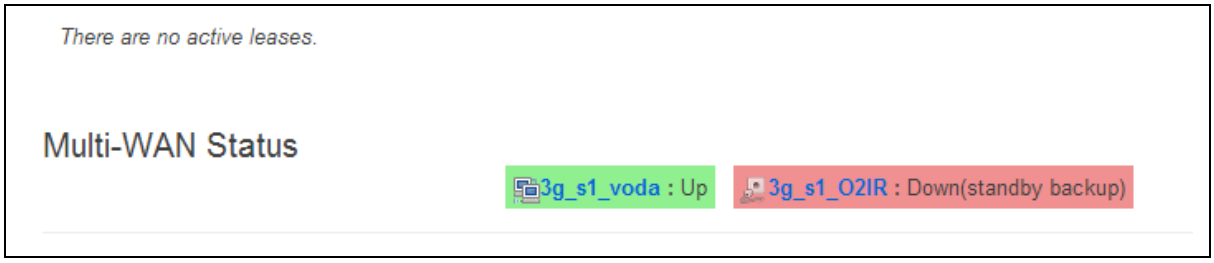



**Figure 107: The status page: multi-WAN status section page**

## 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 108: 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 109: 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 110: 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 111: The add connection watch configuration page**

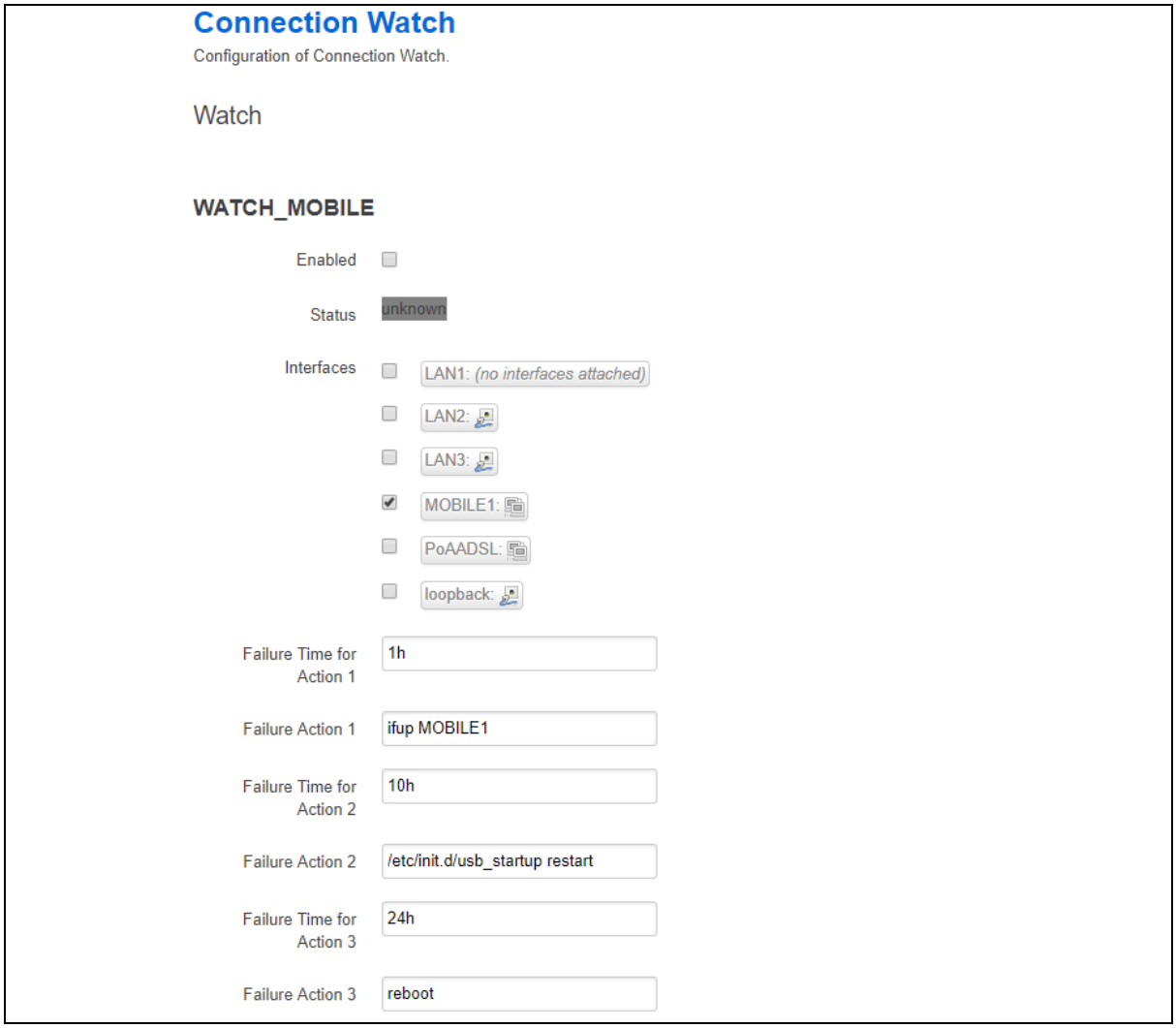



**Figure 112: The connection watch configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: cwatch.@watch[0].enabled<br>Opt: enabled | Enables a cwatch instance.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Interfaces<br>UCI: cwatch.@watch[0].test_ifaces<br>Opt: test_ifaces | Defines the interface name(s) to monitor. Multiple interfaces are delimited by space separator. Example:<br>`option test_ifaces 'WANADSL WANMOBILE`<br>If multiple interfaces are defined the failure action will only be triggered if no traffic is received on all interfaces for the defined period. |
| Web: Failure Time for Action 1<br>UCI: cwatch.@watch[0].failure_time_1<br>Opt: failure_time_1 | Defines a duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days.<br>1h<br>Range — s; m; h; d; |
| Web: Failure Action 1<br>UCI: cwatch.@watch[0].failure_action_1<br>Opt: failure_action_1 | Defines the failure action associated with failure_time_1. Example to force up interface:<br>`option failure_action_1 'ifup wan'`<br>blank<br>Range |

| | |
|---|---|
| Web: Failure Time for Action 2<br>UCI: cwatch.@watch[0].failure_time_2<br>Opt: failure_time_2 | Defines a second duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days.<br>10h<br>Range: s; m; h; d; |
| Web: Failure Action 2<br>UCI: cwatch.@watch[0].failure_action_2<br>Opt: failure_action_2 | Defines the failure action associated with failure_time_2. Example to reset usb:<br>`option failure_action_1 '/etc/init.d/usb_startup restart'`<br>blank<br>Range |
| Web: Failure Time for Action 3<br>UCI: cwatch.@watch[0].failure_time_3<br>Opt: failure_time_3 | Defines a third duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days.<br>24h<br>Range: s; m; h; d; |
| Web: Failure Action 3<br>UCI: cwatch.@watch[0].failure_action_3<br>Opt: failure_action_3 | Defines the failure action associated with failure_time_3. Example to reset usb:<br>`option failure_action_3 'reboot'`<br>blank<br>Range |

**Table 66: 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.

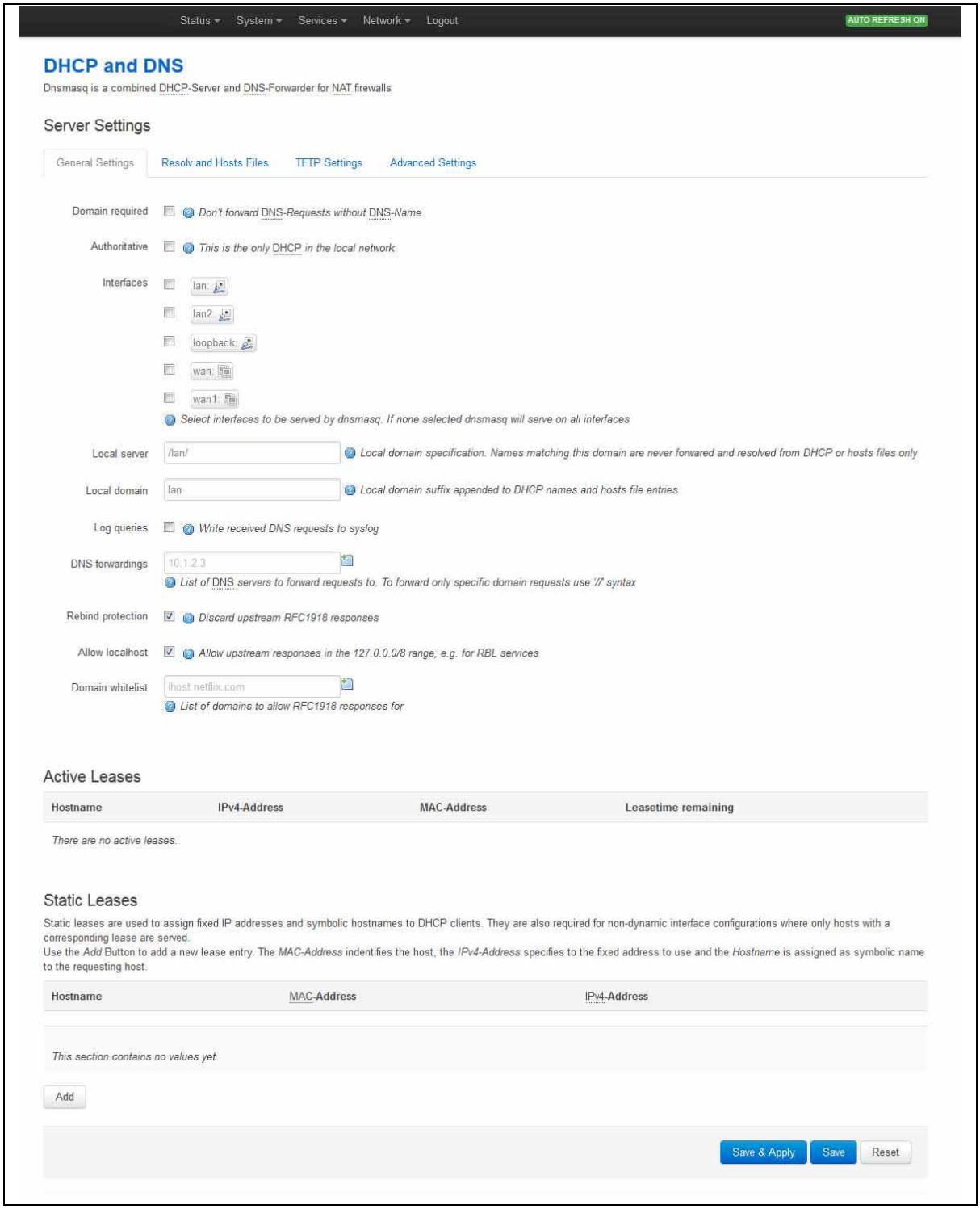



**Figure 113: The DHCP and DNS page**

### 23.2.1 Dnsmasq: general settings

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Domain required<br>UCI: dhcp.@dnsmasq[0].domainneeded<br>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.<br>1 – Enabled.<br>0 – Disabled. |
| Web: Authoritative<br>UCI: dhcp.@dnsmasq[0]. authoritative<br>Opt: authoritative | Forces authoritative mode, this speeds up DHCP leasing. Used if this is the only server in the network.<br>1 – Enabled.<br>0 – Disabled. |
| Web: Interfaces<br>UCI: dhcp.@dnsmasq[0].interface<br>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.<br>Lan – Serve only on LAN interface.<br>Range |
| Web: Local Server<br>UCI: dhcp.@dnsmasq[0].local<br>Opt: local | Specifies the local domain. Names matching this domain are never forwarded and are resolved from DHCP or host files only.<br>/lan/<br>Range |
| Web: Local Domain<br>UCI: dhcp.@dnsmasq[0].domain<br>Opt: domain | Specifies local domain suffix appended to DHCP names and hosts file entries.<br>lan<br>Range |
| Web: Log Queries<br>UCI: dhcp.@dnsmasq[0].logqueries<br>Opt: logqueries | Writes received DNS requests to syslog.<br>0 – Disabled.<br>1 – Enabled. |
| Web: DNS Forwardings<br>UCI: dhcp.@dnsmasq[0].server<br>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.<br>No DNS server configured.<br>Range |
| Web: Rebind Protection<br>UCI: dhcp.@dnsmasq[0].rebind_protection<br>Opt: rebind_protection | Enables DNS rebind attack protection by discarding upstream RFC1918 responses.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Allow Localhost<br>UCI: dhcp.@dnsmasq[0].rebind_localhost<br>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.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Domain Whitelist<br>UCI: dhcp.@dnsmasq[0].rebind_domain<br>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.<br>No list configured.<br>Range |

**Table 67: Information table for general server settings**

### 23.2.2 Dnsmasq: resolv and host files

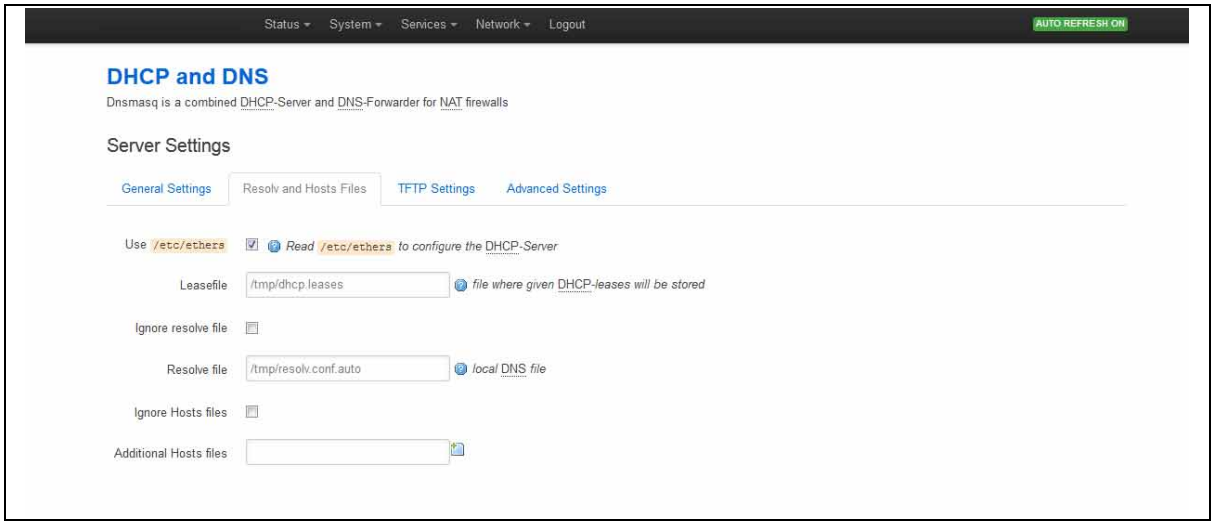



**Figure 114: The resolv and host files section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Use /etc/ethers<br>UCI: dhcp.@dnsmasq[0].readethers<br>Opt: readethers | Defines whether static lease entries are read from /etc/ethers.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Leasefile<br>UCI: dhcp.@dnsmasq[0].leasefile<br>Opt: leasefile | 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.<br>/tmp/dhcp.leases — Store DHCP leases in this file.<br>Range |
| Web: Ignore resolve file<br>UCI: dhcp.@dnsmasq[0].noresolv<br>Opt: noresolv | Defines whether to use the local DNS file for resolving DNS.<br>0 — Use local DNS file.<br>1 — Ignore local DNS file. |
| Web: Resolve file<br>UCI: dhcp.@dnsmasq[0].resolvfile<br>Opt: resolvfile | Defines the local DNS file. Default is /tmp/resolv.conf.auto |
| Web: Ignore Hosts files<br>UCI: dhcp.@dnsmasq[0].nohosts<br>Opt: nohosts | Defines whether to use local host's files for resolving DNS.<br>0 — Use local hosts file.<br>1 — Ignore local hosts file. |
| Web: Additional Hosts files<br>UCI: dhcp.@dnsmasq[0].addnhosts<br>Opt: list addnhosts | Defines local host's files. When using UCI multiple servers should be entered with a space between them. |

**Table 68: Information table for resolv and host files section**

### 23.2.3 Dnsmasq: TFTP settings

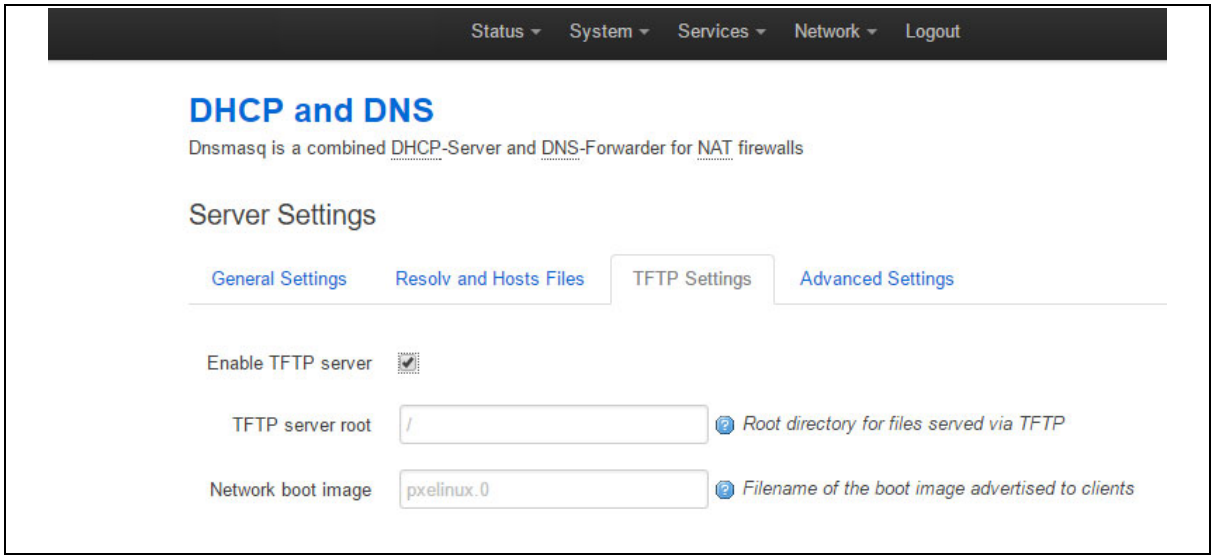



**Figure 115: The TFTP settings section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable TFTP Server<br>UCI: dhcp.@dnsmasq[0].enable_tftp<br>Opt: enable_tftp | Enables the TFTP server.<br>0 Disabled.<br>1 Enabled. |
| Web: Enable TFTP Server<br>UCI: dhcp.@dnsmasq[0].tftp_root<br>Opt: tftp_root | Defines root directory for file served by TFTP. |
| Web: Enable TFTP Server<br>UCI: dhcp.@dnsmasq[0].dhcp_boot<br>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 69: Information table for TFTP settings**

### 23.2.4 Dnsmasq: advanced settings

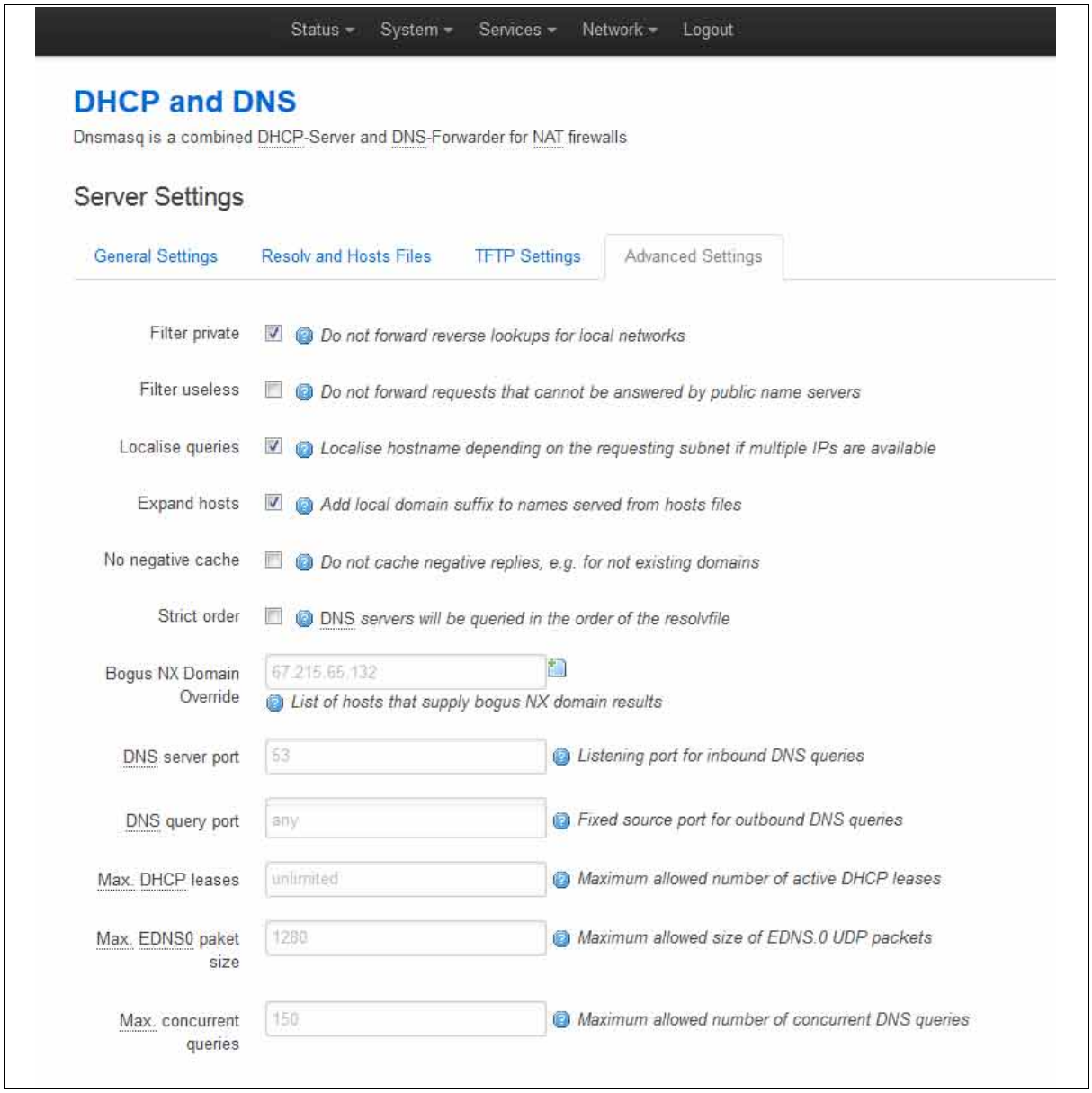



**Figure 116: The advanced settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Filter private<br>UCI: dhcp.@dnsmasq[0].<br>Opt: boguspriv | 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.<br>1 – Enabled.<br>0 – Disabled. |
| Web: Filter useless<br>UCI: dhcp.@dnsmasq[0].filterwin2k<br>Opt: filterwin2k | Enables disallow option for forwarding requests that cannot be answered by public name servers. Normally enabled for dial on demand interfaces.<br>1 – Enabled.<br>0 – Disabled. |

| | |
|---|---|
| Web: Localise queries<br>UCI: dhcp.@dnsmasq[0].localise_queries<br>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.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Expand hosts<br>UCI: dhcp.@dnsmasq[0].expandhosts<br>Opt: expandhosts | Adds a local domain suffix to names served from host files.<br>1 — Enabled.<br>0 — Disabled. |
| Web: No negative cache<br>UCI: dhcp.@dnsmasq[0].nonegcache<br>Opt: nonegcache | Enable this to stop caching of negative replies. For example, non-existing domains.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Strict order<br>UCI: dhcp.@dnsmasq[0].strictorder<br>Opt: strictorder | Enable this to query DNS servers in the order of the resolve file.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Bogus NX Domain override<br>UCI: dhcp.@dnsmasq[0].bogusnxdomain<br>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.<br>Empty list<br>Range |
| Web: DNS server port<br>UCI: dhcp.@dnsmasq[0].port<br>Opt: port | Listening port for inbound DNS queries.<br>53 — Set to 0 to disable DNS functionality.<br>Range — 0 - 65535 |
| Web: DNS query port<br>UCI: dhcp.@dnsmasq[0].queryport<br>Opt: queryport | Defines fixed source port for outbound DNS queries.<br>any<br>Range — any; 0 - 65535 |
| Web: Max DHCP leases<br>UCI: dhcp.@dnsmasq[0].dhcpleasemax<br>Opt:dhcpleasemax | Defines the maximum allowed number of active DHCP leases.<br>unlimited<br>Range |
| Web: Max EDNS0 packet size<br>UCI: dhcp.@dnsmasq[0].ednspacket_max<br>Opt: ednspacket_max | Defines the maximum allowed size of EDNS.0 UDP packets in bytes.<br>1280 — 1280 bytes<br>Range |
| Web: Max concurrent queries<br>UCI: dhcp.@dnsmasq[0].dnsforwardmax<br>Opt: dnsforwardmax | Maximum allowed number of concurrent DNS queries.<br>150 — 1280 bytes<br>Range |

**Table 70: Information table for advanced settings**

### 23.2.5 Active leases

This section displays all currently active leases.

**Figure 117: The active leases section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Hostname<br>UCI: dhcp.@host[0].name<br>Opt: name | Displays the hostname of the client. |
| Web: IPv4 Address<br>UCI: dhcp.@host[0].ip<br>Opt: ip | Displays the IP address of the client. |
| Web: MAC Address<br>UCI: dhcp.@host[0].mac<br>Opt: mac | Displays the MAC address of the client. |
| Web: Lease time remaining<br>UCI: n/a<br>Opt: n/a | Displays the remaining lease time. |

**Table 71: 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.

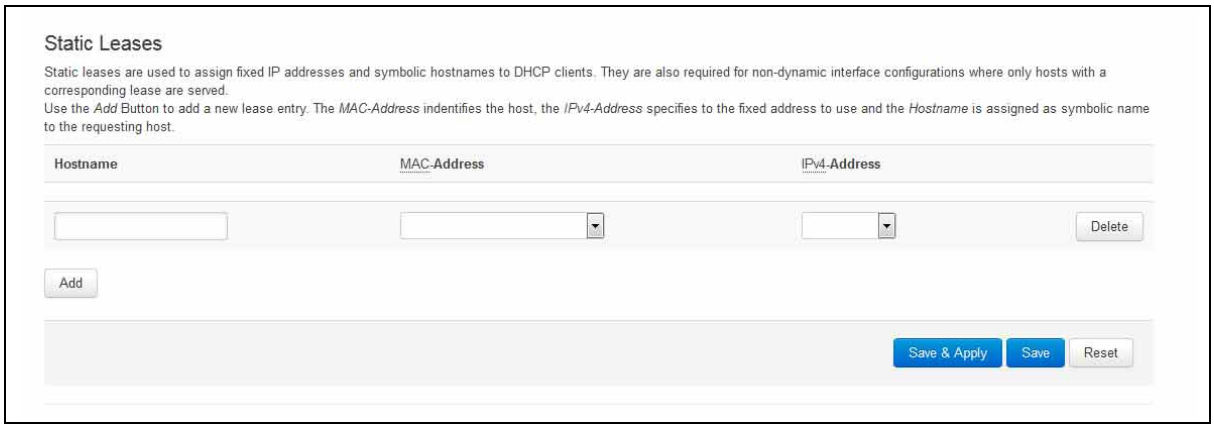



**Figure 118: The static leases section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Hostname<br>UCI: dhcp.@host[0].name<br>Opt: name | Defines the optional symbolic name to assign to this static DHCP entry.<br>1 — Enabled.<br>0 — Disabled. |
| Web: MAC Address<br>UCI: dhcp.@host[0].mac<br>Opt: mac | Defines the hardware address that identifies the host. |
| Web: IPv4 Address<br>UCI: dhcp.@host[0].ip<br>Opt: ip | The IPv4 address specifies the fixed address to use for this host. |

**Table 72: 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
```

| UCI / Package Option | Description |
|---|---|
| Web: n/a<br>UCI: dhcp.<pool_name>.interface<br>Opt: interface | Defines the interface that is served by this DHCP pool. This must be one of the configured interfaces.<br>lan — Enabled.<br>Range |
| Web: n/a<br>UCI: dhcp.<pool_name>.start<br>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.<br>100<br>Range |
| Web: n/a<br>UCI: dhcp.<pool_name>.limit<br>Opt: limit | Defines the offset from the network address for the end of the DHCP pool.<br>150<br>Range — 0 - 255 |

| | |
|---|---|
| Web: n/a<br>UCI: dhcp.<pool_name>.leasetime<br>Opt: leasetime | Defines the lease time of addresses handed out to clients, for example 12h or 30m.<br>12h: 12 hours<br>Range |
| Web: n/a<br>UCI: dhcp.<pool_name>.ignore<br>Opt: ignore | Defines whether this DHCP pool is enabled.<br>0: DHCP pool enabled.<br>1: DHCP pool disabled. |
| Web: n/a<br>UCI: dhcp.<pool_name>.force<br>Opt: force | Forces DHCP serving on the specified interface even if another DHCP server is detected on the same network segment.<br>0: Disabled.<br>1: Enabled. |
| Web: n/a<br>UCI: dhcp.<pool_name>.dhcp_option<br>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.<br>: No options defined.<br>Syntax: Option_number, option_value. |
| Web: n/a<br>UCI: dhcp.<pool_name>.dynamicdhcp<br>Opt: dynamicdhcp | Defines whether to allocate DHCP leases.<br>1: Dynamically allocate leases.<br>0: Use /etc/ethers file for serving DHCP leases. |
| Web: n/a<br>UCI: dhcp.<pool_name>.dynamicdhcp<br>Opt: networkid | Assigns a network-id to all clients that obtain an IP address from this pool. |

**Table 73: 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.

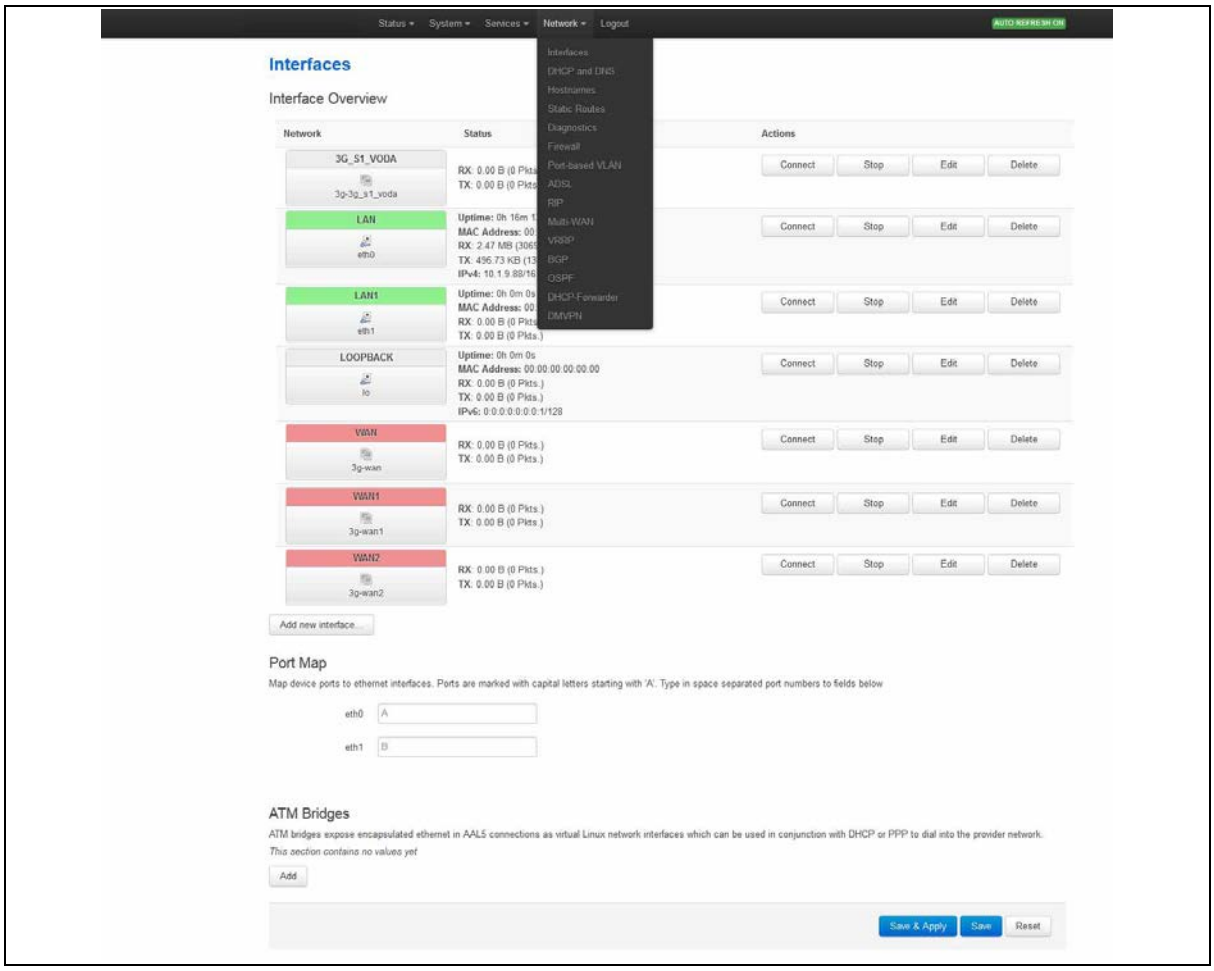



**Figure 119: The interfaces overview page**

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.

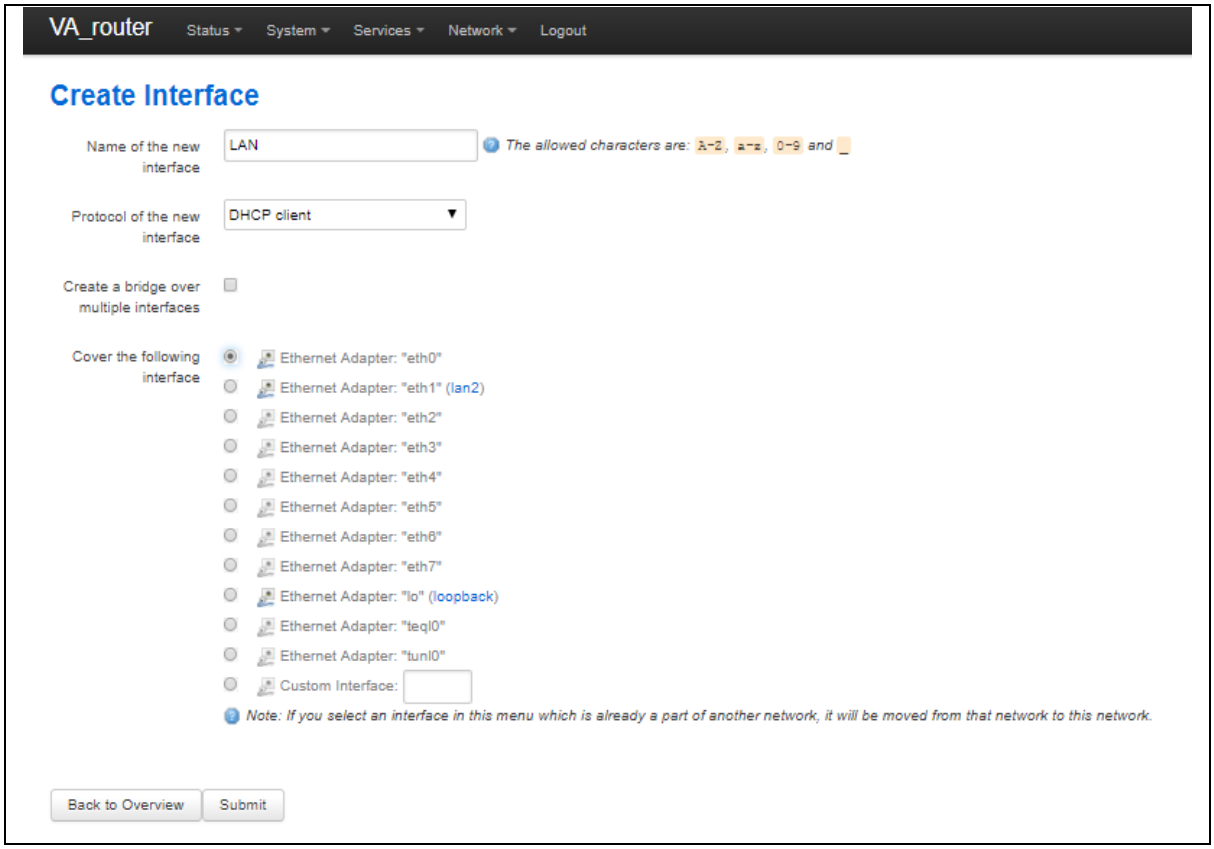



**Figure 120: The create interface page**

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface<br>UCI: network.<if name><br>Opt: config interface</td><td>Assigns a logical name to the interface. The network interface section will assign this name (<if name>).<br>Type the name of the new interface.<br>Allowed characters are A-Z, a-z, 0-9 and _</td></tr>
<tr><td>Web: Protocol of the new interface<br>UCI: network.<if name>.proto<br>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<br>UCI: network.<if name>.type<br>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<br>UCI: network.<if name>.ifname<br>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 74: 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

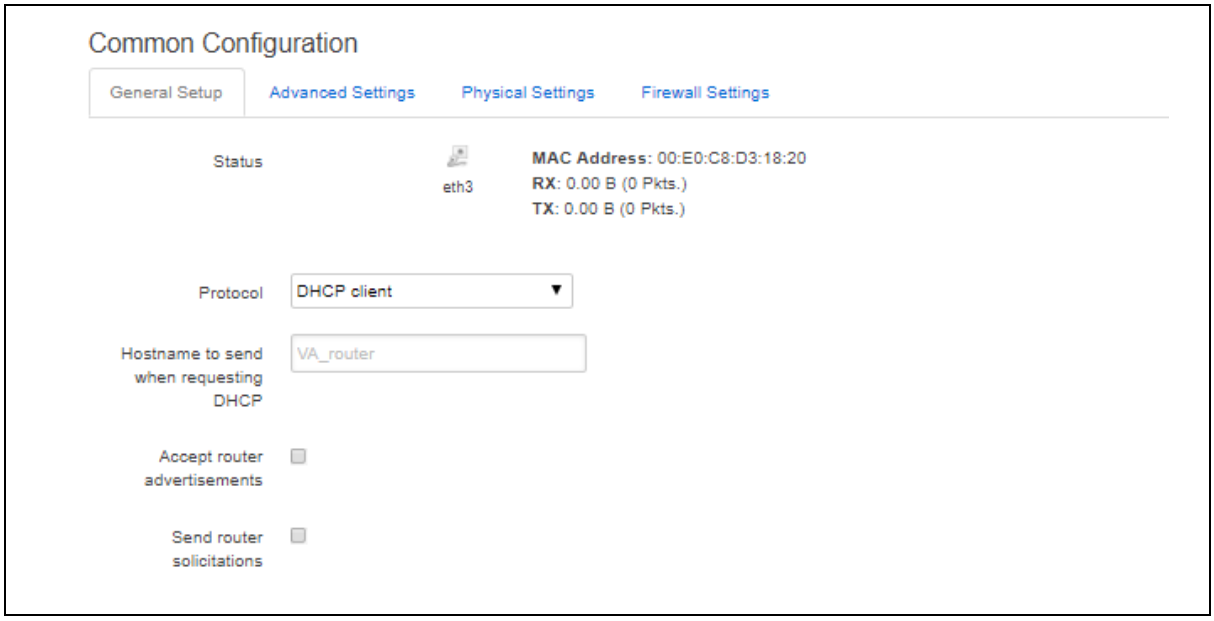



**Figure 121: The interface general setup configuration page for DHCP client protocol**

<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<br>UCI: network.<if name>.proto<br>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.<br>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<br>UCI: network.<if name>.hostname<br>Opt: hostname</td><td>Defines the hostname to include in DHCP requests</td></tr>
<tr><td>Web: Accept router advertisements<br>UCI: network.<if name>.accept_ra<br>Opt: accept_ra</td><td>Specifies whether to accept IPv6 Router Advertisements on this interface (optional).<br>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<br>UCI: network.<if name>.send_rs<br>Opt: send_rs</td><td>Specifies whether to send router Solicitations on this interface (optional).<br>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 75: Information table for general setup configuration settings for DHCP client protocol**

### 24.2.3.2 Common configuration: advanced settings

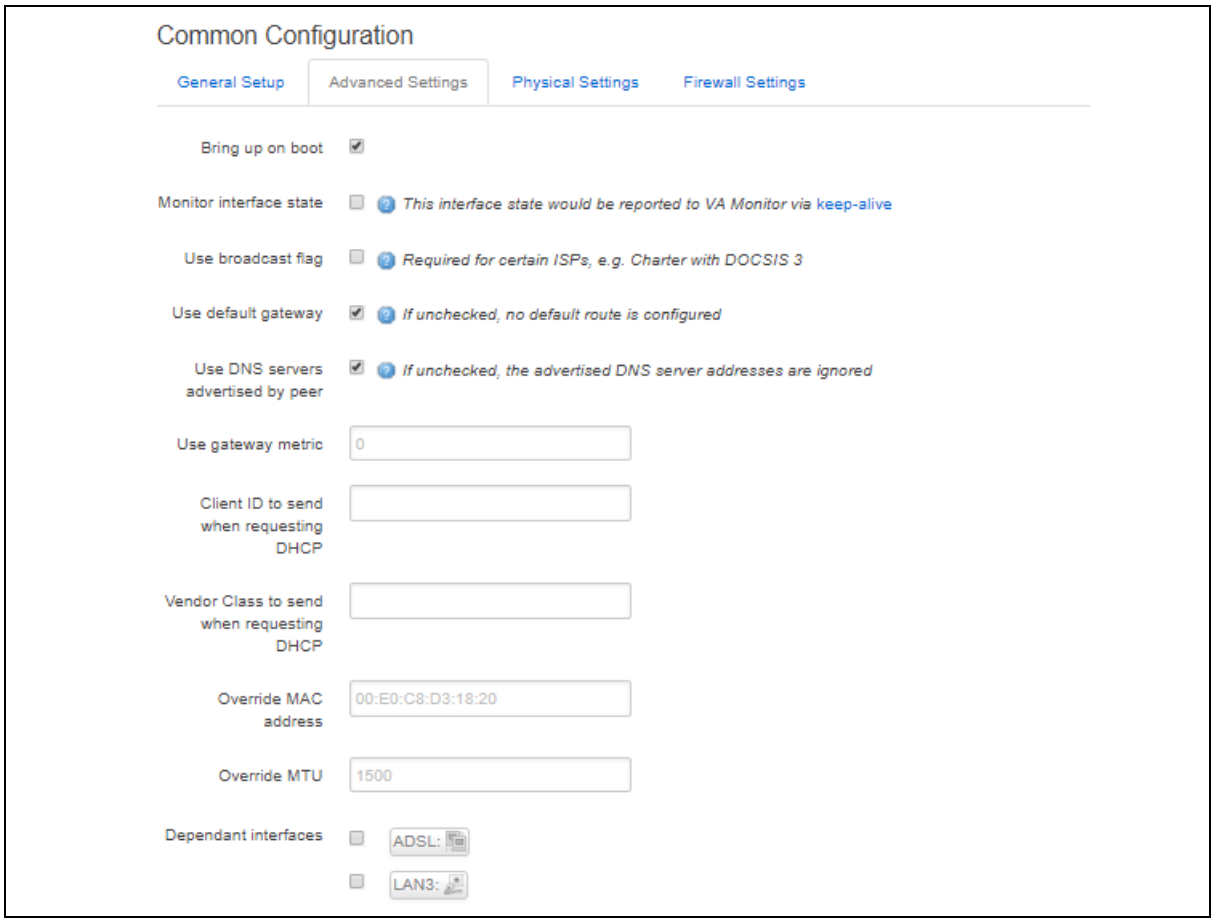



**Figure 122: The interface advanced settings page for DHCP client protocol**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Bring up on boot<br>UCI: network.<if name>.auto<br>Opt: auto | Enables the interface to connect automatically on boot up.<br>0 Disabled.<br>1 Enabled. |
| Web: Monitor interface state<br>UCI: network.<if name>.monitored<br>Opt: monitored | Enabled if status of interface is presented on Monitoring platform.<br>0 Disabled.<br>1 Enabled. |
| Web: Use broadcast flag<br>UCI: network.<if name>.broadcast<br>Opt: broadcast | Enables the broadcast flag in DHCP requests (required for certain ISPs).<br>0 Disabled.<br>1 Enabled. |
| Web: Use default gateway<br>UCI: network.<if name>.gateway<br>Opt: gateway | Defines whether to suppress the DHCP assigned default gateway. When disabled via web option gateway is set to 0.0.0.0.<br>0 Disabled (option gateway set to 0.0.0.0)<br>1 Enabled. |
| Web: Use DNS servers advertised by peer<br>UCI: n/a<br>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. |

| | |
|---|---|
| Web: Use custom DNS servers<br>UCI: network.<if name>.dns<br>Opt: dns | Defines whether to override DHCP assigned DNS servers with configured list of DNS servers.<br>Multiple DNS Servers are separated by a space if using UCI.<br>Example: `option dns '1.1.1.1 2.2.2.2'`<br>0 – Disabled (option gateway set to 0.0.0.0)<br>1 – Enabled. |
| Web: Use gateway metric<br>UCI: network.<if name>.metric<br>Opt: metric | Specifies the default route metric to use for this interface.<br>0 – Disabled.<br>Range |
| Web: Client ID to send when requesting DHCP<br>UCI: network.<if name>. clientid<br>Opt: clientid | Defines whether to override the client identifier in DHCP requests.<br>Blank – Do not override.<br>Range – Override. |
| Web: Vendor Class to send when requesting DHCP<br>UCI: network.<if name>.vendorid<br>Opt: vendorid | Defines whether to override the vendor class in DHCP requests.<br>Blank – Do not override.<br>Range – Override. |
| Web: Override MAC address<br>UCI: network.<if name>.macaddr<br>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<br>UCI: network.<if name>.mtu<br>Opt: mtu | Defines the value to override the default MTU on this interface.<br>1500 – 1500 bytes |
| Web: Dependant Interfaces<br>UCI: network.[if_name].dependants<br>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.<br>Separate multiple interfaces by a space when using UCI.<br>Example: `option dependants 'PPPADSL MOBILE'`<br>This replaces the following previous options in child interfaces.<br>gre – option local_interface<br>lt2p – option src_ipaddr<br>iot – option wan1 wan2<br>6in4 – option ipaddr<br>6to4 – option ipaddr |
| Web: SNMP Alias ifIndex<br>UCI: network.@interface[X].snmp_alias_ifindex<br>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*)<br>Blank – No SNMP interface alias index<br>Range – 0 - 4294966295 |

**Table 76: 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.

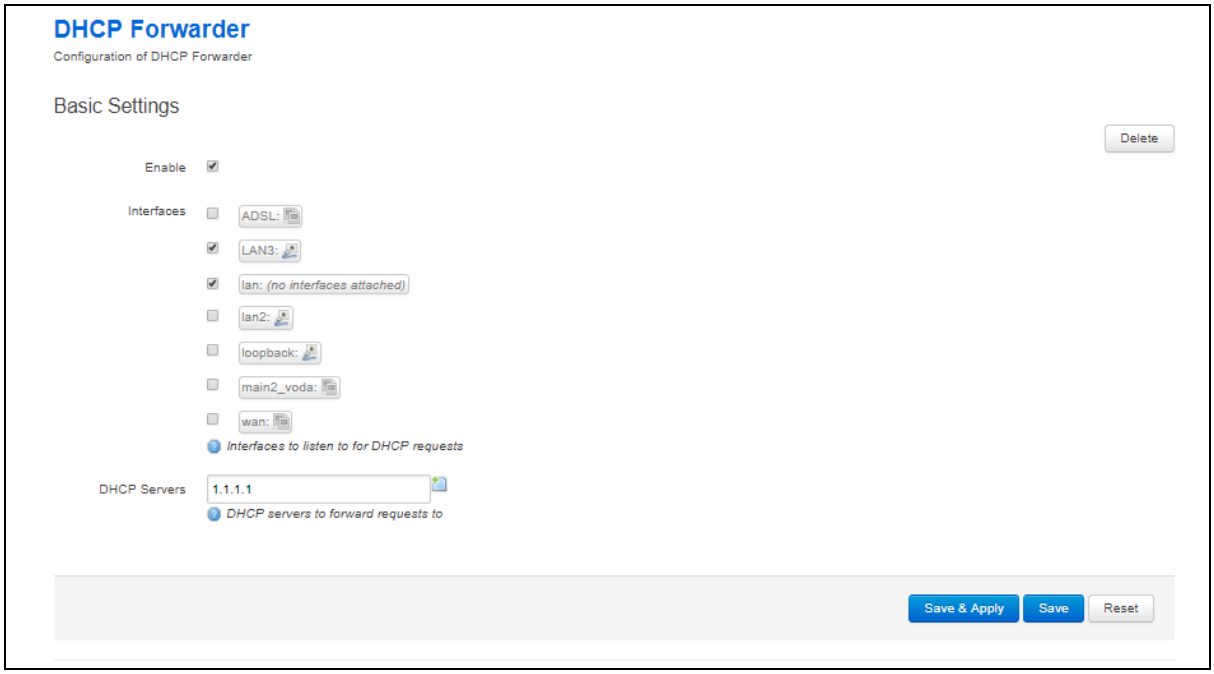



**Figure 123: The DHCP forwarder configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: dhcp_fwd.main.enabled<br>Opt: enabled | Defines whether DHCP forwarding is enabled or disabled.<br>0 — Do not send router solicitations.<br>1 — Send router solicitations. |
| Web: Interfaces<br>UCI: dhcp_fwd.main.listen_interface<br>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:<br>uci set dhcp_fwd.main.listen_interface=LAN1<br>uci add_list dhcp_fwd.main.listen_interface=LAN2<br>or using a list of options via package options<br>list listen_interface 'LAN1'<br>list listen_interface 'LAN2' |
| Web: DHCP Servers<br>UCI: dhcp_fwd.main.server<br>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:<br>uci set dhcp_fwd.main.server=1.1.1.1<br>uci add_list dhcp_fwd.main.main.server=2.2.2.2<br>or using a list of options via package options<br>list server '1.1.1.1'<br>list server '2.2.2.2' |

**Table 77: 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.

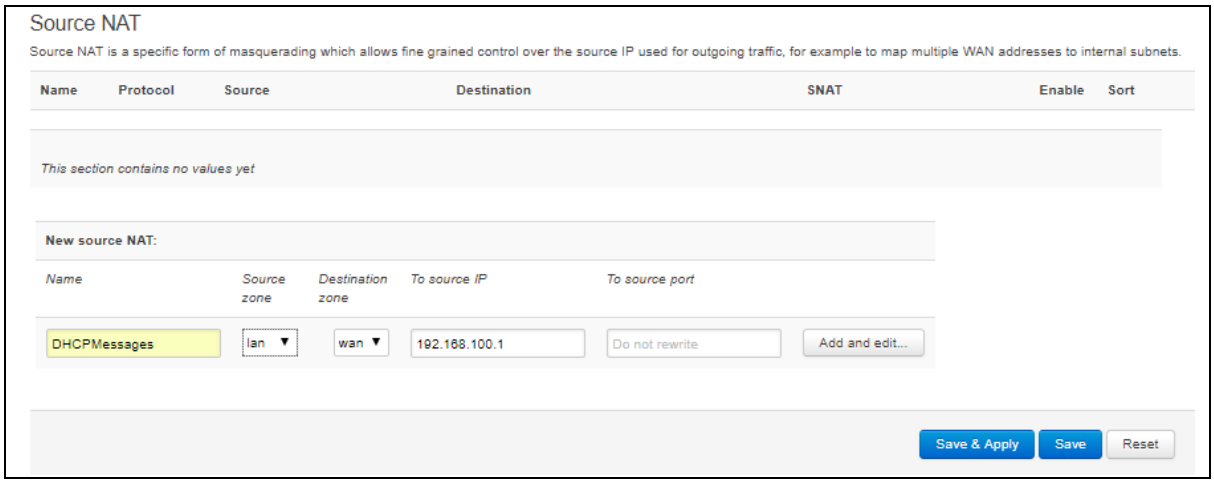



**Figure 124: The firewall – traffic rules configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Name<br>UCI: firewall.@redirect[X].name<br>Opt: name | Defines a name for the source NAT rule. |
| Web: Source Zone<br>UCI: firewall.@redirect[X].src<br>Opt: src | Defines the source interface for the source NAT rule.<br>Select **the interface where the DHCP requests are originating**. |
| Web: Destination Zone<br>UCI: firewall.@redirect[X].dest<br>Opt: dest | Defines destination interface for the source NAT rule.<br>Select **the interface where the DHCP requests are intended to be transmitted**. |
| Web: To source IP<br>UCI: firewall.@redirect[X].src_dip<br>Opt: src_dip | Defines the IP address to rewrite matched traffic souce IP.<br>Select **the source IP address to match the required IPSec rule**. |
| Web: To source port<br>UCI: firewall.@redirect[X].src_dport<br>Opt: src_dport | Defines the port number to rewrite matched traffic souce port number.<br>**Leave empty**. |

**Table 78: Information table for the souce NAT configuration**

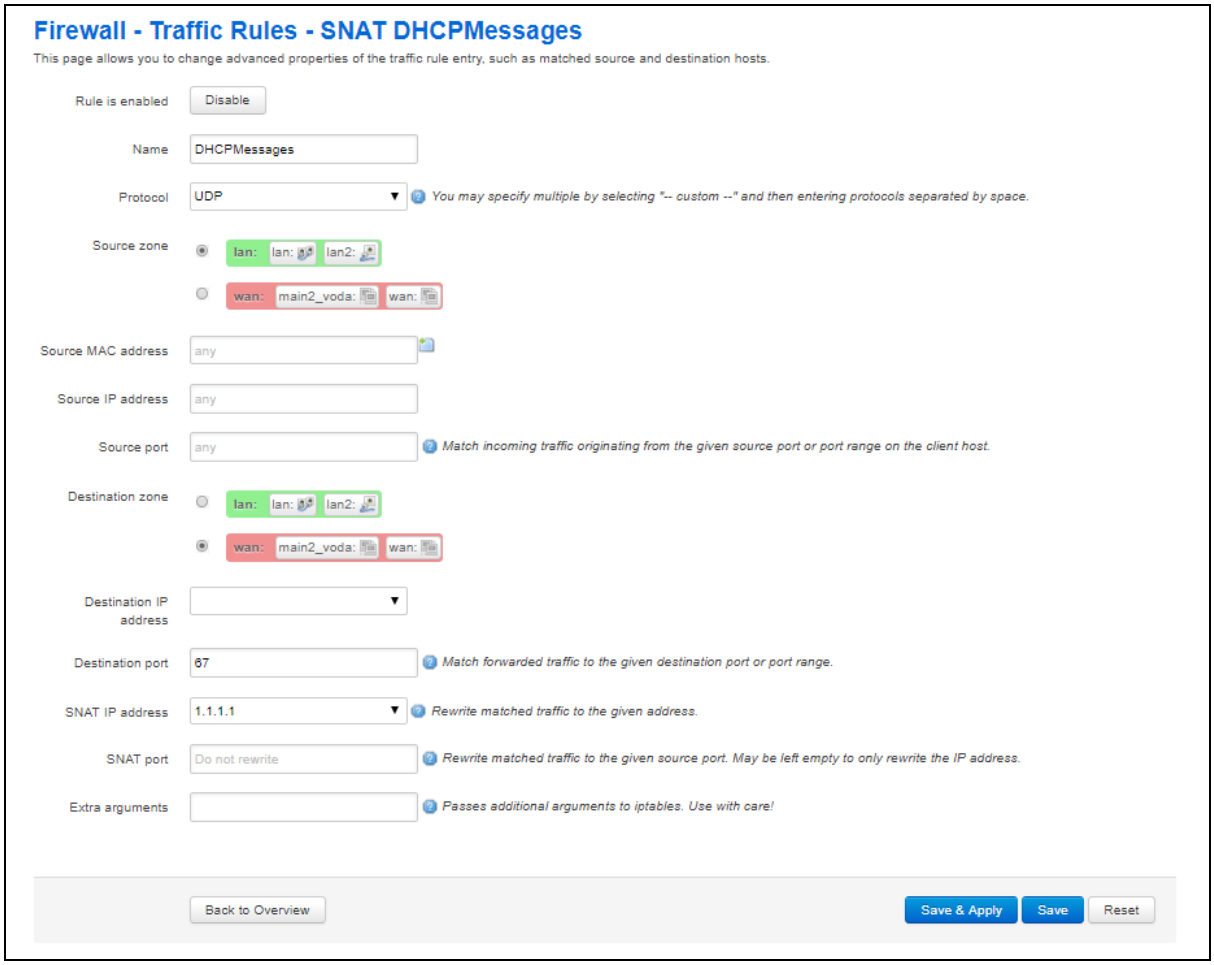



**Figure 125: The firewall – traffic rules – SNAT configuration page**

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled<br>UCI: firewall.@redirect[X].enabled<br>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<br>UCI: firewall.@redirect[X].name<br>Opt: name</td><td>Defines a name for the source NAT rule.</td></tr>
<tr><td>Web: Protocol<br>UCI: firewall.@redirect[X].proto<br>Opt: proto</td><td>Defines the protocol for the souce NAT rule to match.<br>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<br>UCI: firewall.@redirect[X].src<br>Opt: src</td><td>Defines the source interface for the source NAT rule.<br>Select the interface where the DHCP requests are originating.</td></tr>
<tr><td>Web: Destination Zone<br>UCI: firewall.@redirect[X].dest<br>Opt: dest</td><td>Defines destination interface for the source NAT rule.<br>Select the interface where the DHCP requests are intended to be transmitted.</td></tr>
<tr><td>Web: Destination port<br>UCI: firewall.@redirect[X].port<br>Opt: port</td><td>Defines the destination port number to match.<br>Select 67.</td></tr>
<tr><td>Web: SNAT IP address<br>UCI: firewall.@redirect[X].src_dip<br>Opt: src_dip</td><td>Defines the IP address to rewrite matched traffic.<br>Select the source IP address to match the required IPSec rule.</td></tr>
</table>

**Table 79: 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.

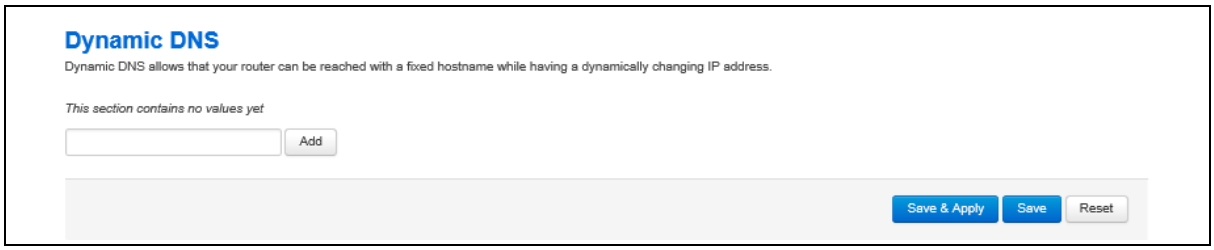



**Figure 126: The Dynamic DNS configuration page**

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

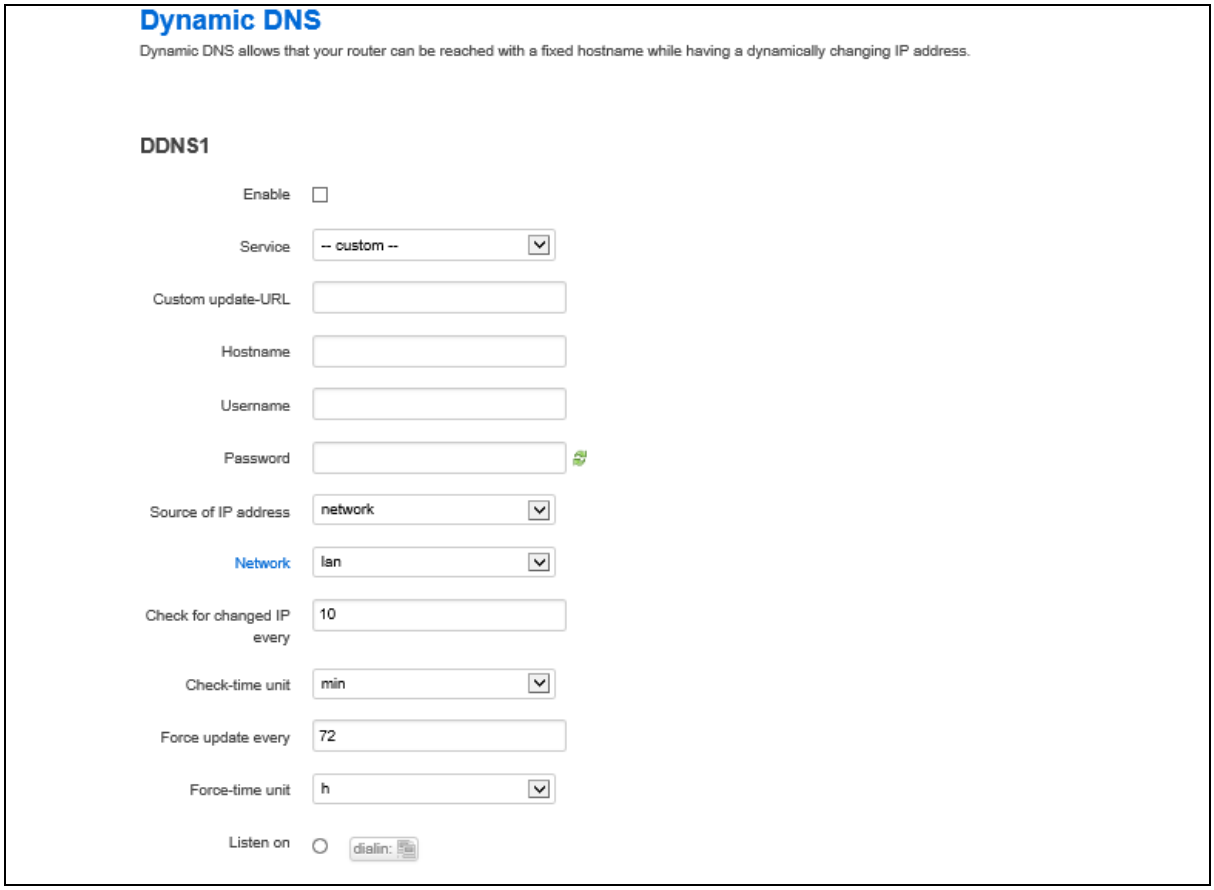



**Figure 127: The dynamic DNS main settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable<br>UCI: ddns.<name>.enabled<br>Opt: enabled | Enables a Dynamic DNS entry on the router.<br>0 Disabled.<br>1 Enabled |
| Web: Service<br>UCI: ddns.<name>.service_name<br>Opt: service_name | Defines the Dynamic DNS provider. |
| Web: Customer update-URL<br>UCI: ddns.<name>.update_url<br>Opt: update_url | Defines the customer DNS provider.<br>Displayed when the service is set to custom in the web interface. |
| Web: Hostname<br>UCI: ddns.<name>.domain<br>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<br>UCI: ddns.<name>.username<br>Opt: username | Defines the user name to use for authenticating domain updates with the selected provider. |
| Web: Password<br>UCI: ddns.<name>.password<br>Opt: password | Defines the password to use for authenticating domain name updates with the selected provider. |
| Web: Source of IP address<br>UCI: ddns.<name>.ip_source<br>Opt: ip_source | Defines the type of interface whose IP needs to be updated.<br>network IP is a associated with a network configuration.<br>interface IP is associated with an interface.<br>web IP is associated with a URL. |

| | |
|---|---|
| Web: Network<br>UCI: ddns.<name>.ip_network<br>Opt: ip_network | Defines the network whose IP needs to be updated.<br>Displayed when the Source of IP address option is set to network.<br>All the configured network interfaces will be shown. |
| Web: Inteface<br>UCI: ddns.<name>.ip_interface<br>Opt: ip_interface | Defines the interface whose IP needs to be updated.<br>Displayed when the Source of IP address option is set to interface.<br>All the configured interfaces will be shown. |
| Web: URL<br>UCI: ddns.<name>.ip_url<br>Opt: ip_url | Defines the URL where the IP downloaded from.<br>Displayed when the Source of IP address option is set to URL. |
| Web: Check for changed IP every<br>UCI: ddns.<name>.check_interval<br>Opt: check_interval | Defines how often to check for an IP change. Used in conjunction with check_unit.<br>10<br>Range |
| Web: Check-time unit<br>UCI: ddns.<name>.check_unit<br>Opt: check_unit | Defines the time unit to use for check for an IP change. Used in conjunction with check_interval.<br>Minutes<br>hours |
| Web: Force update every<br>UCI: ddns.<name>.force_interval<br>Opt: force_interval | Defines how often to force an IP update to the provider. Used in conjunction with force_unit.<br>72 Disabled.<br>Range Enabled |
| Web: Force-time unit<br>UCI: ddns.<name>.force_unit<br>Opt: force_unit | Defines the time unit to use for check for an IP change. Used in conjunction with force_interval.<br>Minutes<br>Hours |
| Web: Listen on<br>UCI: ddns.<name>.interface<br>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.<br>All configured interfaces will be displayed. |

**Table 80: 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 128: The host records add page**

Select **Add**. Enter a hostname and IP address and select **Save & Apply**.

**Figure 129: The host records configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Hostname<br>UCI: network.host.hostname<br>Opt: hostname | Defines the hostname. |
| Web: IP-Address<br>UCI: network.host.addr<br>Opt: addr | Defines the IP address associated with the hostname. |

**Table 81: 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 130: The hostnames add page**

Select **Add**. Enter a hostname and IP address for the PTR record and select **Save & Apply**.

**Figure 131: The hostnames configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Hostname<br>UCI: dhcp.domain.name<br>Opt: name | Defines the domain name for the PTR record. |
| Web: IP-Address<br>UCI: dhcp.domain.ip<br>Opt: ip | Defines the IP address associated with the domain name. |

**Table 82: 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.

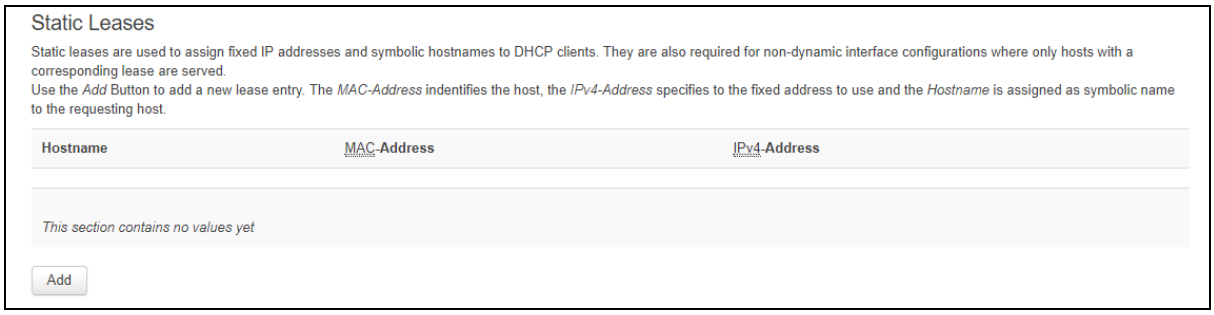



**Figure 132: The static leases add page**

Select **Add**. Enter a hostname, MAC address and IP address for the static lease. Select **Save & Apply**.

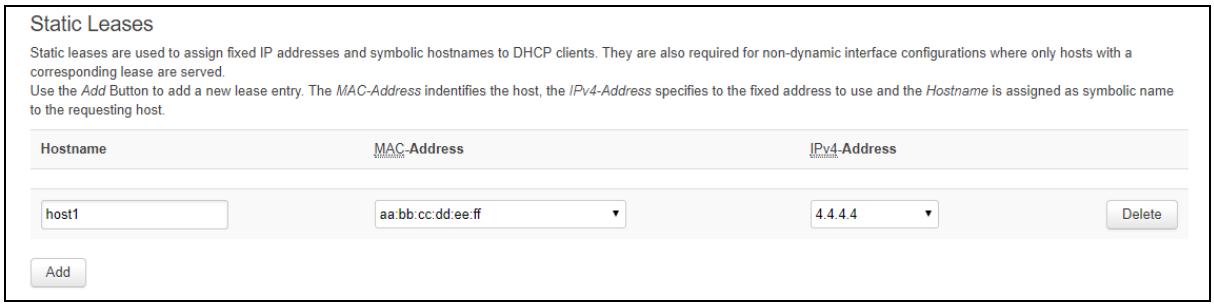



**Figure 133: The static leases configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Hostname<br>UCI: dhcp.host.name<br>Opt: name | Defines the symbolic hostname to assign. |
| Web: MAC-Address<br>UCI: dhcp.host.mac<br>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<br>UCI: dhcp.host.ip<br>Opt: ip | Defines the IP address to be used for this host. |

**Table 83: 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.

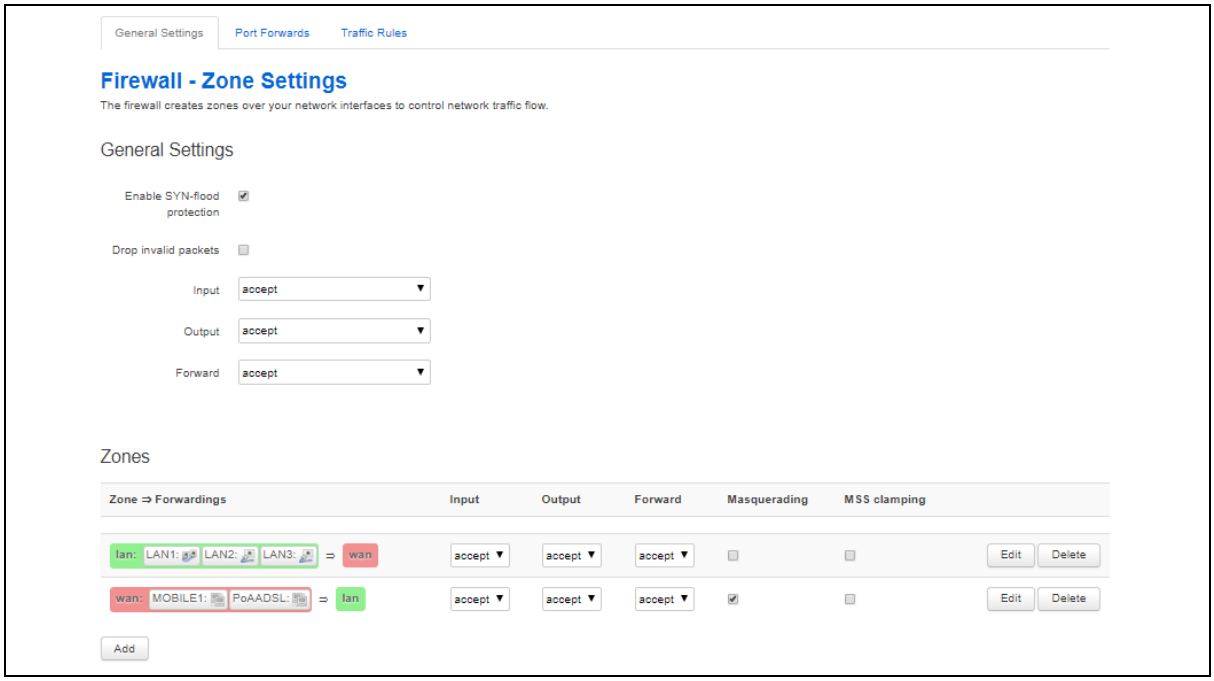



**Figure 134: The firewall zone general settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable SYN-flood protection<br>UCI: firewall.defaults.syn_flood<br>Opt: syn_flood | Enables SYN flood protection.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Drop invalid packets<br>UCI: firewall.defaults.drop_invalid<br>Opt: drop_invalid | Drops packets not matching any active connection.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Input<br>UCI: firewall.defaults.input<br>Opt: input | Default policy for the Input chain.<br>Accept – Accepted packets pass through the firewall.<br>Reject – Rejected packets are blocked by the firewall and ICMP message is returned to the source host.<br>Drop – Dropped packets are blocked by the firewall. |
| Web: Output<br>UCI: firewall.defaults.output<br>Opt: output | Default policy for the Output chain.<br>Accept – Accepted packets pass through the firewall.<br>Reject – Rejected packets are blocked by the firewall and ICMP message is returned to the source host.<br>Drop – Dropped packets are blocked by the firewall. |
| Web: Forward<br>UCI: firewall.defaults.forward<br>Opt: forward | Default policy for the Forward chain.<br>Accept – Accepted packets pass through the firewall.<br>Reject – Rejected packets are blocked by the firewall and ICMP message is returned to the source host.<br>Drop – Dropped packets are blocked by the firewall. |

**Table 84: 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

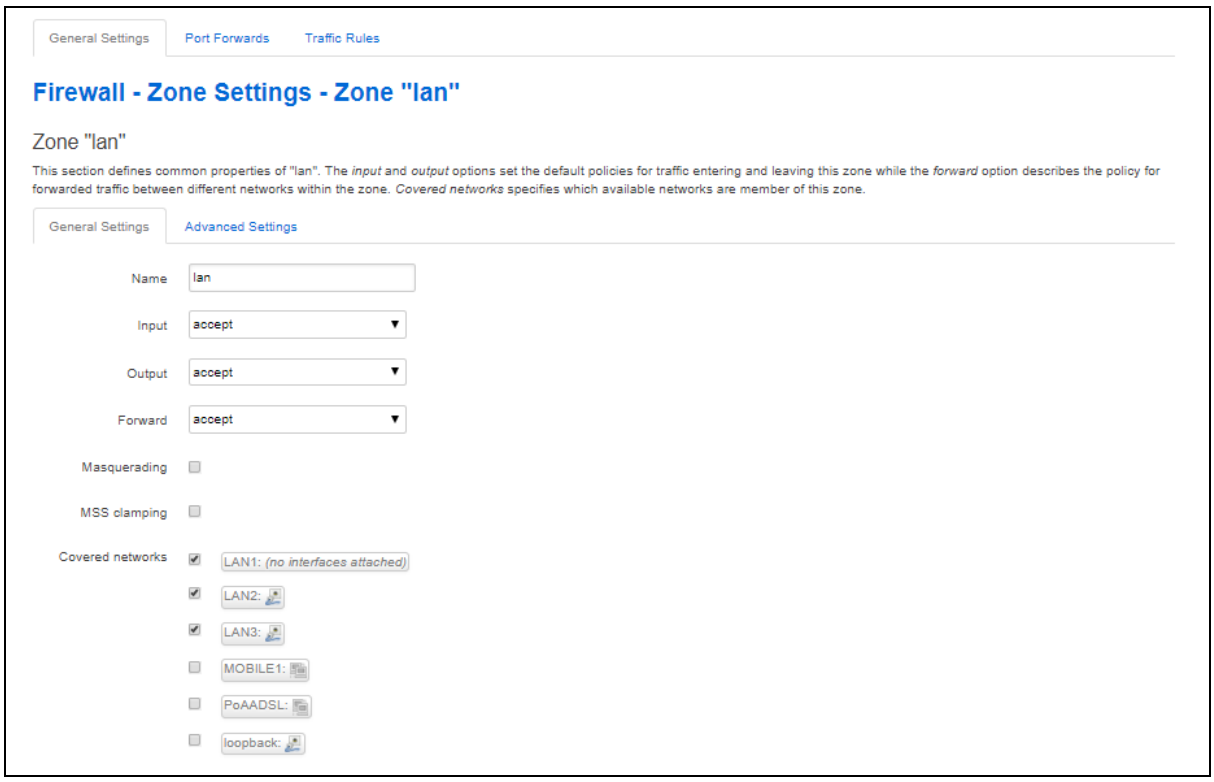



**Figure 135: The firewall zone general settings**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: name<br>UCI: firewall.<zone label>.name<br>Opt: name | 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. |
| Web: Input<br>UCI: firewall.<zone label>.input<br>Opt: input | 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.<br>Accept — Accepted packets pass through the firewall.<br>Reject — Rejected packets are blocked by the firewall and ICMP message is returned to the source host.<br>Drop — Dropped packets are blocked by the firewall. |

<table>
<tr><td>Web: Output<br>UCI: firewall.<zone label>.output<br>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<br>UCI: firewall.<zone label>.forward<br>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<br>UCI: firewall.<zone label>.masq<br>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<br>UCI: firewall.<zone label>.mtu_fix<br>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<br>UCI: firewall.<zone label>.network<br>Opt: network</td><td>Defines a list of interfaces attached to this zone, if omitted, the value of name is used by default.<br>Note: use the uci list syntax to edit this setting through UCI.</td></tr>
</table>

**Table 85: Information table for firewall zone general settings**

### 28.2.1.4 Firewall zone: advanced settings

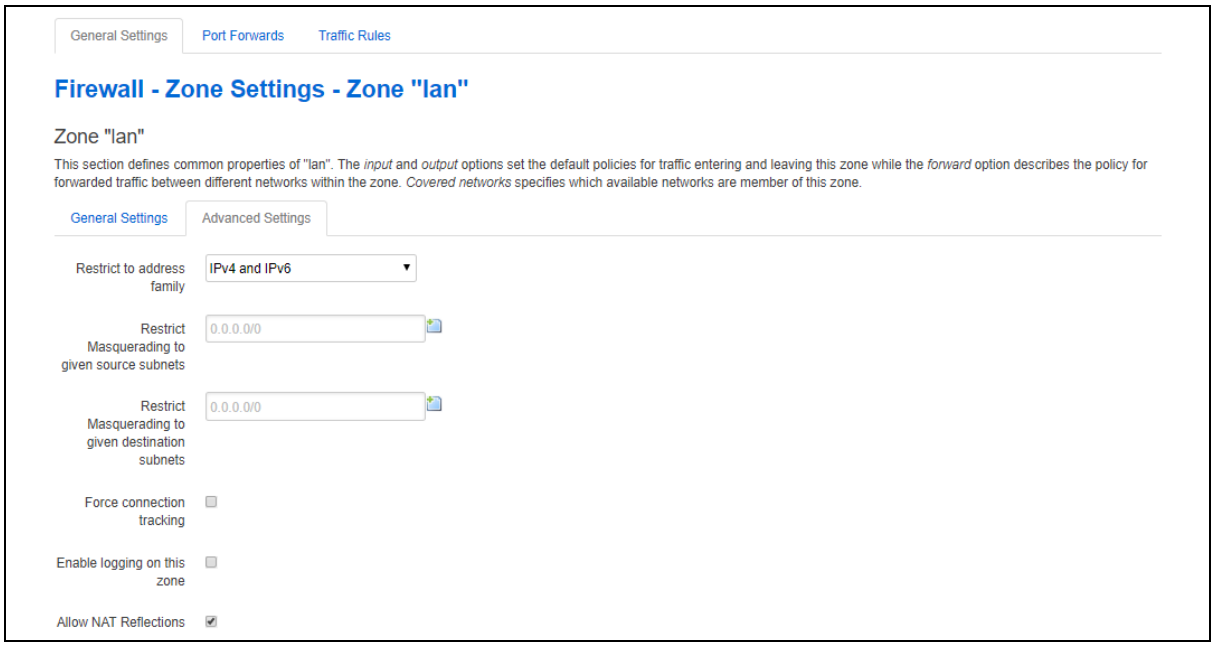



**Figure 136: Firewall zone advanced settings**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Restrict to address family<br>UCI: firewall.<zone label>.family<br>Opt: family | Restricts zone to IPv4, IPv6 or both IPv4 and IPv6.<br>**Option** / **Description** / **UCI**<br>IPv4 and IPv6 / Any address family / any<br>IPv4 only / IPv4 only / ipv4<br>IPv6 only / IPv6 only / Ipv6 |
| Web: Restrict Masquerading to given source subnets.<br>UCI: firewall.<zone label>.masq_src<br>Opt: masq_src | Limits masquerading to the given source subnets. Negation is possible by prefixing the subnet with '!'. Multiple subnets are allowed. |
| Web: Restrict Masquerading to given destination subnets.<br>UCI: firewall.<zone label>.masq_dest<br>Opt: masq_dest | 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'. |
| Web: Force connection tracking<br>UCI: firewall.<zone label>.conntrack<br>Opt: conntrack | Forces connection tracking for this zone.<br>0 / Disabled.<br>1 / If masquerading is used. Otherwise, default is 0. |
| Web: Enable logging on this zone<br>UCI: firewall.<zone label>.log<br>Opt: log | Creates log rules for rejected and dropped traffic in this zone. |
| Web: Allow NAT reflections<br>UCI: firewall.<zone label>.reflection<br>Opt: reflection | Enable/disable all NAT reflections for this zone.<br>**Note**: For configs with a large number of firewall rules, disabling NAT reflection will speed up load of firewall rules on interface start.<br>0 / Disable reflection.<br>1 / Enable reflection. |

| | |
|---|---|
| Web: n/a<br>UCI: firewall.<zone label>.log_limit<br>Opt: log_limit | Limits the amount of log messages per interval. |

**Table 86: 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.

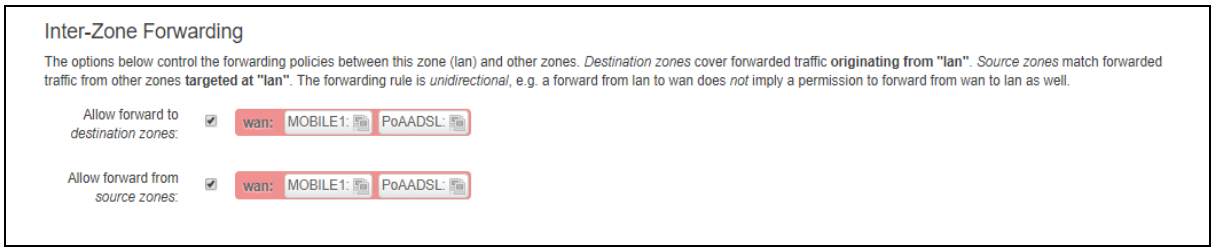



**Figure 137: The inter-zone forwarding section**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Allow forward to destination zones<br>UCI: firewall.<forwarding label>.dest<br>Opt: dest | Allows forward to other zones. Enter the current zone as the source.<br>Enabling this option puts two entries into the firewall file: destination and source. |
| UCI firewall.<forwarding label>.src<br>Opt: src | |
| Web: Allow forward from source zones<br>UCI: firewall.<forwarding label>.dest<br>Opt: dest | Allows forward from other zones. Enter the current zone as the destination.<br>Enabling this option puts two entries into the firewall file: destination and source. |
| UCI: firewall.<forwarding label>.src<br>Opt: src | |

**Table 87: 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**.

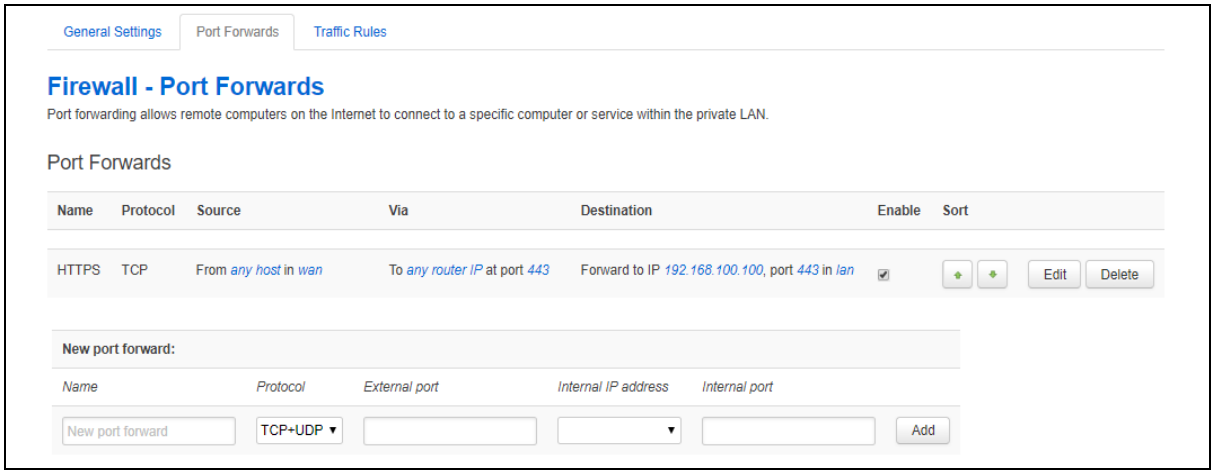



**Figure 138: The firewall port forward page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: name<br>UCI: firewall.<redirect label>.name<br>Opt: name | 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. |
| Web: Protocol<br>UCI: firewall.<redirect label>.proto<br>Opt: proto | Defines layer 4 protocol to match incoming traffic.<br>**Option** / **Description** / **UCI**<br>tcp+udp / Match either TCP or UDP packets. / tcp udp<br>tcp / Match TCP packets only. / tcp<br>udp / Match UDP packets only. / udp |
| Web: External port<br>UCI: firewall.<redirect label>.src_dport<br>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 as start:stop, for example, 2001:2020.<br>Blank / Match traffic to any port.<br>Range / 1 - 65535 |
| Web: Internal IP address<br>UCI: firewall.<redirect label>.dest_ip<br>Opt: dest_ip | Specifies the internal (LAN) IP address for the traffic to be redirected to. |
| Web: Internal port<br>UCI: firewall.<redirect label>.dest_port<br>Opt: dest_port | Specifies the destination tcp/udp port for the redirect traffic. |

**Table 88: 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).

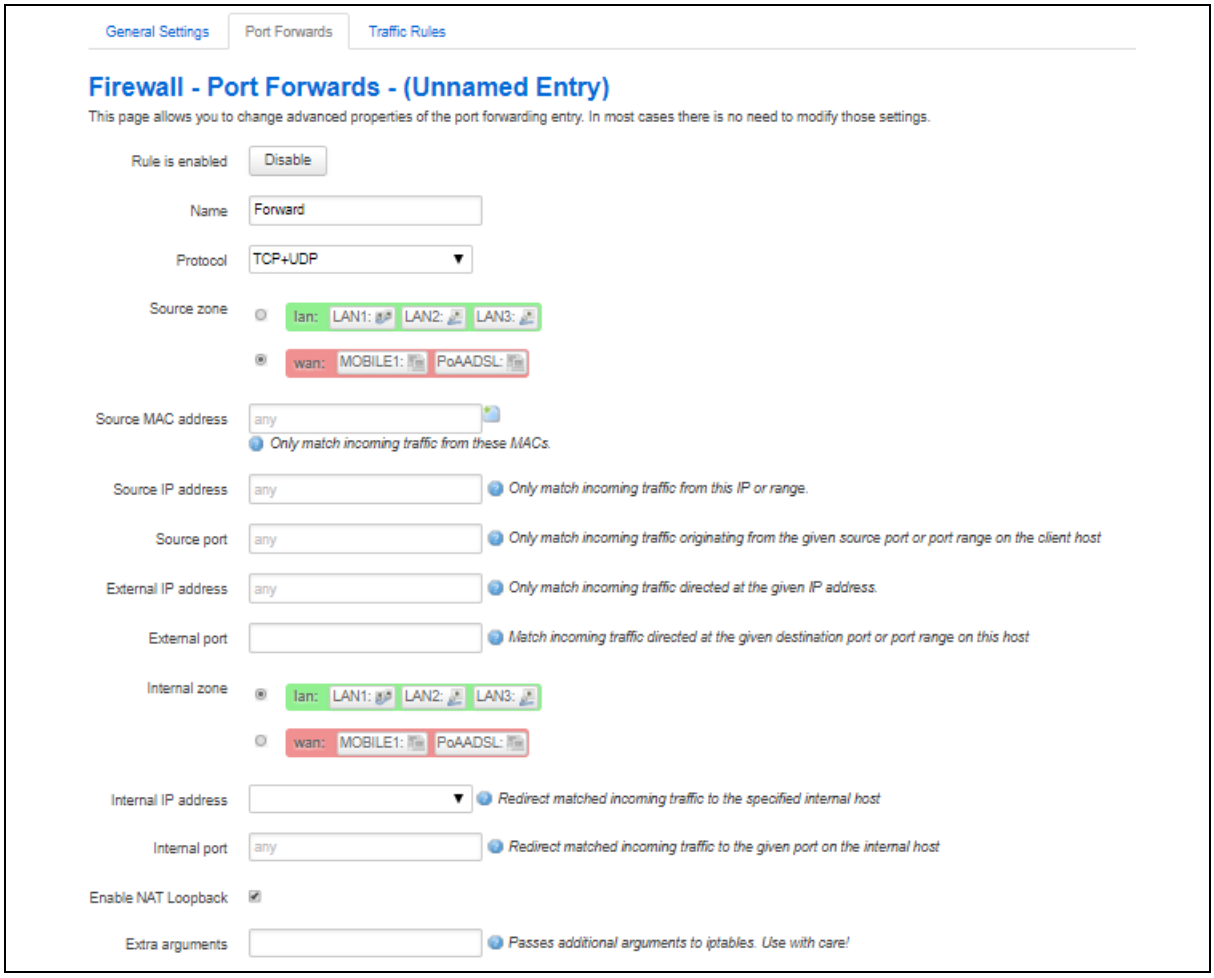



**Figure 139: The firewall port forwards edits page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Rule is enabled<br>UCI: firewall.<redirect label>.enabled<br>Opt: enabled | Specifies if this redirect should be enabled or disabled.<br>0 — Disabled.<br>1 — Enabled. |
| Web: name<br>UCI: firewall.<redirect label>.name<br>Opt: name | 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. |
| Web: Protocol<br>UCI: firewall.<redirect label>.proto<br>Opt: proto | Defines layer 4 protocol to match incoming traffic.<br>**Option** — **Description** — **UCI**<br>tcp+udp — Match either TCP or UDP packets. — tcp udp<br>tcp — Match TCP packets only. — tcp<br>udp — Match UDP packets only. — udp |
| Web: Source zone<br>UCI: firewall.<redirect label>.src<br>Opt: src | 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. |

| | |
|---|---|
| Web: Source MAC address<br>UCI: firewall.<redirect label>.src_mac<br>Opt: list src_mac | Defines the list of source MAC addresses that this redirect will match.<br>Format: aa:bb:cc:dd:ee:ff<br>Multiple RIP interfaces are entered using `uci set` and `uci add_list` commands. Example:<br>`uci set firewall.@redirect[0].src_mac=aa:bb:cc:dd:ee:ff`<br>`uci add_list firewall.@redirect[0].src_mac=12:34:56:78:90:12`<br>or using a list of options via package options<br>`list network 'aa:bb:cc:dd:ee:ff'`<br>`list network '12:34:56:78:90:12'` |
| Web: Source IP address<br>UCI: firewall.<redirect label>.src_ip<br>Opt: src_ip | Defines a source IP address that this redirect will match.<br>Blank: Match traffic from any source IP.<br>Range: A.B.C.D/mask. |
| Web: Source port<br>UCI: firewall.<redirect label>.src_port<br>Opt: src_port | Defines a source IP port that this redirect will match. You can enter multiple ports, using a space separator.<br>For example: option src_port '22 23'<br>*see note below on use with options src_dport and dest_port<br>Blank: Match traffic from any source port.<br>Range: 1 - 65535 |
| Web: External port<br>UCI: firewall.<redirect label>.src_dport<br>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.<br>You can enter multiple ports, using a space separator.<br>For example: option src_dport '22 23'<br>*see note below on use with options src_port and dest_port<br>Blank: Match traffic to any port.<br>Range: 1 – 65535 |
| Web: Internal zone<br>UCI: firewall.<redirect label>.dest<br>Opt: dest | Specifies the traffic destination zone, must refer to one of the defined zone names. |
| Web: Internal IP address<br>UCI: firewall.<redirect label>.dest_ip<br>Opt: dest_ip | Specifies the internal (LAN) IP address for the traffic to be redirected to. |
| Web: Internal port<br>UCI: firewall.<redirect label>.dest_port<br>Opt: dest_port | Specifies the destination tcp/udp port for the redirect traffic. You can enter multiple ports, using a space separator.<br>*For example: option dest_port '22 23'<br>*See note below table on use with options src_port and src_dport. |
| Web: Enable NAT Loopback<br>UCI: firewall.<redirect label>.reflection<br>Opt: reflection | Enable or disable NAT reflection for this redirect.<br>0: Reflection disabled.<br>1: Reflection enabled. |
| Web: Extra arguments<br>UCI: firewall.<redirect label>.extra<br>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 89: 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.

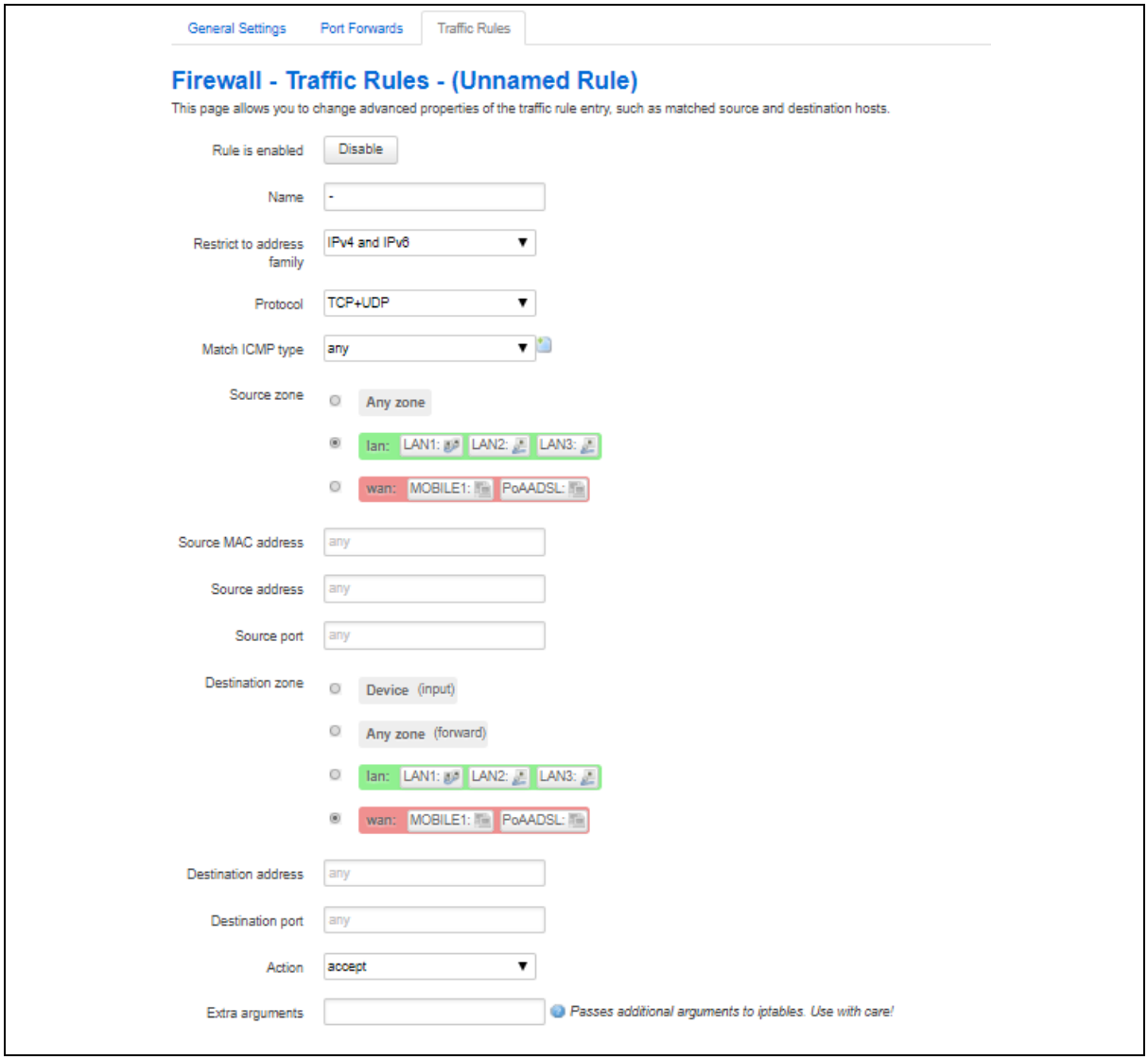



**Figure 140: The firewall traffic rules page**

<table>
<tr><th>Web Field / UCI / Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled<br>UCI: firewall.<rule label>.enabled<br>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<br>UCI: firewall.<rule label>.name<br>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<br>UCI: firewall.<rule label>.family<br>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<br>UCI: firewall.<rule label>.proto<br>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<br>UCI: firewall.<rule label>.icmp_type<br>Opt: icmp_type</td><td>Match specific icmp types.<br>This option is only valid when ICMP is selected as the protocol. ICMP types can be listed as either type names or type numbers.<br>Note: for a full list of valid ICMP type names, see the ICMP Options table below.</td></tr>
<tr><td>Web: Source zone<br>UCI: firewall.<rule label>.src<br>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<br>UCI: firewall.<rule label>.src_mac<br>Opt: src_mac</td><td>Matches incoming traffic from the specified MAC address.<br>The MAC address must be entered in the following format:<br><code>aa:bb:cc:dd:ee:ff:</code><br>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.<br>Example:<br><code>option src_mac 00:E0:C8:12:34:56/24</code><br>will match on all packets with prefix 00:E0:C8.</td></tr>
<tr><td>Web: Source address<br>UCI: firewall.<rule label>.src_ip<br>Opt: src_ip</td><td>Matches incoming traffic from the specified source IP address.</td></tr>
<tr><td>Web: Source port<br>UCI: firewall.<rule label>.src_port<br>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<br>UCI: firewall.<rule label>.dest<br>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<br>UCI: firewall.<rule label>.dest_ip<br>Opt: dest_ip</td><td>For DNAT, redirects matched incoming traffic to the specified internal host.<br>For SNAT, matches traffic directed at the given address.</td></tr>
</table>

<table>
<tr><td>Web: Destination port<br>UCI: firewall.<rule label>.dest_port<br>Opt: dest_port</td><td>For DNAT, redirects matched incoming traffic to the given port on the internal host.<br>For SNAT, matches traffic directed at the given ports.</td></tr>
<tr><td>Web: Action<br>UCI: firewall.<rule label>.target<br>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<br>UCI: firewall.<rule label>.extra<br>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<br>UCI: firewall.<rule label>.reflection<br>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<br>UCI: firewall.<rule label>.limit<br>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<br>UCI: firewall.<rule label>.limit_burst<br>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<br>UCI: firewall.<rule label>.recent<br>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 90: 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 91: 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<br>connection<br>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

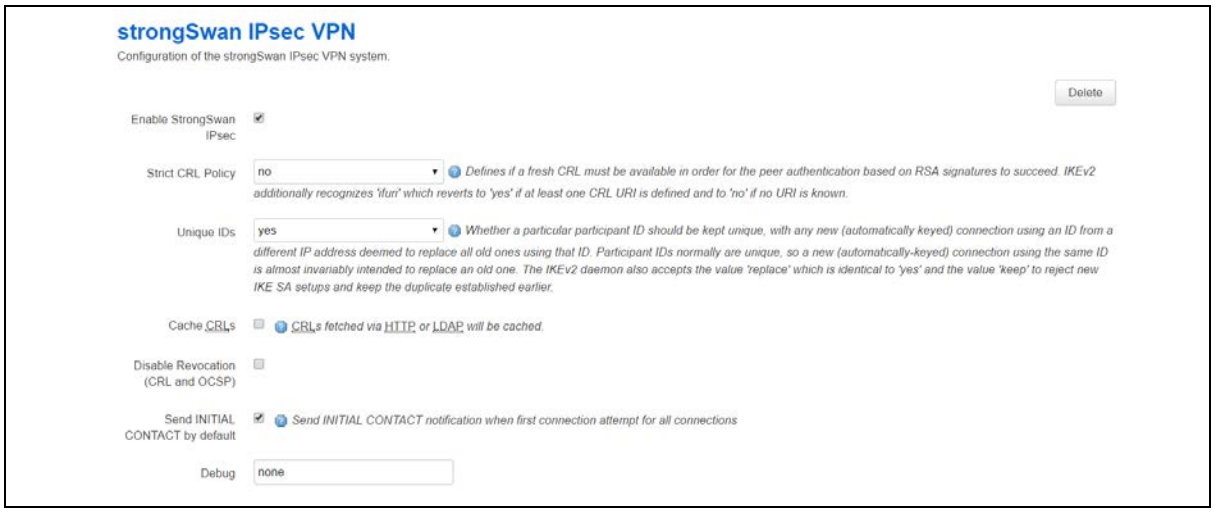



**Figure 141: The common settings section**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Enable strongswan<br>UCI: strongswan.general.enable<br>Opt: enabled | Enables or disables IPSec.<br>0 Disabled.<br>1 Enabled. |
| Web: Strict CRL Policy<br>UCI: strongswan.general.strictcrlpolicy<br>Opt: strictcrlpolicy | Defines if a fresh CRL must be available for the peer authentication based on RSA signatures to succeed.<br>0 Disabled.<br>1 Enabled.<br>ifuri 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. |
| Web: Unique IDs<br>UCI: strongswan.general.uniqueids<br>Opt: uniqueids | 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.<br>Participant IDs normally are unique, so a new (automatically-keyed) connection using the same ID is almost invariably intended to replace an old one.<br>0 Disabled.<br>1 Enabled.<br>replace Identical to Yes.<br>keep Rejects new IKE SA and keep the duplicate established earlier |
| Web: Cache CRLs<br>UCI: strongswan.general.cachecrls<br>Opt: cachecrls | 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.<br>0 Disabled.<br>1 Enabled. |
| Web: Disable Revocation<br>UCI: strongswan.general.revocation_disabled<br>Opt: revocation_disabled | Defines whether disable CRL and OCSP checking for revoked certificates.<br>0 Disabled.<br>1 Enabled. |
| Web: Send INITIAL CONTACT by default<br>UCI: strongswan.general.initial_contact<br>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. This can also be enabled/disabled per connection.<br>0 Does not set initial contact flag.<br>1 Sets initial contact flag on first attempt. |
| Web: Debug<br>UCI: strongswan.general.debug<br>Opt: debug | Enables debugging. This option is used for trouble shooting issues. It is not suitable for a production environment.<br>None Debug disabled.<br>Control Debug enabled. Shows generic control flow with errors and very basic auditing logs.<br>All Debug enabled. Most verbose logging also includes sensitive information such as keys. |

**Table 92: Information table for IPSec common settings**

### 29.2.2 Common settings: configure connection

**Figure 142: The configuring IPSec settings**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: strongswan.@connection[X].enabled<br>Opt: enable | Enables or disables IPSec connection.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Aggressive<br>UCI: strongswan.@connection[X].aggressive<br>Opt: aggressive | Enables or disables IKE aggressive mode.<br>**Note**: using aggressive mode along with PSK authentication is less secure method than main mode and should be avoided.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Name<br>UCI: strongswan.@connection[X].name<br>Opt: name | Specifies a name for the tunnel. |
| Web: Autostart Action<br>UCI: strongswan.@connection[X].auto<br>Opt: auto | Specifies when the tunnel is initiated.<br>start – On start up.<br>route – When traffic routes this way.<br>add – Loads a connection without starting it.<br>ignore – Ignores the connection.<br>always – Actively retries to establish the tunnel if it went down. |
| Web: Connection Type<br>UCI: strongswan.@connection[X].type<br>Opt: type | Defines the type of IPSec connection.<br>tunnel – Connection uses tunnel mode.<br>transport – Connection uses transport mode.<br>pass – Connection does not perform any IPSec processing.<br>drop – Connection drops all the packets. |

**Table 93: Information table for connection settings**

### 29.2.3 Common settings: IP addressing

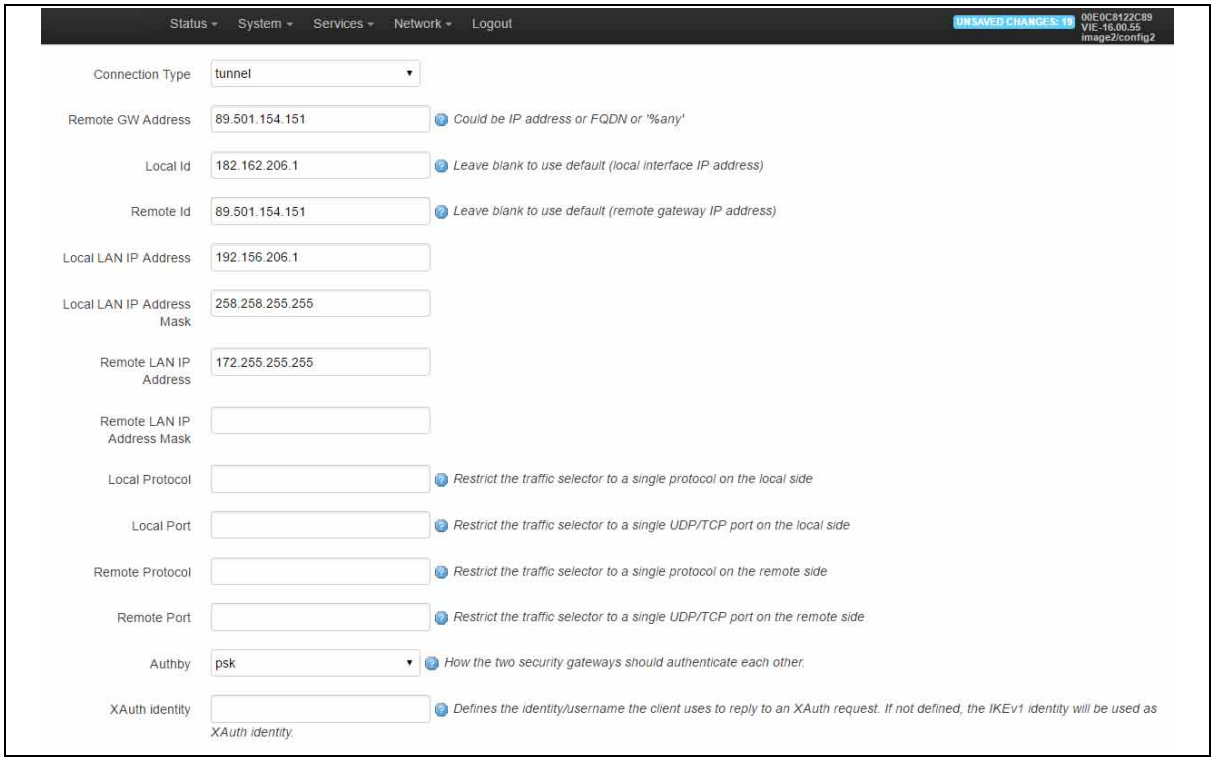



**Figure 143: The IP addressing settings**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Remote GW Address<br>UCI: strongswan.@connection[X]. remoteaddress<br>Opt: remoteaddress | Sets the public IP address of the remote peer. |
| Web: Local ID<br>UCI: strongswan.@connection[X].localid<br>Opt: localid | Defines the local peer identifier. |
| Web: Remote ID<br>UCI: strongswan.@connection[X].remoteid<br>Opt:remoteid | Defines the remote peer identifier. |
| Web: Local LAN IP Address<br>UCI: strongswan.@connection[X]. locallan<br>Opt: locallan | Defines the local IP of LAN. |
| Web: Local LAN IP Address Mask<br>UCI: strongswan.@connection[X]. locallanmask<br>Opt: locallanmask | Defines the subnet of local LAN. |
| Web: Remote LAN IP Address<br>UCI: strongswan.@connection[X]. remotelan<br>Opt:remotelan | Defines the IP address of LAN serviced by remote peer. |
| Web: Remote LAN IP Address Mask<br>UCI: strongswan.@connection[X]. remotelanmask<br>Opt:remotelanmask | Defines the Subnet of remote LAN. |

| | |
|---|---|
| Web: Local Protocol<br>UCI: strongswan.@connection[X].localproto<br>Opt: localproto | Restricts the connection to a single protocol on the local side. |
| Web: Local Port<br>UCI: strongswan.@connection[X].localport<br>Opt: localport | Restricts the connection to a single port on the local side. |
| Web: Remote Protocol<br>UCI: strongswan.@connection[X].remoteproto<br>Opt:remoteproto | Restricts the connection to a single protocol on the remote side. |
| Web: Remote Port<br>UCI: strongswan.@connection[X].remoteport<br>Opt: remoteport | Restricts the connection to a single port on the remote side. |
| Web: Authby<br>UCI: strongswan.@connection[X].authby<br>Opt: authby | Defines how the two secure gateways should authenticate.<br>**Note**: using aggressive mode along with PSK authentication is unsecure and should be avoided.<br>Pubkey: For public key signatures.<br>Rsasig: For RSA digital signatures.<br>ecdsasig: For Elliptic Curve DSA signatures.<br>Psk: Using a preshared key.<br>xauthrsasig: Enables eXtended Authentication (XAuth) with addition to RSA signatures.<br>xauthpsk: Using extended authentication and preshared key.<br>never: Can be used if negotiation is never to be attempted or accepted (shunt connections). |

**Table 94: Information table for IP addressing settings**

### 29.2.4 Common settings: IPSec settings

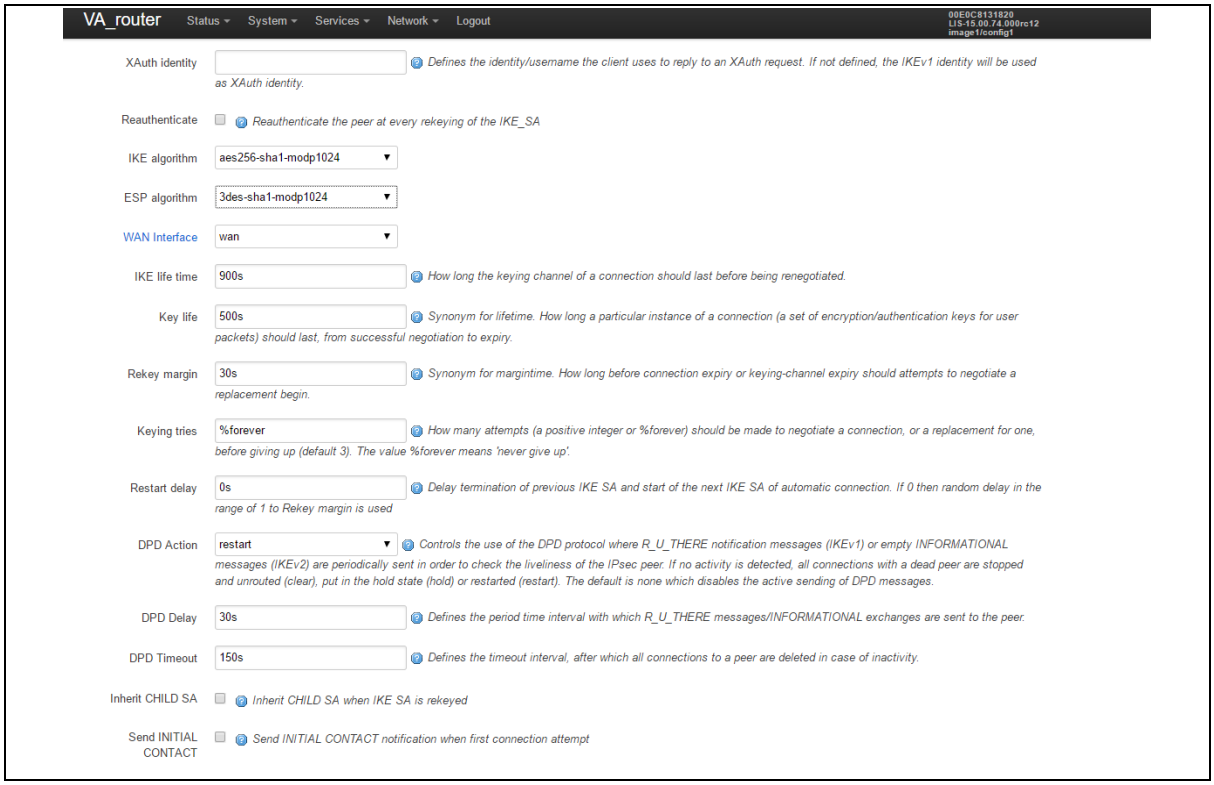



**Figure 144: The IPSec connections settings**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: XAuth Identity<br>UCI: strongswan.@connection[X].xauth_identity<br>Opt: xauth_identity | Defines Xauth ID. |
| Web: IKE Algorithm<br>UCI: strongswan.@connection[X].ike<br>Opt: ike | Specifies the IKE algorithm to use.<br>The format is: encAlgo \| authAlgo \| DHGroup<br>encAlgo:<br>3des<br>aes128<br>aes256<br>serpent<br>twofish<br>blowfish<br>authAlgo:<br>md5<br>sha<br>sha2<br>DHGroup:<br>modp1024<br>modp1536<br>modp2048<br>modp3072<br>modp4096<br>modp6144<br>modp8192<br>For example, a valid IKE algorithm is aes128-sha-modp1536. |

| | |
|---|---|
| Web: ESP algorithm<br>UCI: strongswan.@connection[X].esp<br>Opt: esp | Specifies the esp algorithm to use.<br>The format is: encAlgo \| authAlgo \| DHGroup<br>encAlgo:<br>3des<br>aes128<br>aes256<br>serpent<br>twofish<br>blowfish<br>authAlgo:<br>md5<br>sha<br>sha2<br>DHGroup:<br>modp1024<br>modp1536<br>modp2048<br>modp3072<br>modp4096<br>modp6144<br>modp8192<br>For example, a valid encryption algorithm is: aes128-sha-modp1536.<br>If no DH group is defined then PFS is disabled. |
| Web: WAN Interface<br>UCI: strongswan.@connection[X].waniface<br>Opt: waniface | This is a space-separated list of the WAN interfaces the router will use to establish a tunnel with the secure gateway.<br>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.<br>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<br>UCI: strongswan.@connection[X].ikelifetime<br>Opt:ikelifetime | Specifies how long the keyring channel of a connection (ISAKMP or IKE SA) should last before being renegotiated.<br>3h<br>Timespec: 1d, 3h, 25m, 10s. |
| Web: Key Life<br>UCI: strongswan.@connection[X].keylife<br>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.<br>Normally, the connection is renegotiated (via the keying channel) before it expires (see rekeymargin).<br>1h<br>Timespec: 1d, 1h, 25m, 10s. |
| Web: Rekey Margin<br>UCI: strongswan.@connection[X].rekeymargin<br>Opt: rekeymargin | Specifies how long before connection expiry or keying-channel expiry should attempt to negotiate a replacement begin.<br>Relevant only locally, other end need not agree on it.<br>9m<br>Timespec: 1d, 2h, 9m, 10s. |

| | |
|---|---|
| Web: Restart Delay<br>UCI: strongswan.@connection[X].restartdelay<br>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).<br>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.<br>Relevant only locally, other end need not agree on it.<br>0<br>Timespec: 1d, 2h, 9m, 10s. |
| Web: Keying Tries<br>UCI: strongswan.@connection[X].keyringtries<br>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<br>UCI: strongswan.@connection[X].dpdaction<br>Opt: dpdaction | Defines DPD (Dead Peer Detection) action.<br>None: Disables DPD.<br>Clear: Clear down the tunnel if peer does not respond. Reconnect when traffic brings the tunnel up.<br>Hold: Clear down the tunnel and bring up as soon as the peer is available.<br>Restart: Restarts DPD when no activity is detected. |
| Web: DPD Delay<br>UCI: strongswan.@connection[X].dpddelay<br>Opt: dpddelay | Defines the period time interval with which R_U_THERE messages and INFORMATIONAL exchanges are sent to the peer.<br>These are only sent if no other traffic is received.<br>30s<br>Timespec: 1d, 2h, 25m, 10s. |
| Web: DPD Timeout<br>UCI: strongswan.@connection[X].dpdtimeout<br>Opt: dpdtimeout | Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity.<br>150s<br>Timespec: 1d, 2h, 25m, 10s. |
| Web: Inherit CHILD SA<br>UCI: strongswan.@connection[X].inherit_child<br>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.<br>0: Delete the existing IPSEC SA on IKE rekey<br>1: Maintain the existing IPSEC SA on IKE rekey |
| Web: Send INITIAL CONTACT<br>UCI: strongswan.@connection[X].initial_contact<br>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.<br>0: Do not set initial contact flag<br>1: Set initial contact flag on first attempt |

**Table 95: 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.

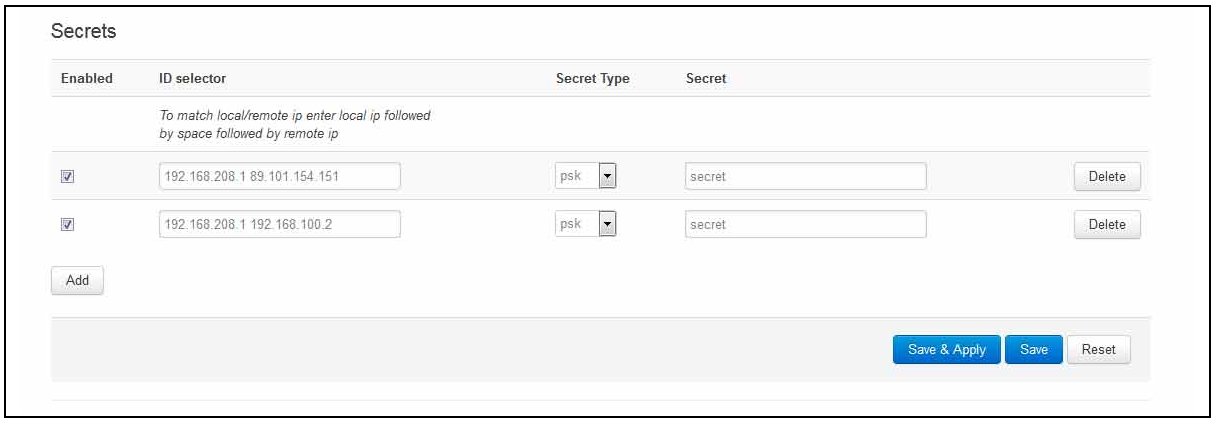



**Figure 145: IPSec secrets settings**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: strongswan.@secret[X].enabled<br>Opt: enabled | Defines whether this set of credentials is to be used or not.<br>0 Disabled.<br>1 Enabled. |
| Web: ID selector<br>UCI: strongswan.@secret[X].idtype<br>Opt: idtype | Defines whether IP address or userfqdn is used. |
| Web: ID selector<br>UCI: strongswan.@secret[X].localaddress<br>Opt: localaddress | Defines the local address this secret applies to. |
| Web: ID selector<br>UCI: strongswan.@secret[X]. remoteaddress<br>Opt: remoteaddress | Defines the remote address this secret applies to. |
| Web: N/A<br>UCI: strongswan.@secret[X].userfqnd<br>Opt: userfqnd | FQDN or Xauth name used of Extended Authentication. This must match xauth_identity from the configuration connection section. |
| Web: Secret Type<br>UCI: strongswan.@secret[X].secrettype<br>Opt: secrettype | Specifies the authentication mechanism to be used by the two peers.<br>Psk Preshared secret<br>Pubkey Public key signatures<br>Rsasig RSA digital signatures<br>Ecdsasig Elliptic Curve DSA signatures<br>Xauth Extended authentication |
| Web: Secret<br>UCI: strongswan.@secret[X].secret<br>Opt: secret | Defines the secret. |

**Table 96: 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

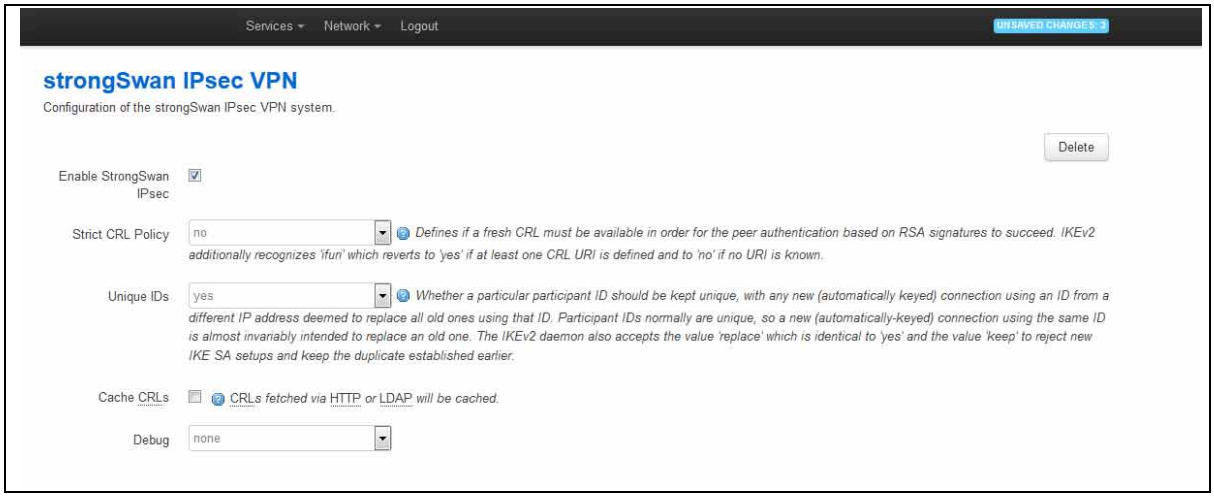



**Figure 146: The common settings section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable strongswan<br>UCI: strongswan.general.enable<br>Opt: enabled | Enables or disables IPSec.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Strict CRL Policy<br>UCI: strongswan.general.strictcrlpolicy<br>Opt: strictcrlpolicy | Defines if a fresh CRL must be available for the peer authentication based on RSA signatures to succeed.<br>0 – Disabled.<br>1 – Enabled.<br>ifuri – 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. |
| Web: Unique IDs<br>UCI: strongswan.general.uniqueids<br>Opt: uniqueids | 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.<br>Participant IDs normally are unique, so a new (automatically-keyed) connection using the same ID is almost invariably intended to replace an old one.<br>0 – Disabled.<br>1 – Enabled.<br>replace – Identical to Yes<br>keep – Rejects new IKE SA and keep the duplicate established earlier |
| Web: Cache CRLs<br>UCI: strongswan.general.cachecrls<br>Opt: cachecrls | 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.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Debug<br>UCI: strongswan.general.debug<br>Opt: debug | Enable debugging. This option is used for trouble shooting issues. It is not suitable for a production environment.<br>None – Debug disabled.<br>Control – Debug enabled. Shows generic control flow with errors and very basic auditing logs.<br>All – Debug enabled. Most verbose logging also includes sensitive information such as keys. |

**Table 97: 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

| | | |
|---|---|---|
| Enabled | ☑ | |
| Aggressive Mode | ☑ | |
| Name | DMVPN_VDF | |
| Autostart Action | ignore | *Operation on startup.**add** loads a connection without starting it. **route** loads a connection and installs kernel traps. If traffic is detected between localLan and remoteLan, a connection is established.**start** loads a connection and brings it up immediately. **ignore** do nothing* |
| Connection Type | transport | |
| Remote GW Address | | *Could be IP address or FQDN or '%any'* |
| Local Id | | *Leave blank to use default (local interface IP address)* |
| Remote Id | | *Leave blank to use default (remote gateway IP address)* |
| Local LAN IP Address | | |
| Local LAN IP Address Mask | | |
| Remote LAN IP Address | | |
| Remote LAN IP Address Mask | | |
| Local Protocol | gre | *Restrict the traffic selector to a single protocol on the local side* |
| Local Port | | *Restrict the traffic selector to a single UDP/TCP port on the local side* |
| Remote Protocol | gre | *Restrict the traffic selector to a single protocol on the remote side* |
| Remote Port | | *Restrict the traffic selector to a single UDP/TCP port on the remote side* |
| Authby | psk | *How the two security gateways should authenticate each other.* |
| XAuth identity | | *Defines the identity/username the client uses to reply to an XAuth request. If not defined, the IKEv1 identity will be used as XAuth identity.* |
| IKE algorithm | aes128-sha1-modp1024 | |
| ESP algorithm | 3des-md5 | |
| WAN Interface | 3GVDF | |
| IKE life time | 3h | *How long the keying channel of a connection should last before being renegotiated.* |
| Key life | 1h | *Synonym for lifetime. How long a particular instance of a connection (a set of encryption/authentication keys for user packets) should last, from successful negotiation to expiry.* |
| Rekey margin | 9m | *Synonym for margintime. How long before connection expiry or keying-channel expiry should attempts to negotiate a replacement begin.* |
| Keyring tries | 3 | *How many attempts (a positive integer or %forever) should be made to negotiate a connection, or a replacement for one, before giving up (default 3). The value %forever means 'never give up'.* |
| DPD Action | none | *Controls the use of the DPD protocol where R_U_THERE notification messages (IKEv1) or empty INFORMATIONAL messages (IKEv2) are periodically sent in order to check the liveliness of the IPsec peer. If no activity is detected, all connections with a dead peer are stopped and unrouted (clear), put in the hold state (hold) or restarted (restart). The default is none which disables the active sending of DPD messages.* |
| DPD Delay | 30s | *Defines the period time interval with which R_U_THERE messages/INFORMATIONAL exchanges are sent to the peer.* |
| DPD Timeout | 30s | *Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity.* |

**Figure 147: The connections settings section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: strongswan.@connection[X].enabled<br>Opt: enable | Enables or disables IPSec connection.<br>0 Disabled.<br>1 Enabled. |
| Web: Aggressive<br>UCI: strongswan.@connection[X].aggressive<br>Opt: aggressive | Enables or disables IKE aggressive mode.<br>**Note**: using aggressive mode along with PSK authentication is less secure method than main mode and should be avoided.<br>0 Disabled.<br>1 Enabled. |
| Web: Name<br>UCI: strongswan.@connection[X].name<br>Opt: name | Specifies a name for the tunnel. |
| Web: Autostart Action<br>UCI: strongswan.@connection[X].auto<br>Opt: auto | Specifies when the tunnel is initiated.<br>start On start up.<br>route When traffic routes this way.<br>add Loads a connection without starting it.<br>ignore Ignores the connection.<br>always Actively retries to establish the tunnel if it went down. |
| Web: Connection Type<br>UCI: strongswan.@connection[X].type<br>Opt: type | Defines the type of IPSec connection.<br>tunnel Connection uses tunnel mode.<br>transport Connection uses transport mode.<br>pass Connection does not perform any IPSec processing.<br>drop Connection drops all the packets. |
| Web: Remote GW Address<br>UCI: strongswan.@connection[X]. remoteaddress<br>Opt: remoteaddress | Sets the public IP address of the remote peer.<br>Leave blank for DMVPN. |
| Web: Local ID<br>UCI: strongswan.@connection[X].localid<br>Opt: localid | Defines the local peer identifier.<br>Leave blank for DMVPN. |
| Web: Remote ID<br>UCI: strongswan.@connection[X].remoteid<br>Opt:remoteid | Defines the remote peer identifier.<br>Leave blank for DMVPN. |
| Web: Local LAN IP Address<br>UCI: strongswan.@connection[X]. locallan<br>Opt: locallan | Defines the local IP of LAN.<br>Leave blank for DMVPN. |
| Web: Local LAN IP Address Mask<br>UCI: strongswan.@connection[X]. locallanmask<br>Opt: locallanmask | Defines the subnet of local LAN.<br>Leave blank for DMVPN. |
| Web: Remote LAN IP Address<br>UCI: strongswan.@connection[X]. remotelan<br>Opt:remotelan | Defines the IP address of LAN serviced by remote peer.<br>Leave blank for DMVPN. |
| Web: Remote LAN IP Address Mask<br>UCI: strongswan.@connection[X]. remotelanmask<br>Opt:remotelanmask | Defines the Subnet of remote LAN.<br>Leave blank for DMVPN. |
| Web: Local Protocol<br>UCI: strongswan.@connection[X].localproto<br>Opt: localproto | Restricts the connection to a single protocol on the local side. |

<table>
<tr><td>Web: Local Port<br>UCI: strongswan.@connection[X].localport<br>Opt: localport</td><td>Restricts the connection to a single port on the local side.</td></tr>
<tr><td>Web: Remote Protocol<br>UCI: strongswan.@connection[X].remoteproto<br>Opt:remoteproto</td><td>Restricts the connection to a single protocol on the remote side.</td></tr>
<tr><td>Web: Remote Port<br>UCI: strongswan.@connection[X].remoteport<br>Opt: remoteport</td><td>Restricts the connection to a single port on the remote side.</td></tr>
<tr><td>Web: Authby<br>UCI: strongswan.@connection[X].authby<br>Opt: authby</td><td>Defines how the two secure gateways should authenticate.<br>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>
<tr><td>Web: XAuth Identity<br>UCI: strongswan.@connection[X].xauth_identity<br>Opt: xauth_identity</td><td>Defines Xauth ID.</td></tr>
<tr><td>Web: IKE Algorithm<br>UCI: strongswan.@connection[X].ike<br>Opt: ike</td><td>Specifies the IKE algorithm to use.<br>The format is: encAlgo | authAlgo | DHGroup:<br>encAlgo:<br>3des<br>aes128<br>aes256<br>serpent<br>twofish<br>blowfish<br>authAlgo:<br>md5<br>sha<br>sha2<br>DHGroup:<br>modp1024<br>modp1536<br>modp2048<br>modp3072<br>modp4096<br>modp6144<br>modp8192<br>For example, a valid IKE algorithm is: aes128-sha-modp1536.</td></tr>
</table>

| | |
|---|---|
| Web: ESP algorithm<br>UCI: strongswan.@connection[X].esp<br>Opt: esp | Specifies the esp algorithm to use.<br>The format is: encAlgo \| authAlgo \| DHGroup<br>encAlgo:<br>3des<br>aes128<br>aes256<br>serpent<br>twofish<br>blowfish<br>authAlgo:<br>md5<br>sha<br>sha2<br>DHGroup:<br>modp1024<br>modp1536<br>modp2048<br>modp3072<br>modp4096<br>modp6144<br>modp8192<br>For example, a valid encryption algorithm is:<br>aes128-sha-modp1536.<br>If no DH group is defined then PFS is disabled. |
| Web: WAN Interface<br>UCI: strongswan.@connection[X].waniface<br>Opt: waniface | This is a space separated list of the WAN interfaces the router will use to establish a tunnel with the secure gateway.<br>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.<br>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<br>UCI: strongswan.@connection[X].ikelifetime<br>Opt:ikelifetime | Specifies how long the keyring channel of a connection (ISAKMP or IKE SA) should last before being renegotiated.<br>3h<br>Timespec: 1d, 3h, 25m, 10s. |
| Web: Key Life<br>UCI: strongswan.@connection[X].keylife<br>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.<br>Normally, the connection is renegotiated (via the keying channel) before it expires (see rekeymargin).<br>1h<br>Timespec: 1d, 1h, 25m, 10s. |
| Web: Rekey Margin<br>UCI: strongswan.@connection[X].rekeymargin<br>Opt: rekeymargin | Specifies how long before connection expiry or keying-channel expiry should attempt to negotiate a replacement begin.<br>Relevant only locally, other end need not agree on it.<br>9m<br>Timespec: 1d, 2h, 9m, 10s. |
| Web: Keyring Tries<br>UCI: strongswan.@connection[X].keyringtries<br>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<br>UCI: strongswan.@connection[X].dpdaction<br>Opt: dpdaction | Defines DPD (Dead Peer Detection) action.<br>None – Disables DPD.<br>Clear – Clear down the tunnel if peer does not respond. Reconnect when traffic brings the tunnel up.<br>Hold – Clear down the tunnel and bring up as soon as the peer is available.<br>Restart – Restarts DPD when no activity is detected. |
| Web: DPD Delay<br>UCI: strongswan.@connection[X].dpddelay<br>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.<br>30s<br>Timespec – 1d, 2h, 25m, 10s. |
| Web: DPD Timeout<br>UCI: strongswan.@connection[X].dpdtimeout<br>Opt: dpdtimeout | Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity.<br>150s<br>Timespec – 1d, 2h, 25m, 10s. |

**Table 98: 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 148: IPSec secrets settings**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: strongswan.@secret[X].enabled<br>Opt: enabled | Defines whether this set of credentials is to be used or not.<br>0 – Disabled.<br>1 – Enabled. |
| Web: ID selector<br>UCI: strongswan.@secret[X].idtype<br>Opt: idtype | Defines whether IP address or userfqdn is used. |
| Web: ID selector<br>UCI: strongswan.@secret[X].localaddress<br>Opt: localaddress | Defines the local address this secret applies to. |
| Web: ID selector<br>UCI: strongswan.@secret[X]. remoteaddress<br>Opt: remoteaddress | Defines the remote address this secret applies to. |

| | |
|---|---|
| Web: N/A<br>UCI: strongswan.@secret[X].userfqnd<br>Opt: userfqnd | FQDN or Xauth name used of Extended Authentication. This must match xauth_identity from the configuration connection section. |
| Web: Secret Type<br>UCI: strongswan.@secret[X].secrettype<br>Opt: secrettype | Specifies the authentication mechanism to be used by the two peers.<br>Psk – Preshared secret<br>Pubkey – Public key signatures<br>Rsasig – RSA digital signatures<br>Ecdsasig – Elliptic Curve DSA signatures<br>Xauth – Extended authentication |
| Web: Secret<br>UCI: strongswan.@secret[X].secret<br>Opt: secret | Defines the secret. |

**Table 99: 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 149: 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.

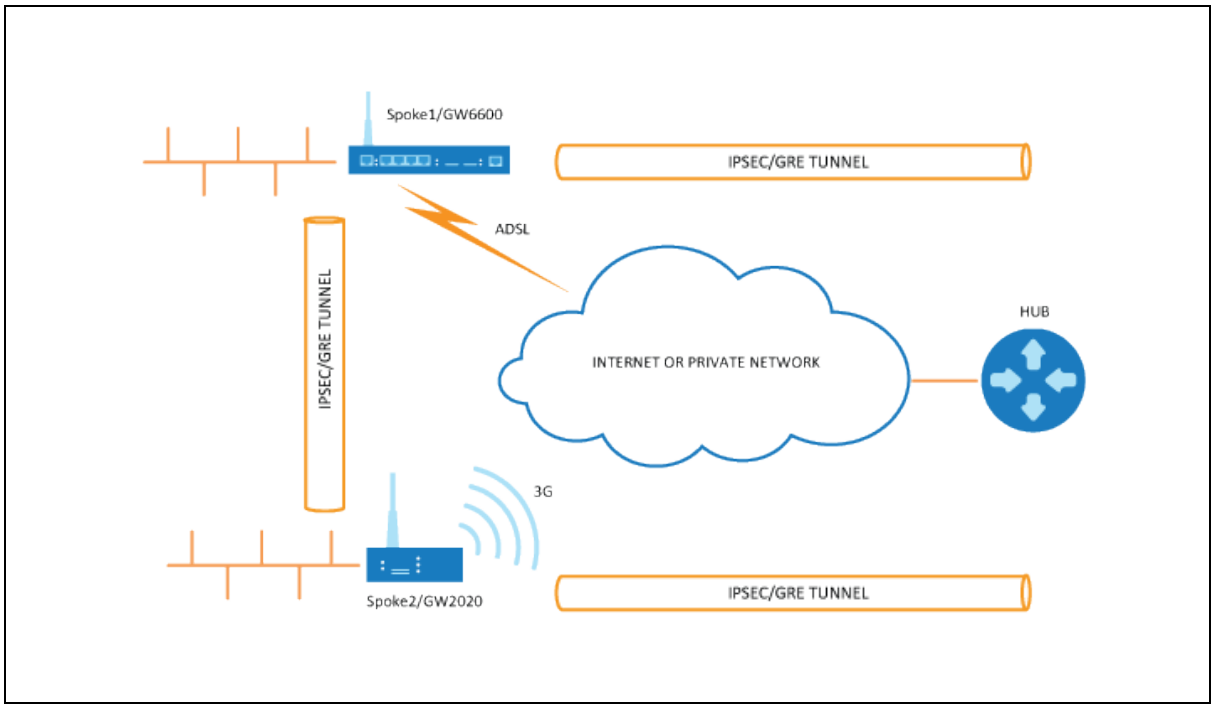



**Figure 150: Network diagram for DMVPN spoke to spoke**

- 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.

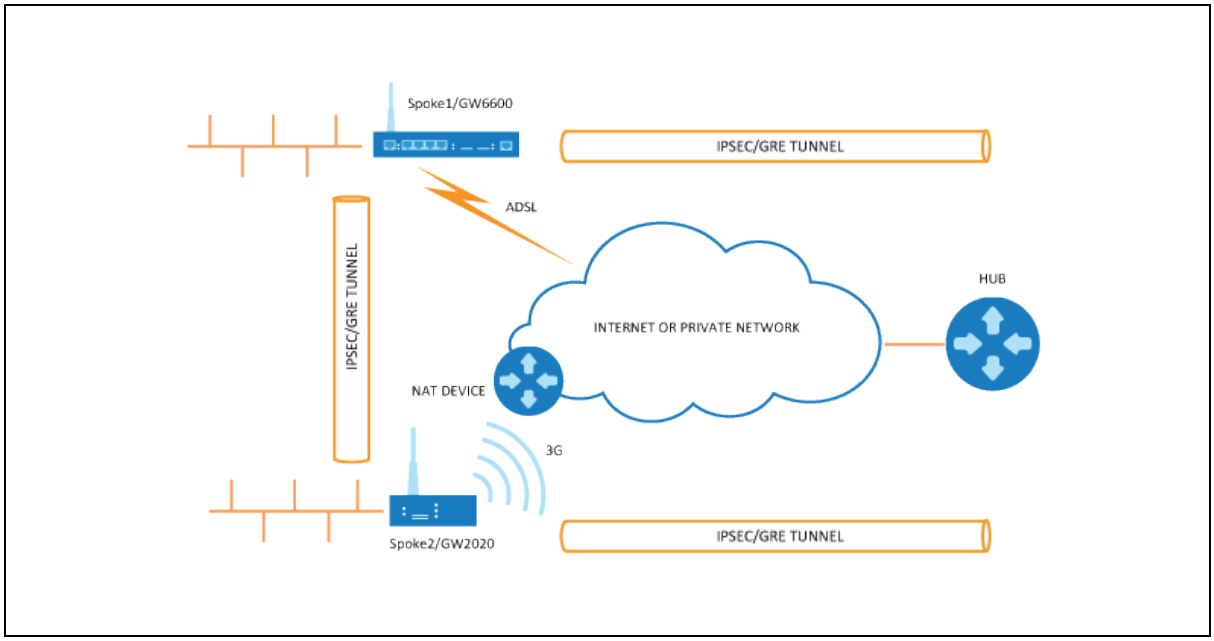



**Figure 151: Network diagram for DMVPN spoke behind NAT**

- 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 152: The DMVPN general section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable DMVPN<br>UCI: dmvpn.common.enabled<br>Opt: enable | Enables DMVPN.<br>0 Disabled.<br>1 Enabled. |
| Web: IPSec template connection<br>UCI: dmvpn.common.ipsec_template_name<br>Opt: ipsec_template_name | Selects the IPSec connection, defined in strongSwan, to be used as a template. |

**Table 100: Information table for DMVPN general settings**

## 30.5.2 DMVPN hub settings

**Figure 153: The DMVPN hub settings**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: GRE Interface<br>UCI: dmvpn.@interface[X].gre_interface<br>Opt: gre_interface | Specifies which GRE interface will be used with this DMVPN configuration. |
| Web: GRE Remote Endpoint IP Address<br>UCI: dmvpn.@interface[X].gre_endpoint_ip<br>Opt: gre_endpoint_ip | Configures the GRE IP address of the hub. |
| Web: GRE Remote Endpoint Mask Length<br>UCI: dmvpn.@interface[X].gre_endpoint_mask_length<br>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<br>UCI: dmvpn.@interface[X].nhs_ip<br>Opt: nhs_ip | Configures the physical IP address for the DMVPN hub. |
| Web: NHRP Authentication<br>UCI: dmvpn.@interface[X].cisco_auth<br>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<br>UCI: dmvpn.@interface[X].holding_time<br>Opt: holding_time | Timeout for cached NHRP requests. |
| Web: Use As Default Route<br>UCI : dmvpn.@interface[X].defaultroute<br>Opt: defaultroute | Adds a default route into tunnel interface.<br>0 Disabled.<br>1 Enabled. |
| Web: Default Route Metric<br>UCI: dmvpn.@interface[X].defaultroutemetric<br>Opt: defaultroutemetric | Metric to use for the default route. |
| Web: LED state indication<br>UCI: dmvpn.@interface[X].led<br>Opt: led | LED to use for indicating if the VPN is up. |

**Table 101: 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 154: 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 155: 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 102: 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<br>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**.

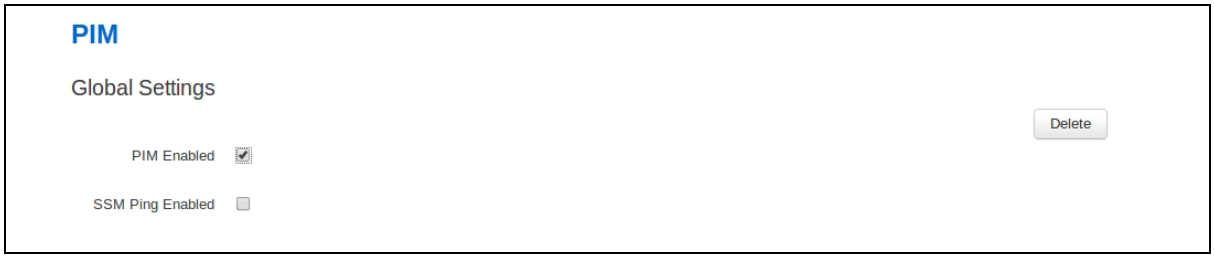



**Figure 156: The global settings interface**

### 31.3.1 Global settings

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: PIM Enabled<br>UCI: pimd.pimd.enabled<br>Opt: enabled | Globally enables PIM on the router.<br>0 Disabled.<br>1 Enabled. |
| Web: SSM Ping Enabled<br>UCI: pimd.pimd.ssmpingd<br>Opt: ssmpingd | Enables answers to SSM pings.<br>0 Disabled.<br>1 Enabled. |

**Table 103: Information table for PIM global settings**

### 31.3.2 Interfaces configuration

**Figure 157: The interfaces configuration section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: pimd.interface[x].enabled<br>Opt: enabled | Enables multicast management of the given interface by the PIM application.<br>0 Disabled.<br>1 Enabled. |
| Web: Interface<br>UCI: pimd.interface[x].interface<br>Opt: interface | Selects the interface to apply PIM settings to. |
| Web: Enable IGMP<br>UCI: pimd.interface[x].igmp<br>Opt: igmp | Enable IGMP on given interface.<br>0 Disabled.<br>1 Enabled.<br>**Note**: you must enable PIM SSM and/or IGMP depending on your requirements.<br>ICMP must be enabled on the interface to the multicast client only. |
| Web: Enable SSM<br>UCI: pimd.interface[x].ssm<br>Opt: ssm | Enable SSM on given interface.<br>0 Disabled.<br>1 Enabled. |

**Table 104: 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<br>Opt: list vlan_qos_map_ingress | VLAN priority code point to socket buffer mapping.<br>Example: network.<if name>. vlan_qos_map_ingress =1:1 |
| UCI: network.<if name>.vlan_qos_map_egress<br>Opt: list vlan_qos_map_egress | Socket buffer to VLAN priority code point mapping.<br>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.

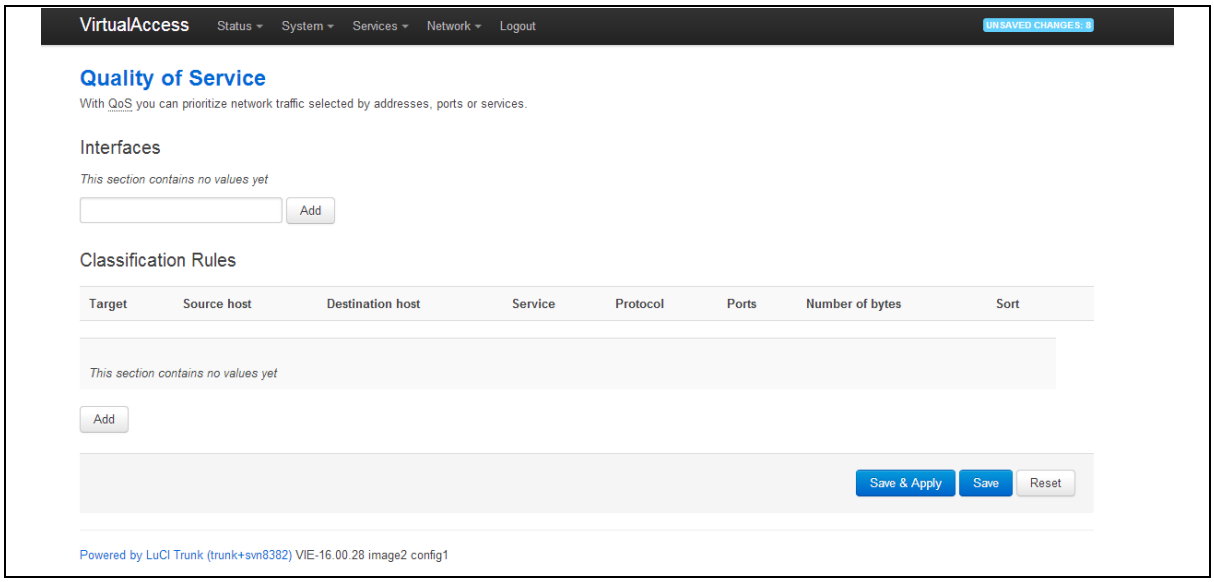



**Figure 158: The quality of service page**

To configure an interface, enter a relevant interface name and click **Add**. The Quality of Service page for that interface appears.

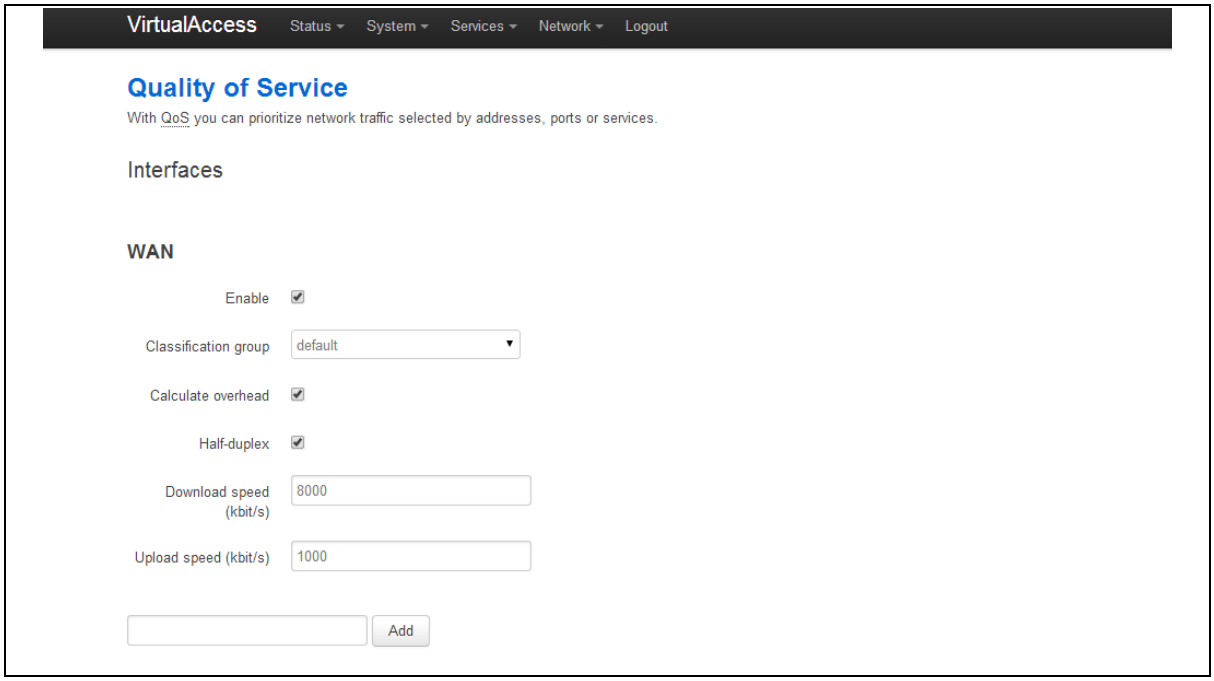



**Figure 159: The quality of service page for WAN interface**

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<br>UCI: qos.[interface].enabled<br>Opt: enabled | Enables or disables QoS interface.<br>1 Enabled.<br>0 Disabled. |
| Web: Classification group<br>UCI: qos. [interface].classgroup<br>Opt: classgroup | Creates a mapping before previously created classgroup and interface to which it should be assigned to. |
| Web: Calculate overhead<br>UCI: qos. [interface].overhead<br>Opt: overhead | Decreases upload and download ratio to prevent link saturation. |
| Web: Half-duplex<br>UCI: qos [interface].halfduplex<br>Opt: halfduplex | Enables or disables half-duplex operation.<br>1 Enabled.<br>0 Disabled. |
| Web: Download speed<br>UCI: qos.[interface].download<br>Opt: download | Download speed limit in kbits/sec. |
| Web: Upload speed<br>UCI: qos.[interface].upload=2000<br>Opt:upload | Upload speed limit in kbits/sec. |

**Table 105: Information table for QoS page**

To add classification rules, click **Add**. TheClassification Rules section appears.

Configure each classification rule with the following parameters.

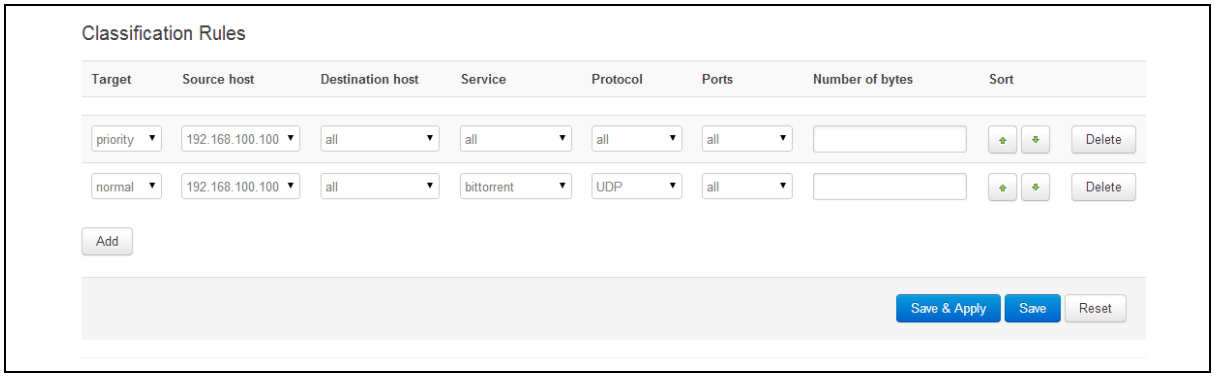



**Figure 160: Parameters for classification rules**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Target<br>UCI:<br>Opt: | Creates and configures selected target bucket.<br>Normal<br>Priority<br>Low<br>Express |
| Web: Source host<br>UCI:<br>Opt: | Source host. |
| Web: Destination host<br>UCI:<br>Opt: | Destination host. |
| Web: Service<br>UCI:<br>Opt: | Selectable service. |
| Web: Protocol<br>UCI:<br>Opt: | Protocol to classify. |
| Web: Ports<br>UCI:<br>Opt: | Upload speed kbits/sec. |
| Web: Number of bytes<br>UCI:<br>Opt: | Number of bytes for bucket. |

**Table 106: 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<br>UCI: qos.[interface].enabled<br>Opt: enabled | Enables or disables QoS interface.<br>1 Enabled.<br>0 Disabled. |
| Web: Classification group<br>UCI: qos. [interface].classgroup<br>Opt: classgroup | Creates a mapping before previously created classgroup and interface to which it should be assigned to. |
| Web: Calculate overhead<br>UCI: qos. [interface].overhead<br>Opt: overhead | Decrease upload and download ratio to prevent link saturation. |
| Web: Half-duplex<br>UCI: qos [interface].halfduplex<br>Opt: halfduplex | Enables or disables half-duplex operation.<br>1 Enabled.<br>0 Disabled. |
| Web: Download speed<br>UCI: qos.[interface].download<br>Opt: download | Download speed limit in kbits/sec. |
| Web: Upload speed<br>UCI: qos.[interface].upload=2000<br>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<br>Opt: Default | Specifies name of classgroup. |
| UCI: qos.Default.classes=Express Normal<br>Opt: classes | Specifies the list of names of classes which should be part of classgroup. |
| qos.Default.default=Normal<br>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<br>Opt: Normal | Specifies class name. |
| UCI: qos.Normal.packetsize=1500<br>Opt: packetsize | Specifies packet size for the class in bytes. |
| UCI: qos.Normal.avgrate=30<br>Opt: avgrate | Average rate for this class, value in % of bandwidth in %. |
| UCI: qos.Normal.priority=5<br>Opt: priority | Specifies priority for the class in %. |
| UCI: qos.Express=class<br>Opt: Express | Specifies class name. |
| UCI: qos.Express.packetsize=1000<br>Opt: packetsize | Specifies packet size for the class in bytes. |
| UCI: qos.Express.maxsize=800<br>Opt: maxsize | Specify max packet size in bytes. |
| UCI: qos.Express.avgrate=50<br>Opt: avgrate | Average rate for this class, value in % of bandwidth in %. |
| UCI: qos.Express.priority=10<br>Opt: priority | Specifies priority for the class in %. |
| UCI: qos.Express.limitrate=10<br>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<br>Opt: classify | Part of classify rule. |
| UCI: qos.@classify[0].target=Express<br>Opt: target | Specifies target class. |
| UCI: qos.@classify[0].proto=udp<br>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.

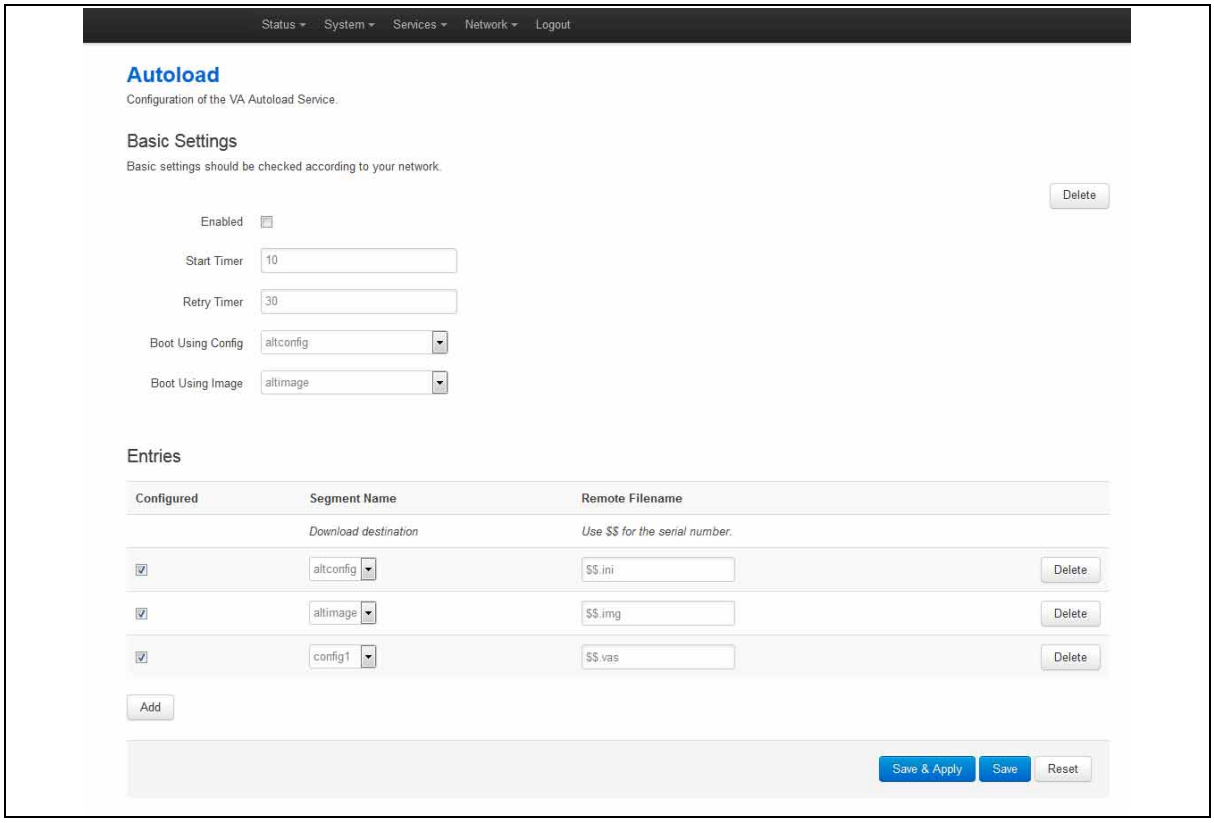



**Figure 161: The autoload settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Basic settings | |
| Web: Enabled<br>UCI: autoload.main.enabled<br>Opt: Enabled | Enables activation at system boot.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Start Timer<br>UCI: autoload.main.StartTimer<br>Opt: StartTimer | Defines how long to wait after the boot up completes before starting activation.<br>10<br>Range: 0-300 secs |
| Web: Retry Timer<br>UCI: autoload.main.RetryTimer<br>Opt: RetryTimer | Defines how many seconds to wait between retries if a download of a particular autoload entry fails.<br>30<br>Range: 0-300 secs |
| Web: N/A<br>UCI: autoload.main.NumberOfRetries<br>Opt: Numberofretries | Defines how many retries to attempt before failing the overall activation sequence, backing off and trying the whole activation sequence again.<br>5<br>Range |
| Web: N/A<br>UCI: autoload.main.BackoffTimer<br>Opt: Backofftimer | 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.<br>15<br>Range |

<table>
<tr><td>Web: Boot Using Config<br>UCI: autoload.main.BootUsingConfig<br>Opt: BootUsingConfig</td><td>Specifies which configuration to boot up with after the activation sequence.<br><table><tr><td>Altconfig</td><td>Alternative configuration</td></tr><tr><td>Config1</td><td>Configuration 1</td></tr><tr><td>Config2</td><td>Configuration 2</td></tr><tr><td>Factconf</td><td>Factory configuration</td></tr></table></td></tr>
<tr><td>Web: Boot Using Image<br>UCI: autoload.main.BootUsingImage<br>Opt: BootUsingImage</td><td>Specifies which image to boot up with after the activation sequence completes successfully.<br><table><tr><td>Altimage</td><td>Alternative image</td></tr><tr><td>Image 1</td><td>image 1</td></tr><tr><td>Image 2</td><td>image 2</td></tr></table></td></tr>
<tr><td colspan="2">Entries</td></tr>
<tr><td>Web: Configured<br>UCI: autoload.@entry[x].Configured<br>Opt: Configured</td><td>Enables the autoload sequence to process this entry.<br><table><tr><td>1</td><td>Enabled.</td></tr><tr><td>0</td><td>Disabled.</td></tr></table></td></tr>
<tr><td>Web: Segment Name<br>UCI: autoload.@entry[x].SegmentName<br>Opt: SegmentName</td><td>Defines where the downloaded file should be stored:<br>(config1 | config2 | altconfig | image1 | image2 | altimage).<br>Typically only altconfig and altimage are used.</td></tr>
<tr><td>Web: RemoteFilename<br>UCI: autoload.@entry[x].RemoteFilename<br>Opt: RemoteFilename</td><td>Defines the name of the file to be downloaded from Activator.<br><table><tr><td>$$.vas</td><td>Notifies activator sequence is complete.</td></tr><tr><td>$$ ini</td><td>Request configuration</td></tr><tr><td>$$ img</td><td>Request firmware</td></tr></table>Note: $$.vas should always be requested last.</td></tr>
</table>

**Table 107: 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.

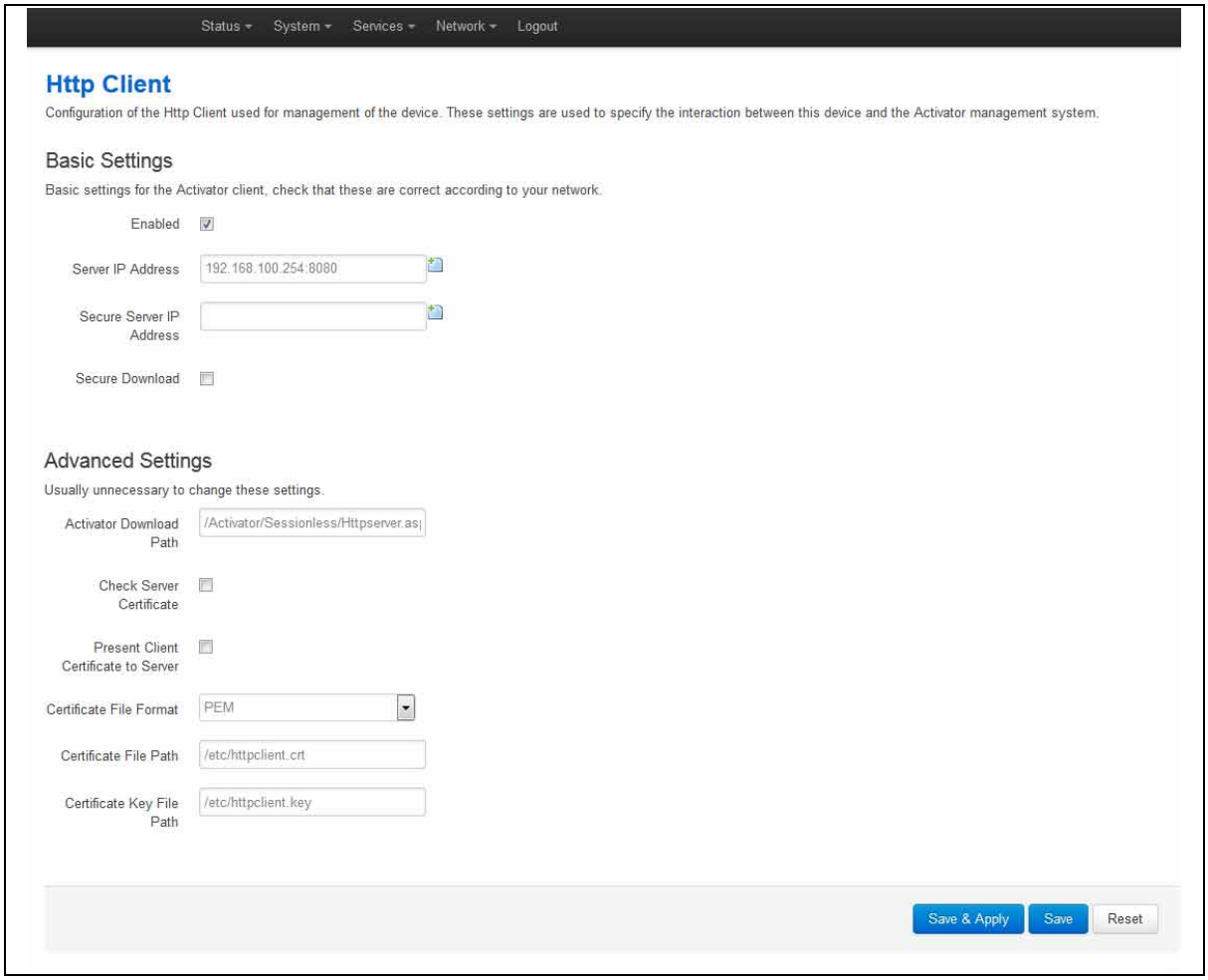



**Figure 162: The HTTP client page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Basic settings | |
| Web: Enabled<br>UCI: httpclient.default.enabled<br>Opt: Enabled | Enables the HTTP client.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Server IP Address<br>UCI: httpclient.default.Fileserver<br>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<br>UCI: httpclient.default.SecureFileServer<br>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<br>UCI: httpclient.default.SecureDownload<br>Opt: SecureDownload | Enables Secure Download (port 443).<br>1 — Enabled.<br>0 — Disabled. |
| Advanced settings | |
| Web: ActivatorDownloadPath<br>UCI: httpclient.default.ActivatorDownloadPath<br>Opt: ActivatorDownloadPath | Specifies the URL on Activator to which the client should send requests.<br>/Activator/Sessionless/Httpserver.asp<br>Range |

| | |
|---|---|
| Web: Check Server Certificate<br>UCI: httpclient.default.ValidateServerCertificateEnabled<br>Opt: ValidateServerCertificateEnabled | Checks for the certificates presence and validity.<br>1 — Enabled.<br>0 — Disabled. |
| Web: Present Client Certificate to Server<br>UCI: httpclient.default. PresentCertificateEnabled<br>Opt: PresentCertificateEnabled | Specifies if the client presents its certificate to the server to identify itself.<br>1 — Enabled.<br>0 — Disabled. |
| Web: CertificateFile Format<br>UCI: httpclient.default.CertificateFormat<br>Opt: CertificateFormat | Specifies the value the client expects to see in the specified field in the server certificate.<br>PEM<br>DER |
| Web: Certificate File Path<br>UCI: httpclient.default.CertificateFile<br>Opt: CertificateFile | Defines the directory/location of the certificate.<br>/etc/httpclient.crt<br>Range |
| Web: Certificate Key File Path<br>UCI: httpclient.default.CertificateKey<br>Opt: CertificateKey | Specifies the directory/location of the certificate key.<br>/etc/httpclient.key<br>Range |
| Web: N/A<br>UCI: ValidateServerCertificateFieldEnabled<br>Opt: ValidateServerCertificate | Defines the field in the server certificate that the client should check.<br>1 — Enabled.<br>0 — Disabled. |
| Web: N/A<br>UCI: httpclient.default.ActivatorChunkyDownloadPath<br>Opt: ActivatorChunkyDownloadPath | Enables partial download activations and active updates.<br>The default value is<br>httpclient.default.ActivatorChunkyDownloadPath=/activator/partial/download<br>The url (on activator) to which the client should send requests for chunky image download. |
| Web: N/A<br>UCI: httpclient.default.ChunkSize<br>Opt: ChunkSize | Specifies the size of each packet payload<br>100k — 100K Bytes<br>1-infinite — Available values |
| Web: N/A<br>UCI: httpclient.default.RateLimit<br>Opt: RateLimit | Throttle activation/active updates traffic received by device to specified limit<br>None — By default there is no limit<br>1-infinite — Available values in kbps |
| Web: N/A<br>UCI: httpclient.default.CAFile<br>Opt: CAFile | Defines path to the certificate authority file stored on the router |
| Web: N/A<br>UCI: httpclient.default.IgnoreServerCertificateStatus<br>Opt: IgnoreServerCertificateStatus | Defines whether to skip the status check on the server certificate.<br>1 — Enabled.<br>0 — Disabled. |

**Table 108: 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<br>UCI: management_users.@user[x].enabled<br>Opt: enable | Enables/creates the user.<br>0 Disabled.<br>1 Enabled. |
| Web: n/a<br>UCI: management_users.@user[x].username<br>Opt: username | Specifies the user's username. |
| Web: n/a<br>UCI: management_users.@user[x].password<br>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.<br>UCI: management_users.@user[x].hashpassword<br>Opt: hashpassword. Note: a SRP user password will be displayed using the srphash option |
| Web: n/a<br>UCI: management_users.@user[x].webuser<br>Opt: webuser | Specifies web access permissions for the user. Note: webuser will only work if linuxuser is set to Enabled.<br>0 Disabled.<br>1 Enabled. |
| Web: n/a<br>UCI: management_users.@user[x].chapuser<br>Opt: chapuser | Specifies CHAP access permissions for the PPP connection. Note: chapuser will only work if linux user is set to Enabled.<br>0 Disabled.<br>1 Enabled. |
| Web: n/a<br>UCI: management_users.@user[x].papuser<br>Opt: papuser | Specifies PAP access permissions for the PPP connection.<br>0 Disabled.<br>1 Enabled. |

| | |
|---|---|
| Web: n/a<br>UCI: management_users.@user[x].srpuser<br>Opt: srpuser | Specifies SRP access permissions for the PPP connection.<br>0 Disabled.<br>1 Enabled. |
| Web: n/a<br>UCI: management_users.@user[x].smsuser<br>Opt: smsuser | Specifies SMS access permissions for the user.<br>0 Disabled.<br>1 Enabled. |
| Web: n/a<br>UCI: linuxuser<br>Opt: linuxuser | Specifies linuxuser access permissions for the user.<br>0 Disabled.<br>1 Enabled. |
| Web: n/a<br>UCI: List allowed_pages<br>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 109: 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.

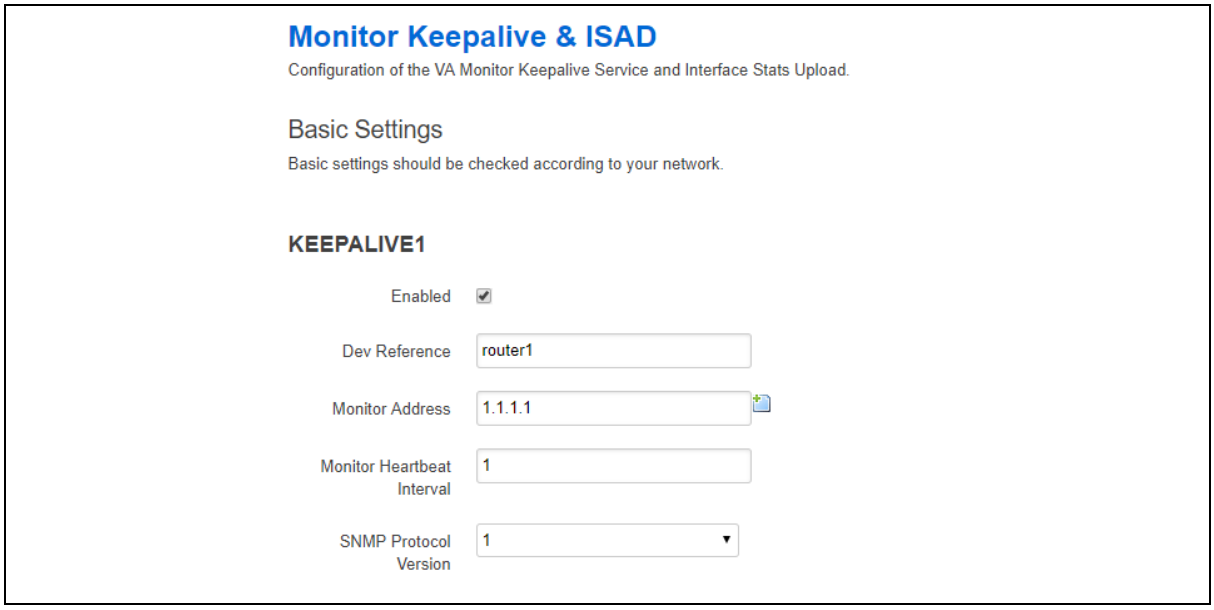



**Figure 163: The Monitor & ISAD keepalive page**

### 35.2.2.1 Basic settings

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: monitor.@keepalive[0].enabled<br>Opt: Enabled | Enables Monitor to send heartbeats to the router.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Dev Reference<br>UCI: monitor.@keepalive[0].dev_reference<br>Opt: dev_reference | Sets a unique identification for this device known to Monitor. |
| Web: Monitor Address<br>UCI: monitor.@keepalive[0].monitor_ip<br>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<br>UCI: monitor.@keepalive[0].interval_min<br>Opt: interval_min | Specifies the interval, in minutes, at which traps are sent.<br>1 — Trap set every 1 minute.<br>Range |
| Web: SNMP Protcol Version<br>UCI: monitor.@keepalive[0].snmp_version<br>Opt: snmp_version | Specifies what SNMP version is sent to remote Manager.<br>1 — snmp version 1<br>2c — SNMP version 2c<br>3 — SNMP version 3 |

**Table 110: Information table for Monitor & ISAD basic configuration**

The figure below shows options that are relevant only if you have selected SNMP version 3.

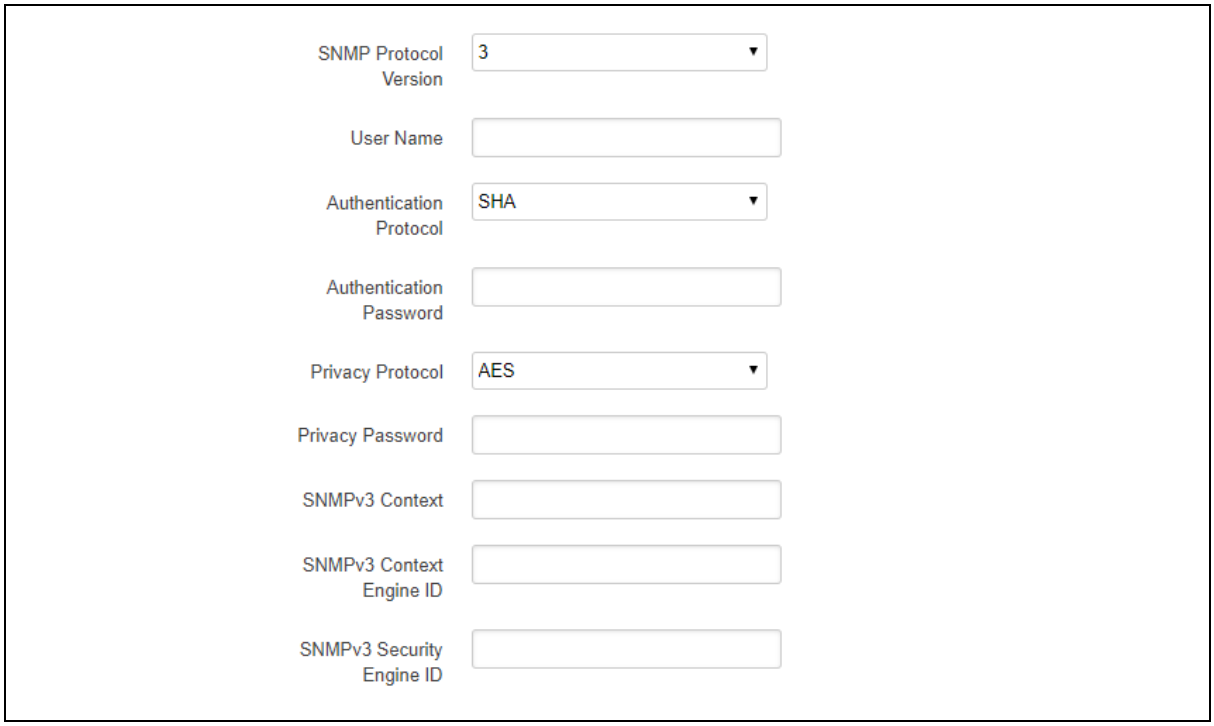



**Figure 164: The Monitor & ISAD keepalive page for SNMP v3**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: User Name<br>UCI: monitor.@keepalive[0].snmp_uname<br>Opt: snmp_uname | Specifies user name.<br>Blank — Default value<br>String |
| Web: Authentication Password<br>UCI: monitor.@keepalive[0].snmp_auth_pass<br>Opt: snmp_auth_pass | Specifies snmpv3 authentication password. |
| Web: Authentication Protocol<br>UCI: monitor.@keepalive[0].snmp_auth_proto<br>Opt: snmp_auth_proto | Specifies snmpv3 authentication protocol.<br>Blank — Default value<br>MD5 — MD5 as authentication protocol<br>SHA — SHA as authentication protocol |
| Web: Privacy Protocol<br>UCI: monitor.@keepalive[0].snmp_priv_proto<br>Opt: snmp_priv_proto | Specifies snmpv3 privacy protocol<br>Blank — Default value<br>AES — AES as privacy protocol<br>DES — MD5 as privacy protocol |
| Web: Privacy Password<br>UCI: monitor.@keepalive[0].snmp_priv_pass<br>Opt: snmp_priv_pass | Specifies snmpv3 privacy password. |
| Web: SNMPv3 Context<br>UCI: monitor.@keepalive[0].snmp_context<br>Opt: snmp_context | Specifies snmpv3 context name. |
| Web: SNMPv3 Context Engine ID<br>UCI: monitor.@keepalive[0].snmp_context_eid<br>Opt: snmp_context_eid | Specifies snmpv3 context engine ID. |

| | |
|---|---|
| Web: SNMPv3 Security Engine ID<br>UCI: monitor.@keepalive[0].snmp_sec_eid<br>Opt: snmp_sec_eid | Specifies snmpv3 security engine ID. |

**Table 111: 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.

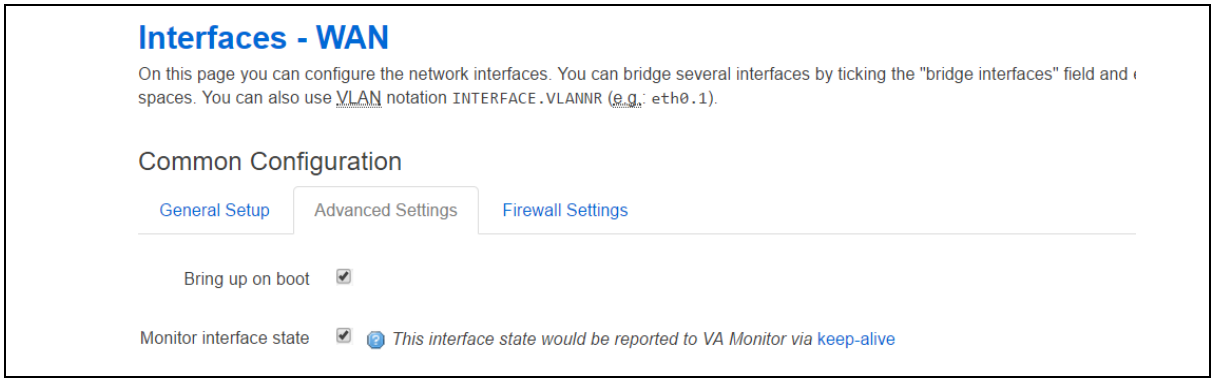



**Figure 165: The interface common configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Monitor interface state<br>UCI: network.@interface[0].monitored<br>Opt: monitored | Enables interface status to be sent in the heartbeat trap to Monitor.<br>0 Disabled.<br>1 Enabled. |

**Table 112: 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.

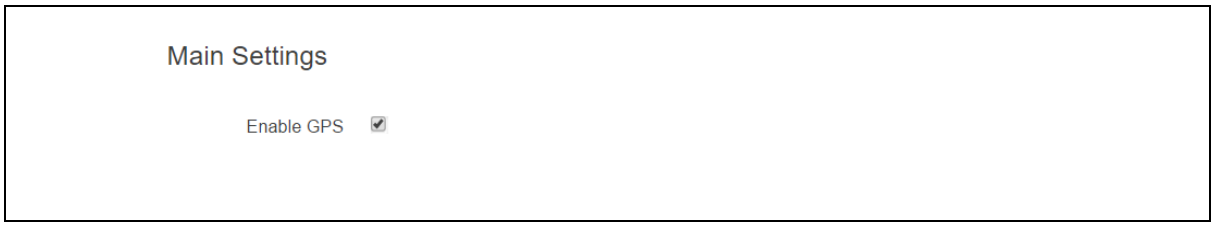



**Figure 166: The GPS configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable GPS<br>UCI: monitor.core.enabled<br>Opt: enabled | Enables GPS coordinates to be sent in the heartbeat keepalive to Monitor.<br>0 – Disabled.<br>1 – Enabled. |

**Table 113: 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**.

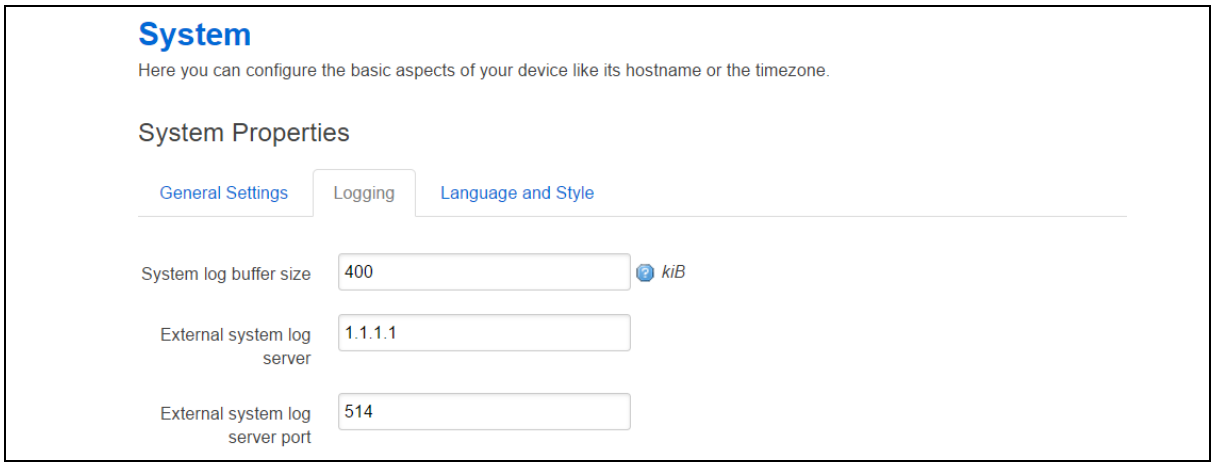



**Figure 167: The GPS status page**

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.

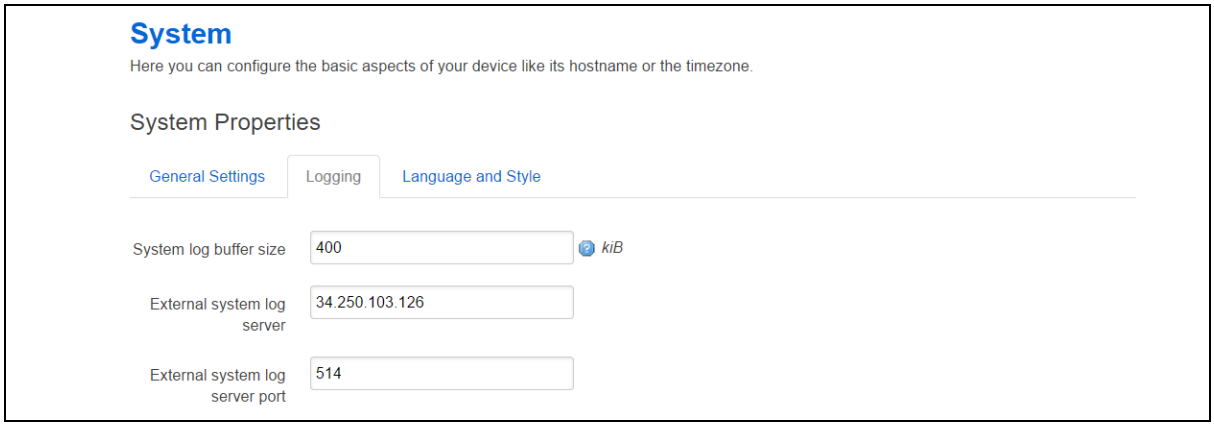



**Figure 168: The system properties page**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: External system log server<br>UCI: system.main.log_ip<br>Opt: log_ip | Defines the external syslog server IP address. |
| Web: External system log server<br>UCI: system.main.log_port<br>Opt: log_port | Defines the external syslog server destination port number for syslog messages.<br>514<br>Range |

**Table 114: 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.

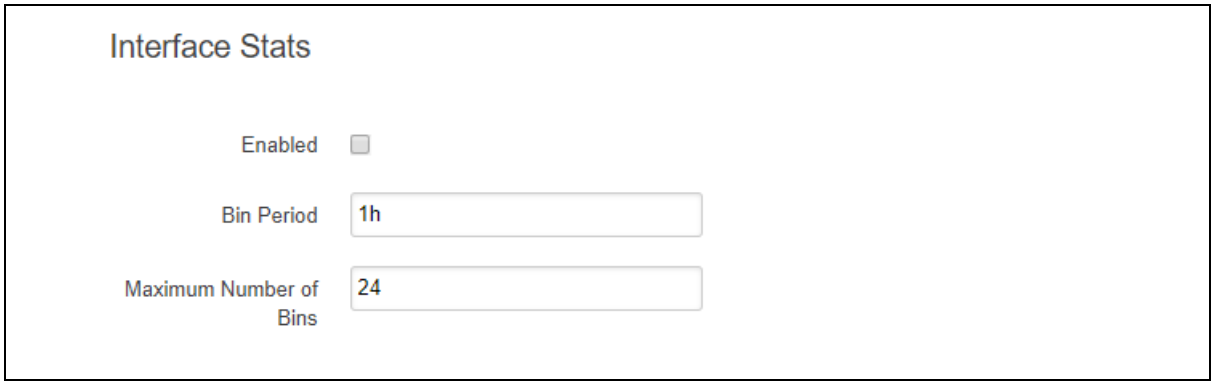



**Figure 169: The Monitor Keepalive & ISAD Interface Stats page**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: monitor.stats.enabled=1<br>Opt: enabled | Enables ISAD.<br>0 Disabled.<br>1 Enabled. |

| | |
|---|---|
| Web: Bin Period<br>UCI: monitor.stats.bin_period<br>Opt: time | Specifies how long to collect data for one bin.<br>Specifies the interval, in minutes, at which traps are sent.<br>1h: Bin collected for 1 hour<br>Range: |
| Web: Maximum Number of Bins<br>UCI: monitor.stats.bin_cache_size<br>Opt: bin_cache_size | Specifies the maximum number of bins to store.<br>Empty: 24<br>Range: |

**Table 115: 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<br>agent<br>com2sec<br>constant | exec<br>group<br>heartbeat<br>informreceiver | inventory<br>inventory_iftable<br>monitor_disk<br>monitor_ioerror | monitor_load<br>monitor_memory<br>monitor_process<br>pass | system<br>trapreceiver<br>usm_user<br>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.

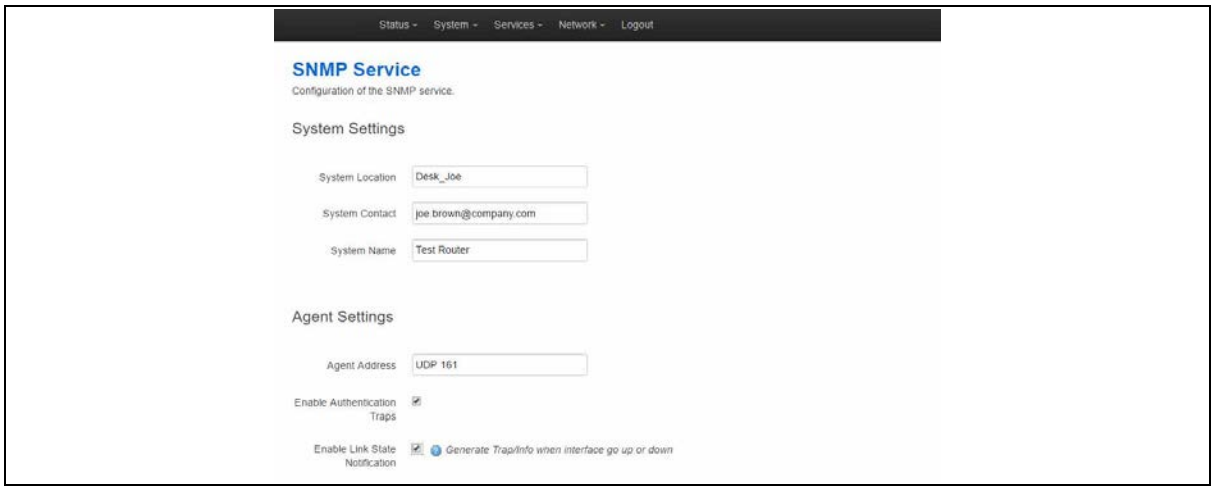



**Figure 170: The SNMP service page**

### 36.2.1 System and agent settings

| Web Field/UCI/Package Option | Description |
|---|---|
| System settings | |
| Web: System Location<br>UCI: snmpd.system[0].sysLocation<br>Opt: sysLocation | Sets the system location, system contact or system name for the agent. This information is reported in the 'system' group in the mibII tree. |
| Web: System Contact<br>UCI: snmpd.system[0].sysContact<br>Opt: sysContact | |
| Web: System Name<br>UCI: snmpd.system[0].sysName<br>Opt: sysName | |
| Agent Settings | |
| Web: Agent Address<br>UCI: snmpd.agent[0].agentaddress<br>Opt: agentaddress | Specifies the address(es) and port(s) on which the agent should listen.<br>[(udp\|tcp):]port[@address][,...] |
| Web: Enable Authentication Traps<br>UCI: snmpd.agent[0].authtrapenabled<br>Opt: authtrapenabled | Enables or disables SNMP authentication trap.<br>0 Disabled.<br>1 Enabled.<br>Note: this is the SNMP poll authentication trap to be set when there is a community mismatch. |
| Web: Enable Link State Notification<br>UCI: snmpd.agent[0].link_updown_notify<br>Opt: link_updown_notify | Generates trap/info when interface goes up or down. When enabled, the router sends a trap notification link up or down.<br>0 Disabled.<br>1 Enabled. |

**Table 116: 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.

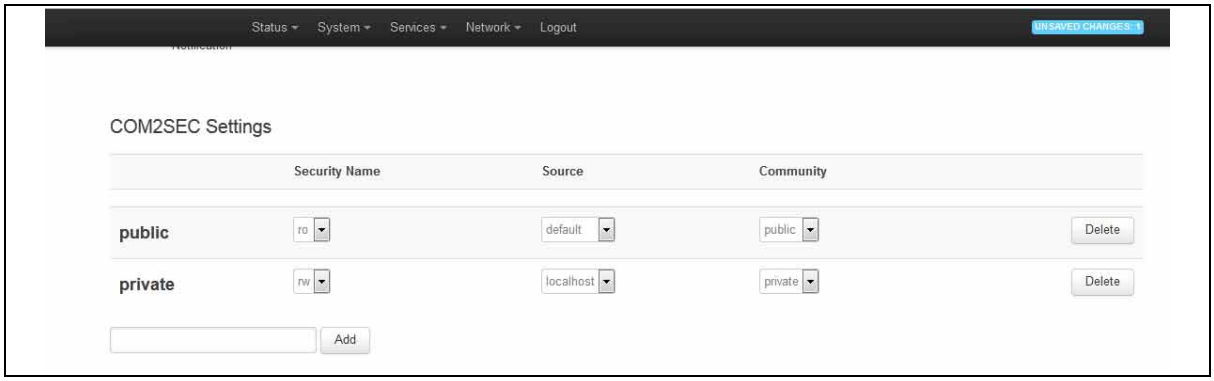



**Figure 171: The COM2Sec settings section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Security Name<br>UCI: snmpd.com2sec[x].secname<br>Opt: secname | Specifies an arbitrary security name for the user. |
| Web: Source<br>UCI: snmpd.com2sec[x].source<br>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<br>UCI: snmpd.com2sec[x].community<br>Opt: community | Specifies the community string being presented in the request. |

**Table 117: Information table for Com2Sec settings**

### 36.2.3 Group settings

Group settings assign community names and SNMP protocols to groups.

**Figure 172: The group settings section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Group<br>UCI: snmpd.group[x].group<br>Opt: group | Specifies an arbitrary group name. |
| Web: Version<br>UCI: snmpd.group[x].version<br>Opt: version | Specifies the SNMP version number being used in the request: v1, v2c and usm (User-based Security Module) are supported.<br>v1 — SNMP v1<br>v2v — SNMP v2<br>usm — SNMP v3<br>any — Any SNMP version |
| Web: Security Name<br>UCI: snmpd.group[x].secname<br>Opt: secname | An already defined security name that is being included in this group. |

**Table 118: 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 173: The view settings section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Name<br>UCI: snmpd.view[x].viewname<br>Opt: viewname | Specifies an arbitrary view name. Typically it describes what the view shows. |
| Web: Type<br>UCI: snmpd.view[x].type<br>Opt: type | 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).<br>included<br>excluded |
| Web: OID<br>UCI: snmpd.view[x].oid<br>Opt: oid | OID to be included in or excluded from the view. Only numerical representation is supported.<br>Example<br>1 — Everything<br>1.3.6.1.2.1.2 — Interfaces table |

**Table 119: 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 174: The access settings section**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Group<br>UCI: snmpd.access[x].group<br>Opt: group | Specifies the group to which access is being granted. |
| Web: Context<br>UCI: snmpd.access[x].context<br>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**.<br>none<br>all |
| Web: Version<br>UCI: snmpd.access[x].version<br>Opt: version | Specifies the SNMP version number being used in the request: any, v1, v2c and usm are supported.<br>v1 — SNMP v1<br>v2v — SNMP v2<br>usm — SNMP v3<br>any — Any SNMP version |
| Web: Level<br>UCI: snmpd.access[x].level<br>Opt: level | Specifies the security level. For SNMP v1 and SNMP v2c level must be **noauth**.<br>noauth<br>auth<br>priv |
| Web: Prefix<br>UCI: snmpd.access[x].prefix<br>Opt: prefix | Prefix specifies how context (above) should be matched against the context of the incoming pdu.<br>exact<br>any<br>all |
| Web: Read<br>UCI: snmpd.access[x].read<br>Opt: read | Specifies the view to be used for read access. |
| Web: Write<br>UCI: snmpd.access[x].write<br>Opt: write | Specifies the view to be used for write access. |
| Web: Notify<br>UCI: snmpd.access[x].notify<br>Opt: notify | Specifies the view to be used for notify access. |

**Table 120: 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 175: The trap receiver settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Host<br>UCI: snmpd.trapreceiver[x].host<br>Opt: host | Host address. Can be either an IP address or an FQDN. |
| Web: Port<br>UCI: snmpd.trapreceiver[x].port<br>Opt: port | UDP port to be used for sending traps.<br>Range<br>162 |
| Web: Version<br>UCI: snmpd.trapreceiver[x].version<br>Opt: version | SNMP version.<br>v1<br>V2 |
| Web: Community<br>UCI: snmpd.trapreceiver[x].community<br>Opt: community | Community to use in trap messages for this host. |

**Table 121: 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 176: The inform receiver settings page**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Host<br>UCI: snmpd.informreceiver[x].host<br>Opt: host | Host address. Can be either an IP address or an FQDN. |
| Web: Port<br>UCI: snmpd.informreceiver[x].port<br>Opt: port | UDP port to be used for sending traps.<br>Range<br>162 |
| Web: Community<br>UCI: snmpd.informreceiver[x].community<br>Opt: community | Community to use in inform messages for this host. |

**Table 122: 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 177: The interface SNMP Alias ifindex field advanced settings page**

| UCI / Package Option | Description |
|---|---|
| Web: SNMP Alias ifindex<br>UCI: network.@interface[X].snmp_alias_ifindex<br>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*)<br>Blank: No SNMP interface alias index<br>Range: 0 - 4294966295 |
| Web: n/a<br>UCI: network.@interface[X].snmp_alias_ifdescr<br>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)<br>Blank: No SNMP interface alias name<br>Range: |

**Table 123: 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 124: 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 125: 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

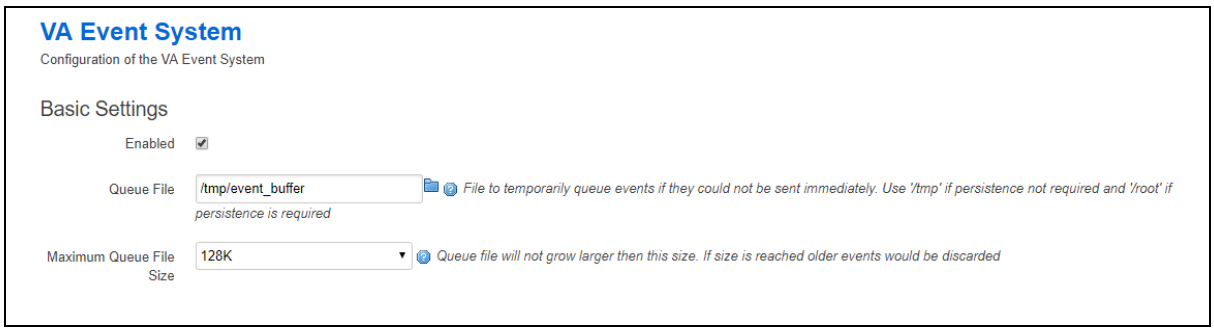



**Figure 178: The VA event system basic settings configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.main.enabled<br>Opt: enabled | Enables VA Event System.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Enabled<br>UCI: va_eventd.main.event_queue_file<br>Opt: event_queue_file | Defines the file to temporarily queue events when they cannot be sent immediately.<br>**Note**: Use `/tmp` path if persistence is not required and `/root` if persistence is required.<br>/tmp/event_buffer — Disabled.<br>1 — Enabled. |
| Web: Enabled<br>UCI: va_eventd.main.event_queue_size<br>Opt: event_queue_size | Defines the file size for the temporary queue. Older events are discarded once file size is reached.<br>128K — 128 Kilobytes<br>Range |

**Table 126: 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. |

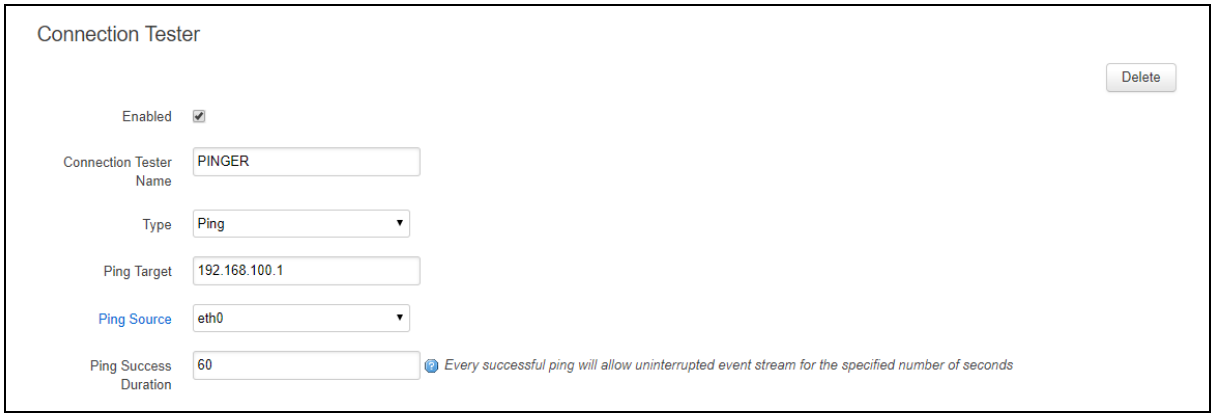



**Figure 179: The VA event system connection tester configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.@conn_tester[0].enabled<br>Opt: enabled | Enables a connection tester.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Connection Tester Name<br>UCI: va_eventd.@conn_tester[0].name<br>Opt: name | Defines the connection tester name.<br>This is used when configuring a connection tester for an event destination. |
| Web: Type<br>UCI: va_eventd.@conn_tester[0].type<br>Opt: type | Defines the connection tester type.<br>**Web Value** — **Description** — **UCI**<br>Ping — Verifies target by ping — ping<br>Link — Verifies target by checking routed interface is up — link |

| | |
|---|---|
| Web: Ping Target<br>UCI: va_eventd.@conn_tester[0].ping_dest_addr<br>Opt: ping_dest_addr | Defines the IP address for the target ping.<br>**Note**: only displayed if connection tester type is set to 'Ping'.<br>[ ] [ ]<br>Range [ ] |
| Web: Ping Source<br>UCI: va_eventd.@conn_tester[0].ping_source<br>Opt: ping_source | Defines an interface or IP address to source the pings from.<br>**Note**: only displayed if connection tester type is set to 'Ping'.<br>eth0 Use eth0 IP for ping source<br>Range [ ] |
| Web: Ping Success Duration<br>UCI: va_eventd.@conn_tester[0].ping_success_duration_sec<br>Opt: ping_success_duration_sec | Defines the duration, in seconds, for which a successful ping defines a connection tester as up.<br>**Note**: only displayed if connection tester type is set to 'Ping'.<br>60 [ ]<br>Range [ ] |
| Web: Link Interface<br>UCI: va_eventd.@conn_tester[0].link_iface<br>Opt: link_iface | Defines the interface to monitor when the connection tester type is set to 'link'. Configured interfaces are listed.<br>**Note**: only displayed if connection tester type is set to 'Link'.<br>[ ] [ ]<br>Range [ ] |

**Table 127: 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.

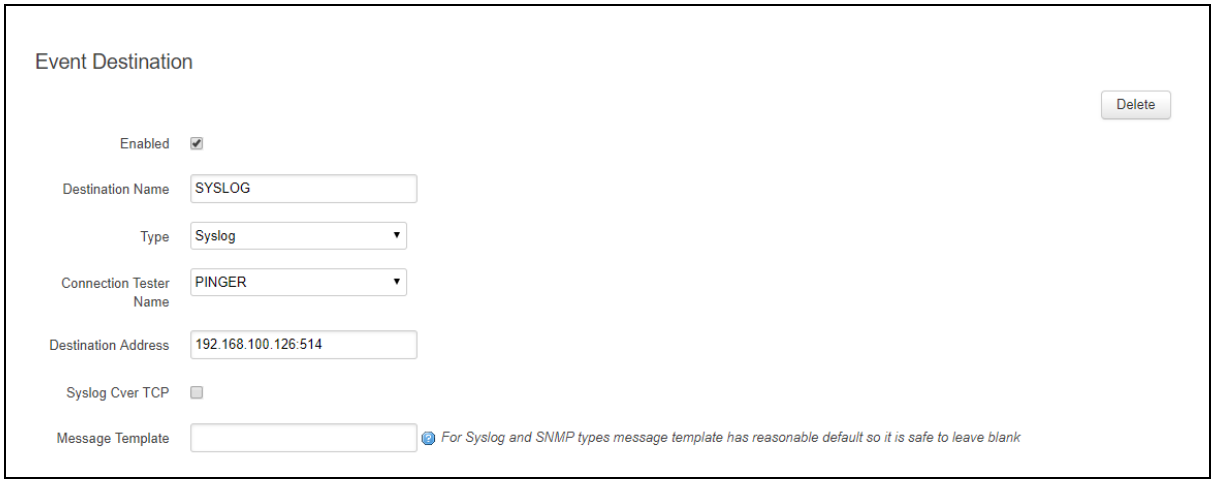



**Figure 180: The VA event system syslog event destination configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.@target[0].enabled<br>Opt: enabled | Enables an event destination. This is used in the event filters section.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Destination name<br>UCI: va_eventd.@target[0].name<br>Opt: name | Defines a name for the event destination.<br>Range |
| Web: Type<br>UCI: va_eventd.@target[0].type<br>Opt: type | Defines the event destination type. For syslog server choose **Syslog**.<br>**Web Value** / **Description** / **UCI**<br>Syslog — syslog<br>SNMP Trap — snmptrap<br>Email — email<br>Execute — exec<br>SMS — sms<br>n/a — File target — file |
| Web: Connection Tester Name<br>UCI: va_eventd.@target[0]. conn_tester<br>Opt: conn_tester | Defines the connection tester (if any) to use to verify the syslog target.<br>None — No connection tester. UCI option not present.<br>Range |
| Web: Destination Address<br>UCI: va_eventd.@target[0]. target_addr<br>Opt: target_addr | Defines the syslog target IP/FQDN and port.<br>Range — a.b.c.d:port or fqdn:port |
| Web: Syslog Over TCP<br>UCI: va_eventd.@target[0].tcp_syslog<br>Opt: tcp_syslog | Defines whether to use TCP for delivery of the syslog event.<br>0 — Use UDP<br>1 — Use TCP |
| Web: Message Template<br>UCI: va_eventd.@target[0].template<br>Opt: template | Defines the message template to use for the event. In general, this should be left empty.<br>See the section on message templates below.<br>Range |

**Table 128: 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.

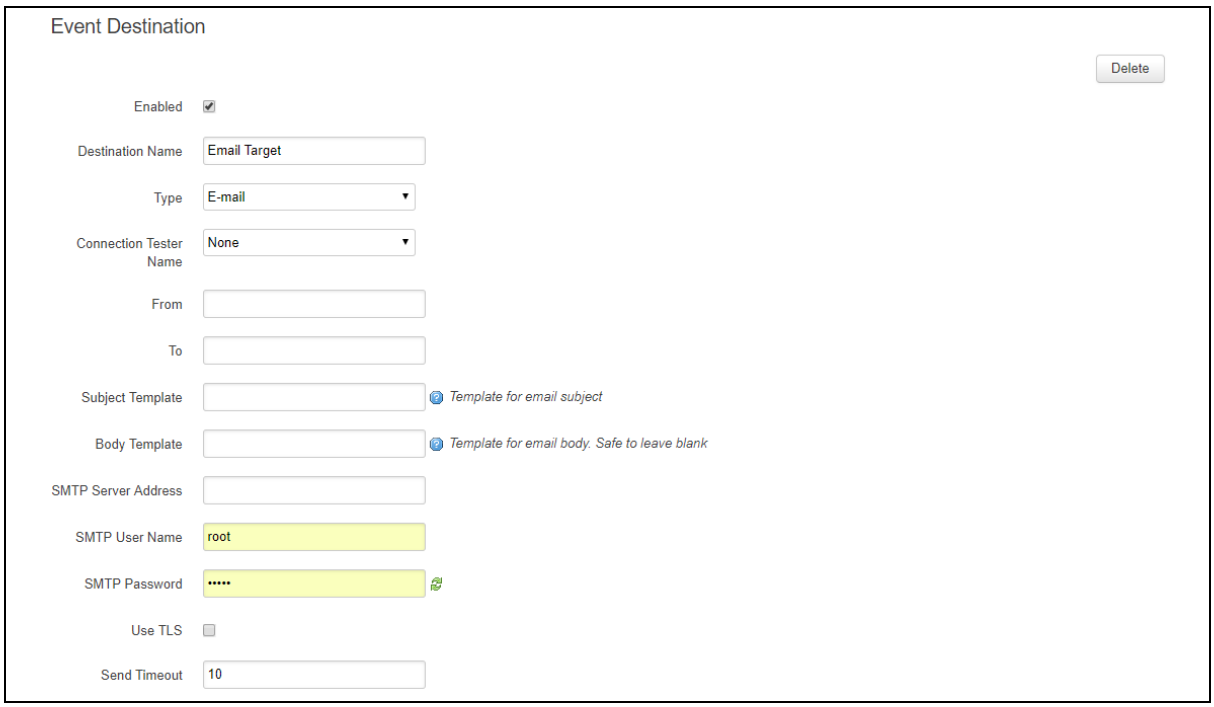



**Figure 181: The VA event system email event destination configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.@target[0].enabled<br>Opt: enabled | Enables an event destination.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Destination name<br>UCI: va_eventd.@target[0].name<br>Opt: name | Defines a name for the event destination.<br>Range |
| Web: Type<br>UCI: va_eventd.@target[0].type<br>Opt: type | Defines the event destination type. For an email server choose **Email**.<br>**Web Value** / **Description** / **UCI**<br>Syslog / Syslog target / syslog<br>SNMP Trap / SNMP target / snmptrap<br>Email / Email target / email<br>Execute / Execure target / exec<br>SMS / SMS target / sms<br>n/a / File target / file |
| Web: Connection Tester Name<br>UCI: va_eventd.@target[0]. conn_tester<br>Opt: conn_tester | Defines the connection tester (if any) to use to verify the email target.<br>None — No connection tester. UCI option not present.<br>Range |
| Web: From<br>UCI: va_eventd.@target[0].from<br>Opt: from | Defines the from address for the email.<br>Range |
| Web: To<br>UCI: va_eventd.@target[0].to<br>Opt: to | Defines to address for the email.<br>Range |

| | |
|---|---|
| Web: Subject Template<br>UCI: va_eventd.@target[0].subject_template<br>Opt: subject_template | Defines subject template for the email. In general, this should be left empty. Example:<br>`va_eventd.@target[0].subject_template="%{severityName} %{eventName}!!!"`<br>See the section on message templates below.<br>Range |
| Web: Body Template<br>UCI: va_eventd.@target[0].body_template<br>Opt: body_template | Defines the email body template. In general, this should be left blank. Example:<br>`va_eventd.@target[0].body_template="%{eventName} (%{class}.%{subclass}) happened!"`<br>See the section on message templates below.<br>Range |
| Web: SMTP Server Address<br>UCI: va_eventd.@target[0].smtp_addr<br>Opt: smtp.addr | Defines the email server address and port.<br>Range: a.b.c.d:port or fqdn:port |
| Web: SMTP User Name<br>UCI: va_eventd.@target[0].smtp_user<br>Opt: smtp_user | Defines user name for SMTP authentication.<br>Range: name@site.com |
| Web: SMTP Password<br>UCI: va_eventd.@target[0].smtp_password<br>Opt: smtp_password | Defines the password for SMTP authentication.<br>Range |
| Web: Use TLS<br>UCI: va_eventd.@target[0].use_tls<br>Opt: use_tls | Enables TLS (Transport Layer Security) support.<br>0<br>1 |
| Web: Send Timeout<br>UCI: va_eventd.@target[0].timeout_sec<br>Opt: timeout_sec | Defines the email send timeout in seconds.<br>10<br>Range |
| Web: Use StartTLS<br>UCI: va_eventd.@target[0]. tls_starttls<br>Opt: tls_starttls | Enables StartTLS support for TLS.<br>(Only displayed when TLS is enabled)<br>0<br>1 |
| Web: Force SSLv3<br>UCI: va_eventd.@target[0].tls_forcessl3<br>Opt: tls_forcessl3 | Enables force SSLv3 for TLS.<br>(Only displayed when TLS is enabled)<br>0<br>1 |

**Table 129: 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.

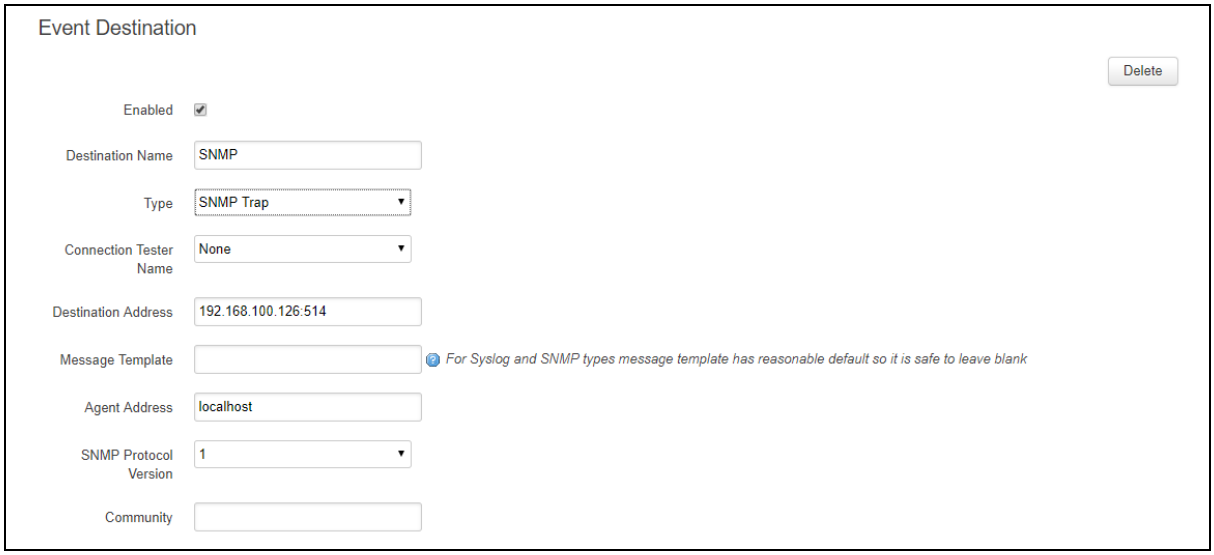



**Figure 182: The VA event system SNMP event destination configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.@target[0].enabled<br>Opt: enabled | Enables an event destination.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Destination name<br>UCI: va_eventd.@target[0].name<br>Opt: name | Defines a name for the event destination.<br>Range |
| Web: Type<br>UCI: va_eventd.@target[0].type<br>Opt: type | Defines the event destination type. For SNMP server choose **SNMP Trap**.<br>**Web Value** / **Description** / **UCI**<br>Syslog / Syslog target / syslog<br>SNMP Trap / SNMP target / snmptrap<br>Email / Email target / email<br>Execute / Execure target / exec<br>SMS / SMS target / sms<br>n/a / File target / file |
| Web: Connection Tester Name<br>UCI: va_eventd.@target[0]. conn_tester<br>Opt: conn_tester | Defines the connection tester (if any) to use to verify the SNMP target.<br>None — No connection tester. UCI option not present.<br>Range |
| Web: Destination Address<br>UCI: va_eventd.@target[0]. target_addr<br>Opt: target_addr | Defines the SNMP target IP/FQDN and port.<br>Range — a.b.c.d:port or fqdn:port |
| Web: Message Template<br>UCI: va_eventd.@target[0].template<br>Opt: template | Defines the message template to use for the event. In general, this should be left empty. Example:<br>`va_eventd.@target[0].template="%{eventName} %{eventSpecificTemplate}"`<br>See the section on message templates below.<br>Range |
| Web: Agent Address<br>UCI: va_eventd.@target[0]. agent_addr<br>Opt: agent_addr | Defines the IP address to source the SNMP trap. (optional)<br>localhost<br>Range |

| | |
|---|---|
| Web: SNMP Protocol Version<br>UCI: va_eventd.@target[0].snmp_version<br>Opt: snmp_version | Defines the SNMP version.<br>1 SNMPv1<br>2c SNMPv2c<br>3 SNMPv3 |
| Web: Community<br>UCI: va_eventd.@target[0].community<br>Opt: community | Defines the community string for SNMPv1.<br>Range |
| Web: Username<br>UCI: va_eventd.@target[0].snmp_uname<br>Opt: snmp_uname | Defines the username for SNMPv3.<br>(Only displayed when SNMP protocol version is SNMPv3)<br>Range |
| Web: Authentication Protocol<br>UCI: va_eventd.@target[0].snmp_auth_proto<br>Opt: snmp_auth_proto | Defines the SNMPv3 authentication protocol<br>(Only displayed when SNMP protocol version is SNMPv3)<br>MD5<br>SHA |
| Web: Authentication Password<br>UCI: va_eventd.@target[0].snmp_auth_pass<br>Opt: snmp_auth_pass | Defines the SNMPv3 authentication password<br>(Only displayed when SNMPv3 authentication protocol is configured)<br>MD5<br>SHA |
| Web: Privacy Protocol<br>UCI: va_eventd.@target[0].snmp_priv_proto<br>Opt: snmp_priv_proto | Defines the SNMPv3 privacy protocol.<br>(Only displayed when SNMP authentication protocol is configured)<br>DES<br>AES |
| Web: Privacy Password<br>UCI: va_eventd.@target[0].snmp_priv_pass<br>Opt: snmp_priv_pass | Defines SNMPv3 privacy password.<br>(Only displayed when SNMP privacy protocol is configured)<br>Range |
| Web: SNMPv3 Context<br>UCI: va_eventd.@target[0].snmp_context<br>Opt: snmp_context | Defines the SNMPv3 context.<br>(Only displayed when SNMP authentication protocol is configured)<br>Range |
| Web: SNMPv3 Context Engine ID<br>UCI: va_eventd.@target[0].snmp_context_eid<br>Opt: snmp_context_eid | Defines the SNMPv3 context engine ID.<br>(Only displayed when SNMP authentication protocol is configured)<br>Range |
| Web: SNMPv3 Security Engine ID<br>UCI: va_eventd.@target[0].snmp_sec_eid<br>Opt: snmp_sec_eid | Defines the SNMPv3 security engine ID.<br>(Only displayed when SNMP authentication protocol is configured)<br>Range |

**Table 130: 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 183: The VA event system Exec event destination configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.@target[0].enabled<br>Opt: enabled | Enables an event destination.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Destination name<br>UCI: va_eventd.@target[0].name<br>Opt: name | Defines a name for the event destination.<br>Range |
| Web: Type<br>UCI: va_eventd.@target[0].type<br>Opt: type | Defines the event destination type. For shell command execution, choose **Execute**.<br>**Web Value** / **Description** / **UCI**<br>Syslog / Syslog target / syslog<br>SNMP Trap / SNMP target / snmptrap<br>Email / Email target / email<br>Execute / Execure target / exec<br>SMS / SMS target / sms<br>n/a / File target / file |
| Web: Connection Tester Name<br>UCI: va_eventd.@target[0]. conn_tester<br>Opt: conn_tester | Defines the connection tester (if any) to use to verify the execute target.<br>None — No connection tester. UCI option not present.<br>Range |
| Web: Command Template<br>UCI: va_eventd.@target[0].cmd_template<br>Opt: cmd_template | Defines the command template to use for the event.<br>Example to log a syslog message:<br>`va_eventd.@target[0].cmd_template="logger -t eventer %{eventName}"`<br>See the section on message templates below.<br>Range |

**Table 131: 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.

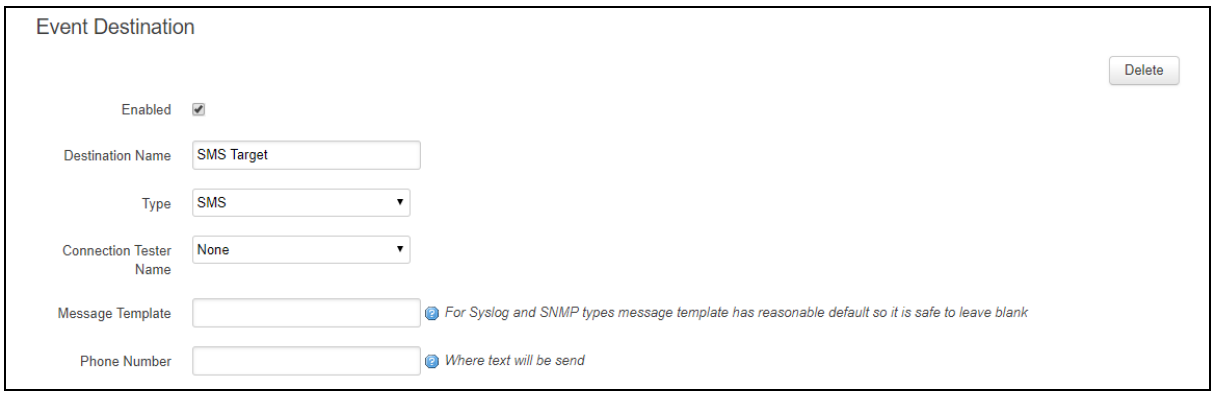



**Figure 184: The VA event system SMS event destination configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.@target[0].enabled<br>Opt: enabled | Enables an event destination.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Destination name<br>UCI: va_eventd.@target[0].name<br>Opt: name | Defines a name for the event destination.<br>Range |
| Web: Type<br>UCI: va_eventd.@target[0].type<br>Opt: type | Defines the event destination type. For SMS destination choose **SMS**.<br>Syslog — syslog<br>SNMP Trap — snmptrap<br>Email — email<br>Execute — exec<br>SMS — sms<br>n/a — file |
| Web: Connection Tester Name<br>UCI: va_eventd.@target[0]. conn_tester<br>Opt: conn_tester | Defines the connection tester (if any) to use to verify the SMS target.<br>None — No connection tester. UCI option not present.<br>Range |
| Web: Message Template<br>UCI: va_eventd.@target[0].template<br>Opt: template | Defines the message template to use for the event. In general, this should be left empty. Example:<br>`va_eventd.@target[0].template="%{eventName}"`<br>See the section on message templates below.<br>Range |
| Web: Phone Number<br>UCI: va_eventd.@target[0].callee<br>Opt: callee | Defines the phone number for sending SMS to.<br>Range |

**Table 132: 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<br>UCI: va_eventd.@target[0].enabled<br>Opt: enabled | Enables an event destination.<br>0 — Disabled.<br>1 — Enabled. |
| Web: n/a<br>UCI: va_eventd.@target[0].name<br>Opt: name | Defines a name for the event destination.<br>Range |
| Web: n/a<br>UCI: va_eventd.@target[0].type<br>Opt: type | Defines the event destination type. For file choose **Syslog**.<br>**Web Value** / **Description** / **UCI**<br>Syslog — syslog<br>SNMP Trap — snmptrap<br>Email — email<br>Execute — exec<br>SMS — sms<br>n/a — file |
| Web: n/a<br>UCI: va_eventd.@target[0].file_name<br>Opt: file_name | Defines a file name for the event destination. (full path)<br>Range |
| Web: n/a<br>UCI: va_eventd.@target[0].max_size_kb<br>Opt: file_name | Defines a file size in kilobits.<br>2048<br>Range |
| Web: n/a<br>UCI: va_eventd.@target[0].template<br>Opt: template | Defines the message template to use for the event. In general, this should be left empty.<br>See the section on message templates below.<br>Range |

**Table 133: 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.

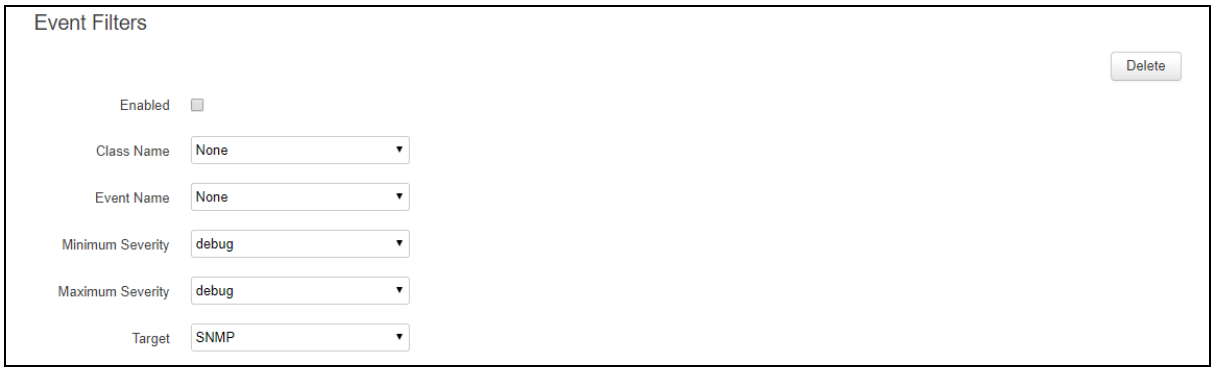



**Figure 185: The VA event system event filters configuration page**

| Web Field/UCI/Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: va_eventd.@forwarding[0].enabled<br>Opt: enabled | Enables an event filter.<br>1 — Disabled.<br>0 — Enabled. |
| Web: Class Name<br>UCI: va_eventd.@forwarding[0].className<br>Opt: className | Only match events with the given class name. Available class names are listed or can be viewed using the command `vae_cli -d` |
| Web: Event Name<br>UCI: va_eventd.@forwarding[0].eventName<br>Opt: eventName | Only match events with the given event name. Available event names are listed. The event name is optional and can be omitted. |
| Web: Minimum Severity<br>UCI: va_eventd.@forwarding[0].severity<br>Opt: severity | Defines the minimum event severity. The minimum severity event is DEBUG. Events generated within the minimum and maximum event severity will be matched.<br>Minimum and maximum severity are specified in the one UCI option and entered using a dash (-) separator in the form minimum-maximum. Example:<br>`va_eventd.@forwarding[0].severity=debug-error`<br>debug — minimum severity<br>info<br>notice<br>warning<br>error<br>critical<br>alert<br>emergency — maximum severity |
| Web: Maximum Severity<br>UCI: va_eventd.@forwarding[0].severity<br>Opt: severity | Defines the maximum event severity. The maximum event severity is EMERGENCY. Events generated within the minimum and maximum event severity will be matched.<br>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:<br>`va_eventd.@forwarding[0].severity=debug-error`<br>debug — minimum severity<br>info<br>notice<br>warning<br>error<br>critical<br>alert<br>emergency — maximum severity |
| Web: Target<br>UCI: va_eventd.@forwarding[0].target<br>Opt: target | Defines the event destination to forward the event to. All configured event destinations will be displayed.<br>Range |

**Table 134: 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.

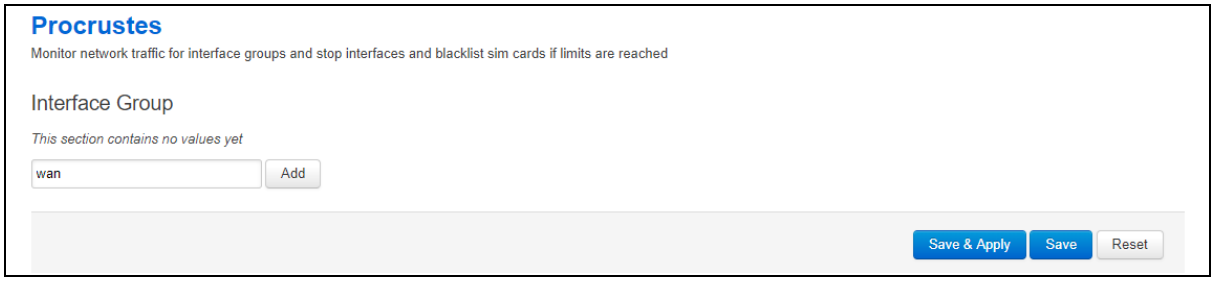



**Figure 186: The data usage page**

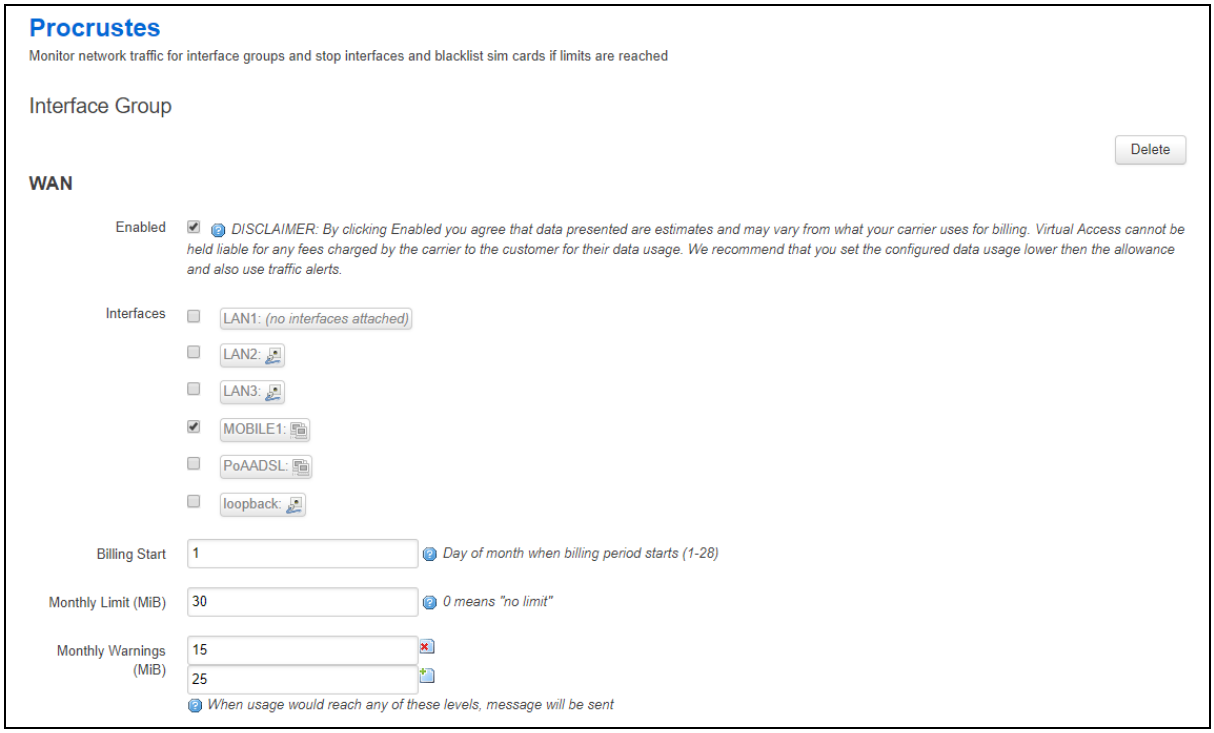



**Figure 187: The data usage configuration page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enabled<br>UCI: procrustes.@limit[0].enabled<br>Opt: enabled | Enable Data Usage monitor on this interface group.<br>0 – Disabled.<br>1 – Enabled. |
| Web: Billing Start<br>UCI: procrustes.@limit[0].billing_period_start_day<br>Opt: billing_period_start_day | Day of month on which the billing period starts.<br>1<br>Range: 1 – 28 |
| Web: Interfaces<br>UCI: procrustes.@limit[0].interfaces<br>Opt: interfaces | Monitor and apply limits to these interfaces as a group. Configure multiple interfaces via UCI using a space separator. Example:<br>`uci set procrustes.@limit[0].interfaces="lan wan"` |
| Web: Monthly Limit (MiB)<br>UCI: procrustes..@limit[0].monthly_data_limit<br>Opt: monthly_data_limit | Defines monthly data traffic limit in mebibytes (MiB). This is total RX and TX on the interface.<br>0 – Zero means no limit.<br>Range |
| Web: Monthly Warnings (MiB)<br>UCI: procrustes.@limit[0].monthly_warning_levels<br>Opt: monthly_warning_levels | 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.<br>Set multiple limits via UCI using a space separator. Example:<br>`uci set procrustes.@limit[0].monthly_warning_levels="15 25"`<br>0 – Zero means no limit.<br>Range |

**Table 135: 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 188: 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

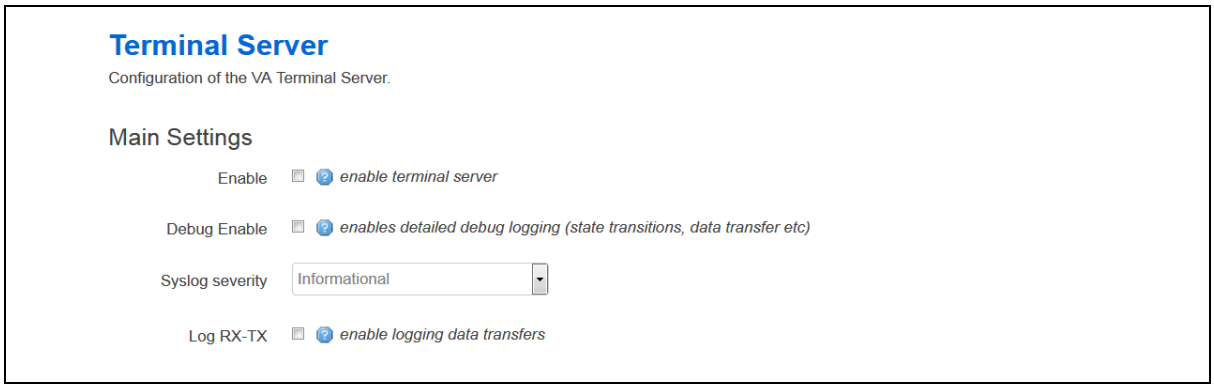



**Figure 189: The terminal server main settings page**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable<br>UCI: tservd.main.enable<br>Opt: enable | Enables Terminal Server on the router.<br>0 Disabled.<br>1 Enabled. |
| Web: Debug Enable<br>UCI: tservd.main.debug_ev_enable<br>Opt: debug_ev_enable | Enables detailed debug logging.<br>0 Disabled.<br>1 Enabled. |
| Web: Syslog severity<br>UCI: tservd.main.log_severity<br>Opt: log_severity | Determines the syslog level. Events up to this priority will be logged.<br>0 Emergency<br>1 Alert<br>2 Critical<br>3 Error<br>4 Warning<br>5 Notice<br>6 Informational<br>7 Debug |
| Web: Log RX-TX<br>UCI: tservd.main.debug_rx_tx_enable<br>Opt: debug_rx_tx_enable | Enables logging data transfers.<br>0 Disabled.<br>1 Enabled. |

**Table 136: 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.

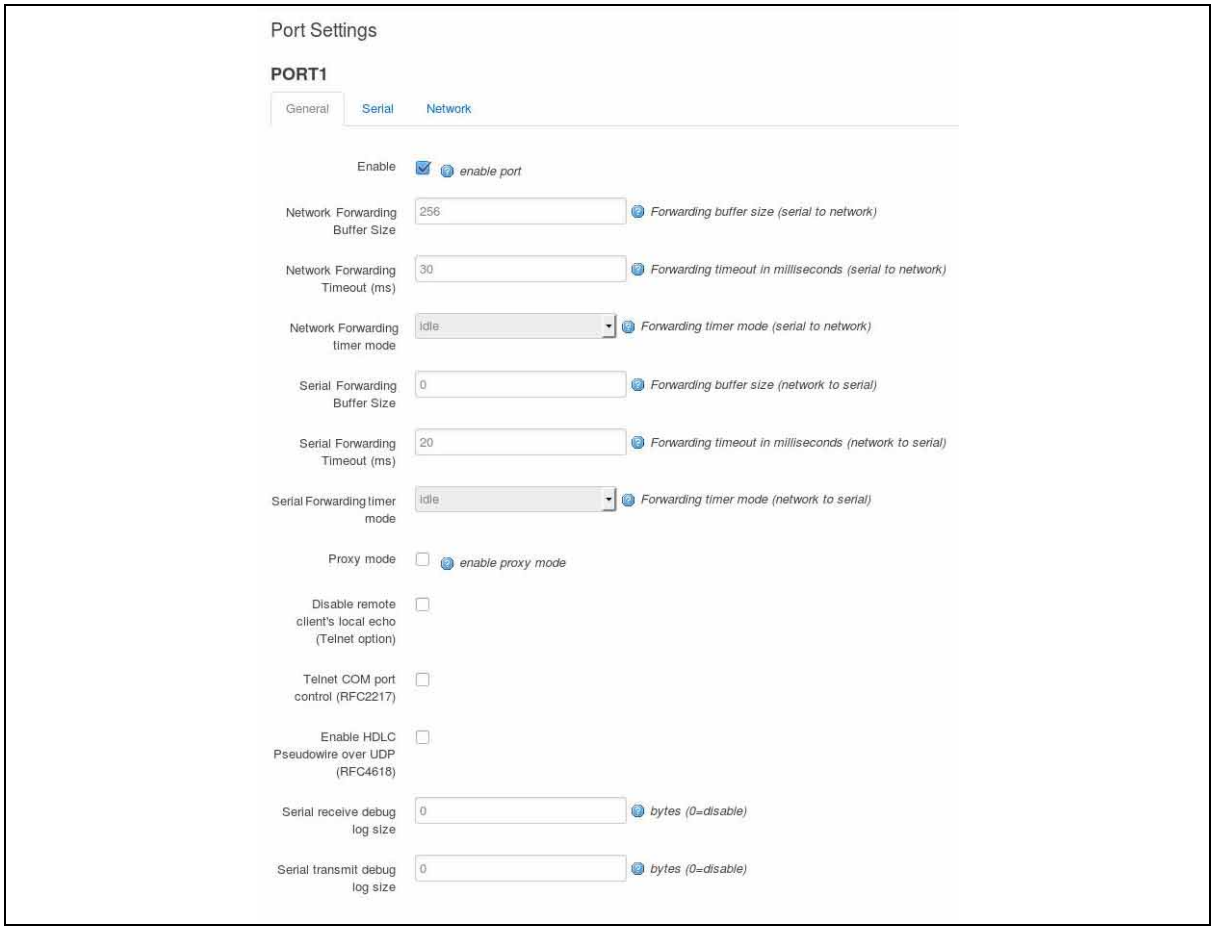



**Figure 190: The general tab fields**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Enable<br>UCI: tservd.@port[0].enable<br>Opt: enable | Enables Terminal Server port.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Network Forwarding Buffer Size<br>UCI: tservd.@port[0].fwd_buffer_size<br>Opt: fwd_buffer_size | Forwarding buffer size in bytes (serial to network).<br>256 — 256 bytes<br>Range — 0-2048 |
| Web: Network Forwarding Timeout(ms)<br>UCI: tservd.@port[0].fwd_timeout<br>Opt: fwd_timeout | Forwarding timeout in milliseconds (serial to network).<br>30 — 30 ms<br>Range — 0-10000 |
| Web: Network Forwarding Timer Mode<br>UCI: tservd.@port[0].fwd_timer_mode<br>Opt: fwd_timer_mode | Forwarding timer mode (serial to network).<br>Idle — Timer is re-started on each received data.<br>Aging — Timer started on the first Rx. |
| Web: Serial Forwarding Buffer Size<br>UCI: tservd.@port[0].sfwd_buffer_size<br>Opt: sfwd_buffer_size | Forwarding buffer size in bytes (network to serial).<br>Set to 0 to use maximum possible network Rx buffer size.<br>0 — 2048 bytes<br>Range — 0-2048 |
| Web: Serial Forwarding Timeout (ms)<br>UCI: tservd.@port[0].sfwd_timeout<br>Opt: sfwd_timeout | Forwarding timeout in milliseconds (network to serial).<br>Set to 0 to forward to serial immediately.<br>20 — 20 ms<br>Range — 0-10000 |

| | |
|---|---|
| Web: Serial Forwarding Timer Mode<br>UCI: tservd.@port[0].sfwd_timer_mode<br>Opt: sfwd_timer_mode | Forwarding timer mode (network to serial).<br>Idle — Timer is re-started on each received data<br>Aging — Timer started on the first Rx. |
| Web: Proxy Mode<br>UCI: tservd.@port[0].proxy_mode<br>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.<br>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:<br>The existing terminal server TCP client connection is disconnected.<br>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.<br>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.<br>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.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Disable Remote Client's Local Echo (Telnet option)<br>UCI: tservd.@port[0].disable_echo<br>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.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Telnet COM Port Control<br>UCI: tservd.@port[0].com_port_control<br>Opt: com_port_control | Set to **1** to enable support for Telnet COM port control (RFC2217).<br>0 — Disabled.<br>1 — Enabled. |
| Web: Enable HDLC Pseudowire over UDP (RFC4618)<br>UCI: tservd.@port[0].hdlc_pw_enabled<br>Opt: hdlc_pw_enabled | Set to **1** to enable HDLC pseudowire over UDP support based on RFC4618. Requires Transport Mode (udpmode) to be enabled.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Serial Receive Debug Log Size<br>UCI: tservd.@port[0].serialRxLogSize<br>Opt: serialRxLogSize | Configures serial receive log size in bytes and enables receive data logging.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Serial Transmit Debug Log Size<br>UCI: tservd.@port[0].serialTxLogSize<br>Opt: serialTxLogSize | Configures serial transmit log size in bytes and enables transmit data logging.<br>0 — Disabled.<br>1 — Enabled. |

**Table 137: 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.

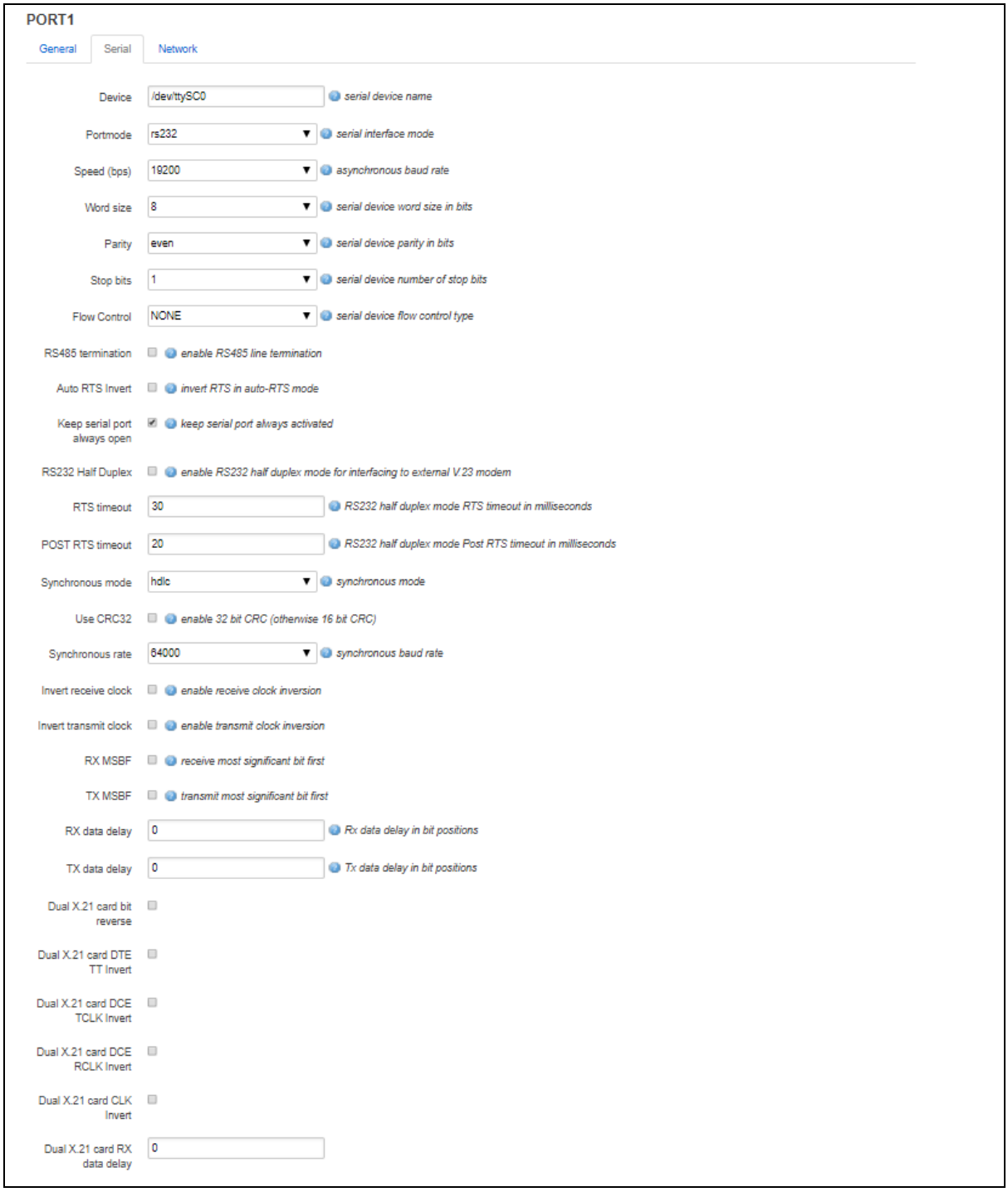



**Figure 191: The serial section fields (port mode RS232)**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Device<br>UCI: tservd.@port[0].devName<br>Opt: devName | Serial device name.<br>/dev/ttySC0 — serial port 1<br>/dev/ttySC1 — serial port 2<br>/dev/ttySC2 — serial port 3<br>/dev/ttySC3 — serial port 4 |
| Web: Port mode<br>UCI: tservd.@port[0].port_mode<br>Opt: port_mode | Sets the serial interface mode.<br>rs232 — RS232 mode.<br>rs485hdx — RS485 2 wire half duplex mode in which transmitter drives RTS.<br>rs485fdx — Rs485 4 wire full duplex mode.<br>v23 — Uses V.23 leased line card driver.<br>x21 — Uses USB serial card in sync mode. |
| Web: Speed (bps)<br>UCI: tservd.@port[0].speed<br>Opt: speed | Serial device speed in baud (bps).<br>9600<br>Range — 115200; 57600; 38400; 19200; 9600 4800; 2400; 1800; 1200; 600; 300; 200; 150; 134; 110; 75; 50 |
| Web: Word size<br>UCI: tservd.@port[0].wsize<br>Opt: wsize | Serial device word size.<br>8<br>Range — 5-8 |
| Web: Parity<br>UCI: tservd.@port[0].parity<br>Opt: parity | Serial device parity.<br>0 — None<br>1 — Even<br>2 — Odd<br>3 — Space |
| Web: Stop Bits<br>UCI: tservd.@port[0].stops<br>Opt: stops | Serial device number of stop bits.<br>1<br>Range — 1-2 |
| Web: Flow Control<br>UCI: tservd.@port[0].fc_mode<br>Opt: fc_mode | Serial flow control mode.<br>0 — None<br>1 — RTS/CTS<br>2 — XON/XOFF |
| Web: RS485 Termination<br>UCI: tservd.@port[0].rs485_line_termination<br>Opt: rs485_line_termination | Enables or disable RS485 termination. Applies only if port mode is set to RS485.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Auto RTS Invert<br>UCI: tservd.@port[0].rtsinvert<br>Opt: rtsinvert | Invert RTS in auto-RTS mode, if port mode is set to RS485.<br>0 — Disabled.<br>1 — Enabled. |
| Web: Keep Serial Port Always Open<br>UCI: tservd.@port[0].tty_always_open<br>Opt: tty_always_open | Keep serial port always open.<br>0 — Disabled.<br>1 — Enabled. |
| Web: RS232 Half Duplex<br>UCI: tservd.@port[0].hd_mode<br>Opt: hd_mode | 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.<br>0 — Full duplex mode.<br>1 — Half duplex mode. |
| Web: RTS Timeout<br>UCI: tservd.@port[0].rts_timeout<br>Opt: rts_timeout | In RS232 half-duplex mode, time in milliseconds between raising RTS and enabling the transmitter. For use with an externally connected V.23 modem.<br>30 — 30ms<br>Range |

| | |
|---|---|
| Web: POST RTS Timeout<br>UCI: tservd.@port[0].post_rts_timeout<br>Opt: post_rts_timeout | 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.<br>20 — 20 ms<br>Range |
| Web: Synchronous mode<br>UCI: tservd.@port[0].sync mode<br>Opt: sync mode | Defines synchronous frame mode. This setting is only displayed if an Atmel USB serial card is enabled.<br>hdlc — HDLC frame mode.<br>transp — Transparent mode. |
| Web: Use CRC32<br>UCI: tservd.@port[0].sync_crc32<br>Opt: sync_crc32 | Defines whether to use CRC32 or CRC16 in HDLC mode. This setting is only displayed if an Atmel USB serial card is enabled.<br>0 — Use CRC16.<br>1 — Use CRC32. |
| Web: DTR control mode<br>UCI: tservd.@port[0].dtr_control_mode<br>Opt: dtr_control_mode | Defines DTR line control modes. This setting is only displayed if an Atmel USB serial card is enabled and port mode is X21.<br>auto — DTR set to on when port is open. Off when the port is closed.<br>on — DTR always on.<br>off — DTR always off.<br>app — DTR controlled by the application.<br>ontx — In HDLC mode DTR is on during frame transmission. |
| Web: RTS control mode<br>UCI: tservd.@port[0].rts_control_mode<br>Opt: rts_control_mode | Defines RTS line control modes. Only displayed if Atmel USB serial card is enabled and port mode is X21.<br>auto — RTS set to on when port is open. Off when the port is closed.<br>on — RTS always on.<br>off — RTS always off.<br>app — RTS controlled by the application.<br>ontx — In HDLC mode RTS is on during frame transmission. |
| Web: Synchronous rate<br>UCI: tservd.@port[0].sync_speed<br>Opt: sync_speed | 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.<br>64000 — 64 kbps<br>Range — 2048000; 1024000; 768000; 512000; 384000; 256000; 128000; 19200; 9600 |
| Web: Invert receive clock<br>UCI: tservd.@port[0].sync_invert_rxclk<br>Opt: sync_invert_rxclk | 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.<br>0 — Normal.<br>1 — Invert. |
| Web: Invert transmit clock<br>UCI: tservd.@port[0].sync_invert_txclk<br>Opt: sync_invert_txclk | 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.<br>0 — Normal.<br>1 — Invert. |
| Web: RX MSBF<br>UCI: tservd.@port[0].sync_rx_msbf<br>Opt: sync_rx_msbf | Defines whether most significant bit is received first. This setting is only displayed if an Atmel USB serial card is enabled.<br>0 — Receive least significant bit first.<br>1 — Receive most significant bit first. |
| Web: TX MSBF<br>UCI: tservd.@port[0].sync_tx_msbf<br>Opt: sync_tx_msbf | Defines whether most significant bit is transmitted first. This setting is only displayed if an Atmel USB serial card is enabled.<br>0 — Transmit least significant bit first.<br>1 — Transmit most significant bit first. |

| | |
|---|---|
| Web: RX data delay<br>UCI: tservd.@port[0].sync_rxdata_dly<br>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.<br>0 \| <br>Range \| |
| Web: TX data delay<br>UCI: tservd.@port[0].sync_txdata_dly<br>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.<br>0 \| <br>Range \| |
| Web: Dual X.21 card bit reverse<br>UCI: tservd.@port[0].bit_reverse<br>Opt: bit_reverse | Enables bit reversal of all bits in 8 byte word during transmission.<br>0 \| Normal.<br>1 \| Reverse. |
| Web: Dual X.21 card DTE TT Invert<br>UCI: tservd.@port[0].dte_tt_inv<br>Opt: dte_tt_inv | Enables X.21 TT clock signal inversion.<br>0 \| Normal.<br>1 \| Invert. |
| Web: Dual X.21 card DCE TCLK Invert<br>UCI: tservd.@port[0].dce_tclk_inv<br>Opt: dce_tclk_inv | Enables X.21 DCE TCLK signal inversion.<br>0 \| Normal.<br>1 \| Invert. |
| Web: Dual X.21 card DCE RCLK Invert<br>UCI: tservd.@port[0].dce_rclk_inv<br>Opt: dce_rclk_inv | Enables X.21 DCE RCLK signal inversion.<br>0 \| Normal.<br>1 \| Invert. |
| Web: Dual X.21 card CLK Invert<br>UCI: tservd.@port[0].x21_clk_invert<br>Opt: x21_clk_invert | Enables X.21 DCE CLK signal inversion.<br>0 \| Normal.<br>1 \| Invert. |
| Web: Dual X.21 card RX data delay<br>UCI: tservd.@port[0] x21_data_delay<br>Opt: x21_data_delay | Sets X.21 card RX data delay in number of bit positions.<br>0 \| <br>Range \| 0 – 7 |
| Web: n/a<br>UCI: tservd.@port[0].sync_tx_idle<br>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.<br>0 \| Tranmit 0 (in HDLC mode)<br>126 \| Transmit flags (in HDLC mode)<br>255 \| Tranmit 1 (in HDLC mode)<br>Range \| 0 – 255 |
| Web: n/a<br>UCI: tservd.@port[0].v23_inband_carrier_signalling<br>Opt: v23_inband_carrier_signalling | Enables signalling of carrier by sending special characters.<br>0 \| Disabled.<br>1 \| Enabled. |
| Web: n/a<br>UCI: tservd.@port[0].v23_inband_carrier_on_char<br>Opt: v23_inband_carrier_on_char | Defines the character decimal to signal remote carrier on.<br>255 \| <br>Range \| 0 - 255 |
| Web: n/a<br>UCI: tservd.@port[0].v23_tx_gain<br>Opt: v23_tx_gain | Defines the transmit gain for v23 mode.<br>2 \| Transmit samples multiplied by 2<br>Range \| |
| Web: n/a<br>UCI: tservd.@port[0].v23_rx_loss<br>Opt: v23_rx_loss | Defines the receive loss for v23 mode.<br>1 \| Receive samples divided by 1.<br>Range \| |
| Web: n/a<br>UCI: tservd.@port[0].v23_rts_to_cts_delay<br>Opt: v23_rts_to_cts_delay | Defines the v23 modem RTS to CTS delay in milliseconds.<br>20 \| <br>Range \| |

| | |
|---|---|
| Web: n/a<br>UCI: tservd.@port[0].v23_is_four_wire<br>Opt: v23_is_four_wire | Defines the V23 modem LIM operation.<br>0 — 2-wire<br>1 — 4-wire |
| Web: n/a<br>UCI: tservd.@port[0].v23_tx_timeout<br>Opt: v23_tx_timeout | Defines the V23 modem receive echo suppression timeout in milliseconds.<br>20<br>Range |
| Web: n/a<br>UCI: tservd.@port[0].v23_tx_rampdown<br>Opt: v23_tx_rampdown | Defines the time in milliseconds it takes the V23 transmitter to rampdown carrier from peak to zero.<br>30<br>Range |
| Web: n/a<br>UCI: tservd.@port[0].v23_tx_maxfill<br>Opt: v23_tx_maxfill | Defines the maximum transmit queue fill level in bytes.<br>127<br>Range — 0 - 255 |

**Table 138: 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.

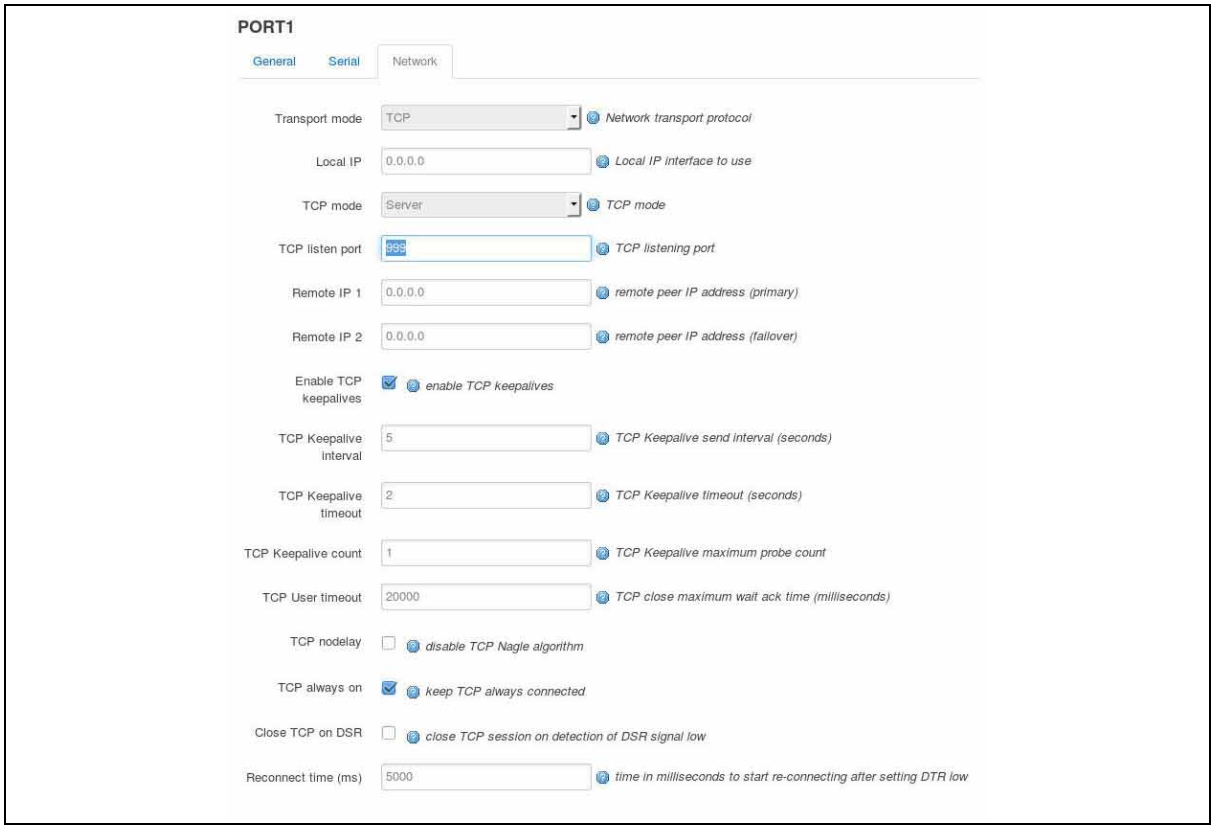



**Figure 192: The port settings network fields (TCP server mode)**

| Web Field / UCI / Package Option | Description |
|---|---|
| Web: Transport Mode<br>UCI: tservd.@port[0].udpMode<br>Opt: udpMode | Selects the transport mode.<br>0 — TCP<br>1 — UDP |
| Web: Local IP<br>UCI: tservd.@port[0].local_ip<br>Opt: local_ip | Local IP address to listen on.<br>0.0.0.0 — Listen on any interface.<br>Range — IPv4 address. |
| Web: TCP Mode<br>UCI: tservd.@port[0].server_mode<br>Opt: server_mode | Select between server and client modes of TCP. Only displayed if Transport Mode is TCP.<br>0 — Client Mode.<br>1 — Server Mode. |
| Web: TCP Listen Port<br>UCI: tservd.@port[0].listen_port<br>Opt: listen_port | TCP listen port for server mode. Only displayed if Transport Mode is TCP and server mode is enabled.<br>999<br>Range — 1 - 65535 |
| Web: Remote TCP Port 1<br>UCI: tservd.@port[0].ip_port1<br>Opt: ip_port1 | Destination peer port IP 1 number. Only displayed if client mode is enabled.<br>951<br>Range — 1 - 65535 |
| Web: Remote TCP Port 2<br>UCI: tservd.@port[0].ip_port2<br>Opt: ip_port2 | Destination peer port IP 2 number for failover. Only displayed if client mode is enabled.<br>951<br>Range — 1 - 65535 |
| Web: Remote IP 1<br>UCI: tservd.@port[0].remote_ip1<br>Opt: remote_ip1 | Destination peer IP 1 address.<br>0.0.0.0<br>Range — IPv4 address. |
| Web: Remote IP 2<br>UCI: tservd.@port[0].remote_ip2<br>Opt: remote_ip2 | Destination peer IP 2 address for failover.<br>0.0.0.0<br>Range — IPv4 address. |
| Web: Enable TCP Keepalives<br>UCI: tservd.@port[0].tcp_keepalives_enabled<br>Opt: tcp_keepalives_enabled | Enable or disables TCP keepalives. Only displayed if Transport Mode is TCP.<br>0 — Disabled.<br>1 — Enabled. |
| Web: TCP Keepalive Interval<br>UCI: tservd.@port[0].tcp_keepalive_interval<br>Opt: tcp_keepalive_interval | Interval in seconds between TCP keepalive probes. Only displayed if Transport Mode is TCP.<br>5 — 5 seconds.<br>Range — 0-65535 |
| Web: TCP Keepalive Timeout<br>UCI: tservd.@port[0].tcp_keepalive_timeout<br>Opt: tcp_keepalive_timeout | Time in seconds to wait for response to a TCP keepalive probe. Only displayed if Transport Mode is TCP.<br>2 — 2 seconds.<br>Range — 0-65535 |
| Web: TCP Keepalive Count<br>UCI: tservd.@port[0].tcp_keepalive_count<br>Opt: tcp_keepalive_count | Number of TCP keepalive probes to send before connection is closed. Only displayed if Transport Mode is TCP.<br>1<br>Range — 0-65535 |
| Web: TCP User Timeout<br>UCI: tservd.@port[0].tcp_user_timeout<br>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.<br>20000 — 20 seconds.<br>Range — 0-65535 |

| | |
|---|---|
| Web: TCP Nodelay<br>UCI: tservd.@port[0].tcp_nodelay<br>Opt: tcp_nodelay | Sets TCP to delay behaviour. Only displayed if Transport Mode is TCP.<br>0 – Normal operation.<br>1 – Disable TCP Nagle algorithm. Only displayed if Transport Mode is TCP. |
| Web: TCP Always on<br>UCI: tservd.@port[0].tcp_always_on<br>Opt: tcp_always_on | Keep TCP session always connected. Only displayed if Transport Mode is TCP and client mode is enabled.<br>0 – Disabled. TCP connection / UDP session is initiated on detecting high state on the DSR interface signal.<br>1 – Enabled. If it disconnects in the established state the TCP connection / UDP session is re-initiated. |
| Web: Close TCP on DSR<br>UCI: tservd.@port[0].close_tcp_on_dsr<br>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.<br>0 – Disabled. Detecting DSR down does not affect the TCP connection.<br>1 – Enabled. Detecting DSR down closes the established TCP connection. |
| Web: Reconnect Time (ms)<br>UCI: tservd.@port[0].disc_time_ms<br>Opt: disc_time_ms | Time in milliseconds to start reconnecting after setting DTR low.<br>5000 – 5 seconds.<br>Range – 0 – 10000 |
| Web: UDP Keepalive Interval<br>UCI: tservd.@port[0].udpKaIntervalMs<br>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.<br>0 – Disabled.<br>Range – 0-65535 |
| Web: UDP Keepalive Count<br>UCI: tservd.@port[0].udpKaCount<br>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.<br>3<br>Range – 0-65535 |
| Web: local UDP Port<br>UCI: tservd.@port[0].udpLocalPort<br>Opt: udpLocalPort | Local UDP port used by terminal server. Only displayed if transport mode is UDP.<br>0<br>Range – 0-65535 |
| Web: remote UDP Port<br>UCI: tservd.@port[0].udpRemotePort<br>Opt: udpRemotePort | Remote UDP port used by terminal server. Only displayed if transport mode is UDP.<br>0<br>Range – 0-65535 |

**Table 139: 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<br>UCI: terminal.console.enabled<br>Opt: enabled | Enables Terminal on the router.<br>0 – Disabled.<br>1 – Enabled. |
| Web: n/a<br>UCI: terminal.console.device<br>Opt: device | String value point at the tty device in /dev folder.<br>None – Default.<br><string> – Device name.( e.g. ttySC0 to use serial port 0) |
| Web: n/a<br>UCI: terminal.console.speed<br>Opt: speed | Set the speed of serial connection.<br>115200 – Default.<br><range> – Supported port speed. |
| Web: n/a<br>UCI: terminal.console.type<br>Opt: type | String value represents supported terminal emulation mode.<br>vt100 – Default.<br><string> – Supported terminal type. |
| Web: n/a<br>UCI: terminal.console.flowcontrol<br>Opt: flowcontrol | Enables hardware flow control RTS/CTS.<br>0 – Disabled.<br>1 – Enabled. |

**Table 140: 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 Serial interface

## 41.1 Overview

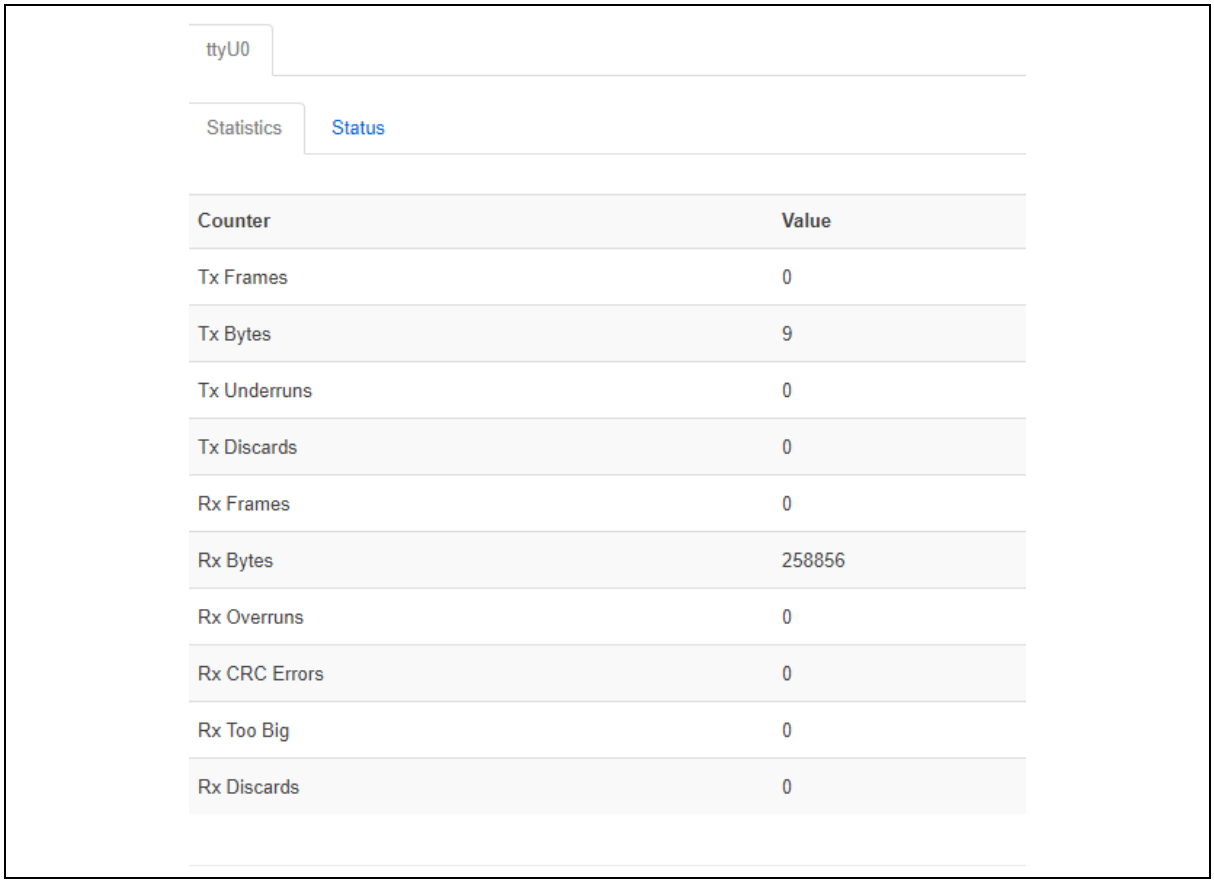
Many different applications and device drivers use the serial interface. You configure the serial interface using the relevant application; for example Terminal Server, therefore there is no standalone serial configuration page.

You can monitor the various serial interfaces using either the command line or the web interface.

## 41.2 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.

### 41.2.1 Serial statistics



ttyU0

Statistics | Status

| Counter | Value |
|---|---|
| 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 |

**Figure 193: The serial statistics page for serial-0**

### 41.2.2 Serial status

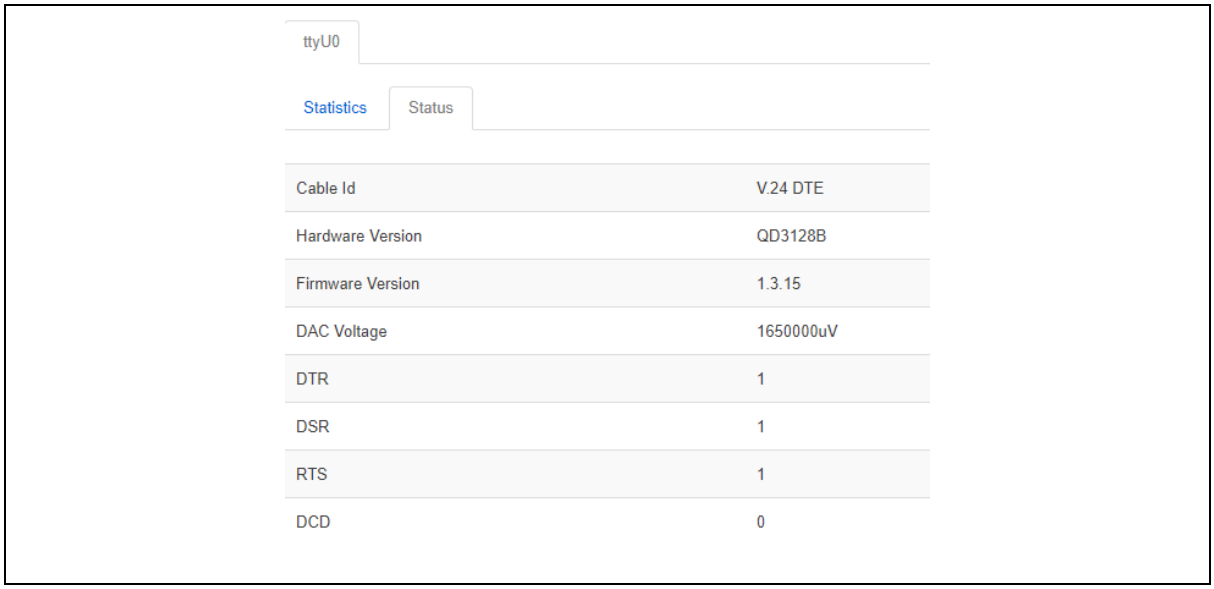



**Figure 194: The serial status page for serial-0**

## 41.3 Monitoring serial interfaces using command line

### 41.3.1 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
```

### 41.3.2 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
```

### 41.3.3 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