www.westermo.com

Merlin 3400 Management Guide

Table of Contents

1. Introduction 10
    1.1. Using this Documentation 10
    1.2. Information Tables 10
    1.3. Diagnostics, Definitions and UCI Commands 11
    1.4. Warning Levels 11
2. Product Overview 13
    2.1. Product Description 13
    2.2. Available Models 13
3. Installation 14
    3.1. Mounting the Router 14
    3.2. Cooling 14
    3.3. Safety Information 14
    3.4. Connecting Cables 16
    3.5. Connecting the Antenna 16
    3.6. Inserting SIM Cards 16
        3.6.1. Inserting SIM 1 Card 17
        3.6.2. Inserting SIM 2 Card 17
    3.7. Powering Up 17
4. Hardware Specification 18
    4.1. Hardware Overview 18
    4.2. Dimensions 20
    4.3. Compliance Information 21
        4.3.1. Agency Approvals and Standards Compliance 21
    4.4. Specifications 21
        4.4.1. Interface Specifications 22
    4.5. Connector Information 23
        4.5.1. Power Input 23
        4.5.2. Serial RJ45 Port 23
        4.5.3. Terminal block 24
        4.5.4. Antennas for model with two SMA Connectors 24
        4.5.5. Antennas for model with four SMA Connectors 24
        4.5.6. Antennas for model with LTE, Wi-Fi and GNSS 24
        4.5.7. SFP Transceivers 24
        4.5.8. Reset Button 25
    4.6. LED Indicators 25
    4.7. Ethernet Port LED Behaviour 26
    4.8. Fibre SFP LED Behaviour 26
5. Factory Configuration Extraction from SIM Card 27
6. Accessing the Router 28
    6.1. Accessing the Router over Ethernet using the Web Interface 28
    6.2. Accessing the Router over Ethernet using an SSH Client 28
    6.3. Accessing the Router over Ethernet using a Telnet Client 29
    6.4. Configuring the Password 30
        6.4.1. Configuring the Password using UCI 31
        6.4.2. Configuring the Password using Package Options 31
    6.5. Accessing the Device Using RADIUS Authentication 31
    6.6. Accessing the Device Using TACACS+ Authentication 33
    6.7. SSH 36
    6.8. Package Dropbear using UCI 39
    6.9. Certs and Private Keys 39
    6.10. Configuring a Router's Web Server 40
        6.10.1. Main Settings 41
    6.11. Basic Authentication (httpd conf) 45

6.12. Securing uhttpd 45
6.13. Displaying Custom Information via Login Screen 46
7. Router File Structure 48
7.1. System Information 48
7.2. Identify Your Software Version 48
7.3. Image Files 49
7.4. Directory Locations for UCI Configuration Files 49
7.5. Viewing and Changing Current Configuration 50
7.6. Configuration File Syntax 50
7.7. Exporting a Configuration File 51
7.7.1. Exporting a Configuration File using the Web Interface 51
7.7.2. Exporting a Configuration File using UCI 51
7.8. Importing a Configuration File 52
7.8.1. Importing a Configuration File using the Web Interface 52
7.8.2. Importing a Configuration File using UCI 53
8. Using the Command Line Interface 54
8.1. Overview of Some Common Commands 54
8.2. Using Unified Configuration Interface (UCI) 56
8.3. Export a Configuration 58
8.4. Show a Configuration Tree 59
8.5. Display Just the Value of an Option 60
8.6. High Level Image Commands 60
8.7. Format of Multiple Rules 60
8.8. Configuration Files 62
8.9. Configuration File Syntax 62
9. Upgrading Router Firmware 64
9.1. Upgrading Router Firmware 64
9.2. Flash Image and Do Not Reboot Option 65
9.3. Update Flash Image and Reboot using New Image Immediately Option 66
9.4. Possible File Corruption 66
9.5. Verify the Firmware Has Been Upgraded Successfully 67
9.6. Upgrading Firmware using CLI 67
9.6.1. Flashing 68
9.6.2. Flash Verification after Flashing 68
9.6.3. Set up an Alternative Image 68
10. System Settings 69
10.1. Syslog Overview 69
10.2. Configuring System Properties 69
10.2.1. General Settings 70
10.2.2. Logging 71
10.2.3. Language and Style 74
10.2.4. Audit Configuration 74
10.2.5. Time Synchronization 75
10.2.6. Console Login Banner 77
10.2.7. System Reboot 78
10.3. System Settings Using Command Line 78
10.3.1. System Settings using UCI 79
10.4. System Diagnostics 80
10.4.1. System Log Messages in RAM 81
10.4.2. System Log Messages in Flash 81
10.4.3. Kernel Messages 82
10.4.4. Syslog Process 83
10.4.5. NTP Process 83
10.5. Advanced Filtering of Syslog Messages 83
10.5.1. Advanced Filtering Using Command Line 83

10.5.2. Filter Definitions ... 84
11. Configuring an Ethernet Interface ... 87
11.1. Configuring an Ethernet Interface using the Web Interface ... 87
11.1.1. Interface Overview: Editing an Existing Interface ... 88
11.1.2. Interface Overview: Creating a New Interface ... 88
11.1.3. Interface Overview: Common Configuration ... 89
11.2. Interface Overview: IP-aliases ... 96
11.2.1. IP-alias using the Web ... 96
11.2.2. IP-aliases: General Setup ... 97
11.2.3. IP-aliases: Advanced Settings ... 98
11.3. Interface Overview: DHCP Server ... 98
11.3.1. DHCP Server: General Setup ... 99
11.3.2. DHCP Server: Advanced Settings ... 101
11.4. Interface Configuration using Command Line ... 101
11.4.1. Interface Configuration using Package Options ... 103
11.4.2. Loopback Interfaces UCI ... 104
11.5. Configuring Port Maps using the Web Interface ... 104
11.6. Configuring Port Maps using UCI ... 105
11.7. Configuring Port Map using Package Options ... 106
11.8. Interface Diagnostics ... 106
11.9. Route Status ... 108
11.10. ARP Table Status ... 108
12. Configuring VLAN ... 109
12.1. Configuring VLAN using the Web Interface ... 109
12.2. General Setup: VLAN ... 111
12.3. Firewall Settings: VLAN ... 112
12.4. Viewing VLAN Interface Settings ... 113
12.5. Configuring VLAN using UCI ... 113
13. Configuring Mobile Manager ... 115
13.1. Configuring Mobile Manager using the Web Interface ... 115
13.1.1. Mobile Manager: Basic Settings ... 115
13.1.2. Mobile Manager: Advanced Settings ... 116
13.1.3. Mobile Manager: LTE Settings ... 118
13.1.4. Mobile Manager: CDMA Settings ... 120
13.1.5. Mobile Manager: Callers ... 123
13.2. Configuring Mobile Manager using UCI ... 123
13.3. Monitoring SMS ... 125
13.3.1. Sending SMS from the Router ... 126
13.3.2. Sending SMS to the Router ... 126
14. Configuring multi-APNs for Mobile Interfaces ... 127
14.1. Multi-APN Overview ... 127
14.2. Configuring multi-APN using the Web Interface ... 127
14.3. Configuring multi-APN using the Command Line ... 128
14.3.1. Configuring Multi-APN using UCI ... 129
14.4. Multi-APN Diagnostics ... 131
14.4.1. Routing table ... 132
15. Configuring a GRE Interface ... 134
15.1. Creating a GRE Connection using the Web Interface ... 134
15.1.1. GRE Connection: Common Configuration: General Setup ... 136
15.1.2. Configuring IPv6 Routes using the Web Interface ... 137
15.1.3. GRE Connection: Common Configuration-advanced Settings ... 138
15.1.4. GRE Connection: Firewall Settings ... 139
15.1.5. GRE Connection: Adding a Static Route ... 140
15.2. Configuring GRE using UCI ... 140
15.3. GRE Diagnostics ... 142

16. Configuring VRF (Virtual Router Forwarding) 145
16.1. Configuring VRF using the Web Interface 145
16.1.1. Configuring a VRF on a Static Route 147
16.2. Configuring the VRFs using the Command Line 147
16.2.1. VRF using UCI 148
16.3. VRF Diagnostics 148
17. Configuring Static Routes 150
17.1. Configuring Static Routes using the Web Interface 150
17.2. Configuring Routes using Command Line 151
17.3. IPv4 Routes using UCI 152
17.4. IPv6 Routes using UCI 152
17.5. Static Routes Diagnostics 153
17.6. Configuring IPv6 Routes using the Web Interface 153
18. Configuring BGP (Border Gateway Protocol) 155
18.1. Configuring BGP using the Web Interface 155
18.2. Optionally Configure a BGP Route Map 157
18.3. Configure BGP Neighbours 159
18.4. Configuring BGP using Command Line 160
18.5. View Routes Statistics 162
19. Configuring OSPF 164
19.1. OSPF Areas 164
19.2. OSPF Neighbours 165
19.3. OSPF Designated Routers 165
19.4. OSPF Neighbour States 166
19.5. OSPF Network Types 166
19.6. The OSPF Hierarchy 167
19.7. OSPF Router Types 168
19.8. Configuring OSPF using the Web Interface 168
19.8.1. Global Settings 169
19.8.2. Topology Configuration 170
19.8.3. Interfaces Configuration 171
19.9. Configuring OSPF using the Command Line 174
19.10. Configuring OSPF using UCI 175
19.11. OSPF Diagnostics 176
19.11.1. Quagga/Zebra Console OSPF 177
19.11.2. OSPF Debug Console 178
20. Configuring VRRP 183
20.1. Configuring Static Routes using the Web Interface 183
20.2. Configuring VRRP using the Web Interface 184
20.3. VRRP Global Settings 184
20.4. VRRP Group Configuration Settings 185
20.5. Configuring VRRP using Command Line 188
20.5.1. VRRP using UCI 189
20.6. VRRP Diagnostics 190
21. Configuring RIP 191
21.1. RIP characteristics 191
21.2. RIP Versions 191
21.3. Configuring RIP using the Web Interface 192
21.3.1. RIP Interfaces Configuration 195
21.3.2. Offset Configuration 196
21.3.3. MD5 authentication key chains 197
21.4. Configuring RIP using Command Line 197
21.4.1. RIP using UCI 199
21.4.2. RIP using Package Options 200
21.5. RIP Diagnostics 201

21.5.1. Tracing RIP Packets 202
21.5.2. Quagga/Zebra console RIP 202
21.5.3. RIP Debug Console 203
22. Configuring PIM and IGMP Interfaces 205
22.1. Configuring PIM and IGMP using the Web Interface 205
22.1.1. Global Settings 206
22.1.2. Interfaces Configuration 206
22.2. Configuring PIM and IGMP using UCI 206
23. Configuring Multi-WAN 208
23.1. Configuring Multi-WAN using the Web Interface 208
23.2. Configuring Multi-WAN using UCI 213
23.3. Multi-WAN Diagnostics 215
24. Automatic Operator Selection 218
24.1. Configuring Automatic Selection via the Web Interface 218
24.1.1. Scenario 1: PMP + Roaming: Pre-empt Enabled 218
24.2. Mobile Manager: Basic Settings 228
24.3. Mobile Manager: Advanced Settings 229
24.4. Mobile Manager: CDMA Settings 230
24.5. Mobile Manager: Callers 233
24.6. Roaming Interface Template: Web Interface 235
24.7. Scenario 2: PMP + Roaming: Pre-empt Disabled 238
24.7.1. Set Multi-WAN Options for Pre-empt Disabled 239
24.8. Scenario 3: No PMP + Roaming 239
24.8.1. Set Options for Automatically Created Interfaces (failover) 240
24.8.2. Basic Settings 240
24.8.3. Caller Settings 241
24.8.4. Configuring no PMP + Roaming using UCI 241
24.8.5. Roaming Interface Template: Web Interface 244
24.8.6. Set Multi-WAN Options for Primary Predefined Interface 247
24.9. Configuring Automatic Operator Selection via UCI 252
24.9.1. PMP + Roaming: Pre-empt & Disabled using UCI 252
24.9.2. Configuring no PMP + Roaming using UCI 258
24.9.3. Automatic Operator Selection Diagnostics via the Web Interface 261
24.10. Automatic Operator Selection Diagnostics via UCI 262
24.10.1. Check Roaming Interfaces Discovered 262
24.10.2. Check Interfaces created in multiwan Package 263
24.10.3. Check Interfaces Created in Network Package 264
24.10.4. Check Current Interface 265
25. Configuring Connection Watch (cwatch) 267
25.1. Configuring Connection Watch using the Web Interface 267
25.2. Configuring cwatch using Command Line 270
25.2.1. cwatch using UCI 271
25.3. cwatch Diagnostics 271
26. Configuring DHCP Server and DNS (Dnsmasq) 273
26.1. Configuring IPv6 Routes using the Web Interface 273
26.2. Configuring DHCP and DNS using the Web Interface 274
26.2.1. Dnsmasq: General Settings 275
26.2.2. Dnsmasq: Resolv and Host Files 276
26.2.3. Dnsmasq: TFTP Settings 277
26.2.4. Dnsmasq: Advanced Settings 278
26.2.5. Active Leases 280
26.2.6. Static Leases: DHCP Server and DNS 280
26.2.7. Configuring DHCP Pools using the Web Interface 281
26.3. Configuring DHCP and DNS using Command Line 283
26.3.1. Configuring Static Leases using Command Line 284

26.3.2. Configuring DHCP Pools using Command Line 284
27. Configuring DHCP Client 286
27.1. Configuring DHCP Client using the Web Interface 286
27.1.1. Creating a New Interface for DHCP Client 287
27.2. Common Configuration 288
27.2.1. Common Configuration: General Setup 289
27.2.2. Common Configuration: Advanced Settings 291
27.3. Configuring DHCP Client using Command Line 293
27.4. DHCP Client Diagnostics 294
28. Configuring DHCP Forwarding 296
28.1. Configuring DHCP using the Web Interface 296
28.2. Configuring DHCP Forwarding using Command Line 297
28.3. DHCP Forwarding over IPSec 298
28.3.1. Configuring Source NAT for DHCP Forwarding over IPSec 298
28.4. Configuring Source NAT for DHCP Forwarding over IPSec 300
28.5. Configuring Source NAT for DHCP Forwarding over IPSec using Command Line .. 303
28.6. DHCP Forwarding Diagnostics 304
29. Configuring Dynamic DNS 306
29.1. Configuring Dynamic DNS using the Web Interface 306
29.2. Dynamic DNS using UCI 309
30. Configuring Host Names 311
30.1. Configuring Local Host Files Entries using the Web Interface 311
30.2. Local Host Records using Command Line 312
30.3. Local Host Records Diagnostics 313
30.4. Configuring PTR Records using the Web Interface 313
30.5. PTR Records using Command Line 314
30.6. PTR Records Diagnostics 314
30.7. Configuring Static Leases using the Web Interface 315
30.8. Configuring Static Leases using Command Line 316
31. Configuring Firewall 318
31.1. Scenario 3: No PMP + Roaming 318
31.2. Configuring Firewall using the Web Interface 318
31.2.1. Firewall - Zone Settings - General Settings page 319
31.2.2. Firewall Port Forwards Page 325
31.2.3. Firewall Traffic Rules Page 329
31.3. Configuring Firewall using UCI 333
31.3.1. Implications of DROP vs. REJECT 337
31.3.2. Connection Tracking 337
31.3.3. Firewall Rule Examples 338
32. Configuring IPSec 345
32.1. Configuring IPSec using the Web Interface 345
32.1.1. Configure Common Settings using the Web Interface 346
32.1.2. Common Settings: Configure Connection using Web Interface 347
32.1.3. Common Settings: IP Addressing using the Web Interface 348
32.2. Common Settings: IPSec Settings using the Web Interface 350
32.3. Configure Secret Settings using the Web Interface 353
32.4. Configuring an IPSec Template for DMVPN via the Web Interface 354
32.5. Configuring IPSec using UCI 354
32.6. Shunt Connection using UCI 357
32.7. Secret Settings using UCI 358
32.8. Configuring an IPSec Template to use with DMVPN using UCI 359
32.9. IPSec Diagnostics 361
33. Configuring SCEP (Simple Certificate Enrolment Protocol) 363
33.1. Configuring SCEP using the Web Interface 363
33.2. Configuring SCEP Certificate using the Command Line 366

33.3. SCEP Diagnostics 369
33.4. Strongswan Process using UCI 370
34. Dynamic Multipoint Virtual Private Network (DMVPN) 371
34.1. Configuring DMVPN using the Web Interface 371
34.2. DMVPN Scenarios 373
34.3. Configuring DMVPN using the Web Interface 375
34.4. DMVPN Diagnostics 377
35. Configuring QoS: VLAN PCP Tagging 381
36. Management Configuration Settings 384
36.1. Autoload: Boot up Activation 384
36.2. Autoload using UCI 387
36.3. HTTP Client: Configuring Activation using the Web Interface 388
36.4. Httpclient: Activator Configuration using UCI 392
36.5. User management using UCI 393
36.6. Configuring Management User Password using UCI 395
36.7. Configuring User Access to Specific Web Pages 396
37. Configuring Monitor 397
37.1. Reporting Device Status to Monitor 397
37.2. Configuring Keepalive Heartbeat using the Web Interface 397
37.3. Configuring keepalive heartbeat using command line 400
37.4. Enabling Interface Status in Keepalive Heartbeat via Web Interface 402
37.5. Enable Interface Status using UCI 403
37.6. Reporting GPS Location to Monitor 403
37.7. GPS Diagnostics 404
37.8. Reporting Syslog to Monitor 405
37.9. Configuring Syslog Events to Monitor using Command Line 406
37.10. Configuring ISAD 407
37.11. ISAD Diagnostics 409
37.12. Speedtest Reporting 410
38. Configuring SNMP 412
38.1. Configuring SNMP using the Web Interface 412
38.1.1. Com2Sec Settings 413
38.1.2. Group Settings using the Web Interface 414
38.1.3. View Settings 414
38.1.4. Access Settings using the Web Interface 415
38.1.5. Trap Receiver 416
38.1.6. Inform Receiver using the Web Interface 417
38.1.7. USM user using Web Interface 417
38.2. Configuring SNMP using Command Line 418
38.2.1. Com2sec using UCI 419
38.2.2. Group Settings using Command Line 420
38.2.3. View Settings using UCI 422
38.2.4. Access Settings Using Command Line 423
38.2.5. SNMP Traps Settings using Command Line 424
38.2.6. SNMP Inform Receiver Settings using Command Line 425
38.2.7. SNMP USM User Settings 426
38.3. Configuring SNMP Interface Alias with Static SNMP Index 427
38.4. Automatic SNMP Traps 428
38.5. SNMP diagnostics 428
39. Event System 431
39.1. Implementation of the Event System 431
39.2. Configuring the Event System using the Web Interface 432
39.3. Connection Tester 432
39.4. Event Destination 433
39.4.1. Syslog Target 434

39.4.2. Email Target 435
39.4.3. SNMP Target 438
39.4.4. Exec Target 440
39.4.5. SMS Target 441
39.4.6. File Target 442
39.5. Event FIlters 443
39.6. Configuring the Event System using Command Line 444
39.6.1. Event System using UCI 446
39.7. Event System Diagnostics 453
40. Configuring Data usage Monitor 457
40.1. Configuring Data using the Web Interface 457
40.2. Configuring Data using Command line 459
40.3. Data Usage Diagnostics 460
41. Configuring Terminal Server 462
41.1. Configuring Terminal Server using the Web Interface 462
41.1.1. Configuring Port Settings 463
41.2. Configuring Terminal Server using UCI 477
41.3. Configuring Terminal Server DSR Signal Management Network 477
41.4. Serial Mode GPIO Control 479
41.5. Terminal server diagnostics 479
42. Configuring Terminal Package 482
42.1. Configuring Terminal Package using UCI 482
42.2. Terminal Diagnostics 483
43. SCADA IEC 104 Gateway 484
43.1. IEC 104 Gateway Configuration using the Web Interface 485
43.1.1. Main Settings 486
43.1.2. Port Settings 486
43.1.3. Port Settings: General 487
43.1.4. Port settings: IEC 104 487
43.1.5. Port Settings: IEC 101 490
43.1.6. Port Settings: DNP3 493
43.1.7. Port Settings: Modbus 496
43.1.8. Port Settings: Advanced 497
43.2. IEC 101 Links 499
43.3. Point Mappings 499
43.4. IEC 104 Gateway Configuration using Command Line 503
43.5. IEC 104 to IEC 101 Conversion (Balanced or Unbalanced) 504
43.6. IEC104 to Modbus Conversion 508
43.6.1. IEC104 to Modbus using UCI 510
43.7. Configuring the Terminal Server 513
43.7.1. Configuring the Terminal Server for IEC104 to DNP3 518
43.7.2. Configuring the Terminal Server for IEC 104 to Modbus over Serial 518
43.8. Configuring IEC61850 to IEC101 Conversion 524
43.8.1. IEC 61850 to IEC 101 Conversion using UCI 527
43.9. SCADA IEC 104 Gateway Diagnostics 531
44. DNP3 Outstation Application 534
44.1. Configuring DNP3 Outstation using the Web Interface 534
44.2. Configuring DNP3 Outstation using Command Line 535
44.3. DNP3 Outstation Diagnostics 536
45. Serial Interface 538
45.1. Monitoring Serial Interfaces using the Web Interface 538
45.2. Set Options for Automatically Created Interfaces (failover) 539
45.3. Monitoring Serial Interfaces using Command Line 539

1. Introduction

This document covers models in the Merlin 3400 Series. For general references, we refer to the Merlin Series throughout.

1.1. 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. 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.

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

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

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

NOTE

These sections can be given a label for identification when using UCI or package options.

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

can be written as:

```
network.routename=route

network.routename.metric=0
```

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

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

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

1.3. Diagnostics, Definitions And UCI Commands

Diagnostics

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

Definitions

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

UCI Commands

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

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

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

1.4. Warning Levels

Warning signs are provided to prevent personal injuries and/or damages to the product. The following levels are used:

Level of warning	Description	Consequence personal injury	Consequence material damage
WARNING	Indicates a potentially hazardous situation	Possible death or major injury	Major damage to the product
CAUTION	Indicates a potentially hazardous situation	Minor or moderate injury	Moderate damage to the product
NOTICE	Provides information in order to avoid misuse of the product, confusion or misunderstanding	No personal injury	Minor damage to the product
NOTE	Used for highlighting general, but important information	No personal injury	Minor damage to the product

Table 1. Warning levels

2. Product Overview

2.1. Product Description

The Merlin 3400 Series

The Merlin series of versatile cellular routers provide high speed data network connectivity for demanding industrial and smart grid applications. Coupled with the Zero Touch deployment software Activator, it provides cost-effective, reliable, and consistent onboarding of routers in large-scale projects.

This compact, rugged unit is suited to tight spaces. Its high MTBF and wide temperature range support maximum service life.

To achieve best-in-class cybersecurity, the Merlin series is equipped with a TPM (Trusted Platform Module) chip that keeps cryptographic keys secure. Secure Boot ensures that the unit boots using only software that is signed and trusted by manufacturer. A set of cybersecurity tools is available as standard including VPN and stateful Firewall support for data security and user authentication.

2.2. Available Models

See datasheet for full list of models available.

3. Installation

3.1. Mounting The Router

The router has an integrated DIN-rail clip. To attach the router to a DIN-rail:

1. Pull down the plastic clasp at rear until it snaps into place in the downward position
2. Position the router so that the top hook of the DIN-clip rests on top of the DIN-rail
3. Allow the router to pivot under gravity so that it sits on the DIN-rail in the vertical position
4. Push up the plastic clasp until it snaps into place, locking the router firmly to the DIN-rail

To remove the router from the DIN-rail, simply reverse the procedure.

3.2. Cooling

This product uses convection cooling. Spacing is recommended for the use of the product in full operating temperature range and service life. To avoid obstructing the airflow around the product, use the following spacing rules.

A minimum spacing of 25 mm (1 inch) above and below, and 10 mm (0.4 inches) left and right of the product is recommended.

3.3. Safety Information

SAFETY DURING INSTALLATION

The product must be installed and operated by qualified service personnel and installed into an apparatus cabinet or similar, where access is restricted to service personnel only.

Before energising and connecting communication cables to the product, ensure a protective earthing conductor is first connected to the protective earthing terminal (only valid for metallic housings). Westermo recommends a cross-sectional area of at least 4 mm^2.

Upon removal of the product, disconnect the product from the power supply and all other communication ports before disconnecting the protective earthing conductor.

SAFETY USE

This equipment is not suitable for use in locations where children are likely to be present.

HAZARDOUS VOLTAGE

Do not open an energised product. Hazardous voltage may occur when connected to a power supply.

PROTECTIVE FUSE

The power supply wiring must be sufficiently fused.

This product has no internal fuse and should be connected via an external fuse for protection.

The Merlin 3400 4G models have a typical current of 400mA drawn at 12 VDC. Hence a 1A fuse is recommended

It must be possible to disconnect manually from the power supply. Ensure compliance to national installation regulations.

POWER SUPPLY CONNECTION

There are safety regulations governing the power source that can be used in conjunction with the product. Refer to the user guide for detailed information.

EARTH CONNECTION

For correct function, the earth connection needs to be properly connected to a designated protective earthing rail (only valid for metallic housings). Torque: 3.1 Nm.

INDOOR USE ONLY

The device is to be connected only to networks without routing to the outside plant. All communication ports are for indoor use only.

REDUCE THE RISK OF FIRE

To reduce the risk of fire, use only telecommunication line cords with a cable diameter of AWG 26 or larger. Regarding power cable dimensions, see chapter Interface Specifications.

RADIO PRODUCTS

Observe the usage limitations of radio products at filling stations, in chemical plants, in systems with explosives or potentially explosive locations.

The product may not be used in airplanes. Exercise particular caution near personal medical aids, such as pacemakers and hearing aids. Never perform work on the antenna system during a thunderstorm.

To fulfill human safety, a minimum separation distance of 20 cm or more should be maintained between the antenna of the product and personnel during operation.

ELECTROSTATIC DISCHARGE (ESD)

Prevent electrostatic discharge damage to internal electronic parts by discharging your body to a grounding point (e.g. use a wrist strap).

HOT SURFACE

Be aware that the surface of this product may become hot. When it is operated at high temperatures, the external surface may exceed Touch Temperature Limit according to the product's relevant electrical safety standard.

CABLE TEMPERATURE RATING FOR FIELD TERMINAL WIRES

There may be a requirement on the minimum temperature rating of the cable to be connected to the field wiring terminals, refer to the user guide for detailed information.

3.4. Connecting Cables

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

3.5. Connecting The Antenna

If only connecting one LTE antenna to a 4G or 5G Redcap model, screw the antenna into the MAIN SMA connector. If you are using more than one LTE antenna, screw the main antenna into the MAIN SMA connector and the secondary antenna into the WAN-AUX SMA connector.

Be careful to select an antenna that ensures the equipment complies with local regulatory requirements.

In North America, Westermo has indicated to PTCRB the following set of antennas for use with this equipment:

Article Number	Model
160-00148	2J 2J7724Ma - Cellular/LTE MIMO Magnetic Mount with two 3m LL100 leads, SMA Male
160-00184	2J 2J2183K-B07H-300LL195-C91GST - Cellular/LTE wall mount antenna with a 3m LL195 lead, SMA Male

Table 2. List of antennas for use with this equipment in North America, as indicated to PTCRB

To comply with the requirements of PTCRB in North America:

1. It is recommended to select an antenna from the list above.
2. Should a user select their own antenna, Westermo strongly recommends that it be mounted with at least 20cm of cable.
3. Where a user selects a directly mounted antenna, it must comply with the OTA requirements of the PTCRB.

3.6. Inserting SIM Cards

On the rear side of the router there are two SIM slots. To access the SIM cards, first remove the SIM cover using a suitable screwdriver (not supplied). Only the proper driver can drive a specific head size without risk of damaging the driver or screw.

A Merlin 3400 dual-radio model has a third SIM slot presented on the front face. The paragraph above applies.

The product datasheet indicates the correct size of SIM card to use. A Mini-SIM card (2FF) is of size 25mm x 15mm x 0.76mm. Do not use a card larger than or smaller than that specified in the product datasheet or you may cause damage to the unit. Using the correct SIM card size helps avoid issues such as incompatibility or improper insertion that can lead to device malfunction.

3.6.1. Inserting SIM 1 Card

Ensure the router is powered off.

- Hold the SIM 1 card with the chip side facing down and the cut corner facing away from you, to the left.
- Gently push the SIM card into the upper SIM slot 1 until it clicks in.

3.6.2. Inserting SIM 2 Card

- If you are using a second SIM, hold the SIM 2 card with the chip side facing up and the cut corner front right facing away from you.
- Gently push the SIM card into the lower SIM slot 2 until it clicks in.

Screw the SIM cover back on with the screwdriver.

3.7. Powering Up

Plug the power cable first into the device and then to a suitable power source. The router takes less than a minute 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.

4. Hardware Specification

4.1. Hardware Overview

The location of interfaces ports and LED indicators for Merlin-3409-T4G-S3-DIO2-AI1-WF-QFR-G-DCi1060 and subsets thereof is shown here.

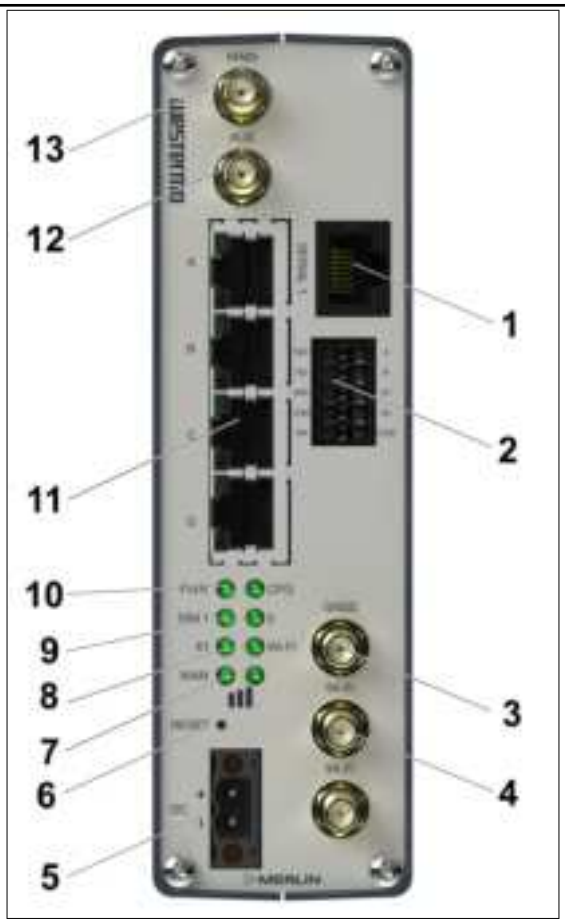


Figure 1. Location of interface ports and LED indicators, illustrated by a Merlin-3409-T4G-S3-DIO2-AI1-WF-QFR-G-DCi1060 model

No.	Description	No.	Description
1	Serial RJ45 port	2	5x2 terminal block
3	GNSS SMA connector	4	Where Wi-Fi feature is present: a pair of Reverse-Polarity Female SMA connectors
5	Power connection	6	Reset button
7	WAN signal strength	8	For future use
9	Where Wi-Fi feature is present: Wi-Fi status LED. Otherwise, for future use	10	SIM LEDs
11	Power and configuration LEDs	12	Ethernet RJ45 ports
13	WAN auxiliary SMA connector	14	WAN primary SMA connector

The location of interface ports and LED indicators for Merlin-3407-T4G-WF-QFR-PFN-G-DCi1060 is shown below.

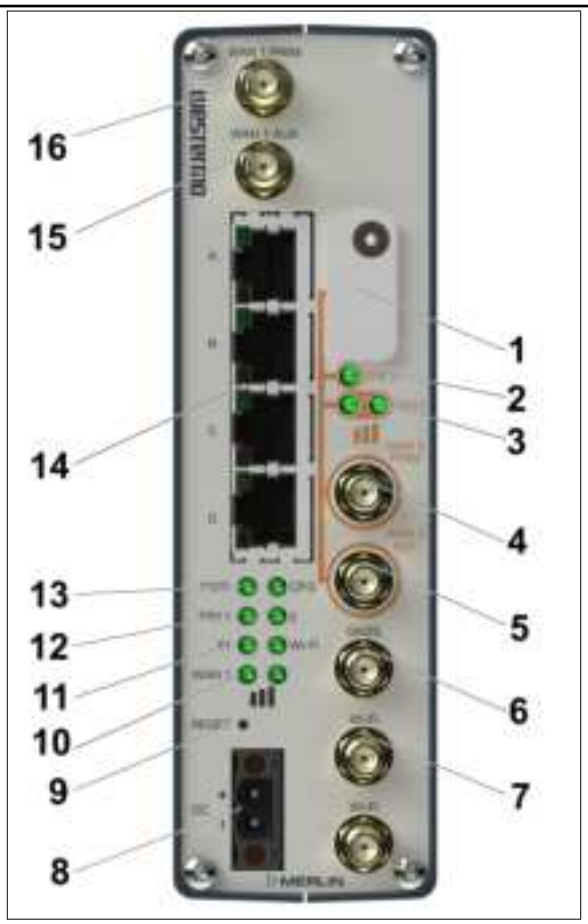


Figure 2. Location of interface ports and LED indicators, illustrated by a Merlin-3407-T4G-WF-QFR-PFN-G-DCi1060 model

No.	Description	No.	Description
1	SIM 3 slot and cover	2	SIM 3 LED
3	WAN signal strength of second cellular radio	4	Primary SMA connector of second cellular radio
5	Auxiliary SMA connector of second cellular radio	6	GNSS SMA connector
7	Pair of Reverse-Polarity Female SMA connectors for Wi-Fi	8	Power connection
9	Reset button	10	WAN signal strength of first cellular radio
11	Wi-Fi status LED	12	SIM LEDs
13	Power and configuration LEDs	14	Ethernet RJ45 ports
15	Auxiliary SMA connector of first cellular radio	16	Primary SMA connector of first cellular radio

The location of interface ports and LED indicators for Merlin-3407-F2G-T4G-PFO-DCi1060 is shown below.

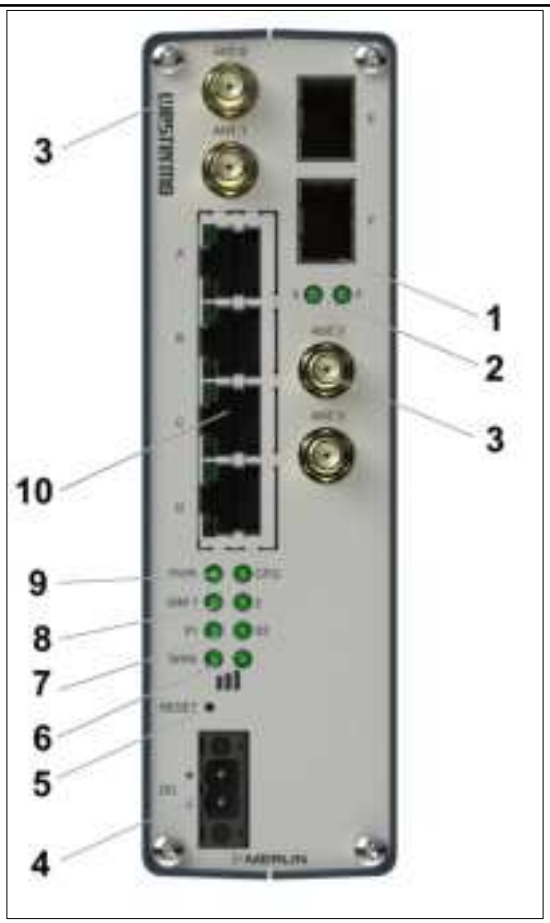


Figure 3. Location of interface ports and LED indicators, illustrated by a Merlin-3407-F2G-T4G-PFO-DCi1060 model

No.	Description	No.	Description
1	Fibre SFP ports	2	Fibre SFP status LEDs
3	Antenna SMA connectors	4	Power connection
5	Reset button	6	WAN signal strength of cellular radio
7	Reserved for future use	8	SIM LEDs
9	Power and configuration LEDs	10	Ethernet RJ45 ports

4.2. Dimensions

Dimensions are stated in millimetres and are regardless of model.

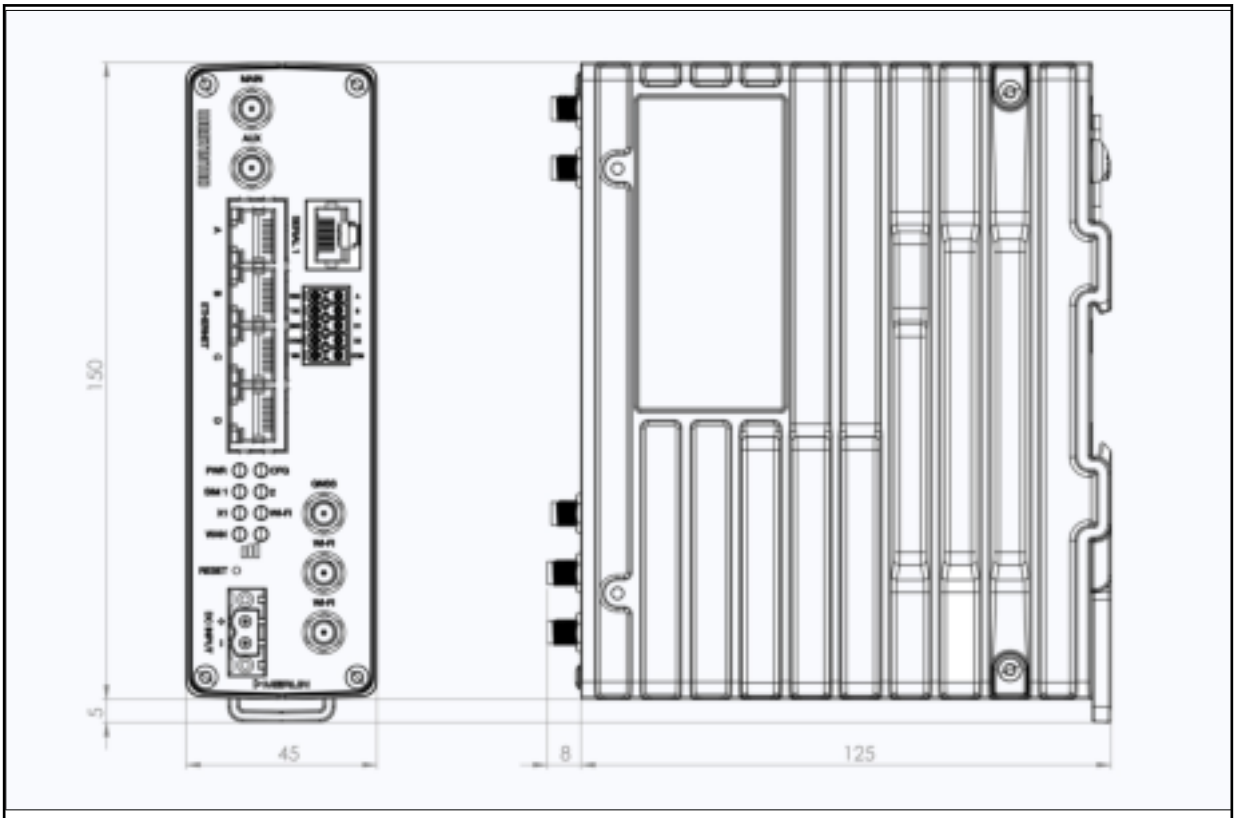


Figure 4. Dimensional drawing

4.3. Compliance Information

4.3.1. Agency Approvals And Standards Compliance

Type	Approval/Compliance
EMC	• EN/IEC 61000-6-2, Immunity industrial environments • EN/IEC 61000-6-4, Emission industrial environments
Safety	• EN 62368-1, Safety Communication Technology

Table 3. Agency approvals and standards compliance

4.4. Specifications

4.4.1. Interface Specifications

DC, Power port	
Operating voltage	9.6 to 60 VDC
Rated current	580 mA at 12 VDC 170 mA at 48 VDC
Rated frequency	DC
Inrush current	2.74×10^{-3} A^2s at 12 VDC
Polarity	Reverse polarity protected
Redundant power input	No
Connector	Push-in spring connectors
Conductor cross section	0.2-2.5 mm^2 (AWG 24-12)
Stripping length cable	7 mm
Tightening torque, screw flange	0.3 Nm
Shielded cable	Not required

Ethernet TX	
Electrical specification	IEEE std 802.3
Data rate	10 Mbit/s, 100 Mbit/s, 1000 Mbits/s manual or auto
Duplex	Full or half, manual or auto
Transmission range	Up to 150 m with CAT5e cable or better
Connection	RJ-45, auto MDI/MDI-X
Cabling	Shielded CAT5e or better is recommended
Number of ports	4

RS-232/485	
Electrical specification	Configurable for EIA RS-232 or EIA RS-422/485
Data rate	RS-232: 50 bit/s - 1 Mbit/s RS-485: 50 bit/s - 20 Mbit/s
Data format	7 or 8 data bits, odd, even or none parity, 1 or 2 stop bits (2 stop bits only when no parity is set)
Transmission range	RS-232: 15 m/49 ft RS-485: Up to 1200 m/0.74 mi, depending on data rate and cable type
Connection	RS-232: RJ-45 according to EIA-561; 3-wire RS-232 is detachable latch terminal RS-485: Detachable latch terminal
Shielded cable	Recommended
Number of ports	Up to 1 RJ45 RS-232, up to 1 3-wire RS-232, up to 1 two-wire RS-485

I/O connection, Relay output	
Maximum voltage/current	Up to 60 VDC
Connect resistance	<100mΩ
Connector	Detachable latch terminal
Conductor cross section	0.14 - 1.5 mm^2 (AWG 28 - 16)
Stripping length cable	7 mm
Number of ports	Up to 2

I/O connection, Digital input	
Maximum voltage	24 VDC
Voltage levels	Logic one: >9.9 VDC Logic zero: <7.4 VDC
Connector	Detachable latch terminal
Conductor cross section	0.14 - 1.5 mm² (AWG 28 - 16)
Stripping length cable	7 mm
Number of inputs	Up to 2

4.5. Connector Information

4.5.1. Power Input

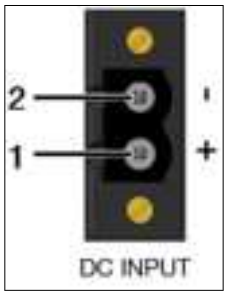
Illustration	Position	Product marking	Direction	Description
	1	DC+	Input	Supply voltage
	2	DC-	Input	Supply voltage

Table 4. Power input

The positive input is marked with a plus sign, "+". The negative input is marked with a minus sign, "-". Connect the voltage to the + pin and the return to the - pin on the power input.

NOTICE - POWER SUPPLY

Where an AC/DC-adapter has not been supplied, a power supply of no greater than 100 W should be used, with a current limit of 1 Amp.

The product can be supplied with power from a source that complies with PS2/LPS according to UL/IEC 62368-1 or with LPS according to UL/IEC 60950-1, the latest edition or equivalent. Such a power supply shall not exceed 100W after 5 seconds.

4.5.2. Serial RJ45 Port

An asynchronous RJ45 serial port may be present on the router. The serial port is named as follows, also its identifier for use within the terminal server configuration:

Label	tservd.port.serialPortName
SERIAL 1	serial1

The RJ45 serial port operates in RS-232 mode.

The pin numbering of the serial port connector is shown below.

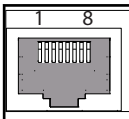


Figure 5. Pin numbering of serial port

4.5.3. Terminal Block

The pins of the terminal block, where present, are shown below,

Illustration	Signal	Direction	Description
	A	-	Half-duplex RS-485
	B	-	Half-duplex RS-485
	D1	In/Out	Digital Input/Output 1
	D2	In/Out	Digital Input/Output 2
	COM	-	Signal Ground, not chassis ground
	RXD	Out	RS-232 Receive data
	TXD	In	RS-232 Transmit data
	GND	-	Signal Ground, not chassis ground. Use with RXD and TXD above
	COM	-	Signal Ground, not chassis ground
	VIN	Out	Analog input. Use with COM above. Measures voltage 1-60 VDC to common ground with a resolution of 0.1V.

Table 5. Terminal block

4.5.4. Antennas For Model With Two SMA Connectors

Where the router has two SMA connectors for mobile radio LTE antennas, one is for MAIN and the other for AUXiliary.

4.5.5. Antennas For Model With Four SMA Connectors

For routers with four SMA connectors marked WAN 1 PRIM, WAN 1 AUX, WAN 2 PRIM and WAN 2 AUX, they are used for dual radio 4G communication. It is recommended to use all four data antenna connectors.

4.5.6. Antennas For Model With LTE, Wi-Fi And GNSS

For routers with LTE, Wi-Fi and GNSS there are five SMA connectors, they are:

- Two LTE antennas for the mobile radio - a MAIN and an AUXiliary
- Single antenna for GNSS/GPS
- A pair of Reverse-Polarity SMA Female SMA connectors for Wi-Fi

4.5.7. SFP Transceivers

Each SFP slot can hold one SFP transceiver. See "*Transceiver User Guide*" for transceiver handling instructions, which also can be downloaded from the product support pages at www.westermo.com/support/product-support.

In the event of contamination, the optical connectors in the SFP transceivers should only be cleaned by the use of forced nitrogen and some kind of cleaning stick. Recommended cleaning fluids are methyl-, ethyl-, isopropyl- or isobutyl alcohol, hexane or naphtha.

HANDLING OF SFP TRANSCEIVERS

SFP transceivers are supplied with plugs to avoid contamination inside the optical port. They are very sensitive to dust and dirt. If the fibre optic cable is disconnected from the product, a protective plug must be used on the transmitter/receiver. The protective plug must be kept on during transportation. The fibre optic cable must be handled the same way.

4.5.8. Reset Button

Use the reset button to request a system reset. When pressing the reset button, all LEDs turn on simultaneously. The length of time holding the reset button will determine its behaviour.

Press duration	PWR/CONFIG LED behaviour	Router behaviour on depress
0-3 seconds	Solid on	Normal reset to running config. No special LED activity.
3-15 seconds	Flashing fast	Releasing 3-15 seconds switches the router back to factory configuration. Note: this will wipe the configurations, both config1 and config2.
15-20 seconds	Solid on	Releasing 15-20 seconds performs a normal reset to running config.
20-30 seconds	Flashing slowly	Releasing 20-30 seconds reboots the router to recovery mode. Only to be done in case of emergency and under the guidance of Westermo support staff. Note: this may wipe the configurations, both config1 and config2.
> 30 seconds	Solid on	Releasing after 30 seconds performs a normal reset.

Table 6. Merlin series router reset behaviour

4.6. LED Indicators

The LED indicators described in this section are all single colour LEDs. When the router is powered on, the power LED is green.

The possible LED states are:

- Off
- Flashing slowly
- Flashing quickly
- On

LED	Status	Description
Booting up		The router takes less than a minute to boot up. During this time, the power LED flashes. Other LEDs display different diagnostic patterns during boot up. Booting is complete when the power LED stops flashing and stays on steady.
Power	On	Power is present
	Off	No power. Boot loader does not exist.
	Flashing	Booting
CFG	On	The router is running a valid configuration file.
	Flashing slowly	The router is running in recovery mode (2.5 flashes/second)
	Flashing quickly	The router is running in factory configuration (5 flashes/second)
SIM	On	SIM selected and registered on the 3G/4G network
	Off	Not selected or SIM not inserted
	Flashing	SIM selected and not registered on the network
Cellular signal strength LED	Both LEDs off	Data link not connected or signal strength <=-113 dBm
	Left LED on Right LED off	Data link connected and signal strength <=-89 dBm
	Left LED off Right LED on	Data link connected and signal strength is between -89 to -69 dBm
	Both LEDs on	Data link connected and signal strength >-69 dBm

Table 7. LED indicators

4.7. Ethernet Port LED Behaviour

There are four Ethernet ports and each has a pair of LEDs: a LINK LED (green) and a SPEED LED (amber).

Figure 6. Merlin Ethernet ports

LINK LED (green)	On	Physical Ethernet link detected
	Off	No physical Ethernet link detected
	Flashing	Data is being transmitted or received over the link
SPEED LED (amber)	On	Link operating at 1000 Mbps
	Off	Link operating at 10 Mbps or at 100 Mbps

Table 8. Ethernet LED behaviour and descriptions

4.8. Fibre SFP LED Behaviour

Each fibre SFP port has a single green-coloured LED.

LINK LED (green)	On	Physical Ethernet link detected.
	Off	No physical Ethernet link detected.
	Flashing	Data is being transmitted or received over the link.

Table 9. Fibre LED behaviour and descriptions

5. Factory Configuration Extraction From SIM Card

Merlin Series 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. Insert the SIM card into SIM slot 1 following the instructions detailed in the Installation chapter.
4. 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 LTE 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.

6. Accessing The Router

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

Configuration Packages Used

Package	Sections
dropbear	dropbear
system	main
uhttpd	main cert

6.1. 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.

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.

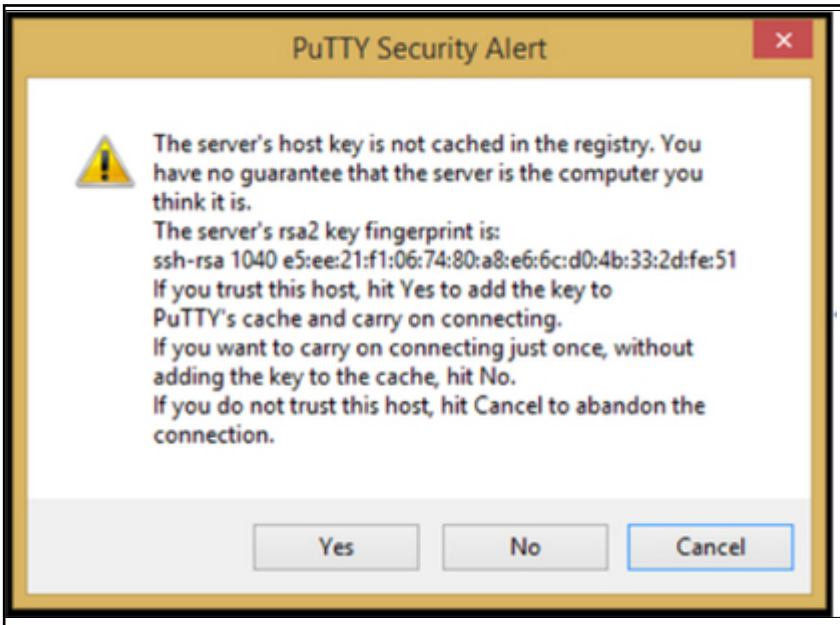
6.2. Accessing The Router Over Ethernet Using An SSH Client

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

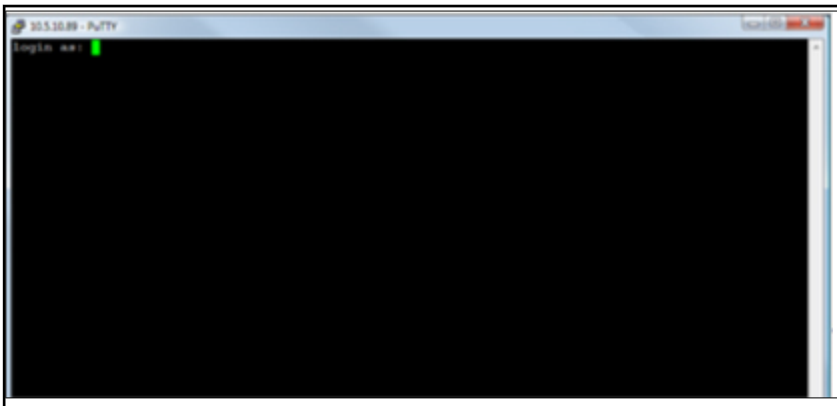
To access CLI over Ethernet start an SSH client and connect to the router's management

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.



Confirming trust of the router's public key over SSH



SSH CLI logon screen

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

Username: **root**

Password: **admin**

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.

6.3. 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
```

To re-enable SSH, enter:

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

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

6.4. Configuring The Password

Configuration packages used

Packages	Sections
system	main

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

The router password section

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

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

NOTE
The username 'root' cannot be changed.

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

6.4.1. 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 a reboot and will now be displayed in encrypted format via the hashpassword option.

6.4.2. 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.

6.5. Accessing The Device Using RADIUS Authentication

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

```
package system

config system 'main'

option hostname 'VirtualAccess'

option timezone 'UTC'

config pam_auth

option enabled 'yes'

option pamservice 'login'

option pammodule 'auth'

option pamcontrol 'sufficient'

option type 'radius'

option servers '192.168.0.1:3333|test|20 192.168.2.5|secret|10'

config pam_auth

option enabled 'yes'

option pamservice 'sshd'

option pammodule 'auth'

option pamcontrol 'sufficient' ...it checks package management_users

option type 'radius'

option servers '192.168.0.1:3333|test|20 192.168.2.5|secret|10'

config 'pam_auth'

option enabled 'yes'

option pamservice 'luci"

option pammodule 'auth'

option pamcontrol 'sufficient'

option type 'radius'

servers '192.168.0.1:3333|test|20 192.168.2.5|secret|10'
```

<table>
<tr><th>UCI/Package Option</th><th>Description</th></tr>
<tr><td>UCI: system.@pam_auth[0].enabled=yes

Opt: enabled</td><td>Enables and disables RADIUS configuration sections.

<table><tr><td>Yes:</td><td>enables the following RADIUS configuration section</td></tr><tr><td>No:</td><td>disables the following RADIUS configuration section</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamservice

Opt: pamservice</td><td>Selects the method which users should be authenticated by.

<table><tr><td>login:</td><td>User connecting over console cable</td></tr><tr><td>sshd:</td><td>User connecting over SSH</td></tr><tr><td>luci:</td><td>User connecting over web</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamcontrol

Opt: pamcontrol</td><td>Specifies authentication behaviour after authentication fails or connection to RADIUS server is broken.

<table><tr><td>Sufficient</td><td>First authenticates against remote RADIUS if password authentication fails then it tries the local database (user defined in package management_users).</td></tr><tr><td>Required:</td><td>If either authentication fails of the RADIUS server is not reachable then the user is not allowed to access the router.</td></tr><tr><td>[success=done new_authtok_reqd=done authinfo_unavail=ignore default=die]:</td><td>Local database is only checked if the RADIUS server is not reachable.</td></tr></table></td></tr>
<tr><td>UCI:

system.@pam_auth[0].pammodule.auth

Opt: pammodule</td><td>Enables user authentication.</td></tr>
<tr><td>UCI: system.@pam_auth[0].type.radius

Opt: type</td><td>Specifies the authentication method.</td></tr>
<tr><td>UCI: system.@pam_auth[0].servers

Opt: servers</td><td>Specifies the RADIUS server along with port number, password and timeout in seconds.

Port and timeout are optional. The default port for RADIUS is 1812; default timeout is 10 seconds.

Multiple servers are entered using a space separator. Syntax:

<server ip address>[:<port>]|<secret>[|timeout]

Examples:

option servers `192.168.0.1test'<br><br>option servers `192.168.0.1|test 192.168.2.5:1234|secret|10'</td></tr>
<tr><td>UCI: system.@pam_auth[1].args=service=ppp
Opt: args</td><td>Additional arguments to pass to TACACS server.</td></tr>
</table>

Information table for RADIUS authentication

6.6. Accessing The Device Using TACACS+ Authentication

You can configure TACACS+ 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 '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'

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

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

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

<table>
<tr><th>UCI/Package Option</th><th>Description</th></tr>
<tr><td>UCI: system.@pam_auth[0].enabled=yes

Opt: enabled</td><td>Enables and disables TACACS configuration sections.

<table><tr><td>Yes:</td><td>enables the following TACACS configuration section</td></tr><tr><td>No:</td><td>disables the following TACACS configuration section</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamservice

Opt: pamservice</td><td>Selects the method which users should be authenticated by.

<table><tr><td>login:</td><td>User connecting over console cable</td></tr><tr><td>sshd:</td><td>User connecting over SSH</td></tr><tr><td>luci:</td><td>User connecting over web</td></tr></table></td></tr>
<tr><td>UCI: system.@pam_auth[0].pamcontrol

Opt: pamcontrol</td><td>Specifies authentication behaviour after authentication fails or connection to TACACS server is broken.

<table><tr><td>Sufficient</td><td>First authenticates against remote TACACS if password authentication fails then it tries the local database (user defined in package management_users).</td></tr><tr><td>Required:</td><td>If either authentication fails of the TACACS server is not reachable then the user is not allowed to access the router.</td></tr><tr><td>[success=done new_authtok_reqd=done authinfo_unavail=ignore default=die]:</td><td>Local database is only checked if the TACACS server is not reachable.</td></tr></table></td></tr>
<tr><td>UCI:

system.@pam_auth[0].pammodule.auth

Opt: pammodule</td><td>Enables user authentication.</td></tr>
<tr><td>UCI: system.@pam_auth[0].type.radius

Opt: type</td><td>Specifies the authentication method.</td></tr>
<tr><td>UCI: system.@pam_auth[0].servers

Opt: servers</td><td>Specifies the TACACS server along with port number, password and timeout in seconds.

Port and timeout are optional. The default port for TACACS is 49.

Multiple servers are entered using a space separator. Syntax:

<server ip address>[:<port>]|<secret>

Examples:

option servers `192.168.0.1test'<br><br>option servers `192.168.0.1|test 192.168.2.5:1234|secret|10'</td></tr>
<tr><td>UCI: system.@pam_auth[1].args=service=ppp
Opt: args</td><td>Additional arguments to pass to TACACS server.</td></tr>
</table>

6.7. 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 using the web interface or through an SSH connection by editing the file stored on: /etc/config_name/dropbear.

Configuration packages used

Package	Sections
dropbear	dropbear

SSH access using the web interface

In the top menu, click **System -> Administration**. The Administration page appears.

Scroll down to the SSH Access section.

The SSH access section

<table>
<tr><th>Web Field/UCI/Package Options</th><th>Description</th></tr>
<tr><td>Web: Interface

UCI: dropbear.@dropbear[0].Interface

Opt: Interface</td><td>Listens only on the selected interface. If you check unspecified, it listens on all interfaces. All configured interfaces will be displayed via the web GUI.

<table><tr><td>(unspecified)</td><td>Listend on all interfaces.</td></tr><tr><td>Range</td><td>Configured interface names.</td></tr></table></td></tr>
<tr><td>Web: Port

UCI: dropbear.@dropbear[0].Port

Opt: port</td><td>Specifies the listening port of the Dropbear instance.

<table><tr><td>22</td><td>Disabled</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: Password authentication

UCI: dropbear.@dropbear[0].PasswordAuth

Opt: PasswordAuth</td><td>If enabled, allows SSH password authentication.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Allow root logins with password

UCI: dropbear.@dropbear[0].RootPasswordAuth

Opt: RootPasswordAuth</td><td>Allows the root user to login with password.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Gateway ports

UCI: dropbear.@dropbear[0].GatewayPorts

Opt: GatewayPorts</td><td>Allows remote hosts to connect to local SSH forwarded ports.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Idle Session Timeout

UCI: dropbear.@dropbear[0].IdleTimeout

Opt: IdleTimeout</td><td>Defines the idle period where the remote session will be closed after the allocated number of seconds of inactivity.

<table><tr><td>30</td><td>30 seconds.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: dropbear.@dropbear[0]. BannerFile

Opt: BannerFile</td><td>Defines a banner file to be displayed during login.

<table><tr><td>/etc/banner</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Maximum login attempts

UCI: dropbear.@dropbear[0].MaxLoginAttempts

Opt: MaxLoginAttempts</td><td>Specifies maximum login failures before session terminates.

<table><tr><td>10</td><td></td></tr><tr><td>0-infinite</td><td></td></tr></table></td></tr>
</table>

6.8. Package Dropbear Using UCI

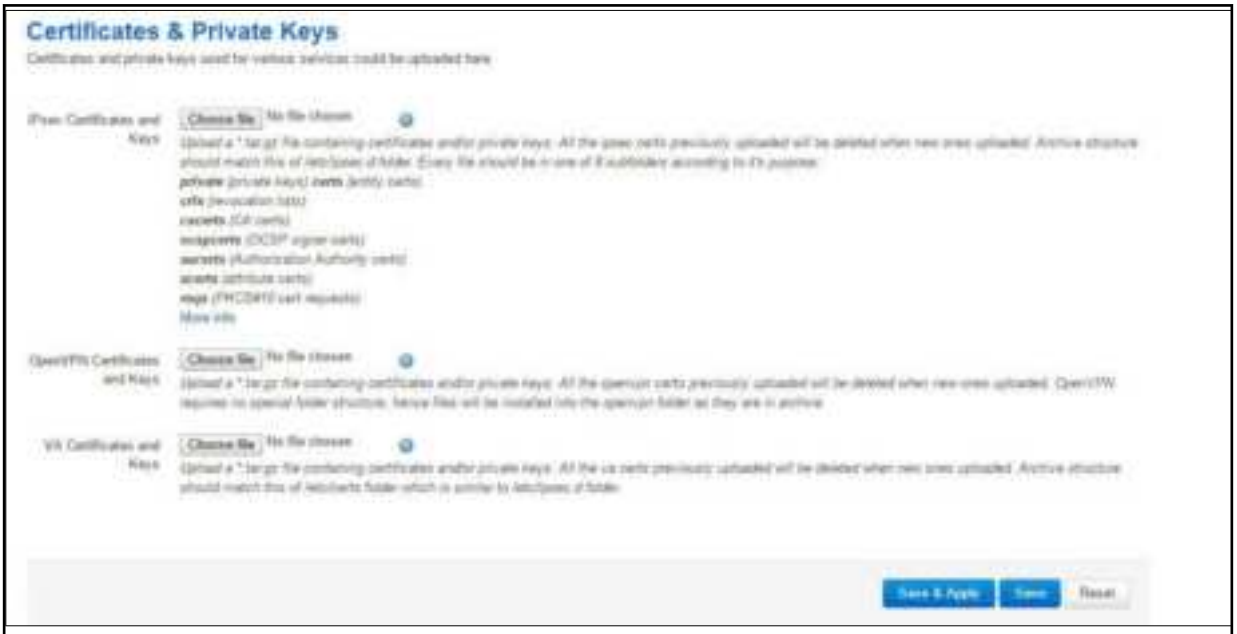
```
root@VA_router:~# uci show dropbear
dropbear.@dropbear[0]=dropbear
dropbear.@dropbear[0].PasswordAuth=on
dropbear.@dropbear[0].RootPasswordAuth=on
dropbear.@dropbear[0].GatewayPorts=0
dropbear.@dropbear[0].IdleTimeout=30
dropbear.@dropbear[0].Port=22
dropbear.@dropbear[0].MaxLoginAttempts=3

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'
```

6.9. 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.



The certificates and 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.

The SSH-keys box

6.10. 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.

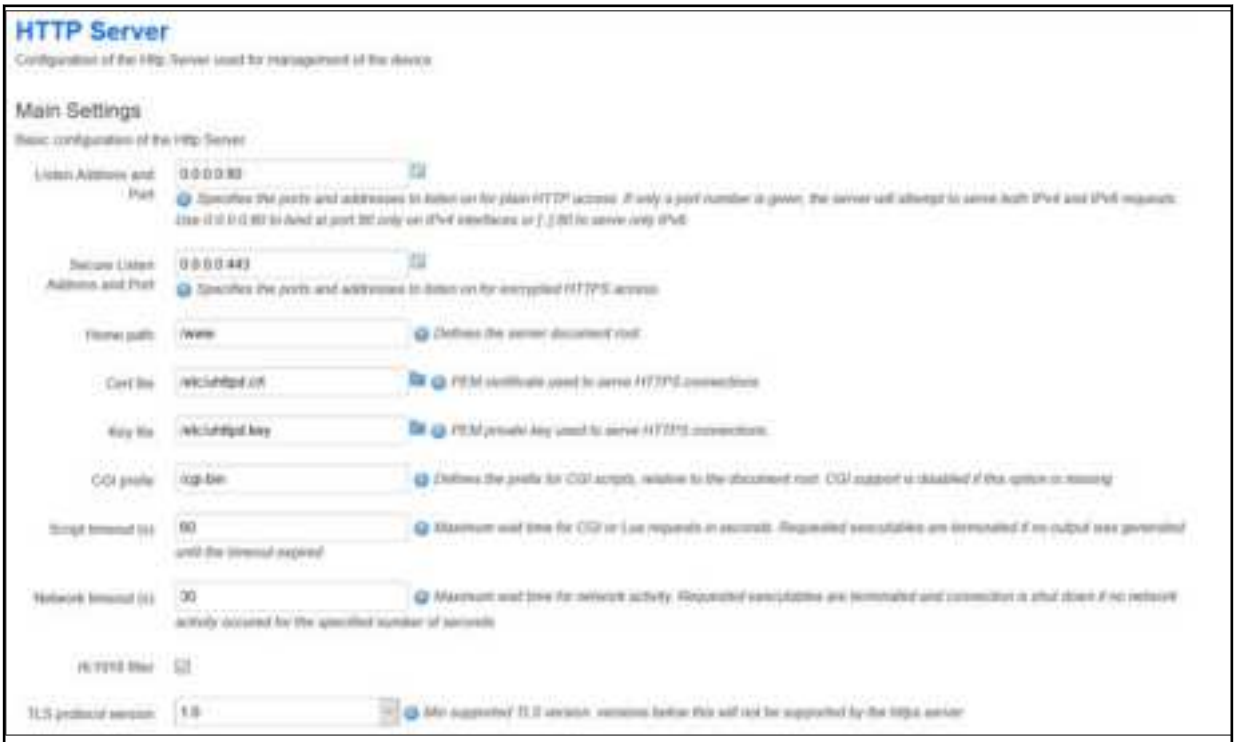
Configuration Packages Used

Package	Sections
uhttpd	main cert

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

Main Settings	Server Configurations
Certificate Settings	SSL certifications

6.10.1. Main Settings



HTTP server main settings

Web Field/UCI/Package Option	Description
Web: Listen Address and Port UCI: uhttpd.main.listen_http 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. 0.0.0.0:80 — Bind at port 80 only on IPv4 interfaces. [::]:80 — Bind at port 80 only on IPv6 interfaces. Range — IP address and/or port
Web: Secure Listen Address and Port UCI: uhttpd.main.listen_https Opt: list listen_https	Specifies the ports and address to listen on for encrypted HTTPS access. The format is the same as listen_http. 0.0.0.0:443 — Bind at port 443 only. [::]:443 Range — IP address and/or port.
Web: Home path UCI: uhttpd.main.home Opt: home	Defines the server document root /www Range

Web: Cert file UCI: uhttpd.main.cert Opt: cert	ASN.1/DER certificate used to serve HTTPS connections. If no listen_https options are given the key options are ignored. /etc/uhttpd.crt Range
Web: Key file UCI: uhttpd.main.key Opt: key	ASN.1/DER private key used to serve HTTPS connections. If no listen_https options are given the key options are ignored. /etc/uhttpd.key Range
Web: CGI profile UCI: uhttpd.main.cgi_prefix Opt: cgi_prefix	Defines the prefix for CGI scripts, relative to the document root. CGI support is disabled if this option is missing. /cgi-bin Range
Web: N/A UCI: uhttpd.main.lua_prefix Opt: lua_prefix	Defines the prefix for dispatching requests to the embedded lua interpreter, relative to the document root. Lua support is disabled if this option is missing. /luci Range
Web: N/A UCI: uhttpd.main.lua_handler Opt: lua_handler	Specifies the lua handler script used to initialise the lua runtime on server start. /usr/lib/lua/luci/sgi/uhttpd.lua Range
Web: Script timeout UCI: uhttpd.main.script_timeout Opt: script_timeout	Sets the maximum wait time for CGI or lua requests in seconds. Requested executables are terminated if no output was generated. 60 Range
Web: Network timeout UCI: uhttpd.main.network_timeout Opt: network_timeout	Maximum wait time for network activity. Requested executables are terminated and the connection is shut down if no network activity occured for the specified number of seconds. 30 Range
Web: rfc 1918 filter UCI: uhttpd.main.rfc1918_filter Opt: rfc1918_filter	Enables option to reject requests from RFC1918 IPs to public server IPs (DNS rebinding counter measure). 0 Disabled 1 Enabled
Web: TLS protocol version UCI: uhttpd.main.tls_version Opt: tls_version	Defines the minimum supported TLS version for the https server. 1.0 1.1 1.2
Web: N/A UCI: uhttpd.main.realm Opt: realm	Defines basic authentication realm when prompting the client for credentials (HTTP 400). OpdenWrt Range

Web: N/A UCI: uhttpd.main.config Opt: config	Config file in Busybox httpd format for additional settings. Currently only used to specify basic auth areas. /etc/http.conf Range
Web: N/A UCI: uhttpd.main.index_page Opt: index_page	Index file to use for directories, for example, add index.php when using php.
Web: N/A UCI: httpd.main.error_page Opt: error_page	Virtual URL of file of CGI script to handle 404 requests. Must begin with '/' (forward slash).
Web: N/A UCI: uhttpd.main.no_symlinks Opt: no_symlinks	Does not follow symbolic links if enabled. 0 Disabled 1 Enabled
Web: N/A UCI: uhttpd.main.no_dirlists Opt: no_symlinks	Does not generate directory listings if enabled. 0 Disabled 1 Enabled

HTTP Server Using Command Line

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.

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.

HTTP server certificate settings

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

HTTP Server Using UCI

```
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
uhttpd.main.tls_version=1.0
```

HTTP Server using Package Options

```
root@VA_router:~# uci export uhttpd
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'
option tls_version '1.0'
```

6.11. 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.

6.12. 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
```

6.13. 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.

Configuration Packages Used

Package	Sections
luci	main
uds	script

Configuring Login Screen Customer 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.

Example login screen displaying serial and signal strength

Login Screen Customer 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%</h>dBm
```

7. Router File Structure

This section describes the file structure and location of essential directories and files on the router.

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
```

7.1. 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.

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
```

7.2. Identify Your Software Version

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

The status page showing software version

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.

7.3. 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”.

7.4. 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.

7.5. 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]
```

7.6. 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.

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
```

7.7. Exporting A Configuration File

A configuration file may be exported either through the UCI interface or through the web interface.

7.7.1. Exporting A Configuration File Using The Web Interface

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.

The flash operations page

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

7.7.2. 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
```

7.8. Importing A Configuration File

A configuration file may be imported either through the UCI interface or through the web interface.

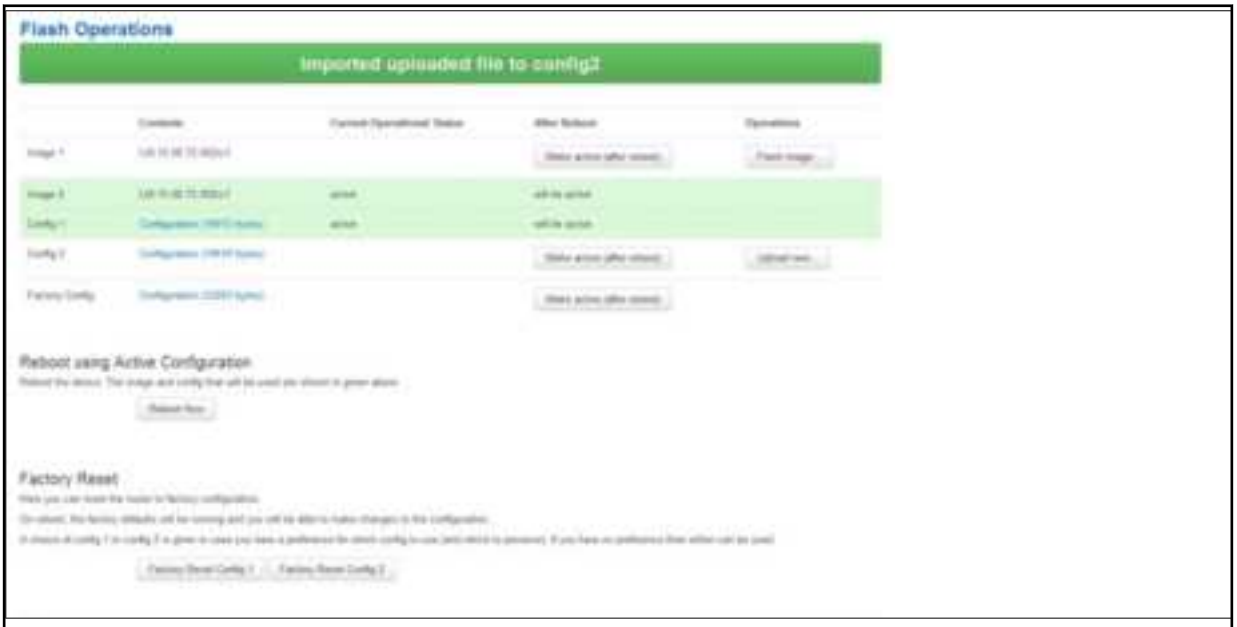
7.8.1. Importing A Configuration File Using The Web Interface

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.

The flash operations page

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



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.

7.8.2. 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.

8. Using The Command Line Interface

This chapter explains how to view the router 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.

8.1. Overview Of Some Common Commands

The router 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	[ksofttirqd_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
```

8.2. 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.

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
```

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.

8.3. 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'
```

8.4. 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
```

8.5. 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:~#
```

8.6. 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
```

8.7. 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
```

8.8. 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 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

8.9. 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.

9. 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.

The router firmware may be upgraded either through the UCI interface or through the web interface.

9.1. Upgrading Router Firmware

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

The flash operations page

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

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

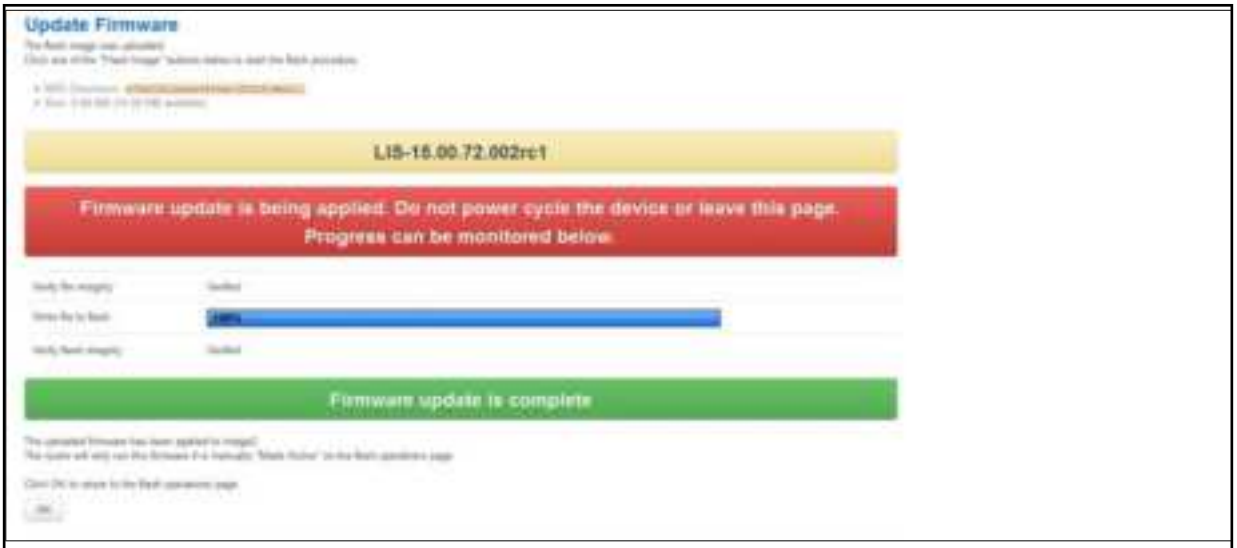
This process may take a while depending on the available connection speed. When the image has loaded, the Update Firmware page appears.

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.

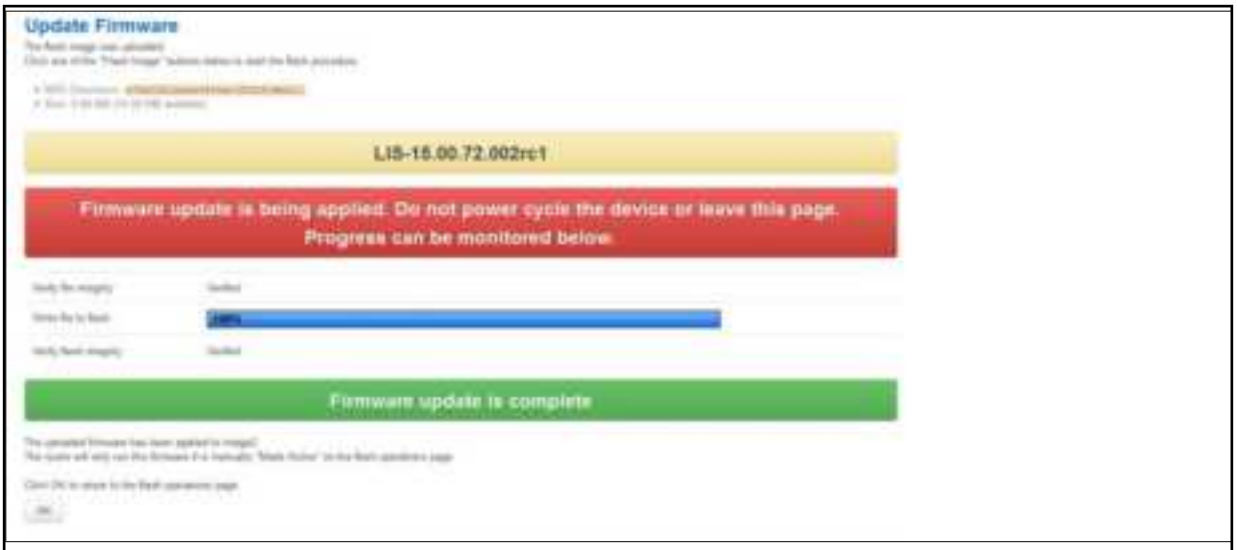
9.2. Flash Image And Do Not Reboot Option



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.

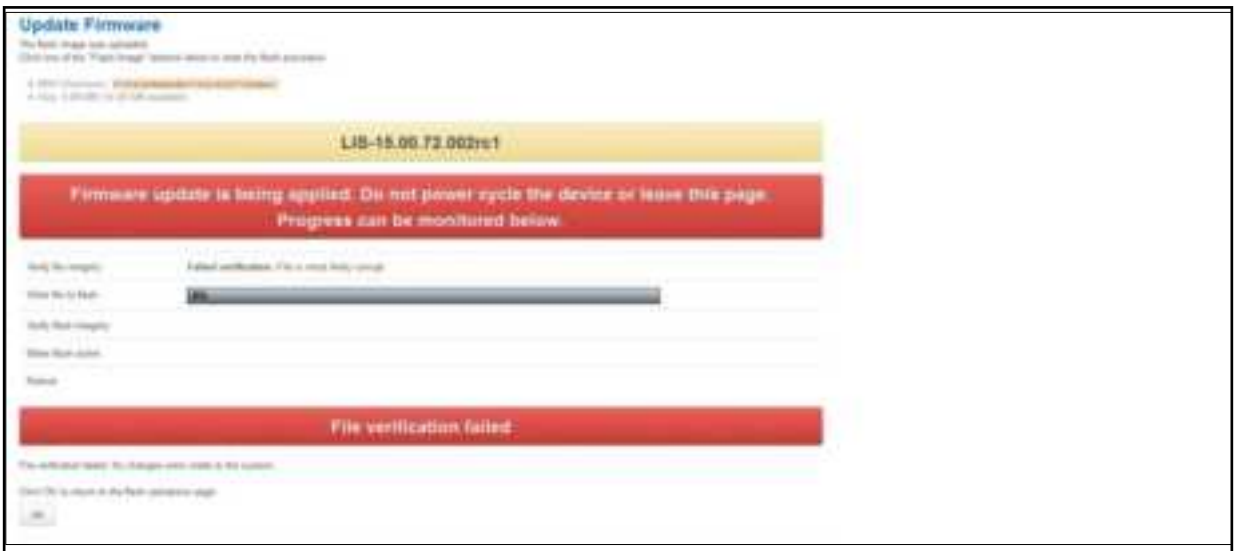
9.3. Update Flash Image And Reboot Using New Image Immediately Option



The firmware update page after 'update flash image an 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.

9.4. Possible File Corruption



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.

9.5. 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.

The system status list showing current firmware version

9.6. Upgrading Firmware Using CLI

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 session to the router.

Enter in the relevant username and password.

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

The output shows the available application:

TFTP using curl

Enter the following command:

```
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 isdifferent 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
```

9.6.1. 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.

9.6.2. 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.

9.6.3. Set Up 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
```

10. 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 bar. It also appears when you open a Telnet or SSH session.

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

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

10.1. Syslog Overview

Most syslog settings appear in the main System Configuration page.

Syslog messages have a timestamp, source facility, priority, and message section. Often the message section begins with an optional tag identifying the usermode program name and process ID responsible for the message.

Messages can be stored locally and also forwarded remotely. Separate filter options apply to each case. At a broad level, you can set the minimum severity level for local and remote targets; only messages with a priority more severe than the configured level will be recorded.

Kernel messages are recorded separately in their own buffer. However, for convenience, these are copied to the system log automatically so that a unified system log is available.

In addition, you can also define filter rules to determine how particular log messages are handled. For example, you may decide that certain debug messages are directed into their own log file, to avoid cluttering up the main system log, and to save bandwidth if delivering to a remote syslog server. You can define filters to be applied to local and remote targets, or both. A filter matches specific log messages and then determines an action for them.

Configuration Package Used

Package	Sections
System	main syslog_filter timeserver
luci	main

10.2. Configuring System Properties

To set your system properties, select **System -> System**. There are five sections in the System page.

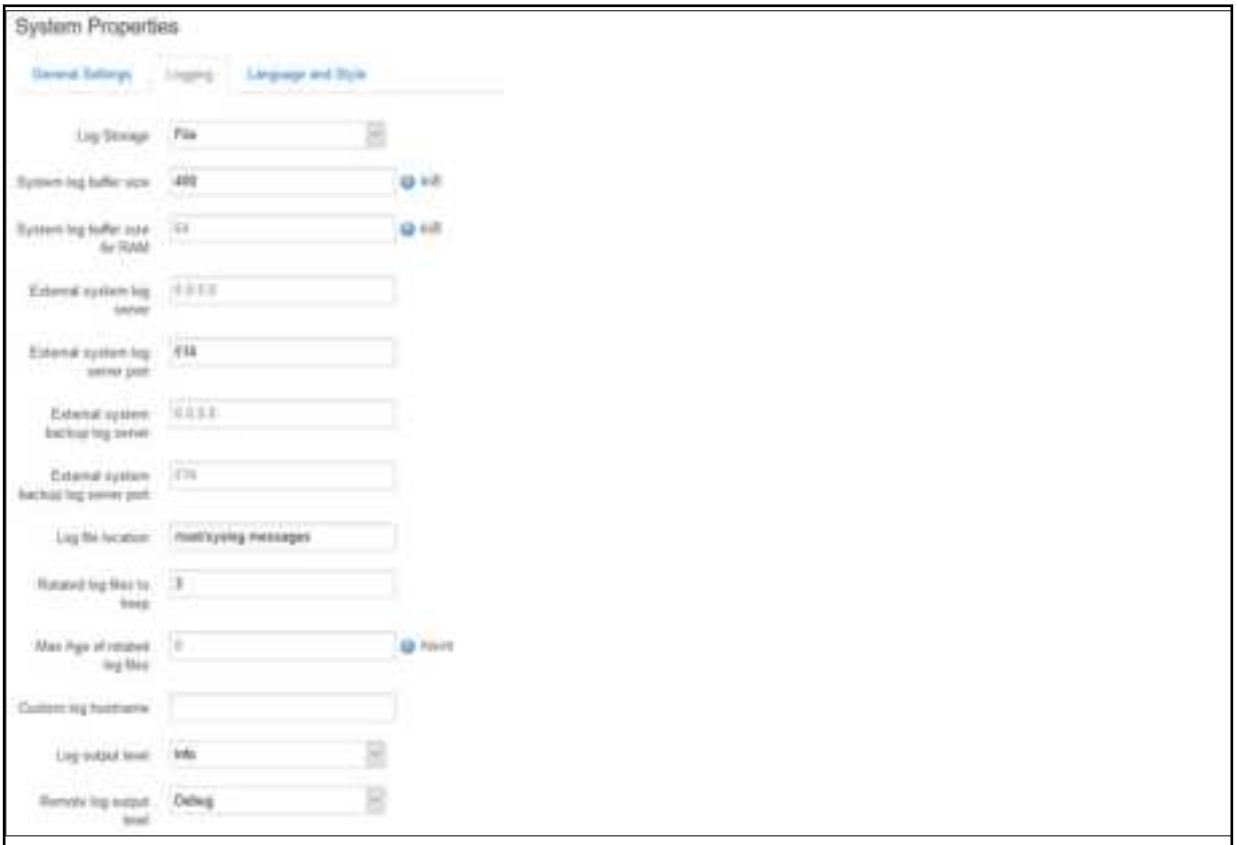
Section	Description
General settings	Configure host name, local time and time zone.
Logging	Configure a router to log to a server. You can configure a syslog client in this section.
Language and style	Configure the router's web language and style.
Time synchronization	Configure the NTP server in this section.
Audit configuration	Configures auditing of configuration changes and shell execution.

10.2.1. General Settings

General settings in system properties

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

10.2.2. Logging



The logging section in system properties

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Log storage

UCI: system.main.log_type

Opt: log_type</td><td>Defines the system log storage type. Messages stored in RAM can be seen using logread.

Note: system log stored in RAM will be lost on reboot.
<table>
<tr><th>Web value</th><th>Description</th><th>UCI</th></tr>
<tr><td>RAM</td><td>Store system log in RAM. Lost on reboot. Viewed using logread.</td><td>circular</td></tr>
<tr><td>File</td><td>Store system log in flash. Maintained through reboot.

Viewed using cat/log</td><td>file</td></tr>
</table></td></tr>
<tr><td>Web: System log buffer size

UCI: system.main.log_size

Opt: log_size</td><td>File log buffer size in KB.

Note: when the file reaches the configured size it is copied to the archive file (log_file_name.0).
<table>
<tr><td>Range</td><td></td></tr>
<tr><td>16</td><td>16KB</td></tr>
</table></td></tr>
<tr><td>Web: System log buffer size for RAM

UCI: system.main.log_size_ram

Opt: log_size_ram</td><td>RAM log buffer size in KB.
<table>
<tr><td>Range</td><td></td></tr>
<tr><td>16</td><td>16K</td></tr>
</table></td></tr>
<tr><td>Web: External system log server

UCI: system.main.log_ip

Opt: log_ip</td><td>External syslog server IP address. If defined, syslog messages will be sent in addition to local storage.</td></tr>
<tr><td>Web: External system log server port

UCI: system.main.log_port

Opt: log_port</td><td>External syslog server port number.
<table>
<tr><td>Range</td><td></td></tr>
<tr><td>514</td><td></td></tr>
</table></td></tr>
<tr><td>Web: External system backup log server

UCI: system.main.log_ip_backup

Opt: log_ip_backup</td><td>Backup external syslog server IP address. If defined, syslog messages will be sent here in addition to the main log server.
<table>
<tr><td>Range</td><td>IP or FQDN</td></tr>
<tr><td>0.0.0.0</td><td></td></tr>
</table></td></tr>
<tr><td>Web: External system backup log server port

UCI: system.main.log_port_backup

Opt: log_port_backup</td><td>External syslog server port number for use with backup server
<table>
<tr><td>Range</td><td></td></tr>
<tr><td>51</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Log file location

UCI: system.main.log_file

Opt: log_file</td><td>Defines the file path for log storage when log storage is set to ‘file’.

Note: when the file reaches the configured size it is copied to the archive file (log_file_name.0).

Set to: root/syslog.messages
<table>
<tr><td>Range</td><td></td></tr>
<tr><td>/root/syslog</td><td></td></tr>
</table></td></tr>
<tr><td>Web: Rotated log files to keep

UCI: system.main.log_file_count

Opt: log_file_count</td><td>Defines the file number of archive files for storage in flash when Log

Storage is set to ‘file’.

When the system log file reaches the configured size it is copied to the archive file (log_file_name.0). Existing archive files are copied to log_file_name.(x+1).
<table>
<tr><td>Range</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td></td><td><table><tr><td>1</td><td>Store 1 archive log file in flash</td></tr></table></td></tr>
<tr><td>Web: Max Age of rotated log files

UCI: system.main.log_age

Opt: log_age</td><td>Defines the maximum duration in hours before archive syslog files are deleted.

Set to 0 to define no age limit.

<table><tr><td>Range</td><td></td></tr><tr><td>0</td><td>No age limit</td></tr></table></td></tr>
<tr><td>Web: Custom log hostname

UCI: system.main.log_hostname

Opt: log_hostname</td><td>Defines a custom host name for syslog messages.

Magic values %hostname (system hostname), %ser (serial), and%mon (Monitor dev_reference) are also recognised.

<table><tr><td>Range</td><td></td></tr><tr><td>Empty</td><td>User router hostname for syslog messages.</td></tr></table></td></tr>
<tr><td>Web: Log output level

UCI: system.main.conloglevel

Opt: conloglevel</td><td>Sets the maximum log output level severity for system events. System events are written to the system log. Messages with a lower level or level equal to the configured level are displayed on the console using the logread command, or alternatively written to a flash file, 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: Remote log output level

UCI: system.main. remoteloglevel

Opt: remoteloglevel</td><td>Sets the maximum log output level severity for system events sent to remote syslog server.

<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: n/a

UCI: system.main.audit_shell

Opt: audit_shell</td><td>Log every command executed in shell.

<table><tr><td>1</td><td>Enable</td></tr><tr><td>0</td><td>Disable</td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: system.main.audit_cfg

Opt: audit_cfg</td><td>Log changes made to configuration file through any interface.

<table><tr><td>1</td><td>Enable</td></tr><tr><td>0</td><td>Disable</td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: system.main.audit_cfg_hul_interval_h ours

Opt: audit_cfg_hul_interval_hours</td><td>Defines the interval, in hours, at which configuration changes are uploaded to Activator.

Set to 0 to disable.

<table><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web Field/UCI/Package Option	Description
	6 — 6 hours
Web: n/a UCI: system.main.audit_cfg_max_size_kb Opt: audit_cfg_max_size_kb	Defines the maximum size audit data can take in flash in 1024 byte units. Range 1024 — 6 hours

10.2.3. Language And Style

The language and style section in system properties

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

10.2.4. Audit Configuration

The audit configuration section in system properties

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Track Config Changes

UCI: system.main.audit_cfg

Opt: audit_cfg</td><td>Any changes made to configuration file through any interface are logged to syslog.</td></tr>
<tr><td>Web: Changelog File Location

UCI: system.main.audit_cfg_log_file

Opt: audit_cfg_log_file</td><td>Defines the location of the configuration change log.

<table><tr><td>Range</td><td></td></tr><tr><td>/root/changelog</td><td></td></tr></table></td></tr>
<tr><td>Web: Logfile Size (kB)

UCI: system.main.audit_cfg_log_size

Opt: audit_cfg_log_size</td><td>Defines the maximum size of the configuration change log file in KB.

<table><tr><td>Range</td><td></td></tr><tr><td>200</td><td>200KB</td></tr></table></td></tr>
<tr><td>Web: Rotated Changelog Files to Keep

UCI: system.main.audit_cfg_log_count

Opt: audit_cfg_log_count</td><td>Defines the maximum number of configuration change log files to store.

<table><tr><td>Range</td><td></td></tr><tr><td>4</td><td>Stoer 4 changelog files before rotating.</td></tr></table></td></tr>
<tr><td>Web: Audit Data Max Size (kB)

UCI: system.main.audit_cfg_max_size_kb

Opt: audit_cfg_max_size_kb</td><td>Defines the maximum size audit data can take in flash in KB.

<table><tr><td>Range</td><td></td></tr><tr><td>1024</td><td></td></tr></table></td></tr>
<tr><td>Web: Config Changes Upload Interval

UCI: system.main.audit_cfg_hul_interval_hours

Opt: audit_cfg_hul_interval_hours</td><td>Defines the interval, in hours, at which configuration change messages are uploaded to Activator.

Set to 0 to disable.

<table><tr><td>Range</td><td></td></tr><tr><td>6</td><td>6 hours</td></tr></table></td></tr>
<tr><td>Web: Shell Audit

UCI: system.main.audit_shell

Opt: audit_shell</td><td>Every command executed in shell is logged to syslog.

<table><tr><td>1</td><td>Enabled</td></tr><tr><td>0</td><td>Disabled</td></tr></table></td></tr>
<tr><td>Web: Web Audit

UCI: luci.main.audit_req

Opt: audit_req</td><td>Enables logging management web access to syslog.

<table><tr><td>1</td><td>Enabled</td></tr><tr><td>0</td><td>Disabled</td></tr></table></td></tr>
<tr><td>Web: Web Audit Level

UCI: luci.main.audit_shell

Opt: audit_level</td><td>Defines the type of web operation to be logged to syslog.

<table><tr><th>Web value</th><th>Description</th><th>UCI</th></tr><tr><td>1</td><td>Actions invoked via web page</td><td>1</td></tr><tr><td>2</td><td>Config and status pages</td><td>2</td></tr><tr><td>3</td><td>config , status and polled pages</td><td>3</td></tr><tr><td>4</td><td>comprehensive URL logging</td><td>4</td></tr></table></td></tr>
</table>

10.2.5. Time Synchronization

The router time must be synchronized 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.

The time synchronization section in system properties

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: NTP update interval

UCI: system.ntp.interval_hours

Opt: interval_hours</td><td>Specifies interval of NTP requests in hours. Default value set to auto.

<table><tr><td>Auto</td><td></td></tr><tr><td>Range</td><td>auto; 1-23</td></tr></table></td></tr>
<tr><td>Web: NTP server candidates

UCI: system.ntp.server

Opt: list server</td><td>Defines the list of NTP servers to poll the time from. If the list is empty, the built-in NTP daemon is not started. Multiple servers can be configured and are separated by a space if using UCI.

By default all fields are set to 0.0.0.0.</td></tr>
<tr><td>Web: Max Round-Tip Time (secs)

UCI: system.ntp.max_ntp_roundtrip_sec Opt: max_ntp_roundtrip_sec</td><td>Defines the maximum time in seconds for an NTP poll. Any polls that take longer than this will be not be used for NTP calculation.

<table><tr><td>2</td><td>Two seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: NTP Server Interface

UCI: system.ntp.listen

Opt: listen</td><td>Defines a list of interfaces that respond to NTP requests. Interfaces should be delimited using space. Example:

option listen ‘LAN1 LAN2’

<table><tr><td>Blank</td><td>Do not respond to NTP requests</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: NTP Server Stratum UCI: system.ntp.stratum

Opt: stratum</td><td>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.

<table><tr><td>Blank</td><td>NTP derive stratum</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: NTP Source Combine Limit

UCI: system.ntp. combinelimit

Opt: combinelimit</td><td>Defines whether to limit sources included in the combining algorithm.

When <code>chronyd</code> has multiple sources available for synchronization, it has to select one source as the synchronization source. The measured offsets and frequencies of the system clock relative to the other sources, however, can be combined with the selected source to improve the accuracy of the system clock.

The <code>combinelimit</code> directive limits which sources are included in the combining algorithm. Their synchronization distance has to be shorter than the distance of the selected source multiplied by the value of the limit. Also, their measured frequencies have to be close to the frequency of the selected source.

<table><tr><td>3</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

10.2.6. Console Login Banner

To configure a message that is displayed after login via SSH, telnet or console, in the top menu, select **System -> Administration.** Navigate to the Console login banner section.

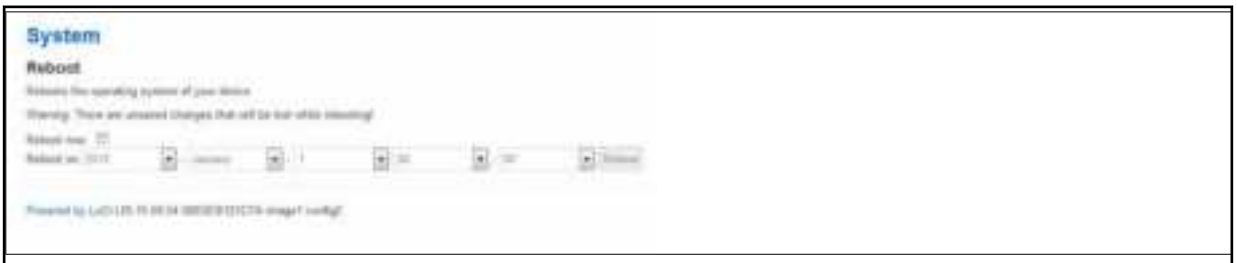
The console login banner in system section

Web Field/UCI/Package Option	Description
Web: Console login banner UCI: system.main.banner list: banner	Defines a login banner that is displayed after log in via SSH, telnet or console

10.2.7. 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.



The system settings reboot page

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

10.3. System Settings Using Command Line

System settings are configured under the system package **/etc/config/system**. There are several configuration sections.

Section	Description
system	General system configuration options
timeserver	Router time and NTP configuration options
syslog_filter	Advanced filter rules (see Advanced filter section)

10.3.1. 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.remoteloglevel=8

system.main.log_file=/root/syslog.messages

system.main.log_size=400

system.main.log_type=file

system.main.log_file_count=3

system.main.conloglevel=8

system.main.cronloglevel=8

system.main.banner=This is a test banner system.ntp.interval_hours=auto

system.ntp.server=0.VA_router.pool.ntp.org 10.10.10.10

system.ntp.combinelimit=3
```

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 remoteloglevel '8'
option log_file '/root/syslog.messages'
option log_size '400'
option log_type 'file'
option log_file_count '3'
option time_save_interval_min "10"
option conloglevel '8'
option cronloglevel '8'
list banner `This is a test banner`
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' option combinelimit '3'
```

10.4. System Diagnostics

System log messages

System log messages comprise of a date, source facility, hostname, severity and message description in the form tag: message.

Facility	Description
auth	Authorisation/security
authpriv	Authorisation (private)
cron	Scheduled jobs
daemon	Background daemons
kern	Kernel messages
local0	hotplug scripts
security	Same as auth
syslog	Internal syslog events
user	General user-mode application messages

Event severity list

Level	Name	Description
0	emerg	System is unusable
1	alert	Immediate action required
2	crit	Critical conditions
3	error	Error conditions
4	warning	Warning conditions
5	notice	Normal but significant
6	info	Informational
7	debug	Debug-level messages
-	none	No priority

10.4.1. System Log Messages In RAM

By default, system log messages are stored in the system log in RAM. To view the system log in RAM, 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.

10.4.2. System Log Messages In Flash

Since logread is limited by memory size and does not survive a reset, it is beneficial to write system messages to flash memory. To do this, modify the system config under the system package.

Set the options **'log_file'**, **'log_size'**, **'log_type'**and **'log_file_count'** as shown 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'

option log_file_count '3'
```

The above commands will take effect after a reboot, or by running the console command:

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

```
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**.

10.4.3. Kernel Messages

To view kernel messages, enter `dmesg`

```
root@VA_router:~# dmesg

[ 0.000000] Linux version 3.10.12 (info@virtualaccess.com) (gcc version 4.8.1 20130401 (prerelease) (Linaro GCC 4.8-2013.04) )
#130 PREEMPT 1970-

01-01T00:00:00Z
```

```
[ 0.000000] SoC: xRX330 rev 1.1

[ 0.000000] bootconsole [early0] enabled

[ 0.000000] CPU0 revision is: 00019556 (MIPS 34Kc)

[ 0.000000] adding memory size:267386880 from DT

[ 0.000000] MIPS: machine is Virtual Access GW6600V series [ 0.000000] Determined physical RAM map:

[ 0.000000] memory: 0ff00000 @ 00000000 (usable) [ 0.000000] User-defined physical RAM map:

[ 0.000000] memory: 07200000 @ 00000000 (usable)
```

NOTE
Kernel messages are also copied to the main system log by default.

10.4.4. Syslog Process

To check the syslog process is running correctly, enter pgrep –fl syslogd

```
root@VA_router:~# pgrep -fl syslogd
5409 /sbin/syslogd -h VARouter -L -R 192.168.14.202:514 -l 7 -r 8 -s 400 -O
/root/syslog.messages -b 3 -C64 -R localhost:2048
```

Changes to the syslog configuration will take effect with a restart of syslogd

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

10.4.5. NTP Process

To check the NTP process is running correctly, enter pgrep –fl chrony

```
root@VA_router:~# pgrep -fl chrony
2553 /usr/sbin/chronyd -f /etc/chrony.conf
```

Changes to the NTP configuration will take effect with a restart of chrony

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

10.5. Advanced Filtering Of Syslog Messages

Syslog messages can be filtered against a series of rules that are checked for each message generated. If a match is found, then the specified action is taken. If no match occurs, then the default action is taken, as defined in the main system logging settings.

A message may match multiple filters. They are processed in the order listed. For example, you may wish to record authorisation messages in the main system log, but also make a copy in a separate authorisation log which can span a much longer period of time.

By default, all matching filters will be applied to each message. However, you can mark a filter to indicate that after it matches, no further filter processing should take place.

The filter rules are defined in a free-form text list in the syslog_filter configuration section. There are two section types, one for messages to be stored locally, and one for messages delivered remotely.

Configuring advanced filters on the web interface is not currently supported; they must be edited using the command line interface.

10.5.1. Advanced Filtering Using Command Line

Filters are defined in the syslog_filter configuration section of the system package. A set of filters can be either local or remote.

- All messages are matched against both local and remote filter rules, if configured.
- Each local filter matched is executed; if there is no match, then the default local logging action applies.
- Any remote filter matched is executed; if there is no match, then the default remote logging action applies.

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

package system


config syslog_filter 'local'

list text "...line 1..."

list text "...line 2..."

list text "...line 3..."

config syslog_filter 'remote'

list text "...line 1..."

list text "...line 2..."

list text "...line 3..."
```

Lines defined here are copied to the router runtime file **/var/conf/syslog.conf** which may be reviewed to determine current rules in use.

10.5.2. Filter Definitions

Each filter ruleset is a series of lines. Each line can be:

- A filter pattern, of the form facility.[op]severity(pattern) target [~]
- A blank line, or comment line, starting with hash (**#**).

If a message does not match any of the filter lines for a destination, local or remote, the default action for that destination is taken.

The sections of a filter pattern break down as shown in the following table.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>facility</td><td>Any keyword or comma-separated list of keywords from the source facility list. See the Source Facilities table in section 6.5.1.1.

Use the wildcard '*' to match all facilities.</td></tr>
<tr><td>severity</td><td>Any keyword from the event severity list (see Event Severity table above). The rule will match all severities more urgent f the message severity level is at least as urgent as this.

Use the wildcard '*' to match all facilities.</td></tr>
<tr><td>op</td><td>Defines an optional severity condition.

<table>
<tr><td>(empty)</td><td>match listed severity, and also anything more severe</td></tr>
<tr><td>!</td><td>match on less urgent severities than that listed</td></tr>
<tr><td>=</td><td>severity must match exactly</td></tr>
<tr><td>!=</td><td>match any severity other than the listed severity</td></tr>
</table>

Examples:

*.debug matches all messages of debug severity and greater (i.e. debug, info, warning, etc

*.=debug matches all debug messages.</td></tr>
<tr><td>pattern</td><td>Defines an optional pattern to match against the message text. The pattern is used to restrict the number of log messages matching this filter.

The pattern syntax is a simple case-insensitive regular expression, using these characters:

<table>
<tr><td>*</td><td>Matches zero or more characters.</td></tr>
<tr><td>?</td><td>Matches any single character (use this for spaces).</td></tr>
<tr><td>!</td><td>Matches anything not matching the following pattern.</td></tr>
<tr><td>^</td><td>Matches the start of a message.</td></tr>
<tr><td>$</td><td>Matches the end of a message.</td></tr>
</table>

Examples:

<table>
<tr><td>(firewall:)</td><td>Match any message containing the string 'firewall:'</td></tr>
<tr><td>(up*eth1)</td><td>Match any UP message referencing eth1</td></tr>
<tr><td>(!mobile)</td><td>Match only messages that do not include the string 'mobile'</td></tr>
<tr><td>(^mobile)</td><td>Match only messages beginning with the string 'mobile'</td></tr>
</table></td></tr>
<tr><td>target</td><td>Defines what to do with the log message when a match occurs. It is optional for remote filters. It can be the name of a disk file, or one of the special target keywords listed below.

<table>
<tr><td>default</td><td>Do whatever the default action is, as if not the filter rule is matched.</td></tr>
<tr><td>ignore</td><td>Never log this message (useful for remote filtering).</td></tr>
<tr><td>console</td><td>Log this message to the console. To view the console use cat/proc/conlog</td></tr>
<tr><td>mem</td><td>Log this message to the memory buffer (logread), if configured.

Note: logread is not stored through reboot.</td></tr>
</table></td></tr>
<tr><td>~</td><td>Optional flag to indicate no further filters should be checked, if this filter matches. This prevents later filters from acting on the same message. For convenience this is automatically implied when a target of ignore is used. A space must be present before the ~ character.

<table>
<tr><td>~</td><td>No further filters should be checked after a match.</td></tr>
<tr><td>(empty)</td><td>Continue checking other filters after a match.</td></tr>
</table></td></tr>
</table>

Filter Examples

Example 1

Log all debug messages to memory buffer. Do not log anywhere else locally.

Log all authorisation facility messages to filepath ‘var/log/auth’. Do not log anywhere else locally.

Log all ipsec messages to filepath ‘va/log/ipsec’. Do not log anywhere else locally.

For everything else, apply default local logging.

No remote filter rules defined, so apply default remote logging to all messages.

```
config syslog_filter 'local'
list text '*.=debug mem ~'
list text 'auth,authpriv.*  /var/log/auth  ~'
list text '*.*(ipsec:) /var/log/ipsec ~'
```

Example 2

As Example 1 but in addition to specified local files, copy auth, authpriv and ipsec to local default log.

```
config syslog_filter 'local'
list text '*.=debug mem ~'
list text 'auth,authpriv.*  /var/log/auth'
list text '*.*(ipsec:) /var/log/ipsec'
list text '*.* default'
```

Example 3

As in Example 2, except **do not** send any auth or auth priv messages remotely.

```
config syslog_filter 'local'
list text '*.=debug mem ~'
list text 'auth,authpriv.*  /var/log/auth'
list text '*.*(ipsec:) /var/log/ipsec'
list text '*.* default'   config syslog_filter 'remote'
```

Example 4

As in Example 3, except only send auth or auth priv messages remotely.

```
config syslog_filter 'local'
list text '*.=debug mem ~'
list text 'auth,authpriv.* /var/log/auth'
list text '*.*(ipsec:) /var/log/ipsec'
list text '*.* default'   config syslog_filter 'remote'
list text 'auth,authpriv.* ~'
list text '*.* ignore'
```

11. Configuring An Ethernet Interface

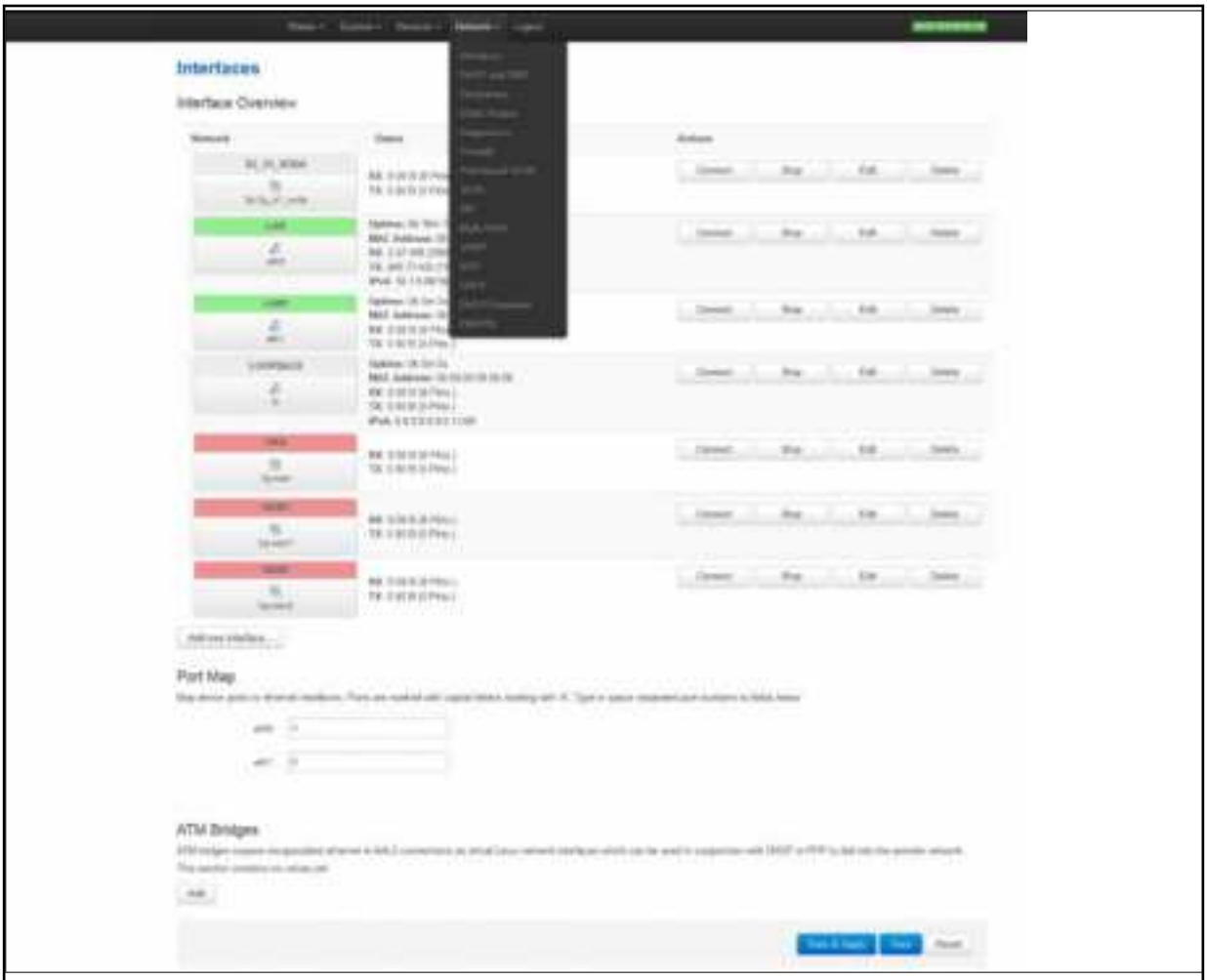
This chapter describes how to configure an Ethernet interface including configuring the interface as a DHCP server, adding the interface to a firewall zone, mapping the physical switch ports and defining loopback interface.

Configuration Packages Used

Package	Sections
Network	interface
	route
	va_switch
	alias
firewall	zone
dhcp	dhcp

11.1. 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.



The interfaces overview page

There are three sections in the Interfaces page.

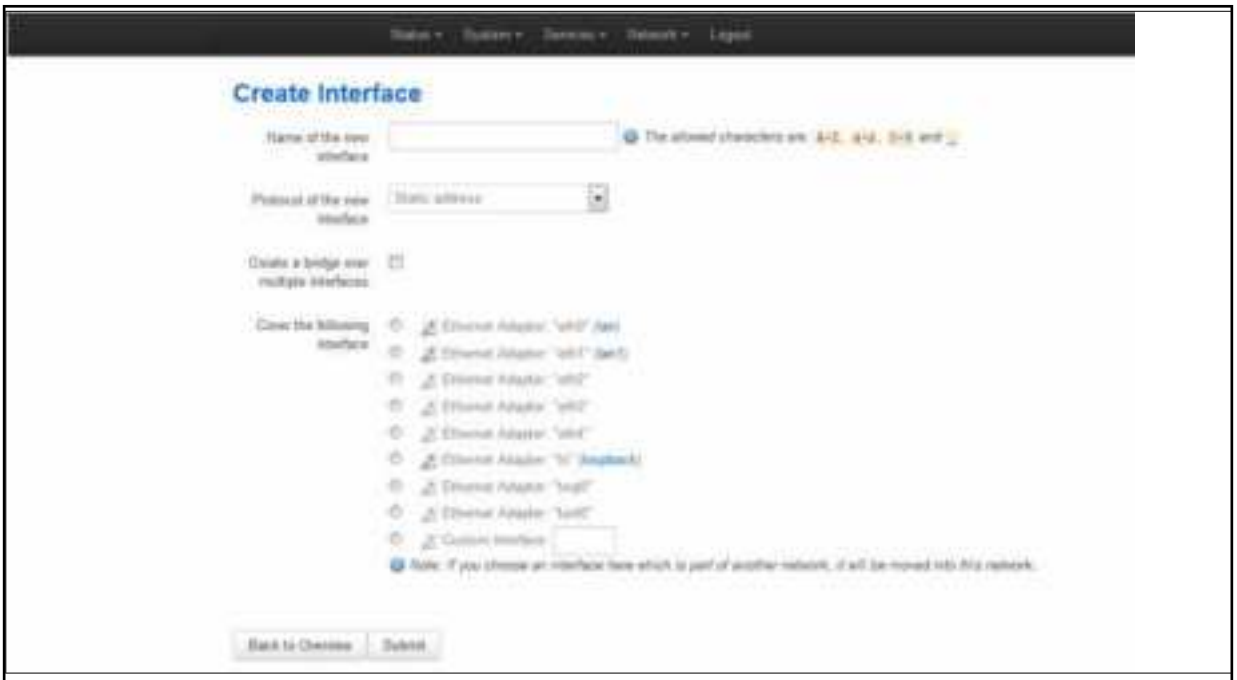
Section	Description
Interface Overview	Shows existing interfaces and their status. You can create new, and edit existing interfaces here.
Port Map	In this section you can map device ports to Ethernet interfaces. Ports are marked with capital letters starting with 'A'. Type in space-separated port character 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.

11.1.1. 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.

11.1.2. 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.



The create interface page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Name of the new interface

UCI: network.<if name>

Opt: config interface</td><td>Assigns a logical name to the interface. The network interface section will assign this name (<if name>).

Type the name of the new interface. Allowed characters are A-Z, a-z, 0-9 and _</td></tr>
<tr><td>Web: Protocol of the new interface

UCI: network.<if name>.proto
Opt: proto</td><td>Specifies what protocol the interface will operate on. Select Static.
<table>
<tr><th>Web</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>none</td></tr>
<tr><td>IPv6-in-IPv4 (RFC4213)</td><td>Used with tunnel brokers.</td><td></td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td><td>gre</td></tr>
<tr><td>IOT</td><td>IOT</td><td>iot</td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td><td>i2tp</td></tr>
<tr><td>L2TPv3</td><td>L2TPv3 Tunnelling Protocol</td><td>i2tpv3</td></tr>
<tr><td>PPP</td><td>Point to Point Tunnelling protocol</td><td>ppp</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td><td>pppoe</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td><td>pppoa</td></tr>
<tr><td>LTE/UMTS/GPRS/EV-DO</td><td>CDMA, UMTS, or GPRS connection using an AT-style 3G modem</td><td>3g</td></tr>
<tr><td>PPP (PSTN-Modem)</td><td>PPP v90 modem</td><td>pppmodem</td></tr>
</table></td></tr>
<tr><td>Web: Create a bridge over multiple interfaces

UCI: network.<if name>.type
Opt: type</td><td>If you select this option, then the new logical interface created will act as a bridging interface between the chosen existing physical interfaces.
<table>
<tr><td>Empty</td><td></td></tr>
<tr><td>Bridge</td><td>Configures a bridge over multiple interfaces</td></tr>
</table></td></tr>
<tr><td>Web: Cover the following interface

UCI: network.<if name>.ifname

Opt: ifname</td><td>Physical interface name to assign to this logical interface. If creating a bridge over multiple interfaces select two interfaces to bridge. When using UCI the interface names should be separated by a space e.g. option ifname ‘eth2 eth3’</td></tr>
</table>

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.

11.1.3. 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.

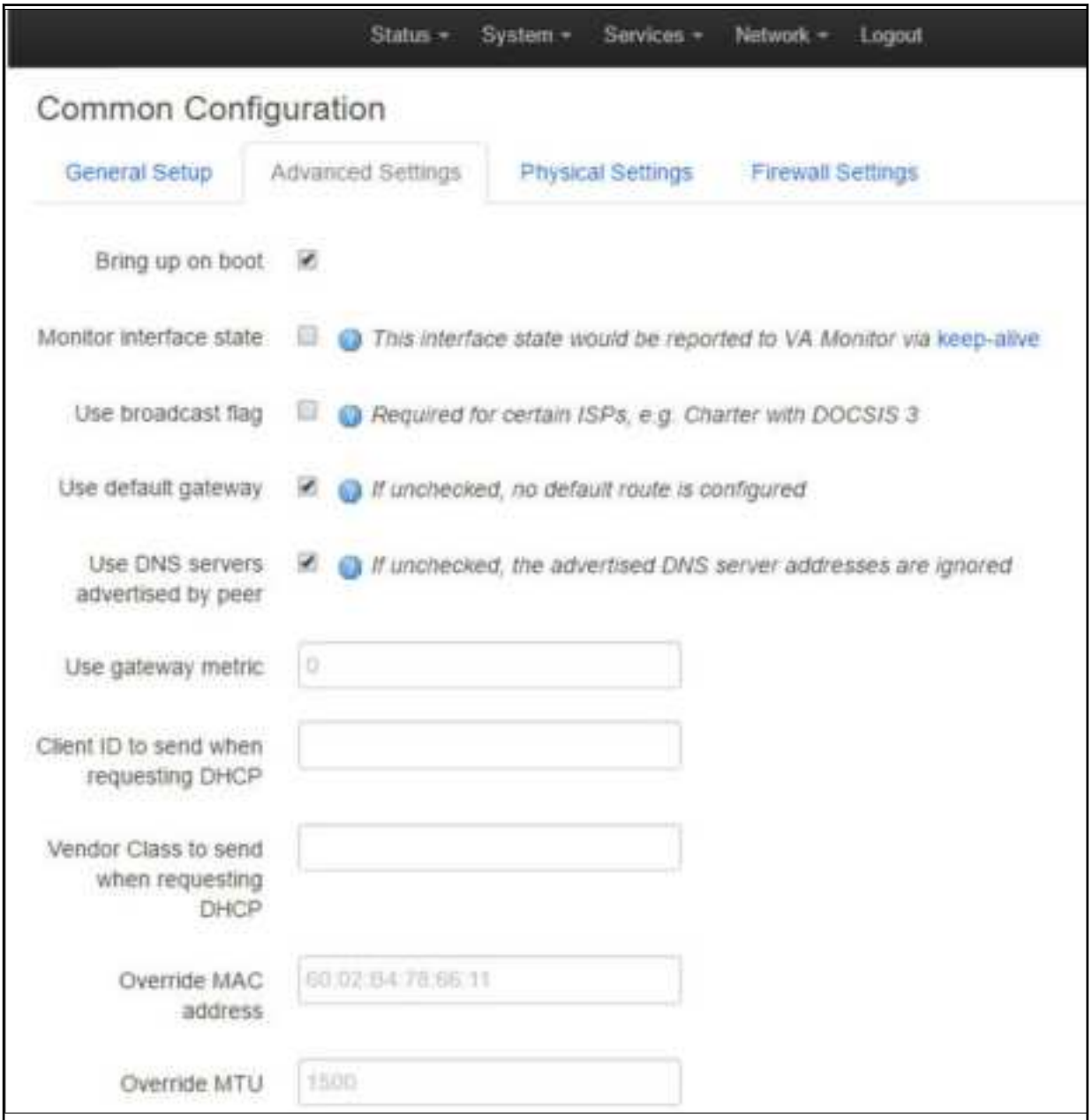
Common Configuration: General Setup

The Ethernet connection common configuration settings page

Web Field/UCI/Package Option	Description
Web: Status	Shows the current status of the interface.
Web: Protocol UCI: network.<if name>.proto Opt: proto	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. Web | Description | UCI Static | Static configuration with fixed address and netmask. | static DHCP Client | Address and netmask are assigned by DHCP. | dhcp Unmanaged | Unspecified | none IPv6-in-IPv4 (RFC4213) | Used with tunnel brokers. | IPv6-over-IPv4 | Stateless IPv6 over IPv4 transport. | GRE | Generic Routing Encapsulation protocol | gre IOT | IOT | iot L2TP | Layer 2 Tunnelling Protocol | i2tp L2TPv3 | L2TPv3 Tunnelling Protocol | l2tpv3 PPP | Point to Point protocol | ppp PPtP | Point to Point Tunnelling protocol | pptp PPPoE | PPP over Ethernet | pppoe PPPoATM | PPP over ATM | pppoa LTE/UMTS/GPRS/EV-DO | CDMA, UMTS or GPRS connection using an AT-style 3G modem | 3g PPP(PSTN-Modem) | PPP v90 modem | pppmodem
Web: IPv4 address UCI: network.<if name>.ipaddr Opt: ipaddr	The IPv4 address of the interface. This is optional if an IPv6 address is provided.
Web: IPv4 netmask UCI: network.<if name>.netmask Opt: netmask	Subnet mask to be applied to the IP address of this interface.
Web: IPv4 gateway UCI: network.<if name>.gateway Opt: gateway	IPv4 default gateway to assign to this interface (optional).
Web: IPv4 broadcast UCI: network.<if name>.broadcast Opt: broadcast	Broadcast address. This is automatically generated if no broadcast address is specified.
Web: Use custom DNS servers UCI: network.<if name>.dns Opt: list dns	List of DNS server IP addresses (optional). Multiple DNS Servers are separated by a space if using UCI.
Web: Accept router advertisements UCI: network.<if name>.accept_ra Opt: accept_ra	Specifies whether to accept IPv6 Router Advertisements on this interface (optional). Note: default is 1 if protocol is set to DHCP, otherwise defaults to 0.
Web: Send router solicitations UCI: network.<if name>.send_rs Opt: send_rs	Specifies whether to send Router Solicitations on this interface (optional). Note: defaults to 1 for static protocol, otherwise defaults to 0.
Web: IPv6 address UCI: network.<if name>.ip6addr	The IPv6 IP address of the interface. Optional if an IPv4 address is provided. CIDR notation for the IPv6 address is required.

Web Field/UCI/Package Option	Description
Opt: ip6addr	
Web: IPv6 gateway UCI: network.<if name>.ip6gw Opt: ip6gw	Assign given IPv6 default gateway to this interface (optional).

Common Configuration: Advanced Settings



The Ethernet connection advanced settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Bring up on boot

UCI: network.<if name>.auto

Opt: auto</td><td>Enables the interface to connect automatically on boot up.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Monitor interface state

UCI: network.<if name>.monitored

Opt: monitored</td><td>Enabled if status of interface is presented on Monitoring platform.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Override MAC address

UCI: network.<if name>.macaddr

Opt: macaddr</td><td>Override the MAC address assigned to this interface. Must be in the form: hh:hh:hh:hh:hh:hh, where h is a hexadecimal number.</td></tr>
<tr><td>Web: Override MTU

UCI: network.<if name>.mtu

Opt: mtu</td><td>Defines the value to override the default MTU on this interface.

<table><tr><td>1500</td><td>1500 bytes</td></tr></table></td></tr>
<tr><td>Web: Use gateway metric

UCI: network.<if name>.metric

Opt: metric</td><td>Specifies the default route metric to use for this interface (optional).

<table><tr><td>0</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Dependant Interfaces

UCI: network.[..x..].dependants

Opt: dependants</td><td>Lists interfaces that are dependant on this parent interface.

Dependant interfaces will go down when parent interface is down and will start or restart when parent interface starts.

Separate multiple interfaces by a space when using UCI

Example: options dependants 'PPPADSL MOBILE'

This replaces the following previous options in child interfaces.

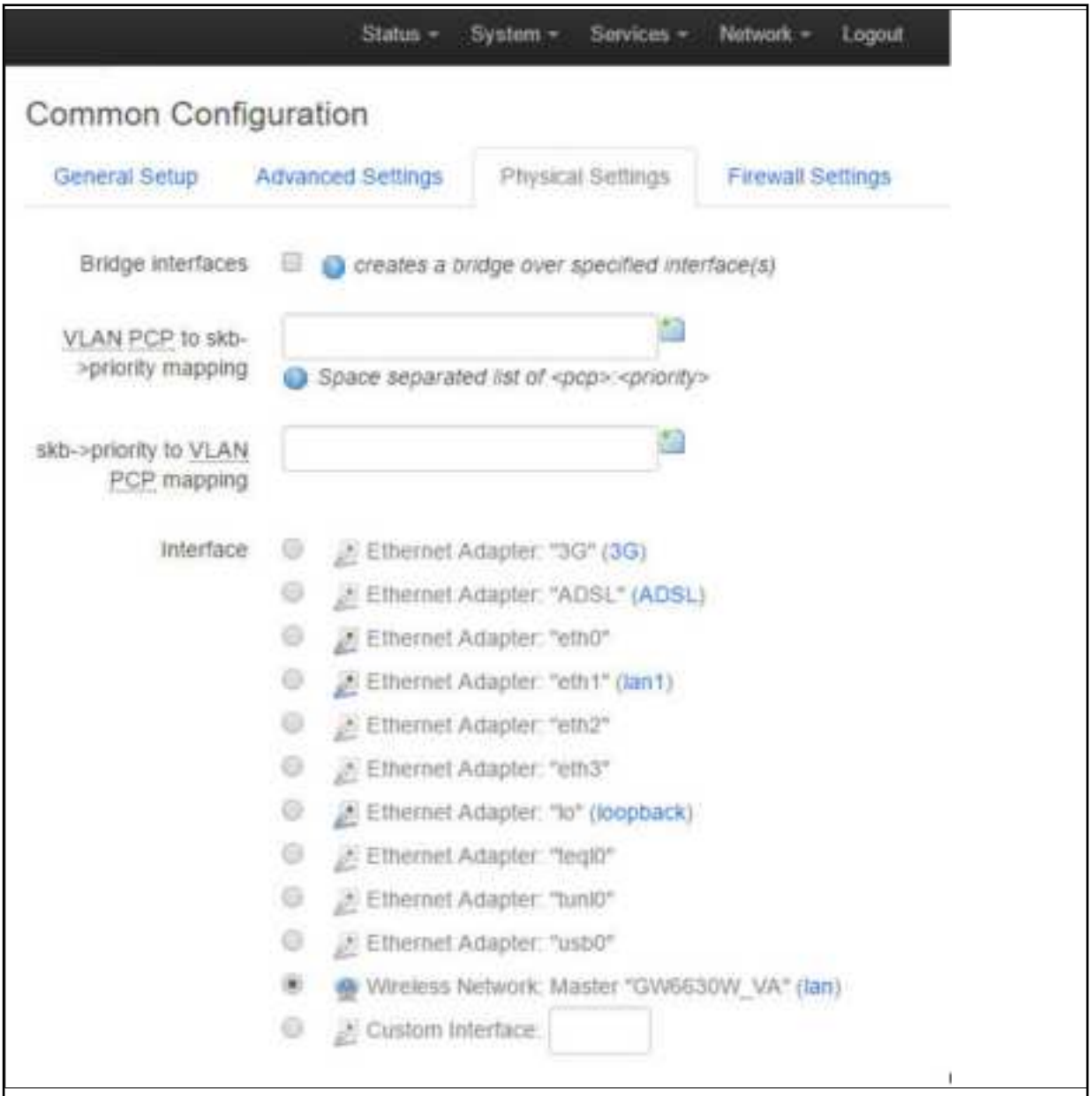
<table><tr><td>gre</td><td>option local_interface</td></tr><tr><td>lt2p</td><td>option src_ipaddr</td></tr><tr><td>iot</td><td>option wan1 wan2</td></tr><tr><td>6in4</td><td>option ipaddr</td></tr><tr><td>6t04</td><td>option ipaddr</td></tr></table></td></tr>
<tr><td>Web: SNMP Alias ifindex

UCI: network.[..x..].snmp_alias_ifindex

Opt: snmp_alias_ifindex</td><td>Defines a static SNMP interface alias index for this interface, that can be polled via the SNMP interface index (snmp_alias_ifindex+1000). See ‘Configuring SNMP’ section for more information.

<table><tr><td>Blank</td><td>No SNMp interface alias index</td></tr><tr><td>Range</td><td>0 - 4294966295</td></tr></table></td></tr>
</table>

Common Configuration: Physical Settings



The common configuration physical settings page

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

Loopback Interfaces

Loopback interfaces are defined in exactly the same way as Ethernet interfaces. For more information, read the section above.

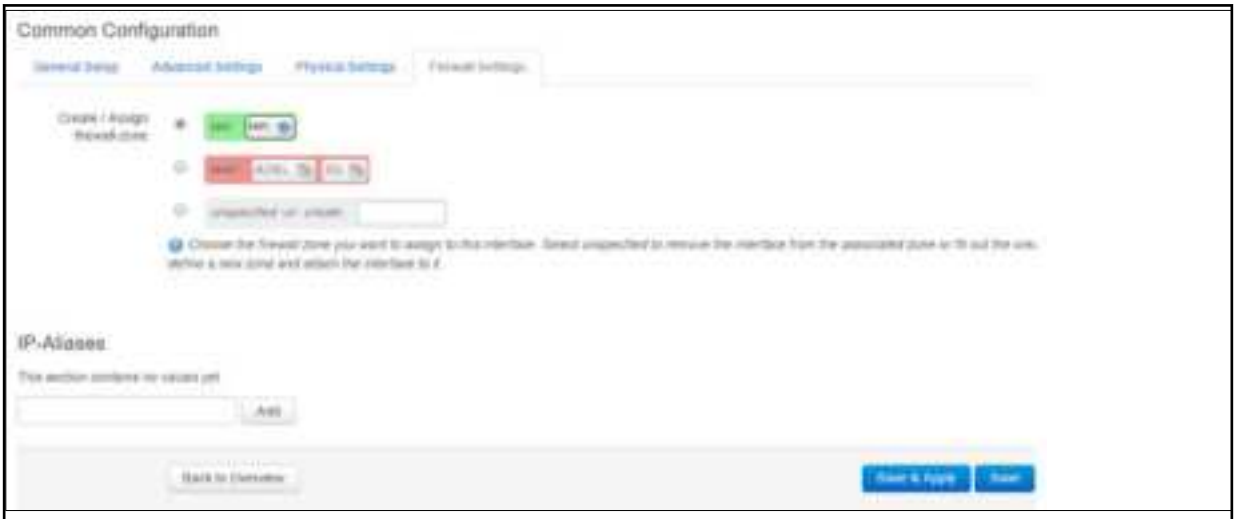
NOTE

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

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.



The common configuration GRE firewall settings page

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.

The common configuration GRE firewall settings page

11.2. Interface Overview: IP-Aliases

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

IP-alias Packages

Package	Section
Network	alias

11.2.1. 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'.

The IP-Aliases section

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

After you have clicked **Add**, the IP-Aliases configuration options page appears.

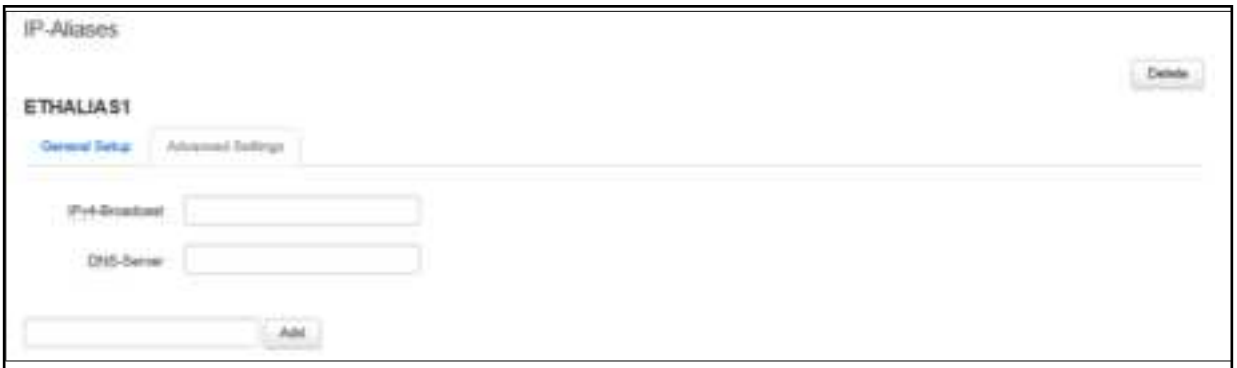
The IP- Alias page is divided into two sub sections: general setup and advanced.

11.2.2. IP-Aliases: General Setup

The IP-Aliases general setup page

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

11.2.3. IP-Aliases: Advanced Settings



The IP-Aliases advanced settings section

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

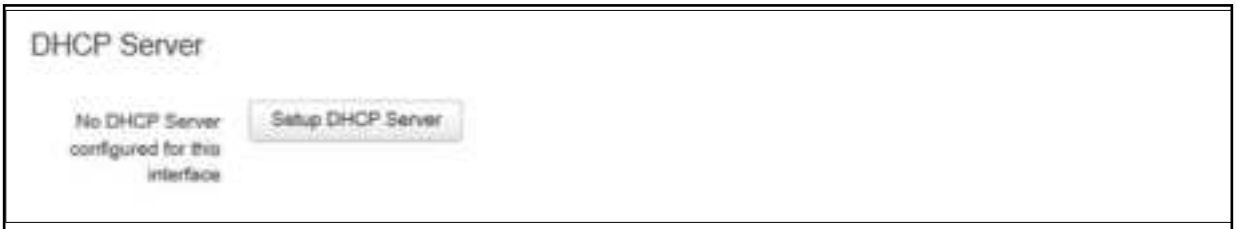
11.3. Interface Overview: DHCP Server

NOTE
This option is only available for interfaces with a static IP address.

DHCP Server: Packages

Package	Sections
dhcp	dhcp

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



The DHCP server settings section

The DHCP Server configuration options appear. The DHCP Server is divided into two sub-sections: General Setup and Advanced Settings.

11.3.1. DHCP Server: General Setup

The DHCP server general setup section

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Ignore interface

UCI: dhcp.@dhcp[x].ignore

Opt: ignore</td><td>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.

<table><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td></tr>
<tr><td>Web: Mode

UCI: dhcp.@dhcp[x].mode

Opt: mode</td><td>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.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>DHCPv4</td><td>DHCP for IPv4</td><td>ipv4</td></tr><tr><td>DHCPv6</td><td>DHCP for IPv6</td><td>ipv6_dhcp</td></tr><tr><td>IPv6 Router Advertisments</td><td>IPv6 RA</td><td>ipv6_ra</td></tr><tr><td>DHCPv6 Prefix Delegation</td><td>DHCPv6 prefix delegation</td><td>ipv6_pd</td></tr></table></td></tr>
<tr><td>Web: Start

UCI: dhcp.@dhcp[x].start

Opt: start</td><td>Defines the offset from the network address for the start of the DHCP pool.

Example: for network address 192.168.100.10/24, start=100, DHCP allocation pool will start at 192.168.100.100.

For subnets greater than /24, it may be greater than 255 to span subnets. Alternatively, specify in IP address notation using the wildcard ‘0’ where the octet is required to inherit bits from the interface IP address.

Example: to define a DHCP scope starting from 10.1.20.0 on an interface with 10.1.0.0/16 address, set start to 0.0.20.1100

<table><tr><td>100</td><td></td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Limit

UCI: dhcp.@dhcp[x].limit

Opt: limit</td><td>Defines the size of the address pool.

Example: for network address 192.168.100.10/24, start=100, limit=150, DHCP allocation pool will be .100 to .249

<table><tr><td>150</td><td></td></tr><tr><td>Range</td><td>0 - 255</td></tr></table></td></tr>
<tr><td>Web: Leasetime

UCI: dhcp.@dhcp[x].leasetime Opt: leasetime</td><td>Defines the lease time of addresses handed out to clients, for example 12h or 30m.

<table><tr><td>12h</td><td>12 hours</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: dhcp.@dhcp[x].interface

Opt: interface</td><td>Defines the interface that is served by this DHCP pool. This must be one of the configured interfaces.

When configured through the web UI this will be automatically populated with the interface name</td></tr>
</table>

11.3.2. DHCP Server: Advanced Settings

The DHCP server advanced settings section

Web Field/UCI/Package Option	Description
Web: Dynamic DHCP UCI: dhcp.@dhcp[x].dynamicdhcp Opt: dynamicdhcp	Defines whether to dynamically allocate DHCP leases. 1 — Dynamically allocate leases. 0 — Use /etc/ethers file for serving DHCP leases.
Web: Force UCI: dhcp.@dhcp[x].force Opt: force	Forces DHCP serving on the specified interface even if another DHCP server is detected on the same network segment. 0 — Disabled. 1 — Enabled.
Web: IPv4-Netmask UCI: dhcp.@dhcp[x].netmask Opt: netmask	Defines a netmask sent to clients that overrides the netmask as calculated from the interface subnet.
Web: DHCP-Options UCI: dhcp.@dhcp[x].dhcp_option Opt: list dhcp_option	Defines additional options to be added for this dhcp pool. For example with 'list dhcp_option 26,1470' or 'list dhcp_option mtu, 1470' you can assign a specific MTU per DHCP pool. Your client must accept the MTU option for this to work. Options that contain multiple values should be separated by a comma. Example: list dhcp_option 6,192.168.2.1,192.168.2.2 No options defined. Syntax: option_number, option_value
Web: n/a UCI: dhcp.@dhcp[x].networkid Opt: networkid	Assigns a network-id to all clients that obtain an IP address from this pool. Use network from interface subnet. Range

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

11.4. Interface Configuration Using Command Line

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

Interface Configuration using UCI

```
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].interface=newinterface
dhcp.@dhcp[0].mode=ipv4
dhcp.@dhcp[0].start=100
dhcp.@dhcp[0].limit=150
dhcp.@dhcp[0].leasetime=12h
```

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

11.4.1. Interface Configuration Using Package Options

```
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 output 'ACCEPT'
option forward 'ACCEPT'
option network 'lan newinterface'

root@VA_router:~# uci export dhcp package dhcp
config dhcp
option interface 'newinterface'
option mode ‘ipv4’
option start '100'
option leasetime '12h'
option limit '150'
```

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

11.4.2. Loopback Interfaces UCI

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

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'
```

NOTE

We highly recommend you **do not** un-assign the 127.0.0.1 IP address from the loopback interface as this action will cause issues with the syslog mechanism and all internal logs will be routed outside the router.

If you must assign an alternative IP address to a loopback interface then you should create the alias of the loopback interface as shown below.

```
Config alias 'loopback_alt'
option interface 'loopback' option proto 'static'
```

```
option ipaddr '10.1.1.10'
option netmask '255.255.255.0'
```

11.5. Configuring Port Maps Using The Web Interface

Port Map Packages

Package	Sections
Network	va_switch

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**.

The interface port map section

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

11.6. 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.

11.7. 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.

ATM Bridges

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

11.8. Interface Diagnostics

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.

Interface Status

```
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)
```

11.9. 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	eth

NOTE

A route will only be displayed in the routing table when the interface is up.

11.10. ARP Table Status

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

```
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
```

12. Configuring VLAN

Maximum number of VLANS supported

Merlin routers support up to 4095 VLANs.

Package	Sections
Network	

12.1. Configuring VLAN Using The Web Interface

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

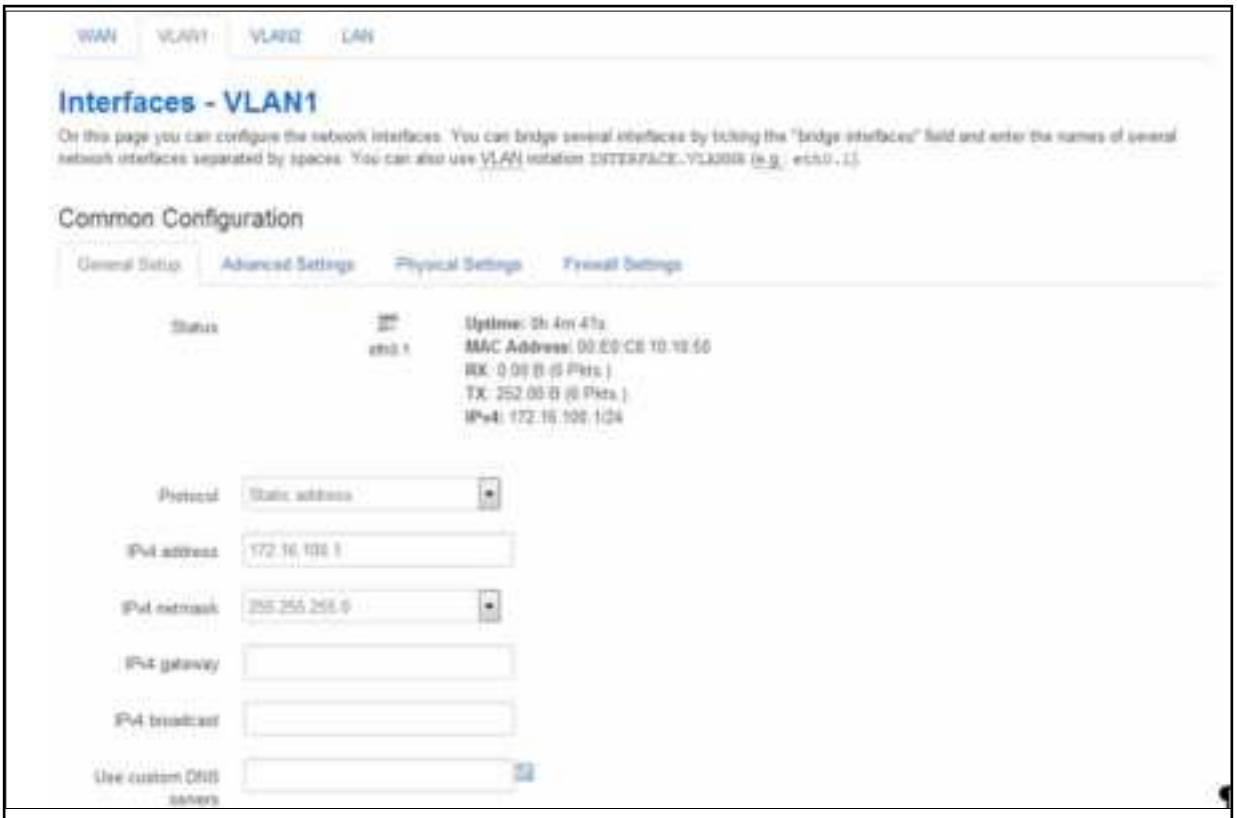
Click **Add new interface**. The Create Interface page appears.

The Create interface page

Web Field/UCI/Package Option	Description
Web: Name of the new interface UCI: network.vlan1=interface Opt: interface	Type the name of the new interface. For example, VLAN1
Web: Protocol of the new interface UCI: network.vlan_tets.proto Opt: proto	Protocol type. Select Static. Option / Description Static: Static configuration with fixed address and netmask. DHCP Client: Address and netmask are assigned by DHCP Unmanaged: Unspecified IPv6-inIPv4 (RFC4213): Used with tunnel brokers IPv6-over-IPv4: Stateless IPv6 over IPv4 transport GRE: Generic Routing Encapsulation protocol IOT: L2TP: Layer 2 Tunnelling Protocol PPP: Point to Point Protocol PPPoE: PPP over Ethernet PPPoATM: PPP over ATM LTE/UMTS/GPRS/EV-DO: CDMA, UMTS, or GPRS connection using an AT-style 3g modem
Web: Create a bridge over multiple interfaces UCI: network.vlan1.type Opt: ifname	Create a bridge over multiple interfaces.
Web: Cover the following interface UCI: network.vlan1.ifname Opt: ifname	Check the Custom Interface radio button. Enter a name, for example, eth0.100. this will assign VLAN 100 to the eth0 interface.

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

12.2. General Setup: VLAN



The common configuration general setup page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Protocol

UCI: network.VLAN1.proto

Opt: proto</td><td>Protocol type.
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6-inIPv4 (RFC4213)</td><td>Used with tunnel brokers</td></tr>
<tr><td>IPv6-over-IPv4</td><td>Stateless IPv6 over IPv4 transport</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation protocol</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td></tr>
<tr><td>PPP</td><td>Point to Point Protocol</td></tr>
<tr><td>PPPoE</td><td>PPP over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>PPP over ATM</td></tr>
<tr><td>LTE/UMTS/GPRS/EV-DO</td><td>CDMA, UMTS, or GPRS connection using an AT-style 3g modem</td></tr>
</table></td></tr>
<tr><td>Web: IPv4 address

UCI: network.VLAN1.ipaddr

Opt: ipaddr</td><td>The IPv4 address of the interface. This is optional if an IPv6 address is provided.</td></tr>
<tr><td>Web: IPv4 netmask

UCI: network.VLAN1.netmask

Opt: netmask</td><td>Subnet mask to be applied to the IP address of this interface.</td></tr>
<tr><td>Web: IPv4 gateway

UCI: network.VLAN1.gateway

Opt: gateway</td><td>IPv4 default gateway to assign to this interface (optional).</td></tr>
<tr><td>Web: Use custom DNS servers

UCI: network.VLAN1.dns

Opt: dns</td><td>List of DNS server IP addresses (optional).</td></tr>
</table>

12.3. 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.

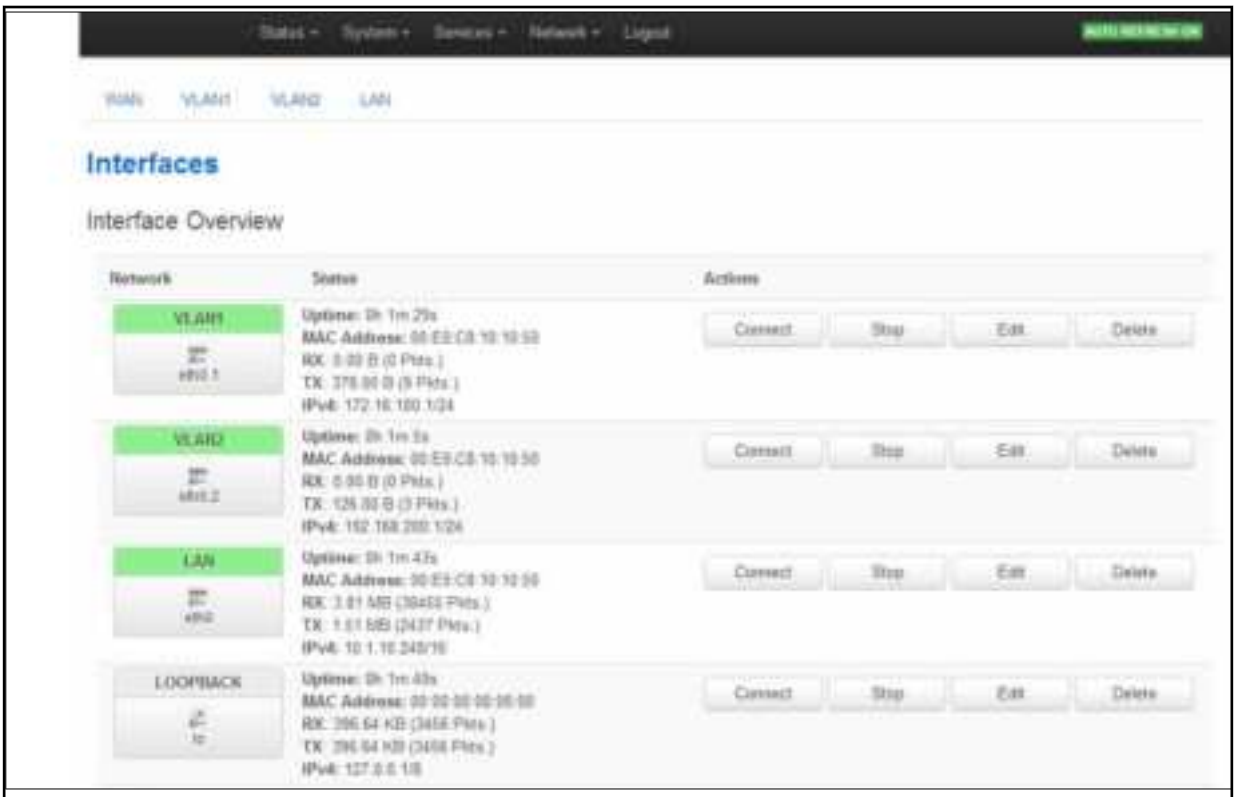
The common configuration firewall settings page

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

12.4. 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.



The interface overview page showing two VLAN interfaces

12.5. Configuring VLAN Using UCI

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.

13. Configuring Mobile Manager

The Mobile Manager feature allows you to configure SIM settings.

Configuration Package Used

Package	Sections
mobile	main
	callers
	roaming_template

13.1. 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	Enable SMS, configure SIM pin code and select roaming SIM.
Advanced	Configure advanced options such as collect ICCIDs and temperature polling interval.
LTE	LTE-specific settings
CDMA*	CDMA configuration.
Callers	Configure callers that can use SMS.
Roaming Interface Template	Configure Preferred Roaming List options.
*Option available only for CDMA modules.	

13.1.1. Mobile Manager: Basic Settings

The mobile manager basic settings page

Web Field/UCI/Package Option	Description
Web: SMS Enable UCI: mobile.main.sms Opt: sms	Enables or disables SMS functionality. Default: 1 — Enabled 0 — Disabled
Web: PIN code for SIM1 UCI: mobile.main.sim1pin Opt: sim1pin	Depending on the SIM card specify the pin code for SIM 1. Default: Blank Range: Depends on the SIM provider
Web: PIN code for SIM2 UCI: mobile.main.sim2pin Opt: sim2pin	Depending on the SIM card specify the pin code for SIM 2. Default: Blank Range: Depends on the SIM provider

13.1.2. Mobile Manager: Advanced Settings

The mobile manager advanced settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Collect ICCIDs

UCI: mobile.main.init_get_iccids

Opt: init_get_iccids</td><td>Enables or disables integrated circuit card identifier ICCIDs collection functionality. If enabled, then both SIM 1 and SIM 2 ICCIDs will be collected; otherwise it will default to SIM 1. This will be displayed under mobile stats.

<table><tr><td>Default: 1</td><td>Enabled</td></tr><tr><td>0</td><td>Disabled</td></tr></table></td></tr>
<tr><td>Web: Force Mode

UCI: mobile.main.force_mode

Opt: force_mode</td><td>Defines whether to operate mobile modem in PPP or Ethernet mode. The mode will be dependent on the service provided by the mobile provider. In general, this is Ethernet mode (default).

Note: It should not be necessary to force PPP mode – contact your support representative for advice.

<table><tr><th>Web</th><th>UCI</th><th>Description</th></tr><tr><td>Automatic</td><td></td><td>Ethernet mode (option not present)</td></tr><tr><td>PPP</td><td>tty</td><td>Enable PPP mode</td></tr></table></td></tr>
<tr><td>Web: Temperature Polling Interval

UCI: mobile.main.temp_poll_interval_sec

Opt: temp_poll_interval_sec</td><td>Defines the time in seconds to poll the mobile module for temperature. Set to 0 to disable.</td></tr>
<tr><td>Web: Automatic Firmware Selection

UCI: mobile.main.enable_firmware_autoselect

Opt: enable_firmware_autoselect</td><td>Enables the selection of an operator-specific firmware in the radio module. The selection is based on the ICCID of the used SIM. At module initialisation the IMSI is checked and if necessary, the correct firmware image in the module will be activated.

Note: activation of the firmware will lead to a delayed startup of the network interface associated with the radio module.

Note: this feature is currently only supported for the Telit LE910NA V2 module. Here Verizon-specific firmware will be selected if the ICCID starts with “891480”.

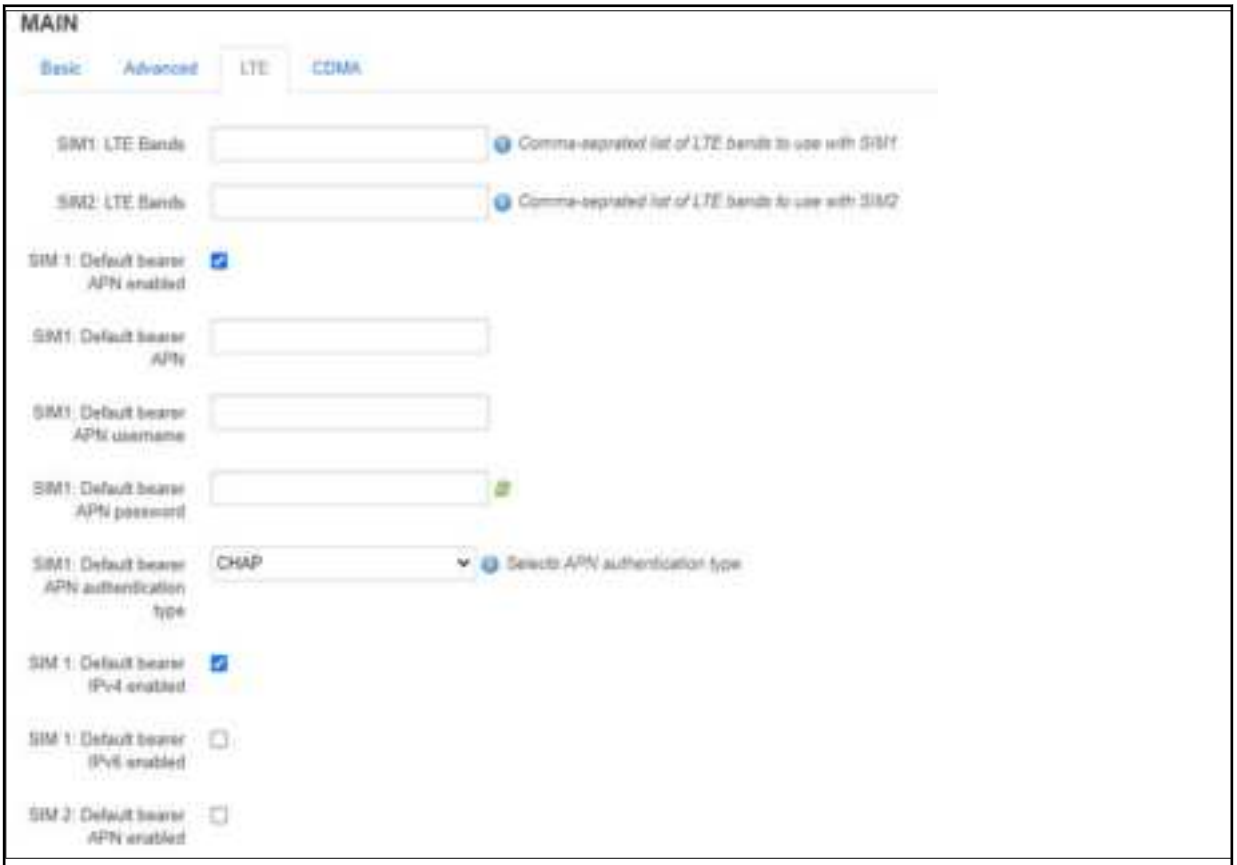
<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Allow USB Power Cycle

UCI: mobile.main.allow_usb_powercycle

Opt: allow_usb_powercycle</td><td>Defines whether to automatically power cycle the USB modem if a mobile module is not detected for 40 seconds.

<table><tr><td>Default: 1</td><td>Enabled</td></tr><tr><td>0</td><td>Disabled</td></tr></table></td></tr>
</table>

13.1.3. Mobile Manager: LTE Settings



The mobile manager LTE settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: SIM1: LTE bands

UCI: mobile.main.sim1_lte_bands

Opt: sim1_lte_bands</td><td>Depending on the SIM card, specify the LTE bands for SIM 1. Comma delimiter. Example:

<code>option sim1_lte_bands '3,20'</code>

Limits LTE bands to 3 and 20

Note: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200, Asiatel and Quectel radio modules.

<table><tr><td>Default:</td><td>Blank</td></tr><tr><td>Range</td><td>LTE bands range from 1 to 70.</td></tr></table></td></tr>
<tr><td>Web: SIM2: LTE bands

UCI: mobile.main.sim2_lte_bands

Opt:sim2_lte_bands</td><td>Depending on the SIM card, specify the LTE bands for SIM 2. Comma delimiter. Example:

<code>option sim2_lte_bands '3,20'</code>

Limits LTE bands to 3 and 20

Note: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200, Asiatel and Quectel radio modules.

<table><tr><td>Default:</td><td>Blank</td></tr><tr><td>Range</td><td>LTE bands range from 1 to 70.</td></tr></table></td></tr>
<tr><td>Web: SIM1: Default bearer APN enabled

UCI: mobile.main.sim1_lte_default_apn_enabled

Opt: sim1_lte_default_apn_enabled</td><td>Enables the use of a specific LTE attach bearer.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: SIM1: Default bearer APN

UCI: mobile.main.sim1_lte_default_apn

Opt: sim1_lte_default_apn</td><td>Specifies the LTE attach bearer APN.</td></tr>
<tr><td>Web: SIM1: Default bearer APN username

UCI: mobile.main.sim1_lte_default_apn_username

Opt: sim1_lte_default_apn_username</td><td>Username for authentication with attach bearer APN.</td></tr>
<tr><td>Web: SIM1: Default bearer APN password

UCI: mobile.main.sim1_lte_default_apn_password

Opt: sim1_lte_default_apn_password</td><td>Password for authentication with attach bearer APN.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>CHAP</td><td>CHAP authentication</td><td>2</td></tr><tr><td>PAP</td><td>PAP authentication</td><td>1</td></tr></table></td></tr>
<tr><td>Web: SIM1: Default bearer IPv4 enabled

UCI: mobile.main.sim1_lte_default_apn_ipv4

Opt: sim1_lte_default_apn_ipv4</td><td>Enables IPv4 for the attach bearer

<table><tr><td>Default: 0</td><td>IPv4 disabled</td></tr><tr><td>1</td><td>IPv4 enabled</td></tr></table></td></tr>
<tr><td>Web: SIM1: Default bearer IPv6 enabled

UCI: mobile.main.sim1_lte_default_apn_ipv6

Opt: sim1_lte_default_apn_ipv6</td><td>Enables IPv6 for the attach bearer.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: SIM2: Default bearer APN

UCI: mobile.main.sim2_lte_default_apn

Opt: sim2_lte_default_apn</td><td>Specifies the LTE attach bearer APN.</td></tr>
<tr><td>Web: SIM2: Default bearer APN username

UCI: mobile.main.sim2_lte_default_apn_username</td><td>Username for authentication with attach bearer APN.</td></tr>
</table>

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Opt: sim2_lte_default_apn_username</td><td></td></tr>
<tr><td>Web: SIM2: Default bearer APN password

UCI: mobile.main.sim2_lte_default_apn_password

Opt: sim2_lte_default_apn_password</td><td>Password for authentication with attach bearer APN.</td></tr>
<tr><td>Web: SIM2: Default bearer APN authentication type

UCI: mobile.main.sim2_lte_default_apn_password

Opt: sim2_lte_default_apn_password</td><td>Selects the APN authentication mechanism.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>CHAP</td><td>CHAP authentication</td><td>2</td></tr><tr><td>PAP</td><td>PAP authentication</td><td>1</td></tr></table></td></tr>
<tr><td>Web: SIM2: Default bearer IPv4 enabled

UCI: mobile.main.sim2_lte_default_apn_ipv4

Opt: sim2_lte_default_apn_ipv4</td><td>Enables IPv4 for the attach bearer

<table><tr><td>Default: 0</td><td>IPv4 disabled</td></tr><tr><td>1</td><td>IPv4 enabled</td></tr></table></td></tr>
<tr><td>Web: SIM2: Default bearer IPv6 enabled

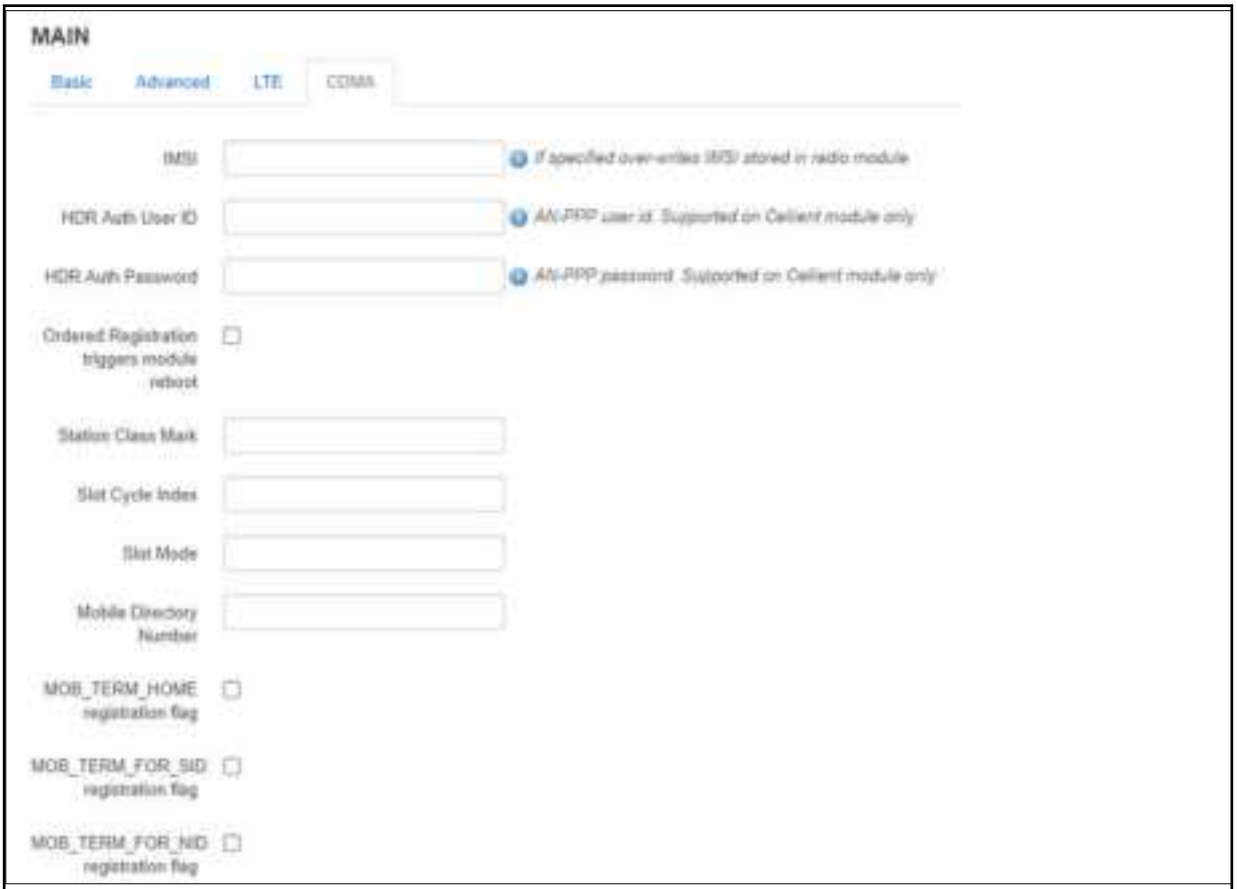
UCI: mobile.main.sim2_lte_default_apn_ipv6

Opt: sim2_lte_default_apn_ipv6</td><td>Enables IPv6 for the attach bearer

<table><tr><td>Default: 0</td><td>IPv6 disabled</td></tr><tr><td>Range</td><td>IPv6 enabled</td></tr></table></td></tr>
</table>

13.1.4. Mobile Manager: CDMA Settings

This configuration page is only supported for CDMA modules.



The mobile manager CDMA page

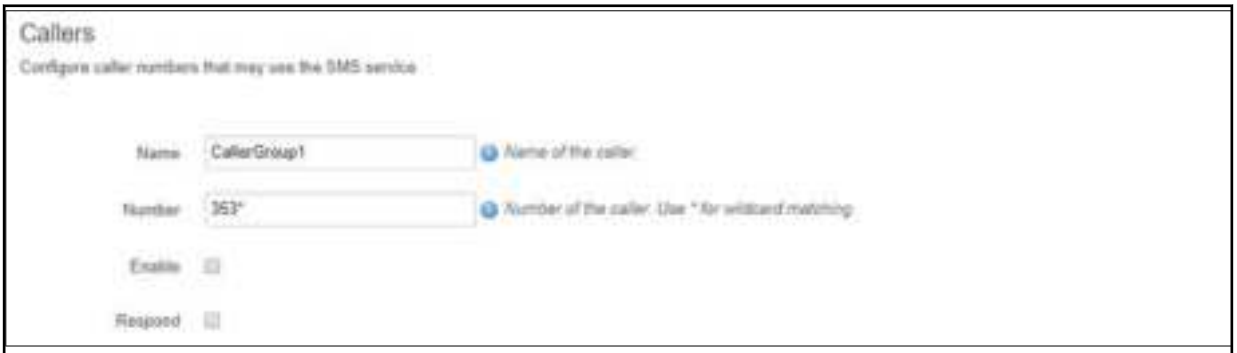
Web Field/UCI/Package Option	Description
Web: IMSI UCI: mobile.main.imsi Opt: imsi	Allows the IMSI (International Mobile Subscriber Identity) to be changed. Default: Programmed in module. Range: Up to 15 digits
Web: HDR Auth User ID UCI: mobile.main.hdr_userid Opt: hdr_userid	AN-PPP user ID. Supported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

Web Field/UCI/Package Option	Description
Opt: cdma_access_overload_control	1 Enabled
Web: Preferred Serving System UCI: mobile.main.cdma_preferred_serving_system Opt: cdma_preferred_serving_system	The CDMA Preferred Serving System(A/B). Default: 5
Web: Digital Analog Mode Preference UCI: cdma_digital_analog_mode_preference Opt: cdma_digital_analog_mode_preference	Digital/analog mode preference. Default: 4
Web: Primary Channel A UCI: mobile.main.cdma_primary_channel_a Opt: cdma_primary_channel_a.	Allows the primary channel (A) to be changed. Default: 283 Range 1-2016. Any band class 5 channel number.
Web: Primary Channel B UCI: mobile.main.cdma_primary_channel_b Opt: cdma_primary_channel_b	Allows the primary channel (B) to be changed. Default: 384 Range 1-2016. Any band class 5 channel number.
Web: Secondary Channel A UCI: mobile.main.cdma_secondary_channel_a Opt: cdma_secondary_channel_a	Allows the secondary channel (A) to be changed. Default: 691 Range 1-2016. Any band class 5 channel number.
Web: Secondary Channel B UCI: mobile.main.cdma_secondary_channel_b Opt: cdma_secondary_channel_b	Allows the secondary channel (B) to be changed. Default: 777 Range 1-2016. Any band class 5 channel number.
Web: Preferred Forward & Reverse RC UCI: mobile.main.cdma_preferred_forward_and_reverse_rc Opt: cdma_preferred_forward_and_reverse_rc	The preferred forward & reverse RC value, this takes the form "forward_rc,reverse_rc" Default: 0,0 Format forward radio channel, reverse radio channel
Web: SID-NID pairs UCI: mobile.main.cdma_sid_nid_pairs Opt: cdma_sid_nid_pairs	Allows specification of SID:NID pairs, this takes the form "SID1,NID1,SID2,NID2, Default: 0,0 Format SID1 (0-65535),NID (0-65535)

Mobile manager: roaming interface template

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

13.1.5. Mobile Manager: Callers



The mobile manager CDMA page

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

13.2. Configuring 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.sim1_lte_bands='3,20'
uci set mobile.main.sim2_lte_bands='4,5'
uci set mobile.main.temp_poll_interval_sec=61
uci set mobile.main.enable_firmware_autoselect=0
uci set mobile.main.allow_usb_powercycle=1
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
```

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'
option sim1_lte_bands '3,20'
option sim2_lte_bands '4,5'
option temp_poll_interval_sec '61'
option enable_firmware_autoselect '0'
option allow_usb_powercycle '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'
```

13.3. 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.

Example of output from system log

To monitor using SSH, login and enter:

```
logread -f &
```

Or, when logging system messages to a flash file at /root/syslog.messages

```
tail -f /root/syslog.messages &
```

13.3.1. 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"
```

13.3.2. 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
```

Multiple commands can be sent in a single SMS using a semicolon (;) separator; for example, to set the router to factory config and then reboot.

```
vacmd set next config factconf;reboot
```

14. Configuring Multi-APNs For Mobile Interfaces

Some of our routers, including the Merlin Series, support simultaneous multiple APN connections to be connected using a single SIM card. Up to four APNs per cellular radio module are currently supported.

Configuration Package Used

Package	Sections
network	interface

14.1. Multi-APN Overview

A PDP (Packet Data Protocol) context is a data structure that exists within the mobile service provider's network that contains a subscriber's session information when the subscriber has an active session. The PDP context data structure contains:

- the subscriber's IP address,
- IMSI (International Mobile Subscriber Identity), and
- APN (Access Point Name

It is sometimes required to connect to several different APNs at the same time. This can be achieved with a single SIM card using separate PDP contexts.

NOTE
The SIM card must allow connection to each of the APNs. Also, two PDP contexts from the same SIM card cannot use the same APN.

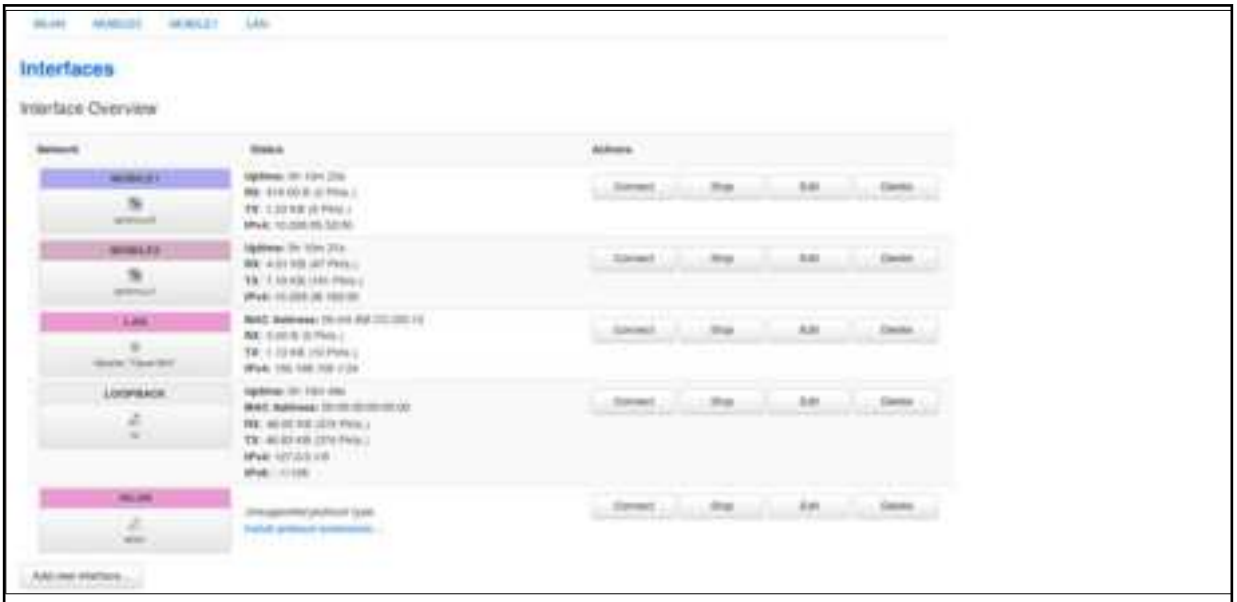
You can use routing and VRF support for each PDP context by referring to the unique interface name that the APN is configured under. Routing and VRF support can be utilised for each PDP context. For more information on these features, read chapters 'Configuring Static Routes' and 'VRF: Virtual Router Forwarding'.

Multi-WAN can control routing to each PDP context in the same way it can control routing to other interfaces. However, in package multiwan option manage_state, set to no for both multiwan interface configurations. Multiwan will then control routing through each PDP context by altering the interface metric to '-1' when it determines the interface has failed its health check.

14.2. Configuring Multi-APN Using The Web Interface

To configure Multi-APN, select **Network -> Interface.** A unique PDP context needs to be configured on each mobile interface. For more information on how to configure a mobile interface, read the chapter 'Configuring a mobile connection'.

NOTE
On each mobile interface set **option sim** to the same number and not to **any**.



The network interface page

On the the desired mobile interface, select **Edit** and then select **Advanced Settings**.

The mobile interface advanced settings page

Web Field/UCI/Package Option	Description
Web: PDP context UCI: network.[interface].pdp_context Opt:pdp_context	Defines the PDP context ID. Should multiple active PDP contexts be supported, you must configure interfaces with different PDP context IDs. 1 Range 1-4

14.3. Configuring Multi-APN Using The Command Line

You can configure multi-APN using the interface configuration section in the network package /etc/config/network using the option pdp_context. The option value should be an integer that is unique to each APN configuration.

14.3.1. Configuring Multi-APN Using UCI

```
root@VA_router:~# uci show network
package network


network.Mobile1=interface
network.Mobile1.proto=3g
network.Mobile1.apn=open.internet
network.Mobile1.username=gprs
network.Mobile1.password=gprs
network.Mobile1.sim=1
network.Mobile1.service=auto
network.Mobile1.metric=1
```

```
network.Mobile1.pdp_context=1
network.Mobile2=interface
network.Mobile2.proto=3g
network.Mobile2.apn=3ireland.ie
network.Mobile2.sim=1
network.Mobile2.service=auto
network.Mobile2.metric=1
network.Mobile2.pdp_context=2
```

Configuring multi-APN using package options

```
root@VA_router:~# uci export
network package network
config interface 'Mobile1'
option proto '3g'
option apn 'open.internet'
option username 'gprs'
option password 'gprs'
option sim '1'
option service 'auto'
option metric '1'
option pdp_context '1'
config interface 'Mobile2'
option proto '3g'
option apn '3ireland.ie'
option sim '1'
option service 'auto'
option metric '1'
option pdp_context '2'
```

Example of simple routing over multi-APN using UCI

```
root@VA_router:~# uci show network
package network
network.Mobile1=interface
network.Mobile1.proto=3g
network.Mobile1.apn=open.internet
network.Mobile1.username=gprs
network.Mobile1.password=gprs
network.Mobile1.sim=1
network.Mobile1.service=auto
network.Mobile1.metric=1
network.Mobile1.pdp_context=1
network.Mobile1.defaultroute=0
network.Mobile2=interface
network.Mobile2.proto=3g
network.Mobile2.apn=3ireland.ie
network.Mobile2.sim=1
network.Mobile2.service=auto
network.Mobile2.metric=1
network.Mobile2.pdp_context=2
network.Mobile1.defaultroute=0
::::
network.8888=route
network.8888.interface=Mobile1
network.8888.target=8.8.8.8
network.8888.netmask=255.255.255.255
network.8844=route
network.8844.interface=Mobile1
network.8844.target=8.8.4.4
network.8844.netmask=255.255.255.255
```

14.4. Multi-APN Diagnostics

Interface Status

When active, to see the status of interfaces with multiple APNs, enter:

```
root@VA_router:~# ifconfig

qmimux0 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00

inet addr:10.205.77.223 P-t-P:10.205.77.223 Mask:255.255.255.192

inet6 addr: fe80::9bb3:25f7:278c:a8f1/64 Scope:Link
```

```
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1

RX packets:5 errors:0 dropped:0 overruns:0 frame:0 TX packets:23 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1

RX bytes:1540 (1.5 KiB) TX bytes:3976 (3.8 KiB)

qmimux1 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00

inet addr:10.209.38.182 P-t-P:10.209.38.182 Mask:255.255.255.252

inet6 addr: fe80::89f2:b5d5:f017:ae91/64 Scope:Link

UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1

RX packets:94 errors:0 dropped:0 overruns:0 frame:0 TX packets:293 errors:0 dropped:0 overruns:0 carrier:0 collisions:0
txqueuelen:1

RX bytes:9032 (8.8 KiB) TX bytes:20860 (20.3 KiB)
```

To check which mobile interface corresponds to the output from the ifconfig command shown above, enter:

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

Interface: Mobile1

Status: Up

Uptime: 00h 05m 30s

IPv4 addresses: 10.202.187.228/29 MAC address: 00:00:00:00:00:00

Device name: "qmimux0"


Interface: Mobile2

Status: Up

Uptime: 00h 05m 27s

IPv4 addresses: 10.201.206.252/29 MAC address: 00:00:00:00:00:00

Device name: "qmimux1"
```

14.4.1. Routing Table

To check the routing table, enter:

```
root@VA_router:~# ip route
8.8.4.4 via 10.204.5.101 dev qmimux0
8.8.8.8 via 10.204.5.101 dev qmimux0
10.204.5.100/30 dev qmimux0 proto kernel scope link src 10.204.5.102
10.209.38.180/30 dev qmimux1 proto kernel scope link src 10.209.38.182
192.168.100.0/24 dev eth0 proto kernel scope link src 192.168.100.1
192.168.101.0/24 dev wlan0 proto kernel scope link src 192.168.101.1
192.168.101.0/24 dev wlan1 proto kernel scope link src 192.168.101.1
```

15. 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.

Configuration Packages Used

Package	Sections
network	interface

15.1. 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.

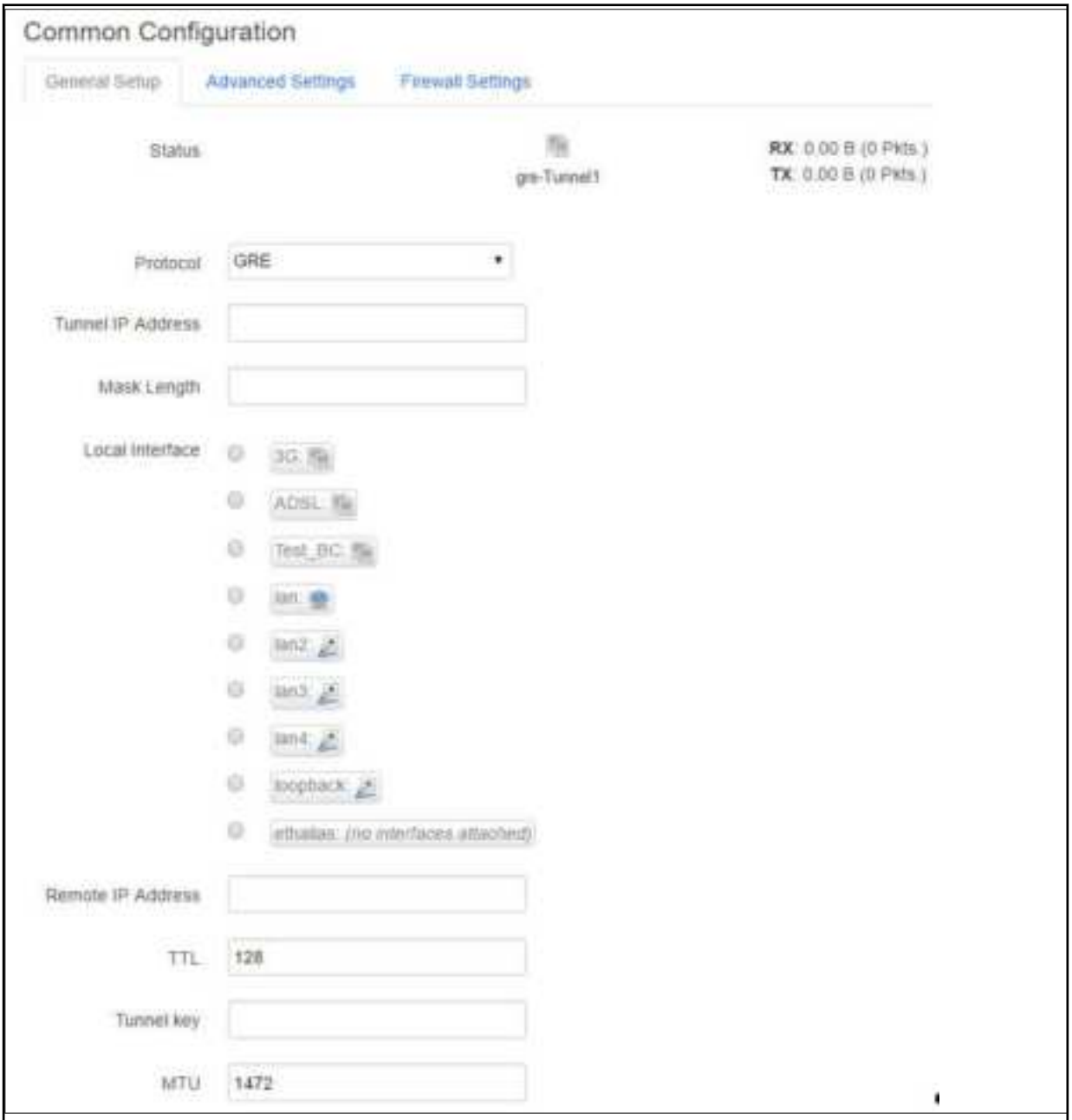
The create interface page

Web Field/UCI/Package Option	Description
Web: Name of the new interface UCI: network. .<if name> Opt: config interface	Assigns a logical name to the GRE tunnel. The network interface section will be assigned this name <if name>. Type the name of the new interface. Allowed characters are A-Z, a-z, 0-9 and _. Must be less than 11 characters.
Web: Protocol of the new interface UCI: network.<if name> .proto Opt: proto	Specifies what protocol the interface will operate on. Select GRE. Option / Description Static / Static configuration with fixed address and netmask. DHCP Client / Address and netmask are assigned by DHCP Unmanaged / Unspecified IPv6-inIPv4 (RFC4213) / Used with tunnel brokers IPv6-over-IPv4 / Stateless IPv6 over IPv4 transport GRE / Generic Routing Encapsulation protocol IOT / L2TP / Layer 2 Tunnelling Protocol PPP / Point to Point Protocol PPPoE / PPP over Ethernet PPPoATM / PPP over ATM LTE/UMTS/GPRS/EV-DO / CDMA, UMTS, or GPRS connection using an AT-style 3g modem
Web: Create a bridge over multiple interfaces UCI: network.<if name> Opt: n/a	Not applicable for GRE.
Create a bridge over multiple interfaces UCI: network.< name> Opt: n/a	Not applicable for GRE.

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.

15.1.1. GRE Connection: Common Configuration: General Setup



The GRE common configuration page

Web Field/UCI/Package Option	Description
Web: Protocol of the new interface UCI: network.<if name>.proto Opt: proto	Shows the protocol the interface will operate on . GRE should be currently selected.
Web: Tunnel IP Address UCI: network.<if name>.ipaddr Opt: ipaddr	Configures local IP address of the GRE interface.
Web: Mask length UCI: network.<if name> .ipaddr Opt: mask_length	Subnet mask, in CIDR notation, to be applied to the tunnel. Typically '30' for point-to-point tunnels. Default: 24 Range: 0-30
Web: Local Interface UCI: network. <if name> .remote_ip Opt: local interface	Specifies which interface is going to be linked with the GRE tunnel interface (optional).
Web: Remote IP address UCI: network. <if name> .remote_ip Opt: remote_ip	For point-to-point tunnels. Specifies remote IP address.
Web: TTL UCI: network. <if name> .key Opt: key	Sets Time-To-Live value on the interface. Default: 128 Range:
Web: Tunnel key UCI: network. <if name> .key Opt: key	Sets GRE tunnel ID key (optional). Usually an integer.
Web: MTU UCI: nwtwork. <if name> .mtu Opt: mtu	Sets GRE MTU (maximum transmission unit) size of PDUs using this interface. Default: 1472 Range:

15.1.2. Configuring IPv6 Routes Using The Web Interface

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

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

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

15.1.3. GRE Connection: Common Configuration-Advanced Settings

The GRE advanced settings page

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

15.1.4. 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.

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.

15.1.5. 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'.

15.2. Configuring GRE Using UCI

GRE using Command Line

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

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

```
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
```

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.

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	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.

15.3. GRE Diagnostics

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	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

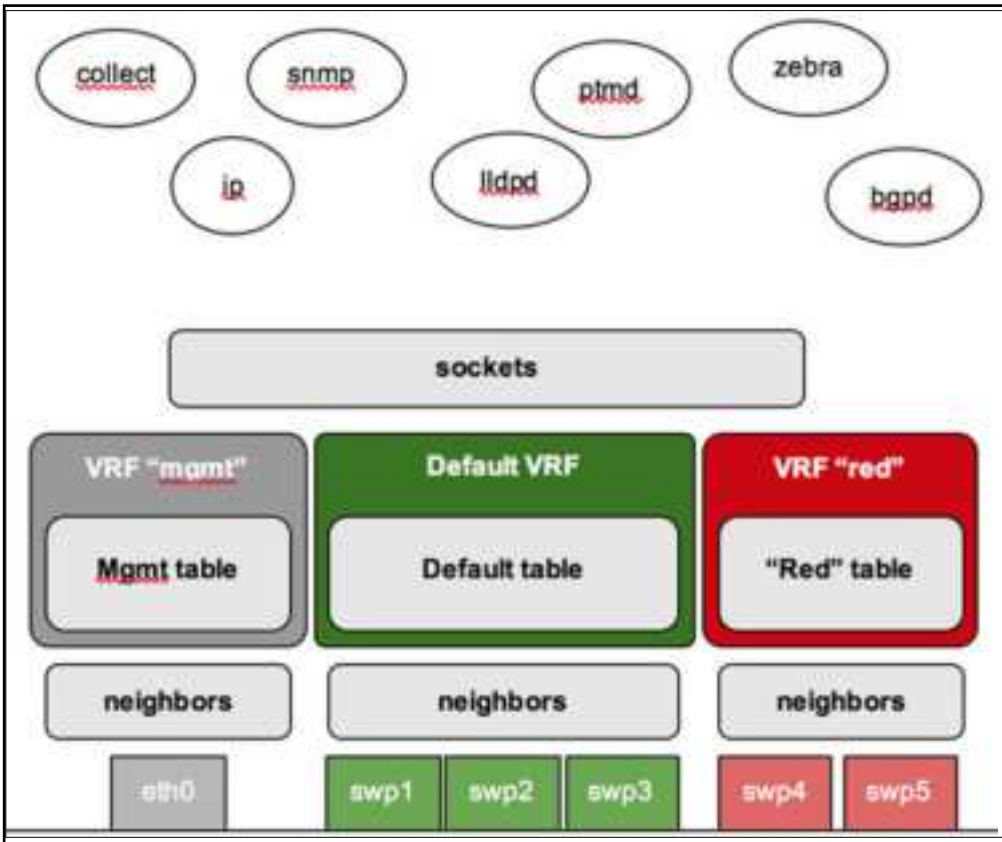
Note: a GRE route will only be displayed in the routing table when the interface is up.

16. Configuring VRF (Virtual Router Forwarding)

Virtual Routing and Forwarding (VRF) is a technology that allows multiple instances of a routing table to exist in a router and work simultaneously. Traffic between routing tables is segregated and so increases security.

VRF overview

An interface is configured to belong to a VRF. Interfaces included in the VRF form an independent routing domain, so routing of incoming and outgoing packets only happens within a VRF. It is also possible to add individual routes to a VRF using static routes.



VRF architecture

Configuration package used

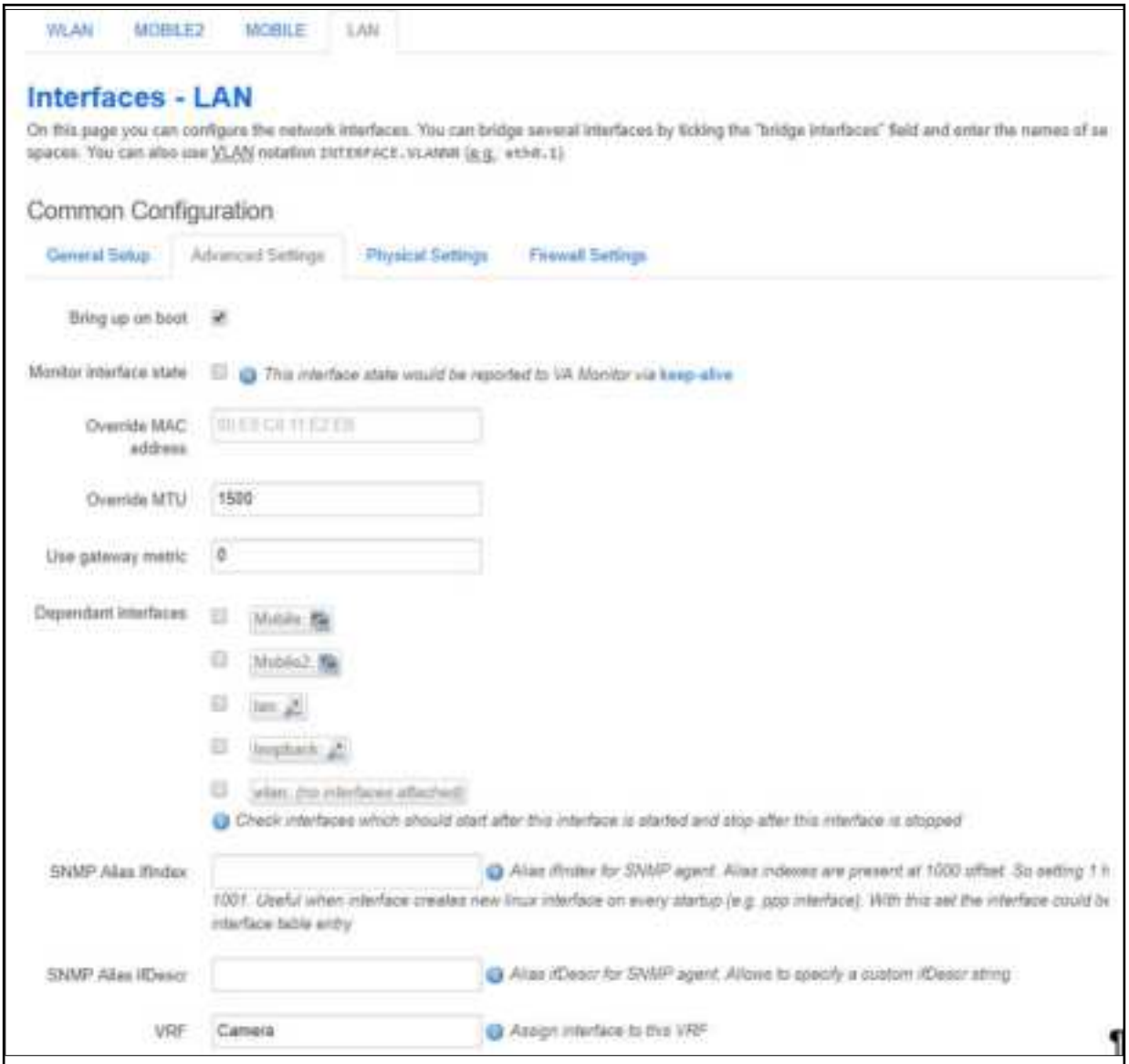
Package	Sections
network	interface
	route

16.1. Configuring VRF Using The Web Interface

Setting the VRF for an Interface

To create VRFs, you must add interfaces. To add an interface to a VRF instance, select **Network - > Interfaces,** select the desired interface to edit then select **Common Configuration - > Advanced Settings**.

Enter in the relevant VRF name in the VRF field.



The interfaces configuration page

Web Field/UCI/Package Option	Description
Web: VRF UCI: network.<if name>.vrf Opt: vrf	Defines the VRF name to which this interface belongs. Note: the name must be consistent across all interfaces that want to reside on that VRF. (empty): Interface is not attached to a VRF Range: 0-15 characters

To add additional interfaces to a VRF, repeat the above for the relevant interface(s).

For example, the above configuration creates a VRF on a LAN interface. To configure this VRF to be used by traffic from a camera on a LAN interface to a VRF on a mobile interface, repeat the above instructions for a mobile interface so the camera VRF will now contain a local network and mobile interface to route traffic.

NOTE

The default VRF is created automatically and is not assigned any VRF name. It is recommended to use this default VRF to access router services and applications; for example, HTTP, SSH, SNMP etc.

16.1.1. Configuring A VRF On A Static Route

Each VRF has its own routing table and static routes can be added to a VRF routing table. To define a static route on a VRF, select **Network - >Static Routes**.

The static routes configuration page

Web Field/UCI/Package Option	Description
Web: VRF UCI: network.route.vrf Opt: vrf	Defines the VRF name. Note: 'none' is a special name to move a route out of a VRF. Example: network.route.vrf=none (empty): Interface is not attached to a VRF Range: 0-15 characters

16.2. Configuring The VRFs Using The Command Line

You configure a VRF using the interface configuration section in the network etc/config/network.

The VRF name must be consistent across all interfaces that want to reside on that VRF.

For the command line examples below, two VRFs called Camera and Management are configured.

VRF using UCI

```
root@VA_router:~# uci show network | grep vrf
network.lan.vrf=Camera
network.Mobile1.vrf=Camera
network.Mobile2.vrf=Management
```

VRF using Package Options

```
root@VA_router:~# uci export network
package network
config interface lan
option vrf 'Camera'
config interface Mobile1
option vrf 'camera'
config interface Mobile2
option vrf 'Manangement'
```

16.2.1. VRF Using UCI

```
root@VA_router:~# uci show network | grep vrf network.lan.vrf=Camera network.Mobile1.vrf=Camera
network.Mobile2.vrf=Management
```

VRF using Package Options

```
root@VA_router:~# uci export network package network
config interface lan
option vrf 'Camera'
config interface Mobile1
```

```
option vrf 'Camera'
config interface Mobile2
option vrf 'Management'
```

To display a list of running VRFs, enter:

```
root@VA_router:~# ip vrf Name Table
Management 10
Camera 10
```

16.3. VRF Diagnostics

To display a list of running VRFs, enter:

```
root@VA_router:~# ip vrf
Name Table
-----------------------------------
Management 10
Camera 10
```

VRF Table

To display the routing table for a VRF, enter the command:

```
ip router list vrf <vrf name>
```

VRF Routes

To display the routing table for a VRF, enter the command:

ip route list vrf <vrf name>.

```
root@VA_router:~# ip route list vrf Camera
default via 10.92.163.130 dev qmimux0
10.92.163.128/30 dev qmimux0 proto kernel scope link src 10.92.163.129
172.16.100.0/24 dev eth1 proto kernel scope link src 172.16.100.1
root@VA_router:~# ip route list vrf Management
default via 10.176.120.94 dev qmimux1
10.176.120.92/30 dev qmimux1 proto kernel scope link src 10.176.120.93
```

17. 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.

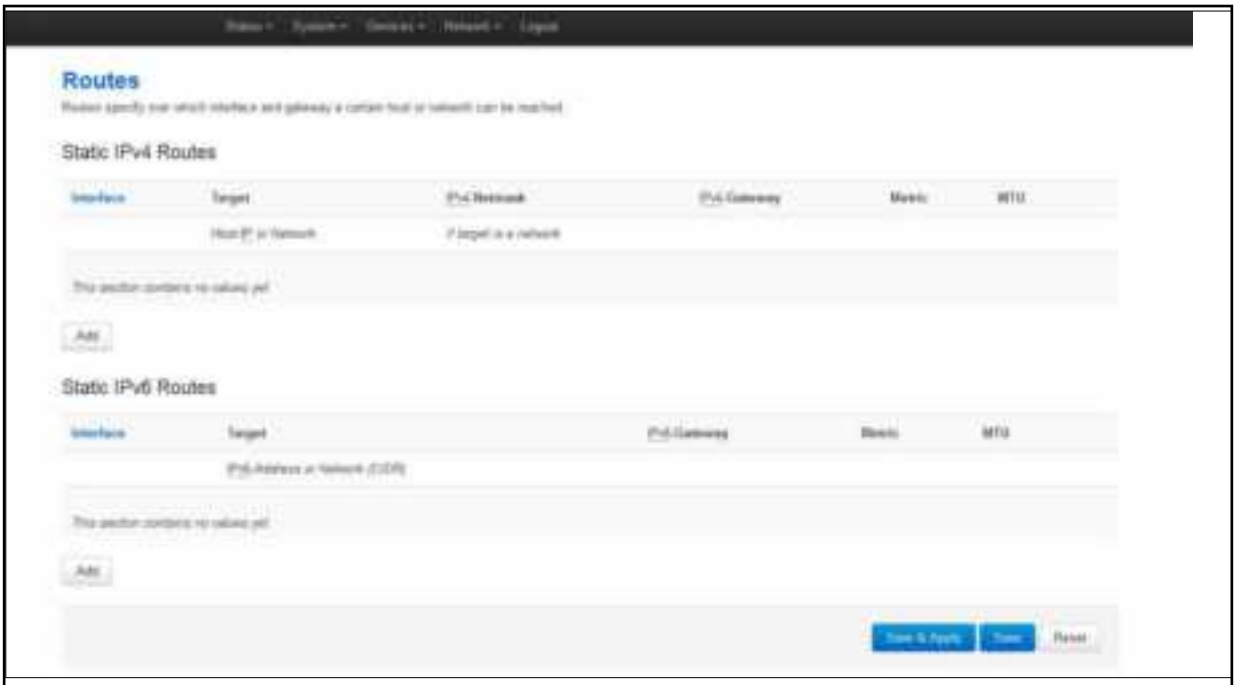
They can be created based on an outgoing interface or next hop IP address.

Configuration Package Used

Package	Sections
network	router

17.1. Configuring Static Routes Using The Web Interface

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



The routes page

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

Web Field/UCI/Package Option	Description
Web: Interface UCI: network.@route[0].interface Opt: interface	Specifies the logical interface name of the parent or master interface this route belongs to. It must refer to one of the defined interface sections.
Web: target UCI: network.@router[0].target Opt: target	Specifies the route network IP address.
Web: netmask UCI: network.@route[0].netmask Opt: netmask	Defines the route netmask. If omitted, 255.255.255.255 is assumed, which makes the target a host address.
Web: Gateway UCI: network.@route[0].gateway Opt: Gateway	Network gateway. If omitted, the gateway from the parent interface is taken. If set to **0.0.0.0** no gateway will be specified for the route.
Web: Metric UCI: network.@route[0].metric Opt: metric	Specifies the route metric to use. Default: 0 Range:
Web: MTU UCI: network.@route[0].mtu Opt:mtu	Defines a specific MTU for this route. If omitted, the MTU from the parent interface will be taken. Default: Blank Range:

17.2. 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'
```

17.3. 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
```

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'
```

17.4. 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
```

IPv6 routers using Package 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'
```

17.5. Static Routes Diagnostics

To show the current routing status, enter:

Root@VA_router:~# route							
Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.100.1	*	255.255.255.0	0	0	0	0	eth 0

NOTE

A route will only be displayed in the routing table when the interface is up.

17.6. Configuring IPv6 Routes Using The Web Interface

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

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

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

18. 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.

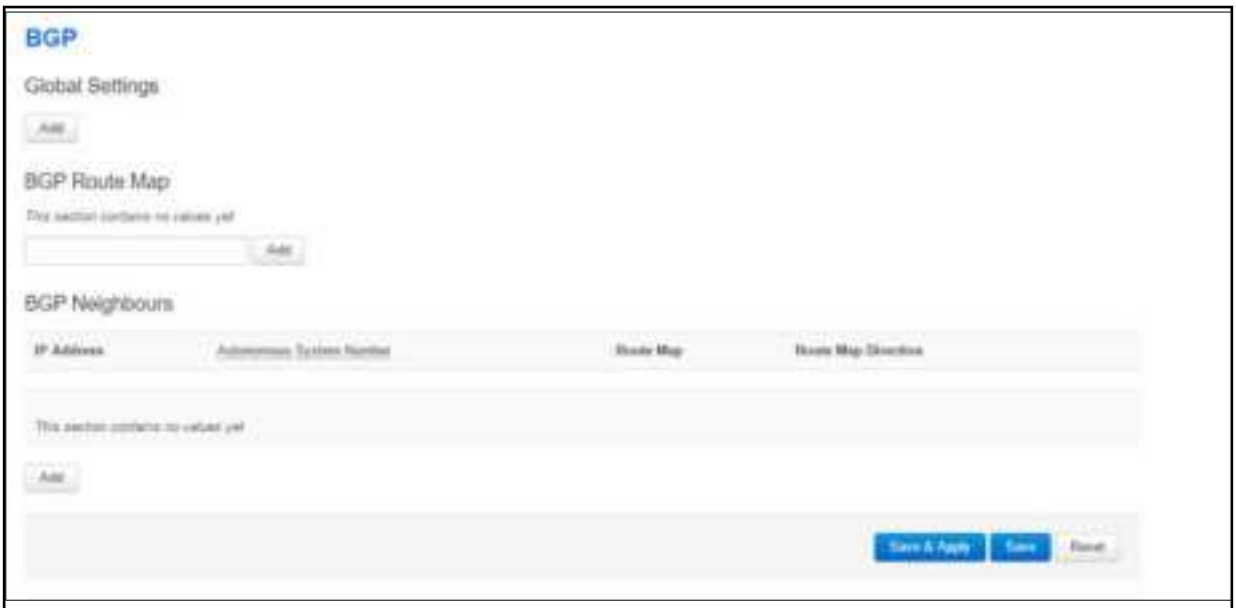
Configuration package used

Package	Sections
bgpd	routing
	peer
	routemap

18.1. Configuring BGP Using The Web Interface

In the top menu, select **Network -> BGP**. The BGP configuration page appears.

The page has three sections: Global Settings, BGP Neighbours and BGP Route Map.



The BGP page

BGP Global Settings

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

The BGP global settings page

Web Field/UCI/Package Option	Description
Web: BGP Enabled UCI: bgpd.bgpd.enabled Opt: enabled	Enables or disables BGP protocol. Default: 0 — Disabled 1 — Enabled
Web: Router ID UCI: bgpd.bgpd.router_id Opt: router_id	Sets a unique router ID in 4 byte format 0.0.0.0.
Web: Scan Time UCI: bgpd.bgpd.scan_time Opt: scan_time	Defines the interval in seconds between RIB scans. Default — 60 seconds Range —
Web: Autonomous System Number UCI: bgpd.bgpd.asn Opt: asn	Defines the ASN for the local router. Type in the ASN. Default — Blank Range — 1-4294967295
Web: Log keepalives UCI: bgpd.bgpd.debug_keepalive Opt: debug_keepalives	Defines whether to enable BGP keepalives to the system log. Default: 0 — Disabled 1 — Enabled
Web: Log events UCI: bgpd.bgpd.debug_events Opt: debug_events	Defines whether to enable BGP event to the system log. Default: 0 — Disabled 1 — Enabled
Web: Log filters UCI: bgpd.bgpd.debug_filters Opt: debug_filters	Defines whether to enable BGP filter events to the system log. Default: 0 — Disabled 1 — Enabled
Web: Log fsm UCI: bgpd.bgpd.debug_fsm Opt: debug_fsm	Defines whether to enable BGP state changes to the system log. Default: 0 — Disabled 1 — Enabled
Web: Log Updates UCI: bgpd.bgpd.debug_updates Opt: debug_updates	Defines whether to enable BGP updates to the system log. Default: 0 — Disabled 1 — Enabled
Web: Network UCI: bgpd.bgpd.network 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. For more information, read the 'Routes' section below.
Web: n/a UCI: bgpd.bgpd.vrf Opt: vrf	Defines the VRF with which to associate this BGP routing instance.

18.2. 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 BGP Route Map configuration section appears. You can configure multiple route maps. The examples below are for a route map named ROUTEMAP.

The route map section

<table>
<tr><th>Web: Match Type</th><th>Defines match type. Available options are as follows:</th></tr>
<tr><td>Web: Order

UCI: bgpd.ROUTEMAP.order

Opt: order</td><td>Defines the route map order number.

<table><tr><td>Default</td><td>Blank</td></tr><tr><td>Range</td><td>1-65535</td></tr></table></td></tr>
<tr><td>Web: Policy Type

UCI: bgpd.ROUTEMAP.permit

Opt: permit</td><td>Defines the actions taken if the entry is matched.

<table><tr><td>Default: Permit</td><td>Permits the route to process the set actions for this entry.</td></tr><tr><td>Deny</td><td>Denies the route.</td></tr></table></td></tr>
<tr><td>UCI: bgpd.ROUTEMAP.match_type

Opt: match_type</td><td>Defines match type. Available options are as follows:

<table><tr><td>IP address</td><td>Matches IP address</td></tr><tr><td>IP Next Hop</td><td>Matches next hop IP address</td></tr><tr><td>AS-Path</td><td>Matches AS path</td></tr><tr><td>Route Metric</td><td>Matches route metric</td></tr><tr><td>BGP Community</td><td>Matches BGP community</td></tr></table></td></tr>
<tr><td>Web: Match value

UCI: bgpd.ROUTEMAP.match

Opt: match</td><td>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.

Enter '-' prefix to deny match.</td></tr>
<tr><td>Web: Set Option

UCI: bgpd.ROUTEMAP.set_type Opt: set_type</td><td>Defines the set option to be processed on a match. Available options are shown below.

<table><tr><td>None</td><td></td></tr><tr><td>IP Next Hop</td><td>Setting option for IP next hop.</td></tr><tr><td>Local Preference</td><td>Setting option for Local Preference</td></tr><tr><td>Router Weight</td><td>Setting option for Route Weight</td></tr><tr><td>BGP MED</td><td>Setting option for BGP multi-exit discriminator (BGP metric)</td></tr><tr><td>AS Path to Prepend</td><td>Setting option to prepend AS to AS path</td></tr><tr><td>BGP Community</td><td>Setting option for BGP community</td></tr><tr><td>IPv6 Next Hop</td><td>Setting option for IPv6 Next Hop Global</td></tr><tr><td>IPv6 Next Hop Local</td><td>Setting option for IPv6 Next Hop Local</td></tr></table></td></tr>
<tr><td>Web: Value

UCI: bgpd.ROUTEMAP.set

Opt: set</td><td>Defines the set value when a match occurs. Value format depends on the set option you have selected.</td></tr>
<tr><td>Web: n/a

UCI: bgpd.ROUTEMAP.routing

Opt: set</td><td>Defines the routing section this BGP route map is related to.</td></tr>
</table>

18.3. Configure BGP Neighbours

To configure BGP neighbours, in the BGP neighbours section, click **Add**. The BGP Neighbours page appears. You can configure multiple BGP neighbours.

The BGP neighbours section

Web Field/UCI/Package Option	Description
Web: IP Address UCI: bgpd.@peer[0].ipaddr Opt: ipaddr	Sets the IP address of the neighbour.
Web: Autonomous System Number UCI: bgpd.@peer[0].asn Opt: asn	Sets the ASN of the remote peer. Default: Blank Range: 1-4294967295
Web: Route Map UCI: bgpd.@peer[0].route_map Opt: route_map	Sets route map name to use with this neighbour.
Web: Route Map Direction UCI: bgpd.@peer[0].route_map_in Opt: route_map_in	Defines what direction to apply to the route map. Default: 1 In 0 Out
Web: IPv6 UCI: bgpd.@peer[0].ipv6 Opt: ipv6	Defines whether the peer is connected over IPv6. Default: 0 1
Web: Local Peer UCI: bgpd.@peer[0].next_hop_self Opt: next_hop_self	Defines an announced route's next hop as being equivalent to the address of the router if it is learned via BGP. Default: 0 1
Web: Holdtime UCI: bgpd.@peer[0].holdtime_sec Opt: holdtime_sec	Defines how long to wait for incoming BGP messages before assuming peer is dead. The timer is reset every time a BGP message is received. Default: 0 Range:
Web: Keepalive Interval UCI: bgpd.@peer[0].keepalive_sec Opt: keepalive_sec	Defines the interval in seconds for between two successive BGP keep alive messages. Default: 0 Range:
Web: Connect Timer UCI: bgpd.@peer[0].connect_sec Opt: connect_sec	Defines how long to wait after interface is up before retrying the connection on it. Default: 0 Range:
Web: n/a UCI: bgpd.@peer[0].routing Opt: routing	Defines the routing section this BGP peer is related to.

18.4. Configuring BGP Using Command Line

Configuring BGP using UCI

You can 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.bgpd.vrf=datavrf
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.@peer[0].ipv6=0
bgpd.@peer[0].next_hop_self=0
bgpd.@peer[0].holdtime_sec=0
bgpd.@peer[0].keepalive_sec=0
bgpd.@peer[0].connect_sec=0
bgpd.@peer[0].routing='bgpd'
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'
bgpd.ROUTEMAP.vrf='bgpd'
```

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

Configuring BGP using Package 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' option ipv6 '0'
option next_hop_self '0'
option holdtime_sec '0'
option keepalive_sec '0'
option connect_sec '0'
option routing 'bgpd'
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'
option routing 'bgpd'
```

18.5. View Routes Statistics

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

The routing table

To view routes via the command line, enter:

root@support:~# route -n							
Kernal 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

19. Configuring OSPF

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 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 for all OSPF routers or 224.0.0.6 for 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).

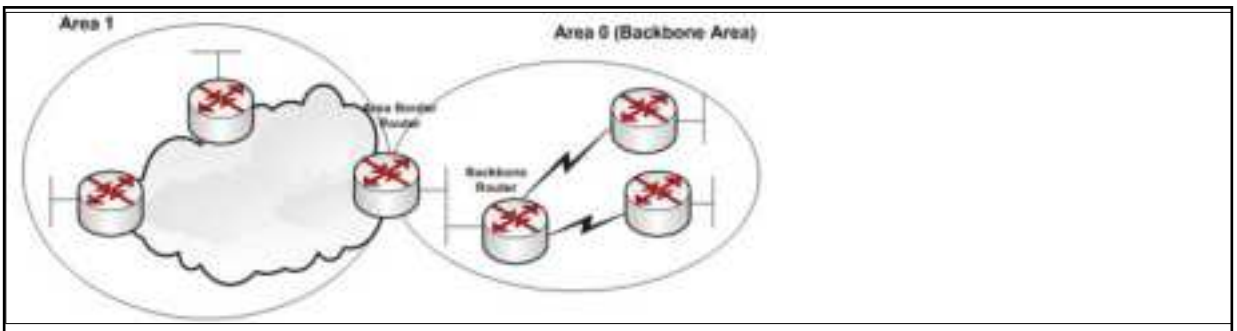
Other characteristics of OSPF include:

- 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.

19.1. OSPF Areas



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.

19.2. 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 of 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

19.3. 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 network, 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 as 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.

19.4. 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 a two-way communication has not yet been initialised.
2-Way	Indicates that bidirectional communication has been established. Recalls that hello packets contain a neighbour field; thus, communication is considered 2-way when 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 synchronised. 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: Full/DR — Indicating that the neighbour is a Designated Router (DR) Full/BDR — Indicationg that the neighbour is a Backup Designated Router (BDR) 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.

19.5. OSPF Network Types

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

State	Description
Broadcast Multi-Access	Indicates a topology where broadcast occurs. Examples include Ethernet, Token Ring and ATM. OSPF characteristics are: OSPF will elect DRs and BDRs Traffic to DRs and BDRs is multicast to 224.0.0.6. Traffic from DRs and BDRs to other routers is multicast to 224.0.0.5 Neighbours do not need to be manually specified.
Point-to-Point	Indicates a topology where two routers are directly connected. An example would be a point-to-point T1. OSPF characteristics are: OSPF will not elect DRs and BDRs All OSPF traffic is multicast to 224.0.0.5 Neighbours do not need to be manually specified
Point-to-Multipoint	Indicates a topology where one interface can connect to multiple destinations. Each connection between a source and destination is treated as a point-to- point link. For example, point to point-to-multipoint frame relay. OSPF characteristics are: OSPF will not elect DRs and BDRs. All OSPF traffic is multicast to 224.0.0.5. Neighbours do not need to be manually specified.
Non-broadcast Multi- access Network (NBMA)	Indicates a topology where one interface can connect to multiple destinations; however, broadcasts cannot be sent across a NBMA network. For example, Frame Relay. OSPF characteristics are: OSPF will elect DRs and BDRs. OSPF neighbours must be manually defined, so all OSPF traffic is unicast instead of multicast. Note: on non-broadcast networks, neighbours must be manually specified, as multicast hellos are not allowed.

19.6. The OSPF Hierarchy

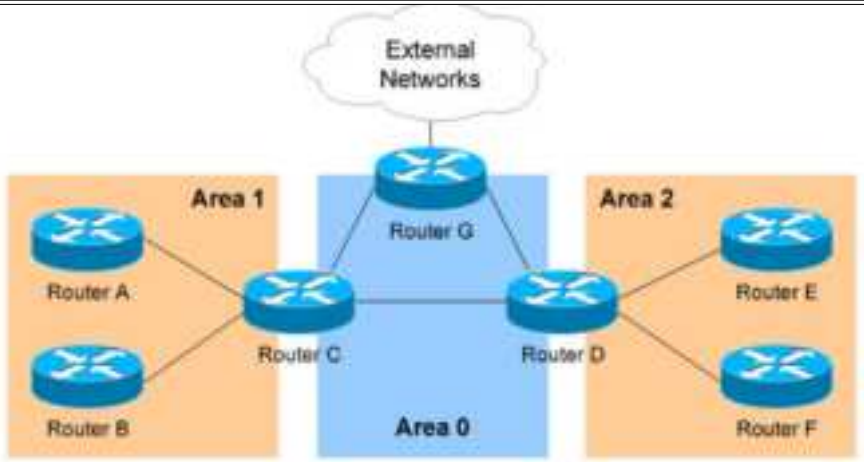
OSPF is a hierarchical system that separates an autonomous system into individual areas. OSPF traffic can either be:

- 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).



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.

19.7. 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.

19.8. Configuring OSPF Using The Web Interface

Configuration Package Used

Package	Sections
ospfd	routing
	network
	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.

19.8.1. Global Settings

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

The OSPF global settings configuration page

Web Field/UCI/Package Option	Description
Web: OSPF Enabled UCI: ospfd.ospfd.enabled Opt: enabled	Enables OSPF advertisements on router. Default: 0 — Disabled 1 — Enabled
Web: Router ID UCI: ospfd.ospfd.router_id Opt: router_id	This sets the router ID of the OSPF process. The router ID may be an IP address of the router, but need not be - it can be any arbitrary 32bit number. However it MUST be unique within the entire OSPF domain to the OSPF speaker. If one is not specified, then ospfd will obtain a router-ID automatically from the zebra daemon. Default — Empty Range —
Web: Make Default Router UCI: ospfd.ospfd.default_info_originate Opt: default_info_originate	Defines whether to originate an AS-External (type-5) LSA describing a default route into all external-routing capable areas, of the specified metric and metric type. Default: 0 — Disabled 1 — Enabled
Web: n/a UCI: ospfd.ospfd.vty_enabled Opt: vty_enabled	Enable vty for OSPFd (telnet to localhost:2604)
Web: n/a UCI: ospfd.ospfd.vrf Opt: vrf	Defines the VRF for OSPF Default — No VRF Range —

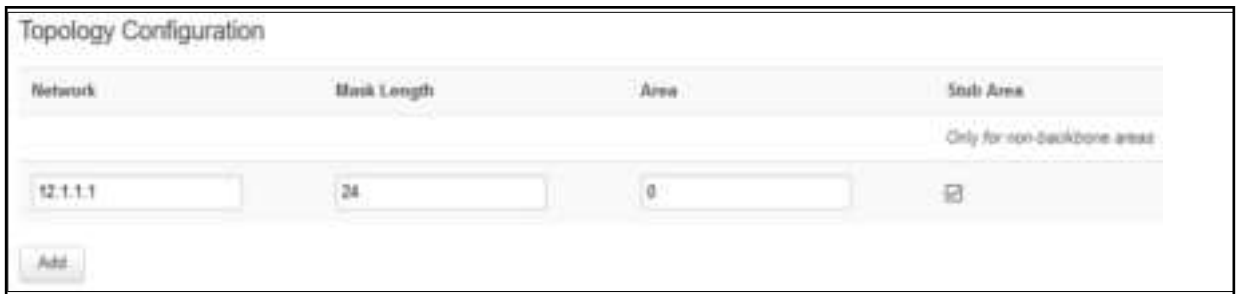
19.8.2. Topology Configuration

The Topology Configuration 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, that is, a 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 enable OSPF on an interface with address 12.1.1.129/25.



The OSPF topology configuration page

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

19.8.3. Interfaces Configuration

The Interfaces Configuration section contains settings to configure the OSPF interface. It defines interface configurations 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.

The OSPF interfaces configuration section

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Interface

UCI: ospfd.@interface[0].ospf_interface Opt: ospf_interface</td><td>Defines the interface name.</td></tr>
<tr><td>Web: Network Type

UCI: ospfd.@interface[0].network_type Opt: network_type</td><td>Defines the network type for specified interface.

<table><tr><td>Default</td><td>Autodetect: it will broadcast. If broadcast is not supported on that interface then use point-to-point.</td></tr><tr><td>broadcast</td><td></td></tr><tr><td>non-broadcast</td><td></td></tr><tr><td>point-to-point</td><td></td></tr><tr><td>point-to-multipoint</td><td></td></tr></table></td></tr>
<tr><td>Web: Passive

UCI: ospfd.@interface[0].passive

Opt: passive</td><td>Does not send hello packets on the given interface, but does advertise the interface as a stub link in the router-LSA (Link State Advertisement) for this router.

This allows you to advertise addresses on such connected interfaces without having to originate AS-External/Type-5 LSAs, which have global flooding scope, as would occur if connected addresses were redistributed into OSPF. This is the only way to advertise non-OSPF links into stub areas.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Hello Interval

UCI: ospfd.@interface[0].hello_interval

Opt: hello_interval</td><td>Defines the number of seconds for the Hello Interval timer value.

A hello packet will be sent every x seconds, where x is the configured hello interval value on the specified interface. This value must be the same for all routers attached to a common network.

The default is every 10 seconds for broadcast and point-to-point interfaces, and 30 seconds for non-broadcast and point-to- multipoint interfaces.

<table><tr><td>Default</td><td>10 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Dead interval

UCI: ospfd.@interface[0].dead_interval

Opt: dead_interval</td><td>Defines the number of seconds for the dead interval timer value used for wait timer and inactivity timer. This value must be the same for all routers attached to a common network.

The default is 40 seconds for broadcast and point-to-point interfaces, and 120 seconds for non-broadcast and point-to- multipoint interfaces. By default, the dead interval timer is four times the hello interval.

<table><tr><td>Default</td><td>40 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Routing priority

UCI: ospfd.@interface[0].priority

Opt: priority</td><td>Defines priority to become the designated router.

A value of 0 means never become a designated router; other values in the range 1-255 are allowed, with 255 being most likely to be a designated router, and 1 being least likely.

<table><tr><td>Default</td><td>1</td></tr><tr><td>Range</td><td>0-255</td></tr></table></td></tr>
<tr><td>Web: Authentication

UCI: ospfd.@interface[0].auth_mode

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

<table><tr><td>Default: no</td><td>No authentication</td></tr></table></td></tr>
</table>

Web Field/UCI/Package Option	Description
	md5 — Sets the interface with OSPF MD5 authentication. text — Sets the interface with OSPF simple password authentication.
Web: Text Auth. Key UCI: ospfd.@interface[0].text_auth_key Opt: text_auth_key	This command sets authentication string for text authentication. text_auth_key option can have length up to 8 characters. Displayed only when authentication is set to **text**.
Web: Key ID UCI: ospfd.@interface[0].key_id Opt: key_id	Specifies key ID. Must be unique and match at both ends. Displayed only when authentication is set to **MD5**.
Web: MD5 Auth. Key UCI: ospfd.@interface[0].md5_auth_key Opt: md5_auth_key	Specifies keyed MD5 chain. Displayed only when authentication is set to **MD5**.

19.9. 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'
```

19.10. Configuring 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.ospfd.vrf=datavrf
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].priority=1
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[0].priority=2
ospfd.@interface[1].auth_mode=md5
ospfd.@interface[1].key_id=1
ospfd.@interface[1].md5_auth_key=test
```

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'
option vrf 'datavrf'
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 priority '1'
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 priority '2'
option auth_mode 'md5'
option key_id '1'
option md5_auth_key 'test'
```

19.11. OSPF Diagnostics

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	U	0	0	0	eth0
192.168.101	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.

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 **<crtl-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
```

19.11.1. Quagga/Zebra Console OSPF

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
```

19.11.2. OSPF Debug Console

When option `vty_enabled` is enabled in the OSPF configuration, the OSPF debug console can be accessed for advanced OSPF debugging. For more information, read the Global Settings section above.

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:

```
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:

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:

```
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:

```
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
```

20. Configuring VRRP

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.

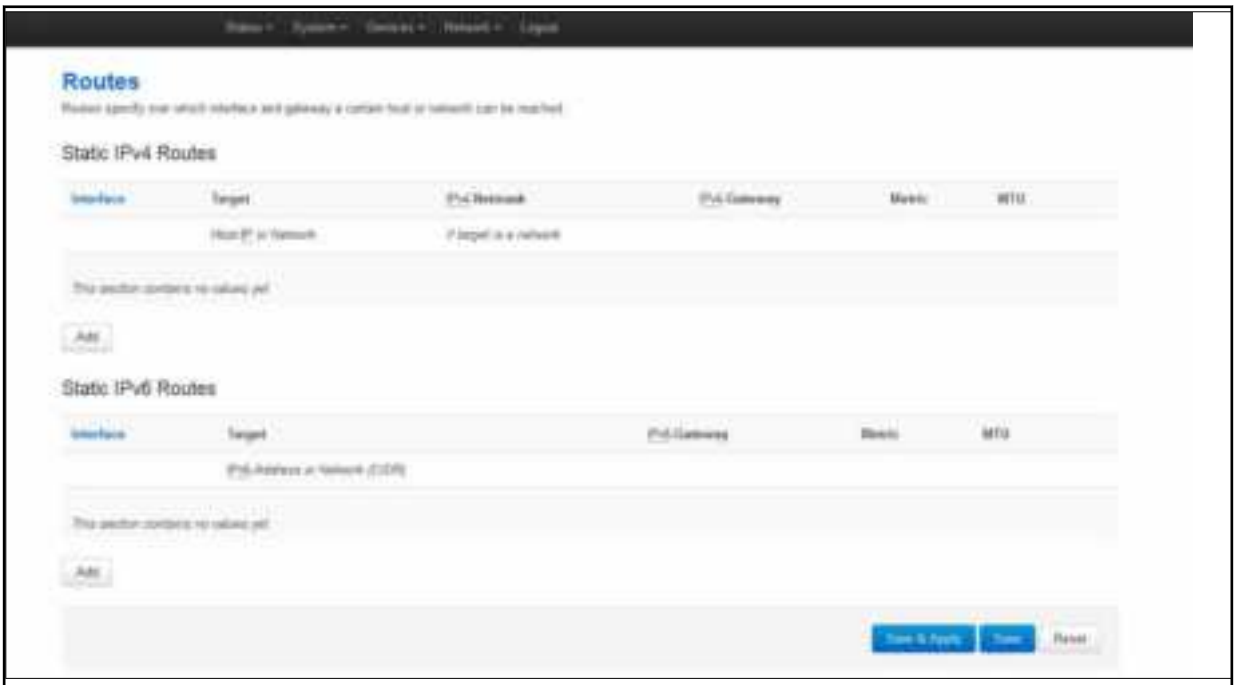
The VRRP protocol is implemented according to internet standard RFC2338.

Configuration package used

Package	Sections
vrrp	main vrrp_group

20.1. Configuring Static Routes Using The Web Interface

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



The routes page

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

Web Field/UCI/Package Option	Description
Web: Interface UCI: network.@route[0].interface Opt: interface	Specifies the logical interface name of the parent or master interface this route belongs to. It must refer to one of the defined interface sections.
Web: target UCI: network.@router[0].target Opt: target	Specifies the route network IP address.
Web: netmask UCI: network.@route[0].netmask Opt: netmask	Defines the route netmask. If omitted, 255.255.255.255 is assumed, which makes the target a host address.
Web: Gateway UCI: network.@route[0].gateway Opt: Gateway	Network gateway. If omitted, the gateway from the parent interface is taken. If set to 0.0.0.0 no gateway will be specified for the route.
Web: Metric UCI: network.@route[0].metric Opt: metric	Specifies the route metric to use. Default: 0 Range:
Web: MTU UCI: network.@route[0].mtu Opt:mtu	Defines a specific MTU for this route. If omitted, the MTU from the parent interface will be taken. Default: Blank Range:

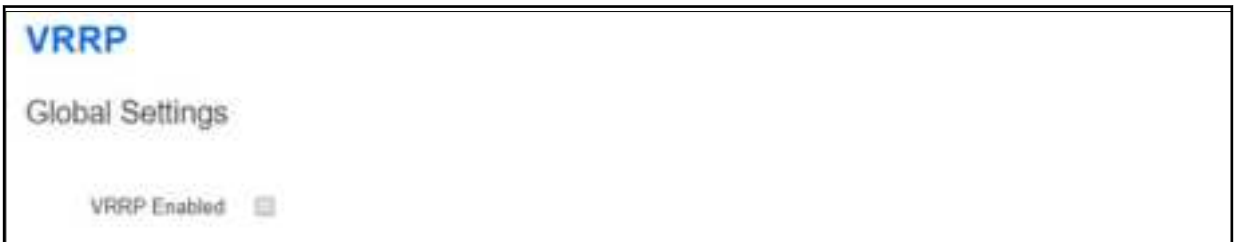
20.2. Configuring VRRP Using The Web Interface

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

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

20.3. VRRP Global Settings

The Global Settings section configures the vrrp package main section. To access configuration settings, click **ADD**.

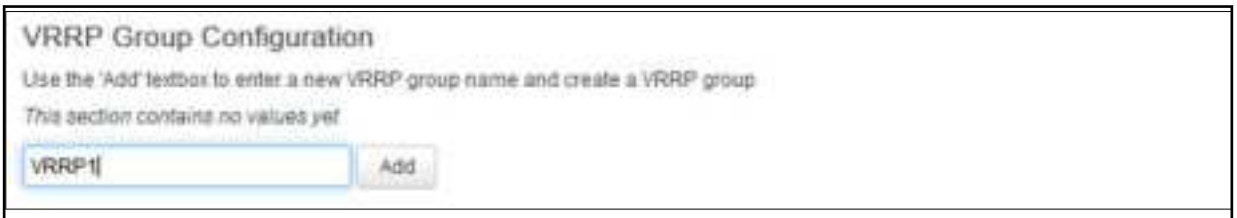


The VRRP global settings configuration page

Web Field/UCI/Package Option	Description
Web: VRRP Enabled UCI: vrrp.main.enabled Opt: Enabled	Globally enables VRRP on the router. Default: 0 — Disabled 1 — Enabled

20.4. VRRP Group Configuration Settings

The VRRP Group Configuration section configures vrrp package vrrp_group section. To access configuration settings, enter a VRRP group name and click **ADD**.



The VRRP group name configuration page

VRRP Group Configuration

Group enabled

Interface

LAN1 (no interfaces attached)

LAN2

LAN3

MOBILE1

PsAADSL

loopback

Interface to serve

Current State

Track interfaces

LAN1 (no interfaces attached)

LAN2

LAN3

MOBILE1

PsAADSL

loopback

Interfaces to monitor

Track IPsec Tunnel

IPsecTunnel1

IPsecTunnel2

IPsec connection(s) to monitor

Track IPsec Fail Time 300 Consider IPsec tunnel failed if tunnel is down for that many seconds

IPsec Connection IPsec connection to bring down/up when VRRP enters BACKUP/MASTER state

Start role BACKUP

Router ID 1

Priority 100

The VRRP group configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Group Enabled

UCI: vrrp.@vrrp_group[X].enabled

Opt: Enabled</td><td>Enables a VRRP group on the router.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Interface

UCI: vrrp.@vrrp_group[X].interface

Opt: interface</td><td>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.

<table><tr><td>Default</td><td>lan</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Track Interfaces

UCI: vrrp.@vrrp_group[X].track_iface

Opt: list track_iface</td><td>Defines one or more WAN interfaces that VRRP should monitor. If a monitored interface goes down on the master VRRP router, it goes into 'Fault' state and the backup VRRP router becomes the master.

Multiple interfaces are entered using uci set and uci add_list

commands. Example:

uci set vrrp.@vrrp_group[0].track_iface=wan1

uci add_list vrrp.@vrrp_group[0].track_iface=wan2

or using a list of options via package options

list track_iface 'wan1' list track_iface 'wan2'

<table><tr><td>Default</td><td>Wan</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Track IPsec Tunnel

UCI: vrrp.@vrrp_group[X].track_ipsec

Opt: list track_ipsec</td><td>Defines one or more IPSec tunnels that VRRP should monitor. If a monitored tunnel goes down on the master VRRP router for the configured Track IPSec Fail Time, it goes into 'Fault' state and the backup VRRP router becomes the master.

Multiple IPSec connections are entered using uci set and uci add_list commands. Example:

uci set vrrp.@vrrp_group[0].track_ipsec=Tunnel1

uci add_list vrrp.@vrrp_group[0].track_ipsec=Tunnel2

or using a list of options via package options

list track_ipsec 'Tunnel1' list track_ipsec 'Tunnel2'

<table><tr><td>Default: Blank</td><td>No IPSec connection to track</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Track IPsec Fail Time

UCI: vrrp.@vrrp_group[X]. track_ipsec_fail_sec

Opt: track_ipsec_fail_sec</td><td>Defines duration in seconds to determine IPSec tunnel failure.

<table><tr><td>Default</td><td>300 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: IPSec connection

UCI: vrrp.@vrrp_group[X].ipsec_connection

Opt: ipsec_connection</td><td>Sets which IPSec connection to bring up or down when VRRP enters 'backup/master' state.

Multiple IPSec connections are entered via the package option using a space separator. Example:

option ipsec_connection 'IPSecTunnel1 IPSecTunnel2'

<table><tr><td>Default</td><td>Blank. No IPSec connection to toggle</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Start role</td><td>Sets the initial role in which a VRRP router starts up. In a cluster of VRRP routes, set one as a master and the others as backup.</td></tr>
</table>

Web Field/UCI/Package Option	Description
UCI: vrrp.@vrrp_group[X].init_state Opt: init_state	Default: BACKUP MASTER
Web: Router ID UCI: vrrp.@vrrp_group[X].router_id 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. Default: 1 Range: 1-255
Web: Priority UCI: vrrp.@vrrp_group[X].priority 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. Default: 100 Range: 0-255
Web: Advert intvl UCI: vrrp.@vrrp_group[X].advert_int_sec Opt: advert_int_sec	Sets the VRRP hello value in seconds. This value must match the value set on a peer. Default: 120 seconds Range: Enable
Web: Password UCI: vrrp.@vrrp_group[X].password Opt: password	Sets the password to use in the VRRP authentication (simple password authentication method). This field may be left blank if no authentication is required.
Web: Virtual IP UCI: vrrp.@vrrp_group[X].virtual_ipaddr Opt: virtual_ipaddr	Sets the virtual IP address and mask in prefix format. For example, '11.1.1.99/24'. All co-operating VRRP routers serving the same LAN must be configured with the same virtual IP address.
Web: GARP delay UCI: vrrp.@vrrp_group[X].garp_delay_sec Opt: garp_delay_sec	Sets the gratuitous ARP message sending delay in seconds.

20.5. Configuring VRRP Using Command Line

The configuration file is stored on /etc/config/vrrp. There are two config sections: main and vrrp_group.

You can configure multiple VRRP groups. By default, all VRRP group instances are named 'vrrp_group'. Instances are 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'
```

20.5.1. 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
```

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'
```

20.6. VRRP Diagnostics

VRRP Process using UCI

The VRRP process has its own subset of commands.

```
root@VA_router:~# /etc/init.d/vrrp
Syntax: /etc/init.d/vrrp [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
```

To restart VRRP, enter:

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

21. Configuring RIP

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.

21.1. 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 utilises 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.

21.2. 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

The Merlin's **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.

21.3. Configuring RIP Using The Web Interface

Configuration package used

Package	Sections
ripd	routing interface key_chain offset

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.

RIP global settings

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

The RIP global settings configurations page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: RIP Enabled

UCI: ripd.ripd.enabled

Opt: enabled</td><td>Enables RIP advertisements on router.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: RIP Version

UCI: ripd.ripd.version

Opt: version</td><td>Specifies the RIP version that will be used. Version 2 is recommended.

<table><tr><td>1</td><td>RIP version 1</td></tr><tr><td>Default: 2</td><td>RIP version 2</td></tr></table></td></tr>
<tr><td>Web: Network/Interface

UCI: ripd.ripd.network

Opt: list network</td><td>Defines the list of the interfaces that will be used to advertise RIP packets.

Format: A.B.C.D/mask or interface name

Multiple RIP interfaces are entered using uci set and uci add_list commands. Example:

uci set ripd.ripd.network=lan1

uci add_list ripd.ripd.network=lan2

or using a list of options via package options

list network ‘lan1’

list network ‘lan2’</td></tr>
<tr><td>Web: RIP Neighbor Address

UCI: ripd.ripd.neighbor

Opt: list neighbor</td><td>Specifies the list of RIP neighbours. When a neighbour does not understand multicast, this command is used to specify neighbours. In some cases, not all routers will be able to understand multicasting, where packets are sent to a network or a group of addresses. In a situation where a neighbour cannot process multicast packets, it is necessary to establish a direct link between routers. The neighbour command allows the network administrator to specify a router as a RIP neighbour.

Multiple RIP neighbours are entered using uci set and uci add_list commands. Example:

uci set ripd.ripd.neighbor=1.1.1.1

uci add_list ripd.ripd.neighbor=2.2.2.2

or using a list of options via package options

list neighbor ‘1.1.1.1’

list neighbor ‘2.2.2.2’</td></tr>
<tr><td>Web: Update Timer

UCI: ripd.ripd.tb_update_sec

Opt: tb_update_sec</td><td>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.

<table><tr><td>Default</td><td>30</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Timeout Timer

UCI: ripd.ripd.tb_timeout_sec

Opt: tb_timeout_sec</td><td>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.

<table><tr><td>Default</td><td>180</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Garbage Collect Timer

UCI: ripd.ripd.tb_garbage_sec Opt: tb_garbage_sec</td><td>Upon expiration of the garbage-collection timer, the route is finally removed from the routing table. This timer starts when Timeout timer expires or when route is advertised as “unreachable”.

The reason for using this two-stage marking and deleting removal method is to give the router that declared the route no longer reachable a chance to propagate</td></tr>
</table>

Web Field/UCI/Package Option	Description
	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. Default: 120 Range:
Web: Make Default Router UCI: ripd.ripd.default_info_originate Opt: default_info_originate	Advertising a default route via RIP. Default: 0 Disabled 1 Enabled
Web: Redistribute Kernel Routes UCI: ripd.ripd.redistribute_kernel_routes Opt: redistribute_kernel_routes	Redistributes routing information from kernel route entries into the RIP tables. Default: 0 Disabled 1 Enabled
Web: n/a UCI: ripd.ripd.vty_enabled Opt: vty_enabled	Enable vty for RIPd (telnet to localhost:2602).

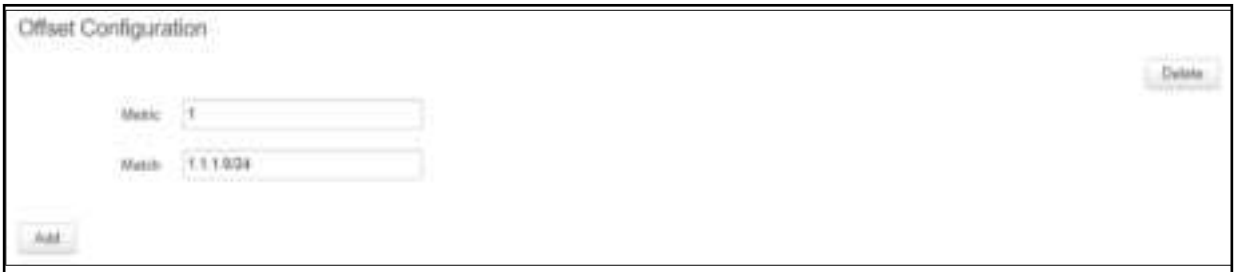
21.3.1. RIP Interfaces Configuration

The RIP interfaces configuration page

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

21.3.2. 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.



The RIP offset configuration section

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

21.3.3. 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.

The MD5 authentication key chains configuration section

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

21.4. 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'
```

21.4.1. 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
```

21.4.2. 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'
```

21.5. RIP Diagnostics

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	U	0	0	0	eth0
192.168.101	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.

21.5.1. 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 foreground, 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
```

21.5.2. Quagga/Zebra Console RIP

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:

```
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
```

21.5.3. RIP Debug Console

When option vty_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:

```
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:

```
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)
```

22. Configuring PIM And IGMP Interfaces

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.

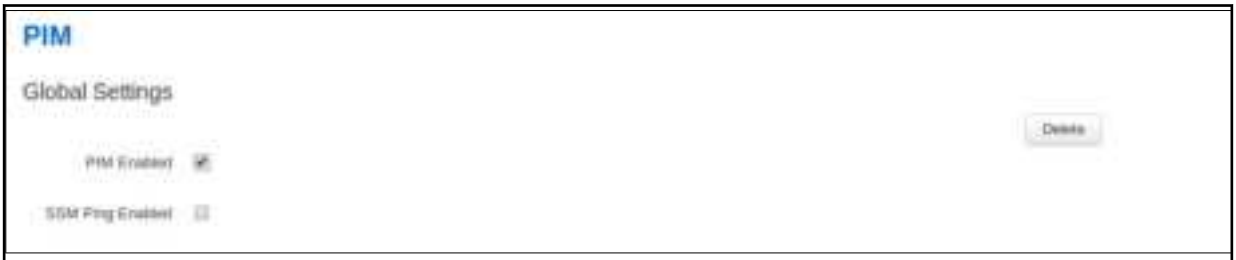
A multicast group is an arbitrary group of receivers that expresses an interest in receiving a particular data stream. The receivers (the designated multicast group) are interested in receiving a data stream from the source. They indicate this by sending an Internet Group Management Protocol (IGMP) host 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 summarise: 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.

Configuration Package Used

Package	Sections
pimd	pimd interface

22.1. 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**.



The PIM global settings interface

22.1.1. Global Settings

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

22.1.2. Interfaces Configuration

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

To save your configuration updates, click **Save & Apply**.

22.2. 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:

```
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
```

23. 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, particularly 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.

Configuration package used

Package	Sections
multiwan	config wan

23.1. Configuring Multi-WAN Using The Web Interface

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

The multi-WAN page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enable

UCI: multiwan.config.enabled

Opt: enabled</td><td>Enables or disables Multi-WAN.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Preempt

UCI: multiwan.config.preempt

Opt: preempt</td><td>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

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Alternate Mode

UCI: multiwan.config.alt_mode

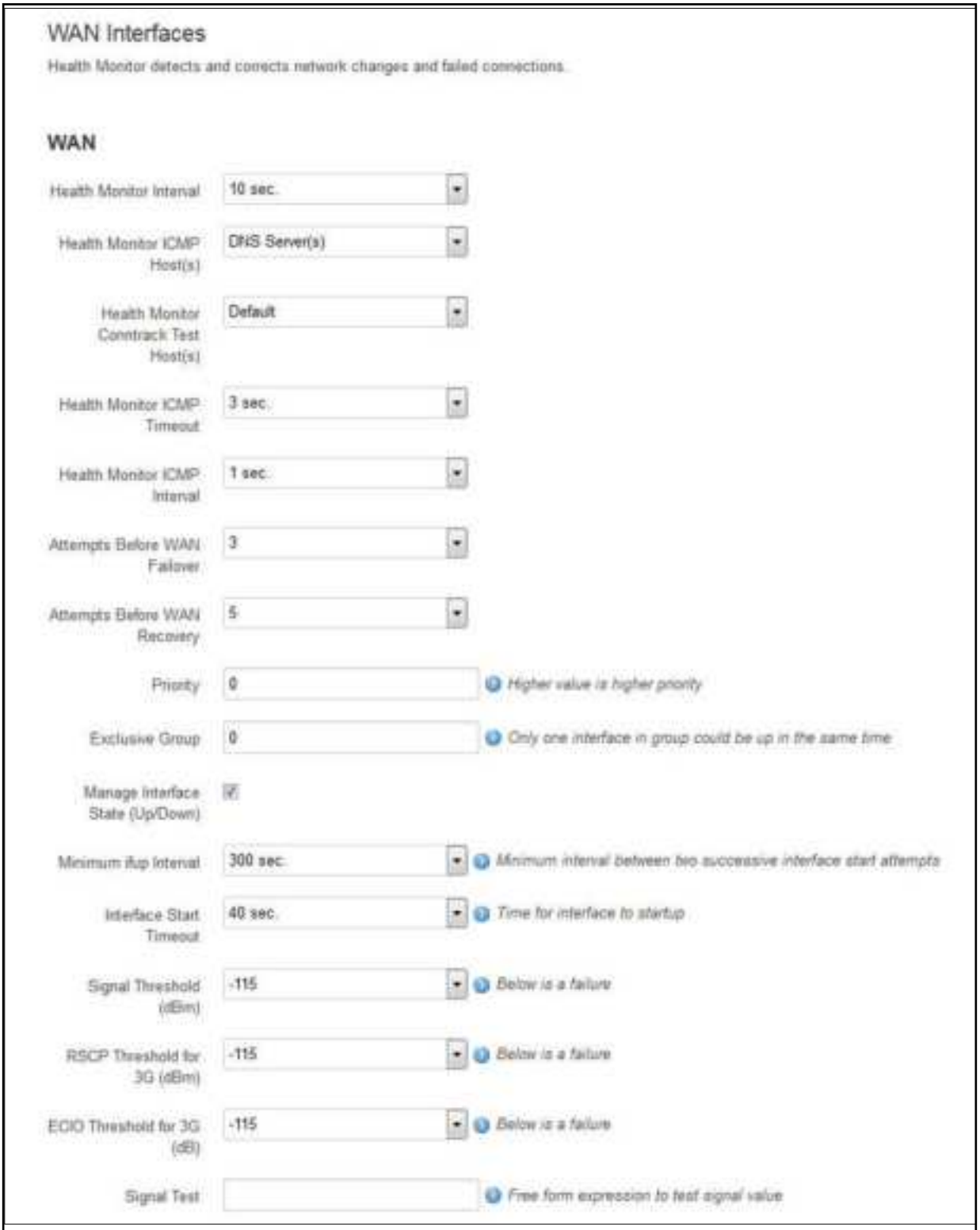
Opt: alt_mode</td><td>Enables or disables alternate mode for Multi-WAN. If enabled the router will use an alternate interface after reboot.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
</table>

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.



Example interface showing failover traffic destination as the added multi-WAN interface

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Health Monitor Interval

UCI: multiwan.wan.health_interval

Opt: health_interval</td><td>Sets the period to check the health status of the interface. The Health Monitor interval will be used for:

• Interface state checks
• Ping interval
• Signal strength checks

The health monitor interval has a granularity of 5 seconds. Configured values will be rounded up to the next 5 second value.

<table><tr><td>Default: 10</td><td>Perform a health check every 10 seconds</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Health Monitor ICMP Host(s)

UCI: multiwan.wan.icmp_hosts

Opt: icmp_hosts</td><td>Sends health ICMPs to configured value DNS servers by default. Configure to any address.

<table><tr><td>Disable</td><td>Disables the option</td></tr><tr><td>Default: DNS servers</td><td>DNS IP addresses will be used</td></tr><tr><td>WAN Gateway</td><td>Gateway IP address will be used</td></tr><tr><td>Custom</td><td>Ability to provide IP address. Multiple pings targets can be entered, comma separated. Pings to both must fail for health check to fail.

Example: option icmp_hosts '1.1.1.1,2.2.2.2'</td></tr></table></td></tr>
<tr><td>Web: Health Monitor Conntrack Test Host(s)

UCI: multiwan.wan.conntrack_hosts

Opt: conntrack_hosts</td><td>Conntrack is the feature used to track if there is any traffic to and from an IP destination within the health interval.

The Conntrack_hosts option defines the IP for conntrack to track, usually the icmp_host IP is used.

If traffic to the conntrack_hosts IP is detected then multiwan does not send a ping health check to the icmp_host; otherwise a ping is sent as normal to the icmp_host.

By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated.

Conntrack is generally used to limit the traffic sent on a GSM network.

<table><tr><td>Default</td><td>Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.</td></tr><tr><td>Disable</td><td>Conntrack disabled.</td></tr><tr><td>Custom</td><td>Specifies an IP other than the icmp_host for conntrack to track.</td></tr></table></td></tr>
<tr><td>Web: Health Monitor ICMP Timeout

UCI: multiwan.wan.timeout

Opt: timeout</td><td>Sets Ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.

<table><tr><td>Default: 3</td><td>Wait 3 seconds for ping reply.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Health Monitor ICMP Interval

UCI: multiwan.wan.icmp_interval

Opt: icmp_interval</td><td>Defines the interval between multiple pings sent at each health check

<table><tr><td>Default</td><td>1</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Health Monitor ICMP Count

UCI: multiwan.wan.icmp_count

Opt: icmp_count</td><td>Defines the number of pings to send at each health check.

<table><tr><td>Default</td><td>1</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Attempts Before WAN Failover

UCI: multiwan.wan.health_fail_retries

Opt: health_fail_retries</td><td>Sets the amount of health monitor retries before the interface is considered a failure.

<table><tr><td>Default</td><td>3</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web Field/UCI/Package Option	Description
Web: Attempts Before WAN Recovery UCI: multiwan.wan.health_recovery_retries Opt: health_recovery_retries	Sets the number of health monitor checks before the interface is considered healthy. Only relevant if pre-empt mode is enabled. Default: 5 Range:
Web: Priority UCI: multiwan.wan.priority Opt: priority	Specifies the priority of the interface. The higher the value, the higher the priority. Default: 0 Range:
Web: Manage Interface State (Up/Down) UCI: multiwan.wan.manage_state Opt: manage_state	Defines whether multi-wan will start and stop the interface. Default: 0 Disabled 1 Enabled
Web: Exclusive Group UCI: multiwan.wan.exclusive_group Opt: exclusive_group	Defines the group to which the interface belongs; only one interface can be active. Default: 0 Range:
Web: Minimum ifup Interval UCI: multiwan.wan.ifup_retry_sec Opt: ifup_retry_sec	Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled. Default: 300 Retry primary interface every 300 seconds 1 Enabled
Web: Interface Start Timeout UCI: multiwan.wan.ifup_timeout Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail. Default: 40 seconds Range:
Web: Signal Threshold (dBm) UCI: multiwan.wan.signal_threshold Opt: signal_threshold	Specifies the minimum signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for sig_dbm in mobile diagnostics.-115. Default: Disabled Range: -46 to -115 dBm
Web: RSCP Threshold (dBm) UCI: multiwan.wan.rscp_threshold Opt: rscp_threshold	Specifies the minimum RSCP signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for rscp_dbm in mobile diagnostics. Default: -115 Disabled Range: -46 to -115 dBm
Web: ECIO Threshold (dB) UCI: multiwan.wan.ecio_threshold Opt: ecio_threshold	Specifies the minimum ECIO signal strength in dB before considering if the interface fails signal health check. Uses the value stored for ecio_db in mobile diagnostics. Default: -115 Disabled Range: -46 to -115 dBm
Web: Signal Test UCI: multiwan.wan.signal_test Opt: signal_test	Defines a script to test various signal characteristics in multiwan signal test. For example: option signal_test '(tech == 0) then (sig_dbm > -70) else (rscp_dbm > -105 and ecio_db > -15)' This states that when technology is GSM, a health fail is determined when signal strength is less than -70dBm. When technology is not GSM a health fail occurs when either rscp_dbm falls below -105dBm or ecio_db falls below -15dB Note: a signal test can also take a UDS script name as a parameter, for example: `Option signal_test 'uds(script_name)'`

Web Field/UCI/Package Option	Description
	Tech values are: 0 GSM 1 GSM Compact 2 UTRAN 3 GSM W/EGPRS 4 UTRAN w/HSPDA 5 UTRAN w/HSUPA 6 UTRAN W/HSUPA and HSDPA 7 E-UTRAN

23.2. 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.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
```

23.3. Multi-WAN Diagnostics

The multiwan 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 multiwan 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 multiwan 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 multiwan 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.

24. Automatic Operator Selection

This section describes how to configure and operate the Automatic Operator Selection feature of the Merlin router.

When the roaming SIM is connected, the radio module has the ability to scan available networks. The router, using mobile and multiwan packages, finds available networks to create and sort interfaces according to their signal strength. These interfaces are used for failover purposes.

Configuration Package Used

Package	Sections
Multiwan	General, interfaces
Mobile	Main, template interfaces
Network	2G/3G/4G interface

24.1. Configuring Automatic 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, particularly 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 typically three scenarios:

- Primary Mobile Provider (PMP) + roaming: pre-empt enabled
- PMP + roaming: pre-empt disabled
- No PMP + roaming

24.1.1. Scenario 1: PMP + Roaming: Pre-Empt Enabled

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.

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. 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.
4. 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).
5. Disconnect auto-generated interface.
6. If the interface was disconnected due to health check failure thenconnect 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.

Create a primary predefined interface

In the web interface top menu, go to **Network -> Interfaces**. The Interfaces page appears.

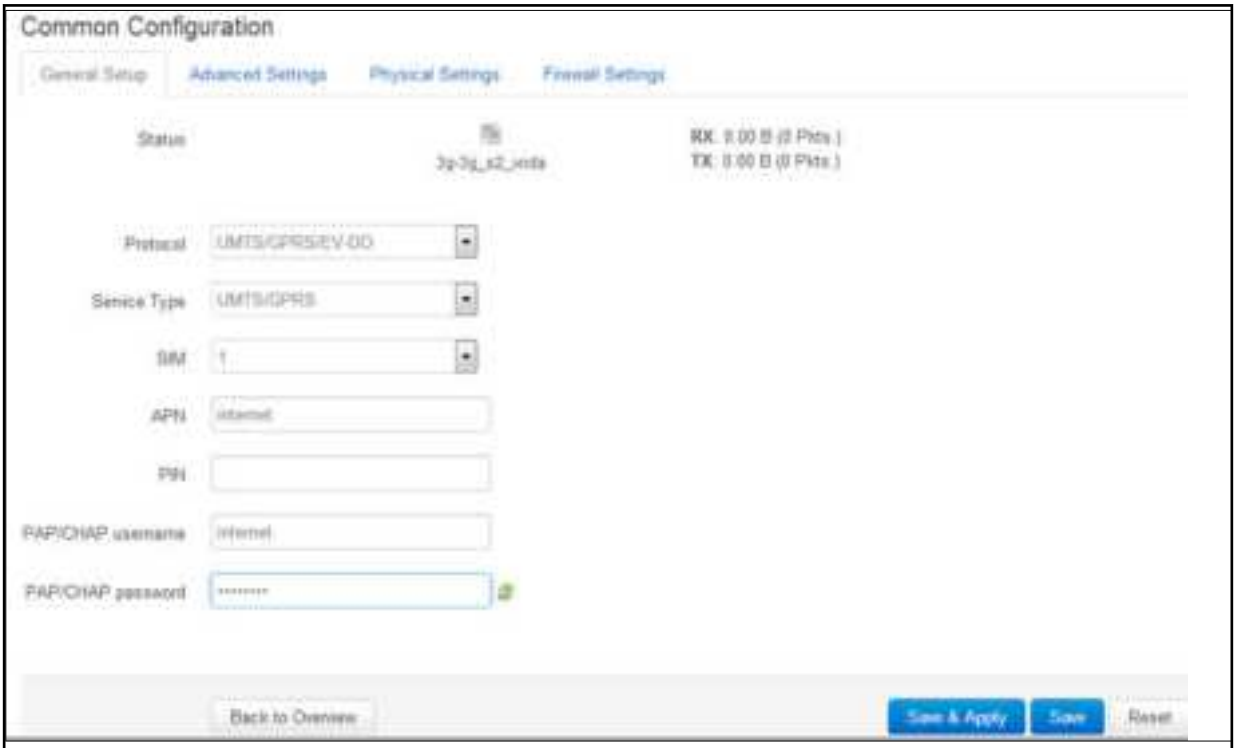
The interface overview page

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

The create interface page

Web Field/UCI/Package Option	Description
Web: Name of the new interface UCI: network.3g_s<sim-number>_<short-operator-name> Opt: 3g_<sim-number>_<short-operator-name>	Type the name of the new interface. Type the interface name in following format: 3g_s<sim-number>_<short-operator-name>. Where<sim-number> is number of roaming SIM (1 or 2) and <short-operator-name> is first four alphanumeric characters of operator name (as reported by 'AT+COPS=?' command). Type the short operator name in lower case, for example: Operator name: First four alphanumeric numbers Vodafone UK: voda 02 -UK: o2uk Orange: oran
Web: Protocol of the new interface UCI: network.[..x..].proto Opt: proto	Protocol type. Select LTE/UMTS/GPRS/EV-DO. Option: Description Static: Static configuration with fixed address and netmask. DHCP Client: Address and netmask are assigned by DHCP. Unmanaged: Unspecified IPv6 -in-IPv4 (RFC4213): IPv4 tunnels that carry IPv6 IPv6 over IPv4: IPv6 over IPv4 tunnel GRE: Generic Routing Encapsulation IOT: L2: Layer 2 Tunnelling Protocol PPP: Point to Point Protocol PPPoE: Point to Point Protocol over Ethernet PPPoATM: Point to Point Protocol over ATM LTE/UMTS/GPRS/EV-DO: CDMA, UMTS or GPRS connection using an AT-style 3G modem
Web: Create a bridge over multiple interfaces UCI: network.[..x..].type Opt: type	Enables bridge between two interfaces. 0: Disabled 1: Enabled
Web: Cover the following interface UCI: Opt: ifname	Selects interfaces for bridge connection.

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



The common configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Protocol

UCI: network.[..x..].proto

Opt: proto</td><td>Protocol type. Select LTE/UMTS/GPRS/EV-DO.

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Static</td><td>Static configuration with fixed address and netmask.</td></tr>
<tr><td>DHCP Client</td><td>Address and netmask are assigned by DHCP.</td></tr>
<tr><td>Unmanaged</td><td>Unspecified</td></tr>
<tr><td>IPv6 -in-IPv4 (RFC4213)</td><td>IPv4 tunnels that carry IPv6</td></tr>
<tr><td>IPv6 over IPv4</td><td>IPv6 over IPv4 tunnel</td></tr>
<tr><td>GRE</td><td>Generic Routing Encapsulation</td></tr>
<tr><td>IOT</td><td></td></tr>
<tr><td>L2TP</td><td>Layer 2 Tunnelling Protocol</td></tr>
<tr><td>PPP</td><td>Point to Point Protocol</td></tr>
<tr><td>PPPoE</td><td>Point to Point Protocol over Ethernet</td></tr>
<tr><td>PPPoATM</td><td>Point to Point Protocol over ATM</td></tr>
<tr><td>LTE/UMTS/GPRS/EV-DO</td><td>CDMA, UMTS or GPRS connection using an AT-style 3G modem</td></tr>
</table></td></tr>
<tr><td>Web: Service Type

UCI: network.[..x..].service

Opt: service</td><td>Service type that will be used to connect to the network.

<table>
<tr><td>gprs_only</td><td>Allows GSM module to only connect to GPRS network.</td></tr>
<tr><td>lte_only</td><td>Allows GSM module to only connect to LTE network.</td></tr>
<tr><td>cdma</td><td>Allows GSM module to only connect to CDMA network.</td></tr>
<tr><td>auto</td><td>GSM module will automatically detect the best available technology code.</td></tr>
</table></td></tr>
<tr><td>Web: SIM

UCI: network.[..x..].sim

Opt: sim</td><td>Select SIM1 or SIM2.

<table>
<tr><td>auto</td><td>Automatically detects which SIM slot is used.</td></tr>
<tr><td>SIM 1</td><td>Selects SIM from slot 1</td></tr>
<tr><td>SIM 2</td><td>Selects SIM from slot 2</td></tr>
</table></td></tr>
<tr><td>Web: APN

UCI: network.[..x..].apn

Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: APN username

UCI: network.[..x..].username

Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: APN password

UCI: network.[..x..].password

Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: Modem Configuration

UCI: N/A

Opt: N/A</td><td>Click the link if you need to configure additional options from Mobile Manager.</td></tr>
</table>

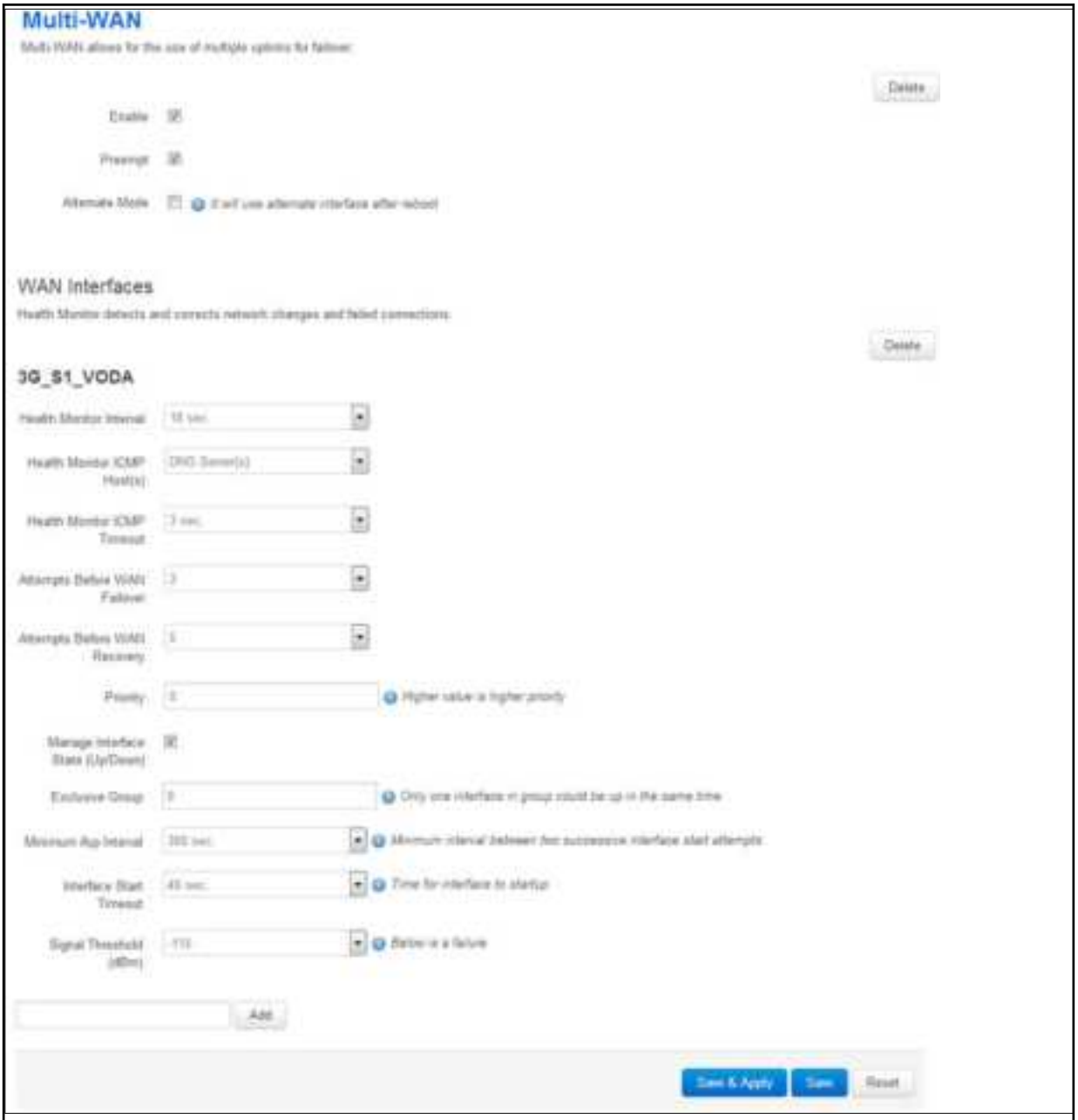
Click **Save & Apply**.

Set Multi-WAN Options For Primary Predefined Interface

On the web interface go to **Network -> Multi-Wan**. The Multi-WAN page appears.

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.



The Multi-WAN page

Web Field/UCI/Package Option	Description
Web: Enable UCI: multiwan.config.enabled Opt: enabled	Enables multiwan. 0 Disabled 1 Enabled
Web: Preempt UCI: multiwan.config.preempt Opt: preempt	Enables or disables pre-emption for multiwan. If enabled, the router will keep trying to connect to a higher priority interface depending on timer set. 0 Disabled 1 Enabled
Web: Alternate Mode UCI: multiwan.config.alt Opt: alt	Enables or disables alternate mode for multiwan. If enabled, the router will use an alternate interface after reboot. 0 Disabled 1 Enabled
Web: WAN Interfaces UCI: multiwan.3g_s<sim-number>_<short-operator-name> Opt: 3g_s<sim-number>_<short- operator-name>	Provide the same interface name as chosen in multiwan section below and click **Add**.
Web: Health Monitor Interval UCI: multiwan.[..x..].health_interval Opt: health_interval	Sets the period to check the health status of the interface. The Health Monitor interval will be used for: • Interface state checks • Ping interval • Signal strength checks
Web: Health Monitor ICMP Host(s) UCI: multiwan.[..x..].icmp_hosts Opt: icmp_hosts	Specifies the target IP address for ICMP packets. Disable — Disables the option DNS servers — DNS IP addresses will be used. WAN Gateway — Gateway IP address will be used. Custom — Ability to provide IP address.
Web: Health Monitor Conntrack Test Host(s) UCI: multiwan.wan.conntrack_hosts Opt: conntrack_hosts	Conntrack is the feature used to track if there is any traffic to and from an IP destination within the health interval. Conntrack_hosts option defines the IP for conntrack to track – usually the icmp_host IP is used. If traffic to the conntrack_hosts IP is detected then multiwan does not send a ping health check to the icmp_host otherwise a ping is sent as normal to the icmp_host. By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated. Conntrack is generally used to limit the traffic sent on a GSM network. Default — Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes. Disable — Conntrack is Disabled. Custom — Specifies an IP other than the icmp_host for conntrack to track.
Web: Health Monitor ICMP Timeout UCI: multiwan.[..x..].timeout Opt: timeout	Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at. 3 — Wait 3 seconds for ping reply. Range

Web Field/UCI/Package Option	Description
Web: Health Monitor ICMP Interval UCI: multiwan.wan.icmp_interval Opt: icmp_interval	Defines the interval between multiple pings sent at each health check. 1 | Range |
Web: Health Monitor ICMP Count UCI: multiwan.wan.icmp_count Opt: icmp_count	Defines the number of pings to send at each health check. 1 | Range | Enabled
Web: Attempts Before WAN Failover UCI: multiwan. [..x..].health_fail_retries Opt: health_fail_retries	Sets the amount of health monitor retries before the interface is considered a failure. 3 | Range |
Web: Attempts Before WAN Recovery UCI: multiwan. [..x..].health_recovery_retries Opt: health_recovery_retries	Sets the number of health monitor checks before the interface is considered healthy. Only relevent if pre-empt mode is enabled. 5 | Range |
Web: Priority UCI: multiwan.[..x..].priority Opt: priority	Specifies the priority of the interface. The higher the value, the higher the priority. This multiwan interface priority must be higher than the one specified in the priority field in the 'Roaming Interface Template' page described in the following section. 0 | Range |
Web: Exclusive Group UCI: multiwan.[..x..].exclusive_group Opt: exclusive_group	Defines the group to which the interface belongs; only one interface can be active. 0 | Range |
Web: Manage Interface State (Up/Down) UCI: multiwan.[..x..].manage_state Opt: manage_state	Defines whether multiwan will start and stop the interface. Select **Enabled**. 0 | Disabled 1 | Enabled
Web: Minimum ifup Interval UCI: multiwan.[..x..].ifup_retry_sec Opt: ifup_retry_sec	Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled.
Web: Interface Start Timeout UCI: multiwan.[..x..].ifup_timeout Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail. Choose timer greater than 120 seconds. 40 | 40 Seconds Range |
Web: Signal Threshold (dBm) UCI: multiwan.[..x..].signal_threshold Opt: signal_threshold	Specifies the minimum signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for sig_dbm in mobile diagnostics. -115 | Disabled Range | -46 to -115dBm
Web: RSCP Threshold (dBm) UCI: multiwan.[..x..].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.

Web Field/UCI/Package Option	Description
Opt: rscp_threshold	
Web: ECIO Threshold (dB) UCI: multiwan.[..x..].ecio_threshold Opt: ecio_threshold	Specifies the minimum ECIO signal strength in dB before considering if the interface fails signal health check. Uses the value stored for ecio_db in mobile diagnostics.
Web: Signal Test UCI: multiwan.[..x..].signal_test Opt: signal_test	Defines script to test various signal characteristics in multiwan signal test. For example: option signal_test '(tech == 0) then (sig_dbm > -70) else (rscp_dbm > -105 and ecio_db > -15)' This states that when technology is GSM a health fail is determined when signal strength is less than -70dBm. When technology is not GSM a health fail occurs when either rscp_dbm falls below -105dBm or ecio_db falls below -15dB. Tech values are: 0 GSM 1 GSM Compact 2 UTRAN 3 GSM w/EGPRS 4 UTRAN w/HSPDA 5 UTRAN w/HSUPA 6 UTRAN w/HSUPA and HSDPA 7 E-UTRAN

Click **Save**.

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 five sections in the mobile manager page:

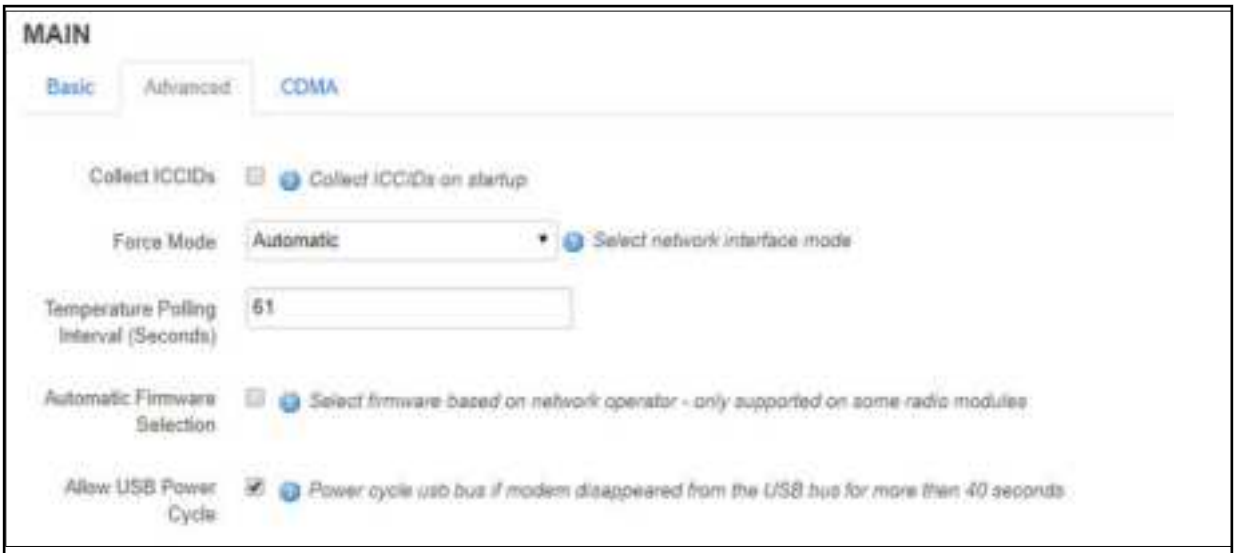
Section	Description
Basic settings	Enable SMS, configure SIM pin code, select roaming SIM, collect ICCCIDs and set IMSI.
Advanced	Configure advanced options such as collect ICCIDs and temperature polling interval.
CDMA*	CDMA configuration
Callers	Configure callers that can use SMS.
Roaming Interface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

24.2. Mobile Manager: Basic Settings

The mobile manager basic page

Web Field/UCI/Package Option	Description
Web: SMS Enable UCI: mobile.main.sms Opt: sms	Enables or disables SMS functionality. 0 Disabled 1 Enabled
Web: PIN code for SIM1 UCI: mobile.main.sim1pin Opt: sim1pin	Depending on the SIM card specifies the pin code for SIM 1. Blank Disabled Range Depends on the SIM provider
Web: PIN code for SIM2 UCI: mobile.main.sim2pin Opt: sim2pin	Depending on the SIM card specify the pin code for SIM 2. Blank Range Depends on the SIM provider
Web: LTE bands for SIM1 UCI: mobile.main.sim1_lte_bands Opt: sim1_lte_bands	Depending on the SIM card specify the LTE bands for SIM 1. Comma delimiter. Example: option sim1_lte_bands '3,20' Limits LTE bands to 3 and 20. Note: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200 and Asiatel.
Web: LTE bands for SIM2 UCI: mobile.main.sim2_lte_bands Opt:sim2_lte_bands	Depending on the SIM card specifies the LTE bands for SIM 2. Comma delimiter. Example: option sim1_lte_bands '3,20' Limits LTE bands to 3 and 20. Note: currently only supported by Hucom/Wetelcom, SIMCom7100, Cellient MPL200 and Asiatel.

24.3. Mobile Manager: Advanced Settings



The mobile manager advanced page

Web Field/UCI/Package Option	Description
Web: Collect ICCIDs UCI: mobile.main.init_get_iccids Opt: init_get_iccids	Enables or disables integrated circuit card identifier ICCID's collection functionality. If enabled then both SIM 1 and SIM 2 ICCIDs will be collected otherwise it will default to SIM 1. This will be displayed under mobile stats. 0 Disabled 1 Enabled
Web: Force Mode UCI: mobile.main.force_mode Opt: force_mode	Defines whether to operate mobile modem in PPP or Ethernet mode. The mode will be dependent on the service provided by the mobile provider. In general, this is Ethernet mode (default). Automatic Ethernet mode (option not present) PPP Enable PPP mode
Web: Temperature Polling Interval UCI: mobile.main.temp_poll_interval_sec Opt: temp_poll_interval_sec	Defines the time in seconds to poll the mobile module for temperature. Set to **0** to disable. 61 61 seconds 1 Enabled
Web: Automatic Firmware Selection UCI: mobile.main.enable_firmware_autoselect Opt: enable_firmware_autoselect	Defines whether to use time obtained from the mobile carrier to update the system clock when NTP is enabled. 0 Disabled 1 Enabled
Web: Allow USB Power Cycle UCI: mobile.main.allow_usb_powercycle Opt: allow_usb_powercycle	Enables the selection of an operator-specific firmware in the radio module. The selection is based on the ICCID of the used SIM. At module initialisation the IMSI is checked and if necessary the correct firmware image in the module will be activated. **Note:** activation of the firmware will lead to delayed startup of the network interface associated with the radio module. **Note:** this feature is currently only supported for the Telit LE910NA V2 module. Here a Verizon-specific firmware will be selected if the ICCID starts with "891480". 0 Disabled 1 Enabled
Web: n/a UCI: mobile.main.disable_time Opt: disable_time	Defines whether to use time obtained from the mobile carrier to update the system clock when NTP is enabled. 0 Disabled 1 Enabled

24.4. Mobile Manager: CDMA Settings

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

MAIN

Basic | Advanced | CDMA

IMSI

If specified over-writes IMSI stored in radio module

HDR Auth User ID

AN-PPP user id. Supported on Cellient module only

HDR Auth Password

AN-PPP password. Supported on Cellient module only

Ordered Registration triggers module reboot

Station Class Mark

Slot Cycle Index

Slot Mode

Mobile Directory Number

MOB_TERM_HOME registration flag

MOB_TERM_FOR_SID registration flag

MOB_TERM_FOR_NID

The mobile manager CDMA page

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

Web Field/UCI/Package Option	Description
Web: Preferred Serving System UCI: mobile.main.cdma_preferred_serving_system Opt: cdma_preferred_serving_system	The CDMA Preferred Serving System(A/B). Default: 5 Range: 0-7
Web: Digital Analog Mode Preference UCI: cdma_digital_analog_mode_preference Opt: cdma_digital_analog_mode_preference	Digital/Analog Mode Preference. Default: 4 Range:
Web: Primary Channel A UCI: mobile.main.cdma_primary_channel_a Opt: cdma_primary_channel_a.	Allows the primary channel (A) to be changed. Default: 283 Range: 1-2016 Any band class 5 channel number
Web: Primary Channel B UCI: mobile.main.cdma_primary_channel_b Opt: cdma_primary_channel_b	Allows the primary channel (B) to be changed. Default: 384 Range: 1-2016 Any band class 5 channel number
Web: Secondary Channel A UCI: mobile.main.cdma_secondary_channel_a Opt: cdma_secondary_channel_a	Allows the secondary channel (A) to be changed. Default: 691 Range: 1-2016 Any band class 5 channel number
Web: Secondary Channel B UCI: mobile.main.cdma_secondary_channel_b Opt: cdma_secondary_channel_b	Allows the secondary channel (B) to be changed. Default: 777 Range: 1-2016 Any band class 5 channel number
Web: Preferred Forward & Reverse RC UCI: mobile.main.cdma_preferred_forward_and_re verse_rc Opt: cdma_preferred_forward_and_reverse_rc	The Preferred Forward & Reverse RC value, this takes the form "forward_rc,reverse_rc" Format: forward radio channel, reverse radio channel Default: 0,0
Web: SID-NID pairs UCI: mobile.main.cdma_sid_nid_pairs Opt: cdma_sid_nid_pairs	Allows specification of SID:NID pairs, this takes the form "SID1,NID1,SID2,NID2, ⁞⁞ Format: SID1 (0-65535),NID (0-65535) Default: 0,65535

24.5. Mobile Manager: Callers

The mobile manager CDMA page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Name

UCI: mobile.@caller[0].name

Opt: name</td><td>Name assigned to the caller.

<table><tr><td>Default</td><td>Blank</td></tr><tr><td>Range</td><td>No limit</td></tr></table></td></tr>
<tr><td>Web: Number

UCI: mobile.@caller[0].number

Opt: number</td><td>Number of the caller allowed to SMS the router. Add in specific caller numbers, or use the * wildcard symbol.

<table><tr><td>Default</td><td>Blank</td></tr><tr><td>Range</td><td>No limit</td></tr><tr><td>Characters</td><td>Global value (*) is accepted.

International value (+) is accepted</td></tr></table></td></tr>
<tr><td>Web: Enable

UCI: mobile.@caller[0].enabled

Opt: enabled</td><td>Enables or disables incoming caller ID.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Respond

UCI: mobile.@caller[0].respond

Opt: respond</td><td>If checked, the router will return an SMS. Select Respond if you want the router to reply.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
</table>

24.6. Roaming Interface Template: Web Interface

The roaming interface template page

Web Field/UCI/Package Option	Description
Web: Interface Signal Sort UCI: mobile.@roaming_template[0].sort_sig_ strength Opt: sort_sig_strength	Sorts interfaces by signal strength priority, so those that have a better signal strength will be tried first. 0 Disabled 1 Enabled
Web: Roaming SIM UCI: mobile.main.roaming_sim Opt: roaming_sim	Sets in which slot to insert roaming SIM card. 1 SIM slot 1 2 SIM slot 2
Web: Firewall Zone UCI: mobile.@roaming_template[0].firewall_ zone Opt: firewall_zone	Adds all generated interfaces to this zone. Select existing zone or click **unspecified** or **create** to create new zone.
Web: APN UCI: mobile.@roaming_template[0].apn Opt: apn	APN name of Mobile Network Operator.
Web: PIN UCI: mobile.@roaming_template[0].pincode Opt: pincode	SIM card's PIN number.
Web: PAP/CHAP username UCI: mobile.@roaming_template[0].username Opt: username	Username used to connect to APN.
Web: PAP/CHAP password UCI: mobile.@roaming_template[0].password Opt: password	Password used to connect to APN.
Web: Service Order UCI: mobile.@roaming_template[0].service_or der Opt: service_order	Defines a space separated list of services, in preferred order. Valid options are gprs, umts, lte, auto. If no valid_service order is defined, then the configured Service Type is used. Example: mobile.@roaming_template[0].service_order="gprs umts lte auto" Default: Blank. Automatically detect best service. Range: gprs umts lte auto
Web: Health Monitor Interval UCI:	Sets the period, in seconds, to check the health status of the interface. The Health Monitor interval will be used for: • Interface state checks • Ping interval • Signal strength checks Default: 10. Health checks every 10 seconds. Range:
Web: Health Monitor ICMP Host(s) UCI: mobile.@roaming_template[0].icmp_host s	Specifies target IP address for ICMP packets. Web: Description: UCI

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Opt: icmp_hosts</td><td>
<table>
<tr><td>Disable</td><td>Disables the option.</td><td>disable</td></tr>
<tr><td>DNS servers</td><td>DNS IP addresses will be used.</td><td>dns</td></tr>
<tr><td>WAN gateway</td><td>Gateway IP address will be used.</td><td>gateway</td></tr>
<tr><td>custom</td><td>Ability to provide IP address. Multiple pings targets can be entered, comma separated. Pings to both must fail for health check to fail. Example:

option icmp_hosts '1.1.1.1,2.2.2.2</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Health Monitor Conntrack Test Host(s)

UCI: mobile.@roaming_template[0].conntrack_ hosts

Opt: conntrack_hosts</td><td>
Conntrack is the feature used to track if there is any traffic to and from an IP destination within the health interval.

The Conntrack_hosts option defines the IP for conntrack to track, usually the icmp_host IP is used.

If traffic to the conntrack_hosts IP is detected then multiwan does not send a ping health check to the icmp_host; otherwise a ping is sent as normal to the icmp_host.

By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated.

Conntrack is generally used to limit the traffic sent on a GSM network.
<table>
<tr><th>Web</th><th>Description</th><th>UCI</th></tr>
<tr><td>Default</td><td>Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.</td><td></td></tr>
<tr><td>Disable</td><td>Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.</td><td>disable</td></tr>
<tr><td>custom</td><td>Specifies an IP other than the icmp_host for conntrack to track.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Health Monitor ICMP Timeout

UCI: mobile.@roaming_template[0].timeout

Opt: timeout</td><td>
Specifies the time in seconds that Health Monitor ICMP will timeout at.

Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.
<table>
<tr><td>Default</td><td>3. Wait 3 seconds for ping reply.</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Health Monitor ICMP Interval

UCI: mobile.@roaming_template[0].interval

Opt: icmp_interval</td><td>
Defines the interval, in seconds, between multiple pings sent at each health check.
<table>
<tr><td>Default</td><td>1</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Attempts Before WAN Failover

UCI:

mobile.@roaming_template[1].health_fail

_retries

Opt: health_fail_retries</td><td>
Defines the number of health check failures before interface is disconnected.
<table>
<tr><td>Default</td><td>3</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Attempts Before WAN Recovery

UCI: mobile.@roaming_template[0].health_recovery_retries

Opt: health_recovery_retries</td><td>
Sets the number of health check passes before the interface is considered healthy. This field is not used for a roaming template.
<table>
<tr><td>Default</td><td>5</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
</table>

Web Field/UCI/Package Option	Description
Web: Priority UCI: mobile.@roaming_template[0].priority Opt: priority	Type the priority number. The higher the value, the higher the priority. This multi-WAN interface priority must be lower than the one specified in the priority field for the PMP interface. Default: 0 Range:
Web: Multi-WAN: Exclusive Group UCI: mobile.@roaming_template[0].multiwan_exclusive_group Opt: multiwan_exclusive_group	Specifies the Multi-WAN group for the generated roaming interfaces. Defaults to '3g' if not specified.
Web: Minimum ifup interval UCI: multiwan.wan.ifup_retry_sec Opt: ifup_retry_sec	Not used for a roaming interface. Default: 300. Retry primary interface every 300 seconds. Range:
Web: Interface Start Timeout UCI: mobile.@roaming_template[0].ifup_timeo ut_sec Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail. Default: 40 1: Enabled
Web: Signal Threshold (dBm) UCI: mobile.@roaming_template[0].signal_threshold Opt: signal_threshold	Specifies the minimum RSCP signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for rscp_dbm in mobile diagnostics. Default: -115dBm Range: -46 to -115 dBm

When you have configured your settings, click **Save & Apply**.

The reboot page

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

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

24.7. 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.

24.7.1. 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 Pre-empt is not selected.

The Multi-WAN page, pre-empt not selected

Click **Save & Apply**.

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

The system reboot page

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

24.8. 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.

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 disconnect the failed interface and attempt to connect to the next best roaming interface.

24.8.1. 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 five sections in the mobile manager page:

Section	Description
Basic settings	Enable SMS, configure SIM pin code, select roaming SIM, collect ICCCIDs and set IMSI.
Advanced	Configure advanced options such as collect ICCIDs and temperature polling interval.
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.	

24.8.2. Basic Settings

Web Field/UCI/Package Option	Description
Web: SMS Enable UCI: mobile.main.sms Opt: sms	Enables SMS. Default: yes – Enabled no – Disabled
Web: Collect ICCIDs UCI: mobile.main.init_get_iccids Opt: init_get_iccids	Enables or disables integrated circuit card identifier ICCID's collection functionality. If enabled then both SIM 1 and SIM 2 ICCIDs will be collected otherwise it will default to SIM 1. This will be display under mobile stats. Default: yes – Enabled no – Disabled
Web: PIN code for SIM1 UCI: mobile.main.sim2pin Opt: sim2pin	Depending on the SIM card specify the pin code for SIM 1. Default – Blank Range –
Web: PIN code for SIM2 UCI: mobile.main.sim2pin Opt: sim2pin	Depending on the SIM card specify the pin code for SIM 2. Default – Blank Range –
Web: HDR Auto User ID UCI: mobile.main.hdr_userid Opt: hdr_userid	AN-PPP user ID. Supported on Cellient (CDMA) modem only. Default – Blank Range –

24.8.3. Caller Settings

Web Field/UCI/Package Option	Description
Web: Name UCI: mobile.@caller[0].name Opt: name	Name assigned to the caller. Default: Blank Range:
Web: Number UCI: mobile.main.init_get_iccids Opt: init_get_iccids	Number of the caller allowed to SMS the router. Add in specific caller numbers or use the wildcard symbol. Default: Blank Range:
Web: Enable UCI: mobile.main.sim2pin Opt: sim2pin	Enables or disables incoming caller ID. Default: no — Disabled yes — Enabled
Web: Respond UCI: mobile.@caller[0].respond Opt: respond	If checked, the router will return an SMS. Select Respond if you want the router to reply. Default: 0 — Disabled 1 — Enabled

24.8.4. 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
```

24.8.5. Roaming Interface Template: Web Interface

The roaming interface template page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Interface Signal Sort UCI:

mobile.@roaming_template[0].sort_sig_

strength

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

UCI: mobile.main.roaming_sim

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 UCI:

mobile.@roaming_template[0].firewall_

zone

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: APN

UCI: mobile.@roaming_template[0].apn

Opt: apn</td><td>APN name of Mobile Network Operator.</td></tr>
<tr><td>Web: PIN

UCI: mobile.@roaming_template[0].pincode

Opt: pincode</td><td>SIM card's PIN number.</td></tr>
<tr><td>Web: PAP/CHAP username

UCI: mobile.@roaming_template[0].username

Opt: username</td><td>Username used to connect to APN.</td></tr>
<tr><td>Web: PAP/CHAP password

UCI: mobile.@roaming_template[0].password

Opt: password</td><td>Password used to connect to APN.</td></tr>
<tr><td>Web: Service Order

UCI:

mobile.@roaming_template[0].service_or

der

Opt: service_order</td><td>Defines a space separated list of services, in preferred order. Valid options are gprs, umts, lte, auto.

If no valid_service order is defined, then the configured Service Type is used. Example:

mobile.@roaming_template[0].service_order=”gprs umts lte auto”

<table><tr><td>Default</td><td>Blank. Automatically detect best service.</td></tr><tr><td>Range</td><td>gprs umts lte auto</td></tr></table></td></tr>
<tr><td>Web: Health Monitor Interval

UCI:</td><td>Sets the period, in seconds, to check the health status of the interface.

The Health Monitor interval will be used for:

• Interface state checks
• Ping interval
• Signal strength checks

<table><tr><td>Default</td><td>10. Health checks every 10 seconds.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Health Monitor ICMP Host(s)

UCI: mobile.@roaming_template[0].icmp_host s</td><td>Specifies target IP address for ICMP packets.

<table><tr><td>Web</td><td>Description</td><td>UCI</td></tr></table></td></tr>
</table>

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Opt: icmp_hosts</td><td>
<table>
<tr><td>Disable</td><td>Disables the option.</td><td>disable</td></tr>
<tr><td>DNS servers</td><td>DNS IP addresses will be used.</td><td>dns</td></tr>
<tr><td>WAN gateway</td><td>Gateway IP address will be used.</td><td>gateway</td></tr>
<tr><td>custom</td><td>Ability to provide IP address. Multiple pings targets can be entered, comma separated. Pings to both must fail for health check to fail. Example:

option icmp_hosts '1.1.1.1,2.2.2.2</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Health Monitor Conntrack Test Host(s)

UCI: mobile.@roaming_template[0].conntrack_ hosts

Opt: conntrack_hosts</td><td>
Conntrack is the feature used to track if there is any traffic to and from an IP destination within the health interval.

The Conntrack_hosts option defines the IP for conntrack to track, usually the icmp_host IP is used.

If traffic to the conntrack_hosts IP is detected then multiwan does not send a ping health check to the icmp_host; otherwise a ping is sent as normal to the icmp_host.

By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated.

Conntrack is generally used to limit the traffic sent on a GSM network.
<table>
<tr><th>Web</th><th>Description</th><th>UCI</th></tr>
<tr><td>Default</td><td>Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.</td><td></td></tr>
<tr><td>Disable</td><td>Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.</td><td>disable</td></tr>
<tr><td>custom</td><td>Specifies an IP other than the icmp_host for conntrack to track.</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Health Monitor ICMP Timeout

UCI: mobile.@roaming_template[0].timeout

Opt: timeout</td><td>
Specifies the time in seconds that Health Monitor ICMP will timeout at.

Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.
<table>
<tr><td>Default</td><td>3. Wait 3 seconds for ping reply.</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Health Monitor ICMP Interval

UCI: mobile.@roaming_template[0].interval

Opt: icmp_interval</td><td>
Defines the interval, in seconds, between multiple pings sent at each health check.
<table>
<tr><td>Default</td><td>1</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Attempts Before WAN Failover

UCI:

mobile.@roaming_template[1].health_fail

_retries

Opt: health_fail_retries</td><td>
Defines the number of health check failures before interface is disconnected.
<table>
<tr><td>Default</td><td>3</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
<tr><td>Web: Attempts Before WAN Recovery

UCI: mobile.@roaming_template[0].health_recovery_retries

Opt: health_recovery_retries</td><td>
Sets the number of health check passes before the interface is considered healthy. This field is not used for a roaming template.
<table>
<tr><td>Default</td><td>5</td></tr>
<tr><td>Range</td><td></td></tr>
</table>
</td></tr>
</table>

Web Field/UCI/Package Option	Description
Web: Priority UCI: mobile.@roaming_template[0].priority Opt: priority	Type the priority number. The higher the value, the higher the priority. This multi-WAN interface priority must be lower than the one specified in the priority field for the PMP interface. Default: 0 Range:
Web: Multi-WAN: Exclusive Group UCI: mobile.@roaming_template[0].multiwan_exclusive_group Opt: multiwan_exclusive_group	Specifies the Multi-WAN group for the generated roaming interfaces. Defaults to '3g' if not specified.
Web: Minimum ifup interval UCI: multiwan.wan.ifup_retry_sec Opt: ifup_retry_sec	Not used for a roaming interface. Default: 300. Retry primary interface every 300 seconds. Range:
Web: Interface Start Timeout UCI: mobile.@roaming_template[0].ifup_timeo ut_sec Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail. Default: 40 1: Enabled
Web: Signal Threshold (dBm) UCI: mobile.@roaming_template[0].signal_threshold Opt: signal_threshold	Specifies the minimum RSCP signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for rscp_dbm in mobile diagnostics. Default: -115dBm Range: -46 to -115 dBm

When you have configured your settings, click **Save & Apply**.

The reboot page

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

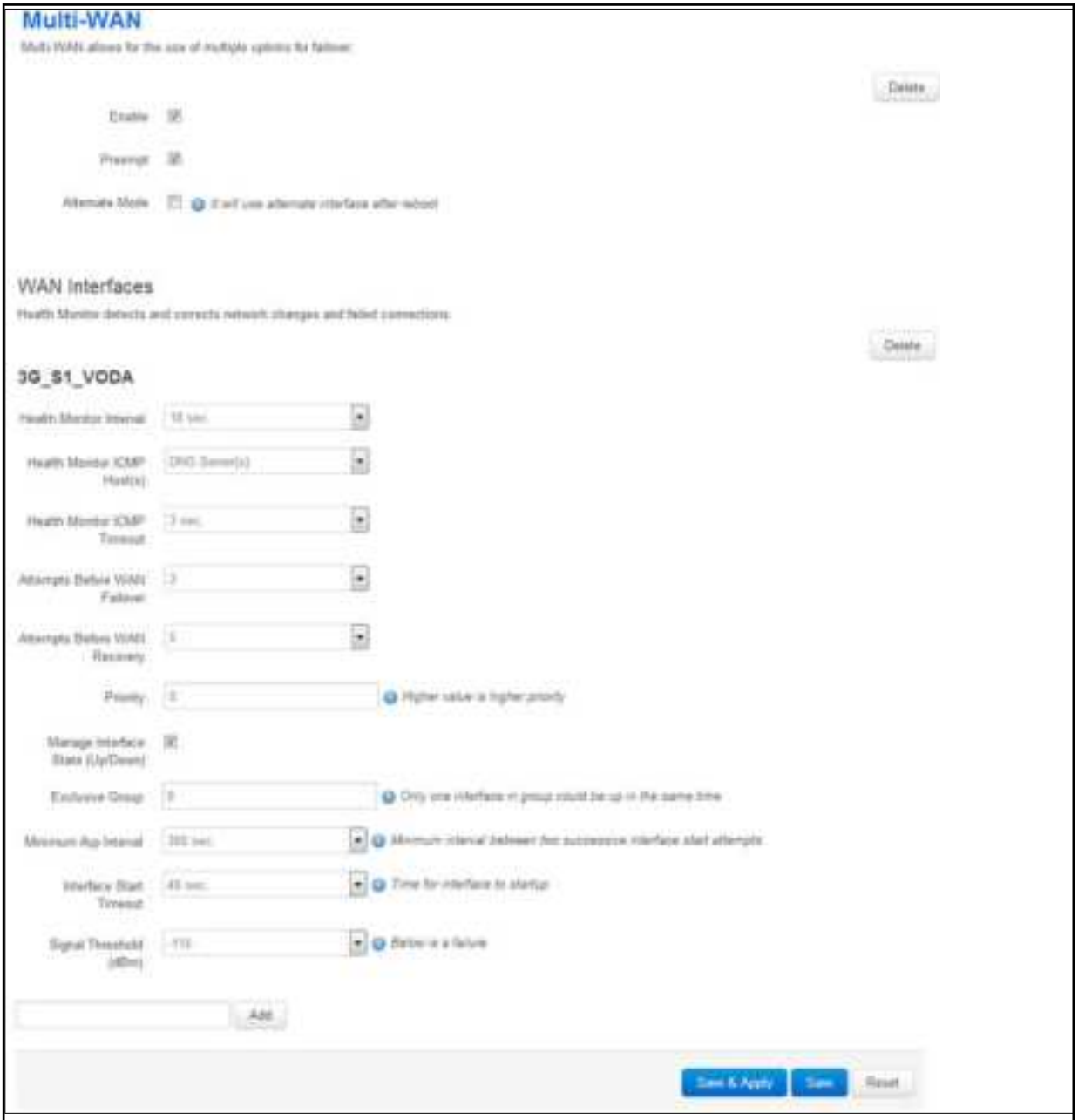
Check the **Reboot now** check box and then click **Reboot**.

24.8.6. Set Multi-WAN Options For Primary Predefined Interface

On the web interface go to **Network -> Multi-Wan**. The Multi-WAN page appears.

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.



The Multi-WAN page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enable

UCI: multiwan.config.enabled

Opt: enabled</td><td>Enables multiwan.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Preempt

UCI: multiwan.config.preempt

Opt: preempt</td><td>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.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Alternate Mode

UCI: multiwan.config.alt

Opt: alt</td><td>Enables or disables alternate mode for multiwan. If enabled, the router will use an alternate interface after reboot.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: WAN Interfaces

UCI: multiwan.3g_s<sim-

number>_<short-operator-name>

Opt: 3g_s<sim-number>_<short- operator-name></td><td>Provide the same interface name as chosen in multiwan section below and click Add.</td></tr>
<tr><td>Web: Health Monitor Interval

UCI: multiwan.[..x..].health_interval

Opt: health_interval</td><td>Sets the period to check the health status of the interface. The Health Monitor interval will be used for:

• Interface state checks
• Ping interval
• Signal strength checks</td></tr>
<tr><td>Web: Health Monitor ICMP Host(s)

UCI: multiwan.[..x..].icmp_hosts

Opt: icmp_hosts</td><td>Specifies the 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 Conntrack Test Host(s)

UCI: multiwan.wan.conntrack_hosts

Opt: conntrack_hosts</td><td>Conntrack is the feature used to track if there is any traffic to and from an IP destination within the health interval.

Conntrack_hosts option defines the IP for conntrack to track – usually the icmp_host IP is used.

If traffic to the conntrack_hosts IP is detected then multiwan does not send a ping health check to the icmp_host otherwise a ping is sent as normal to the icmp_host.

By default the conntrack_hosts is checked if the health interval is greater than 5 minutes. This time threshold currently cannot be manipulated.

Conntrack is generally used to limit the traffic sent on a GSM network.

<table><tr><td>Default</td><td>Conntrack checks for traffic from icmp_host IP when health_interval is greater than 5 minutes.</td></tr><tr><td>Disable</td><td>Conntrack is Disabled.</td></tr><tr><td>Custom</td><td>Specifies an IP other than the icmp_host for conntrack to track.</td></tr></table></td></tr>
<tr><td>Web: Health Monitor ICMP Timeout

UCI: multiwan.[..x..].timeout

Opt: timeout</td><td>Sets ping timeout in seconds. Choose the time in seconds that the health monitor ICMP will timeout at.

<table><tr><td>3</td><td>Wait 3 seconds for ping reply.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

Web Field/UCI/Package Option	Description
Web: Health Monitor ICMP Interval UCI: multiwan.wan.icmp_interval Opt: icmp_interval	Defines the interval between multiple pings sent at each health check. 1 | Range |
Web: Health Monitor ICMP Count UCI: multiwan.wan.icmp_count Opt: icmp_count	Defines the number of pings to send at each health check. 1 | Range | Enabled
Web: Attempts Before WAN Failover UCI: multiwan. [..x..].health_fail_retries Opt: health_fail_retries	Sets the amount of health monitor retries before the interface is considered a failure. 3 | Range |
Web: Attempts Before WAN Recovery UCI: multiwan. [..x..].health_recovery_retries Opt: health_recovery_retries	Sets the number of health monitor checks before the interface is considered healthy. Only relevent if pre-empt mode is enabled. 5 | Range |
Web: Priority UCI: multiwan.[..x..].priority Opt: priority	Specifies the priority of the interface. The higher the value, the higher the priority. This multiwan interface priority must be higher than the one specified in the priority field in the 'Roaming Interface Template' page described in the following section. 0 | Range |
Web: Exclusive Group UCI: multiwan.[..x..].exclusive_group Opt: exclusive_group	Defines the group to which the interface belongs; only one interface can be active. 0 | Range |
Web: Manage Interface State (Up/Down) UCI: multiwan.[..x..].manage_state Opt: manage_state	Defines whether multiwan will start and stop the interface. Select **Enabled**. 0 | Disabled 1 | Enabled
Web: Minimum ifup Interval UCI: multiwan.[..x..].ifup_retry_sec Opt: ifup_retry_sec	Specifies the interval in seconds before retrying the primary interface when pre-empt mode is enabled.
Web: Interface Start Timeout UCI: multiwan.[..x..].ifup_timeout Opt: ifup_timeout	Specifies the time in seconds for interface to start up. If it is not up after this period, it will be considered a fail. Choose timer greater than 120 seconds. 40 | 40 Seconds Range |
Web: Signal Threshold (dBm) UCI: multiwan.[..x..].signal_threshold Opt: signal_threshold	Specifies the minimum signal strength in dBm before considering if the interface fails signal health check. Uses the value stored for sig_dbm in mobile diagnostics. -115 | Disabled Range | -46 to -115dBm
Web: RSCP Threshold (dBm) UCI: multiwan.[..x..].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.

Web Field/UCI/Package Option	Description
Opt: rscp_threshold	
Web: ECIO Threshold (dB) UCI: multiwan.[..x..].ecio_threshold Opt: ecio_threshold	Specifies the minimum ECIO signal strength in dB before considering if the interface fails signal health check. Uses the value stored for ecio_db in mobile diagnostics.
Web: Signal Test UCI: multiwan.[..x..].signal_test Opt: signal_test	Defines script to test various signal characteristics in multiwan signal test. For example: option signal_test '(tech == 0) then (sig_dbm > -70) else (rscp_dbm > -105 and ecio_db > -15)' This states that when technology is GSM a health fail is determined when signal strength is less than -70dBm. When technology is not GSM a health fail occurs when either rscp_dbm falls below -105dBm or ecio_db falls below -15dB. Tech values are: 0 GSM 1 GSM Compact 2 UTRAN 3 GSM w/EGPRS 4 UTRAN w/HSPDA 5 UTRAN w/HSUPA 6 UTRAN w/HSUPA and HSDPA 7 E-UTRAN

Click **Save**.

24.9. Configuring Automatic Operator Selection Via UCI

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.

24.9.1. PMP + Roaming: Pre-Empt & Disabled Using UCI

PMP interface configuration

The PMP interface is configured in the network package /etc/config/network. To view the network configuration file, enter:

To view the mobile 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_sl_voda'
option auto '0'
option proto '3g'
option service_order 'auto lte umts gprs'
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_order='auto lte umts gprs'
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
```

Roaming Interface Configuration Using UCI

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
```

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

24.9.2. 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:

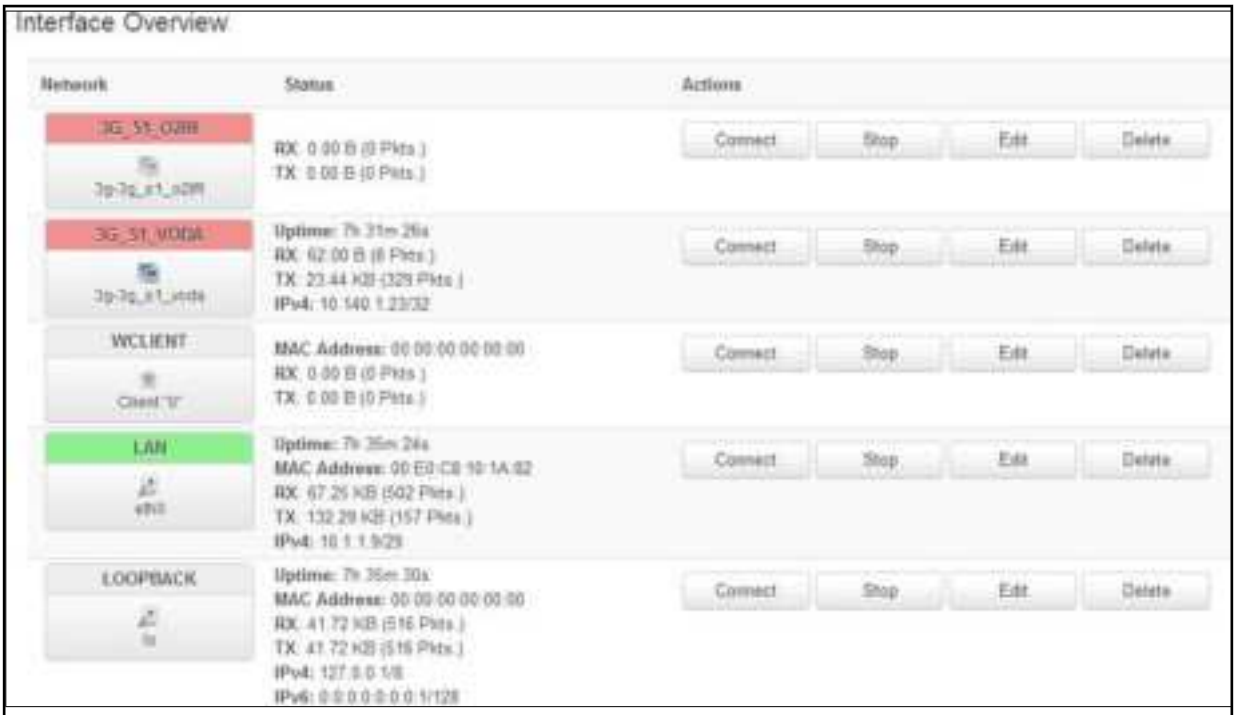
```
root@VA_router:~# uci show multiwan
multiwan.config=multiwan
multiwan.config.enabled=yes
multiwan.config.preempt=no
multiwan.config.alt_mode=no
```

24.9.3. Automatic Operator Selection Diagnostics Via The Web Interface

When interfaces are auto-created they are presented in the network and in the multiwan package.

To check interfaces created in the multiwan package, from the top menu, select **Network -> Multi-WAN**.

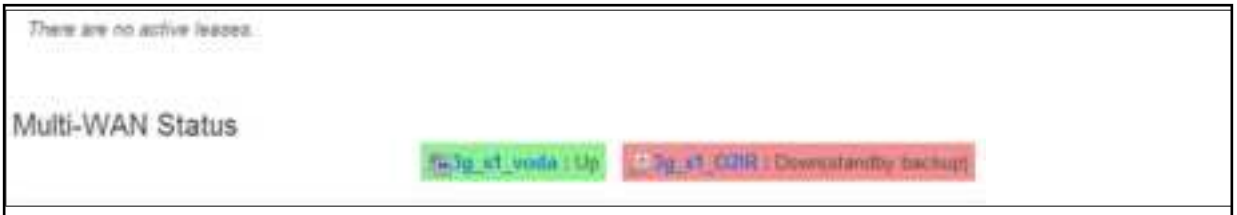
To check interfaces that have been created in the network package, from the top menu, select **Network -> Interfaces**.



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.



The multi-WAN status page

24.10. Automatic Operator Selection Diagnostics Via UCI

In this section, you can use UCI to check status of packages and various interfaces associated with automatic operator selection.

24.10.1. Check Roaming Interfaces Discovered

Roaming interfaces discovered during roaming search are stored at: **/var/const_state/roaming**.

This file contains a section for each discovered operator/service combination, along with signal strength, if tested. Time taken to scan is also available along with the time of scan and number of services found.

To check roaming interfaces discovered, enter:

```
root@VA_router:~# cat /var/const_state/roaming
roaming.main2_voda_lte=service
roaming.main2_voda_lte.name=vodafone IE
roaming.main2_voda_lte.shortname=voda IE
roaming.main2_voda_lte.opnum=27201
roaming.main2_voda_lte.interface=main2_voda
roaming.main2_voda_lte.servicetype=7
roaming.main2_voda_lte.sim=2
roaming.main2_voda_lte.tested=0
roaming.main2_voda_lte.signalstrength=0
roaming.main2_voda_umts=service
roaming.main2_voda_umts.name=vodafone IE
roaming.main2_voda_umts.shortname=voda IE
roaming.main2_voda_umts.opnum=27201
roaming.main2_voda_umts.interface=main2_voda
roaming.main2_voda_umts.servicetype=2
roaming.main2_voda_umts.sim=2
roaming.main2_voda_umts.tested=1
roaming.main2_voda_umts.signalstrength=-79
roaming.main2_voda_gprs=service
roaming.main2_voda_gprs.name=vodafone IE
roaming.main2_voda_gprs.shortname=voda IE
```

```
roaming.main2_voda_gprs.opnum=27201
roaming.main2_voda_gprs.interface=main2_voda
roaming.main2_voda_gprs.servicetype=0
roaming.main2_voda_gprs.sim=2
roaming.main2_voda_gprs.tested=0
roaming.main2_voda_gprs.signalstrength=0
roaming.main2_o2IR_umts=service
roaming.main2_o2IR_umts.name=o2 IRL
roaming.main2_o2IR_umts.shortname=o2 - IRL
roaming.main2_o2IR_umts.opnum=27202
roaming.main2_o2IR_umts.interface=main2_o2IR
roaming.main2_o2IR_umts.servicetype=2
roaming.main2_o2IR_umts.sim=2
roaming.main2_o2IR_umts.tested=1
roaming.main2_o2IR_umts.signalstrength=-85
roaming.main2_o2IR_gprs=service
roaming.main2_o2IR_gprs.name=o2 IRL
roaming.main2_o2IR_gprs.shortname=o2 - IRL
roaming.main2_o2IR_gprs.opnum=27202
roaming.main2_o2IR_gprs.interface=main2_o2IR
roaming.main2_o2IR_gprs.servicetype=0
roaming.main2_o2IR_gprs.sim=2
roaming.main2_o2IR_gprs.tested=0
roaming.main2_o2IR_gprs.signalstrength=0
roaming.status=status
roaming.status.num_services=5
roaming.status.scan_update_time=Thu May 12 05:02:38 2022
roaming.status.scan_duration=185
```

Roaming operators are also stored in MIB `vaModemRoaming.mib`.

24.10.2. Check Interfaces Created In Multiwan Package

To check interfaces created in the multiwan package, enter:

```
root@VA_router:~# cat/var/const_state/multiwan
multiwan.main2_3IRL=interface
multiwan.main2_3IRL.timeout=disable
multiwan.main2_3IRL.health_recovery_retries=5
multiwan.main2_3IRL.exclusive_group=3g
multiwan.main2_3IRL.manage_state=yes
multiwan.main2_3IRL.signal_threshold=-80
multiwan.main2_3IRL.ifup_timeout_sec=150
multiwan.main2_3IRL.icmp_hosts=disable
multiwan.main2_3IRL.health_interval=4
multiwan.main2_3IRL.priority=5
multiwan.main2_3IRL.ifup_retry_sec=120
multiwan.main2_3IRL.health_fail_retries=3
multiwan.main2_o2IR=interface
multiwan.main2_o2IR.timeout=disable
multiwan.main2_o2IR.health_recovery_retries=5
multiwan.main2_o2IR.exclusive_group=3g
multiwan.main2_o2IR.manage_state=yes
multiwan.main2_o2IR.signal_threshold=-80
multiwan.main2_o2IR.ifup_timeout_sec=150
multiwan.main2_o2IR.icmp_hosts=disable
multiwan.main2_o2IR.health_interval=4
multiwan.main2_o2IR.priority=5
multiwan.main2_o2IR.ifup_retry_sec=120
multiwan.main2_o2IR.health_fail_retries=3
```

24.10.3. Check Interfaces Created In Network Package

To check interfaces created in the network package, enter:

```
root@VA_router:~# cat /var/const_state/network
network.main2_3IRL=interface
network.main2_3IRL.snmp_alias_ifindex=3
network.main2_3IRL.sim=2
network.main2_3IRL.defaultroute=yes
network.main2_3IRL.username=campen1
network.main2_3IRL.apn=vpn.amylan.co.uk
network.main2_3IRL.opformat=2
network.main2_3IRL.phy=1-1
network.main2_3IRL.roaming_sim=2
network.main2_3IRL.operator=27205
network.main2_3IRL.password=campen1
network.main2_3IRL.auto=no
network.main2_3IRL.service_order=auto
network.main2_3IRL.proto=3g
network.main2_o2IR=interface
network.main2_o2IR.snmp_alias_ifindex=3
network.main2_o2IR.sim=2
network.main2_o2IR.defaultroute=yes
network.main2_o2IR.username=campen1
network.main2_o2IR.apn=vpn.amylan.co.uk
network.main2_o2IR.opformat=2
network.main2_o2IR.phy=1-1
network.main2_o2IR.roaming_sim=2
network.main2_o2IR.operator=27202
network.main2_o2IR.password=campen1
network.main2_o2IR.auto=no
network.main2_o2IR.service_order=auto
network.main2_o2IR.proto=3g
```

24.10.4. Check Current Interface

To check the SIM status of the interface you are currently using, enter:

```
root@VA_router:~# cat /var/const_state/mobile
mobile.3g_1_1=status
mobile.3g_1_1.sim2_iccid=89314404000075920976
mobile.3g_1_1.imei=866802020194140
mobile.3g_1_1.hw_rev=4534B04SIM7100E
mobile.3g_1_1.sim_select=yes
```

To check mobile status of the interface you are currently using, enter

```
root@VA_router:~# cat /var/state/mobile
mobile.3g_1_1=status
mobile.3g_1_1.auto_info=/tmp/3g_1-1.auto
mobile.3g_1_1.scan_update_time=Thu May 12 05:02:38 2018
mobile.3g_1_1.imsi=204043726930595
mobile.3g_1_1.imsi2=204043726930595
mobile.3g_1_1.lte_band=3
mobile.3g_1_1.last_error=no network service
mobile.3g_1_1.mcc=272 mobile.3g_1_1.last_error_time=2022-02-22 10:41:27
mobile.3g_1_1.lac=11
mobile.3g_1_1.cell=46542698
mobile.3g_1_1.mnc=05
mobile.3g_1_1.operator_code=27205
mobile.3g_1_1.operator_name=3 IRL DATA ONLY
mobile.3g_1_1.rscp_dbm=-86
mobile.3g_1_1.ecio_db=-8.5
mobile.3g_1_1.sig_dbm=-51
mobile.3g_1_1.temperature=37
mobile.3g_1_1.vam_state=connecting
mobile.3g_1_1.sim_slot=2
mobile.3g_1_1.sim_in=yes
mobile.3g_1_1.technology=UMTS
mobile.3g_1_1.registered=Roaming
mobile.3g_1_1.reg_code=5
mobile.3g_1_1.registered_pkt=Searching
mobile.3g_1_1.reg_code_pkt=2
```

25. 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. Recovery actions are prioritised based on their configured failure periods, the smallest failure period having the lowest priority. Lowest priority actions are repeated until the next highest priority action executes at which point it then stops leaving only the new action to execute at configured intervals.

Example:

- Failure time 1 = 1 hour; Failure action 1= interface up
- Failure time 2 = 10 hours; Failure action 2 = interface restart
- Failure time 3 = 24 hours; Failure action 3 = reboot

In the above example action execution priorities are action 3 > action 2 > action 1. In the case of failure to detect incoming packets, action 1 is triggered first and is executed at intervals of one hour until action 2 is due. When action 2 is executed, action 1 gets disabled and thereafter only action 2 is executed every 10 hours until action 3 is due.

If the status of the interface is detected as 'up' at any stage then no subsequent failure action will occur and all failure timers are reset. In the case of any subsequent failure, all failure actions are re-enabled and the action sequence is repeated.

Configuration package used

Package	Sections
cwatch	watch

25.1. Configuring Connection Watch Using The Web Interface

To configure Connection Watch using the web interface, select **Services - >Connection Watch**. The Connection Watch page appears.

The add connection watch configuration page

If no Connection Watch configuration exists in the configuration file, first enter a name for the Connection Watch instance and select **Add**.

The connection watch configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: cwatch.@watch[0].enabled

Opt: enabled</td><td>Enables a cwatch instance.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Interfaces

UCI: cwatch.@watch[0].test_ifaces

Opt: test_ifaces</td><td>Defines the interface name(s) to monitor. Multiple interfaces are delimited by space separator. Example:

option test_ifaces 'WANADSL WANMOBILE'

If multiple interfaces are defined the failure action will only be triggered if no traffic is received on all interfaces for the defined period.</td></tr>
<tr><td>Web: Failure Time for Action 1

UCI: cwatch.@watch[0].failure_time_1

Opt: failure_time_1</td><td>Defines a duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days.

<table><tr><td>Default</td><td>1 hour</td></tr><tr><td>Range</td><td>s; m; h; d;</td></tr></table></td></tr>
<tr><td>Web: Failure Action 1

UCI: cwatch.@watch[0].failure_action_1

Opt: failure_action_1</td><td>Defines the failure action associated with failure_time_1. Example to force up interface:

option failure_action_1 'ifup wan'

<table><tr><td>Default</td><td>blank</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Failure Grace Time 1 UCI:

cwatch.@watch[0].failure_grace_time_1

Opt: failure_grace_time_1</td><td>Defines a grace time during which interface activity will be

ignored after 'Failure Action 1' is executed.

Connection Watch will assume the interface to be down during the grace period and will not reset the failure action timers even if packets are received during this grace time.

This can be used to overcome the situation where packets can be received after a failure action even though the interface eventually fails to connect.

For example, during a USB restart on a mobile interface, a small amount of packets can be registered as being received while a mobile connection is attempted but fails registration.

<table><tr><td>Default: 0</td><td>No grace time</td></tr><tr><td>Range</td><td>s; m; h; d;</td></tr></table></td></tr>
<tr><td>Web: Failure Time for Action 2

UCI: cwatch.@watch[0].failure_time_2

Opt: failure_time_2</td><td>Defines a second duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days.

<table><tr><td>Default</td><td>10 hours</td></tr><tr><td>Range</td><td>s; m; h; d;</td></tr></table></td></tr>
<tr><td>Web: Failure Action 2

UCI: cwatch.@watch[0].failure_action_2

Opt: failure_action_2</td><td>Defines the failure action associated with failure_time_2. Example to reset usb:

option failure_action_1 '/etc/init.d/usb_startup restart'

<table><tr><td>Default</td><td>Blank</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Failure Grace Time 2 UCI:

cwatch.@watch[0].failure_grace_time_2

Opt: failure_grace_time_2</td><td>Defines a grace time during which interface activity will be

ignored after 'Failure Action 2' is executed.

Connection Watch will assume the interface to be down during the grace period and will not reset the failure action timers even if packets are received during this grace time.</td></tr>
</table>

Web Field/UCI/Package Option	Description
	This can be used to overcome the situation where packets can be received after a failure action even though the interface eventually fails to connect. For example, during a USB restart on a mobile interface, a small amount of packets can be registered as being received while a mobile connection is attempted but fails registration. Default: 0 — No grace time Range — s; m; h; d;
Web: Failure Time for Action 3 UCI: cwatch.@watch[0].failure_time_3 Opt: failure_time_3	Defines a third duration to monitor an interface for receive traffic. Duration can be specified in seconds, minutes, hours, days. Default — 24 hours Range — s; m; h; d;
Web: Failure Action 3 UCI: cwatch.@watch[0].failure_action_3 Opt: failure_action_3	Defines the failure action associated with failure_time_3. Example to reset usb: option failure_action_3 'reboot' Default — Blank Range —
Web: Failure Grace Time 3 UCI: cwatch.@watch[0].failure_grace_time_3 Opt: failure_grace_time_3	Defines a grace time during which interface activity will be ignored after 'Failure Action 3' is executed. Connection Watch will assume the interface to be down during the grace period and will not reset the failure action timers even if packets are received during this grace time. This can be used to overcome the situation where packets can be received after a failure action even though the interface eventually fails to connect. For example, during a USB restart on a mobile interface, a small amount of packets can be registered as being received while a mobile connection is attempted but fails registration. Default: 0 — No grace time Range — s; m; h; d;

25.2. 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'
```

25.2.1. 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
```

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_grace_time_1 `30s`
option failure_time_2 '10h'
option failure_action_2 '/etc/init.d/usb_startup restart'
option failure_grace_time_2 `2m`
option failure_time_3 '24h'
option failure_action_3 'reboot'
```

25.3. Cwatch Diagnostics

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 grace time [x]
```

```
cwatch[x]: Watch WATCH_MOBILE executed action 2 grace time [x]
cwatch[x]: Watch WATCH_MOBILE executed action 3 grace time [x]
```

A syslog message will be generated if there is a problem with the configured cwatch instance.

```
cwatch[x]: Watch WATCH_MOBILE test_ifaces not defined. Watch ignored
```

26. 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.

Configuration Package Used

Package	Sections
dchp	dnsmasq
	dhcp
	host

26.1. Configuring IPv6 Routes Using The Web Interface

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

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

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

26.2. Configuring DHCP And DNS Using The Web Interface

In the top menu, select **Network -> DHCP and DNS**. The DHCP and DNS page appears.

The DHCP and DNS page

There are three sections: Server Settings, Active Leases, and Static Leases.

26.2.1. Dnsmasq: General Settings

Web Field/UCI/Package Option	Description
Web: Domain required UCI: dhcp.@dnsmasq[0].domainneeded Opt: domainneeded	Defines whether to forward DNS requests without a DNS name. Dnsmasq will never forward queries for plain names, without dots or domain parts, to upstream nameservers. If the name is not known from /etc/hosts or DHCP then a "not found" answer is returned. Default: 0 – Disabled 1 – Enabled
Web: Authoritative UCI: dhcp.@dnsmasq[0]. authoritative Opt: authoritative	Forces authoritative mode. This speeds up DHCP leasing. Used if this is the only server in the network. Default: 0 – Disabled 1 – Enabled
Web: Interfaces UCI: dhcp.@dnsmasq[0].interface Opt: list interface	Defines the list of interfaces to be served by dnsmasq. If you do not select a specific interface, dnsmasq will serve on all interfaces. Configured interfaces are shown via the web GUI. Default – Lan. Serve only on LAN interface Range –
Web: Local Server UCI: dhcp.@dnsmasq[0].local Opt: local	Specifies the local domain. Names matching this domain are never forwarded and are resolved from DHCP or host files only. Default – /lan/ Range –
Web: Local Domain UCI: dhcp.@dnsmasq[0].domain Opt: domain	Specifies local domain suffix appended to DHCP names and hosts file entries. Default – lan Range –
Web: Log Queries UCI: dhcp.@dnsmasq[0].logqueries Opt: logqueries	Writes received DNS requests to syslog. Default: 0 – Disabled 1 – Enabled
Web: DNS Forwardings UCI: dhcp.@dnsmasq[0].server Opt: list server	List of DNS servers to forward requests to. To forward specific domain requests only, use // syntax. When using UCI, enter multiple servers with a space between them. Default – No DNS server configured Range –
Web: Rebind Protection UCI: dhcp.@dnsmasq[0].rebind_protection Opt: rebind_protection	Enables DNS rebind attack protection by discarding upstream RFC1918 responses. Default: 0 – Disabled 1 – Enabled
Web: Allow Localhost UCI: dhcp.@dnsmasq[0].rebind_localhost Opt: rebind_localhost	Defines whether to allow upstream responses in the 127.0.0.0/8 range. This is required for DNS-based blacklist services. Only takes effect if rebind protection is enabled. Default: 0 – Disabled 1 – Enabled
Web: Domain Whitelist UCI: dhcp.@dnsmasq[0].rebind_domain Opt: list rebind_domain	Defines the list of domains to allow RFC1918 responses to. Only takes effect if rebind protection is enabled. When using UCI multiple servers, enter the domains with a space between them. Default – No list configured

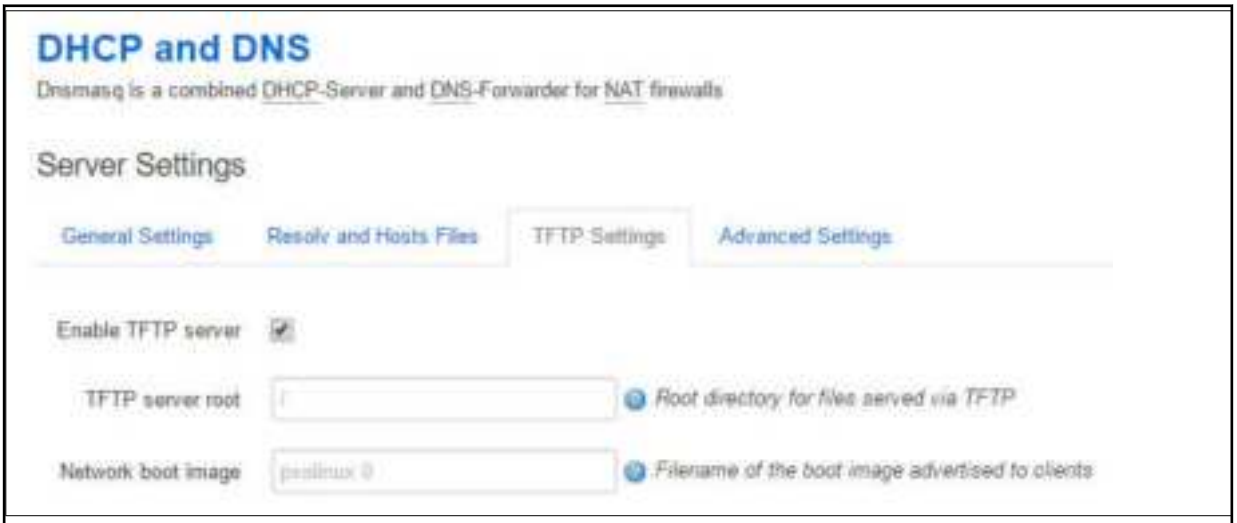
Web Field/UCI/Package Option	Description
	Range

26.2.2. Dnsmasq: Resolv And Host Files

The resolv and host files section

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

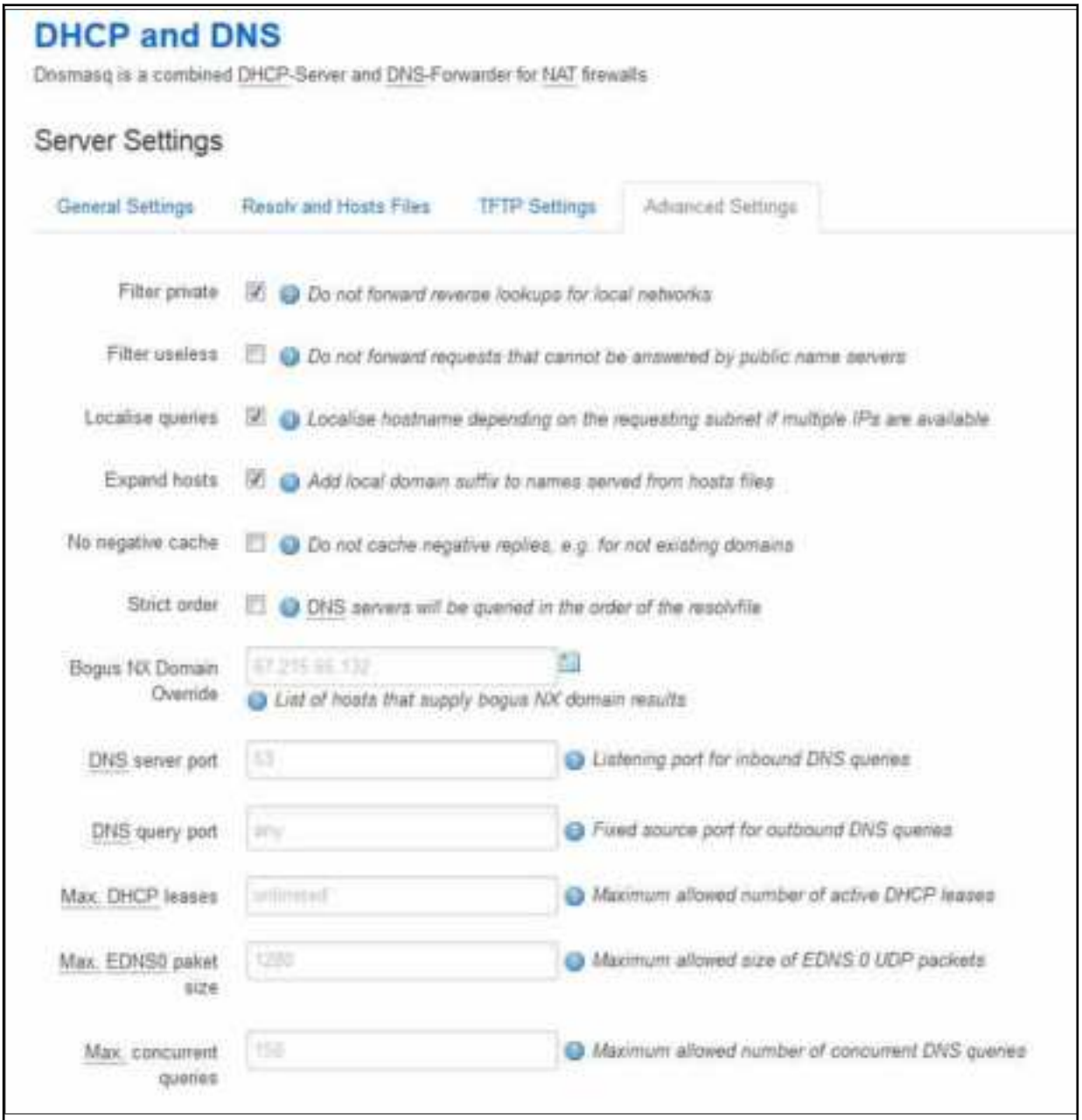
26.2.3. Dnsmasq: TFTP Settings



The TFTP settings section

Web Field/UCI/Package Option	Description
Web: Enable TFTP server UCI: dhcp.@dnsmasq[0].enable_tftp Opt: enable_tftp	Enables the TFTP server. Default: 0 — Disabled 1 — Enabled
Web: TFTP server Root UCI: dhcp.@dnsmasq[0].tftp_root Opt: tftp_root	Defines root directory for file served by TFTP.
Web: Network boot image UCI: dhcp.@dnsmasq[0].dhcp_boot Opt: dhcp_boot	Defines the filename of the boot image advertised to clients. This specifies BOOTP options, in most cases just the file name.

26.2.4. Dnsmasq: Advanced Settings



The advanced settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Filter private

UCI: dhcp.@dnsmasq[0].

Opt: boguspriv</td><td>Enables disallow option for forwarding reverse lookups for local networks. This rejects reverse lookups to private IP ranges where no corresponding entry exists in /etc/hosts.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>Default: 1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Filter useless

UCI: dhcp.@dnsmasq[0].filterwin2k

Opt: filterwin2k</td><td>Enables disallow option for forwarding requests that cannot be answered by public name servers. Normally enabled for dial on demand interfaces.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Localise queries

UCI: dhcp.@dnsmasq[0].localise_queries

Opt: localise_queries</td><td>Defines whether to use an IP address to match the incoming interface if multiple addresses are assigned to a host name in /etc/hosts.</td></tr>
<tr><td>Web: Expand hosts

UCI: dhcp.@dnsmasq[0].expandhosts

Opt: expandhosts</td><td>Adds a local domain suffix to names served from host files.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>Default: 1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: No negative cache

UCI: dhcp.@dnsmasq[0].nonegcache

Opt: nonegcache</td><td>Enable this to stop caching of negative replies. For example, non- existing domains.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Strict order

UCI: dhcp.@dnsmasq[0].strictorder

Opt: strictorder</td><td>Enable this to query DNS servers in the order of the resolve file.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Bogus NX Domain override

UCI: dhcp.@dnsmasq[0].bogusnxdomain

Opt: list bogusnxdomain</td><td>A list of hosts that supply bogus NX domain results. When using UCI multiple servers, enter the server names with a space between them.

<table><tr><td>Default</td><td>Empty list</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: DNS server port

UCI: dhcp.@dnsmasq[0].port

Opt: port</td><td>Listening port for inbound DNS queries.

<table><tr><td>Default: 53</td><td>Set to 0 to disable DNS functionality.</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: DNS query port

UCI: dhcp.@dnsmasq[0].queryport

Opt: queryport</td><td>Defines fixed source port for outbound DNS queries.

<table><tr><td>Default</td><td>Any</td></tr><tr><td>Range</td><td>any; 0-65535</td></tr></table></td></tr>
<tr><td>Web: Max DHCP leases

UCI: dhcp.@dnsmasq[0].dhcpleasemax

Opt: dhcpleasemax</td><td>Defines the maximum allowed number of active DHCP leases.

<table><tr><td>Default</td><td>unlimited</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Max EDNS0 packet size

UCI: dhcp.@dnsmasq[0].ednspacket_max

Opt: ednspacket_max</td><td>Defines the maximum allowed size of EDNS.0 UDP packets in bytes.

<table><tr><td>Default</td><td>1280 bytes</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Max concurrent queries

UCI: dhcp.@dnsmasq[0].dnsforwardmax

Opt: dnsforwardmax</td><td>Maximum allowed number of concurrent DNS queries.

<table><tr><td>Default: 150</td><td>1280 bytes</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

26.2.5. Active Leases

The active leases section

This section displays all currently active leases.

Web Field/UCI/Package Option	Description
Web: Hostname UCI: n/a Opt: n/a	Displays the hostname of the client.
Web: IPv4 Address UCI: n/a Opt: n/a	Displays the IP address of the client.
Web: MAC Address UCI: n/a Opt: n/a	Displays the MAC address of the client.
Web: Lease time remaining UCI: n/a Opt: n/a	Displays the remaining lease time.

26.2.6. Static Leases: DHCP Server And DNS

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.

The static leases section

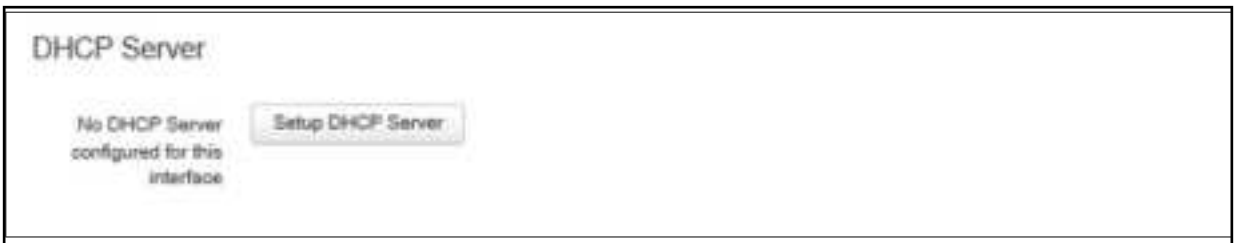
Web Field/UCI/Package Option	Description
Web: Hostname UCI: dhcp.@host[0].name Opt: name	Defines the optional symbolic name to assign to this static DHCP entry. Default: 0 — Disabled 1 — Enabled
Web: MAC Address UCI: dhcp.@host[0].mac Opt: mac	Defines the hardware address that identifies the host.
Web: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

26.2.7. Configuring DHCP Pools Using The Web Interface

DHCP pools are configured via the interface configuration.

Select **Network -> Interfaces**. Choose the interface you want to add the DHCP pool to and select **Edit.** Scroll to **DNCP Server** section.

Note: this section is only available for interfaces with a static IP address. To assign a DHCP Server to the interface, click **Setup DHCP Server**.



The DHCP server settings

The DHCP Server configuration options will appear. The DHCP Server is divided into two sub sections: General Setup and Advanced Settings.

DHCP server advanced settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Ignore interface

UCI: dhcp.@dhcp[x].ignore

Opt: ignore</td><td>Defines whether the DHCP pool should be enabled for this interface. If not specified for the DHCP pool then the default is disabled i.e. dhcp pool enabled.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Mode

UCI: dhcp.@dhcp[x].mode

Opt: mode</td><td>Defines whether the DHCP pool should be enabled for this interface. If not specified for the DHCP pool then the default is disabled i.e. dhcp pool enabled.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>Default: DHCPv4</td><td>DHCP for IPv4</td><td>ipv4</td></tr><tr><td>DHCPv6</td><td>DHCP for IPv6</td><td>ipv6_dhcp</td></tr><tr><td>IPv6 Router Advertisements</td><td>IPv6 RA</td><td>ipv6_ra</td></tr><tr><td>DHCPv6 Prefix Delegation</td><td>DHCPv6 prefix delegation</td><td>ipv6_pd</td></tr></table></td></tr>
<tr><td>Web: Start

UCI: dhcp.@dhcp[x].start

Opt: start</td><td>Defines the offset from the network address for the start of the DHCP pool.

Example: for network address 192.168.100.10/24, start=100, DHCP allocation pool will start at 192.168.100.100.

For subnets greater than /24, it may be greater than 255 to span subnets. Alternatively, specify in IP address notation using the wildcard ‘0’ where the octet is required to inherit bits from the interface IP address.

Example: to define a DHCP scope starting from 10.1.20.0 on an interface with 10.1.0.0/16 address, set start to 0.0.20.1

<table><tr><td>Default</td><td>100</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Limit

UCI: dhcp.@dhcp[x].limit

Opt: limit</td><td>Defines the size of the address pool.

Example: For network address 192.168.100.10/24, start=100, limit=150, DHCP allocation pool will be .100 to .249

<table><tr><td>Default: 150</td><td>Limits DHCP allocation pool to 50 available addresses</td></tr><tr><td>Range</td><td>0-255</td></tr></table></td></tr>
<tr><td>Web: Leasetime

UCI: dhcp.@dhcp[x].leasetime

Opt: leasetime</td><td>Defines the lease time of addresses handed out to clients, for example 12h or 30m.

<table><tr><td>Default</td><td>12 hours</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: dhcp.@dhcp[x].interface

Opt: interface</td><td>Defines the interface that is served by this DHCP pool. This must be one of the configured interfaces.

When configured through the web UI this will be automatically populated with the interface name.</td></tr>
</table>

DHCP Server: Advanced Settings

The DHCP server advanced settings section

Web Field/UCI/Package Option	Description
Web: Dynamic DHCP UCI: dhcp.@dhcp[x].dynamicdhcp Opt: dynamicdhcp	Defines whether to dynamically allocate DHCP leases. Default 1 — Dynamically allocate leases 0 — Use /etc/ethers file for serving DHCP leases
Web: Force UCI: dhcp.@dhcp[x].force Opt: force	Forces DHCP serving on the specified interface even if another DHCP server is detected on the same network segment. 0 — Disabled 1 — Enabled
Web: IPv4-Netmask UCI: dhcp.@dhcp[x].netmask Opt: netmask	Defines a netmask sent to clients that overrides the netmask as calculated from the interface subnet. Default — Use netmask from interface subnet Range —
Web: DHCP-Options UCI: dhcp.@dhcp[x].dhcp_option Opt: list dhcp_option	Defines additional options to be added for this dhcp pool. For example, with 'list dhcp_option 26,1470' or 'list dhcp_option mtu, 1470' you can assign a specific MTU per DHCP pool. Your client must accept the MTU option for this to work. Options that contain multiple values should be separated by a comma. Example: list dhcp_option 6,192.168.2.1,192.168.2.2 Default — No options defined Syntax — Option_number, option_value
Web: n/a UCI: dhcp.@dhcp[x].networkid Opt: networkid	Assigns a network-id to all clients that obtain an IP address from this pool. Default — Use network from interface subnet Range —

26.3. Configuring DHCP And DNS Using Command Line

Possible section types of the DHCP configuration file include Common Options (dnsmasq), DHCP Pools (dhcp) and Static Leases (host). Not all types may appear in the file and most of them are only needed for special configurations.

The configuration section type **dnsmasq** determines values and options relevant to the overall operation of dnsmasq and the DHCP options on all interfaces served.

26.3.1. Configuring Static Leases Using Command Line

Static leases are configured under the **dhcp** package, stored at **/etc/config/dhcp**. By default, all static leases instances are named **host**. The static lease is identified by @host then the static lease position in the package as a number. For example, for the first static lease in the package using UCI:

```
dhcp.@host[0]=dhcp
dhcp.@host[0].name=mypc
```

Or using package options:

```
config host
option name 'mypc'
```

However, to better identify, it is recommended to give the static lease instance a name. For example, to create a static instance named mypc.

To define a named static lease instance using UCI, enter:

```
dhcp.mypc=host
dhcp.mypc.name=mypc
```

To define a named static lease instance using package options, enter:

```
config dhcp 'mypc'
option name 'mypc'
```

The following examples using UCI and package options add 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.

Example of Static Leases using UCI

```
root@VA_router:~# uci show dhcp.mypc
dhcp.mypc=host
dhcp.mypc.ip=192.168.1.2
dhcp.mypc.mac=00:11:22:33:44:55
dhcp.mypc.name=mypc
```

Example of Static Leases using Package Options

```
root@VA_router:~# uci export dhcp package dhcp
config host 'mypc'
option ip '192.168.1.2'
option mac '00:11:22:33:44:55'
option name 'mypc'
```

26.3.2. Configuring DHCP Pools Using Command Line

DHCP pools are configured under the dhcp package, stored at **/etc/config/dhcp**.

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.

You can configure multiple dhcp pools.

By default, all dhcp pool instances are named 'dhcp'. The instance is identified by @dhcp then the dhcp pool position in the package as a number. For example, for the first dhcp pool in the package using UCI:

```
dhcp.@dhcp[0]=dhcp
dhcp.@dhcp[0].interface=LAN
```

Or using package options:

```
config dhcp
option interface 'LAN'
```

However, to better identify, it is recommended to give the dhcp pool instance a name. For example, to create a dhcp pool instance named LAN.

To define a named dhcp pool instance using UCI, enter:

```
dhcp.LAN=dhcp
dhcp.LAN.interface=LAN
```

To define a named dhcp pool instance using package options, enter:

```
config dhcp 'LAN'
option interface 'LAN'
```

Configuring DHCP pools using UCI

```
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
```

Configuring DHCP pools using package options

```
root@VA_router:~# uci export dhcp
package dhcp
...
config 'dhcp' 'LAN'
option 'interface' 'LAN' option 'start' '100' option 'limit' '150' option 'leasetime' '12h' option ignore 0
```

27. Configuring DHCP Client

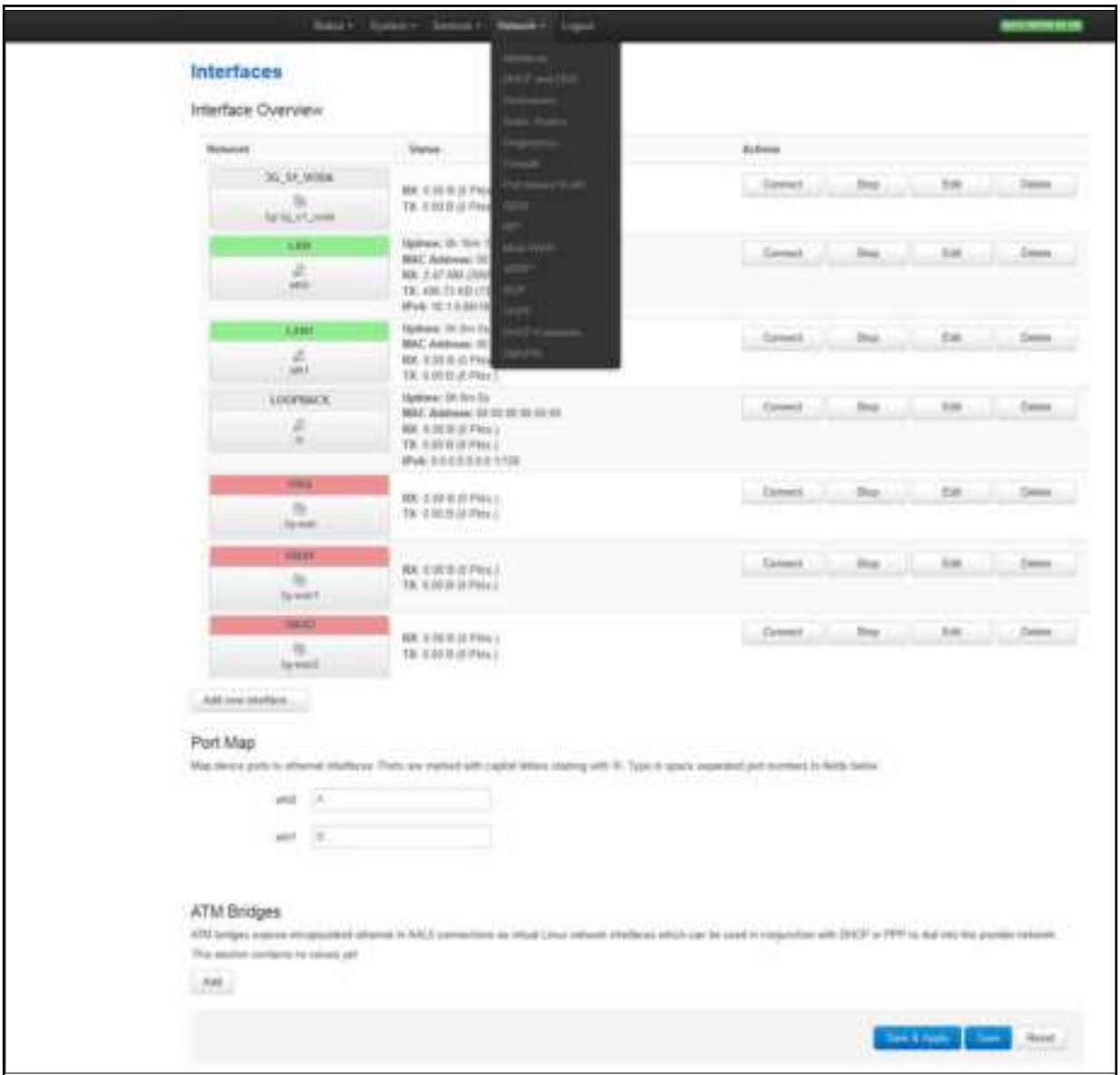
This section describes how to configure an interface as a DHCP client. This section will only detail the configuration for DHCP client. For information on how to configure other interface options such as firewall zone, mapping of switch ports, etc, read the standard interface configuration document.

Configuration Packages Used

Package	Sections
Network	interface

27.1. 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.



The interfaces overview page

There are three sections in the Interfaces page.

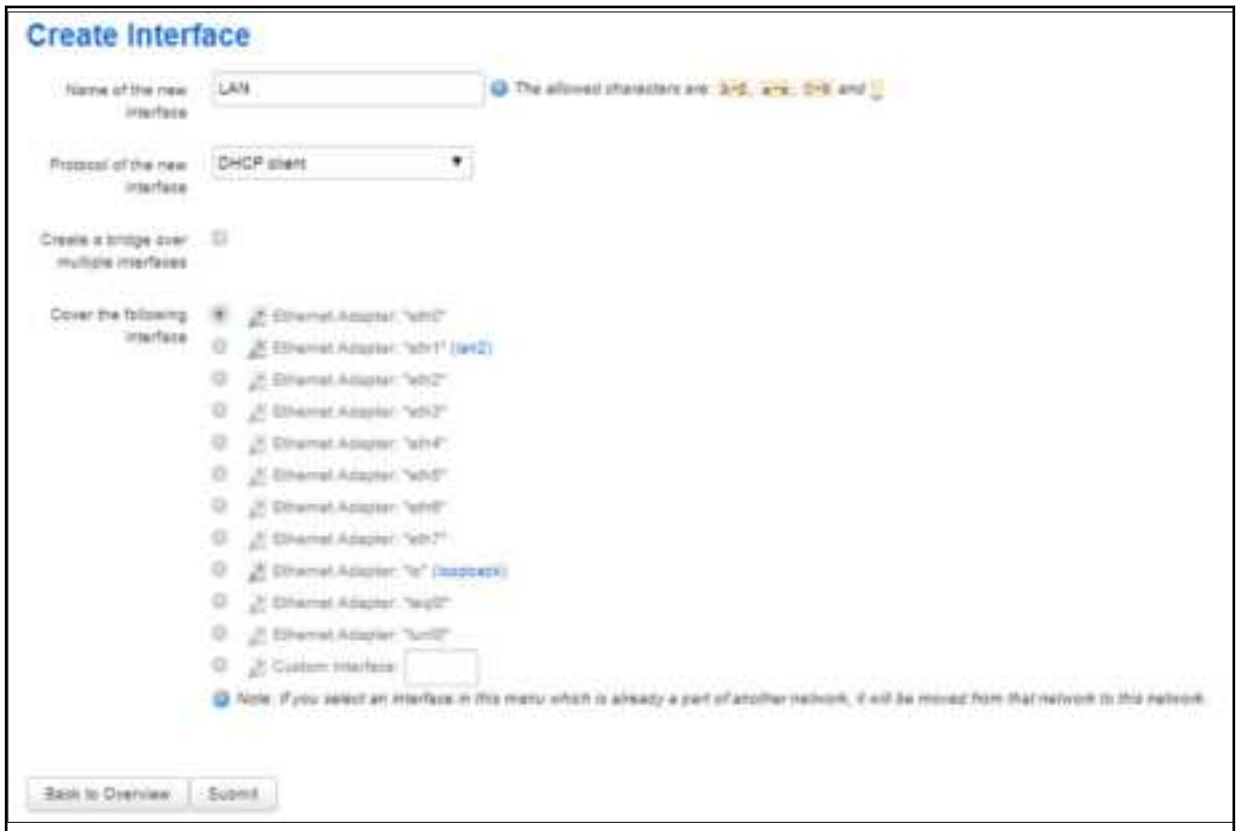
Section	Description
Interface Overview	Shows existing interfaces and their status. You can create new, and edit existing interfaces here.
Port Map	In this section you can map device ports to Ethernet interfaces. Ports are marked with capital letters starting with 'A'. Type in space-separated port character 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.

Edit 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.

27.1.1. 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.



The create interface page

Web Field/UCI/Package Option	Description
Web: Name of the new interface UCI: network.<if name> Opt: config interface	Assigns a logical name to the interface. The network interface section will assign this name (<if name>). Type the name of the new interface. Allowed characters are A-Z, a-z, 0-9 and _
Web: Protocol of the new interface UCI: network.<if name>.proto Opt: proto	Specifies what protocol the interface will operate on. Select DHCP Client. Option / Description / UCI: Static / static configuration with fixed address and netmask. / static DHCP Client / Address and netmask are assigned by DHCP / dhcp Unmanaged / Unspecified / Empty IPv6-inIPv4 (RFC4213) / Used with tunnel brokers / IPv6-over-IPv4 / Stateless IPv6 over IPv4 transport / GRE / Generic Routing Encapsulation protocol / IOT / / L2TP / Layer 2 Tunnelling Protocol / PPP / Point to Point Protocol / PPPoE / PPP over Ethernet / PPPoATM / PPP over ATM / LTE/UMTS/GPRS/EV-DO / CDMA, UMTS or GPRS connection using an AT-style 3G modem /
Web: Create a bridge over multiple interfaces UCI: network.<if name>.type 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. Default / Empty Bridge / Configures a bridge over multiple interfaces
Web: Cover the following interface UCI: network.<if name>.ifname 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'.

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.

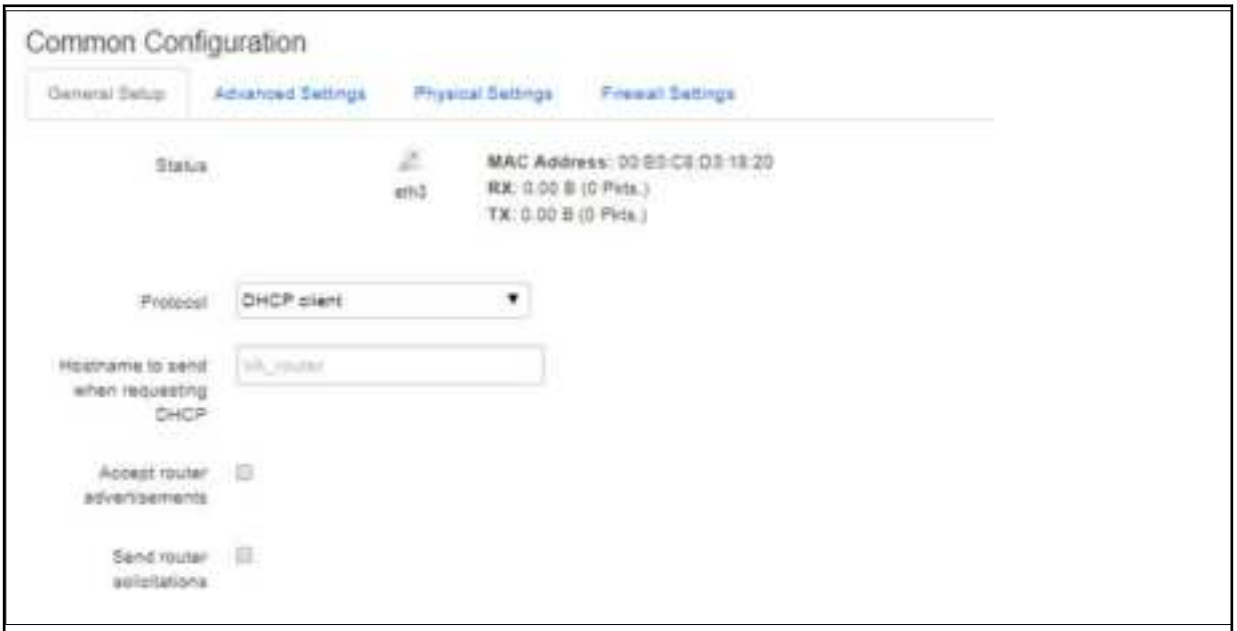
27.2. Common Configuration

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.

27.2.1. Common Configuration: General Setup



The interface general setup configuration page for DHCP client protocol

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Status</td><td>Shows the current status of the interface.</td></tr>
<tr><td>Web: Protocol

UCI: network.<if name>.proto Opt: proto</td><td>Protocol type. The interface protocol may be one of the options shown below. The protocol selected in the previous step will be displayed as default but can be changed if required.

Select DHCP Client.
<table>
<tr><th>Option</th><th>Description</th><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-inIPv4 (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: Hostname to send when requesting DHCP

UCI: network.<if name>.hostname

Opt: hostname</td><td>Defines the hostname to include in DHCP requests</td></tr>
<tr><td>Web: Accept router advertisements

UCI: network.<if name>.accept_ra

Opt: accept_ra</td><td>Specifies whether to accept IPv6 Router Advertisements on this interface (optional).

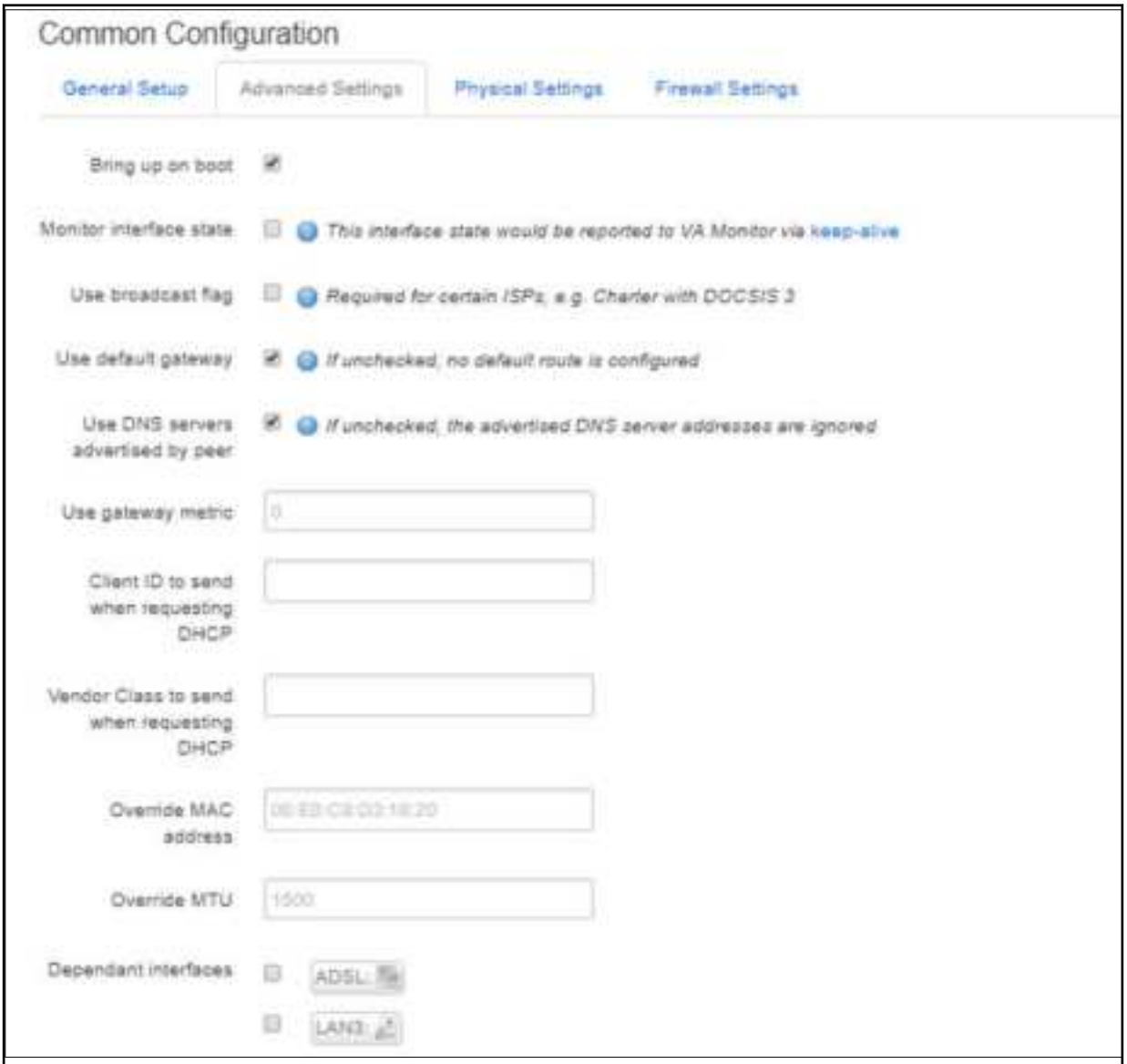
Note: default is 1 if protocol is set to DHCP, otherwise the setting defaults to 0.
<table>
<tr><td>0</td><td>Does not accept IPv6 router advertisements</td></tr>
<tr><td>Default</td><td>Accepts IPv6 router advertisements</td></tr>
</table></td></tr>
<tr><td>Web: Send router solicitations

UCI: network.<if name>.send_rs

Opt: send_rs</td><td>Specifies whether to send router solicitations on this interface (optional).

Note: defaults to 1 for static protocol, otherwise the setting defaults to 0.
<table>
<tr><td>Default 0</td><td>Does not send router solicitations</td></tr>
<tr><td>1</td><td>Sends router solicitations</td></tr>
</table></td></tr>
</table>

27.2.2. Common Configuration: Advanced Settings



The interface advanced settings page for DHCP client protocol

Web Field/UCI/Package Option	Description
Web: Bring up on boot UCI: network.<if name>.auto Opt: auto	Enables the interface to connect automatically on boot up. 0 Disabled 1 Enabled
Web: Monitor interface state UCI: network.<if name>.monitored Opt: monitored	Enabled if the status of the interface is presented on the monitoring platform. 0 Disabled 1 Enabled
Web: Use broadcast flag UCI: network.<if name>.broadcast Opt: broadcast	Enables the broadcast flag in DHCP requests (required for certain ISPs). 0 Disabled 1 Enabled
Web: Use default gateway UCI: network.<if name>.gateway Opt: gateway	Defines whether to suppress the DHCP assigned default gateway. When disabled via web option, the gateway is set to 0.0.0.0. 0 Disabled 1 Enabled
Web: Use DNS servers advertised by peer UCI: n/a Opt: n/a	Defines whether to override DHCP assigned DNS servers with configured list of DNS servers. When unchecked allows configuration of custom DNS servers via web. There is no uci option set when checking or unchecking this option.
Web: Use custom DNS servers UCI: network.<if name>.dns Opt: dns	Defines whether to override DHCP assigned DNS servers with configured list of DNS servers. Multiple DNS Servers are separated by a space if using UCI. Example: option dns '1.1.1.1 2.2.2.2' Default. 0 Disabled. (option gateway set to 0.0.0.0) 1 Enabled
Web: Use gateway metric UCI: network.<if name>.metric Opt: metric	Specifies the default route metric to use for this interface. Default. 0 Disabled Range
Web: Client ID to send when requesting DHCP UCI: network.< name> clientid Opt: clientid	Defines whether to override the client identifier in DHCP requests. Default Blank. Do not override Range Override
Web: Vendor Class to send when requesting DHCP UCI: network.<name>.vendorid Opt: vendorid	Defines whether to override the vendor class in DHCP requests Default Blank. Do not override Range OVerride
Web: Override MAC address UCI: network.< name>.macaddr Opt: macaddr	Overrides the MAC address assigned to this interface. Must be in the form: hh:hh:hh:hh:hh:hh, where h is a hexadecimal number.
Web: Override MTU UCI: network.<name>.mtu Opt: mtu	Defines the value to override the default MTU on this interface. 1500 1500 bytes Default 1500 bytes
Web: Dependant Interfaces UCI: network.[if_name].dependants	Lists interfaces that are dependant 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.

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Opt: dependants</td><td>Separate multiple interfaces by a space when using UCI. Example: option dependants 'PPPADSL MOBILE'

This replaces the following previous options in child interfaces.

<table><tr><td>gre</td><td>option local_interface</td></tr><tr><td>lt2p</td><td>option src_ipaddr</td></tr><tr><td>iot</td><td>option wan1 wan2</td></tr><tr><td>6in4</td><td>option ipaddr</td></tr><tr><td>6to4</td><td>option ipaddr</td></tr></table></td></tr>
<tr><td>Web: SNMP Alias ifIndex

UCI: network.@interface[X].snmp_alias_ifindex

Opt: snmp_alias_ifindex</td><td>Defines a static SNMP interface alias index for this interface, that can be polled using via the SNMP interface index (snmp_alias_ifindex+1000)

<table><tr><td>Blank</td><td>No SNMP interface alias index</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
</table>

27.3. Configuring DHCP Client Using Command Line

The configuration files for DHCP client are stored on **/etc/config/network**

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
```

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'
```

27.4. DHCP Client Diagnostics

Interface Status

To view the IP address of DHCP client interface, enter:

```
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)
```

ARP Table Status

```
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
```

Route Status

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.

28. Configuring DHCP Forwarding

This section describes how to configure the router to forward DHCP requests from an interface to a network DHCP server.

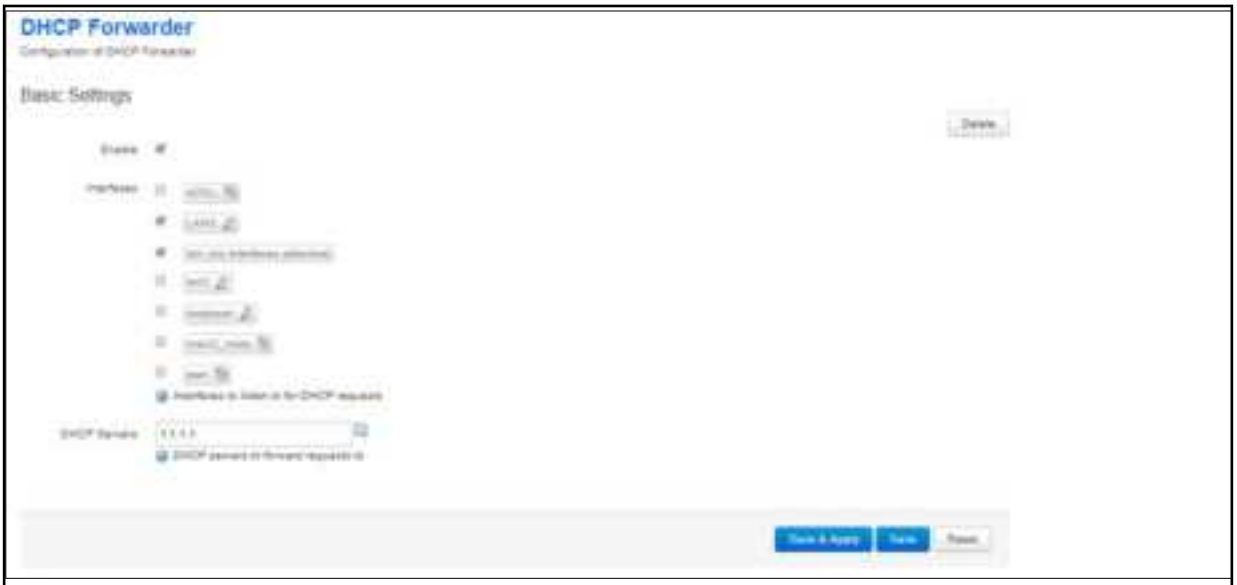
Configuration Packages Used

Package	Sections
dhcp_fwd	dhcpfwd

28.1. Configuring DHCP 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.



The DHCP forwarder configuration page

Web Field/UCI/Package Option	Description
Web: Enabled UCI: dhcp_fwd.main.enabled Opt: enabled	Defines whether DHCP forwarding is enabled or disabled. Default: 0 — Do not send router solicitations 1 — Send router solicitations
Web: Interfaces UCI: dhcp_fwd.main.listen_interface Opt: list listen_interface	Defines a list of the source interface name(s) to forward DHCP messages from. Multiple interface_name(s) are entered using uci set and uci add_list commands. Example: uci set dhcp_fwd.main.listen_interface=LAN1 uci add_list dhcp_fwd.main.listen_interface=LAN2 or using a list of options via package options list listen_interface 'LAN1' list listen_interface 'LAN2'
Web: DHCP Servers UCI: dhcp_fwd.main.server Opt: list server	Defines a list of the network DHCP servers to forward DHCP messages to. Multiple interface_name(s) are entered using uci set and uci add_list commands. Example: uci set dhcp_fwd.main.server=1.1.1.1 uci add_list dhcp_fwd.main.main.server=2.2.2.2 or using a list of options via package options list server '1.1.1.1' list server '2.2.2.2'

28.2. Configuring DHCP Forwarding Using Command Line

The configuration files for DHCP client are stored in **/etc/config/dhcp_fwd**

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
```

DHCP Forwarding using Package Options

```
root@VA_router:~# uci export dhcp_fwd
package dhcp_fwd
config dhcpfwd 'main'
 option enabled '1'
list listen_interface 'lan2'
list server '1.1.1.1'
```

28.3. 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.

Package	Sections
firewall	redirect

28.3.1. Configuring Source NAT For DHCP Forwarding Over IPSec

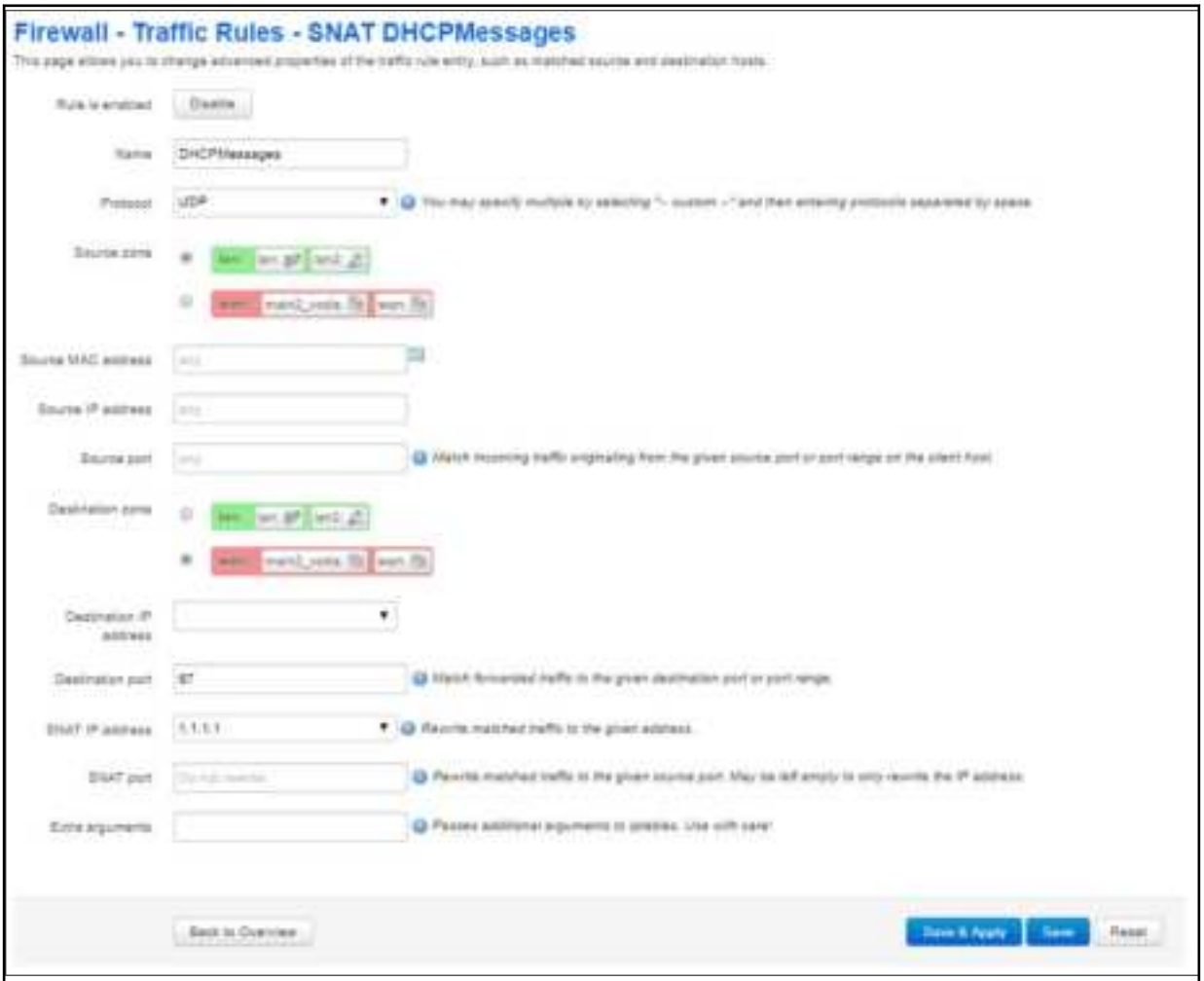
To enter a source NAT rule, browse to **Network -> Firewall**. Select **Traffic Rules** tab. The Firewall - Traffic Rules page appears.

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' chapter of this manual.

The firewall > traffic rules configuration page

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



The firewall > traffic rules > SNAT configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled

UCI: firewall.@redirect[X].enabled

Opt: enabled</td><td>Defines whether source NAT rule is enabled.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Name

UCI: firewall.@redirect[X].name

Opt: name</td><td>Defines a name for the source NAT rule.</td></tr>
<tr><td>Web: Protocol

UCI: firewall.@redirect[X].proto

Opt: proto</td><td>Defines the protocol for the source NAT rule to match. Select UDP.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>All protocols</td><td>Match all protocols</td><td>all</td></tr><tr><td>TCP+UDP</td><td>Match TCP and UCP protocols</td><td>tcp upd</td></tr><tr><td>TCP</td><td>Match TCP protocol</td><td>tcp</td></tr><tr><td>UCP</td><td>Match UDP protocol</td><td>udp</td></tr><tr><td>ICMP</td><td>Match ICMP protocol</td><td>icmp</td></tr><tr><td>Custom</td><td>Enter custom protocol</td><td></td></tr></table></td></tr>
<tr><td>Web: Source Zone

UCI: firewall.@redirect[X].src

Opt: src</td><td>Defines the source interface for the source NAT rule.

Select the interface where the DHCP requests are originating.</td></tr>
<tr><td>Web: Destination Zone

UCI: firewall.@redirect[X].dest

Opt: dest</td><td>Defines destination interface for the source NAT rule.

Select the interface where the DHCP requests are intended to be transmitted.</td></tr>
<tr><td>Web: Destination port

UCI: firewall.@redirect[X].port

Opt: port</td><td>Defines the destination port number to match. Select 67.</td></tr>
<tr><td>Web: SNAT IP address

UCI: firewall.@redirect[X].src_dip

Opt: src_dip</td><td>Defines the IP address to rewrite matched traffic.

Select the source IP address to match the required IPSec rule.</td></tr>
</table>

28.4. Configuring Source NAT For DHCP Forwarding Over IPSec

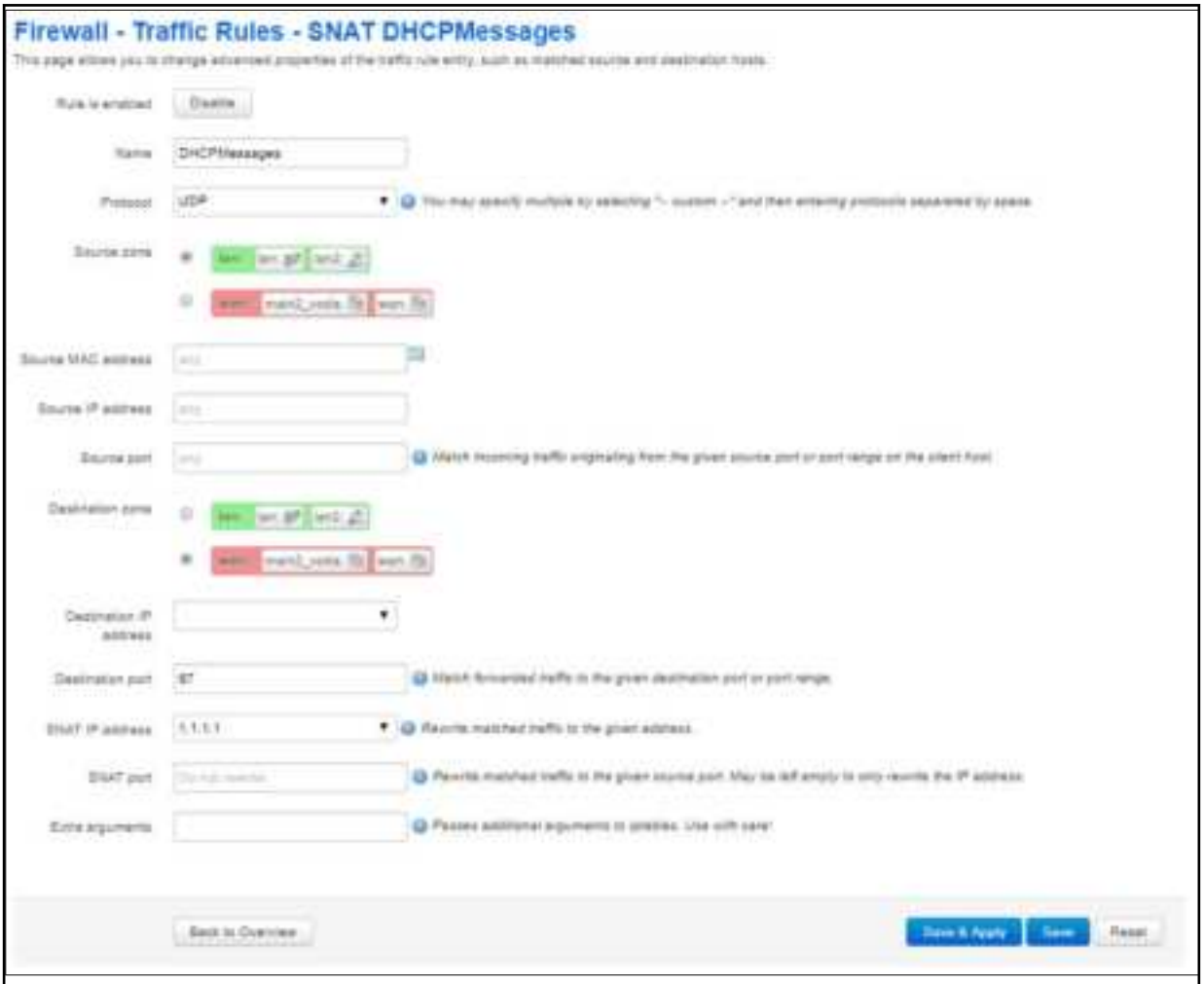
To enter a source NAT rule, browse to **Network -> Firewall**. Select **Traffic Rules** tab. The Firewall - Traffic Rules page appears.

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' chapter of this manual.

The firewall > traffic rules configuration page

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



The firewall > traffic rules > SNAT configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled

UCI: firewall.@redirect[X].enabled

Opt: enabled</td><td>Defines whether source NAT rule is enabled.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Name

UCI: firewall.@redirect[X].name

Opt: name</td><td>Defines a name for the source NAT rule.</td></tr>
<tr><td>Web: Protocol

UCI: firewall.@redirect[X].proto

Opt: proto</td><td>Defines the protocol for the source NAT rule to match. Select UDP.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>All protocols</td><td>Match all protocols</td><td>all</td></tr><tr><td>TCP+UDP</td><td>Match TCP and UCP protocols</td><td>tcp upd</td></tr><tr><td>TCP</td><td>Match TCP protocol</td><td>tcp</td></tr><tr><td>UCP</td><td>Match UDP protocol</td><td>udp</td></tr><tr><td>ICMP</td><td>Match ICMP protocol</td><td>icmp</td></tr><tr><td>Custom</td><td>Enter custom protocol</td><td></td></tr></table></td></tr>
<tr><td>Web: Source Zone

UCI: firewall.@redirect[X].src

Opt: src</td><td>Defines the source interface for the source NAT rule.

Select the interface where the DHCP requests are originating.</td></tr>
<tr><td>Web: Destination Zone

UCI: firewall.@redirect[X].dest

Opt: dest</td><td>Defines destination interface for the source NAT rule.

Select the interface where the DHCP requests are intended to be transmitted.</td></tr>
<tr><td>Web: Destination port

UCI: firewall.@redirect[X].port

Opt: port</td><td>Defines the destination port number to match. Select 67.</td></tr>
<tr><td>Web: SNAT IP address

UCI: firewall.@redirect[X].src_dip

Opt: src_dip</td><td>Defines the IP address to rewrite matched traffic.

Select the source IP address to match the required IPSec rule.</td></tr>
</table>

28.5. Configuring Source NAT For DHCP Forwarding Over IPSec Using Command Line

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
```

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'
```

28.6. DHCP Forwarding Diagnostics

Tracing DHCP Packets

To trace DHCP packets on any interface on the router, enter:

```
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 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 -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
```

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
```

29. Configuring Dynamic DNS

Dynamic DNS (DDNS) functionality on a Merlin 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.

Configuration Packages Used

Package	Sections
ddns	service

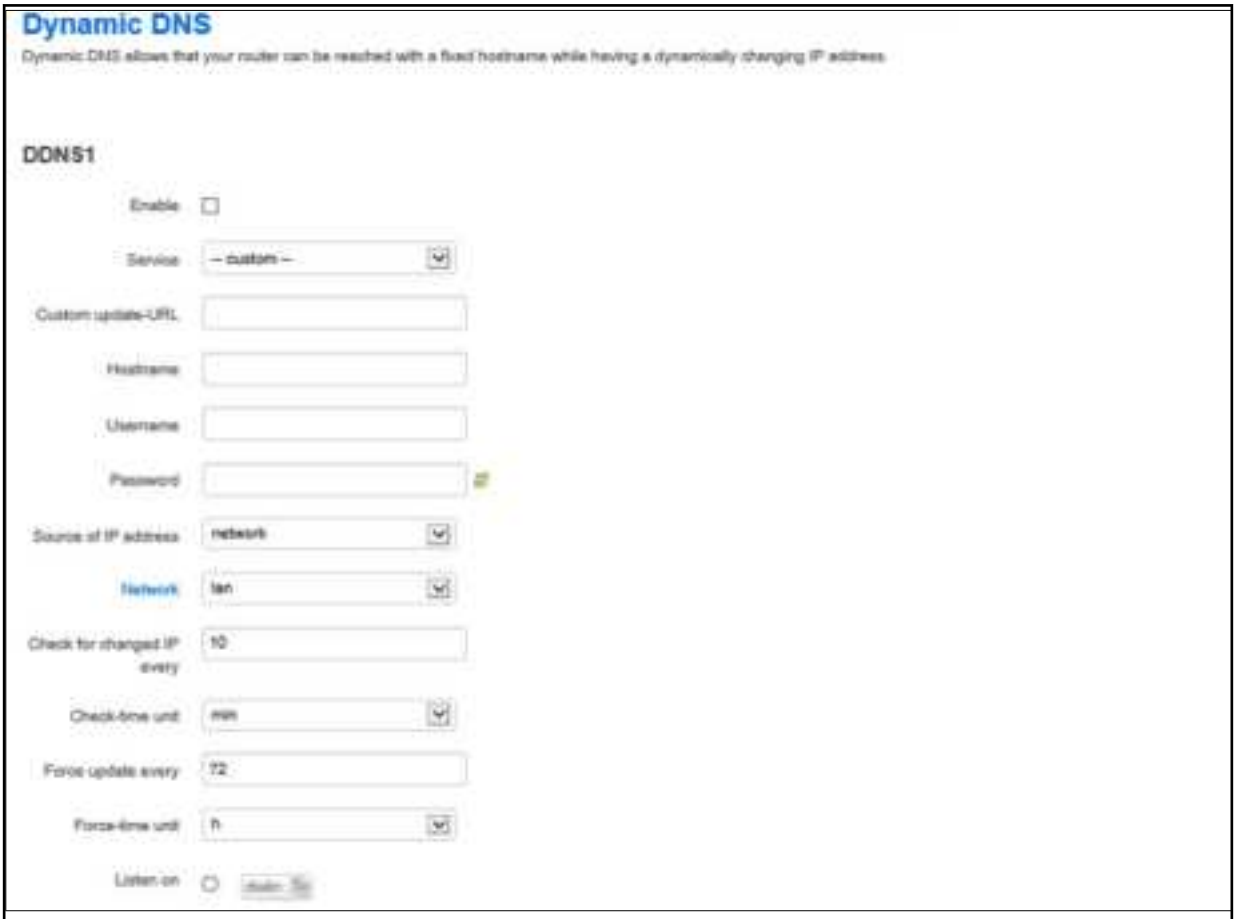
29.1. Configuring Dynamic DNS Using The Web Interface

In the top menu, select **Services -> Dynamic DNS**. The Dynamic DNS configuration page appears.

The Dynamic DNS configuration page

Enter a name that will be used for the dynamic DNS section in the configuration. Select Add. The Dynamic DNS configuration options appear.

Dynamic DNS Settings



The Dynamic DNS main settings page

Web Field/UCI/Package Option	Description
Web: Enabled UCI: `ddns. <name> .enabled` Opt: enabled	Enables a dynamic DNS entry on the router 0. Default: Disabled 1: Enabled
Web: Service UCI: `ddns. <name> .service_name` Opt: service_name	Defines the dynamic DNS provider.
Web: Customer update-URL UCI: `ddns. <name> .update_url` Opt: update_url	Defines the customer DNS provider. Displayed when the service is set to custom in the web interface.
Web: Hostname UCI: `ddns.<name>.domain` Opt: domain	Defines the fully qualified domain name associated with this entry. This is the name to update with the new IP address as needed.
Web: Password UCI: ddns.<name>.password Opt: password	Defines the password to use for authenticating domain name updates with the selected provider.
Web: Source of IP address UCI: `ddns.<name>.ip_source` Opt: ip_source	Defines the type of interface whose IP needs to be updated. network: IP is associated with a network configuration. interface: IP is associated with an interface. web: IP is associated with a URL.
Web: Network UCI: `ddns.<name>ip_network` Opt: ip_network	Defines the network whose IP needs to be updated. Displayed when the Source of IP address option is set to network. All the configured network interfaces will be shown.
Web: Interface UCI: `ddns.<name>.ip_interface` Opt: ip_interface	Defines the interface whose IP needs to be updated. Displayed when the Source of IP address option is set to interface. All the configured interfaces will be shown.
Web: URL UCI: `ddns.<name>.ip_url` Opt: ip_url	Defines the URL where the IP downloaded from. Displayed when the Source of IP address option is set to URL.
Web: Check for changed IP every UCI: `ddns.<name>.check_interval` Opt: check_interval	Defines how often to check for an IP change. Used in conjunction with check_unit. Default: 10 Range:
Web: Check-time unit UCI: `ddns.<name>.check_unit` Opt: check_unit	Defines the time unit to use for check for an IP change. Used in conjunction with check_interval. Default: 72 Range: Enabled
Web: Force update every UCI: `ddns.<name>.force_interval` Opt: force_interval	Defines how often to force an IP update to the provider. Used in conjunction with force_unit. Default: 72. Disabled Range: Enabled

Web Field/UCI/Package Option	Description
Web: Force-time unit UCI: `ddns.<name>.force_unit` Opt: force_unit	Defines the time unit to use for check for an IP change. Used in conjunction with force_interval. Default: Hours Range: Minutes-hours
Web: Listen on UCI: `ddns.<name>.interface` Opt: interface	Defines the interface for ddns monitoring. Typically, this will be the same as the interface whose IP is being updated – as defined ip_network or ip_interface. All configured interfaces will be displayed.

29.2. Dynamic DNS Using UCI

Dynamic DNS uses the ddns package **/etc/config/ddns**

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'
```

30. Configuring Host Names

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

Local Host File Records

The hosts file is an operating system 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 router's local host file contains:

```
127.0.0.1 localhost

::1 ip6-localhost ip6-loopback
```

The local host file is stored at **/etc/hosts**

Configuration Packages Used

Packages	Sections
network	host

30.1. 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.

The host records page

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

The hosts records configuration page

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

30.2. Local Host Records Using Command Line

Local host records are configured in the host section of the network package

/etc/config/network.

You can configure multiple hosts.

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'
```

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
```

Local Host Records using Package Options

```
root@VA_router:~# uci export network package network
::::
config host
option hostname 'Device1'
option addr '1.1.1.1'
```

30.3. Local Host Records Diagnostics

Hosts Files

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
```

30.4. Configuring PTR Records Using The Web Interface

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.

Configuration Packages Used

Package	Sections
dhcp	domain

In the top menu, select **Network -> Hostnames**. The Hostnames configuration page appears.

The hostnames entries page

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

The hostnames configuration page

30.5. 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'
```

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
```

PTR Records using Package Options

```
root@VA_router:~# uci export dhcp package dhcp
::::
config domain
option name 'Domain1'
option ip '2.2.2.2'
```

30.6. PTR Records Diagnostics

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
```

30.7. Configuring Static Leases Using The Web Interface

Static leases are used to assign fixed IP addresses and symbolic hostnames to DHCP clients based on the MAC (hardware) address.

They are also required for non dynamic interface configurations where only hosts with a corresponding lease are served.

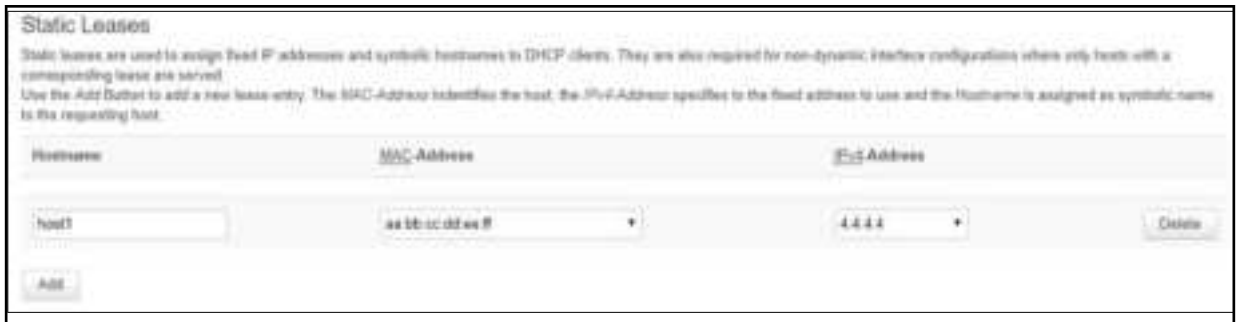
Package	Sections
dhcp	host

In the top menu, select **Network -> DHCP and DNS**. The DHCP and DNS configuration page appears.

Browse to **Static leases** section.

The static leases add page

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



The static leases configuration page

Web Field/UCI/Package Option	Description
Web: Hostname UCI: dhcp.host.name Opt: name	Defines the symbolic hostname to assign.
Web: MAC-Address UCI: dhcp.host.mac Opt: mac	Defines the MAC address for this host. MAC addresses should be entered in the format aa:bb:cc:dd:ee:ff
Web: IPv4-Address UCI: dhcp.host.ip Opt: ip	Defines the IP address to be used for this host.

30.8. Configuring Static Leases Using Command Line

Static leases are configured under the **dhcp** package, stored at **/etc/config/dhcp**. By default, all static leases instances are named **host**. The static lease is identified by @host then the static lease position in the package as a number. For example, for the first static lease in the package using UCI:

```
dhcp.@host[0]=dhcp
dhcp.@host[0].name=mypc
```

Or using package options:

```
config host
option name 'mypc'
```

However, to better identify, it is recommended to give the static lease instance a name. For example, to create a static instance named mypc.

To define a named static lease instance using UCI, enter:

```
dhcp.mypc=host
dhcp.mypc.name=mypc
```

To define a named static lease instance using package options, enter:

```
config dhcp 'mypc'
option name 'mypc'
```

The following examples using UCI and package options add 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.

Example of Static Leases using UCI

```
root@VA_router:~# uci show dhcp.mypc

dhcp.mypc=host

dhcp.mypc.ip=192.168.1.2

dhcp.mypc.mac=00:11:22:33:44:55

dhcp.mypc.name=mypc
```

Example of Static Leases using Package Options

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

config host 'mypc'

option ip '192.168.1.2'

option mac '00:11:22:33:44:55'

option name 'mypc'
```

31. 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 configuring 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'.

Configuration Package Used

Package	Sections
firewall	

31.1. 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.

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 disconnect the failed interface and attempt to connect to the next best roaming interface.

31.2. Configuring Firewall Using The Web Interface

In the top menu, select **Network -> Firewall**. The Firewall page appears. It is divided into three sections:

Section	Description
General 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.

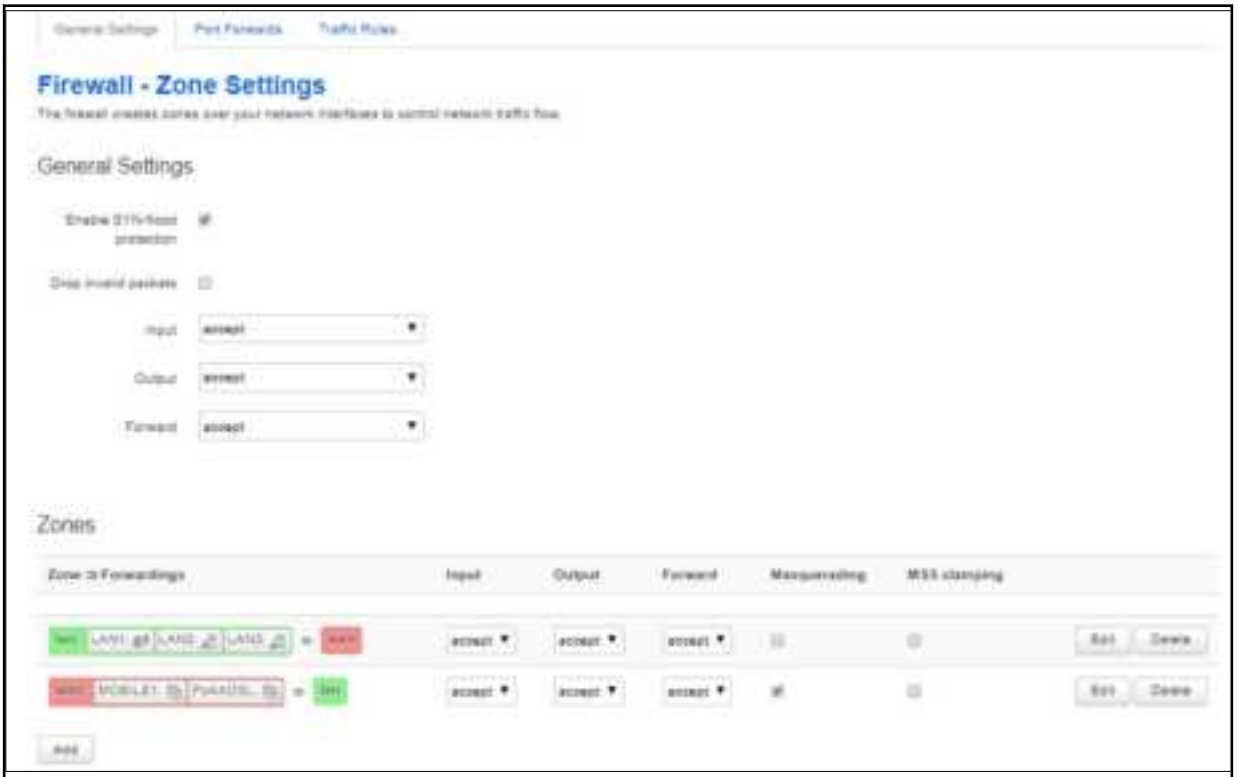
31.2.1. Firewall - Zone Settings - General Settings Page

The General Setting page is divided into two sections.

Section	Description
General Settings	Defines the global firewall setting 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.

Firewall - Zone Settings - General Settings section

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.



The Firewall - Zone settings - general settings page

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

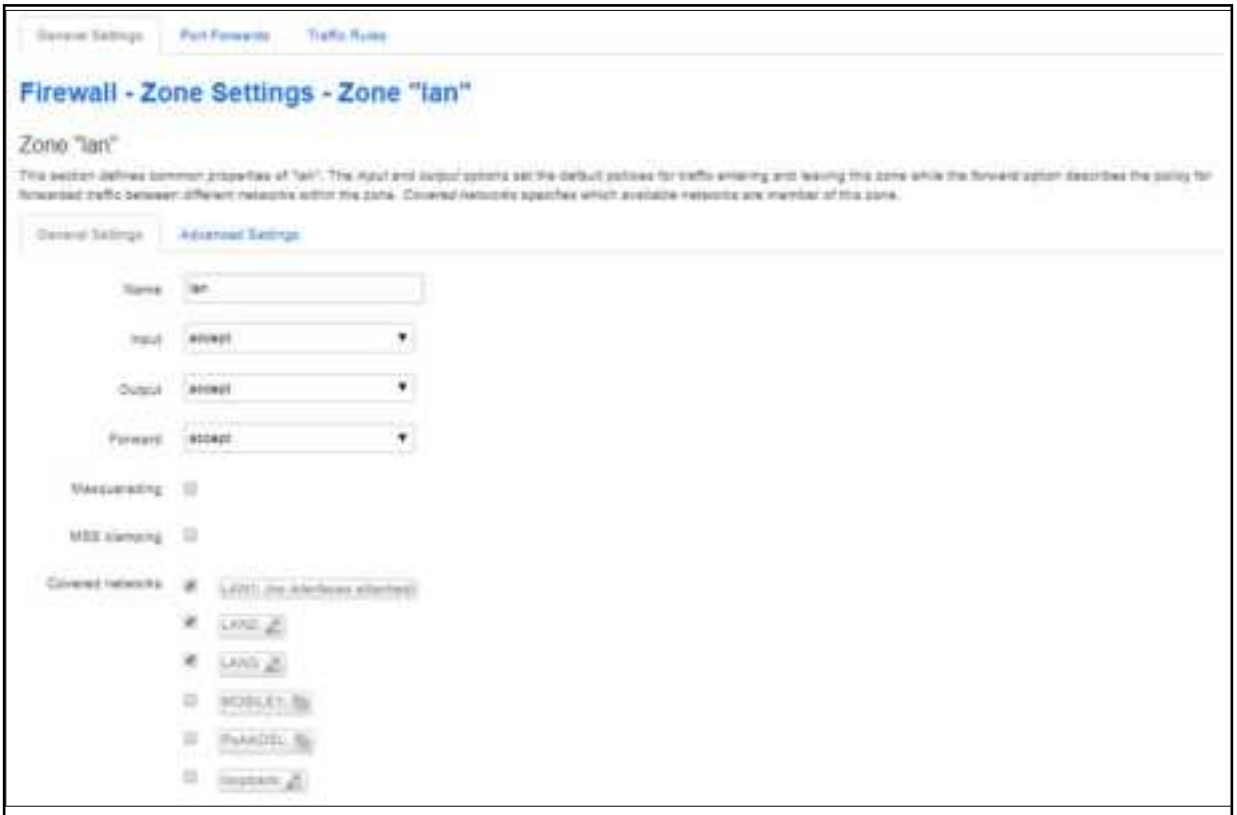
Firewall - Zone Settings - Zones Settings section

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 Firewall - Zone Settings - Zone "[name]" page appears.

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.

There are two sections in Firewall - Zone Settings - Zone "x" page: General Settings and Advanced settings

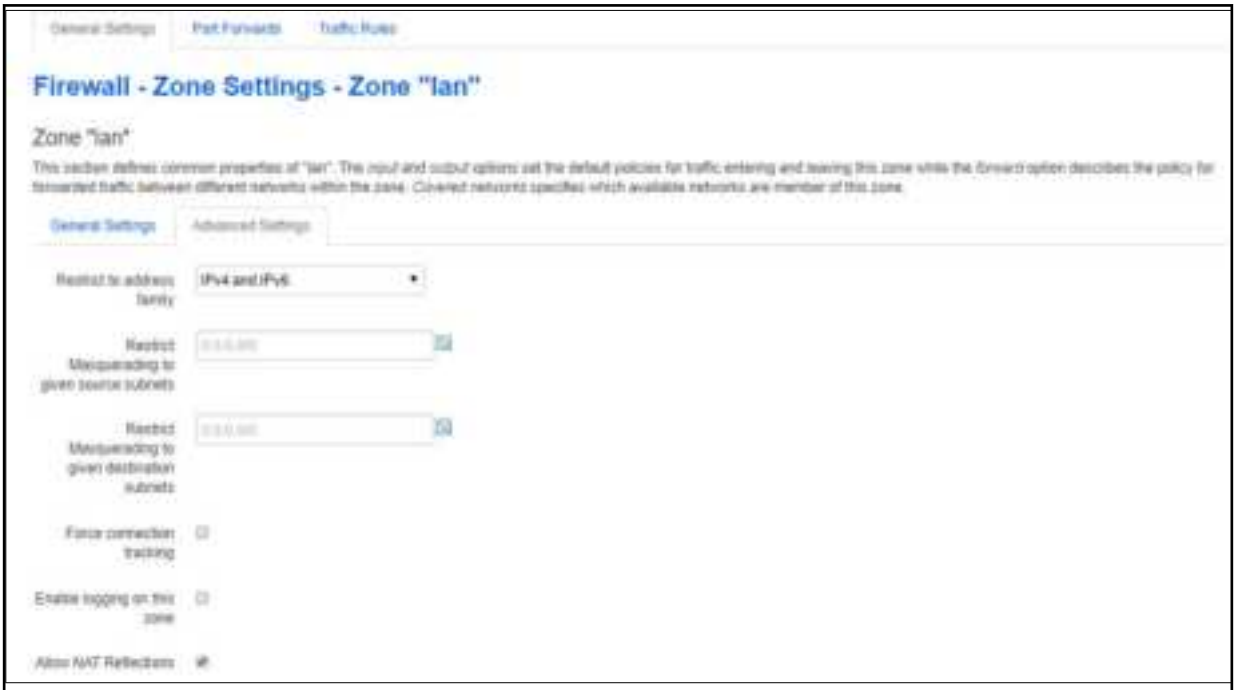
Firewall zone: general settings



The Firewall - zone settings - general settings tab

Web Field/UCI/Package Option	Description
Web: name UCI: firewall.<zone label>.name 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 UCI: firewall. <zone label>.input 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. Accept: Accepted packets pass through the firewall. Reject: Rejected packets are blocked by the firewall and ICMP message is returned to the source host. Default: Drop: Dropped packets are blocked by the firewall.
Web: Output UCI: firewall. <zone label>.output Opt: output	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. Accept: Accepted packets pass through the firewall. Reject: Rejected packets are blocked by the firewall and ICMP message is returned to the source host. Default: Drop: Dropped packets are blocked by the firewall.
Web: Forward UCI: firewall. <zone label>.forward Opt: forward	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. Accept: Accepted packets pass through the firewall. Reject: Rejected packets are blocked by the firewall and ICMP message is returned to the source host. Default: Drop: Dropped packets are blocked by the firewall.
Web: Masquerading UCI: firewall. <zone label>.masq Opt: masq	Specifies whether outgoing zone traffic should be masqueraded (NATTED). This is typically enabled on the wan zone.
Web: MSS Clamping UCI: firewall. <zone label>.mtu_fix Opt: mtu_fix	Enables MSS clamping for outgoing zone traffic. Subnets are allowed. Default: 0: Disabled 1: Enabled
Web: Covered Networks UCI: firewall.<zone label>.network Opt:	Defines a list of interfaces attached to this zone, if omitted, the value of name is used by default. Note: use the uci list syntax to edit this setting though UCI.

Firewall zone: advanced settings



The Firewall - zone settings - advanced settings tab

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Restrict to address family

UCI: firewall. <zone label>.family

Opt: family</td><td>Restricts zone to IPv4, IPv6 or both IPv4 and IPv6.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>IPv4 and IPv6</td><td>Any address family</td><td>any</td></tr><tr><td>IPv4 only</td><td>IPv4 only</td><td>ipv4</td></tr><tr><td>IPv6 only</td><td>IPv6 only</td><td>ipv6</td></tr></table></td></tr>
<tr><td>Web: Restrict Masquerading to given source subnets

UCI: firewall. <zone label>.masq_src

Opt: masq_src</td><td>Limits masquerading to the given source subnets. Negation is possible by prefixing the subnet with '!'. Multiple subnets are allowed.</td></tr>
<tr><td>Web: Restrict Masquerading to given destination subnets

UCI: firewall. <zone label>.masq_dest

Opt: masq_dest</td><td>Limits masquerading to the given destination subnets. Negation is possible by prefixing the subnet with '!'. Multiple subnets are allowed. Multiple IP addresses/subnets should be separated by a space, for example: option masq_dest '1.1.1.1 2.2.2.0/24'.</td></tr>
<tr><td>Web: Force connection tracking

UCI: firewall. <zone label>.conntrack

Opt: conntrack</td><td>Forces connection tracking for this zone.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>Default: 1</td><td>If masquerading is used. Otherwise, default is 0.</td></tr></table></td></tr>
<tr><td>Web: Enable logging on this zone

UCI: firewall. <zone label>.log

Opt: log</td><td>Creates log rules for rejected and dropped traffic in this zone.</td></tr>
<tr><td>Web: Allow NAT reflections

UCI: firewall. <zone label>.reflection

Opt: reflection</td><td>Enable/disable all NAT reflections for this zone.

Note: for configs with a large number of firewall rules, disabling NAT reflection will speed up load of firewall rules on interface start.

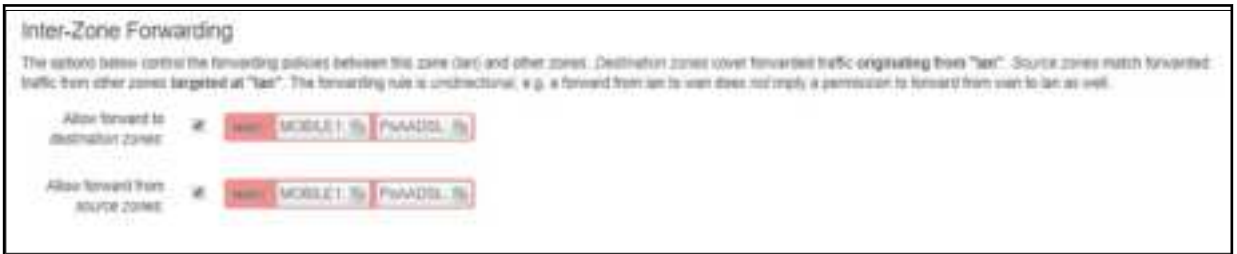
<table><tr><td>0</td><td>Disable reflection</td></tr><tr><td>Default: 1</td><td>Enable reflection</td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: firewall. <zone label>.log_limit

Opt: log_limit</td><td>Limits the amount of log messages per interval.</td></tr>
</table>

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.



The inter-zone forwarding section

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

31.2.2. Firewall Port Forwards Page

Port forwards are also known as redirects. The Firewall Port Forwards 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.

The firewall port forward page

To edit an existing port forward select **edit**. To add a new port forward select **add**.

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: name

UCI: firewall.<redirect label>.name

Opt: name</td><td>Sets the port forwarding name. For Web UI generated redirects the

<redirect label> takes the form of @redirect[x], where x is an integer starting from 0.</td></tr>
<tr><td>Web: Protocol

UCI: firewall.<redirect label>.proto

Opt: proto</td><td>Defines layer 4 protocol to match incoming traffic.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>Default: tcp+udp</td><td>Match either TCP or UDP packets</td><td>tcp udp</td></tr><tr><td>tcp</td><td>Match TCP packets only</td><td>tcp</td></tr><tr><td>udp</td><td>Match UDP packets only</td><td>udp</td></tr></table></td></tr>
<tr><td>Web: External port

UCI: firewall.<redirect label>.src_dport

Opt: src_dport</td><td>Specifies the incoming TCP/UDP port or port range to match. This is the incoming destination port specified by the external host. Port ranges specified as start:stop, for example, 2001:2020.

<table><tr><td>Default: Blank</td><td>Match traffic to any port</td></tr><tr><td>Range</td><td>1-65535</td></tr></table></td></tr>
<tr><td>Web: Internal IP address

UCI: firewall.<redirect label>.dest_ip

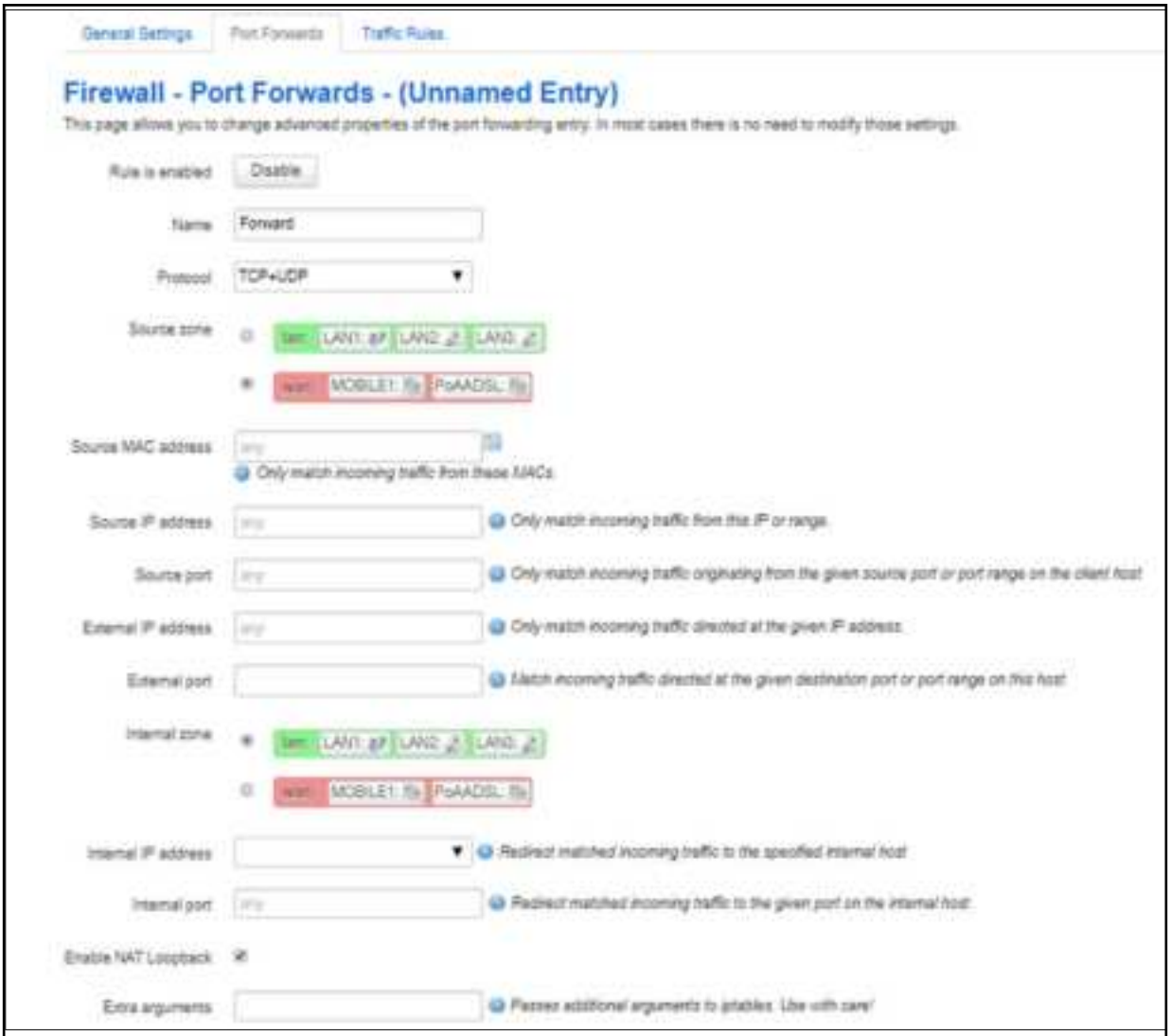
Opt: dest_ip</td><td>Specifies the internal (LAN) IP address for the traffic to be redirected to.</td></tr>
<tr><td>Web: Internal port

UCI: firewall.<redirect label>.dest_port

Opt: dest_port</td><td>Specifies the destination tcp/udp port for the redirect traffic.</td></tr>
</table>

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).



The firewall port forwards edits page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled

UCI: firewall.<redirect label>.enabled

Opt: enabled</td><td>Specifies if this redirect should be enabled or disabled.

<table><tr><td>0</td><td>Disabled</td></tr><tr><td>Default: 1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Name

UCI: firewall.<redirect label>.name

Opt: name</td><td>Sets the port forwarding name. For Web UI generated redirects the <redirect label> takes the form of @redirect[x], where x is an integer starting from 0.</td></tr>
<tr><td>Web: Protocol

UCI: firewall.<redirect label>.proto

Opt: proto</td><td>Defines layer 4 protocol to match incoming traffic.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>Default: tcp+udp</td><td>Match either TCP or UDP packets</td><td>tcp udp</td></tr><tr><td>tcp</td><td>Match TCP packets only</td><td>tcp</td></tr><tr><td>udp</td><td>Match UDP packets only</td><td>udp</td></tr></table></td></tr>
<tr><td>Web: Source zone

UCI: firewall.<redirect label>.src

Opt: src</td><td>Specifies the traffic source zone. It must refer to one of the defined zone names. When using the web interface, this is set to WAN initially.</td></tr>
<tr><td>Web: Source MAC address

UCI: firewall.<redirect label>.src_mac

Opt: list src_mac</td><td>Defines the list of source MAC addresses that this redirect will match.

Format: aa:bb:cc:dd:ee:ff

Multiple RIP interfaces are entered using <code>uci set</code> and <code>uci add_list</code> commands. Example:

<code>uci set firewall.@redirect[0].src_mac=aa:bb:cc:dd:ee:ff</code>

<code>uci add_list firewall.@redirect[0].src_mac=12:34:56:78:90:12</code>

or using a list of options via package options

<code>list network 'aa:bb:cc:dd:ee:ff'</code>

<code>list network '12:34:56:78:90:12'</code></td></tr>
<tr><td>Web: Source IP address

UCI: firewall.<redirect label>.src_ip

Opt: src_ip</td><td>Defines a source IP address that this redirect will match.

<table><tr><td>Default: Blank</td><td>Match traffic from any source IP</td></tr><tr><td>Range</td><td>A.B.C.D/mask</td></tr></table></td></tr>
<tr><td>Web: Source port

UCI: firewall.<redirect label>.src_port

Opt: src_port</td><td>Defines a source IP port that this redirect will match. You can enter multiple ports, using a space separator.

*For example: option src_port '22 23'

*see note below on use with options src_dport and dest_port

<table><tr><td>Default: Blank</td><td>Match traffic from any source IP</td></tr><tr><td>Range</td><td>1 - 65535</td></tr></table></td></tr>
<tr><td>Web: External port

UCI: firewall.<redirect label>.src_dport

Opt: src_dport</td><td>Specifies the incoming TCP/UDP port or port range to match. This is the incoming destination port specified by the external host.

Port ranges specified in format start:stop, for example, 2001:2020.

You can enter multiple ports, using a space separator.

*For example: option src_dport '22 23'</td></tr>
</table>

Web Field/UCI/Package Option	Description
	*see note below on use with options src_port and dest_port Default: Blank — Match traffic from any source IP Range — 1 - 65535
Web: Internal zone UCI: firewall.<redirect label>.dest Opt: dest	Specifies the traffic destination zone, must refer to one of the defined zone names.
Web: Internal IP address UCI: firewall.<redirect label>.dest_port Opt: dest_port	Specifies the internal (LAN) IP address for the traffic to be redirected to.
Web: Internal zone UCI: firewall.<redirect label>.dest_ip Opt: dest_ip	Specifies the destination tcp/udp port for the redirect traffic. You can enter multiple ports, using a space separator. *For example: option dest_port '22 23' *See note below table on use with options src_port and src_dport.
Web: Enable NAT Loopback UCI: firewall.<redirect label>.reflection Opt: reflection	Enable or disable NAT reflection for this redirect. 0 — Reflection disabled Default: 1 — Reflection enabled
Web: Extra arguments UCI: firewall.<redirect label>.extra Opt: extra	Passes extra arguments to IP tables. This is useful to specify additional match options, like -m policy --dir in for IPSec. The arguments are entered as text strings.

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:

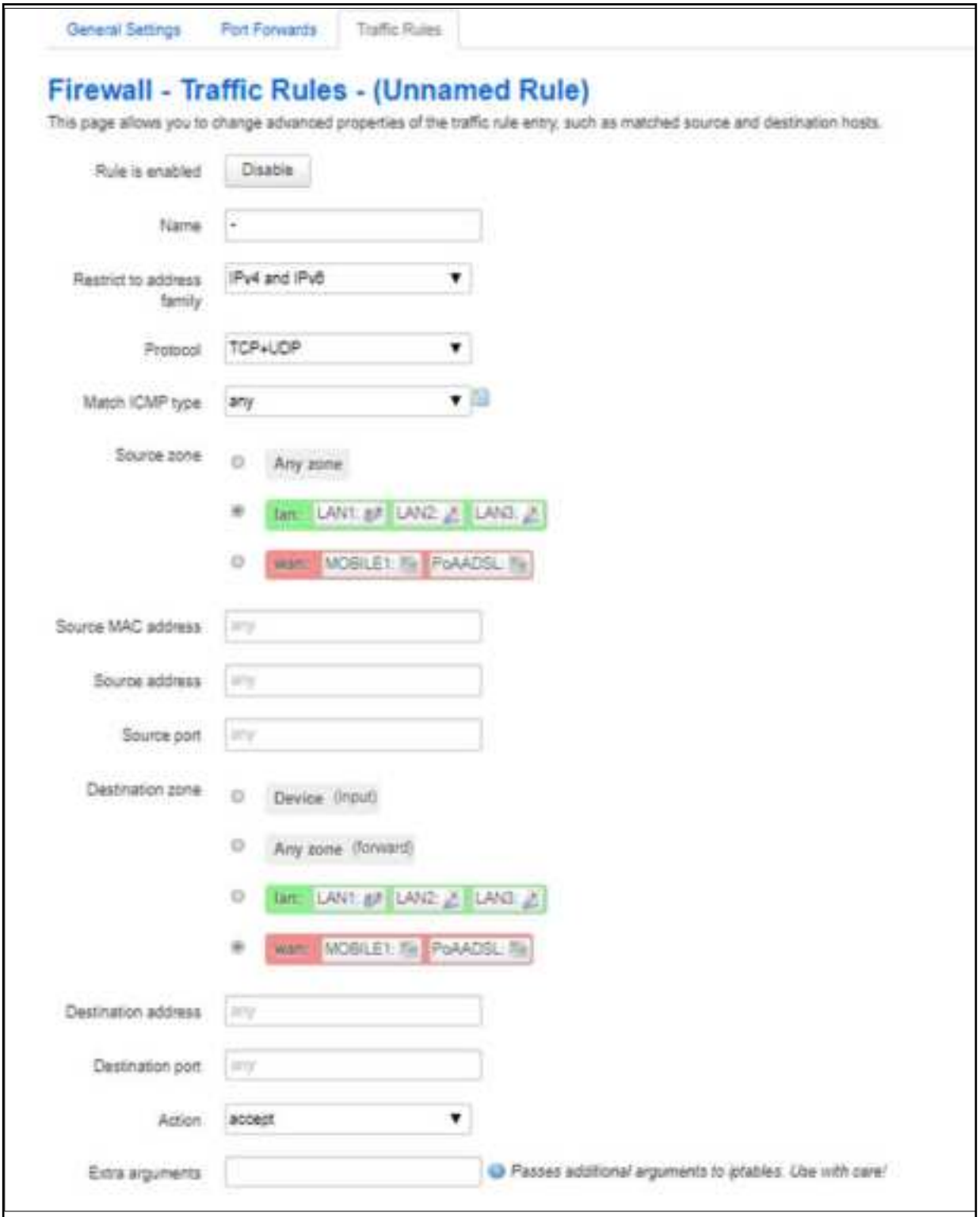
```
option src_dport '21 22 23'

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'
```

31.2.3. Firewall Traffic Rules Page

Rules can be defined to allow or restrict access to specific ports, hosts or protocols.



The firewall traffic rules page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Rule is enabled

UCI: firewall.<rule label>.enabled

Opt: enabled</td><td>Enables or disables traffic rule.

<table><tr><td>0</td><td>Rule is disabled</td></tr><tr><td>Default: 1</td><td>Rule is enabled</td></tr></table></td></tr>
<tr><td>Web: Name

UCI: firewall.<rule label>.name

Opt: name</td><td>Select a descriptive name limited to less than 11 characters. No spaces are allowed in the naming convention.</td></tr>
<tr><td>Web: Restrict to address family

UCI: firewall.<rule label>.family

Opt: family</td><td>Restrict to protocol family.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>Default: IPv4 and IPv6</td><td>Traffic rule applies to any address family</td><td>any</td></tr><tr><td>IPv4 only</td><td>IPv4 only</td><td>ipv4</td></tr><tr><td>IPv6 only</td><td>IPv6 only</td><td>ipv6</td></tr></table></td></tr>
<tr><td>Web: Protocol

UCI: firewall.<rule label>.proto

Opt: proto</td><td>Matches incoming traffic using the given protocol.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>Default: tcp+udp</td><td>Applies rule to TCP and UDP only</td><td>tcp udp</td></tr><tr><td>tcp</td><td>Applies rule to TCP only</td><td>tcp</td></tr><tr><td>udp</td><td>Applies rule to UDP only</td><td>udp</td></tr><tr><td>ICMP</td><td>Applies rule to ICMP only</td><td>icmp</td></tr><tr><td>custom</td><td>Specify protocol from /etc/protocols</td><td></td></tr></table></td></tr>
<tr><td>Web: Match ICMP type

UCI: firewall.<rule label>.icmp_type

Opt: icmp_type</td><td>Match specific icmp types.

This option is only valid when ICMP is selected as the protocol. ICMP types can be listed as either type names or type numbers.

Note: for a full list of valid ICMP type names, see the ICMP Options table below.</td></tr>
<tr><td>Web: Source zone

UCI: firewall.<rule label>.src

Opt: src</td><td>Specifies the traffic source zone, must refer to one of the defined zone names. For typical port forwards, this is usually WAN.</td></tr>
<tr><td>Web: Source MAC address

UCI: firewall.<rule label>.src_mac

Opt: src_mac</td><td>Matches incoming traffic from the specified MAC address. The MAC address must be entered in the following format: aa:bb:cc:dd:ee:ff:

To only match the first portion of the MAC address append

<code>/prefix</code> to the option value, where <code>prefix</code> defines the bits from the start of the MAC to match on.

Example:

<code>option src_mac 00:E0:C8:12:34:56/24</code>

will match on all packets with prefix <code>00:E0:C8</code>.</td></tr>
<tr><td>Web: Source address

UCI: firewall.<rule label>.src_ip

Opt: src_ip</td><td>Matches incoming traffic from the specified source IP address.</td></tr>
<tr><td>Web: Source port

UCI: firewall.<rule label>.src_port

Opt: src_port</td><td>Matches incoming traffic originating from the given source port or port range on the client host.</td></tr>
<tr><td>Web: Destination zone

UCI: firewall.<rule label>.dest Opt: dest</td><td>Specifies the traffic destination zone. Must refer to one of the defined zone names.</td></tr>
</table>

Web Field/UCI/Package Option	Description
Web: Destination address UCI: firewall.<rule label>.dest_ip Opt: dest_ip	For DNAT, redirects matched incoming traffic to the specified internal host. For SNAT, matches traffic directed at the given address.
Web: Destination port UCI: firewall.<rule label>.dest_port Opt: dest_port	For DNAT, redirects matched incoming traffic to the given port on the internal host. For SNAT, matches traffic directed at the given ports.
Web: Action UCI: firewall.<rule label>.target Opt: target	Action to take when rule is matched. Option | Description | UCI drop | Drop matching traffic | drop Default: accept | Allow matching traffic | accept reject | Reject matching traffic | reject don't track | Disable connection tracking for the rule. See the Connection Tracking section below for more information. | No Track
Web: Extra arguments UCI: firewall.<rule label>.extra Opt: extra	Passes extra arguments to IP tables. This is useful to specify additional match options, like -m policy --dir in for IPSec.
Web: n/a UCI: firewall.<rule label>.reflection Opt: reflection	Disables NAT reflection for this redirect if set to 0. Applicable to DNAT targets.
Web: n/a UCI: firewall.<rule label>.limit Opt: limit	Sets maximum average matching rate; specified as a number, with an optional / second, /minute, /hour or /day suffix. Example: 3/hour.
Web: n/a UCI: firewall.<rule label>.limit_burst Opt: limit_burst	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.
Web: n/a UCI: firewall.<rule label>.recent Opt: recent	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.

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

31.3. Configuring Firewall Using UCI

Firewall is configured under the firewall package /etc/config/firewall.

There are six config sections: defaults, zone, forwarding, redirect, rule and include. You can configure multiple zone, forwarding and redirect sections.

Firewall: General Settings

To set general (default) settings, enter:

```
uci add firewall defaults
uci set firewall.@defaults[0].syn_flood=1
uci setfirewall.@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.

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, enter:

```
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=lan1wifi_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
```

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, enter:

```
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
```

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, enter:

```
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
```

Firewall Traffic Rules

By default, all traffic rules 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, enter:

```
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 setfirewall.@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
```

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
pth	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.

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.

31.3.1. 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)

31.3.2. 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, you must enable the conntrack option in the corresponding zone to disable NOTRACK. It should appear as option 'conntrack' '1' in the right zone in /etc/config/firewall.

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.

31.3.3. Firewall Rule Examples

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.

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'
```

Source NAT (SNAT)

Source NAT changes an outgoing packet destined for the system so that it 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 pubic WAN IP address.

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 will 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
```

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
```

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
```

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 21:00 and 09:00

```
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
```

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
```

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'
```

IP Spoofing Prevention Mechanism

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'
```

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
```

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
```

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
```

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
```

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.

Firewall Management

After a configuration change, to rebuild the 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
```

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
```

32. 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. Merlin 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 your router is not limited in any way by software; the only hardware limitation is the amount of RAM installed on the device.

Configuration package used

Package	Sections
strongswan	general connection secret

32.1. 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	

32.1.1. Configure Common Settings Using The Web Interface

The common settings section

Web Field/UCI/Package Option	Description
Web: Enable strongswan UCI: strongswan.general.enable Opt: enabled	Enables or disables IPSec. 0 Disabled Default: 1 Enabled
Web: Strict CRL Policy UCI: strongswan.general.strictcrlpolicy Opt: strictcrlpolicy	Defines if a fresh CRL must be available for the peer authentication based on RSA signatures to succeed. 0 Disabled Default: 1 Enabled
Web: Unique IDs UCI: strongswan.general.uniqueids 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. Participant IDs normally are unique, so a new, automatically- keyed, connection using the same ID is almost invariably intended to replace an old one. 0 Disabled Default: 1 Enabled
Web: Cache CRLs UCI: strongswan.general.cachecrls 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. 0 Disabled Default: 1 Enabled replace Identical to Yes keep Rejects new IKE SA and keeps the duplicate established earlier
Web: Disable Revocation UCI: strongswan.general.revocation_disabled Opt: revocation_disabled	Defines whether disable CRL and OCSP checking for revoked certificates. 0 Disabled Default: 1 Enabled
Web: Send INITIAL CONTACT by default UCI: strongswan.general.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 or disabled per connection.

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Opt: initial_contact</td><td></td></tr>
<tr><td>Web: Debug

UCI: strongswan.general.debug

Opt: debug</td><td>Enables debugging. This option is used for trouble shooting issues. It is not suitable for a production environment.
<table>
<tr><td>Default: None</td><td>Debug disabled</td></tr>
<tr><td>Control</td><td>Debug enabled. Shows generic control flow with errors and very basic auditing logs.</td></tr>
<tr><td>All</td><td>Debug enabled. Most verbose logging also includes sensitive information such as keys.</td></tr>
</table></td></tr>
</table>

32.1.2. Common Settings: Configure Connection Using Web Interface

The configure connection page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: strongswan.@connection[X].enabled

Opt: enable</td><td>Enables or disables an IPSec connection.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Aggressive

UCI: strongswan.@connection[X].aggressive

Opt: aggressive</td><td>Enables or disables IKE aggressive mode.

Note: using aggressive mode along with PSK authentication is a less secure method than main mode and should be avoided.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Name

UCI: strongswan.@connection[X].name Opt: name</td><td>Specifies a name for the tunnel.</td></tr>
<tr><td>Web: Autostart Action

UCI: strongswan.@connection[X].auto

Opt: auto</td><td>Specifies when the tunnel is initiated.

<table><tr><td>start</td><td>On start up.</td></tr><tr><td>Default: route</td><td>When traffic routes this way.</td></tr><tr><td>add</td><td>Loads a connection without starting it.</td></tr><tr><td>ignore</td><td>Ignores the connection.</td></tr><tr><td>always</td><td>Actively retries to establish the tunnel if it went down.</td></tr></table></td></tr>
<tr><td>Web: Connection Type

UCI: strongswan.@connection[X].type

Opt: type</td><td>Defines the type of IPSec connection.

<table><tr><td>Default: tunnel</td><td>Connection uses tunnel mode.</td></tr><tr><td>transport</td><td>Connection uses transport mode.</td></tr><tr><td>pass</td><td>Connection does not perform any IPSec processing</td></tr><tr><td>drop</td><td>Connection drops all the packets</td></tr></table></td></tr>
</table>

32.1.3. Common Settings: IP Addressing Using The Web Interface

The IP addressing settings

Web Field/UCI/Package Option	Description
Web: Remote GW Address UCI: strongswan.@connection[X]. remoteaddress Opt: remoteaddress	Sets the public IP address of the remote peer.
Web: Local ID UCI: strongswan.@connection[X].localid Opt: localid	Defines the local peer identifier.
Web: Remote ID UCI: strongswan.@connection[X].remoteid Opt:remoteid	Defines the remote peer identifier.
Web: Local LAN IP Address UCI: strongswan.@connection[X]. locallan Opt: locallan	Defines the local IP of LAN.
Web: Local LAN IP Address Mask UCI: strongswan.@connection[X]. locallanmask Opt: locallanmask	Defines the subnet of local LAN.
Web: Remote LAN IP Address UCI: strongswan.@connection[X]. remotelan Opt:remotelan	Defines the IP address of LAN serviced by remote peer.
Web: Remote LAN IP Address Mask UCI: strongswan.@connection[X]. remotelanmask Opt:remotelanmask	Defines the Subnet of remote LAN.
Web: Local Protocol UCI: strongswan.@connection[X].localproto Opt: localproto	Restricts the connection to a single port on the local side.
Web: Local Port UCI: strongswan.@connection[X].localport Opt: localport	Restricts the connection to a single protocol on the local side.
Web: Remote Protocol UCI: strongswan.@connection[X].remoteproto Opt: remoteproto	Restricts the connection to a single protocol on the remote side.
Web: Remote Port UCI: strongswan.@connection[X].remoteport Opt: remoteport	Restricts the connection to a single port on the remote side.
Web: Authby UCI: strongswan.@connection[X].authby Opt: authby	Defines how the two secure gateways should authenticate. Note: using aggressive mode along with PSK authentication is unsecure and should be avoided. Pubkey — For public key signatures. Rsasig — For RSA digital signatures. ecdsasig — For elliptic curve DSA signatures.

Web Field/UCI/Package Option	Description
	Psk — Using a preshared key. zauthrsasig — Enables eXtended Authentication (ZAtuh) with addition to RSA signatures. zauthpsk — Using extended authentication and preshared key. never — Can be used if negotiation is never to be attempted or accepted (shunt connection)

32.2. Common Settings: IPSec Settings Using The Web Interface

The common settings: IPSec settings page

Figure 7. Common settings: IPSec settings

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

Web Field/UCI/Package Option	Description
	modp8192 For example, a valid IKE algorithm is: aes128-sha-modp1536.
Web: WAN Interface UCI: strongswan.@connection[X].waniface Opt: waniface	This is a space-separated list of the WAN interfaces the router will use to establish a tunnel with the secure gateway. On the web, a list of the interface names is automatically generated. If you want to specify more than one interface use the "custom" value. Example: if you have a 3G WAN interface called 'wan' and a WAN ADSL interface called 'dsl' and wanted to use one of these interfaces for this IPSec connection, you would use: 'wan adsl'.
Web: IKE Life Time UCI: strongswan.@connection[X].ikelifetime Opt: ikelifetime	Specifies how long the keyring channel of a connection (ISAKMP or IKE SA) should last before being renegotiated. Default: 3 hr Timespec: 1d, 1h, 25m, 10s
Web: Key Life UCI: strongswan.@connection[X].keylife Opt: keylife	Specifies how long a particular instance of a connection (a set of encryption/authentication keys for user packets) should last, from successful negotiation to expiry. Normally, the connection is renegotiated (via the keying channel) before it expires (see rekeymargin). Default: 1h Timespec: 1d, 1h, 25m, 10s
Web: Rekey Margin UCI: strongswan.@connection[X].rekeymargin Opt: rekeymargin	Specifies how long before connection expiry or keying- channel expiry should attempt to negotiate a replacement begin. Relevant only locally, other end need not agree on it. Default: 9m Timespec: 1d, 2h, 9m, 10s
Web: Restart Delay UCI: strongswan.@connection[X].restartdelay Opt: restartdelay	Defines specific delay when re-establishing a connection. Previously if `close_action=restart`, then the new option `restartdelay` controls how many seconds it waits before attempting to re-establish the tunnel to allow the headend some time to tidy up. If not set, it defaults to zero, which means that the previous behaviour of choosing a random time interval in the range `0..RekeyMargin` seconds takes effect. Relevant only locally, other end need not agree on it. Default: 0 Timespec: 1d, 2h, 9m, 10s
Web: Keying Tries UCI: strongswan.@connection[X].keyringtries Opt: keyringtries	Specifies how many attempts, for example, 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, the other end need not agree on it.
Web: DPD Action UCI: strongswan.@connection[X].dpdaction Opt: dpdaction	Defines DPD (Dead Peer Detection) action. Default: None — Disables DPD. Clear — Clear down the tunnel if peer does not respond. Reconnect when traffic brings the tunnel up. Hold — Clear down the tunnel and bring up as soon as the peer is available. Restart — Restarts DPD when no activity is detected.
Web: DPD Delay UCI: strongswan.@connection[X].dpddelay Opt: dpddelay	Defines the period time interval with which R_U_THERE messages and INFORMATIONAL exchanges are sent to the peer. These are only sent if no other traffic is received. Default: 30s Timespec: 1d, 2h, 25m, 10s
Web: DPD Timeout UCI: strongswan.@connection[X].dpdtimeout Opt: dpdtimeout	Defines the timeout interval, after which all connections to a peer are deleted in case of inactivity. Default: 150s Timespec: 1d, 2h, 25m, 10s
Web: Inherit CHILD SA UCI:strongswan.@connection[X].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.

Web Field/UCI/Package Option	Description
Opt: inherit_child	Default: 0 — Delete the existing IPCES SA on IKE rekey 1 — Maintain the existing IPSEC SA on IKE rekey
Web: Send INITIAL CONTACT UCI: strongswan.@connection[X].initial_contact Opt: initial_contact	Defines whether the first attempt to contact a remote peer by this strongswan instance sets the initial_contact flag which should cause compliant peers to automatically bring down any previous sessions. Default: 0 — Do not set initial contact flag 1 — Set initital contact flag on first attempt.

32.3. Configure Secret Settings Using The Web Interface

Each tunnel requires settings to configure how the local end point of the tunnel proves its identity to the remote end point.

The IPSec secret settings section

Figure 8. Configure secret settings

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

32.4. 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	

32.5. Configuring IPSec Using UCI

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
```

Connection Settings

NOTE
Xauth is not supported in IKEv2.

```
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'
```

32.6. Shunt Connection Using UCI

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.

32.7. Secret Settings Using UCI

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'
```

32.8. Configuring An IPSec Template To Use With DMVPN Using UCI

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'
```

32.9. IPSec Diagnostics

IPSec Status using the Web Interface

In the top menu, click **Status -> IPSec**. The IPSec Connections page appears.

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.

IPSec Diagnostics using UCI

Viewing 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
```

IPSec Status

```
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
```

33. Configuring SCEP (Simple Certificate Enrolment Protocol)

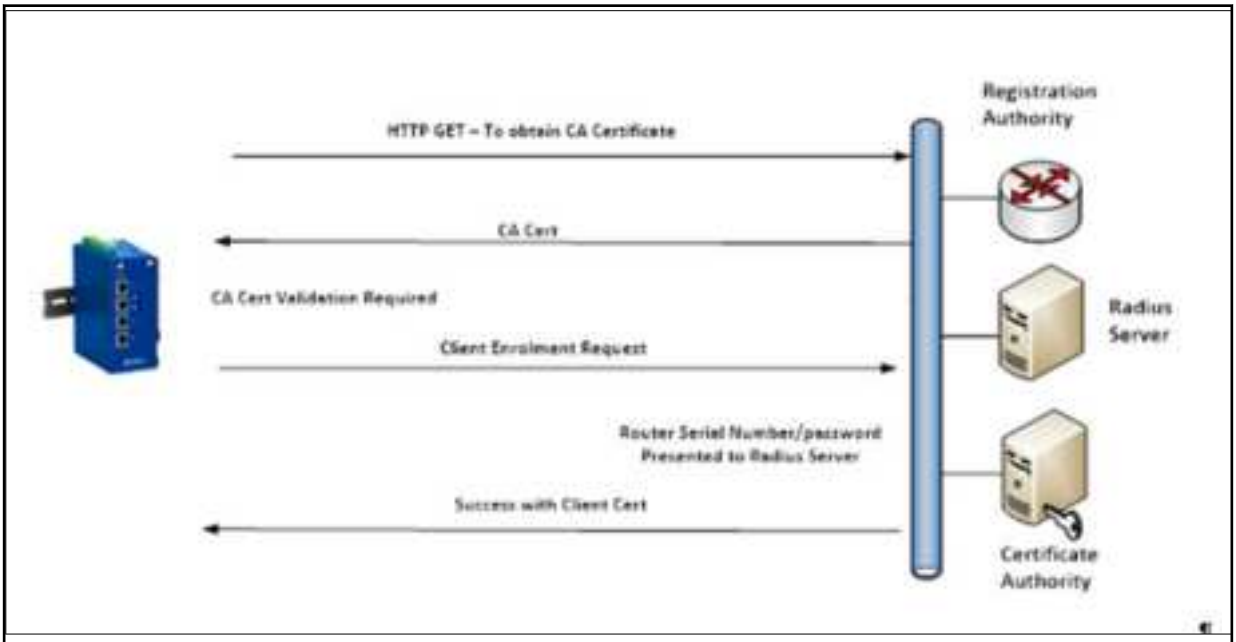
SCEP is a method for automatically obtaining x.509 certificates for IPSec validation. This protocol is commonly used in a Private Key Infrastructure (PKI).

The SCEP method has the following steps:

- Obtain a copy of the Certificate Authority (CA) certificate and validate it.
- Generate a Certificate Signing Request (CSR) and send it securely to the CA.
- Re-enrol as necessary to obtain a new certificate prior to the expiration of the current certificate.

This section only details the SCEP portion of an IPSec configuration. For more

information on configuring general IPSec, read the chapter 'Configuring IPSec'.



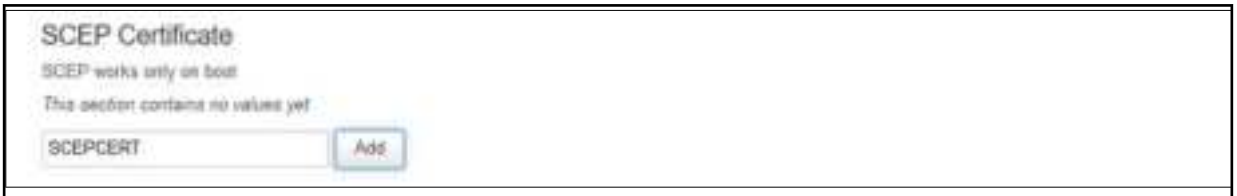
The SCEP process between router and PKI infrastructure

Configuration package used

Package	Sections
strongswan	scep_cert

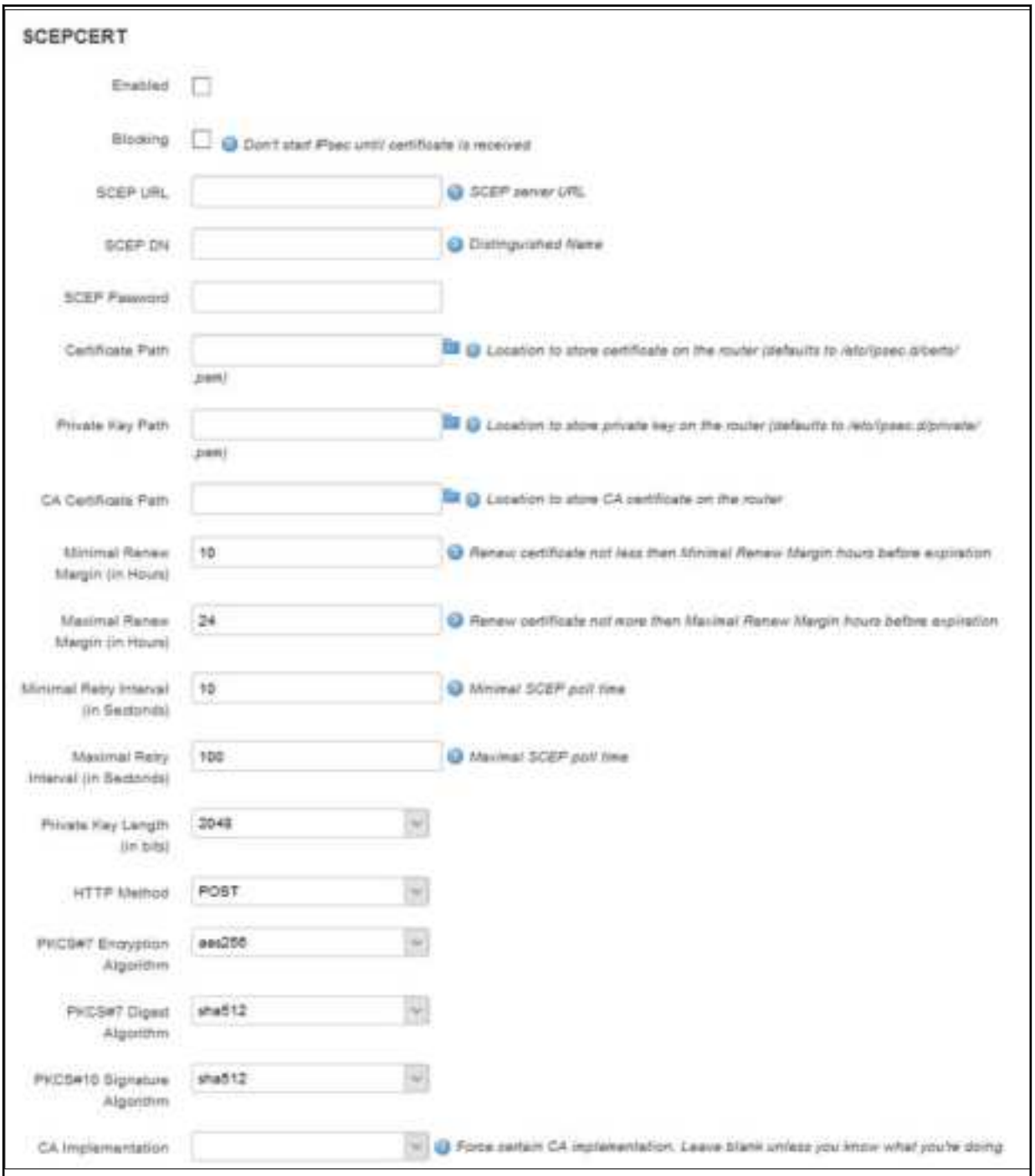
33.1. Configuring SCEP Using The Web Interface

To define an automatically enrolled certificate, using SCEP, select **Services -> IPSec**. Scroll down to the SCEP Certificate section. Enter a name for the SCEP section and select **Add**.



SCEP certificate name section

The SCEP certificate configuration section options appear.



The SCEP certificate section options

Web Field/UCI/Package Option	Description
Web: Enabled UCI: strongswan.@scep_cert[0].enabled Opt: enabled	Defines whether SCEP automatic enrolment is enabled. 0 – Disabled Default: 1 – Enabled
Web: Blocking UCI: strongswan.@scep_cert[0].blocking Opt: blocking	Defines whether to wait until the certificate is received before starting IPSec. Default: 0 – Wait until the certificate is received before starting IPSec. 1 – Do not wait until the certificate is received.
Web: SCEP URL UCI: strongswan.@scep_cert[0].url Opt: url	Defines the URL for the SCEP server.
Web: SCEP DN UCI: strongswan.@scep_cert[0].dn Opt: dn	Defines the Distinguished Name to use for new certificate. Note: substring `%serial`will be replaced with a router's serial number.
Web: SCEP Password UCI: strongswan.@scep_cert[0].scep_psk Opt: scep_psk	Defines a SCEP password.
Web: Certificate Path UCI: strongswan.@scep_cert[0].cert_path Opt: cert_path	Defines the filepath to store the certificate on the router (absolute of relative). Default: Empty – /etc/ipsec.d/certs/.pem
Web: Private Key Path UCI: strongswan.@scep_cert[0].key_path Opt: key_path	Defines the filepath to store the private key on the router (absolute of relative). Default: Empty – /etc/ipsec.d/certs/.pem
Web: CA Certificate Path UCI: strongswan.@scep_cert[0].cacert Opt: cacert	Defines the filepath to store the CA certificate on the router (absolute of relative). Default: Empty – /etc/ipsec.d/certs/.pem
Web: Minimal Renewal Margin (Hours) UCI: strongswan.@scep_cert[0].minmargin_hrs Opt: minmargin_hrs	Defines the minimum duration, in hours, from certificate expiration for renewal of certificate. Note: a random value between minimal and maximal renewal margin will be used. Default: – 10 hours
Web: Maximal Renewal Margin (Hours) UCI: strongswan.@scep_cert[0].maxmargin_hrs Opt: maxmargin_hrs	Defines the maximum duration, in hours, from certificate expiration for renewal of certificate. Note: the retry interval will be set to a random value between minimal and maximal renewal margin. Default: – 24 hours
Web: Minimal Retry Interval (Seconds) UCI: strongswan.@scep_cert[0].minretry Opt: minretry	Defines the minimal poll time, in seconds. Note: the retry interval will be set to a random value between minimal and maximal renewal margin. The retry interval is used when the server replies with PENDING status for initial request (also called manual mode). Default: – 10 hours
Web: Maximal Retry Interval (Seconds) UCI: strongswan.@scep_cert[0].maxretry	Defines the maximal poll time, in seconds.

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Opt: maxretry</td><td>Note: the retry interval will be set to a random value between minimal and maximal renewal margin. The retry interval is used when the server replies with PENDING status for initial request (also called manual mode).

<table><tr><td>Default:</td><td>100 seconds</td></tr></table></td></tr>
<tr><td>Web: Private Key Length (in bits)

UCI: strongswan.@scep_cert[0].key_len

Opt: key_len</td><td>Defines the private key length.

<table><tr><td>Default: 2048</td><td>2048 bits</td></tr><tr><td>4096</td><td>4096 bits</td></tr><tr><td>6144</td><td>6144 bits</td></tr><tr><td>8192</td><td>8192 bits</td></tr><tr><td>--custom--</td><td>Defines custom length</td></tr></table></td></tr>
<tr><td>Web: HTTP Method

UCI: strongswan.@scep_cert[0].method

Opt: method</td><td>Defines the HTTP method used for client enrolment.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>Default: GET</td><td>HTTP GET</td><td>get</td></tr><tr><td>POST</td><td>HTTP POST</td><td>post</td></tr></table></td></tr>
<tr><td>Web: PKCS#7 Encryption Algorithm

UCI: strongswan.@scep_cert[0].pkcs7_enc_algo

Opt: pkcs7_enc_algo</td><td>Defines the symetric encryption algorithm to use.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>Default: aes256</td><td></td><td>aes256</td></tr><tr><td>aes192</td><td></td><td>aes192</td></tr><tr><td>aes128</td><td></td><td>aes128</td></tr><tr><td>3des</td><td></td><td>3des</td></tr></table></td></tr>
<tr><td>Web: PKCS#7 Digest Algorithm

UCI: strongswan.@scep_cert[0].pkcs7_dgst_algo

Opt: pkcs7_dgst_algo</td><td>Defines the hash algorithm for pkcs7 digest calculation.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>Default: sha512</td><td></td><td>sha512</td></tr><tr><td>sha384</td><td></td><td>sha284</td></tr><tr><td>sha256</td><td></td><td>sha256</td></tr><tr><td>sha1</td><td></td><td>sha1</td></tr><tr><td>md5</td><td></td><td>md5</td></tr></table></td></tr>
<tr><td>Web: PKCS#7 Signature Algorithm

UCI: strongswan.@scep_cert[0].pkcs10_sig_algo

Opt: pkcs10_sig_algo</td><td>Defines the hash algorithm for pkcs10 signature.

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>Default: sha512</td><td></td><td>sha512</td></tr><tr><td>sha384</td><td></td><td>sha284</td></tr><tr><td>sha256</td><td></td><td>sha256</td></tr><tr><td>sha1</td><td></td><td>sha1</td></tr><tr><td>md5</td><td></td><td>md5</td></tr></table></td></tr>
<tr><td>Web: CA Implementation

UCI: strongswan.@scep_cert[0].caimpl

Opt: caimpi</td><td>Defines the SCEP server implementation

<table><tr><th>Web</th><th>Description</th><th>UCI</th></tr><tr><td>Default: Empty</td><td>Automatically deducted from URL.</td><td></td></tr><tr><td>Microsoft CA</td><td>Mircosoft CA</td><td>ms</td></tr><tr><td>EJB CA</td><td>Enterprise Java Beans Certificate Authority</td><td>ejbca</td></tr></table></td></tr>
</table>

33.2. Configuring SCEP Certificate Using The Command Line

SCEP is configured using the **scep_cert** configuration section in the strongswan package

/etc/config/strongswan.

You can configure multiple SCEP configuration sections.

By default, all SCEP certificate instances are named 'scep_cert'. The SCEP certificate instance is identified by @scep_cert then the SCEP certificate position in the package as a number. For example, for the first SCEP certificate in the package using UCI, enter:

```
strongswan.@scep_cert[0]=scep_cert
strongswan.@scep_cert[0].enabled=1
```

Or using package options, enter:

```
config scep_cert
option enabled '1'
```

However, to better identify it, we recommend giving the SCEP certificate instance a name. For example, a SCEP certificate named 'SCEPCERT' will be strongswan.SCEPCERT.

To define a named SCEP certificate instance using UCI, enter:

```
strongswan.SCEPCERT=scep_cert
strongswan.SCEPCERT.enabled=1
```

To define a named SCEP certificate instance using package options, enter:

```
config scep_cert 'SCEPCERT'
option 'enabled' '1'
```

SCEP certificate using UCI

```
root@VA_router:~# uci show strongswan
package strongswan
::::
strongswan.SCEPCERT=scep_cert
strongswan.SCEPCERT.enabled=1
strongswan.SCEPCERT.url=url
strongswan.SCEPCERT.dn=dn
strongswan.SCEPCERT.scep_psk=password
strongswan.SCEPCERT.cert_path=/etc/ipsec.d/certs/
strongswan.SCEPCERT.key_path=/etc/ipsec.d/private/
strongswan.SCEPCERT.cacert=/etc/ipsec.d/cacerts/
strongswan.SCEPCERT.minmargin_hrs=10
strongswan.SCEPCERT.maxmargin_hrs=240
strongswan.SCEPCERT.minretry=10
strongswan.SCEPCERT.maxretry=100
strongswan.SCEPCERT.key_len=2048
strongswan.SCEPCERT.method=get
strongswan.SCEPCERT.pkcs7_enc_algo=aes256
strongswan.SCEPCERT.pkcs7_dgst_algo=sha512
strongswan.SCEPCERT.pkcs10_sig_algo=sha512
strongswan.SCEPCERT.caimpl=ms
```

SCEP certificate using package options

```
root@VA_router:~# uci export strongswan
package strongswan
####
config scep_cert 'SCEPCERT'
option enabled '1'
option url 'url' option dn 'dn'
option scep_psk 'password'
option cert_path '/etc/ipsec.d/certs/'
option key_path '/etc/ipsec.d/private/'
option cacert '/etc/ipsec.d/cacerts/'
option minmargin_hrs '10'
option maxmargin_hrs '240'
option minretry '10'
option maxretry '100'
option key_len '2048' option method 'get'
option pkcs7_enc_algo 'aes256'
option pkcs7_dgst_algo 'sha512'
option pkcs10_sig_algo 'sha512'
option caimpl 'ms'
```

33.3. SCEP Diagnostics

Syslog

SCEP certificate status can be monitored via the system log. An example of SCEP syslog messages is shown below.

```
Aug 14 04:51:01 user.notice 00E0C81604BE ipsec: ca cert

'/etc/ipsec.d/cacerts/vaebjtest' expired or not yet downloaded

Aug 14 04:51:01 authpriv.info 00E0C81604BE scepclient[9146]: loaded plugins: curl aes des sha1 sha2 md5 random x509 pkcs1 pkcs7 pem openssl gmp

Aug 14 04:51:01 authpriv.info 00E0C81604BE scepclient[9146]: building CRED_CONTAINER - PKCS7 failed, tried 2 builders

Aug 14 04:51:01 authpriv.info 00E0C81604BE scepclient[9146]: unable to parse PKCS#7, assuming plain CA cert

Aug 14 04:51:01 authpriv.info 00E0C81604BE scepclient[9146]: written ca cert file '/etc/ipsec.d/cacerts/vaebjtest' (1200 bytes)

Aug 14 04:51:01 authpriv.info 00E0C81604BE ipsec_starter[9172]: Starting strongSwan 5.0.2 IPsec [starter]...

Aug 14 04:51:01 daemon.info 00E0C81604BE ipsec: 00[DMN] Starting IKE charon daemon (strongSwan 5.0.2, Linux 3.18.11, mips)

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loading ca certificates from '/etc/ipsec.d/cacerts'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loaded ca certificate "CN=VAejbcaTestCA, O=VA, C=IE" from '/etc/ipsec.d/cacerts/vaebjtest'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loading aa certificates from '/etc/ipsec.d/aacerts'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loading ocsp signer certificates from '/etc/ipsec.d/ocspcerts'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loading attribute certificates from '/etc/ipsec.d/acerts'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loading crls from '/etc/ipsec.d/crls'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loading secrets from '/etc/ipsec.secrets'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loading secrets from '/var/conf/ipsec.secrets'

Aug 14 04:51:02 daemon.info 00E0C81604BE ipsec: 00[CFG] loaded RSA private key from '/etc/ipsec.d/private/ejb_cert.pem'
```

33.4. Strongswan Process Using UCI

The strongswan process has its own subset of commands.

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

Syntax: /etc/init.d/dsl_control [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
```

To restart strongswan, enter:

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

34. 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.

Prerequisites for Configuring DMVPN

Before configuring DMVPN, you must first configure:

- A GRE interface; read the previous chapter, 'Configuring GRE interfaces'.
- An IPSec connection to use as a template. Read the previous chapter 'Configuring IPSec'.

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 peer's 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 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.

34.1. Configuring DMVPN Using The Web Interface

Configuration Packages Used

Package	Sections
network	For configuring GRE tunnels.
strongswan	For enabling and configuring the IPSec connection template
dmvpn	

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.

There are two sections in the DMVPN page: General and DMVPN Hub Settings.

DMVPN General Settings

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

The DMVPN general section

Web Field/UCI/Package Option	Description
Web: Enable DMVPN UCI: dmvpn.common.enabled Opt: enable	Enables DMVPN. Default: 0 — Disabled 1md5 — Enabled
Web: IPSec template connection UCI: dmvpn.common.ipsec_template_name Opt: ipsec_template_name	Selects the IPSec connection, defined in strongSwan, to be used as a template.

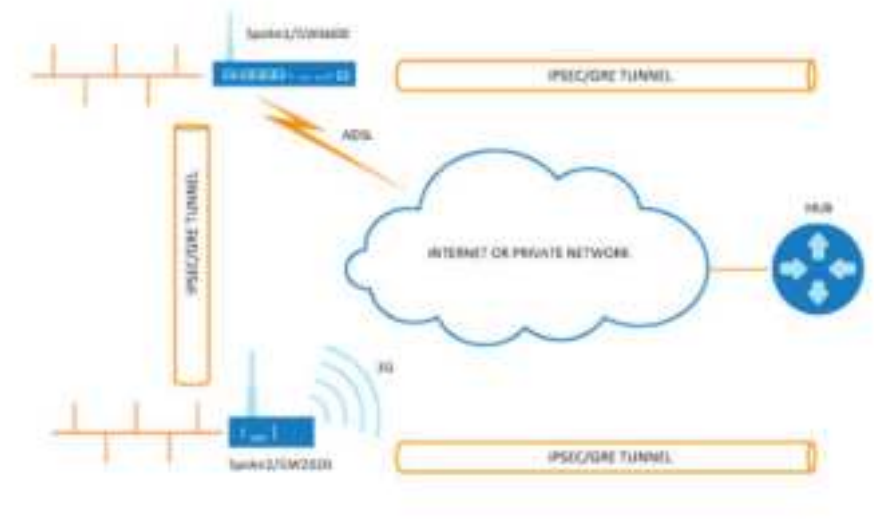
DMVPN Hub Settings

The DMVPN hub settings section

Web Field/UCI/Option	Description
Web: GRE Interface UCI: dmvpn.@interface[X].gre_interface Opt: gre_interface	Specifies which GRE interface will be used with this DMVPN configuration.
Web: GRE Remote Endpoint IP Address UCI: dmvpn.@interface[X].gre_endpoint_ip Opt: gre_endpoint_ip	Configures the GRE IP address of the hub.
Web: GRE Remote Endpoint Mask Length UCI: dmvpn.@interface[X].gre_endpoint_mask_length Opt: gre_endpoint_mask_length	Configures the length of the mask of the GRE interface on the hub. For example, if the mask is 255.255.0.0 the length will be 16.
Web: DMVPN Hub IP Address UCI: dmvpn.@interface[X].nhs_ip Opt: nhs_ip	Configures the physical IP address for the DMVPN hub.
Web: NHRP Authentication UCI: dmvpn.@interface[X].cisco_auth Opt: cisco_auth	Enables authentication on NHRP. The password will be applied in plaintext to the outgoing NHRP packets. Maximum length is 8 characters.
Web: NHRP Holding Time UCI: dmvpn.@interface[X].holding_time Opt: holding_time	Timeout for cached NHRP requests.
Web: Use as Default Route UCI: dmvpn.@interface[X].defaultroute Opt: defaultroute	Adds a default route into tunnel interface. incomplete: Resolution request sent. 1: Negative cached.
Web: Default Route Metric UCI: dmvpn.@interface[X].defaultroutemetric Opt: defaultroutemetric	Metric to use for the default route.
Web: LED state indication UCI: dmvpn.@interface[X].led Opt: led	LED to use for indicating if the VPN is up.

34.2. DMVPN Scenarios

Scenario 1



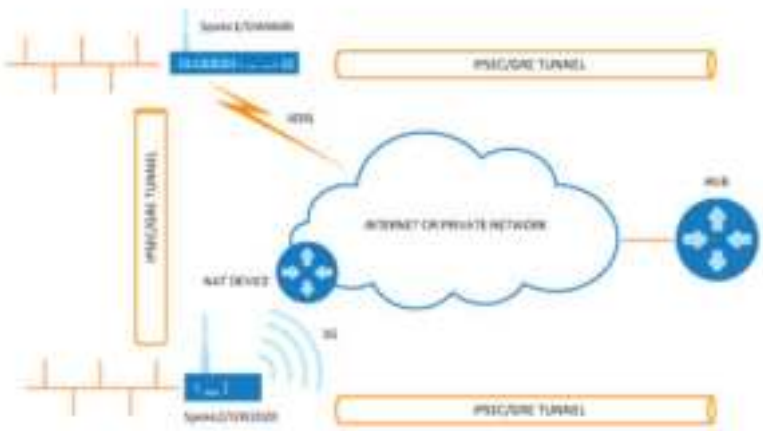
Network diagram for DMVPN spoke to spoke

Spoke1, spoke2 and a hub are in the same public or private network.

- 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, spoke 1 and spoke 2 can send traffic directly to each other.

Scenario 2

Spoke1 is in a private (NAT-ed) network, spoke2 and hub are in public network.



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.

34.3. Configuring DMVPN Using The Web Interface

Configuration Packages Used

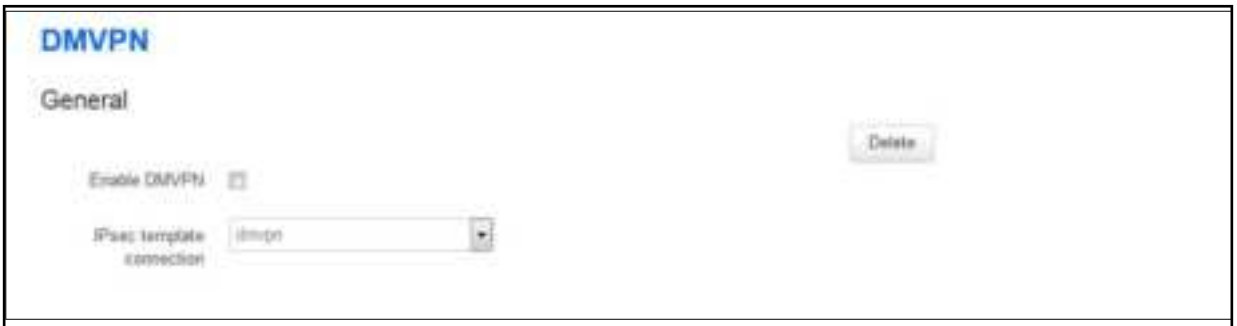
Package	Sections
network	For configuring GRE tunnels.
strongswan	For enabling and configuring the IPSec connection template
dmvpn	

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.

There are two sections in the DMVPN page: General and DMVPN Hub Settings.

DMVPN General Settings

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



The DMVPN general section

Web Field/UCI/Package Option	Description
Web: Enable DMVPN UCI: dmvpn.common.enabled Opt: enable	Enables DMVPN. Default: 0 — Disabled 1md5 — Enabled
Web: IPSec template connection UCI: dmvpn.common.ipsec_template_name Opt: ipsec_template_name	Selects the IPSec connection, defined in strongSwan, to be used as a template.

DMVPN Hub Settings

The DMVPN hub settings section

Web Field/UCI/Option	Description
Web: GRE Interface UCI: dmvpn.@interface[X].gre_interface Opt: gre_interface	Specifies which GRE interface will be used with this DMVPN configuration.
Web: GRE Remote Endpoint IP Address UCI: dmvpn.@interface[X].gre_endpoint_ip Opt: gre_endpoint_ip	Configures the GRE IP address of the hub.
Web: GRE Remote Endpoint Mask Length UCI: dmvpn.@interface[X].gre_endpoint_mask_length Opt: gre_endpoint_mask_length	Configures the length of the mask of the GRE interface on the hub. For example, if the mask is 255.255.0.0 the length will be 16.
Web: DMVPN Hub IP Address UCI: dmvpn.@interface[X].nhs_ip Opt: nhs_ip	Configures the physical IP address for the DMVPN hub.
Web: NHRP Authentication UCI: dmvpn.@interface[X].cisco_auth Opt: cisco_auth	Enables authentication on NHRP. The password will be applied in plaintext to the outgoing NHRP packets. Maximum length is 8 characters.
Web: NHRP Holding Time UCI: dmvpn.@interface[X].holding_time Opt: holding_time	Timeout for cached NHRP requests.
Web: Use as Default Route UCI: dmvpn.@interface[X].defaultroute Opt: defaultroute	Adds a default route into tunnel interface. incomplete: Resolution request sent. 1: Negative cached.
Web: Default Route Metric UCI: dmvpn.@interface[X].defaultroutemetric Opt: defaultroutemetric	Metric to use for the default route.
Web: LED state indication UCI: dmvpn.@interface[X].led Opt: led	LED to use for indicating if the VPN is up.

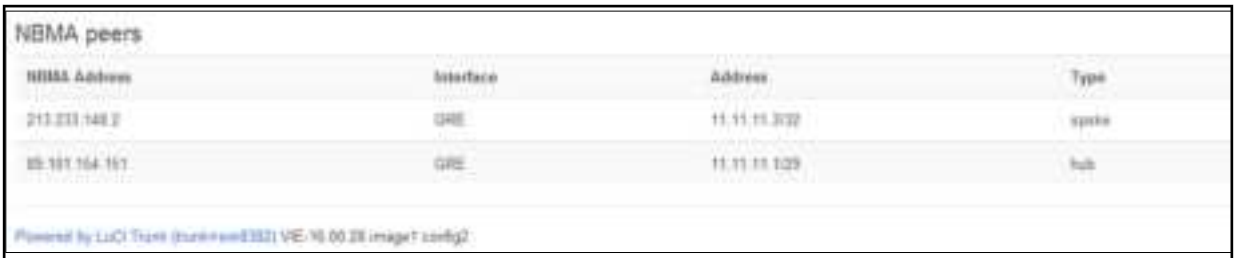
34.4. DMVPN Diagnostics

In the top menu, click **Status -> IPSec**. The IPSec Connections page appears.

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**.



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.

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_id874dc90_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
```

35. Configuring QoS: VLAN PCP Tagging

Merlin 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 forspecific 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 Options	Description
UCI: network.<if name>.vlan_qos_map_ingress Opt: list vlan_qos_map_ingress	VLAN priority code point to socket buffer mapping. Example: network.<if name>. vlan_qos_map_ingress =1:1
UCI: network.<if name>.vlan_qos_map_egress Opt: list vlan_qos_map_egress	Socket buffer to VLAN priority code point mapping. Example: network.<if name>. vlan_qos_map_egress =0:1

The above sample configuration specifies that any frames on VLAN2, VLAN3 and VLAN4 will be processed or have their PCP value adjusted according to QoS values set.

VLAN1 VLAN1 is an untagged VLAN so there are no 802.1Q tags on the frames.

VLAN2 Any frames received on VLAN2 destined to VLAN2 with PCP priority of 1 will be forwarded without altering the priority; it will be still set to 1. Any frames received on VLAN2 destined to VLAN2 with a PCP priority set to 0 will have a priority of 1 set as they leave the router on VLAN2.

VLAN3 Any frames received on VLAN3 destined to VLAN3 with a PCP priority of 3 will be forwarded without altering the priority; it will be still set to 3. Any frames received on VLAN3 destined to VLAN2 with PCP priority set to 0 will have a priority of 3 set as they leave the router on VLAN3.

VLAN4 Any frames received on VLAN4 destined to VLAN2 with PCP priority of 5 will be forwarded without altering the priority; it will be still set to 5. Any frames received on VLAN4 destined to VLAN2 with PCP priority set to 0 will have a priority of 5 set as they leave the router on VLAN4.

Four queues are supported and are structured as follows: · Queue 1: PCP values 0 and 1 - Default · Queue 2: PCP values 2 and 3 - Normal · Queue 3: PCP values 4 and 5 - High · Queue 4: PCP values 6 and 7 - Express

Value 7 is the highest priority and 0 is the lowest. These queues prioritise 802.1Q tagged frames as they are received on the port, these are hardware defined.

When 802.1Q frames are received on the port they are processed according to the above queues on arrival, even if not defined in the configuration. Then if value '`vlan_qos_map_ingress`' is configured you can modify the PCP priority for egress if the frame was to be forwarded on another tagged interface.

When frames are received on an untagged VLAN interface configured with '`vlan_qos_map_egress`' and are destined to tagged interface, 802.1Q tag will be created with a default priority of 0 and then the priority will be set according to the PCP value specified as the frames leave port.

36. Management Configuration Settings

This chapter contains the configuration sections and parameters required to manage and monitor your device using Activator and Monitor.

Activator

Activator is a 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 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.

Monitor

Monitor is a proprietary tool, based on SNMP protocol, to monitor wide networks of deployed routers. The router is 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.

Configuration Packages Used

Package	Sections
autoload	main
httpclient	default
management_users	user

36.1. 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).

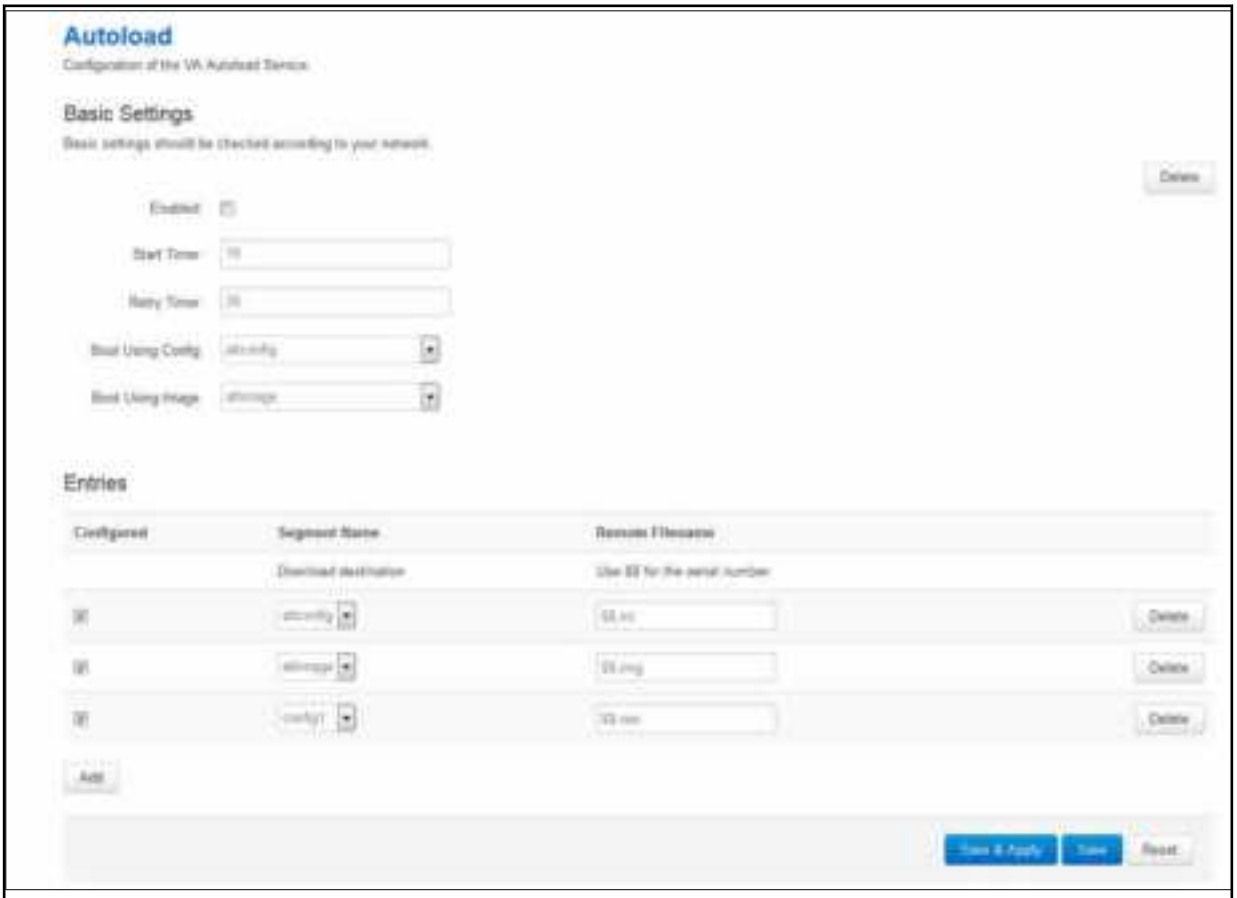
Autoload may already have been set at factory config level. If you wish to enable autoload services, proceed through the following steps.

Autoload Packages

Package	Sections
autoload	main

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.



The autoload settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td colspan="2">Basic Settings</td></tr>
<tr><td>Web: Enabled

UCI: autoload.main.enabled

Opt: Enabled</td><td>Enables activation at system boot.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Start Timer

UCI: autoload.main.StartTimer

Opt: StartTimer</td><td>Defines how long to wait after the boot up completes before starting activation.

<table><tr><td>Default:</td><td>10</td></tr><tr><td>Range</td><td>0-300 secs</td></tr></table></td></tr>
<tr><td>Web: Retry Timer

UCI: autoload.main.RetryTimer

Opt: RetryTimer</td><td>Defines how many seconds to wait between retries if a download of a particular autoload entry fails.

<table><tr><td>Default:</td><td>10</td></tr><tr><td>Range</td><td>0-300 Secs</td></tr></table></td></tr>
<tr><td>Web: N/A

UCI: autoload.main.NumberOfRetries

Opt: Numberofretries</td><td>Defines how many retries to attempt before failing the overall activation sequence, backing off and trying the whole activation sequence again.

<table><tr><td>Default:</td><td>5</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: N/A

UCI: autoload.main.BackoffTimer

Opt: Backofftimer</td><td>Defines how many minutes to back off for if a download and all retires fail. After the backoff period, the entire autoload sequence will start again.

<table><tr><td>Default:</td><td>15</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Boot Using Config

UCI: autoload.main.BootUsingConfig

Opt: BootUsingConfig</td><td>Specifies which configuration to boot up with after the activation sequence.

<table><tr><td>Default: 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

UCI: autoload.main.BootUsingImage

Opt: BootUsingImage</td><td>Specifies which image to boot up with after the activation sequence completes successfully.

<table><tr><td>Default: 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

UCI: autoload.@entry[x].Configured

Opt: Configured</td><td>Enables the autoload sequence to process this entry.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Segment Name

UCI: autoload.@entry[x].SegmentName

Opt: SegmentName</td><td>Defines where the downloaded file should be stored:

(config1 | config2 | altconfig | image1 | image2| altimage).

Typically only altconfig and altimage are used.</td></tr>
<tr><td>Web: RemoteFilename

UCI: autoload.@entry[x].RemoteFilename

Opt: RemoteFilename</td><td>Defines the name of the file to be downloaded from Activator.

<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>

36.2. 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"
```

36.3. 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.

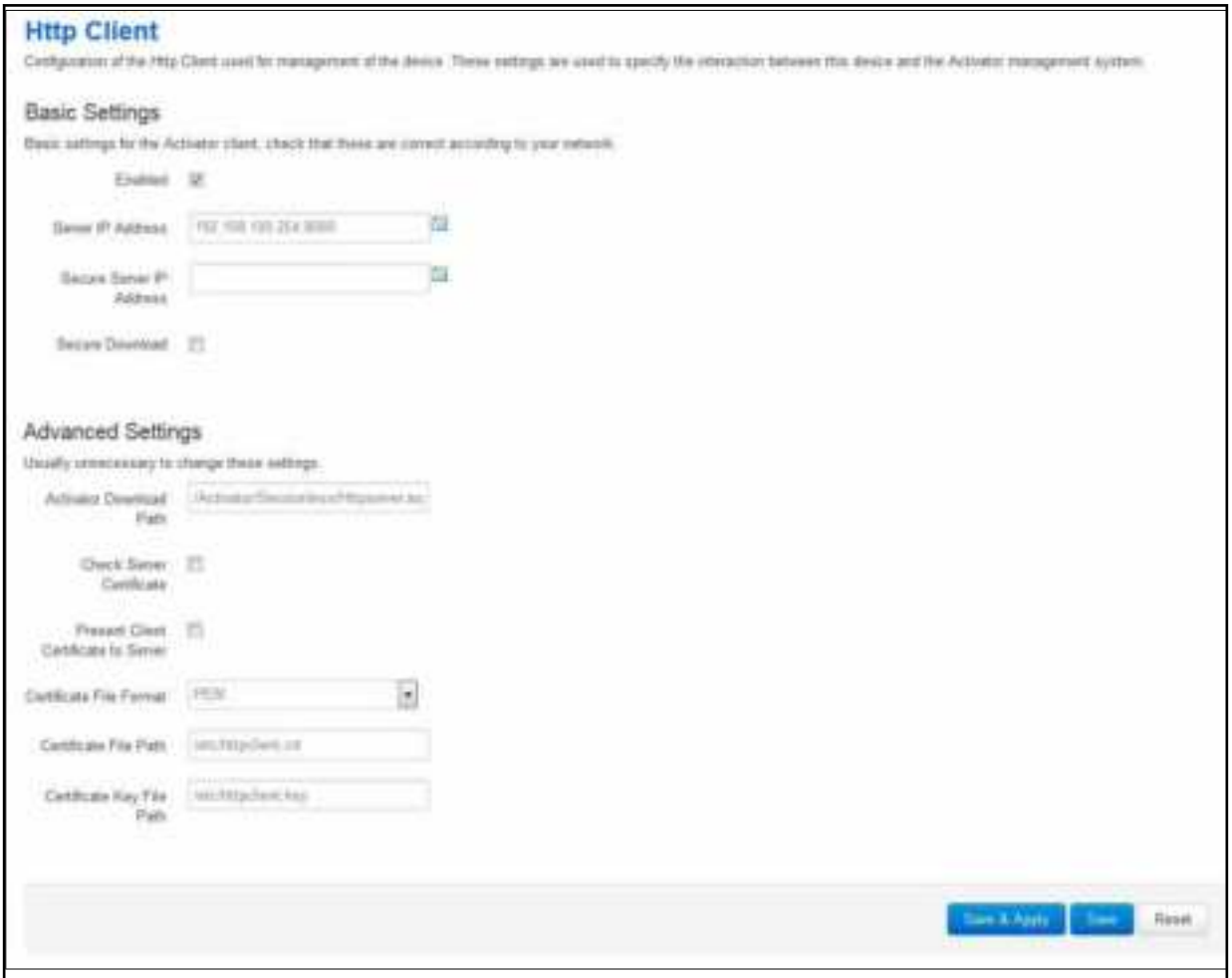
The httpclient core section configures the basic functionality of the module used for retrieving files from Activator during the activation process.

HTTP Client Configuration Packages

Package	Sections
Httpclient	default

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.



The HTTP client page

Web Field/UCI/ Package Option	Description
Basic Settings	
Web: Enabled UCI: httpclient.default.enabled Opt: Enabled	Enables the HTTP client. Default: 0 — Disabled 1 — Enabled
Web: Server IP Address UCI: httpclient.default.Fileserver Opt: list Fileserver	Specifies the address of Activator that uses http port 80. This can be an IP address or FQDN. The syntax should be x.x.x.x:80 or FQDN:80. Multiple servers should be separated by a space using UCI.
Web: Secure Server IP Address UCI: httpclient.default.SecureFileServer Opt: list SecureFileServer	Specifies the address of Secure Activator that uses port 443. This can be an IP address or FQDN. The syntax should be x.x.x.x:443 or FQDN:443. Multiple servers should be separated by a space using UCI.
Web: Secure Download UCI: httpclient.default.SecureDownload Opt: SecureDownload	Enables Secure Download (port 443). Default: 0 — Disabled 1 — Enabled
Advanced Settings	
Web: Activator Download Path UCI: httpclient.default.ActivatorDownloadPath Opt: ActivatorDownloadPath	Specifies the URL on Activator to which the client should send requests. Default: /Activator/Sessionless/Httpserver.asp
Web: Check Server Certificate UCI: httpclient.default.ValidateServerCertificateEnabled Opt: ValidateServerCertificateEnabled	Checks for the certificate's presence and validity. Default: 0 — Disabled 1 — Enabled
Web: Present Client Certificate to Server UCI: httpclient.default.ValidateServerCertificateEnabled	Specifies if the client presents its certificate to the server to identify itself. Default: 0 — Disabled 1 — Enabled

Web Field/UCI/ Package Option	Description
Opt: ValidateServerCertificateEnabled	
Web: Certificate File Format UCI: httpclient.default.CertificateFormat Opt: CertificateFormat	Specifies the value the client expects to see in the specified field in the server certificate. Default: PEM DER
Web: Certificate File Path UCI: httpclient.default.CertificateFile Opt: CertificateFile	Defines the directory/location of the certificate. Default: /etc/httpclient.crt
Web: Certificate Key File Path UCI: httpclient.default.CertificateKey Opt: CertificateKey	Specifies the directory/location of the certificate key. Default: /etc/httpclient.key
Web: N/A UCI: httpclient.default.ActivatorChunkyDownloadPath Opt: ActivatorChunkyDownloadPath	Enables partial download activations and active updates. The default value is: `httpclient.default.ActivatorChunkyDownloadPath=/activator/partial/download` The URL, on Activator, to which the client should send requests for chunky image download.
Web: N/A UCI: httpclient.default.ChunkSize Opt: ChunkSize	Specifies the size of each packet payload. Default: 100k bytes 1-infinite Available values in kbps
Web: N/A UCI: httpclient.default.RateLimit Opt: RateLimit	Throttle activation/active updates traffic received by device to specified limit. Default: None By default, there is no limit. 1-infinite Available values in kbps
Web: N/A UCI: httpclient.default.CAFile Opt: CAFile	Defines the path to the certificate authority file stored on the router.
Web: N/A UCI: httpclient.default.IgnoreServerCertificateStatus	Defines whether to skip the status check on the server certificate. Default: 0 Disabled 1 Enabled

Web Field/UCI/ Package Option	Description
Opt: IgnoreServerCertificate Status	

36.4. 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
```

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'
```

36.5. User Management Using UCI

User management is not currently available using the web interface. You can configure the feature using UCI or Activator.

Configuration packages used

Package	Sections
management_users	Users

Configuring User Management

You can create different users on the system by defining them in the user management configuration file. This gives users access to different services.

Web Field/UCI/Package Option	Description
General settings	
Web: n/a UCI: management_users.@user[x].enabled Opt: enable	Enables/creates the user. Default: 0 Disabled 1 Enabled
Web: n/a UCI: management_users.@user[x].username Opt: username	Specifies the user's username.
Web: n/a UCI: management_users.@user[x].password Opt: password	Specifies the user's password. When entering the user password enter in plain text using the password option. After reboot the password is displayed encrypted via the CLI using the hashpassword option. UCI: `management_users.@user[x].hashpassword` Opt: hashpassword. Note: a SRP user password will be displayed using the srphash option.
Web: n/a UCI: management_users.@user[x].webuser Opt: webuser	Specifies web access permissions for the user. Note: webuser will only work if linuxuser is set to Enabled. Default: 0 Disabled 1 Enabled
Web: n/a UCI: management_users.@user[x].chapuser Opt: chapuser	Specifies CHAP access permissions for the PPP connection. Note: chapuser will only work if linux user is set to no. Default: 0 Disabled 1 Enabled
Web: n/a UCI: management_users.@user[x].papuser Opt: papuser	Specifies PAP access permissions for the PPP connection. Default: 0 Disabled 1 Enabled
Web: n/a UCI: management_users.@user[x].srpuser Opt: srpuser	Specifies SRP access permissions for the PPP connection. Default: 0 Disabled 1 Enabled
Web: n/a UCI: management_users.@user[x].smsuser Opt: smsuser	Specifies SMS access permissions for the user. Default: 0 Disabled 1 Enabled
Web: n/a UCI: linuxuser Opt: linuxuser	Specifies linuxuser access permissions for the user. Default: 0 Disabled 1 Enabled
Web: n/a UCI: List allowed_pages Opt: list allowed_pages	Specifies which pages the user can view. Multiple pages should be entered using a space to separate if using UCI.

Note:

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.

User Management using UCI Example

```
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
```

User Management using Package Options Example

```
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'
option smsuser '0'
```

36.6. Configuring 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].password=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.

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.

36.7. Configuring User Access To Specific Web Pages

To specify particular pages a user can view, add the list allowed_pages. Examples are:

```
list allowed_pages '/admin/status'
```

The user can view admin status page only.

```
List allowed_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.

37. Configuring Monitor

Introduction

Our monitoring system (Monitor) is a secure portal that provides:

- Central 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
- Interface statistic collection (ISAD)

37.1. 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

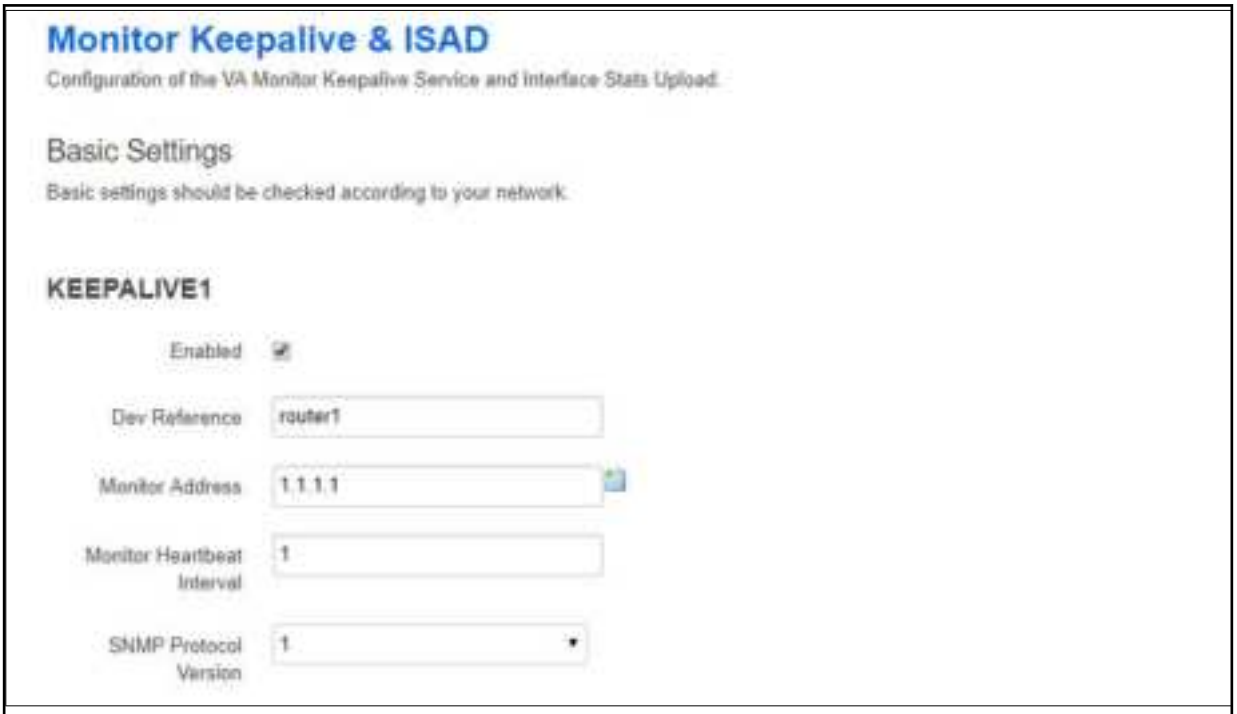
Configuration Package Used

Package	Sections
monitor	keepalive
network	interface

37.2. 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.



The Monitor & ISAD keepalive page

Basic Settings

Web Field/UCI/Package Option	Description
Web: Enabled UCI: monitor.@keepalive[0].enabled Opt: Enabled	Enables Monitor to send heartbeats to the router. Default: 0 — Disabled 1 — Enabled
Web: Dev Reference UCI: monitor.@keepalive[0].dev_reference Opt: dev_reference	Sets a unique identification for this device known to Monitor.
Web: Monitor Address UCI: monitor.@keepalive[0].monitor_ip Opt: list monitor_ip	Defines the IP address of Monitor. It is possible to specify multiple addresses to which SNMP heartbeat traps will be sent. To configure via UCI use a space separator. Example: `monitor.@keepalive[0].monitor_ip=1.1.1.1 2.2.2.2`
Web: Montor Heartbeat Interval UCI: monitor.@keepalive[0].interval_min Opt: interval_min	Specifies the interval, in minutes, at which traps are sent. Default: 1 — SNMP version 1 2c — SNMP version 2c 3 — SNMP version 3

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

SNMP Protocol Version	3
User Name	
Authentication Protocol	SHA
Authentication Password	
Privacy Protocol	AES
Privacy Password	
SNMPv3 Context	
SNMPv3 Context Engine ID	
SNMPv3 Security Engine ID	

The Monitor & ISAD keepalive page for SNMP v3

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

37.3. Configuring Keepalive Heartbeat Using Command Line

Keepalive is configured under the monitor package.

By default, all keepalive instances are named 'keepalive', instances are 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'
```

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
```

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'
```

37.4. Enabling Interface Status In Keepalive Heartbeat Via Web Interface

The keepalive heartbeat can send information on multiple interfaces. 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.

The interface common configuration page

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

37.5. Enable Interface Status Using UCI

Interface status is configured under the network package.

```
root@VA_router:~# uci show network
network.@interface[0]=interface
::::
network.@interface[0].monitored=1
::::
```

Enable Interface Status using Package Option

```
root@VA_router:~# uci export network
package network
config interface 'WAN'
::::
option monitored '1'
::::
```

37.6. Reporting GPS Location To Monitor

To allow Monitor to display a router GPS location, you can configure the GPS coordinates to be sent in the heartbeat keepalive from the router.

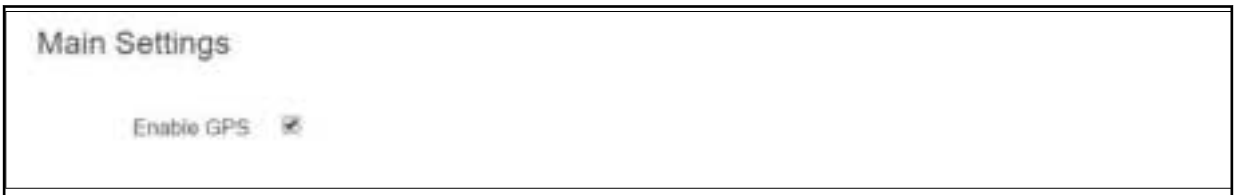
GPS location is only available in supported hardware models.

Ensure monitor heartbeat is correctly configured as in the previous section.

Package	Sections
gpsd	gpsd

Configuring GPS Location via the Web Interface

Select **Services -> GPS**. The GPS configuration page appears. The web interface configures a gpsd section named core.



The GPS configuration page

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

Configuring GPS Location using Command Line

GPS location is configured under the gpsd package.

GPS Location using UCI

```
root@VA_router:~# uci show gpsd
gpsd.core=gpsd
gpsd.core.enabled=1
```

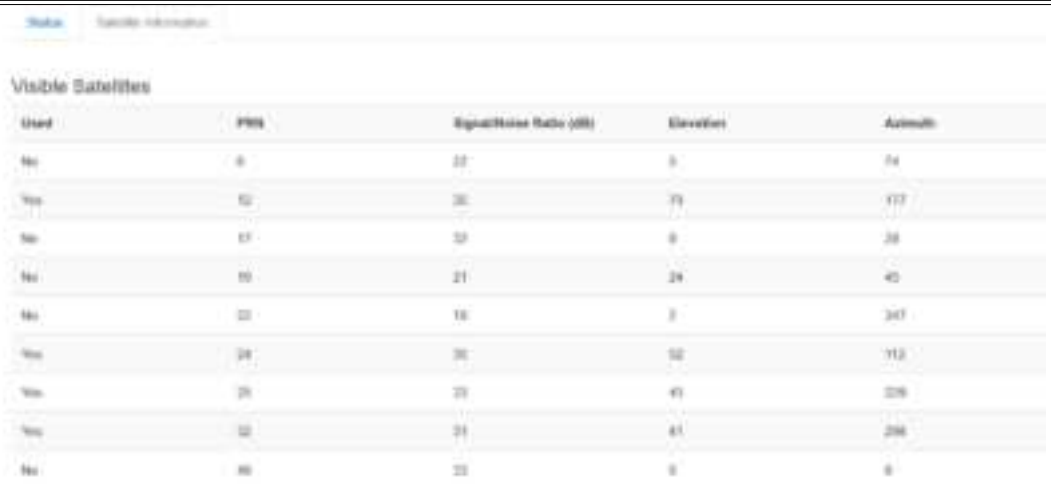
GPS Location using Package Options

```
root@VA_router:~# uci export gpsd
package gpsd
config gpsd 'core'
option enabled '1'
```

37.7. GPS Diagnostics

To view information on GPS coordinates via the web interface, select **Status -> GPS Information**. There are two tabs: Status and Satellite Information.

The GPS status page



The GPS visible satellites page

To view GPS coordinates via command line, enter:

```
root@VA_router:~# gpspeek

Fix: 3D,1495467700,53.342529,-6.241236,27.700000,202.600000,0.000000,0.000000
```

37.8. Reporting Syslog To Monitor

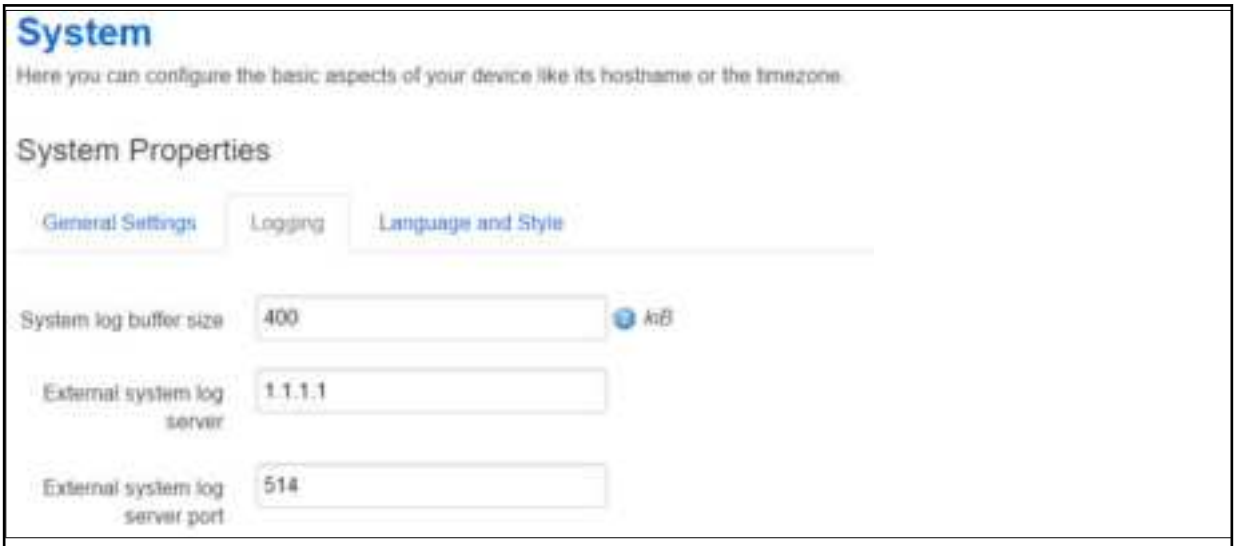
Configuration Package Used

Package	Sections
system	main

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.



The system properties page

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

37.9. Configuring Syslog Events To Monitor Using Command Line

Syslog is configured under the system package.

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
```

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'
```

37.10. 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 40.2 above. Interfaces should have option monitored enabled as part of the collection.

ISAD replaces the deprecated SLA feature.

Package	Sections
monitor	interface_stats

Configuring ISAD using the web interface

The Monitor keepalive & ISAD interface stats page

Select **Services -> Monitor**. The Monitor Keepalive & ISAD page appears. ISAD is configured under the **Interface Stats** section.

Web Field/UCI/Package Option	Description
Web: Enabled UCI: monitor.stats.enabled=1 Opt: enabled	Enables ISAD. Default: 0 — Disabled 1 — Enabled
Web: Bin Period UCI: monitor.stats.bin_period Opt: time	Specifies how long to collect data for one bin. Specifies the interval, in minutes, at which traps are sent. Default: 1 hr — Bin collected for 1 hour.
Web: Maximum Number of Bins UCI: monitor.stats.bin_cache_size Opt: bin_cache_size	Specifies the maximum number of bins to store. Default: Empty — 24

Configuring ISAD using the command line

ISAD is configured under the monitor package.

Configuring ISAD using UCI

```
root@VA_router:~# uci show monitor
monitor.keepalivev1=keepalive
monitor.keepalivev1.enabled=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
```

Configuring 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'
```

37.11. ISAD Diagnostics

Checking Process

To check to see if ISAD is running, enter:

```
root@VA_router:~# pgrep -fl isad
5303 /usr/sbin/isad -b 60 -s 10 -c 200 -u /var/state /var/const_state
```

Checking Bin Statistics

To check if stats are being collected, enter:

```
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
```

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'
```

37.12. Speedtest Reporting

To assist in determining WAN line speed characteristics the router can be configured to:

- Implement a Discard Protocol (RFC863)
- Implement a Character Generation Protocol (RFC864)

NOTE

A central client is required to generate the speedtest traffic and produce the measurement reports.

Configuration is not currently available via the web interface.

Web Field/UCI/Package Option	Description
Web: n/a UCI: monitor.speedtest.discard_enabled Opt: discard_enabled	Enables listening on TCP port 9 and discarding all received data. Default: 0 — Disabled 1 — Enabled
Web: n/a UCI: monitor.speedtest.chargen_enabled Opt: chargen_enabled	Enables listening on TCP port 19 and streaming data to the connected client at maximum possible speed. Default: 0 — Disabled 1 — Enabled

Configuring Speedtest via the Command Line

Speedtest options are configured in the speedtest configuration section of the monitor package.

Speedtest using UCI

```
root@VA_router:~# uci show monitor

monitor.speedtest=speedtest

monitor.speedtest.discard_enabled

monitor.speedtest.chargen_enabled
```

Speedtest using Package Options

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

package monitor


config speedtest

option discard_enabled '0'

option chargen_enabled '0'
```

38. 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 (Objective Identifiers) defined in MIBs. In addition, information from the router can be pushed to a network management station in the form of SNMP traps.

Configuration Package Used

Package	Sections				
snmpd	access agent com2sec constant	exec group heartbeat informreceiver	inventory inventory_iftable monitor_disk monitor_ioerror	monitor_load monitor_memory monitor_process pass	system trapreceiver usm_user view

The SNMP application has several configuration sections:

System and Agent	Configures the SNMP agent
Com2Sec	Maps SNMP community names into an arbitrary security name.
Group	Assigns community names and SNMP protocols to groups.
View and Access	Creates views and sub-views of the whole available SNMP tree and grants specific access to those views on a group by group basis.
usm_user	Defines a user for SNMPv3 USM.
Trap Receiver	Sets the address of a notification receiver that should be sent SNMPv1 TRAPs and SNMPv2c TRAP2s.
Inform Receiver	Sets the address of a notification receiver that should be sent SNMPv2 INFORM notifications respectively.

38.1. Configuring SNMP Using The Web Interface

In the top menu, select **Services -> SNMP**. The SNMP Service page appears.

The SNMP service page

System and Agent Settings

Web Field/UCI/Package Option	Description
System Settings	
Web: System Location UCI: snmpd.system[0].sysLocation 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 UCI: snmpd.system[0].sysContact Opt: sysContact	
Web: System Name UCI: snmpd.system[0].sysName Opt: sysName	
Agent Settings	
Web: Agent Address UCI: snmpd.agent[0].agentaddress Opt: agentaddress	Specifies the address(es) and port(s) on which the agent should listen. [(udp|tcp):][address:]port [,...] Example: `udp:127.0.0.1:161, tcp:161, localhost:9161`
Web: Enable Authentication Traps UCI: snmpd.agent[0].authtrapenabled Opt: authtrapenabled	Enables or disables SNMP authentication trap. Default: 0 — Disabled 1 — Enabled Note: this is the SNMP poll authentication trap you set when there is a community mismatch.
Web: Enable Link State Notification UCI: snmpd.agent[0].link_updown_notify 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. Default: 0 — Disabled 1 — Enabled

38.1.1. 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.

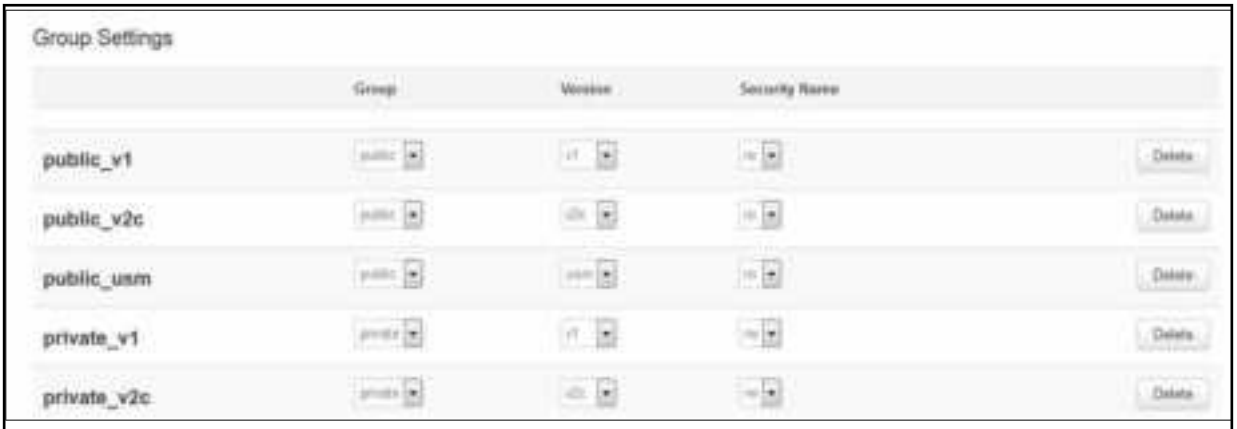
A community string is a password that is applied to a device to restrict both read-only and read-write access to the SNMP data on the device. These community strings should be chosen carefully to ensure they are not trivial. They should also be changed at regular intervals and in accordance with network security policies.

The COM2SEC settings section

Web Field/UCI/Package Option	Description
Web: Security Name UCI: snmpd.com2sec[x].secname Opt: secname	Specifies an arbitrary security name for the user.
Web: Source UCI: snmpd.com2sec[x].source Opt: source	A hostname, localhost or a subnet specified as a.b.c.d/mask or a.b.c.d/bits or 'default' for no restrictions.
Web: Community UCI: snmpd.com2sec[x].community Opt: community	Specifies the community string being presented in the request.

38.1.2. Group Settings Using The Web Interface

Group settings assign community names and SNMP protocols to groups.



The group settings section

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

38.1.3. 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.

The view settings section

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

38.1.4. Access Settings Using The Web Interface

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.

The access settings section

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Group

UCI: snmpd.access[x].group

Opt: group</td><td>Specifies the group to which access is being granted.</td></tr>
<tr><td>Web: Context

UCI: snmpd.access[x].context

Opt: context</td><td>SNMPv3 request context is matched against the value according to the prefix below. For SNMP v1 and SNMP v2c, the context must be none.

<table><tr><td>Default:</td><td>none</td></tr><tr><td></td><td>all</td></tr></table></td></tr>
<tr><td>Web: Version

UCI: snmpd.access[x].version

Opt: version</td><td>Specifies the SNMP version number being used in the request: any, v1, v2c and usm are supported.

<table><tr><td>Default: v1</td><td>SNMP v1</td></tr><tr><td>v2v</td><td>SNMP v2</td></tr><tr><td>usm</td><td>SNMP v3</td></tr><tr><td>any</td><td>Any SNMP version</td></tr></table></td></tr>
<tr><td>Web: Level

UCI: snmpd.access[x].level

Opt: level</td><td>Specifies the security level. For SNMP v1 and SNMP v2c the level must be noauth.

<table><tr><td>Default</td><td>nouath</td></tr><tr><td></td><td>auth</td></tr><tr><td></td><td>priv</td></tr></table></td></tr>
<tr><td>Web: Prefix

UCI: snmpd.access[x].prefix

Opt: prefix</td><td>Specifies how the context should be matched against the context of the incoming pdu.

<table><tr><td>Default:</td><td>exact</td></tr><tr><td></td><td>any</td></tr><tr><td></td><td>all</td></tr></table></td></tr>
<tr><td>Web: Read

UCI: snmpd.access[x].read

Opt: read</td><td>Specifies the view to be used for read access.</td></tr>
<tr><td>Web: Write

UCI: snmpd.access[x].write

Opt: write</td><td>Specifies the view to be used for write access.</td></tr>
<tr><td>Web: Notify

UCI: snmpd.access[x].notify

Opt: notify</td><td>Specifies the view to be used for notify access.</td></tr>
</table>

38.1.5. Trap Receiver

Trap receiver settings define a notification receiver that should be sent SNMPv1 TRAPs and SNMPv2c TRAP2.

Web Field/UCI/Package Option	Description
Web: Host UCI: snmpd.trapreceiver[x].host Opt: host	Host address. Can be either an IP address or an FQDN.
Web: Port UCI: snmpd.trapreceiver[x].port Opt: port	UDP port to be used for sending traps. Default: 162
Web: Version UCI: snmpd.trapreceiver[x].version Opt: version	SNMP version. Default: v1 v2
Web: Community UCI: snmpd.trapreceiver[x].community Opt: community	Community to use in trap messages for this host.

38.1.6. Inform Receiver Using The Web Interface

Inform receiver settings define a notification receiver that should be sent SNMPv2c INFORM notifications.

The inform receiver settings page

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

38.1.7. USM User Using Web Interface

Configure a user for for SNMPv3 USM (User Based Security Model).

The USM user settings page

Web Field/UCI/Package Option	Description
Web: Username UCI: snmpd.@usm_user[0].name Opt: name	Defines a USM username.
Web: Auth Protocol UCI: snmpd.@usm_user[0].auth_protocol Opt: auth_protocol	Defines the authentication protocol to use. Note: if omitted the user will be defined as noauth user. MD5 SHA
Web: Auth Password UCI: snmpd.@usm_user[0].auth_password Opt: auth_password	Defines the authentication password. Note: password must be at least 8 characters long.
Web: Priv Protocol UCI: snmpd.@usm_user[0].priv_protocol Opt: priv_protocol	Defines the privacy protocol to use. Note: if omitted the user will be defined as authNoPriv user. MD5 SHA
Web: Priv Password UCI: snmpd.@usm_user[0].priv_password Opt: priv_password	Defines the privacy password. Note: the password must be at least 8 characters long.
Web: OID UCI: snmpd.@usm_user[0].oid Opt: oid	Defines the OID branch to restrict this user to. Similar to view restrictions in v1 and v2c

38.2. Configuring SNMP Using Command Line

SNMP is configured under the snmpd package. The configuration files are stored on **/etc/config/snmpd**.

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
```

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,localhost:9161'
```

38.2.1. Com2sec Using UCI

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.

```
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
```

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'
```

38.2.2. Group Settings Using Command Line

The following example specifies that a request from the security name “ro” using snmp v1, v2c or USM (User Based Security Model for SNMPv3) are all mapped to the “public” group. Similarly, requests from the security name “rw” in all protocols are mapped to the “private” group.

```
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
```

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'
```

38.2.3. View Settings Using UCI

The following example defines two views, one for the entire system and another for only mib2.

```
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
```

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'
```

38.2.4. Access Settings Using Command Line

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.

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'
```

38.2.5. SNMP Traps Settings Using Command Line

By default, all SNMP trap instances are named 'trapreceiver, it is identified by @trapreceiver then the trap receiver position in the package as a number. For example, for the first trap receiver in the package using UCI:

```
snmpd.@trapreceiver[0]=trapreceiver
snmpd.@trapreceiver[0].host=1.1.1.1:161
```

Or using package options:

```
config trapreceiver
```

```
option host '1.1.1.1:161'
```

However, to better identify it, it is recommended to give the trap receiver instance a name. For example, to create a trap receiver instance named TrapRecv1.

To define a named trap receiver instance using UCI, enter:

```
snmpd.TrapRecv1=TrapRecv1
snmpd.TrapRecv1.host=1.1.1.1:161
```

To define a named trap receiver instance using package options, enter:

```
config trapreceiver TrapRecv1
option host '1.1.1.1:161'
```

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
```

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'
```

38.2.6. SNMP Inform Receiver Settings Using Command Line

By default, all SNMP inform receiver instances are named 'informreceiver', it is identified by @informreceiver then the inform receiver position in the package as a number. For example, for the first inform receiver in the package using UCI:

```
snmpd.@informreceiver [0]=informreceiver
snmpd.@informreceiver [0].host=1.1.1.1
```

Or using package options:

```
config informreceiver
option host '1.1.1.1'
```

However, to better identify it, it is recommended to give the inform receiver instance a name. For example, to create a inform receiver instance named InformRecv1.

To define a named trap receiver instance using UCI, enter:

```
snmpd.InformRecv1=InformRecv1
snmpd.InformRecv1.host=1.1.1.1
```

To define a named trap receiver instance using package options, enter:

```
config informreceiver InformRecv1
option host '1.1.1.1'
```

SNMP Inform Receiver using UCI

```
snmpd.@informreceiver[0]=informreceiver
snmpd.@informreceiver[0].host=1.1.1.1
snmpd.@informreceiver[0].port=67
snmpd.@informreceiver[0].community=private
```

SNMP Inform Receiver using Package Options

```
config informreceiver
option host '1.1.1.1'
option port '67'
option community 'private'
```

38.2.7. SNMP USM User Settings

By default, all USM User instances are named 'usm_user', it is identified by @usm_user then the USM user position in the package as a number. For example, for the first USM User in the package using UCI:

```
snmpd.@usm_user[0]=usm_user
snmpd.@usm_user[0].name=username
```

Or using package options:

```
config usm_user
option name 'username'
```

However, to better identify it, it is recommended to give the usm_user instance a name. For example, to create a usm_user instance named User1.

To define a named usm_user instance using UCI, enter:

```
snmpd.User1=User1
snmpd.User1.name=username
```

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

```
config usm_user 'User1'
option name 'username'
```

SNMP USM using UCI

```
snmpd.@usm_user[0]=usm_user
snmpd.@usm_user[0].name=username
snmpd.@usm_user[0].auth_protocol=SHA
snmpd.@usm_user[0].auth_password=password
snmpd.@usm_user[0].priv_protocol=AES
snmpd.@usm_user[0].priv_password=password
snmpd.@usm_user[0].oid=1.2.3.4
```

SNMP USM using Package Options

```
config usm_user
option name 'username'
option auth_protocol 'SHA'
option auth_password 'password'
option priv_protocol 'AES'
option priv_password 'aespassword'
option oid '1.2.3.4'
```

38.3. 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.

Package	Sections
network	interface

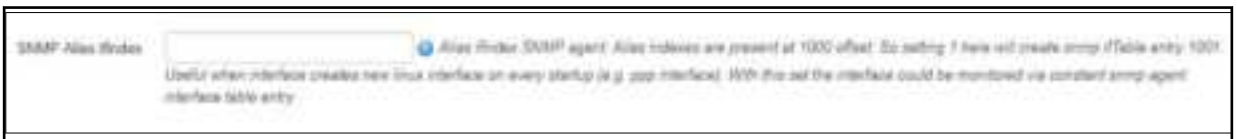
Configuring SNMP Interface Alias

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, configure the SNMP manager 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.



The interface SNMP alias ifindex field advanced settings page

UCI/Package Option	Description
Web: SNMP Alias ifindex UCI: network.@interface[X].snmp_alias_ifindex Opt: snmp_alias_ifindex	Defines a static SNMP interface alias index for this interface that can be polled using via the SNMP interface index. snmp_alias_ifindex+1000 Default: Blank — No SNMP interface alias index Range: 0-4294966295
Web: n/a UCI: network.@interface[X].snmp_alias_ifdescr Opt: snmp_alias_ifdescr	Defines an alias name to be reported for the UCI name in the enterprise MIB for UCI interfaces, and in alias entries in the ifIndex table. If present, this option supercedes the default ifDescr value (usually the UCI interface name, or configured ifName). Default: Blank — No SNMP interface alias name Range

38.4. Automatic SNMP Traps

Last Gasp

The router will automatically generate an SNMP trap when power loss is detected, and attempt to deliver to the configured trap receiver - ORK firmware family only.

NOTE

Whether the hardware is able to deliver the last gasp depends on the hold up time on the particular hardware model and the network conditions.

Event	SNMP Trap Format
Shutdown	{ SNMPv1 { Trap(28) E:8072.4 192.168.100.1 enterpriseSpecific s=2 8382 }

Cold Start

On completion of system start up, the router will generate a cold start SNMP trap and deliver to the configured trap receiver.

Event	SNMP Trap Format
Startup	{ SNMPv1 { Trap(29) E:8072.3.2.10 192.168.100.1 coldStart 9 } }

38.5. SNMP Diagnostics

SNMP Process

To check the SNMP process is running correctly, enter:

```
root@VA_router:~# pgrep -fl snmpd

6970 /usr/sbin/snmpd -Lsd0-6 -p /var/run/snmpd.pid -m -c /var/conf/snmpd.conf
```

SNMP Port

To check the SNMP service is listening on the configured port, enter:

```
root@VA_router:~# netstat -pantu | grep snmp

udp        0 0 0.0.0.0:161      0.0.0.0:*           6970/snmpd
```

Reviving SNMP Values

SNMP values can be queried by an `snmpwalk` or `snmpget` command either locally or remotely.

snmpwalk

To create an `snmpwalk` locally, enter `snmpwalk`. An example snmpwalk is shown below:

```
root@VA_router:~# snmpwalk -c public -v 1localhost .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
```

snmpget

To create an `snmpget` command locally, enter:

```
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"
```

SNMP Status

To see an overview including tx/rx packets and uptime of the SNMP process, enter:

```
root@VA_router:~# snmpstatus -c public -v 2c localhost
[UDP: [0.0.0.0]-&gt;[127.0.0.1]:161]=&gt;[Virtual Access GWXXXX, SN#
00E0C812D1A0, EDG-21.00.07.008] Up: 0:17:05.87
```

39. Event System

Merlin routers feature an event system. It allows you to forward Merlin 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 the web interface and the command line.

Configuration Package Used

Package	Section
va_eventd	main
	forwarding
	target
	conn_tester

39.1. 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 and 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, an 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.

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`.

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.

The attributes of a target vary significantly depending on its type.

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.

39.2. 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.

39.3. 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.

The event system connection tester configuration page

Web Field/UCI/Package Option	Description
Web: Enabled UCI: va_eventd.@conn_tester[0].enabled Opt: enabled	Enables a connection tester. Default: 0 — Disabled 1 — Enabled
Web: Connection Tester Name UCI: va_eventd.@conn_tester[0].name Opt: name	Defines the connection tester name. This is used when configuring a connection tester for an event destination.
Web: Type UCI: va_eventd.@conn_tester[0].type Opt: type	Defines the connection tester type. Web Value: Ping — Description: Verifies target by ping — UCI: ping
Web: Ping Target UCI: va_eventd.@conn_tester[0].ping_dest_ad dr Opt: ping_dest_addr	Defines the IP address for the target ping. Note: only displayed if connection tester type is set to 'Ping'.
Web: Ping Source UCI: va_eventd.@conn_tester[0].ping_source Opt: ping_source	Defines an interface or IP address to source the pings from.
Web: Ping Success Duration UCI: va_eventd.@conn_tester[0].ping_success_duration_sec Opt: ping_success_duration_sec	Defines the duration, in seconds, for which a successful ping defines a connection tester as up. Note: only displayed if connection tester type is set to 'Ping'.
Web: Link Interface UCI: va_eventd.@conn_tester[0].link_iface Opt: link_iface	Defines the interface to monitor when the connection tester type is set to 'link'. Configured interfaces are listed. Note: only displayed if configured syslog server.

39.4. 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.

39.4.1. Syslog Target

When a syslog target receives an event, it sends it to the configured syslog server.

The event system syslog event destination configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: va_eventd.@target[0].enabled

Opt: enabled</td><td>Enables an event destination. This is used in the event filters section.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Destination name

UCI: va_eventd.@target[0].name

Opt: name</td><td>Defines a name for the event destination.</td></tr>
<tr><td>Web: Type

UCI: va_eventd.@target[0].type

Opt: type</td><td>Defines the event destination type. For syslog server choose Syslog.

<table><tr><td>Web Value</td><td>Description</td><td>UCI</td></tr><tr><td>Syslog</td><td>Syslog server</td><td>syslog</td></tr><tr><td>Email</td><td>Event sent via email</td><td></td></tr><tr><td>SNMP</td><td>Event sent via SNMP trap</td><td></td></tr><tr><td>Exec</td><td>Command executed when event occurs</td><td></td></tr><tr><td>SMS</td><td>Event send via SMS</td><td></td></tr><tr><td>File</td><td>File targer</td><td></td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name

UCI: va_eventd.@target[0]. conn_tester

Opt: conn_tester</td><td>Defines the connection tester (if any) to use to verify the syslog target.

<table><tr><td>Default: None</td><td>No connection tester. UCI option not present.</td></tr><tr><td>Range:</td><td></td></tr></table></td></tr>
<tr><td>Web: Destination Address

UCI: va_eventd.@target[0]. target_addr

Opt: target_addr</td><td>Defines the syslog target IP/FQDN and port.

<table><tr><td>Default:</td><td></td></tr><tr><td>Range</td><td>a.b.c.d : port or fqdn : port</td></tr></table></td></tr>
<tr><td>Web: Syslog Over TCP

UCI: va_eventd.@target[0].tcp_syslog

Opt: tcp_syslog</td><td>Defines whether to use TCP for delivery of the syslog event.

<table><tr><td>Default: 0</td><td>Use UDP</td></tr><tr><td>1</td><td>Use TCP</td></tr></table></td></tr>
<tr><td>Web: Message Template

UCI: va_eventd.@target[0].template

Opt: template</td><td>Defines the message template to use for the event. In general, this should be left empty.

See the section on message templates below.</td></tr>
<tr><td>Web: n/a

UCI: va_eventd.@target[0].facility

Opt: facility</td><td>Defines a custom facility to overwrite existing facility on syslog messages before delivery to syslog target.

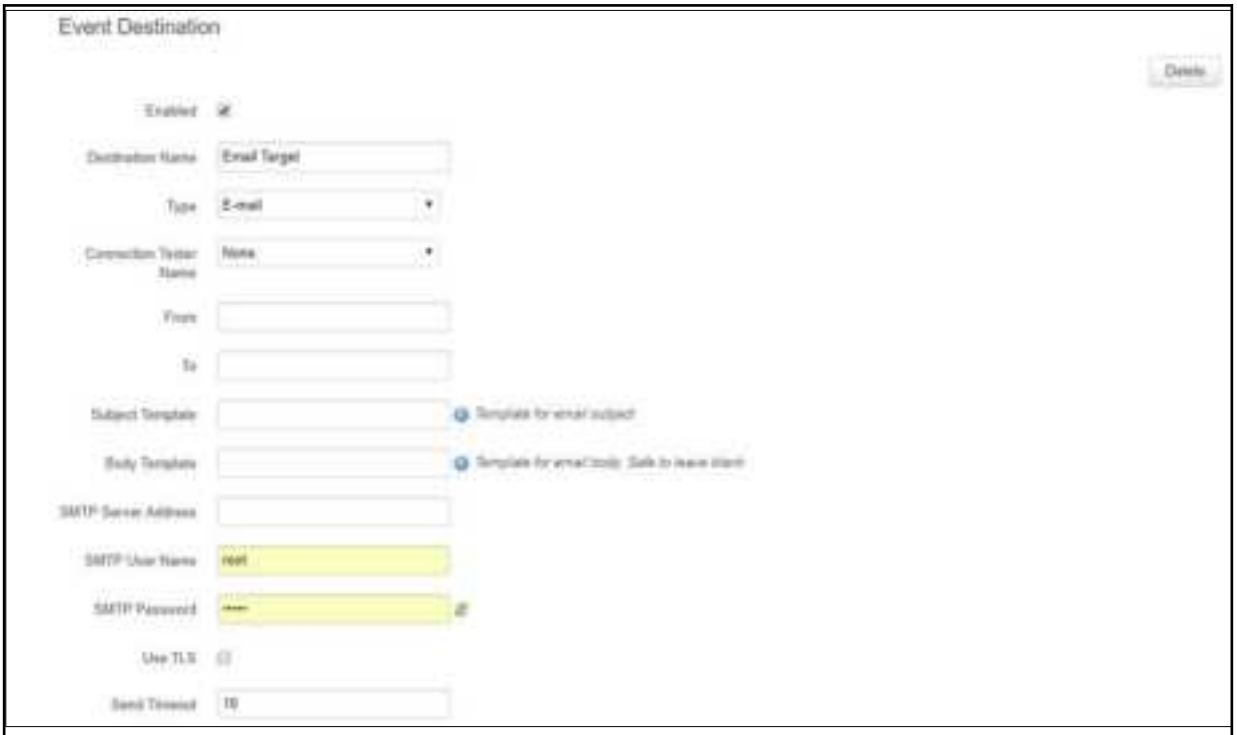
<table><tr><td>Default:</td><td>Does not overwrite existing facility.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: va_eventd.@target[0].severity Opt: severity</td><td>Defines a custom severity to overwrite existing severity on syslog messages before delivery to syslog target.

<table><tr><td>Default:</td><td>Does not overwrite existing facility.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
</table>

39.4.2. Email Target

When an email target receives an event, it sends it to the configured email address.



The event system email event destination configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: va_eventd.@target[0].enabled

Opt: enabled</td><td>Enables an event destination.

<table><tr><td>Default: 0</td><td>disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Destination name

UCI: va_eventd.@target[0].name

Opt: name</td><td>Defines a name for the event destination.</td></tr>
<tr><td>Web: Type

UCI: va_eventd.@target[0].type

Opt: type</td><td>Defines the event destination type. For an email server choose Email.

<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td>Syslog target</td><td>syslog</td></tr><tr><td>SNMP Trap</td><td>SNMP target</td><td>snmptrap</td></tr><tr><td>Email</td><td>Email target</td><td>email</td></tr><tr><td>Execute</td><td>Execute target</td><td>exec</td></tr><tr><td>SMS</td><td>SMS target</td><td>sms</td></tr><tr><td>File</td><td>File target</td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name

UCI: va_eventd.@target[0]. conn_tester

Opt: conn_tester</td><td>Defines the connection tester (if any) to use to verify the email target.

<table><tr><td>Default:</td><td>none. No connection tester. UCI option not present.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: From

UCI: va_eventd.@target[0].from

Opt: from</td><td>Defines the ‘from’ address for the email.</td></tr>
<tr><td>Web: To

UCI: va_eventd.@target[0].to

Opt: to</td><td>Defines the ‘to’ address for the email.</td></tr>
<tr><td>Web: Subject Template

UCI: va_eventd.@target[0].subject_template

Opt: subject_template</td><td>Defines subject template for the email. In general, this should be left empty. Example:

<code>va_eventd.@target[0].subject_template="%{severityName} %{eventName}!!!"</code>

See the section on message templates below.</td></tr>
<tr><td>Web: Body Template

UCI: va_eventd.@target[0].body_template

Opt: body_template</td><td>Defines the email body template. In general, this should be left blank. Example:

va_eventd.@target[0].body_template="%{eventName} (%{class}.%{subclass}) happened!"

See the section on message templates below.</td></tr>
<tr><td>Web: SMTP Server Address

UCI: va_eventd.@target[0].smtp_addr

Opt: smtp.addr</td><td>Defines the email server address and port.

<table><tr><td>Default:</td><td></td></tr><tr><td>Range</td><td>a.b.c.d:port or fqdn:port</td></tr></table></td></tr>
<tr><td>Web: SMTP User Name

UCI: va_eventd.@target[0].smtp_user

Opt: smtp_user</td><td>Defines user name for SMTP authentication.</td></tr>
<tr><td>Web: SMTP Password

UCI: va_eventd.@target[0].smtp_password

Opt: smtp_password</td><td>Defines the password for SMTP authentication.</td></tr>
</table>

Web Field/UCI/Package Option	Description
Web: Use TLS UCI: va_eventd.@target[0].use_tls Opt: use_tls	Enables TLS (Transport Layer Security) support. Default: 0 Range 0-1
Web: Send Timeout UCI: va_eventd.@target[0].timeout_sec Opt: timeout_sec	Defines the email send timeout in seconds. Default: 10 Range
Web: Use StartTLS UCI: va_eventd.@target[0]. tls_starttls Opt: tls_starttls	Enables StartTLS support for TLS. (Only displayed when TLS is enabled) Default: 0 Range 0-1
Web: Force SSLv3 UCI: va_eventd.@target[0].tls_forcessl3 Opt: tls_forcessl3	Enables force SSLv3 for TLS. (Only displayed when TLS is enabled)

39.4.3. SNMP Target

When a SNMP target receives an event, it sends it in a trap to the configured SNMP manager.

The event system SNMP event destination configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: va_eventd.@target[0].enabled

Opt: enabled</td><td>Enables an event destination.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Destination name

UCI: va_eventd.@target[0].name

Opt: name</td><td>Defines a name for the event destination.</td></tr>
<tr><td>Web: Type

UCI: va_eventd.@target[0].type

Opt: type</td><td>Defines the event destination type. For SNMP server, choose SNMP Trap.

<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td>Syslog target</td><td>syslog</td></tr><tr><td>SNMP Trap</td><td>SNMP target</td><td>snmptrap</td></tr><tr><td>Email</td><td>Email target</td><td>email</td></tr><tr><td>Execute</td><td>Execute target</td><td>exec</td></tr><tr><td>SMS</td><td>SMS target</td><td>sms</td></tr><tr><td>File</td><td>File target</td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name

UCI: va_eventd.@target[0]. conn_tester

Opt: conn_tester</td><td>Defines the connection tester (if any) to use to verify the SNMP target.

<table><tr><td>Default:</td><td>None. No connection tester. UCI option not present.</td></tr></table></td></tr>
<tr><td>Web: Destination Address

UCI: va_eventd.@target[0]. target_addr

Opt: target_addr</td><td>Defines the SNMP target IP/FQDN and port.

<table><tr><td>Range:</td><td>a.b.c.d:port or fqdn:port</td></tr></table></td></tr>
<tr><td>Web: Message Template

UCI: va_eventd.@target[0].template Opt: template</td><td>Defines the message template to use for the event. In general, this should be left empty. Example:

<code>va_eventd.@target[0].template=”%{eventName} %{eventSpecificTemplate}”</code>

See the section on message templates below.</td></tr>
<tr><td>Web: Agent Address

UCI: va_eventd.@target[0]. agent_addr

Opt: agent_addr</td><td>Defines the IP address to source the SNMP trap. (optional)

<table><tr><td>Default: localhost</td><td>Local IP</td></tr><tr><td>Range</td><td>Localhost or IP address</td></tr></table></td></tr>
<tr><td>Web: SNMP Protocol Version

UCI: va_eventd.@target[0].snmp_version

Opt: snmp_version</td><td>Defines the SNMP version.

<table><tr><td>Default: 1</td><td>SNMPv1</td></tr><tr><td>2c</td><td>SNMPv2c</td></tr><tr><td>3</td><td>SNMPv3</td></tr></table></td></tr>
<tr><td>Web: Community

UCI: va_eventd.@target[0].community

Opt: community</td><td>Defines the community string for SNMPv1.</td></tr>
<tr><td>Web: Username

UCI: va_eventd.@target[0].snmp_uname

Opt: snmp_uname</td><td>Defines the username for SNMPv3. Only displayed when SNMP protocol version is SNMPv3</td></tr>
<tr><td>Web: Authentication Protocol

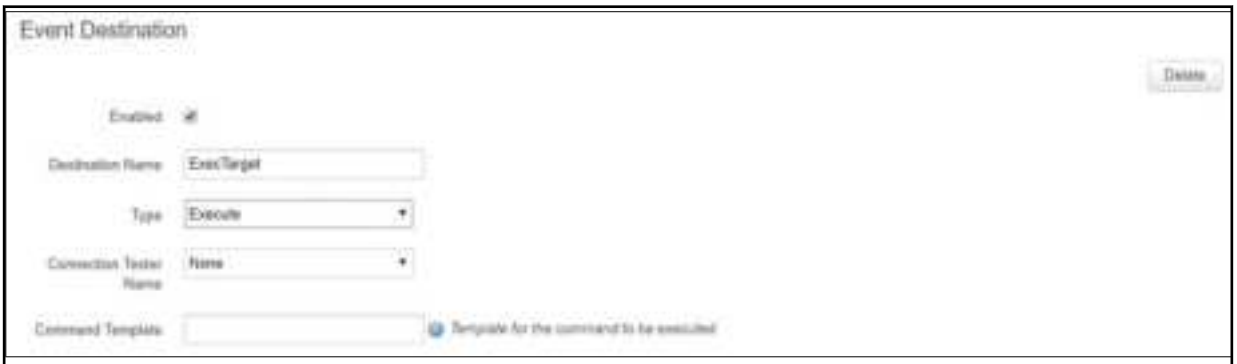
UCI: va_eventd.@target[0].snmp_auth_proto

Opt: snmp_auth_proto</td><td>Defines the SNMPv3 authentication protocol. Only displayed when SNMP protocol version is SNMPv3.</td></tr>
</table>

Web: Authentication Password UCI: va_eventd.@target[0].snmp_auth_pass Opt: snmp_auth_pass	Defines the SNMPv3 authentication password. Only displayed when SNMPv3 authentication protocol is configured. MD5 SHA
Web: Privacy Protocol UCI: va_eventd.@target[0].snmp_priv_proto Opt: snmp_priv_proto	Defines the SNMPv3 privacy protocol. Only displayed when SNMP authentication protocol is configured. MD5 SHA
Web: Privacy Password UCI: va_eventd.@target[0].snmp_priv_pass Opt: snmp_priv_pass	Defines SNMPv3 privacy password. Only displayed when SNMP privacy protocol is configured. DES AES
Web: SNMPv3 Context UCI: va_eventd.@target[0].snmp_context Opt: snmp_context	Defines the SNMPv3 context. Only displayed when SNMP authentication protocol is configured.
Web: SNMPv3 Context Engine ID UCI: va_eventd.@target[0].snmp_context_eid Opt: snmp_context_eid	Defines the SNMPv3 context engine ID. Only displayed when SNMP authentication protocol is configured.
Web: SNMPv3 Security Engine ID UCI: va_eventd.@target[0].snmp_sec_eid Opt: snmp_sec_eid	Defines the SNMPv3 security engine ID. Only displayed when SNMP authentication protocol is configured.

39.4.4. Exec Target

When an Execute target receives an event, it executes a shell command.



The event system exec event destination configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: va_eventd.@target[0].enabled

Opt: enabled</td><td>Enables an event destination.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Destination name

UCI: va_eventd.@target[0].name

Opt: name</td><td>Defines a name for the event destination.</td></tr>
<tr><td>Web: Type

UCI: va_eventd.@target[0].type

Opt: type</td><td>Defines the event destination type. For shell command execution, choose Execute.

<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td>Syslog target</td><td>syslog</td></tr><tr><td>SNMP Trap</td><td>SNMP target</td><td>snmptrap</td></tr><tr><td>Email</td><td>Email target</td><td>email</td></tr><tr><td>Execute</td><td>Execute target</td><td>exec</td></tr><tr><td>SMS</td><td>SMS target</td><td>sms</td></tr><tr><td>File</td><td>File target</td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name

UCI: va_eventd.@target[0]. conn_tester

Opt: conn_tester</td><td>Defines the connection tester, if any, to use to verify the execute target.

<table><tr><td>Default: None</td><td>No connection tester. UCI option not present.</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Command Template

UCI: va_eventd.@target[0].cmd_template

Opt: cmd_template</td><td>Defines the command template to use for the event. Example to log a syslog message:

<code>va_eventd.@target[0].cmd_template="logger -t eventer %{eventName}"</code>

See the section on message templates below.</td></tr>
</table>

39.4.5. SMS Target

When an SMS target receives an event, it sends an SMS message.

The event system SMS event destination configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: va_eventd.@target[0].enabled

Opt: enabled</td><td>Enables an event destination.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Destination name

UCI: va_eventd.@target[0].name

Opt: name</td><td>Defines a name for the event destination.</td></tr>
<tr><td>Web: Type

UCI: va_eventd.@target[0].type

Opt: type</td><td>Defines the event destination type. For SMS destination, choose SMS.

<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td>Syslog target</td><td>syslog</td></tr><tr><td>SNMP Trap</td><td>SNMP target</td><td>snmptrap</td></tr><tr><td>Email</td><td>Email target</td><td>email</td></tr><tr><td>Execute</td><td>Execute target</td><td>exec</td></tr><tr><td>SMS</td><td>SMS target</td><td>sms</td></tr><tr><td>File</td><td>File target</td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name

UCI: va_eventd.@target[0]. conn_tester

Opt: conn_tester</td><td>Defines the connection tester, if any, to use to verify the SMS target.

<table><tr><td>Default: None</td><td>No connection tester. UCI option not present.</td></tr></table></td></tr>
<tr><td>Web: Message Template

UCI: va_eventd.@target[0].template

Opt: template</td><td>Defines the message template to use for the event. In general, this should be left empty. Example:

<code>va_eventd.@target[0].template="%{eventName}"</code>

See the section on message templates below.</td></tr>
<tr><td>Web: Phone Number

UCI: va_eventd.@target[0].callee

Opt: callee</td><td>Defines the phone number for sending SMS to.</td></tr>
</table>

39.4.6. File Target

When file target receives an event, it logs to a file.

The event system file event destination configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: va_eventd.@target[0].enabled

Opt: enabled</td><td>Enables an event destination.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Destination Name

UCI: va_eventd.@target[0].name

Opt: name</td><td>Defines a name for the event destination.</td></tr>
<tr><td>Web: Type

UCI: va_eventd.@target[0].type

Opt: type</td><td>Defines the event destination type. For file choose File.

<table><tr><th>Web Value</th><th>Description</th><th>UCI</th></tr><tr><td>Syslog</td><td>Syslog target</td><td>syslog</td></tr><tr><td>SNMP Trap</td><td>SNMP target</td><td>snmptrap</td></tr><tr><td>Email</td><td>Email target</td><td>email</td></tr><tr><td>Execute</td><td>Execute target</td><td>exec</td></tr><tr><td>SMS</td><td>SMS target</td><td>sms</td></tr><tr><td>File</td><td>File target</td><td>file</td></tr></table></td></tr>
<tr><td>Web: Connection Tester Name

UCI: va_eventd.@target[0]. conn_tester

Opt: conn_tester</td><td>Defines the connection tester (if any) to use to verify the File target.

<table><tr><td>Default: None</td><td>No connection tester. UCI option not present</td></tr></table></td></tr>
<tr><td>Web: Message Template

UCI: va_eventd.@target[0].template

Opt: template</td><td>Defines the message template to use for the event. In general, this should be left empty.

See the section on message templates below.</td></tr>
<tr><td>Web: File Name

UCI: va_eventd.@target[0].file_name

Opt: file_name</td><td>Defines a file name for the event destination. Full path.</td></tr>
<tr><td>Web: Max Size (KiB)

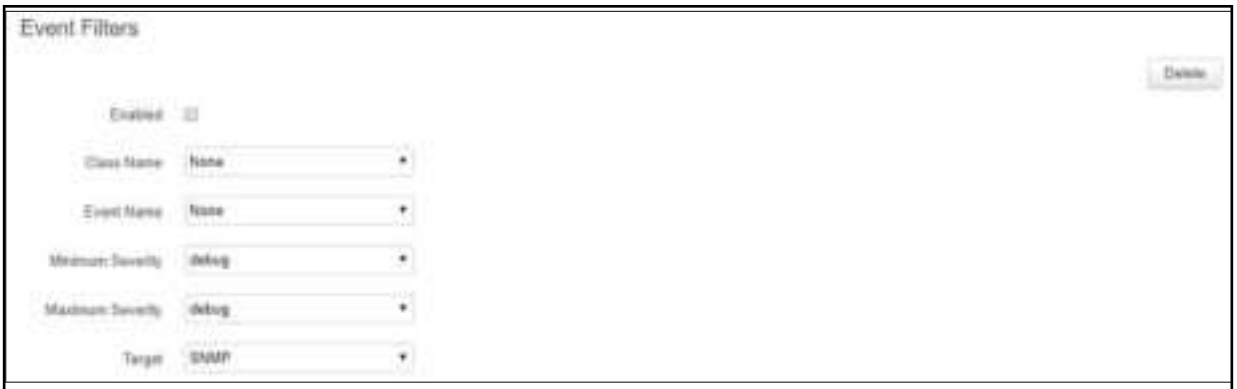
UCI: va_eventd.@target[0].max_size_kb

Opt: file_name</td><td>Defines a file size in kilobits.

<table><tr><td>Default:</td><td>2048</td></tr></table></td></tr>
</table>

39.5. Event FIlters

Event filters are used 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.



The event system event filters configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enabled

UCI: va_eventd.@forwarding[0].enabled Opt: enabled</td><td>Enables an event filter.

<table><tr><td>Default: 1</td><td>Disabled</td></tr><tr><td>0</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Class Name

UCI: va_eventd.@forwarding[0].className

Opt: className</td><td>Only match events with the given class name. Available class names are listed or can be viewed using the command <code>vae_cli -d</code></td></tr>
<tr><td>Web: Event Name

UCI: va_eventd.@forwarding[0].eventName

Opt: eventName</td><td>Only match events with the given event name. Available event names are listed. The event name is optional and can be omitted.</td></tr>
<tr><td>Web: Minimum Severity

UCI: va_eventd.@forwarding[0].severity

Opt: severity</td><td>Defines the minimum event severity. The minimum severity event is DEBUG. Events generated within the minimum and maximum event severity will be matched.

Minimum and maximum severity are specified in the one UCI option and entered using a dash (-) separator in the form minimum-maximum. Example:

<code>va_eventd.@forwarding[0].severity=debug-error</code>

<table><tr><td>debug</td><td>minimum severity</td></tr><tr><td>info</td><td></td></tr><tr><td>notice</td><td></td></tr><tr><td>warning</td><td></td></tr><tr><td>error</td><td></td></tr><tr><td>critical</td><td></td></tr><tr><td>alert</td><td></td></tr><tr><td>emergency</td><td>maximum severity</td></tr></table></td></tr>
<tr><td>Web: Maximum Severity

UCI: va_eventd.@forwarding[0].severity

Opt: severity</td><td>Defines the maximum event severity. The maximum event severity is EMERGENCY. Events generated within the minimum and maximum event severity will be matched.

The UCI command for specifying minimum and maximum severity is the same and is entered with two parameters using a dash (-) separator minimum-maximum. Example:

<code>va_eventd.@forwarding[0].severity=debug-error</code>

<table><tr><td>debug</td><td>minimum severity</td></tr><tr><td>info</td><td></td></tr><tr><td>notice</td><td></td></tr><tr><td>warning</td><td></td></tr><tr><td>error</td><td></td></tr><tr><td>critical</td><td></td></tr><tr><td>alert</td><td></td></tr><tr><td>emergency</td><td>maximum severity</td></tr></table></td></tr>
<tr><td>Web: Target

UCI: va_eventd.@forwarding[0].target

Opt: target</td><td>Defines the event destination to forward the event to. All configured event destinations will be displayed.</td></tr>
</table>

39.6. 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, enter:

```
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, enter:

```
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.@forwarding[0]=forwarding
va_eventd.@forwarding[0].enabled=1
```

Or using package options:

```
config forwarding
option enabled '1'
```

39.6.1. 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 (<.from%3Dfrom@example.com>)
va_eventd.@target[2].to=to@example.com (<.to%3Dto@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
```

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'
```

39.7. Event System Diagnostics

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	Login SSH	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	LoginConsole	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 IKE %{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 established
wifi	1	WiFiConnectedToAp	notice	WiFi %{p1} connected to AP %{p2}
wifi	2	WiFiDisconnectedFromAP	notice	WiFi %{p1} disconnected from AP
wifi	3	WiFiStationAttached	notice	WiFi station %{p2} connected to ..
wifi	4	WiFiStationDetached	notice	WiFi station %{p2} disconnected ..
wifi	5	WiFiStationAttachFailed	notice	WiFi station %{p2} failed to con..
ppp	1	LinkUp	informat	PPP for interface %{p2} (protoco..

pp	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	nformat	NTP adjust by %{p1}
ntp	3	QueryTimeout	warning	NTP query to %{p1} timed out. Ne..
ntp	4	QueryFailed	warning	NTP query failed: %{p1}

40. Configuring Data Usage Monitor

Merlin 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.

NOTICE

DISCLAIMER: data usage statistics calculated by this data usage feature are best estimates and may vary from the mobile carrier statistics that are used for billing. Westermo 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.

Configuration Package Used

Package	Sections
procrustes	limit

40.1. Configuring Data Using The Web Interface

Select **Services -> Data Usage**. The Data Usage page appears.

The data usage page

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.

The data usage configuration page

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

40.2. Configuring Data Using Command Line

Data usage is configured under the **procrustes** package **/etc/config/procrustes**.

By default, all limit instances are named 'limit', and are 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, enter:

```
config limit
option enabled '1'
```

However, to better identify instances, 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.

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
```

Procrustes using package options

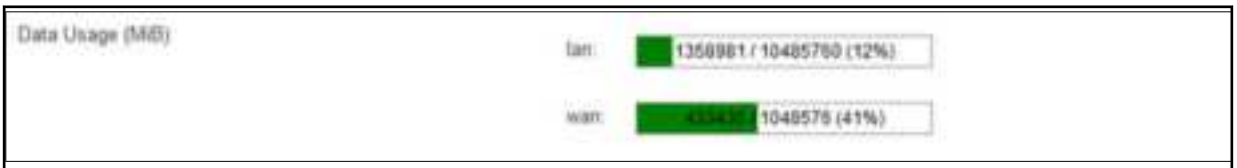
```
root@VA_router:~# uci export procrustes

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'
```

40.3. Data Usage Diagnostics

Data Usage Status using the Web Interface

Select **Status -> Overview**. The Status page appears.

To check current data usage, scroll to **Network -> Data Usage (MiB)** row. Data usage is presented as progress bar.



The data usage status progress bar

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

Viewing Data Usage

The router has a 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, to check total data usage, enter:

```
root@VA_router:~# cat /var/state/procrustes

procrustes.lan.total_bytes=215780

procrustes.wan.total_bytes=433436
```

Additional Debugging Commands

Additional useful debugging commands via the command line are described in the table below.

Diagnostic Command	Description
`logread | grep procrustes`	Shows logs related to "procrustes" only
`is /root/procrustes/sim_blacklist/`	Shows list of blacklisted SIM iccids

41. Configuring Terminal Server

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: the terminal server makes a TCP connection to external TCP server.
- UDP endpoint: the terminal server forwards data between a UDP stream and a serial port.

Configuration packages used

Package	Sections
tservd	main
	port

41.1. 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 sections:

- Main Settings are to enable the terminal server, syslog settings, and to enable log setting.
- The 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.

Configure Main Settings

The terminal server main settings page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enable

UCI: tservd.main.enable

Opt: enable</td><td>Enables Terminal Server on the router.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Debug Enable

UCI: tservd.main.debug_ev_enable

Opt: debug_ev_enable</td><td>Enables detailed debug logging.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Syslog severity

UCI: tservd.main.log_severity

Opt: log_severity</td><td>Determines the syslog level. Events up to this priority will be logged.

<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: Log RX-TX

UCI: tservd.main.debug_rx_tx_enable

Opt: debug_rx_tx_enable</td><td>Enables logging data transfers.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
</table>

41.1.1. Configuring Port Settings

The Port Settings section is divided into 3 sub-sections:

- General
- Serial
- Network

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.

Port Settings

PORT1

General | Serial | Network

Enable — enable port

Network Forwarding Buffer Size — 256 — Forwarding buffer size (serial to network)

Network Forwarding Timeout (ms) — 30 — Forwarding timeout in milliseconds (serial to network)

Network Forwarding timer mode — idle — Forwarding timer mode (serial to network)

Serial Forwarding Buffer Size — 0 — Forwarding buffer size (network to serial)

Serial Forwarding Timeout (ms) — 20 — Forwarding timeout in milliseconds (network to serial)

Serial Forwarding timer mode — idle — Forwarding timer mode (network to serial)

Proxy mode — enable proxy mode

Disable remote client's local echo (Telnet option)

Telnet COM port control (RFC2217)

Enable HDLC Pseudowire over UDP (RFC4618)

Serial receive debug log size — 0 — bytes (0=disable)

Serial transmit debug log size — 0 — bytes (0=disable)

The port settings general section

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Enable

UCI: tservd.@port[0].enable

Opt: enable</td><td>Enables terminal server port.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Network Forwarding Buffer Size

UCI: tservd.@port[0].fwd_buffer_size

Opt: fwd_buffer_size</td><td>Forwarding buffer size in bytes (serial to network).

<table><tr><td>Default:</td><td>256 bytes</td></tr><tr><td>Range</td><td>0-2048</td></tr></table></td></tr>
<tr><td>Web: Network Forwarding Timeout(ms)

UCI: tservd.@port[0].fwd_timeout

Opt: fwd_timeout</td><td>Forwarding timeout in milliseconds (serial to network).

<table><tr><td>Default:</td><td>30ms</td></tr><tr><td>Range</td><td>0-10000</td></tr></table></td></tr>
<tr><td>Web: Network Forwarding Timer Mode

UCI: tservd.@port[0].fwd_timer_mode

Opt: fwd_timer_mode</td><td>Forwarding timer mode (serial to network).

<table><tr><td>Default: Idle</td><td>Timer is restarted on each received data.</td></tr><tr><td>Aging</td><td>Timer is started on the first Rx.</td></tr></table></td></tr>
<tr><td>Web: Serial Forwarding Buffer Size

UCI: tservd.@port[0].sfwd_buffer_size

Opt: sfwd_buffer_size</td><td>Forwarding buffer size in bytes (network to serial).

Set to 0 to use maximum possible network Rx buffer size.

<table><tr><td>Default:</td><td>20ms</td></tr><tr><td>Range</td><td>0-10000</td></tr></table></td></tr>
<tr><td>Web: Serial Forwarding Timeout (ms)

UCI: tservd.@port[0].sfwd_timeout

Opt: sfwd_timeout</td><td>Forwarding timeout in milliseconds (network to serial).

Set to 0 to forward to serial immediately.

<table><tr><td>Default:</td><td>20ms</td></tr><tr><td>Range</td><td>0-10000</td></tr></table></td></tr>
<tr><td>Web: Serial Forwarding Timer Mode

UCI: tservd.@port[0].sfwd_timer_mode

Opt: sfwd_timer_mode</td><td>Forwarding timer mode (network to serial).

<table><tr><td>Default: Idle</td><td>Timer is restarted on each received data.</td></tr><tr><td>Aging</td><td>Timer started on the first Rx.</td></tr></table></td></tr>
<tr><td>Web: Proxy Mode

UCI: tservd.@port[0].proxy_mode

Opt: proxy_mode</td><td>Defines if a special proxy mode should be configured to allow ‘hijacking’ of the terminal server. It allows a connection to be made from a remote location and redirect terminal server data temporarily for troubleshooting.

When enabled, a TCP proxy server is started which listens for an incoming TCP connection from a remote peer.

Once an incoming new TCP connection on the proxy server TCP port is accepted:

• The existing terminal server TCP client connection is disconnected.
• The terminal server automatically reconnects the TCP client side but this time to the local loopback address 127.0.0.1 and to the local proxies TCP port number.
• When the proxy server has both local and remote TCP sessions connected it simply forwards the data between the two connections, taking into account the flow control.
• When either side TCP socket closes, the main terminal server client reconnects to the normal IP destination and the server proxy returns to listening for another connection from the far end.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Disable Remote Client’s Local Echo (Telnet option)</td><td>Set to 1 to send IAC WILL ECHO Telnet option to remote client forcing it to disable local echo. For server mode only.</td></tr>
</table>

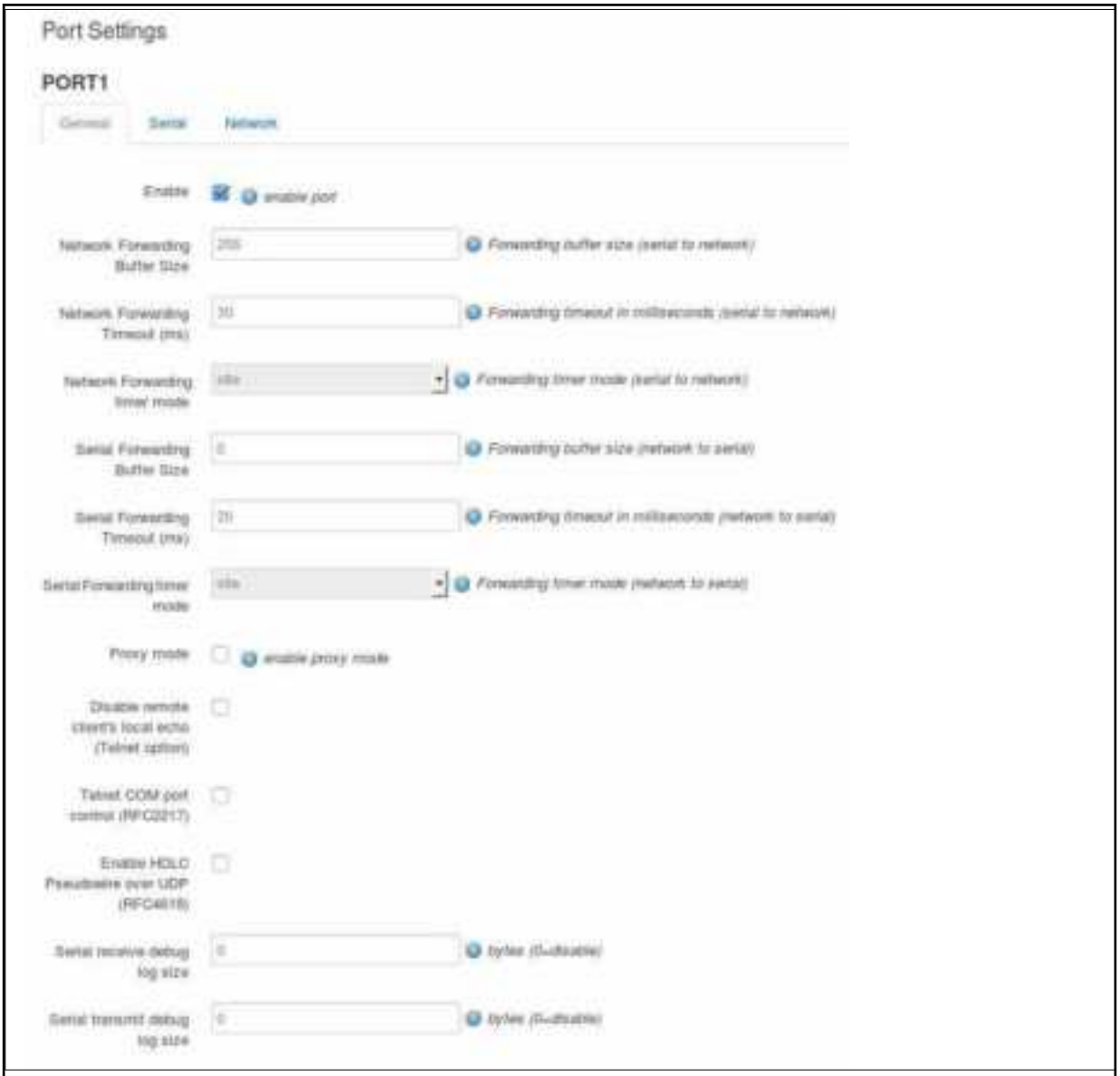
UCI: tservd.@port[0].disable_echo Opt: disable_echo	Default: 0 Disabled 1 Enabled
Web: Telnet COM Port Control UCI: tservd.@port[0].com_port_control Opt: com_port_control	Set to **1** to enable support for Telnet COM port control (RFC2217). Default: 0 Disabled 1 Enabled
Web: Enable HDLC Pseudowire over UDP (RFC4618) UCI: tservd.@port[0].hdlc_pw_enabled Opt: hdlc_pw_enabled	Set to **1** to enable HDLC pseudowire over UDP support based on RFC4618. Requires Transport Mode (udpmode) to be enabled. Default: 0 Disabled 1 Enabled
Web: Serial Receive Debug Log Size UCI: tservd.@port[0].serialRxLogSize Opt: serialRxLogSize	Configures serial receive log size in bytes and enables receive data logging. Default: 0 Disabled 1 Enabled
Web: Serial Transmit Debug Log Size UCI: tservd.@port[0].serialTxLogSize Opt: serialTxLogSize	Configures serial transmit log size in bytes and enables transmit data logging. Default: 0 Disabled 1 Enabled

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.

The figure below shows the options available if you have selected RS232 mode.



The serial section fields (port mode RS232)

The figure below shows the options available if you have selected RS485 mode.

The serial section fields (port mode RS485)

The figure below shows the options available if you have selected X.21 mode.

The serial section fields (port mode X.21)

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Device

UCI: tservd.@port[0].devName

Opt: devName</td><td>Serial device name.

<table><tr><td>Default: /dev/ttySC0</td><td>serial port 1</td></tr><tr><td>/dev/ttySC1</td><td>serial port 2</td></tr><tr><td>/dev/ttySC2</td><td>serial port 3</td></tr><tr><td>/dev/ttySC3</td><td>serial port 4</td></tr></table></td></tr>
<tr><td>Web: Port mode

UCI: tservd.@port[0].port_mode

Opt: port_mode</td><td>Sets the serial interface mode.

<table><tr><td>Default: rs232</td><td>RS232 mode</td></tr><tr><td>rs485hdx</td><td>RS485 2-wire half-duplex mode in which the transmitter drives the RTS.</td></tr><tr><td>rs485fdx</td><td>RS485 4-wire full-duplex mode.</td></tr><tr><td>v23</td><td>Uses V.23 leased line card driver.</td></tr><tr><td>x21</td><td>Uses USB serial card in sync mode.</td></tr></table></td></tr>
<tr><td>Web: GPIO Control

UCI: tservd.@port[1].serial_mode)gpio_control

Opt: serial_mode_gpio_control</td><td>Enables or disables software control of the port mode between RS232 and RS485. Applies only to port 1 (ttySC1) and not to port 0.

Note: the port mode is set with the option port mode described above.

<table><tr><td>Default: 0</td><td>Port mode is configurable by hardware settings and is not user configurable. Set to 0 for port 0.</td></tr><tr><td>1</td><td>Enabled. Port mode is configurable by software settings. This is applicable to serial port 1 on devices that are capable of RS485.</td></tr></table></td></tr>
<tr><td>Web: Speed (bps)

UCI: tservd.@port[0].speed

Opt: speed</td><td>Serial device speed in baud (bps).

<table><tr><td>Default:</td><td>9600</td></tr><tr><td>Range</td><td>115200; 57600; 38400; 19200; 9600; 4800; 2400; 1800; 1200; 600; 300; 200; 150; 134; 110; 75; 50</td></tr></table></td></tr>
<tr><td>Web: Word size

UCI: tservd.@port[0].wsize

Opt: wsize</td><td>Serial device word size.

<table><tr><td>Default:</td><td>8</td></tr><tr><td>Range</td><td>5-8</td></tr></table></td></tr>
<tr><td>Web: Parity

UCI: tservd.@port[0].parity

Opt: parity</td><td>Serial device parity

<table><tr><td>Default: 0</td><td>None</td></tr><tr><td>1</td><td>Even</td></tr><tr><td>2</td><td>odd</td></tr><tr><td>3</td><td>Space</td></tr></table></td></tr>
<tr><td>Web: Stop Bits

UCI: tservd.@port[0].stops

Opt: stops</td><td>Serial device number of stop bits.

<table><tr><td>Default: 1</td><td></td></tr><tr><td>Range</td><td>1-2</td></tr></table></td></tr>
<tr><td>Web: Flow Control

UCI: tservd.@port[0].fc_mode

Opt: fc_mode</td><td>Serial flow control mode.

<table><tr><td>Default: 0</td><td>None</td></tr><tr><td>1</td><td>RTS/CTS</td></tr><tr><td>2</td><td>XON/XOFF</td></tr></table></td></tr>
<tr><td>Web: RS485 Termination

UCI: tservd.@port[0].rs485_line_termination

Opt: rs485_line_termination</td><td>Enables or disables RS485 termination. Applies only if port mode is set to RS485.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Auto RTS Invert

UCI: tservd.@port[0].rtsinvert</td><td>Invert RTS in auto-RTS mode, if port mode is set to RS485.

<table><tr><td>Default: 0</td><td>Disabled</td></tr></table></td></tr>
</table>

<table>
<tr><td>Opt: rtsinvert</td><td><table><tr><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>Web: Keep Serial Port Always Open

UCI: tservd.@port[0].tty_always_open

Opt: tty_always_open</td><td>Keep serial port always open.</td></tr>
<tr><td>Web: RS232 Half Duplex

UCI: tservd.@port[0].hd_mode

Opt: hd_mode</td><td>Defines whether to enable special mode in the asynchronous serial driver for communication to an externally connected V.23 half-duplex modem.

Note: this setting does not enable half-duplex mode in the serial hardware of the router.

<table><tr><td>Default</td><td>30ms</td></tr></table></td></tr>
<tr><td>Web: RTS Timeout

UCI: tservd.@port[0].rts_timeout

Opt: rts_timeout</td><td>In RS232 half-duplex mode, time in milliseconds between raising RTS and enabling the transmitter. For use with an externally connected V.23 modem.

<table><tr><td>Default</td><td>30ms</td></tr></table></td></tr>
<tr><td>Web: POST RTS Timeout

UCI: tservd.@port[0].post_rts_timeout

Opt: post_rts_timeout</td><td>In RS232 half-duplex mode, sets the time in milliseconds between dropping RTS (transmission finished) and enabling the receiver. For use with externally connected V.23 modem.

<table><tr><td>Default</td><td>20ms</td></tr></table></td></tr>
<tr><td>Web: Synchronous mode

UCI: tservd.@port[0].sync mode

Opt: sync mode</td><td>Defines synchronous frame mode. This setting is only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default: hdlc</td><td>HDLC frame mode</td></tr><tr><td>transp</td><td>Transparent mode</td></tr></table></td></tr>
<tr><td>Web: Use CRC32

UCI: tservd.@port[0].sync_crc32

Opt: sync_crc32</td><td>Defines whether to use CRC32 or CRC16 in HDLC mode. This setting is only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default: 0</td><td>Use CRC16</td></tr><tr><td>1</td><td>Use CRC32</td></tr></table></td></tr>
<tr><td>Web: DTR control mode

UCI: tservd.@port[0].dtr_control_mode

Opt: dtr_control_mode</td><td>Defines DTR line control modes. This setting is only displayed if an Atmel USB serial card is enabled and port mode is X21.

<table><tr><td>Default: auto</td><td>Defines DTR line control modes. This setting is only displayed if an Atmel USB serial card is enabled and port mode is X21.</td></tr><tr><td>on</td><td>DTR always on</td></tr><tr><td>off</td><td>DTR always off</td></tr><tr><td>app</td><td>DTR controlled by the application</td></tr><tr><td>ontx</td><td>In HDLC mode DTR is on during frame transmission.</td></tr></table></td></tr>
<tr><td>Web: RTS control mode

UCI: tservd.@port[0].rts_control_mode

Opt: rts_control_mode</td><td>Defines RTS line control modes. Only displayed if an Atmel USB serial card is enabled and port mode is X21.

<table><tr><td>Default: auto</td><td>Defines RTS line control modes. This setting is only displayed if an Atmel USB serial card is enabled and port mode is X21.</td></tr><tr><td>on</td><td>RTS always on</td></tr><tr><td>off</td><td>RTS always off</td></tr><tr><td>app</td><td>RTS controlled by the application</td></tr><tr><td>ontx</td><td>In HDLC mode RTS is on during frame transmission.</td></tr></table></td></tr>
<tr><td>Web: Synchronous rate

UCI: tservd.@port[0].sync_speed

Opt: sync_speed</td><td>Defines the synchronous speed in bps. Set to 0 for external clock. If not set to 0, an internal clock is used. This setting is only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default: 64000</td><td>64 kbps</td></tr><tr><td>Range</td><td>2048000; 1024000; 768000; 512000; 384000; 256000; 128000; 19200; 9600</td></tr></table></td></tr>
</table>

<table>
<tr><td>Web: Invert receive clock

UCI: tservd.@port[0].sync_invert_rxclk

Opt: sync_invert_rxclk</td><td>Defines receive clock inversion. Normal clock data is sampled on falling edge. Inverted clock data is sampled on rising edge. This setting is only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default: 0</td><td>Normal</td></tr><tr><td>1</td><td>Invert</td></tr></table></td></tr>
<tr><td>Web: Invert transmit clock

UCI: tservd.@port[0].sync_invert_txclk

Opt: sync_invert_txclk</td><td>Defines transmit clock inversion. Normal clock data transmitted on falling edge. Inverted clock data transmitted on rising edge. Only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default: 0</td><td>Normal</td></tr><tr><td>1</td><td>Invert</td></tr></table></td></tr>
<tr><td>Web: RX MSBF

UCI: tservd.@port[0].sync_rx_msbf

Opt: sync_rx_msbf</td><td>Defines whether most significant bit is received first. This setting is only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default: 0</td><td>Receive least significant bit first</td></tr><tr><td>1</td><td>Receive most significant bit first</td></tr></table></td></tr>
<tr><td>Web: TX MSBF

UCI: tservd.@port[0].sync_tx_msbf

Opt: sync_tx_msbf</td><td>Defines whether most significant bit is transmitted first. This setting is only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default: 0</td><td>Transmit least significant bit first</td></tr><tr><td>1</td><td>Transmit most significant bit first</td></tr></table></td></tr>
<tr><td>Web: RX data delay

UCI: tservd.@port[0].sync_rxdata_dly

Opt: sync_rxdata_dly</td><td>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.

<table><tr><td>Default</td><td>0</td></tr></table></td></tr>
<tr><td>Web: TX data delay

UCI: tservd.@port[0].sync_txdata_dly

Opt: sync_txdata_dly</td><td>Defines the number of bit positions to delay the output of data from the detecting clock edge. This setting is only displayed if an Atmel USB serial card is enabled.

<table><tr><td>Default</td><td>0</td></tr></table></td></tr>
<tr><td>Web: Dual X.21 card bit reverse

UCI: tservd.@port[0].bit_reverse

Opt: bit_reverse</td><td>Enables bit reversal of all bits in 8 byte word during transmission.

<table><tr><td>Default: 0</td><td>Normal</td></tr><tr><td>1</td><td>Reverse</td></tr></table></td></tr>
<tr><td>Web: Dual X.21 card DTE TT Invert

UCI: tservd.@port[0].dte_tt_inv

Opt: dte_tt_inv</td><td>Enables X.21 TT clock signal inversion.

<table><tr><td>Default: 0</td><td>Normal</td></tr><tr><td>1</td><td>Invert</td></tr></table></td></tr>
<tr><td>Web: Dual X.21 card DCE TCLK

Invert UCI: tservd.@port[0].dce_tclk_inv

Opt: dce_tclk_inv</td><td>Enables X.21 DCE TCLK signal inversion.

<table><tr><td>Default: 0</td><td>Normal</td></tr><tr><td>1</td><td>Invert</td></tr></table></td></tr>
<tr><td>Web: Dual X.21 card DCE RCLK Invert

UCI: tservd.@port[0].dce_rclk_inv

Opt: dce_rclk_inv</td><td>Enables X.21 DCE RCLK signal inversion.

<table><tr><td>Default: 0</td><td>Normal</td></tr><tr><td>1</td><td>Invert</td></tr></table></td></tr>
<tr><td>Web: Dual X.21 card CLK Invert

UCI: tservd.@port[0].x21_clk_invert

Opt: x21_clk_invert</td><td>Enables X.21 DCE CLK signal inversion.

<table><tr><td>Default: 0</td><td>Normal</td></tr><tr><td>1</td><td>Invert</td></tr></table></td></tr>
<tr><td>Web: Dual X.21 card RX data delay

UCI: tservd.@port[0] x21_data_delay</td><td>Sets X.21 card RX data delay in number of bit positions.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-7</td></tr></table></td></tr>
</table>

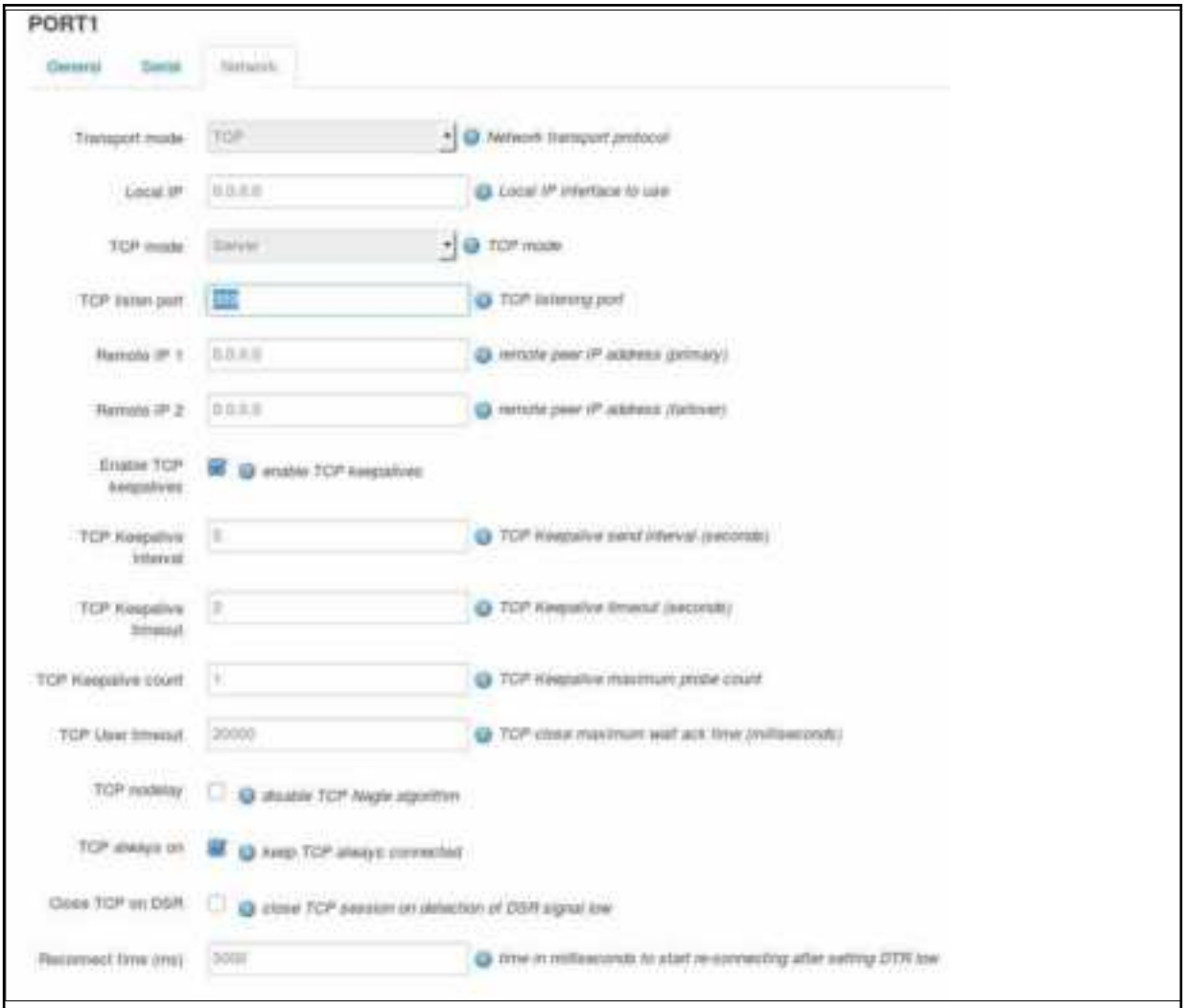
Opt: x21_data_delay	
Web: n/a UCI: tservd.@port[0].sync_tx_idle Opt: sync_tx_idle	Defines the value of idle character (decimal) to transmit in case of transmit underrun. In HDLC mode, this configures inter-frame fill. Default: 0 — Transmit 0 (in HDLC mode) 126 — Transmit flags (in HDLC mode) 255 — Transmit 1 (in HDLC mode) Range — 0-255
Web: n/a UCI: tservd.@port[0].v23_inband_carrier_signalling Opt: v23_inband_carrier_signalling	Enables signalling of carrier by sending special characters Default: 0 — Disabled 1 — Enabled
Web: n/a UCI: tservd.@port[0].v23_inband_carrier_on_char Opt: v23_inband_carrier_on_char	Defines the character decimal to signal remote carrier on. Default — 255 Range — 0-255
Web: n/a UCI: tservd.@port[0].v23_tx_gain Opt: v23_tx_gain	Defines the transmit gain for v23 mode. Default — 2 Range — 0-255
Web: n/a UCI: tservd.@port[0].v23_rx_loss Opt: v23_rx_loss	Defines the receive loss for v23 mode. Default: 1 — Receives samples divided by 1
Web: n/a UCI: tservd.@port[0].v23_rts_to_cts_delay Opt: v23_rts_to_cts_delay	Defines the v23 modem RTS to CTS delay in milliseconds Default — 20
Web: n/a UCI: tservd.@port[0].v23_is_four_wire Opt: v23_is_four_wire	Defines the V23 modem LIM operation. Default: 0 — 2-wire 1 — 4-wire
Web: n/a UCI: tservd.@port[0].v23_tx_timeout Opt: v23_tx_timeout	Defines the V23 modem receive echo suppression timeout in milliseconds. Default — 20
Web: n/a UCI: tservd.@port[0].v23_tx_rampdown Opt: v23_tx_rampdown	Defines the time, in milliseconds, it takes the V23 transmitter to rampdown carrier from peak to zero. Default — 30
Web: n/a UCI: tservd.@port[0].v23_tx_maxfill Opt: v23_tx_maxfill	Defines the maximum transmit queue fill level in bytes. Default — 127 Range — 0-255

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.



The port settings network fields (TCP server mode)

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

<table>
<tr><td>Opt: tcp_keepalive_count</td><td><table><tr><td>Default</td><td>1</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: TCP User Timeout

UCI: tservd.@port[0].tcp_user_timeout

Opt: tcp_user_timeout</td><td>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.

<table><tr><td>Default: 20000</td><td>20 seconds</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: TCP Nodelay

UCI: tservd.@port[0].tcp_nodelay

Opt: tcp_nodelay</td><td>Sets TCP to delay behaviour. Only displayed if transport mode is TCP.

<table><tr><td>Default: 0</td><td>Normal operation</td></tr><tr><td>1</td><td>Disable TCP Nagle algorithm. Only display if transport mode is TCP.</td></tr></table></td></tr>
<tr><td>Web: TCP Always on

UCI: tservd.@port[0].tcp_always_on

Opt: tcp_always_on</td><td>Keep TCP session always connected. Only displayed if transport mode is TCP and client mode is enabled.</td></tr>
<tr><td>Web: Close TCP on DSR

UCI: tservd.@port[0].close_tcp_on_dsr

Opt: close_tcp_on_dsr</td><td>Close TCP session on detection of DSR signal low. Only displayed if Transport Mode is TCP and client mode is enabled.

<table><tr><td>Default: 0</td><td>Disabled. TCP connection/UDP session is initiated on detecting high state on the DSR interface signal.</td></tr><tr><td>1</td><td>Enabled. If it disconnects in the established state the TCP connection/UDP session is re-initiated.</td></tr></table></td></tr>
<tr><td>Web: Reconnect Time (ms)

UCI: tservd.@port[0].disc_time_ms

Opt: disc_time_ms</td><td>Time in milliseconds to start reconnecting after setting DTR low.

<table><tr><td>Default: 50000</td><td>5 seconds</td></tr><tr><td>Range</td><td>0-10000</td></tr></table></td></tr>
<tr><td>Web: UDP Keepalive Interval

UCI: tservd.@port[0].udpKaIntervalMs

Opt: udpKaIntervalMs</td><td>Defines time in milliseconds to send UDP keepalives (empty UDP packets) when no data to send. Only displayed if transport mode is UDP.

<table><tr><td>Default: 0</td><td>Disabled</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: UDP Keepalive Count

UCI: tservd.@port[0].udpKaCount

Opt: udpKaCount</td><td>Defines the maximum number of remote UDP keepalives not received before UDP stream is considered broken. Only displayed if transport mode is UDP.

<table><tr><td>Default</td><td>3</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: local UDP Port

UCI: tservd.@port[0].udpLocalPort

Opt: udpLocalPort</td><td>Local UDP port used by terminal server. Only displayed if transport mode is UDP.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: remote UDP Port

UCI: tservd.@port[0].udpRemotePort

Opt: udpRemotePort</td><td>Remote UDP port used by terminal server. Only displayed if transport mode is UDP.</td></tr>
</table>

41.2. Configuring 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
```

Configuring 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'
```

41.3. Configuring Terminal Server DSR Signal Management Network

On the IP network side, the terminal server can operate in one of three modes:

- TCP Client
- TCP Server
- UDP

Based on the chosen network configuration, the DSR behaviour may vary.

DSR Signal Behaviour in TCP Client Mode

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 detecting DSR down signal and closes the established TCP connection.

If option `close_tcp_on_dsr` is disabled then detecting DSR down does not affect the TCP connection.

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.

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.

TCP Connection Re-initiation

After TCP connection 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.

DSR Signal Behaviour in TCP Server Mode

TCP Connection Initiation at Startup

After a short startup delay, the terminal server starts listening for an incoming TCP connection from the remote peer.

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.

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.

DSR Signal Behaviour in UDP Mode

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.

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.

UDP Session Reset

After UDP session clearing the terminal server takes action to reset up 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.

41.4. Serial Mode GPIO Control

On some models of Merlin routers it is possible to change the physical transmission mode between RS232 and RS485. This is only applicable to the second serial port on the routers: /dev/ttySC1.

To enable `serial_mode_gpio_control` set the option to **1**.

Use the portmode option in addition to `serial_mode_gpio_control` to select between RS232, RS485 full duplex, RS485 half duplex, X.21 and V.23.

Checking the Current serial_mode_gpio_control

To check if terminal server is running, enter the following command:

```
root@VA_router:~# uci show tservd | grep serial_mode_gpio_control
```

The output of the above command will look similar to the example below if serial_mode_gpio_control is enabled for the second serial port.

```
tservd.port0.serial_mode_gpio_control=0

tservd.port1.serial_mode_gpio_control=1
```

41.5. Terminal Server Diagnostics

The tservd process has to be running otherwise diagnostics options for terminal server will not be available.

Checking the Terminal Server Process

```
root@VA_router:~# -fl tservd

1264 root        1032 S tservd
```

If Terminal Server is running it will be shown with its process ID.

Terminal Server Statistics

To view 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)
```

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
```

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
```

Terminal Server Advanced Debugging

To view terminal server advanced debug commands for the terminal server, enter:

```
root@VA_router:~# tserv

=== Termserv diagnostics. Command syntax: ===

tserv show stats - show statistics

tserv clear stats - clear statistics

tserv show serial - show serial interface status

tserv send serial0 &lt;data&gt;- 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 quit - terminate termserv process
```

42. Configuring Terminal Package

Terminal package is used to automatically add entries to getty to inittab for extra incoming console/terminal connections.

Configuration Packages Used

Package	Sections
terminal	terminal

Terminal package is not available to configure using the web interface.

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

42.1. 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
```

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'
```

42.2. Terminal Diagnostics

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
```

43. SCADA IEC 104 Gateway

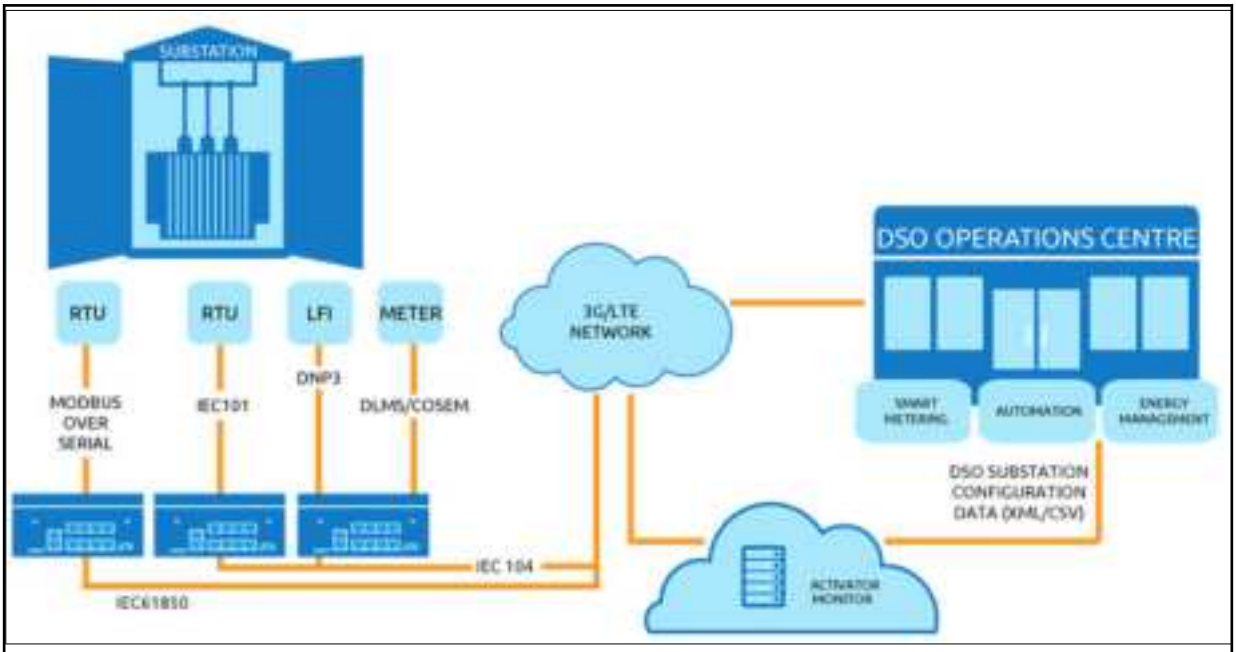
Supervisory control and data acquisition (SCADA) systems are used by industrial organisations and companies to control and monitor physical processes, examples of which are electricity transmission, gas and oil transportation in pipelines, water distribution and traffic lights control. Alarm handling is usually an important part of most SCADA implementations.

SCADA systems usually consist of:

- Supervisory computers
- Remote terminal units (RTUs)
- Programmable logic controllers)PLCs)

The IEC 104 gateway feature on the router is used for SCADA protocol conversion where the SCADA master is running IEC 104 protocol: ·

- IEC 104 to IEC 101 conversion (balanced and unbalanced)
- IEC 104 to DNP3
- IEC 104 to MODBUS (serial to TCP)
- IEC 612850 to IEC 101 unbalanced conversion



Example network for IEC 104 to RTU protocol conversion

Configuration for the above conversions is done in two parts:

- IEC 104 gateway (iecd package), and
- Terminal server (tservd package)

The IEC 104 gateway handles the protocol processing while the terminal server handles low level serial communication.

NOTE

The terminal server is not required for IEC 104 to Modbus TCP.

Configuration Packages Used

Package	Sections
iecd	main port point
tservd	main port

43.1. IEC 104 Gateway Configuration Using The Web Interface

In the top menu, select **Services -> IEC104 Gateway**. The IEC 104 gateway page appears.

The IEC 104 gateway configuration page

There are four sections in the IEC 104 Gateway page:

Section	Description
Main Settings	Enables the IEC 104 gateway.
Port Settings	Sets the IEC 104 SCADA master communication settings and the protocol methods used by the RTUs: • IEC 101 unbalanced or balanced • DNP3 • Modbus over serial • Modbus over TCP
IEC101 Links	Defines the IEC 101 slave links used in IEC 101 conversion. Each link is defined by a `config iec101link` section block. There is a maximum of 32 links supported. In IEC 101 unbalanced mode all of these links can be used. In IEC 101 balanced mode only one outstation per serial port is assumed since these will be point to point links.
Points	Configures the data point mappings. Note: there are no data point mappings in IEC104 to IEC101 conversion.

43.1.1. Main Settings

The IEC 104 gateway main settings configuration page

Web Field/UCI/Package Option	Description
Web: Enable UCI: iecd.main.enable Opt: enable	Enables IEC104 gateway. Default: 0 — Disabled 1 — Enabled

43.1.2. Port Settings

The port configuration will depend on the desired protocol conversion. There are 5 sections.

Section	Description
General	Enables an IEC104 port and selects the RTU protocol method.
IEC104	Defines the IEC104 gateway configuration for communication with the SCADA Master.
IEC101	Defines the IEC104 to IEC101 conversion parameters.
DNP3	Defines the IEC104 to DNP3 conversion parameters.
Modbus	Defines the IEC104 to MODBUS conversion parameters (Modbus over serial or Modbus over TCP).
Advanced	Defines logging and TCP keepalive options for all conversion methods.

In the Port Settings section, enter a text name that will be used for the iecd port section, for example, Port1. Select **Add**. The IECD port configuration options appear.

43.1.3. Port Settings: General

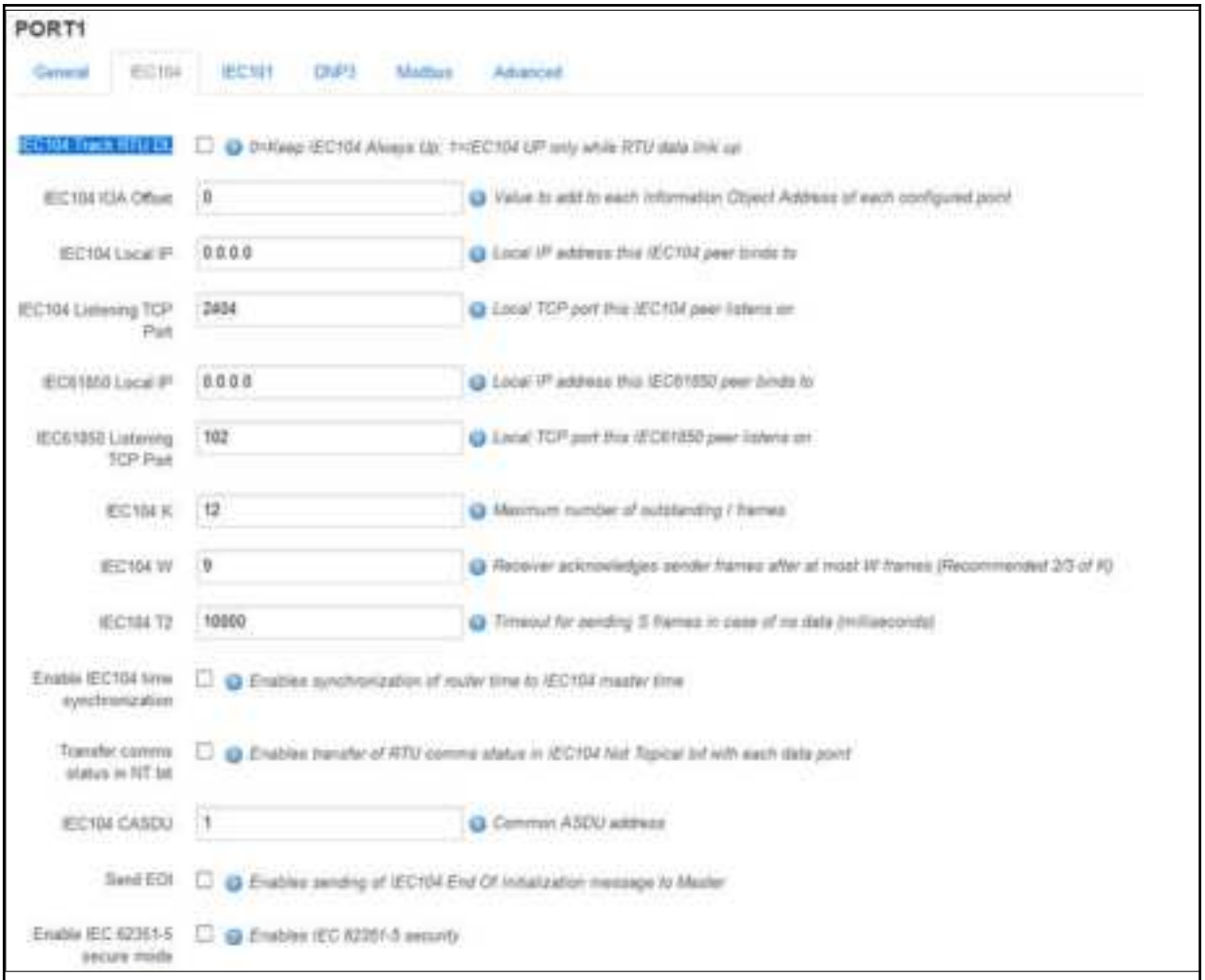
In this section you can configure general port settings. Check **Enable** to enable the port and select the appropriate RTU protocol from the Slave Protocol and Master Protocol drop-down menus.

The IEC 104 gateway port settings: general configuration page

Web Field/UCI/Package Option	Description
Web: Enable UCI: iecd.<port>.enable Opt: enable	Enables an IECD port.
Web: Slave Protocol UCI: iecd.<port>.slave_protocol Opt: slave_protocol	Defines the protocol method used by the SCADA master to connect to this router (acting as slave).
Web: Master Protocol UCI: iecd.<port>.master_protocol Opt: master_protocol	Defines the protocol method used by this router (acting as a master) to connect to the outstations.
Web: n/a UCI: iecd.<port>.pointmap_file Opt: pointmap_file	Defines the path to the points map file, for example: `/root/iecd/iecd_points1.csv`

43.1.4. Port Settings: IEC 104

In this section you can configure IEC104 settings.



The IEC 104 gateway port IEC 104 configuration page

Web Field/UCI/Package Option	Description
Web: IEC104 Track RTU DL UCI: iecd.<port>.iec104_track_rtu_dl Opt: iec104_track_rtu_dl	Defines whether the IEC 104 link follows the state of the RTU data link. Default: 0 — Always listens and accepts connection from the IEC 104 master. This means IEC 104 is always up independently of the RTU protocol. 1 — IEC 104 is up only while RTU data link is up. The IEC 104 socket is closed and IEC 104 will only start listening when RTU data link is up.
Web: IEC104 IOA Offset UCI: iecd.<port>.ioa_offset Opt: ioa_offset	Defines the value to add to each Information Object Address of each configured point. Default: 0
Web: IEC104 Local IP UCI: iecd.<port>.iec104_local_ip Opt: iec104_local_ip	Defines the local IP address this IEC 104 peer binds to. Default: 0.0.0.0 Blind to outgoing port.
Web: IEC104 Listening TCP Port UCI: iecd.<port>.iec104_local_tcpport Opt: iec104_local_tcpport	Defines the local TCP port this IEC 104 peer listens on. Default: 2404 Range: 1-65535
Web: IEC61850 Local IP UCI: iecd.<port>.iec61850_local_ip Opt: iec61850_local_ip	Defines the local IP address this IEC 61850 peer binds to. Default: 0.0.0.0 Blind to outgoing port. Range:
Web: IEC61850 Listening TCP Port UCI: iecd.<port>.iec61850_local_tcpport Opt: iec61850_local_tcpport	Defines the local TCP port this IEC 61850 peer listens on. Default: 102 Range: 1-655535
Web: IEC104 K UCI: iecd.<port>.iec104_k Opt: iec104_k	Defines the maximum number of outstanding I frames. Default: 12 Range:
Web: IEC104 W UCI: iecd.<port>.iec104_w Opt: iec104_w	Defines the number of frames after which the receiver will acknowledge. It is recommended that this value is 2/3 the value of IEC 104K. Default: 9 Range:
Web: IEC104 T2 UCI: iecd.<port>.iec104_t2 Opt: iec104_t2	Defines the timeout in milliseconds for sending S frames in case of no data. Default: 10000 milliseconds. Range:
Web: Enable IEC104 time synchronization UCI: iecd.<port>.iec104_sync_time Opt: iec104_sync_time	Enables synchronisation of router time to IEC 104 master time. Default: 1 — Enable synchronisation. 0 — Disable synchronisation.
Web: Transfer comms status in NT bit UCI: iecd.<port>.iec104_comms_status_nt Opt: iec104_comms_status_nt	Enables transfer of RTU comms status in IEC 104 Not Topical bit with each data point. Default: 1:
Web: IEC104 CASDU UCI: iecd.<port>.iec104_casdu	Defines IEC104 common ASDU address. Default: 1

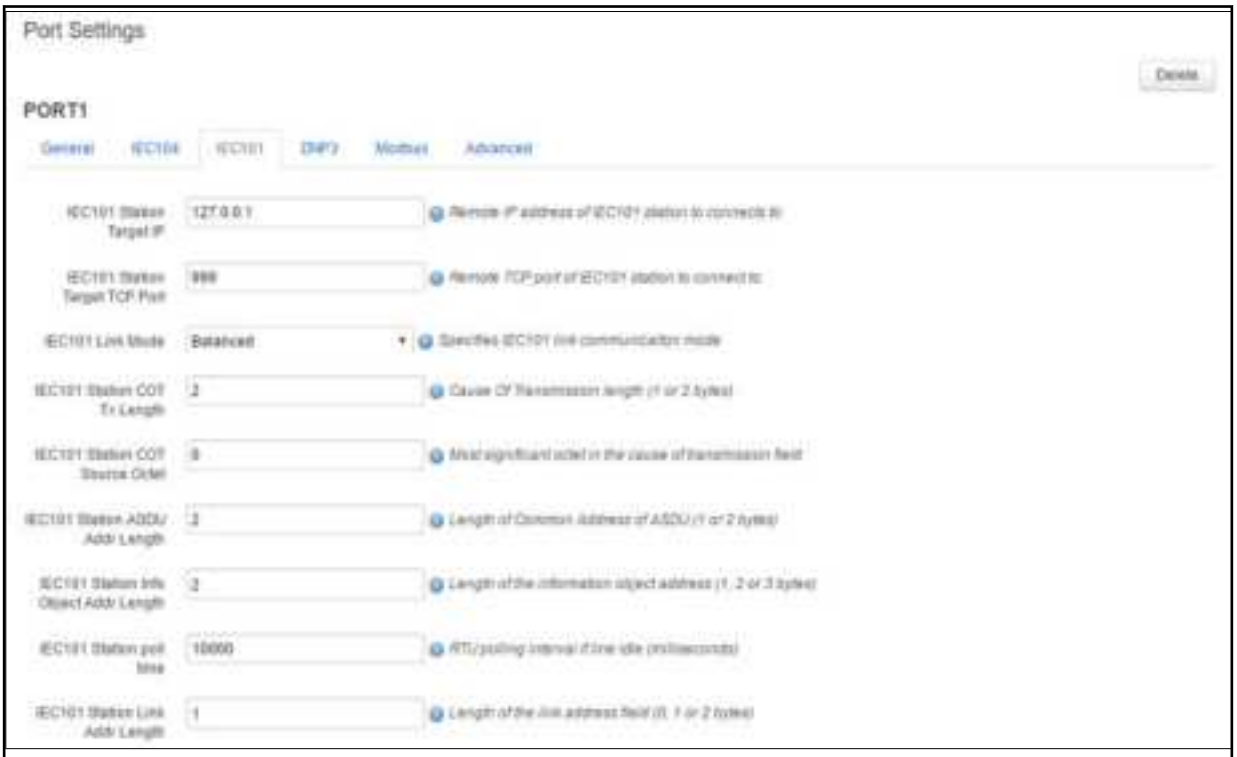
Web Field/UCI/Package Option	Description
Opt: iec104_casdu	Range
Web: Send EOI UCI: iecd.<port>.iec104_send_eoi Opt: iec104_send_eoi	Enables sending of IEC 104 End Of Initialisation message to the master. Default: 0 1
Web: Enable IEC 62351-5 secure mode UCI: iecd.<port>.iec104_secure_on Opt: iec104_secure_on	Enables IEC 62351-5 security. Default: 0 1
Web: n/a UCI: iecd. <port>.iec104_rtu_dl_start_dt Opt: iec104_rtu_dl_start_dt	Defines the start operation of the RTU data link. Default: 0 – The RTU data link is always started and established at startup and kept up. 1 – The RTU data link layer is started and established when IEC104 is up and the START DT message from the IEC104 master is received. When the RTU data link comes up: send START DT CONF to the IEC104 master.
Web: n/a UCI: iecd.<port>iec104_gi_resp_time Opt: iec104_gi_resp_time	Defines the time in milliseconds between sending successive general interrogation response messages. Default: 200 milliseconds Range 50-1000
Web: n/a UCI: iecd.<port>.iec104_txq_size Opt: iec104_txq_size	Defines the maximum size of transmit ASDU queue in the application layer (number of frames). Default: 128 Range 2-256
Web: n/a UCI: iecd.<port>.iec104_cmd_delay_time Opt: iec104_cmd_delay_time	Defines the maximum allowable received command age in milliseconds. If set to 0, control commands time verification is disabled. Default: 0 – 5000 milliseconds 1 – 1000-60000
Web: n/a UCI: iecd.<port>.iec104_fsm_debug_on Opt: iec104_fsm_debug_on	Enables the log for IEC 104 state transitions and events. Default: 0 – Enable 1 – Disable
Web: n/a UCI: iecd.<port>.iec104_dump_data Opt: iec104_dump_data	Enables RX/TX Hex dump. Default: 0 – Enable 1 – Disable
Web: n/a UCI: iecd.<port>.iec104_trace_on Opt: iec104_trace_on	Enables protocol tracing. Default: 0 – Enable 1 – Disable

43.1.5. Port Settings: IEC 101

The IEC 104 to IEC 101 conversion feature on the router allows converting commands in the control direction, and the responses and process data in the monitor direction, between the SCADA master running the IEC 104 protocol and the remote RTUs running IEC101 protocol over a serial interface.

IEC 104 to IEC 101 conversion can be configured for two modes:

IEC 101 Mode	Description
Unbalanced	In IEC 101 unbalanced mode, the router supports communication of up to 32 IEC 101 slaves connected onto the same serial interface.
Balanced	IEC 101 balanced mode is used in point-to-point configuration. That is, the router is communicating to a single IEC 101 outstation on the serial interface. Each peer, either the controlling station (Master) or controlled station (RTU) can initiate communication in balanced mode.



The IEC 104 gateway port IEC101 configuration page

Web Field/UCI/Package Option	Description
Web: IEC101 Station Target IP UCI: iecd.<port>.iec101_target_ip Opt: iec101_target_ip	Defines the remote IP address of the IEC101 station to connect to. Default: 127.0.0.1 Range:
Web: IEC101 Station Target TCP Port UCI: iecd.<port>.iec101_target_tcpport Opt: iec101_target_tcpport	Defines the remote TCP port of the IEC101 station to connect to. Default: 999 Range:
Web: IEC101 Link Mode UCI: iecd.<port>.iec101_mode Opt: iec101_mode	Defines the IEC101link communication mode. Default: Unbalanced Balanced
Web: IEC101 Station COT Tx Length UCI: iecd.<port>.iec101_cot_tx_length Opt: iec101_cot_tx_length	Defines the Cause of Transmission length (1 or 2 bytes). Default: 2 bytes Range:
Web: IEC101 Station COT Source Length UCI: iecd.<port>.iec101_cot_source_octet Opt: iec101_cot_source_octet	Defines the most significant octet in the Cause of Transmission field. Default: 0
Web: IEC101 Station ASDU Addr Length UCI: iecd.<port>.iec101_asdu_addrlen Opt: iec101_asdu_addrlen	Defines the length of Common Address of ASDU (1 or 2 bytes). Default: 2 bytes Range:
Web: IEC101 Station Info Object Addr Length UCI: iecd.<port>.iec101_info_obj_addrlen Opt: iec101_info_obj_addrlen	Defines the length of the Information Object Address (1, 2 or 3 bytes). Default: 2 bytes Range:
Web: IEC101 Station Poll Time UCI: iecd.<port>.iec101_data_polling_time Opt: iec101_data_polling_time	Defines the RTU polling interval in milliseconds if the line is idle. Default: 1000 milliseconds Range:
Web: IEC101 Link Addr Length UCI: iecd.<port>.iec101_link_addrlen Opt: iec101_link_addrlen	Defines the length of the link address field (0, 1 or 2 bytes). Default:1 bytes Range:
Web: n/a UCI: iecd.<port>.iec101_ack_delay Opt: iec101_ack_delay	Defines the time to wait in milliseconds for an IEC101 ACK. Default: 0 seconds Range:
Web: n/a UCI: iecd.<port>.iec101_frame_rsp_time Opt: iec101_frame_rsp_time	Defines the maximum number of milliseconds before resending an IEC101 frame. Default: 3 Range:
Web: n/a UCI: iecd.<port>.iec101_max_tx_retry Opt: iec101_max_tx_retry	Defines the maximum number of times to retry sending an IEC101 frame. Default: 3 Range:
Web: n/a UCI: iecd.<port>.iec101_txq_size	Defines the size of transmit ASDU queue (number of frames) in the IEC101 link layer.

Web Field/UCI/Package Option	Description
Opt: iec101_txq_size	Default: 128 Range:
Web: n/a UCI: iecd.<port>.iec101_send_spont_delay_acq Opt: iec101_send_spont_delay_acq	Defines whether to send DELAY ACQUISITION SPONTANEOUS message as part of 'Acquisition of Transmission Delay' procedure. Note: this option is used in the scenario where an IEC104 Master is talking to an IEC101 RTU. Default: 0 — Do not send DELAY ACQUISITION SPONTANEOUS message. 1 — Send DELAY ACQUISITION SPONTANEOUS message.
Web: n/a UCI: iecd.<port>.iec101_fsm_debug_on Opt: iec101_fsm_debug_on	Enables logging IEC104 state transitions and events. Default: 0 1
Web: n/a UCI: iecd.<port>.iec101_dump_data Opt: iec101_dump_data	Enables RX/TX Hex dump. Default: 0 1

43.1.6. Port Settings: DNP3

The IEC 104 to DNP3 conversion feature on the router allows converting commands in the control direction, and the responses and process data in the monitor direction, between the SCADA master running the IEC104 protocol and the remote RTU running DNP3 over serial protocol.

The IEC 104 gateway port DNP3 configuration page

Web Field/UCI/Package Option	Description
Web: DNP3 Station Target IP UCI: iecd.<port>.dnp3_target_ip Opt: dnp3_target_ip	Defines the remote IP address of the DNP3 station to connect to.
Web: DNP3 Station Target TCP Port UCI: iecd.<port>.dnp3_target_tcpport Opt: dnp3_target_tcpport	Defines the remote TCP port of the DNP3 station to connect to. Default: 999 Range:
Web: DNP3 Master Station Address UCI: iecd.<port>.dnp3_dl_srcaddr Opt: dnp3_dl_srcaddr	Defines the local (Master) DNP3 address. Default: 0 Range:
Web: DNP3 Outstation Address UCI: iecd.<port>.dnp3_dl_dstaddr Opt: dnp3_dl_dstaddr	Defines the remote (Outstation) DNP3 address. Default: 0 Range:
Web: Enable DNP3 Data Link Confirms UCI: iecd.<port>.dnp3_dl_cfrm_user_data Opt: dnp3_dl_cfrm_user_data	Enables DNP3 data link layer user data confirmations. Default: 0 Range:
Web: DNP3 Data Link Keep Alive UCI: iecd.<port>.dnp3_dl_keep_alive_int Opt: dnp3_dl_keep_alive_int	Defines the DNP3 data link keepalive interval in milliseconds (0 to disable). Default: 15000 milliseconds Range:
Web: DNP3 Frame Response Time UCI: iecd.<port>.dnp3_dl_frame_rsp_time Opt: dnp3_dl_frame_rsp_time	Defines the maximum amount of time in milliseconds to receive a frame acknowledgement from the DNP3 outstation. Default: 1000 milliseconds Range:
Web: DNP3 Maximum Frame Retry UCI: iecd.<port>.dnp3_dl_max_tx_retry Opt: dnp3_dl_max_tx_retry	Defines the maximum number of times to retry confirmed frame delivery to the DNP3 outstation.
Web: DNP3 Outstation Poll Time UCI: iecd.<port>.dnp3_app_poll_time Opt: dnp3_app_poll_time	Defines the DNP3 outstation poll time in milliseconds. Default: 3 Range:
Web: Enable DNP3 Unsolicited Responses UCI: iecd.<port>.dnp3_app_unsol_enable Opt: dnp3_app_unsol_enable	Enables DNP3 application level unsolicited responses. Default: 0 Disabled 1 Enabled
Web: Enable DNP3 Time Synchronization UCI: iecd.<port>.dnp3_app_sync_time Opt: dnp3_app_sync_time	Enables DNP3 time synchronisation. Default: 0 Disabled 1 Enabled
Web: n/a UCI: iecd.<port>.dnp3_dl_utxq_size Opt: dnp3_dl_utxq_size	Defines the size of DNP3 data link transmit unconfirmed service frame queue (number of frames). Default: 128 Range: 2-256
Web: n/a	Defines the size of DNP3 data link transmit confirmed service frame queue (number of frames).

Web Field/UCI/Package Option	Description
UCI: iecd.<port>.dnp3_dl_ctxq_size Opt: dnp3_dl_ctxq_size	Default: 128 Range: 2-256
Web: n/a UCI: iecd.<port>.dnp3_app_read_attr Opt: dnp3_app_read_attr	Enables reading DNP3 device attributes at the start of the session. This feature is useful for debugging and is not recommended for production.
Web: n/a UCI: iecd.<port>.dnp3_app_firstpoll_delay Opt: dnp3_app_firstpoll_delay	Defines initial timeout from start-up in milliseconds before performing the first DNP3 integrity poll. Default: 5000 milliseconds Range: 5000-65535
Web: n/a UCI: iecd.<port>.dnp3_app_evpoll_time Opt: dnp3_app_evpoll_time	Defines DNP3 outstation event polling interval in milliseconds. Default: 3000 milliseconds Range: 1000-65535
Web: n/a UCI: iecd.<port>.dnp3_app_frag_rx_time Opt: dnp3_app_frag_rx_time	Defines DNP3 application level fragment response timeout. Default: 1000 milliseconds Range: 1000-65535
Web: n/a UCI: iecd.<port>.dnp3_app_txq_size Opt:n/a	Defines DNP3 application level transmit queue size (number of frames). Default: 64 Range: 2-256
Web: n/a UCI: iecd.<port>.dnp3_app_output_mode Opt: dnp3_app_output_mode	Defines a decimal code that controls how the router sends a DNP3 binary output command to a DNP3 RTU. The most commonly used model is **Select/Operate**. **Note**: this command is used where the router is acting as a DNP3 master. Default: 0 — Use WRITE command 1 — Use Slect/Operate message sequence 2 — Use Direct Operate message
Web: n/a UCI: iecd.<port>.dnp3_app_evpoll_mode Opt: dnp3_app_evpoll_mode	Defines DNP3 outstation event polling interval in milliseconds. Default: 3000 milliseconds Range: 1000-65535
Web: n/a UCI: iecd.<port>.dnp3_fsm_debug_on Opt: dnp3_fsm_debug_on	Enables DNP3 link and application level state machine transition and event logging into syslog. Default: 1 — Enabled 0 — Disabled
Web: n/a UCI: iecd.<port>.dnp3_object_parser_debug_on Opt: dnp3_object_parser_debug_on	Enables or disables logging low level debug information when parsing DNP3 objects in the received DNP3 slave messages Default: 1 — Enabled 2 — Disabled
Web: n/a UCI: iecd.<port> .dnp3_dump_data Opt: dnp3_dump_data	Enables RX/TX Hex dump. Default: 1 — Enabled 2 — Disabled
Web: n/a UCI: iecd.<port>.dnp3_trace_on	Enables DNP3 protocol tracing. Default: 1 — Enabled

Web Field/UCI/Package Option	Description
Opt: dnp3_trace_on	2 Disabled

43.1.7. Port Settings: Modbus

The IEC 104 to Modbus conversion feature on the router allows converting commands in the control direction and the responses and process data in the monitor direction between the SCADA Master running the IEC 104 protocol and the remote RTUs running Modbus protocol.

The router software supports two variations of the Modbus protocol:

- Modbus over serial: the Modbus devices are connected to the serial interface of the router.
- Modbus TCP: the Modbus devices are located on the IP network reachable from the router.

In the Modbus over serial variation, currently the router supports Modbus 'RTU mode' frame format of the Modbus specification only.

The IEC 104 gateway port modbus configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Modbus Protocol

UCI: iecd.<port>.modbus_protocol

Opt: modbus_protocol</td><td>Defines the protocol variation used by RTU that connects to this router.

<table><tr><th>Option</th><th>Description</th><th>UCI</th></tr><tr><td>Modbus serial</td><td>Modbus over serial</td><td>modbus_serial</td></tr><tr><td>Modbus TCP</td><td>Modbus over TCP</td><td>modbus_tcp</td></tr></table></td></tr>
<tr><td>Web: Modbus local IP

UCI: iecd.<port>.modbus_local_ip

Opt: modbus_local_ip</td><td>Defines the local IP to use in Modbus mode.

<table><tr><td>Default</td><td>0.0.0.0</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus local port

UCI: iecd.<port>.modbus_local_port

Opt: modbus_local_port</td><td>Defines the local port to use in Modbus mode.

<table><tr><td>Default</td><td>888</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus remote IP

UCI: iecd.<port>.modbus_remote_ip

Opt: modbus_remote_ip</td><td>Defines the remote IP address.

<table><tr><td>Default</td><td>127.0.0.1</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus remote port

UCI: iecd.<port>.modbus_remote_port

Opt: modbus_remote_port</td><td>Defines the remote port.

<table><tr><td>Default</td><td>999</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus polling time

UCI: iecd.<port>.modbus_polling_time

Opt: modbus_polling_time</td><td>Defines the slave polling interval in milliseconds.

<table><tr><td>Default</td><td>3000</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: Modbus frame response time

UCI: iecd.<port>.modbus_resp_time

Opt: modbus_resp_time</td><td>Defines in milliseconds the maximum time allowed to receive a response frame from a Modbus slave.

<table><tr><td>Default</td><td>1000</td></tr><tr><td>Range</td><td></td></tr></table></td></tr>
<tr><td>Web: n/a

UCI: iecd.<port>.modbus_dump_data

Opt: modbus_dump_data</td><td>Enables RX/TX Hex dump.</td></tr>
<tr><td>Web: n/a

UCI: iecd.<port>.modbus_trace_on

Opt: modbus_trace_on</td><td>Enables Modbus protocol tracing.</td></tr>
<tr><td>Web: n/a

UCI: iecd.<port>.modbus_fsm_debug_on

Opt: modbus_fsm_debug_on</td><td>Enables Modbus state machine debugging.</td></tr>
</table>

43.1.8. Port Settings: Advanced

In this section you can configure the advanced port settings.

The IEC 104 gateway port advanced configuration page

Web Field/UCI/Package Option	Description
Web: Syslog severity UCI: iecd.<port>.loglevel Opt: loglevel	Defines the lowest severity used for logging events by iecd. 0 Emergency 1 Alert 2 Critical 3 Error 4 Warning 5 Notice 6 Informational 7 Debug
Web: Enable TCP keepalives UCI: iecd.<port>.tcp_keepalive_enabled Opt: tcp_keepalive_enabled	Defines whether to enable TCP keepalive. Default: 1 Enabled 2 Disabled
Web: TCP Keepalive interval UCI: iecd.<port>.tcp_keepalive_interva l Opt: tcp_keepalive_interval	Defines the TCP keepalive interval in seconds. Default 5 seconds Range
Web: TCP Keepalive timeout UCI: iecd.<port>.tcp_keepalive_timeout Opt: tcp_keepalive_timeout	Defines the TCP keepalive timeout in seconds. Default 5 seconds Range
Web: TCP Keepalive count UCI: iecd.<port>.tcp_keepalive_count Opt: tcp_keepalive_count	Defines the number of unanswered keepalives before terminating the TCP session. Default 3 seconds Range
Web: n/a UCI: iecd.<port>.tcp_user_timeout Opt: tcp_user_timeout	Defines the maximum time in milliseconds to wait for a TCP ACK after data transmission before closing the connection in TCP established state. Set to **0** to use kernel defaults (about 15-20 minutes). Default 2000 milliseconds Range

43.2. IEC 101 Links

The following section defines the IEC 101 slave links used in IEC 101 conversion. Each link is defined by a config iec101link section block. There is a maximum of 32 links supported.

In IEC 101 unbalanced mode all of these links can be used. However, as IEC 101 balanced mode is used in a point to point scenario, it is assumed there will be only one outstation per serial port. Only the first link configured for that port will be used. Each peer, either the controlling station (Master) or the controlled station (RTU) can initiate communication in balanced mode.

IEC101 slave links configuration page

Web Field/UCI/Package Option	Description
Web: Port Number UCI: iecd.iec101link[x].portno Opt: portno	Defines the serial port number to which this point belongs.
Web: IEC101 Link Address UCI: iecd.iec101link[x].address Opt: address	Defines the IEC101 station link address.
Web: IEC101 Link ASDU Station UCI: iecd. iec101link[x].asduaddr Opt: asduaddr	Defines the IEC101 station common ASDU address.

43.3. Point Mappings

IEC 104 point mappings are used for DNP3 and Modbus conversion only.

The point mappings comprise the information necessary to perform conversion between each data variable (point) on the remote RTU and the corresponding variable in the IEC 104 domain.

Modbus TCP requires a device route file (**/root/iecd/devroute.csv**) to map the point configuration to an IP address. For more information, read the Modbus route file section below.

There is a maximum of 1200 point mappings supported per serial port.

The IEC 104 gateway point mapping configuration page

<table>
<tr><th>Web Field/UCI/Package Option</th><th>Description</th></tr>
<tr><td>Web: Port Number

UCI: iecd.point[x].portno

Opt: portno</td><td>Defines the port number to which this point belongs (1 to 4). This corresponds to the serial port number.</td></tr>
<tr><td>Web: IEC101 Type ID

UCI: iecd.point[x].iec101_type_id

Opt: iec101_type_id</td><td>Defines the IEC101 type ID (data type). All types are defined in IEC-60870-5-104

<table>
<tr><td>1</td><td>Single point information</td></tr>
<tr><td>2</td><td>Double point information</td></tr>
<tr><td>7</td><td>Bitstring 32 bits</td></tr>
<tr><td>9</td><td>Measured normalised value short signed</td></tr>
<tr><td>11</td><td>Measured scaled value short signed</td></tr>
<tr><td>13</td><td>IEEE STD 754 - Short floating point number</td></tr>
<tr><td>14</td><td>IEEE STD 754 - Short floating point number with time tag CP24Time2a</td></tr>
<tr><td>15</td><td>Integrated totals, 32 bit signed integer</td></tr>
<tr><td>20</td><td>Packed single point information with status change detection</td></tr>
<tr><td>21</td><td>Measured normalised value short signed without quality descriptor</td></tr>
<tr><td>30</td><td>Single point information with time tag CP56Time2a</td></tr>
<tr><td>31</td><td>Double point information with time tag CP56Time2a</td></tr>
<tr><td>33</td><td>Bitstring of 32 bits with time tag CP56Time2a</td></tr>
<tr><td>34</td><td>Measure normalised value short signed time tag CP56Time2a</td></tr>
<tr><td>35</td><td>Measured value, scaled value with time tag CP56Time2a</td></tr>
<tr><td>36</td><td>Measured value, short floating point number with time tag CP56Time 2a</td></tr>
<tr><td>37</td><td>Integrated totals with time tag CP56Time2a</td></tr>
</table></td></tr>
<tr><td>Web: IEC104 Type ID

UCI: iecd.point[x].iec104_type_id

Opt: iec104_type_id</td><td>Defines the IEC104 type ID (data type). All types are defined in IEC-60870-5-104

<table>
<tr><td>1</td><td>Single point information</td></tr>
<tr><td>2</td><td>Double point information</td></tr>
<tr><td>7</td><td>Bitstring 32 bits</td></tr>
<tr><td>9</td><td>Measured normalised value short signed</td></tr>
<tr><td>11</td><td>Measured scaled value short signed</td></tr>
<tr><td>13</td><td>IEEE STD 754 - Short floating point number</td></tr>
<tr><td>14</td><td>IEEE STD 754 - Short floating point number with time tag CP24Time2a</td></tr>
<tr><td>15</td><td>Integrated totals, 32 bit signed integer</td></tr>
<tr><td>20</td><td>Packed single point information with status change detection</td></tr>
<tr><td>21</td><td>Measured normalised value short signed without quality descriptor</td></tr>
<tr><td>30</td><td>Single point information with time tag CP56Time2a</td></tr>
<tr><td>31</td><td>Double point information with time tag CP56Time2a</td></tr>
<tr><td>33</td><td>Bitstring of 32 bits with time tag CP56Time2a</td></tr>
<tr><td>34</td><td>Measure normalised value short signed time tag CP56Time2a</td></tr>
<tr><td>35</td><td>Measured value, scaled value with time tag CP56Time2a</td></tr>
<tr><td>36</td><td>Measured value, short floating point number with time tag CP56Time 2a</td></tr>
<tr><td>37</td><td>Integrated totals with time tag CP56Time2a</td></tr>
</table></td></tr>
<tr><td>Web: IEC101 IOA

UCI: iecd.point[x].iec101_ioa

Opt: iec101_ioa</td><td>Defines IEC104 information object address. This is how remote an IEC104 SCADA master knows one point from another.

<table>
<tr><td>Range</td><td>1-116777215</td></tr>
</table></td></tr>
<tr><td>Web: IEC104 IOA

UCI: iecd.point[x].iec104_ioa

Opt: iec104_ioa</td><td>Defines IEC104 information object address. This is how a remote IEC104 SCADA master knows one point from another.

<table>
<tr><td>Range</td><td>1-116777215</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Web: Device Addr

UCI: iecd.point[x].devaddr

Opt: devaddr</td><td>Defines the Modbus device address of the RTU (Modbus slave address). Used for identifying the point mapping to IP address in the device route file for Modbus TCP.

This is not used in DNP3 mode.</td></tr>
<tr><td>Web: DNP3 Options

UCI: iecd.point[x].dnp3options

Opt: dnp3options</td><td>For DNP3. Defines the DNP3 options bitmap.

<table><tr><td>Default</td><td>1</td></tr><tr><td>Range</td><td>1-116777215</td></tr></table></td></tr>
<tr><td>Web: Group

UCI: iecd.point[x].group

Opt: group</td><td>For DNP3. Defines the DNP3 group number to which this data point maps to.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-255</td></tr></table>
For Modbus. Defines the Modbus data type.

<table><tr><td>Default: 0</td><td>Discreet input</td></tr><tr><td>1</td><td>Input register</td></tr><tr><td>2</td><td>Holding register</td></tr><tr><td>3</td><td>Coil</td></tr></table></td></tr>
<tr><td>Web: Index

UCI: iecd.point[x].index

Opt: index</td><td>For DNP3. Defines the DNP3 point index.

For Modbus. Defines the Modbus data index (point number).

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: Index2

UCI: iecd.point[x].index2

Opt: index2</td><td>For DNP3. Defines the DNP3 secondary point index.

For Modbus. Defines the Modbus data index (point number).

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: Modbus options

UCI: iecd.point[x].mb_options

Opt: mb_options</td><td>For Modbus. Defines the Modbus options bitmap.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: Modbus bitmap mask

UCI: iecd.point[x].bitmap_mask

Opt: bitmap_mask</td><td>For Modbus. Defines the Modbus bitmap mask.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: Modbus CtrlMode index

UCI: iecd.point[x].ctrlmode_index

Opt: ctrlmode_index</td><td>For Modbus. Defines the Modbus control mode register index.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: Modbus CtrlMode value

UCI: iecd.point[x].ctrlmode_val

Opt: ctrlmode_val</td><td>For Modbus. Defines the Modbus control mode register value.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: Local Digital Output GPIO number

UCI: iecd.point[x].local_gpio_output_nr

Opt: local_gpio_output_nr</td><td>Defines the local digital output GPIO number.

<table><tr><td>Default</td><td>0</td></tr><tr><td>Range</td><td>0-65535</td></tr></table></td></tr>
<tr><td>Web: IEC61850 DO

UCI: iecd.point[x].iec61850_do</td><td>Defines the IEC61850 Data Object reference. (Maximum 32 characters).

<table><tr><td>Default</td><td>0</td></tr></table></td></tr>
</table>

Opt: iec61850_do	Range: 0-32 characters
Web: n/a UCI: iecd.point[x].dword Opt: dword	Defines the DWORD type. Relevant for Modbus data types IR (input registers) and HR (holding registers). Default: 0 — Data point is treated as 16 bit wide 1 — Data point is treated as 32 bit wide. Two consecutive 16 bit registers are read from the Modbus device.

MODBUS device route file

If the configured MODBUS protocol variation is Modbus TCP, then the device route file at

/root/iecd/devroute.csv is used to map the device address `(iecd.point[x].devaddr)` from the point mapping to the remote IP address of the Modbus TCP slave device.

The devroute.csv file entries will have the following format:

```
<Modbus device addr>, <IP address>
```

For example, for the point mapping file, enter:

```
config point
option portno 1
option iec104_type_id 30
option iec104_ioa 64213
option devaddr 1
option group 0
option index 2
```

For the devroute.csv entry, enter:

```
1,192.168.0.106
```

43.4. IEC 104 Gateway Configuration Using Command Line

The IEC104 gateway uses the iecd package /etc/config/iecd. You can configure multiple port, iec101link and points sections.

By default, IEC104 gateway port instances are named port. It is identified by @port followed by the port position in the package as a number. For example, for the first port in the package using UCI:

```
iecd.@port[0]=port
iecd.@port[0].enable=1
```

Or using package options:

```
config port
option enable '1'
```

By default, all IEC104 gateway IEC101 link instances are named iec101link, the instance is identified by @iec101link followed by the link position in the package as a number.

For example, for the first IEC101 link in the package using UCI:

```
iecd.@iec101link[0]=iec101link
iecd.@iec101link[0].portno=1
```

Or using package options:

```
config iec101link
option portno '1'
```

By default, all IEC104 gateway point instances are named point, it is identified by @point followed by the point position in the package as a number. For example, for the first point in the package using UCI:

```
iecd.@point[0]=point
iecd.@point[0].portno=1
```

Or using package options:

```
config point
option portno '1'
```

43.5. IEC 104 To IEC 101 Conversion (Balanced Or Unbalanced)

The following example shows IEC 104 to IEC 101 unbalanced conversion with one IEC 101 link.

To configure IEC 104 to IEC 101 balanced conversion set option iec101_mode to

IEC 104 to IEC 101 using UCI

```
root@VA_router:~# uci show iecd
iecd.main=iecd
iecd.main.enable=1
iecd.port1=port
iecd.port1.enable=1
iecd.port1.loglevel=5
iecd.port1.tcp_keepalive_enabled=1
iecd.port1.tcp_keepalive_interval=5
iecd.port1.tcp_keepalive_timeout=5
iecd.port1.tcp_keepalive_count=3
iecd.port1.tcp_user_timeout=20000
iecd.port1.master_protocol=iec101
iecd.port1.slave_protocol=iec104
iecd.port1.ioa_offset=0
iecd.port1.pointmap_file=/root/iecd/iecd_points1.csv
iecd.port1.iec104_local_ip=0.0.0.0
iecd.port1.iec104_local_tcpport=2404
iecd.port1.iec104_k=12
iecd.port1.iec104_w=9
iecd.port1.iec104_t2=10000
iecd.port1.iec104_gi_resp_time=200
iecd.port1.iec104_txq_size=128
iecd.port1.iec104_sync_time=1
iecd.port1.iec104_time_tagged_cmds=0
iecd.port1.iec104_cmd_delay_time=5000
iecd.port1.iec104_fsm_debug_on=0
iecd.port1.iec104_dump_data=0
iecd.port1.iec104_trace_on=0

#IEC101 conversion options
iecd.port1.iec101_target_ip=127.0.0.1
iecd.port1.iec101_target_tcpport=999
iecd.port1.iec101_mode=unbalanced #balanced or unbalanced
iecd.port1.iec101_cot_tx_length=1
iecd.port1.iec101_cot_source_octet=0
iecd.port1.iec101_asdu_addrlen=1
iecd.port1.iec101_info_obj_addrlen=2
```

```
iecd.port1.iec101_data_polling_time=500

iecd.port1.iec101_ack_delay=0

iecd.port1.iec101_link_addrlen=1

iecd.port1.iec101_frame_rsp_time=2000

iecd.port1.iec101_max_tx_retry=3

iecd.port1.iec101_txq_size=128

iecd.port1.iec101_send_spont_delay_acq=1

iecd.port1.iec101_fsm_debug_on=0

iecd.port1.iec101_dump_data=0

iecd.port1.iec101_trace_on=0
```

The following section defines IEC101 slave links used in IEC101 unbalanced mode on each link is defined by a config block 'config iec101link'

To add more links repeat the section block for each added link. # Maximum 32 links are supported

```
iecd.@iec101link[0]=iec101link

iecd.@iec101link[0].portno=1

iecd.@iec101link[0].address=6

iecd.@iec101link[0].asduaddr=6

#No data point mappings for IEC104 to IEC101 conversion
```

IEC 104 to IEC 101 using Package Options

```
root@VA_router:~# uci export iecd
package iecd
config iecd 'main'
option enable '1'
config port 'port1'
option enable '1'
option loglevel '5'
option tcp_keepalive_enabled '1'
option tcp_keepalive_interval '5'
option tcp_keepalive_timeout '5'
option tcp_keepalive_count '3'
option tcp_user_timeout '20000'
option master_protocol 'iec101'
option slave_protocol 'iec104'
option ioa_offset '0'
option pointmap_file '/root/iecd/iecd_points1.csv'
option iec104_local_ip '0.0.0.0'
option iec104_local_tcpport '2404'
option iec104_k '12'
option iec104_w '9'
option iec104_t2 '10000'
option iec104_gi_resp_time '200'
option iec104_txq_size '128'
option iec104_sync_time '1'
option iec104_time_tagged_cmds '0'
option iec104_cmd_delay_time '5000'
option iec104_fsm_debug_on '0'
option iec104_dump_data '0'
option iec104_trace_on '0'
#IEC101 conversion options
option iec101_target_ip '127.0.0.1'
option iec101_target_tcpport '999'
option iec101_mode 'unbalanced' #balanced or unbalanced
```

```
option iec101_cot_tx_length '1'

option iec101_cot_source_octet '0'

option iec101_asdu_addrlen '1'

option iec101_info_obj_addrlen '2'

option iec101_data_polling_time '500'

option iec101_ack_delay '0'

option iec101_link_addrlen '1'

option iec101_frame_rsp_time '2000'

option iec101_max_tx_retry '3'

option iec101_txq_size '128'

option iec101_send_spont_delay_acq '1'

option iec101_fsm_debug_on '0'

option iec101_dump_data '0'

option iec101_trace_on '0'

# The following section defines IEC101 slave links used in IEC101 unbalanced mode on

# Each link is defined by a config block 'config iec101link'

# To add more links repeat the section block for each added link. To remove links, simply remove the link block from the configuration

# Maximum 32 links are supported

# Definition of options within the section block:

# portno - port number to which this point belongs (1 to 4) # address - IEC101 slave link address

# asduaddr IEC101 slave common ASDU address

config iec101link

option portno 1

option address 6

option asduaddr 6

#No data point mappings for IEC104 to IEC101 conversion
```

43.6. IEC104 To Modbus Conversion

The following example shows IEC104 to Modbus over serial.

To configure Modbus TCP, set `option modbus_protocol` to **`modbus_tcp`**.

When configuring Modbus TCP, the device route file at /root/iecd/devroute.csv must be configured to map the device address `option devaddr` from the point mapping to the remote IP address of the Modbus TCP slave device.

The devroute.csv file entries will have the following format:

```
<Modbus device addr>, <IP address>
```

For example, for the point mapping file:

```
config point
option portno 1
option iec104_type_id 30
option iec104_ioa 64213
option devaddr 1
option group 0
option index 2
```

For the devroute.csv entry:

```
1,192.168.0.106
```

43.6.1. IEC104 To Modbus Using UCI

```
root@VA_router:~# uci show iecd
iecd.main=iecd
iecd.main.enable=1
iecd.port1=port
iecd.port1.enable=1
iecd.port1.loglevel=5
iecd.port1.tcp_keepalive_enabled=1
iecd.port1.tcp_keepalive_interval=5
iecd.port1.tcp_keepalive_timeout=5
iecd.port1.tcp_keepalive_count=3
iecd.port1.tcp_user_timeout=20000
iecd.port1.master_protocol=modbus
iecd.port1.slave_protocol=iec104
iecd.port1.ioa_offset=0
iecd.port1.pointmap_file=/root/iecd/iecd_points1.csv
iecd.port1.iec104_local_ip=0.0.0.0
iecd.port1.iec104_local_tcpport=2404
iecd.port1.iec104_k=12
iecd.port1.iec104_w=9
iecd.port1.iec104_t2=10000
iecd.port1.iec104_gi_resp_time=200
iecd.port1.iec104_txq_size=128
iecd.port1.iec104_sync_time=1
iecd.port1.iec104_time_tagged_cmds=0
iecd.port1.iec104_cmd_delay_time=5000
iecd.port1.iec104_fsm_debug_on=0
iecd.port1.iec104_dump_data=0
iecd.port1.iec104_trace_on=0
iecd.port1.iec101_cot_source_octet=0
```

Modbus Conversion Options

```
iecd.port1.modbus_protocol=modbus_serial
iecd.port1.modbus_local_ip=0.0.0.0
iecd.port1.modbus_local_port=888
iecd.port1.modbus_remote_ip=127.0.0.1
iecd.port1.modbus_remote_port=999
iecd.port1.modbus_polling_time=3000
iecd.port1.modbus_resp_time=1000
iecd.port1.modbus_dump_data=0
iecd.port1.modbus_trace_on=0
iecd.port1.modbus_fsm_debug_on=0
```

Modbus Data Point Mappings

```
iecd.@point[0]=point
iecd.@point[0].portno=1
iecd.@point[0].iec104_type_id=36
iecd.@point[0].iec104_ioa=6620161
iecd.@point[0].iec101_ioa=0
iecd.@point[0].devaddr=11
iecd.@point[0].group=1
iecd.@point[0].index=18459
iecd.@point[0].dword=1
iecd.@point[1]=point
iecd.@point[1].portno=1
iecd.@point[1].iec104_type_id=36
iecd.@point[1].iec104_ioa=6620162
iecd.@point[1].iec101_ioa=0
iecd.@point[1].devaddr=11
iecd.@point[1].group=1
iecd.@point[1].index=18461
iecd.@point[1].dword=1
```

IEC 104 to Modbus using Package Options

```
root@VA_router:~# uci export iecd
package iecd
config iecd 'main'
option enable '1'

config port 'port1'
option enable '1'
option loglevel '5'
option tcp_keepalive_enabled '1'
option tcp_keepalive_interval '5'
option tcp_keepalive_timeout '5'
option tcp_keepalive_count '3'
option tcp_user_timeout '20000' option master_protocol 'modbus' option slave_protocol 'iec104' option ioa_offset '0'
option pointmap_file '/root/iecd/iecd_points1.csv' option iec104_local_ip '0.0.0.0'
option iec104_local_tcpport '2404'
option iec104_k '12'
option iec104_w '9'
option iec104_t2 '10000'
option iec104_gi_resp_time '200'
option iec104_txq_size '128'
option iec104_sync_time '1'
option iec104_time_tagged_cmds '0'
option iec104_cmd_delay_time '5000'
option iec104_fsm_debug_on '0'
option iec104_dump_data '0'
option iec104_trace_on '0'
option iec101_cot_source_octet '0'
```

Modbus Conversion Options

```
option modbus_protocol 'modbus_serial'
option modbus_local_ip '0.0.0.0'
option modbus_local_port '888'
option modbus_remote_ip '127.0.0.1'
option modbus_remote_port '999'
option modbus_polling_time '3000'
option modbus_resp_time '1000'
option modbus_dump_data '0'
option modbus_trace_on '0'
option modbus_fsm_debug_on '0'

config point
option portno '1'
option iec104_type_id '36'
option iec104_ioa '6620161'
option iec101_ioa '0'
option devaddr '11'
option group '1'
option index '18459'
option dword '1'

config point
option portno '1'
option iec104_type_id '36'
option iec104_ioa '6620162'
option iec101_ioa '0'
option devaddr '11'
option group '1'
option index '18461'
option dword '1'
```

43.7. Configuring The Terminal Server

The terminal server is used to control the data from the serial port over the IP network.

The terminal server configuration can be found at **Services -> Terminal Server**. The Terminal Server Configuration page appears. You must configure two main sections: Main Settings and Port Settings.

The terminal server for IEC104 to each of the RTU protocol conversions differ only slightly. This section shows the command line options for configuring the terminal server for IEC104 conversion.

For more detailed information on configuring the terminal server using the web GUI and

option values, read the chapter, ‘Configuring terminal server’.

Configuring the Terminal Server for IEC 104 to IEC 101 using UCI

```
root@VA_router:~# uci show tservd
tservd.main=tservd
tservd.main.enable=1
tservd.main.debug_ev_enable=0
tservd.main.log_severity=5
tservd.main.debug_rx_tx_enable=0
tservd.port1=port
tservd.port1.enable=1
tservd.port1.devName=/dev/ttySC0
tservd.port1.ip_port1=0
tservd.port1.ip_port2=0
tservd.port1.remote_ip1=0.0.0.0
tservd.port1.remote_ip2=0.0.0.0
tservd.port1.tcp_always_on=1
tservd.port1.close_tcp_on_dsr=0
tservd.port1.tty_always_open=1
tservd.port1.fwd_timeout=0
tservd.port1.fwd_timer_mode=idle
tservd.port1.fwd_buffer_size=1
tservd.port1.sfwd_buffer_size=0
tservd.port1.sfwd_timeout=0
tservd.port1.sfwd_timer_mode=idle
tservd.port1.speed=9600
tservd.port1.wsize=8
tservd.port1.parity=1
tservd.port1.stops=1
tservd.port1.fc_mode=0
tservd.port1.disc_time_ms=5000
tservd.port1.server_mode=1
tservd.port1.proxy_mode=0
tservd.port1.local_ip=0.0.0.0
tservd.port1.listen_port=999
tservd.port1.udpMode=0
tservd.port1.udpLocalPort=0
tservd.port1.udpRemotePort=0
tservd.port1.udpKaIntervalMs=0
```

```
tservd.port1.udpKaCount=3

tservd.port1.serial_mode_gpio_control=1

tservd.port1.tcp_nodelay=1

tservd.port1.portmode=rs232
```

Configuring IEC 104 to IEC 101 using Package Options

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

package tservd


config tservd main

# set to 1 to enable terminal server

option enable 1

# enables detailed debug logging (state transitions, data transfer etc)

option debug_ev_enable 0

# sets syslog level (0 to 7), default is 6

option log_severity 5
```

```
option debug_rx_tx_enable 0

config port 'port1'

# enables this port option enable 1

# serial device name

option devName '/dev/ttySC0'

# destination peer port IP number (two number for failover) option ip_port1 0

option ip_port2 0

# destination peer ip address (two addresses for failover) option remote_ip1 '0.0.0.0'

option remote_ip2 '0.0.0.0'

# keep TCP session always connected option tcp_always_on 1

# close TCP session on detection of DSR signal low option close_tcp_on_dsr 0

# keep serial port always open (if option not present, default is 0) option tty_always_open 1

# Forwarding timeout in milliseconds (serial to network)

option fwd_timeout 0

# Forwarding timer mode (serial to network), 'idle'=timer re-started on each received data,

# 'aging'=timer started on first rx

option fwd_timer_mode 'idle'

# Forwarding buffer size (serial to network)

option fwd_buffer_size 1

# Forwarding buffer size (network to serial), 0=use maximum possible network rx buffer size

option sfwd_buffer_size 0

# Forwarding timeout in milliseconds (network to serial), 0=forward to serial immediately

option sfwd_timeout 0

# Forwarding timer mode (network to serial), 'idle'=timer re-started on each received data,

# 'aging'=timer started on first rx

option sfwd_timer_mode 'idle'

# serial device speed in baud

option speed 9600

# serial device word size (5,6,7,8)

option wsize 8

# serial device parity (0=none, 1=even, 2=odd)

option parity 1

# serial device number of stop bits (1 or 2)

option stops 1

# serial flow control mode (0=none, 1=RTS CTS, 2=XONXOFF)

option fc_mode 0
```

```
# time in milliseconds to start re-connecting after setting DTR low
option disc_time_ms 5000
# TCP server mode
option server_mode 1
# Proxy mode (off by default)
option proxy_mode 0
# Local IP address to listen on (0.0.0.0=listen on any interface)
option local_ip '0.0.0.0'
# TCP listen port for server mode
option listen_port 999
# UDP mode
option udpMode 0
# UDP local port UDP mode
option udpLocalPort 0
# UDP port for UDP mode
option udpRemotePort 0
# If set to non zero, send empty UDP packets every this many milliseconds to remote peer
option udpKaIntervalMs 0
# Max number of consecutive remote UDP keepalive missed (not received) before UDP
# session considered broken
option udpKaCount 3
option serial_mode_gpio_control 1
option tcp_nodelay 1
# rs232 - RS-232 mode, rs485hdx - rs485 2 wire half duplex mode in which transmitter drives
# RTS. rs485fdx - RS485 4 wire full duplex mode. 'v23' - using V.23 leased line card driver.
# x21 - use USB serial card in sync mode
option portmode 'rs232'
```

43.7.1. Configuring The Terminal Server For IEC104 To DNP3

The terminal server configuration for IEC104 to DNP3 is the same as for IEC104 to IEC101 except for serial device parity which is set to **none**.

Parity setting using uci:

```
tservd.port1.parity=1
```

Parity setting using package options:

```
option parity 0
```

43.7.2. Configuring The Terminal Server For IEC 104 To Modbus Over Serial

The terminal server is only used for IEC104 to Modbus over serial. It is not used for Modbus over TCP.

The options necessary for IEC104 to Modbus configuration are listed below. These options are for the first serial port only.

IEC104 to Modbus over Serial using UCI

```
root@VA_router:~# uci show tservd
tservd.main=tservd
tservd.main.enable=1
tservd.main.debug_ev_
enable=0 tservd.main.log_severity=5
tservd.main.debug_rx_tx_enable=0
tservd.port1=port tservd.port1.enable=1
tservd.port1.devName=/dev/ttySC0
tservd.port1.ip_port1=999
tservd.port1.ip_port2=999
tservd.port1.remote_ip1=127.0.0.1
tservd.port1.remote_ip2=127.0.0.1
tservd.port1.tcp_always_on=1
tservd.port1.close_tcp_on_dsr=0
tservd.port1.tty_always_open=1
tservd.port1.fwd_timeout=10
tservd.port1.fwd_timer_mode=idle
tservd.port1.fwd_buffer_size=300
tservd.port1.sfwd_buffer_size=0
tservd.port1.sfwd_timeout=0
tservd.port1.sfwd_timer_mode=idle
tservd.port1.speed=19200
tservd.port1.wsize=8
tservd.port1.parity=1
tservd.port1.stops=1
tservd.port1.fc_mode=0
tservd.port1.disc_time_ms=5000
tservd.port1.server_mode=1
tservd.port1.proxy_mode=0
tservd.port1.local_ip=0.0.0.0
tservd.port1.listen_port=999
tservd.port1.udpMode=1
tservd.port1.udpLocalPort=999
tservd.port1.udpRemotePort=888
tservd.port1.udpKaIntervalMs=0
tservd.port1.udpKaCount=3
```

```
tservd.port1.serial_mode_gpio_control=1

tservd.port1.portmode=rs232
```

IEC104 to Modbus over serial using package options

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

config tservd main
# set to 1 to enable terminal server
option enable 1
# enables detailed debug logging (state transitions, data transfer etc)
option debug_ev_enable 0
# sets syslog level (0 to 7), default is 6
option log_severity 5
option debug_rx_tx_enable 0
config port 'port1'
# enables this port
option enable 1

# serial device name
option devName '/dev/ttySC0'
destination peer port IP number (two number for failover)
option ip_port1 999
option ip_port2 999
# destination peer ip address (two addresses for failover)
option remote_ip1 '127.0.0.1'
option remote_ip2 '127.0.0.1'
# keep TCP session always connected
option tcp_always_on 1
# close TCP session on detection of DSR signal low
option close_tcp_on_dsr 0
# keep serial port always open (if option not present, default is 0)
option tty_always_open 1
# Forwarding timeout in milliseconds (serial to network)
option fwd_timeout 10
# Forwarding timer mode (serial to network), 'idle'=timer re-started on each received data,
# 'aging'=timer started on first rx
option fwd_timer_mode 'idle'
# Forwarding buffer size (serial to network)
option fwd_buffer_size 300
# Forwarding buffer size (network to serial), 0=use maximum possible network rx buffer size
```

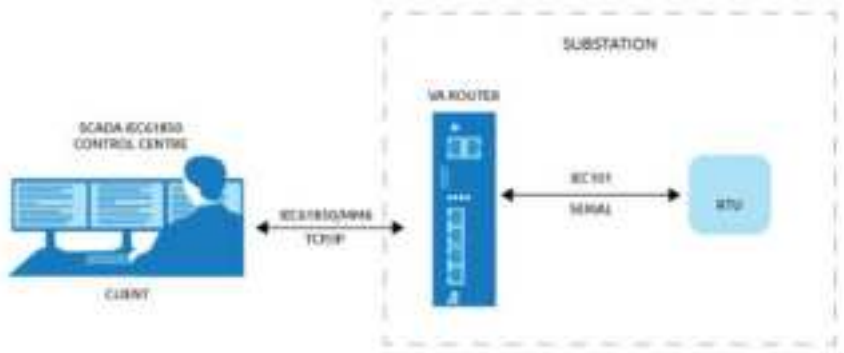
```
option sfwd_buffer_size 0
# Forwarding timeout in milliseconds (network to serial), 0=forward to serial immediately
option sfwd_timeout 0
# Forwarding timer mode (network to serial), 'idle'=timer re-started on each received data,
# 'aging'=timer started on first rx
option sfwd_timer_mode 'idle'
# serial device speed in baud
option speed 19200
# serial device word size (5,6,7,8)
option wsize 8
# serial device parity (0=none, 1=even, 2=odd)
option parity 1
# serial device number of stop bits (1 or 2)
option stops 1
# serial flow control mode (0=none, 1=RTS CTS, 2=XONXOFF)
option fc_mode 0
# time in milliseconds to start re-connecting after setting DTR low
option disc_time_ms 5000
# TCP server mode
option server_mode 1
# Proxy mode (off by default)
option proxy_mode 0
# Local IP address to listen on (0.0.0.0=listen on any interface)
option local_ip '0.0.0.0'
# TCP listen port for server mode
option listen_port 999
# UDP mode
option udpMode 1
# UDP local port UDP mode
option udpLocalPort 999
# UDP port for UDP mode
option udpRemotePort 888
# If set to non zero, send empty UDP packets every this many milliseconds to remote peer
option udpKaIntervalMs 0
# Max number of consecutive remote UDP keepalive missed (not received) before UDP
# session considered broken
```

```
option udpKaCount 3

option serial_mode_gpio_control 1

# rs232 - RS-232 mode, rs485hdx - rs485 2 wire half duplex mode in which transmitter drives

# RTS. rs485fdx - RS485 4 wire full duplex mode. 'v23' - using V.23 leased line card driver.

# x21 - use USB serial card in sync mode

option portmode 'rs232'
```

43.8. Configuring IEC61850 To IEC101 Conversion

The IEC61850 to IEC101-unbalanced conversion feature of the router allows converting commands in the control direction and the responses and process data in the monitor direction between the SCADA master running the IEC61850 protocol and the remote RTUs running IEC101 protocol in unbalanced mode over serial interface.

In IEC101 unbalanced mode, the router supports communication of up to 32 IEC101 slaves connected onto the same serial interface.



Example of IEC61850 to IEC 101 conversion scenario

The IEC104 gateway and terminal server are used for IEC61850 to IEC101 conversion, as in the other protocol conversions however the IEC62850 options are currently not available via the web interface. The following section details command line configuration.

Web Field/UCI/Package Option	Description
iecd port Config Section	
Web: n/a UCI: iecd.<port>.slave_protocol Opt: slave_protocol	Defines what protocol the SCADA control centre is using to connect to this gateway. Default: iec104 | IC104 iec61850 | IEC61850
Web: n/a UCI: iecd.<port>.iec61850_local_ip Opt: iec61850_local_ip	Defines the local IP address this IEC61850 peer binds to.
Web: n/a UCI: iecd.<port>.iec61850_local_tcpport Opt: iec61850_local_tcpport	Defines the local TCP port this IEC104 peer listens on. Default | 2404 Range | 1-65535
iecd point Config Section	
Web: n/a UCI: iecd.point[x].iec61850_ld Opt: iec61850_ld	Defines the IEC61850 logical device name. For example: `option iec61850_ld 'SENSORS'` Range | 0-32 characters
Web: n/a UCI: iecd.point[x].iec61850_ln Opt: iec61850_ln	Defines the IEC61850 logical node name. For example: `option iec61850_ln 'LLN0'` Range | 0-32 characters
Web: n/a UCI: iecd.point[x].iec61850_do Opt: iec61850_do	Defines the IEC61850 data object name. For example: `option iec61850_do 'SPS01'` Range | 0-32 characters
Web: n/a UCI: iecd.point[x] iec101_type_id Opt: iec101_type_id	Defines the IEC104 type ID (data type). For example: 1 – Single Point Information 2 – Double Point Information All types are defined in IEC-60870-5-101.
Web: IEC101 IOA UCI: iecd.point[x].iec101_ioa Opt: iec101_ioa	Defines IEC101 information object address. Default: 1 | Single point Information Range | 1-16777215

Relation of IEC 101 Data Types to IEC 61859 Data Type

Supported data type combinations are listed below:

option iec101_type_id (IEC101 explanation)	option iec61850_do (IEC61850 explanation)	IEC101 point R/W	IEC61850 point R/W
'1' SPI (Single Point Information)	'SPS' Single-point status	read only	read only
'1' SPI (Single Point Information)	'SPC' Controllable single-point	read-write	read-write
'1' SPI (Single Point Information)	'SPG' Single point setting	--	write only
'3' DPI (Double Point Information)	'DPS' Double-point status	read only	read only
'3' DPI (Double Point Information)	'DPC' Controllable double-point	read-write	read-write
'11' Measured value, scaled value short signed	'INS' Integer status	read only	read only
'11' Measured value, scaled value short signed	'STV' Status value	read only	read only
'11' Measured value, scaled value short signed	'ENS' Enumerated Status	read only	read only
'11' Measured value, scaled value short signed	'ENC' Controllable enumerated status	read-write	read-write
'11' Measured value, scaled value short signed	'ENG' Enumerated status setting	--	write only
'11' Measured value, scaled value short signed	'INC' Controllable integer status	read-write	read-write
'11' Measured value, scaled value short signed	'CMD' Command	--	write only
'11' Measured value, scaled value short signed	'ING' Integer status setting	--	write only
'11' Measured value, scaled value short signed	'MV' Measured Value	read only	read only
'13' Measured value, short floating point number	'MV' Measured Value	read only	read only
'13' Measured value, short floating point number	'APC' Controllable analog set point	read-write	read-write
'13' Measured value, short floating point number	'SPV' Set point value	--	write only
'13' Measured value, short floating point number	'ASG' Analog setting	--	write only

43.8.1. IEC 61850 To IEC 101 Conversion Using UCI

```
root@VA_router:~# uci show iecd
iecd.main=iecd iecd.main.enable=1
iecd.port1=port iecd.port1.enable=1
iecd.port1.loglevel=5
iecd.port1.tcp_keepalive_enabled=1
iecd.port1.tcp_keepalive_interval=5
iecd.port1.tcp_keepalive_timeout=5
iecd.port1.tcp_keepalive_count=3
iecd.port1.tcp_user_timeout=20000
iecd.port1.master_protocol=iec101
iecd.port1.slave_protocol=iec61850
iecd.port1.ioa_offset=0
iecd.port1.pointmap_file=/root/iecd/iecd_points1.csv
iecd.port1.iec104_local_ip=0.0.0.0
iecd.port1.iec104_local_tcpport=2404
iecd.port1.iec104_k=12
iecd.port1.iec104_w=9
iecd.port1.iec104_t2=10000
iecd.port1.iec104_gi_resp_time=200
iecd.port1.iec104_txq_size=128
iecd.port1.iec104_sync_time=1
iecd.port1.iec104_time_tagged_cmds=0
iecd.port1.iec104_cmd_delay_time=5000
iecd.port1.iec104_fsm_debug_on=0
iecd.port1.iec104_dump_data=0
iecd.port1.iec104_trace_on=0
iecd.port1.iec61850_local_ip=0.0.0.0
iecd.port1.iec61850_local_tcpport=104
iecd.port1.iec101_target_ip=127.0.0.1
iecd.port1.iec101_target_tcpport=999
iecd.port1.iec101_mode=unbalanced
iecd.port1.iec101_cot_tx_length=1
iecd.port1.iec101_cot_source_octet=0
iecd.port1.iec101_asdu_addrlen=1
```

```
iecd.port1.iec101_info_obj_addrlen=2
iecd.port1.iec101_data_polling_time=500
iecd.port1.iec101_ack_delay=0
iecd.port1.iec101_link_addrlen=1
iecd.port1.iec101_frame_rsp_time=2000
iecd.port1.iec101_max_tx_retry=3
iecd.port1.iec101_txq_size=128
iecd.port1.iec101_send_spont_delay_acq=1
iecd.port1.iec101_fsm_debug_on=0
iecd.port1.iec101_dump_data=0
iecd.port1.iec101_trace_on=0
iecd.@iec101link[0]=iec101link
iecd.@iec101link[0].portno=1
iecd.@iec101link[0].address=6
iecd.@iec101link[0].asduaddr=6
iecd.@point[0]=point
iecd.@point[0].portno=1
iecd.@point[0].iec61850_ld=SENSORS
iecd.@point[0].iec61850_ln=LLN0
iecd.@point[0].iec61850_do=SPSS01
iecd.@point[0].iec104_type_id=1
iecd.@point[0].iec104_ioa=5
iecd.@point[0].iec101_type_id=1
iecd.@point[0].iec101_ioa=5
iecd.@point[0].devaddr=1
iecd.@point[0].group=1
iecd.@point[0].index=0
iecd.@point[0].dword=0
iecd.@point[1]=point
iecd.@point[1].portno=1
iecd.@point[1].iec61850_ld=SENSORS
iecd.@point[1].iec61850_ln=LLN0
iecd.@point[1].iec61850_do=SPSS02
iecd.@point[1].iec104_type_id=1
iecd.@point[1].iec104_ioa=6
iecd.@point[1].iec101_type_id=1
```

```
iecd.@point[1].iec101_ioa=6
iecd.@point[1].devaddr=1
iecd.@point[1].group=1
iecd.@point[1].index=0
iecd.@point[1].dword=0
```

IEC 61850 to IEC 101 Conversion using Package Options

```
root@VA_router:~# uci export iecd package iecd
config iecd 'main'
option enable '1'
config port 'port1'
option enable '1'
option loglevel '5'
option tcp_keepalive_enabled '1'
option tcp_keepalive_interval '5'
option tcp_keepalive_timeout '5'
option tcp_keepalive_count '3'
option tcp_user_timeout '20000' option master_protocol 'iec101' option slave_protocol 'iec61850' option ioa_offset '0'
option pointmap_file '/root/iecd/iecd_points1.csv'
# IEC104 related settings
option iec104_local_ip '0.0.0.0'
option iec104_local_tcpport '2404'
option iec104_k '12'
option iec104_w '9'
option iec104_t2 '10000'
option iec104_gi_resp_time '200'
option iec104_txq_size '128'
option iec104_sync_time '1'
option iec104_time_tagged_cmds '0'
option iec104_cmd_delay_time '5000'
option iec104_fsm_debug_on '0'
```

```
option iec104_dump_data '0'

option iec104_trace_on '0'

# IEC61850 related settings

option iec61850_local_ip '0.0.0.0'

option iec61850_local_tcpport '104'

option iec101_target_ip '127.0.0.1'

option iec101_target_tcpport '999' option iec101_mode 'unbalanced' option iec101_cot_tx_length '1'

option iec101_cot_source_octet '0'

option iec101_asdu_addrlen '1'

option iec101_info_obj_addrlen '2'

option iec101_data_polling_time '500'

option iec101_ack_delay '0'

option iec101_link_addrlen '1'

option iec101_frame_rsp_time '2000'

option iec101_max_tx_retry '3'

option iec101_txq_size '128'

option iec101_send_spont_delay_acq '1'

option iec101_fsm_debug_on '0'

option iec101_dump_data '0'

option iec101_trace_on '0'

# The following section defines IEC101 slave links used in IEC101 unbalanced mode on

# Each link is defined by a config block 'config iec101link'

# To add more links repeat the section block for each added link. To remove links, simply remove the link block from the
configuration

# Maximum 32 links are supported #

# Definition of options within the section block:

# portno - port number to which this point belongs (1 to 4) # address - IEC101 slave link address

# asduaddr IEC101 slave common ASDU address
```

```
config iec101link
option portno 1
option address 6
option asduaddr 6
config point
option portno '1'
option iec61850_ld 'SENSORS'
option iec61850_ln 'LLN0'
option iec61850_do 'SPSS01'
option iec104_type_id '1'
option iec104_ioa '5'
option iec101_type_id 1
option iec101_ioa '5'
option devaddr '1'
option group '1'
option index '0'
option dword '0'
config point
option portno '1'
option iec61850_ld 'SENSORS'
option iec61850_ln 'LLN0'
option iec61850_do 'SPSS02'
option iec104_type_id '1'
option iec104_ioa '6'
option iec101_type_id 1
option iec101_ioa '6'
option devaddr '1'
option group '1'
option index '0'
option dword '0'
```

43.9. SCADA IEC 104 Gateway Diagnostics

The iecd and tserv background services are started automatically at router power up. You can manually stop, start or restart these services as follows:

iecd

```
/etc/init.d/iecd stop – stops IECD service

/etc/init.d/iecd start – starts IECD service

/etc/init.d/iecd restart – stops and starts IECD service
```

tservd

```
/etc/init.d/tservd stop – stops TSERVD service

/etc/init.d/ tservd start – starts TSERVD service

/etc/init.d/ tservd restart – stops and starts TSERVD service
```

Events

The diagnosing and protocol tracing on the router the following features are available:

- Viewing syslog events (error messages)
- Running and viewing protocol traces (using syslog)
- Viewing statistic counters and debug information using diagnostic commands

To see the appropriate debug information, you must enable different debug options.

The following table summarizes various options for tracing and diagnostics of the IEC 104 to IEC 101/DNP3/ Modbus conversion:

Diagnostic Feature	IEC 104	IEC 101	DNP3	Modbus
Protocol Tracing	option log_severity '7' option iec104_trace_on '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option iec101_trace_on '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option dnp3_trace_on '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option modbus_trace_on '1' /etc/init.d/iecd restart logread -f
Viewing Rx / Tx Hex dump	option log_severity '7' option iec104_dump_data '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option iec101_dump_data '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option dnp3_dump_data '1' /etc/init.d/iecd restart logread -f	option log_severity '7' option modbus_dump_data '1' /etc/init.d/iecd restart logread -f
Viewing Statistics	iec show stats	iec show stats	iec show stats	iec show stats
Clearing Statistics	iec clear stats	iec clear stats	iec clear stats	iec clear stats
Viewing debug information	N/a	N/a	N/a	iec show modbus debug
View point loaded points	iec show points	iec show points	iec show points	iec show points

Viewing Statistics

To view IEC 104 gateway statistics, enter:

```
root@VA_router:~/iecd# iec show stats
```

Modbus Stats

```
Modbus DL Frames Rx 20 Tx 3845 TxErrs 0
Modbus DL CRCErrs 0 Bad Addr 0 LengthErrs 0 UnknownPeer 0 SessionClose 0
Modbus App PDU Rx 20 PDU Tx 3845 PDU Rx Errors 0 PDU Rx Exception 0
Modbus App PDU Rx Timeout 3825 Unknown DevAddr 0 Rx Unexpected FC 0
Modbus App PDU TxQ Overrun 0
```

IEC104 Stats

```
IEC104 DL state: CLOSED
IEC104 DL uptime: 0 hrs 0 mins 0 secs
IEC104 DL PktsRx 15 PktsTx 21 TxQ Overrun 0
IEC104 App ASDU Rx 6 ASDU Tx 12 Bad ASDU 0
```

Viewing Point Mappings

To view IEC 104 gateway point mappings, enter:

```
root@VA_router:~/iecd# iec show points
==== IEC104 point map: ====
IEC 104 Types Legend:
---------------
SPI: Single point information (1 bit)
DPI: Double point information (2 bit)
MVA: Measured normalized value (16 bit signed)
MVAFP: Measured value, floating point number (32 bit signed)
SVA: Measured scaled value (16 bit signed)
BSTR32: Bitstring of 32 bits
IT: Integrated Total (Counter 32 bit)
CP24: with 3 octet time tag CP24Time2a
CP56: with 7 octet time tag CP56Time2a
NQD: Without quality descriptor
---------------
(#1) IOA=64213, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 1, Type 0 (Discreet Input (1bit))
(#2) IOA=64214, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 2, Type 0 (Discreet Input (1bit))
(#3) IOA=64215, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 9, Type 0 (Discreet Input (1bit))
(#4) IOA=64216, Val=0x00000000, IEC104TypeId=30 (SPI-CP56) DevAddr 1 Modbus pt 10, Type 0 (Discreet Input (1bit))
(#5) IOA=64217, Val=0x00000000, IEC104TypeId=34 (MVA-CP56) DevAddr 1 Modbus pt 2, Type 1 (Input Register (16 bit))
(#6) IOA=64218, Val=0x00000000, IEC104TypeId=34 (MVA-CP56) DevAddr 1 Modbus pt 7, Type 1 (Input Register (16 bit))
(#7) IOA=64219, Val=0x00000000, IEC104TypeId=34 (MVA-CP56) DevAddr 1 Modbus pt 1, Type 2 (Holding Register (16 bit))
```

44. DNP3 Outstation Application

Merlin routers have a feature that allows the router to operate as a DNP3 outstation application. A DNP3 SCADA master can poll the router and obtain the following information:

- Router uptime in seconds.
- The serial number of the router.
- The status of up to two router interfaces.

Configuration Packages Used

Package	Sections
dnposd	dnposd

44.1. Configuring DNP3 Outstation Using The Web Interface

To configure the DNP3 outstation, from the top menu select **Services -> DNP3 Outstation**.

Check the **Enable** box and fill in your unique parameters.

The router listens for inbound UDP connections from the SCADA master on the specified port.

The web automatically names the `dnposd` config section 'main'.

DNP3 outstation settings

Web Field/UCI/Package Option	Description
Web: Enable UCI: dnposd.main.enabled Opt: enabled	Enables the DNP3 outstation application on the router. Default: 0 — Disabled 1 — Enabled
Web: Local DNP Address UCI: dnposd.main.local_address Opt: local_address	Defines the DNP3 address of the router. Default: Blank Range 0-65535
Web: Master DNP Address UCI: dnposd.main.master_address Opt: master_address	Defines the DNP3 address of the SCADA master. Default: Blank Range 0-65535
Web: Master IP Address UCI: dnposd.main.master_host Opt: master_host	Defines the IP address of the SCADA master. Only requests from this host will be processed.
Web: Local Port UCI: dnposd.main.local_port Opt: local_port	Defines the UDP port for the router to listen on for incoming DNP3 messages from the SCADA master. Default: 20000 Range 0-65535
Web: Monitor Interface1 UCI: dnposd.main. monitor_if1 Opt: monitor_if1	Defines the first interface to monitor for status. Note: the interface names need to exactly match to the physical names. You can view the physical name by using the ifconfig command via command line. Default: Blank Range 0-65535
Web: Monitor Interface2 UCI: dnposd.main. monitor_if2 Opt: monitor_if2	Defines the second interface to monitor for status. Note: the interface names need to exactly match to the physical names. You can view the physical name by using the ifconfig command via command line. Default: Blank Range 0-65535

44.2. Configuring DNP3 Outstation Using Command Line

DNP3 outstation is configured under the `dnposd` package /etc/config/dnp3osd

DNP3 Outstation using UCI

```
root@VA_router:~# uci show dnposd
dnposd.main=dnposd
dnposd.main.local_port=20000
dnposd.main.enabled=yes
dnposd.main.local_address=1
dnposd.main.master_address=2
dnposd.main.master_host=10.1.10.21
dnposd.main.monitor_if1=wwan0
dnposd.main.monitor_if2=pppoa-DSL
```

Modify these commands by running a `uci set <parameter>` command followed by `uci commit`.

DNP3 Outstation using Package Options

```
root@VA_router:~# uci export dnposd
package dnposd
config dnposd 'main'
option local_port '20000'
option enabled 'yes'
option local_address '1'
option master_address '2'
option master_host '10.1.10.21'
option monitor_if1 'wwan0'
option monitor_if2 'pppoa-DSL'
```

44.3. DNP3 Outstation Diagnostics

Restarting dnposd

To restart dnposd service, enter:

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

Tracing DNP3 Packets

By default, the DNP3 outstation listens on UDP port 20000. To trace UDP packets on port 20000, enter:

```
root@VA_router:~# tcpdump -i any -n udp -p port 20000 &
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 65535 bytes
```

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

```
root@VA_router:~# fg

tcpdump -i any -n udp -p port 20000

^C

33 packets captured

33 packets received by filter

0 packets dropped by kernel
```

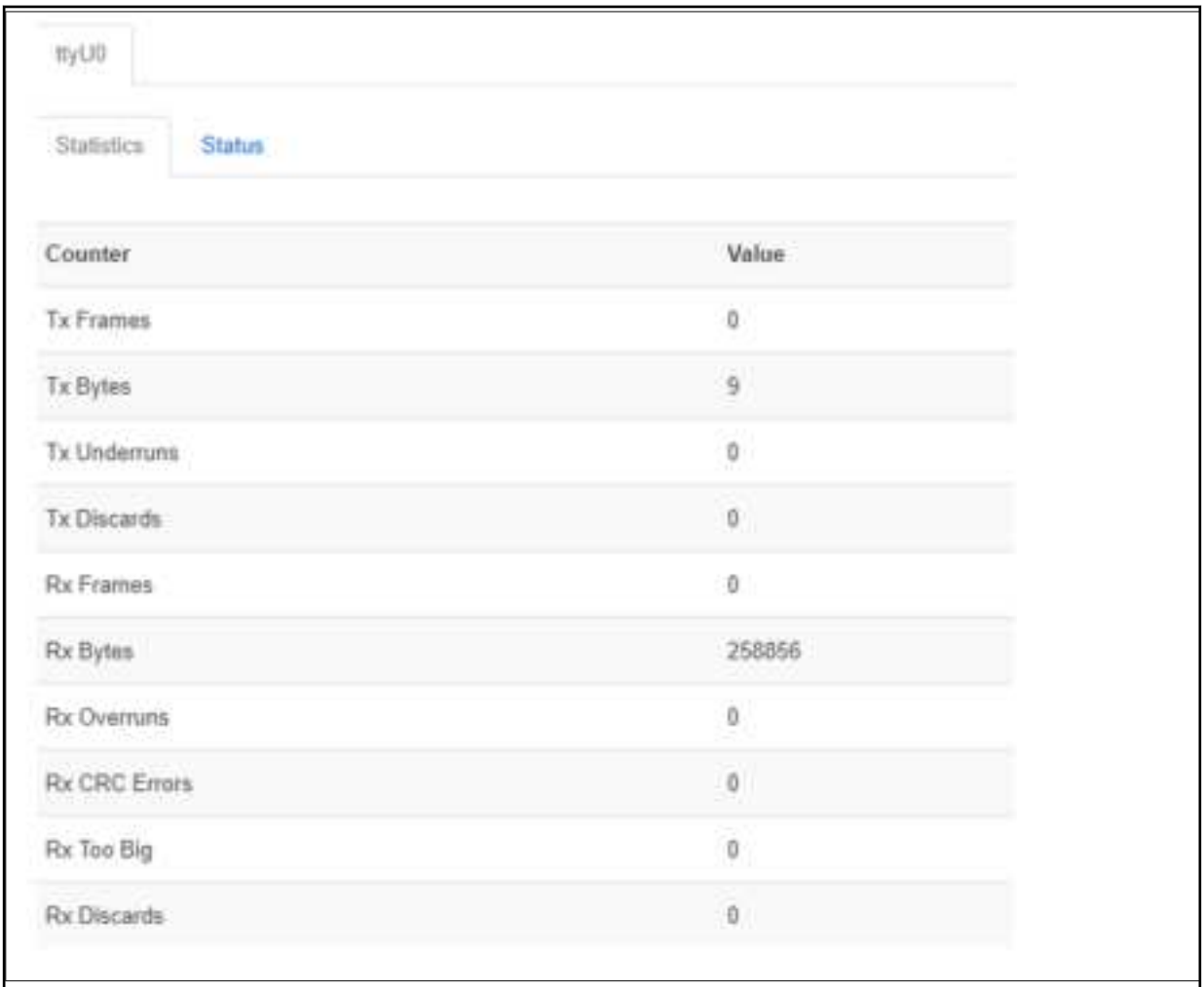
45. Serial Interface

Many different applications and device drivers use the serial interface. You configure the serial interface using the relevant application; for example Terminal Server, therefore there is no standalone serial configuration page. You can monitor the various serial interfaces using either the command line or the web interface.

45.1. Monitoring Serial Interfaces Using The Web Interface

In the top menu, select **Status -> Serial Interfaces**. Depending on the number of serial interfaces present on 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.

Serial Statistics



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

The serial statistics page for serial-0

The serial status page for serial-0

45.2. 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 five sections in the mobile manager page:

Section	Description
Basic settings	Enable SMS, configure SIM pin code, select roaming SIM, collect ICCCIDs and set IMSI.
Advanced	Configure advanced options such as collect ICCIDs and temperature polling interval.
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.	

45.3. Monitoring Serial Interfaces Using Command Line

Serial Statistics using Command Line

To view serial statistics, enter:

```
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
```

Serial Status using Command Line

To view serial statistics, enter:

```
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
```

Resetting Serial Statistics

To reset serial statistics, enter:

```
root@VirtualAccess:~# serial_stats_reset ttyU0
Serial interface statistics reset
```

You can reset statistics for all or individual serial interfaces.

Westermo • Metallverksgatan 6, SE-721 30 Västerås, Sweden
Tel +46 16 42 80 00 Fax +46 16 42 80 01
E-mail: info@westermo.com
www.westermo.com

EN_REV D 2026 07 @ Westermo Network Technologies AB