

DATACOM



DmOS

DATACOM OPERATING SYSTEM

Version 7.0.2

QUICK CONFIGURATION GUIDE

204.4292.29 - December 29, 2021

Contacts

Technical Support

Datacom has available a support portal - DmSupport, to help the customers in use and config of our equipment.

Access to the DmSupport can be made through link: <https://supportcenter.datacom.com.br>

In this site the following are available: firmwares, technical datasheets, config guide, MIBs and manuals for download. In addition, it allows opening of calls for assistance with our technical team.

Telephone Number: **+55 51 3933-3122**

We would like to highlight that our assistance through telephone support is available from Monday through Friday from 08:00 AM through 05:30 PM.

Important: For support assistance 24x7, please request a quotation to our sales department.

General Information

For any other additional information, please visit the <https://www.datacom.com.br/en> or call:

DATACOM

Rua América, 1000

92990-000 - Eldorado do Sul - RS - Brazil

+55 51 3933-3000

Product Documentation

This document is part of a set of documents prepared to provide all necessary information about DATACOM products.

Software Platform

- **Quick Configuration Guide** - Provides instructions on how to set functionalities in a quick manner in the equipment
- **Troubleshooting Guide** - Provides instructions on how to analyze, identify and solve problems with the product
- **Command Reference** - Provides all the commands related to the product
- **Release Notes** - Provides instructions on the new functionalities, identified defects and compatibilities between Software and Hardware

Hardware Platform

- **Datasheet** - Provides the Hardware and Software technical characteristics of product
- **Installation Guide** - Provides instructions on the procedures covering product installation

The availability of some documents can vary depending on the type of product.

Access <https://supportcenter.datacom.com.br> to locate the related documents or contact the Technical Support for additional information.



Introduction to the document

About this Document

The present document is a set of instructions that provide a quick and objective explanation on the use of the functionalities available in the product. It also covers the initial configs that are generally required immediately after installation of the product.

This document was developed to be used as an eventual source for solution of technical issues, and for this reason its sequential reading is not mandatory. However, if you are setting the equipment and is not familiar with the product, reading of the document since the beginning is recommended.

It is presumed that the individual or individuals that manage any aspect of the product should have the basic knowledge of Ethernet, network protocols and communication networks in general.

Audience







This guide is directed to network administrators, technicians or teams qualified to install, set, plan and maintain this product.

Conventions

To facilitate understanding of the present manual, the following conventions were adopted:

Icons

Icon	Type	Description
	Note	The notes explain in a better manner a detail included in the text.

Icon	Type	Description
	Note	WEEE Directive Symbol (Applicable in the European Union and other European countries with separate collection systems). This symbol on the product or its packaging indicates that this product must not be disposed of with other waste. Instead, it is your responsibility to dispose of your waste equipment by handing it over to a designated collection point for the recycling of waste electrical and electronic equipment. The separate collection and recycling of your waste equipment at the time of disposal will help conserve natural resources and ensure that it is recycled in a manner that protects human health and the environment. For more information about where you can drop off your consumer waste equipment for recycling, please contact your local city recycling office or the dealer from whom you originally purchased the product.
	Warning	This symbols means that, case the procedure was not correctly followed, may exist electrical shock risk.
	Warning	Represents laser radiation. It is necessary to avoid eye and skin exposure.
	Warning	Non-ionizing radiation emission.
	Caution	This symbol means that this text is very important and, if the orientations were not correct followed, it may cause damage or hazard.
	Caution	Indicates that equipment, or a part is ESDS (Electrostatic Discharge Sensitive). It should not be handled without grounding wrist strap or equivalent.



A warning icon requests special attention to the conditions that, if not avoided, may cause physical damages to the equipment.



A caution icon requests special attention to the conditions that, if not avoided, may result in risk of death of serious injury.

Table of Contents

Contacts	2
Product Documentation	3
Introduction to the document	4
1 Basic Management	17
1.1 First login	17
1.1.1 Installing and energizing the equipment	17
1.1.2 Accessing the equipment using the console port	17
1.1.3 Accessing the equipment using the out-of-band management port	17
1.1.4 Accessing the equipment for the first time	18
1.2 Firmware Management	19
1.2.1 DmOS Software Update	19
1.2.2 Firmware downgrade	20
1.2.3 ONUs Software Update	21
1.3 CLI Overview	22
1.3.1 Operational Mode	22
1.3.2 Configuration Mode	23
1.3.3 Configuration Types	24
1.3.4 Create Alias Command	25
1.4 Configuration Management	26
1.4.1 Stored Configurations	26
1.4.2 Restoring Configuration	27
1.4.3 Restoring Factory Config	27
1.5 Files Management	27
1.5.1 Saving the Configuration in File	28
1.5.2 Exporting the Files	28
1.5.3 Importing the Files	28
1.5.4 File Handling	28
1.5.5 Updating config with a saved file	29
1.5.6 File edit	29
1.5.7 Exporting the SNMP MIBs	29
2 Equipment Management	30
2.1 Password Configuration	30
2.2 Password Reset	30
2.2.1 Configuration reset	31
2.2.2 Admin user password reset	31

2.3 Licenses Configuration	31
2.3.1 Enabling MPLS license	32
2.3.2 Verifying Licensing	32
2.4 Product Model Configuration	32
2.4.1 DM4270 24XS Product Model Configuration	33
2.5 Forwarding Resources Configuration	33
2.6 Management Configuration	35
2.6.1 Configuring Out-of-Band Management	35
2.6.2 Configuring In-Band Management	36
2.7 CLI Access Configuration	36
2.7.1 Generating SSH Keys	37
2.7.2 Enabling SSH Legacy Support	37
2.7.3 Configuring Maximum Connections of SSH and Telnet	37
2.7.4 Enabling Telnet Service	37
2.8 Hostname Configuration	38
2.8.1 Configuring Hostname	38
2.9 Banner Configuration	38
2.9.1 Configuring Banner in Single Line	38
2.9.2 Configuring Banner in Multiple Line	38
2.9.3 Verifying Banner	39
2.10 System Clock and Date Configuration	39
2.10.1 Configuring System Clock	39
2.10.2 Configuring Timezone	39
2.10.3 Verifying Clock	40
3 Network Management	41
3.1 LLDP Configuration	41
3.1.1 Configuring LLDP between two Neighbors	41
3.1.2 Verifying LLDP	41
3.2 SNTP Configuration	42
3.2.1 Configuring SNTP	42
3.2.2 Configuring SNTP with Authentication	43
3.2.3 Verifying SNTP	43
3.3 Syslog Configuration	43
3.3.1 Configuring Remote Syslog	44
3.3.2 Verifying Syslog	45
3.4 SNMP Configuration	45
3.4.1 Configuring SNMPv2	46
3.4.2 Configuring SNMPv3	47

3.4.3 Configuring the sending of SNMP notifications	48
3.4.4 Configuring SNMP parameters	50
3.4.5 Verifying SNMP	53
3.5 Ping	54
3.6 Traceroute	55
3.7 SSH Client	55
3.8 Telnet Client	56
3.9 Tcpdump	56
3.9.1 Example of using filters	57
3.9.2 Generate and export pcap file	57
4 OAM	58
4.1 CFM Configuration	58
4.1.1 Configuring CFM	58
4.1.2 Configuring CFM with QinQ	60
4.1.3 Enabling Alarm Indication Signal (ETH-AIS)	62
4.1.4 Enabling Action Block	63
4.1.5 Enabling Action Shutdown	65
4.1.6 Fault management	66
4.1.7 Ethernet Delay Measurement (ETH-DM)	66
4.1.8 Verifying CFM	68
4.2 EFM Configuration	68
4.2.1 Enabling EFM	69
4.2.2 Verifying EFM	69
4.3 RDM Configuration	69
4.3.1 Configuring RDM as a slave	70
4.3.2 Verifying RDM	71
4.4 Traffic Loop Configuration	71
4.4.1 Configuring Traffic Loop for L2 Traffic Validation	72
4.5 TWAMP Configuration	72
4.5.1 Configuring a TWAMP session	73
4.5.2 Configuring ACLs in TWAMP Reflector	75
4.5.3 Configuring TWAMP in VRF	75
4.5.4 Calculating the maximum number of session in TWAMP Reflector	75
4.5.5 Verifying TWAMP	76
4.6 sFLOW Configuration	76
4.6.1 Configuring sFLOW	77
4.7 Task scheduling configuration	78
4.7.1 Automatic reboot configuration	78

4.7.2 Automatic backup configuration	78
4.7.3 Executing a task manually	79
4.7.4 Running a task from a pattern	79
4.7.5 Verifying Assistant Task	81
4.8 User Defined Counters Configuration	81
4.8.1 VLAN counters configuration	81
4.8.2 Interface counters configuration	82
4.8.3 Verifying Counters	82
5 Users Authentication	83
5.1 Local Users Configuration	83
5.1.1 Creating a new Local User	84
5.1.2 Deleting a Local User	84
5.2 TACACS+ Configuration	84
5.2.1 Configuring a TACACS+ Server	84
5.3 RADIUS Configuration	86
5.3.1 Configuring a RADIUS Server	86
5.4 Authentication Order Configuration	87
5.4.1 Configuring RADIUS with higher priority	87
5.4.2 Configuring TACACS+ with higher priority	87
6 Interfaces	88
6.1 Ethernet Interfaces Configuration	88
6.1.1 Configuring Ethernet Interfaces	88
6.1.2 Configuring Ethernet Interfaces Range	89
6.1.3 Configuring Ethernet Interfaces Description	89
6.1.4 Configuring Ethernet Interfaces MTU	90
6.1.5 Configuring Ethernet Interfaces TPID	90
6.1.6 Configuring a 10Gbps Interface to operate in 1Gbps	91
6.1.7 Verifying Ethernet Interfaces	92
6.2 Link Aggregation Configuration	92
6.2.1 Configuring a LAG in static mode	92
6.2.2 Configuring a LAG in dynamic mode (LACP)	93
6.2.3 Configuring the load balancing hash	94
6.2.4 Configuring the load balancing	94
6.2.5 Configuring maximum and minimum number of active links in a LAG	95
6.2.6 Verifying Link Aggregation	96
6.3 Backup-Link Configuration	96
6.3.1 Backup-Link Configuration	96
6.3.2 Verifying Backup-Link	98

6.4 Port Mirroring Configuration	99
6.4.1 Configuring Port Mirroring for received traffic	99
6.4.2 Configuring Port Mirroring for transmitted traffic	99
6.4.3 Configuring Port Mirroring for transmitted and received traffic	100
6.5 Link Flap Detection Configuration	100
6.5.1 Configuring Link Flap Detection on Ethernet Interface	100
6.5.2 Verifying Link Flap	101
6.6 Hold Time Configuration	101
6.6.1 Hold Time Configuration	102
6.6.2 Verifying Hold Time	102
6.7 DWDM Interface Configuration	102
6.7.1 DWDM Interface Configuration	102
6.7.2 Verifying DWDM Interface	103
7 GPON	105
7.1 Basic Operation of GPON	105
7.1.1 Setting the GPON interface	105
7.1.2 Setting the ONUs Authentication Method	106
7.1.3 Discovering the ONUS	107
7.1.4 ONU Provisioning	107
7.1.5 ONU Removing	107
7.1.6 GPON basic verification	108
7.2 GPON Profiles	108
7.2.1 Loading the Default Profiles	109
7.2.2 Bandwidth Profile	110
7.2.3 Line Profile	110
7.2.4 Media Profile	111
7.2.5 SIP Agent Profile	112
7.2.6 SNMP Profile	113
7.2.7 GEM Traffic Agent Profile	114
7.2.8 Residential Gateway Profile (RG-Profile)	115
7.2.9 TR-069 ACS Profile	118
7.2.10 Verifying GPON profiles	119
7.3 GPON Service Type	119
7.3.1 Service VLAN N:1	119
7.3.2 Service VLAN 1:1	120
7.3.3 Service VLAN TLS	120
7.4 Mapping the Service Port	120
7.4.1 Service Port - Transparent	120

7.4.2 Service Port - Replace	121
7.4.3 Service Port - Add	121
7.5 Setting GPON Application	121
7.5.1 Configuring a N:1 Application with ONU bridge	122
7.5.2 Configuring a 1:1 Application with ONU bridge	123
7.5.3 Configuring a TLS Application with ONU router	124
7.5.4 Configuring a GPON application with MPLS	125
7.5.5 Verifying GPON applications	125
7.6 Automatic Provisioning of ONUs	126
7.6.1 Verifying GPON	127
7.6.2 Verifying automatic provisioning	128
8 Switching	129
8.1 MAC Table Configuration	129
8.1.1 Configuring Aging Time	129
8.1.2 Disabling MAC learning	130
8.1.3 Verifying MAC Address Table	130
8.2 VLAN Configuration	130
8.2.1 Configuring VLAN with Tagged Interfaces	130
8.2.2 Configuring VLAN with Untagged Interfaces	131
8.2.3 Configuring VLAN Translate	132
8.2.4 Configuring QinQ	133
8.2.5 Configuring Selective QinQ	133
8.2.6 Verifying VLAN Configuration	134
8.3 RSTP Configuration	135
8.3.1 Configuring a Basic RSTP	135
8.3.2 Applying RSTP parameters	136
8.3.3 Verifying RSTP	137
8.4 MSTP Configuration	137
8.4.1 Configuring MSTP for load balancing	138
8.4.2 Verifying MSTP	139
8.5 EAPS Configuration	140
8.5.1 Configuring a Basic Ring EAPS	140
8.5.2 Verifying EAPS	141
8.6 ERPS Configuration	141
8.6.1 Configuring an ERPS Single-ring	142
8.6.2 Configuring an ERPS Multi-ring	143
8.6.3 Verifying ERPS	147
8.7 L2CP Configuration	148

8.7.1 Configuring L2CP in extended mode	148
8.7.2 Configuring L2CP by Specific Protocol	149
8.7.3 Configuring L2CP Extended PDU transparency	150
8.7.4 Verifying L2CP	150
8.7.5 PDUs default behavior in OLT platforms	151
8.7.6 PDUs default behavior in Switch platforms	151
8.8 Loopback Detection Configuration	151
8.8.1 Configuring Loopback Detection for access network	152
8.8.2 Verifying Loopback Detection	152
8.9 DHCP Relay L2 Configuration	153
8.9.1 Configuring Circuit ID	153
8.9.2 Configuring filter by MAC/IP	154
8.9.3 Verifying DHCP Relay	155
9 IP Services	156
9.1 IP Addresses Configuration	156
9.1.1 Configuring IPv4 addresses	156
9.1.2 Configuring IPv6 addresses	156
9.1.3 Verifying IP Address	157
9.1.4 MTU configuration in L3 interfaces	157
9.2 IPv6 SLAAC Configuration	158
9.2.1 Verifying IPv6 SLAAC	159
9.3 L3 DHCP Relay Configuration	159
9.3.1 Configuring the L3 DHCP Relay	160
9.3.2 Configuring the DHCP Option globally	161
9.3.3 Configuring the DHCP Option by interface	161
10 Routing	163
10.1 Static Routing Configuration	163
10.1.1 Configuring a Default Static Route	163
10.1.2 Verifying Static Routes	164
10.2 Black hole route configuration	165
10.2.1 IPv4 black hole route configuration	165
10.2.2 IPv6 black hole route configuration	165
10.2.3 Summarization with black hole routes	166
10.3 VLAN Routing Configuration	167
10.3.1 Configuring a Basic Routing between VLANs	167
10.3.2 Verifying Routes	168
10.4 VRF Configuration	168
10.4.1 Configuring a IPV4 VRF Lite	168

10.4.2 Enabling Route Leaking between IPv4 VRFs	170
10.4.3 Configuring a IPV6 VRF Lite	172
10.4.4 Enabling Route Leaking between IPv6 VRFs	174
10.4.5 Verifying VRFs	176
10.5 PBR Configuration	176
10.5.1 Verifying PBR	177
10.6 OSPFv2 Configuration	178
10.6.1 Configuring OSPFv2 in Point to Point Network	178
10.6.2 Configuring OSPFv2 in Broadcast Network	179
10.6.3 Configuring the area in OSPFv2	182
10.6.4 Filtering received OSPFv2 routes	183
10.6.5 Filtering redistributed routes into OSPFv2	184
10.6.6 Enabling ECMP in OSPFv2	185
10.6.7 Route summarization in OSPFv2	187
10.6.8 Verifying OSPFv2	188
10.7 OSPFv3 Configuration	188
10.7.1 Configuring OSPFv3 Point to Point	189
10.7.2 Enabling ECMP in OSPFv3	190
10.7.3 Verifying OSPFv3	192
10.8 BGP Configuration	192
10.8.1 Configuring a eBGP IPv4 Single Homed	192
10.8.2 Configuring route-maps and IPv4 prefix-lists	194
10.8.3 Configuring a iBGP IPv6 Single Homed	196
10.8.4 Configuring route-maps and IPv6 prefix-lists	198
10.8.5 Configuring BGP Communities	199
10.8.6 Verifying BGP	204
10.9 VRRP Configuration	205
10.9.1 Configuring a VRRPv2 to provide High-Availability	205
10.9.2 Verifying VRRP	207
10.10 BFD Configuration	207
10.10.1 Configuring BFD in OSPFv2	208
10.10.2 Verifying BFD	209
11 MPLS	210
11.1 LDP Configuration	210
11.1.1 Verifying LDP	212
11.2 RSVP Configuration	212
11.2.1 Configuring RSVP affinity	212
11.2.2 Configuring RSVP explicit path	219

11.2.3 Verifying RSVP	225
11.3 VPWS Configuration	226
11.3.1 Configuring a VPWS with PW type VLAN - Case 1	229
11.3.2 Configuring a VPWS with VLAN PW type - Case 2	230
11.3.3 Configuring a QinQ VPWS with PW type VLAN	231
11.3.4 Configuring a VPWS with Ethernet PW type - Case 1	233
11.3.5 Configuring a VPWS with Ethernet PW type - Case 2	234
11.3.6 Configuring a VPWS with RSVP	235
11.3.7 Configuring a VPWS with PW Backup	236
11.3.8 Enabling FAT in a L2VPN	240
11.3.9 Configuring a VPWS with GPON access	240
11.3.10 Configuring a VPWS with PW type VLAN and Service Port access	242
11.3.11 Configuring a VPWS with PW type VLAN and Ethernet Loop access	243
11.4 VPLS Configuration	245
11.4.1 Configuring a VPLS with Ethernet PW type	249
11.4.2 Configuring a VPLS with VLAN PW type - Case 1	251
11.4.3 Configuring a VPLS with VLAN PW type - Case 2	253
11.4.4 Configuring a QinQ VPLS with VLAN PW type - Case 1	255
11.4.5 Configuring a QinQ VPLS with VLAN PW type - Case 2	257
11.4.6 Configuring a H-VPLS	258
11.4.7 VPLS with RSVP	260
11.4.8 Enabling TLS in a VPLS	261
11.4.9 Limiting MAC addresses in a VPLS	261
11.4.10 Configuring VPLS with GPON access	262
11.4.11 Configuring VPLS with PW type VLAN and Service Port access	264
11.4.12 Configuring VPLS with PW type Ethernet and Service Port access	266
11.5 Verifying L2VPN	268
11.6 L3VPN Configuration	268
11.6.1 Configuring a L3VPN Site-to-Site	268
11.6.2 Configuring a L3VPN Hub and Spoke	271
11.6.3 Configuring BGP between PEs and CEs	275
11.6.4 Enabling AS Override	277
11.6.5 Enabling Allow AS In	277
11.6.6 Configuring OSPF between PEs and CEs	277
11.6.7 Verifying L3VPNs	279
11.7 6VPE Configuration	279
11.7.1 Configuring a 6VPE L3VPN Site-to-Site	279
11.7.2 Configuring a 6VPE L3VPN Hub and Spoke	283
11.7.3 Configuring BGP between PEs and CEs	287

11.7.4 Enabling AS Override	288
11.7.5 Enabling Allow AS In	289
11.7.6 Verifying 6VPE	289
11.8 BGP Free Core Configuration	289
12 Multicast	292
12.1 IGMP Snooping Configuration	292
12.1.1 Configuring IGMP Snooping in Ethernet Application	292
12.1.2 Configuring IGMP Snooping in GPON Application	293
12.1.3 Verifying IGMP	294
13 QoS	295
13.1 Congestion Control Configuration	295
13.1.1 Configuring WFQ Scheduler	295
13.2 Traffic Shapping Configuration	296
13.2.1 Configuring Rate Limit on Interface	296
13.3 Traffic Policing Configuration	296
13.3.1 Configuring Traffic Policing based on VLANs	297
13.3.2 Configuring Traffic Policing based on inner VLAN	298
13.3.3 Configuring Traffic Policing based on PCP	299
13.3.4 Configuring Traffic Policing based on DSCP	299
13.3.5 Configuring Hierarchical Traffic Policing based on PCP	300
13.3.6 Verifying QoS policers	302
14 Security	303
14.1 Storm Control Configuration	303
14.1.1 Configuring Storm Control	303
14.1.2 Verifying Storm Control	304
14.2 ACL Configuration	304
14.2.1 Configuring ACL L2 to deny traffic of a VLAN	304
14.2.2 Configuring ACL L3 to deny traffic of IPv4 Address	305
14.2.3 Configuring an ACL for CPU protection	305
14.2.4 Configuring an ACL for CPU based packets	307
14.2.5 Verifying ACLs	308
14.3 Anti IP Spoofing Configuration	308
14.3.1 Configuring Anti IP Spoofing for specific IPv4 and MAC address	308
14.3.2 Configuring Anti IP Spoofing for specific IPv4 address	309
14.3.3 Configuring Anti IP Spoofing for all IPv6 addresses	309
14.3.4 Configuring Anti IP Spoofing for all IPv4 and IPv6 addresses	309
14.3.5 Verifying Anti IP Spoofing	310

14.4 MAC Limit Configuration 310

 14.4.1 Configuring MAC Limit on Interface 310

 14.4.2 Configuring MAC Limit on VLAN 310

 14.4.3 Verifying MAC Limit 311

14.5 CPU DoS Protect Configuration 311

 14.5.1 Configuring the CPU DoS Protect 312

 14.5.2 Configuring the CPU DoS Protect per Protocol 312

 14.5.3 Verifying the CPU DoS Protect 312

Legal Note 313

Warranty 313

1 Basic Management

This chapter contains the following sections:

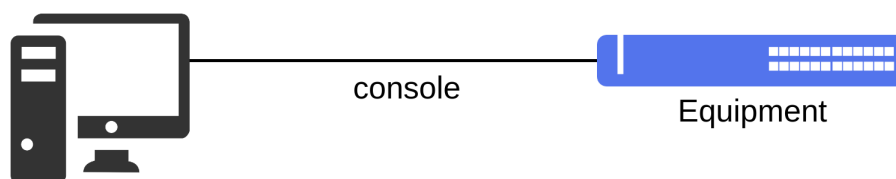
- First login
- Firmware Management
- CLI Overview
- Configuration Management
- Files Management

1.1 First login

1.1.1 Installing and energizing the equipment

Please check the detailed instructions in the equipment **Installation Guide**.

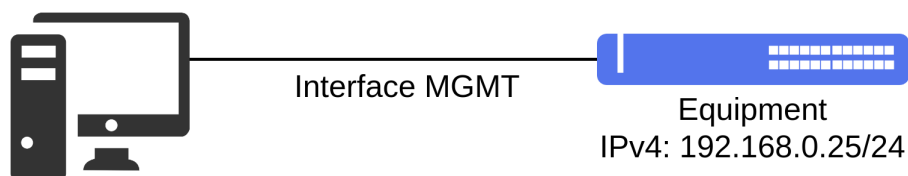
1.1.2 Accessing the equipment using the console port



Accessing the equipment using the console port

Access to equipment CLI can be carried out through the console port. A serial cable is required and a terminal emulator such as Hyper Terminal or similar is needed. The terminal settings should be set as **9600 8N1** without hardware or software flow control.

1.1.3 Accessing the equipment using the out-of-band management port



Accessing the equipment using the out-of-band management port

The equipment CLI can also be accessed through the management port, called MGMT. The MGMT port is an Ethernet port dedicated for management and is not enabled to be used by L2 or L3 protocols.

To access the CLI, a LAN cable should be connected to the MGMT port and an IP address should be configured in the auxiliary PC network card. The equipment default IP address is **192.168.0.25/24**. A SSH application in the auxiliary PC is required to connected to the equipment.

1.1.4 Accessing the equipment for the first time

To login to the equipment for the first time, the **admin** user and **admin** default password must be used.

```
login: admin
Password: admin
Welcome to the DmOS CLI
```



For security purposes, it is highly recommended to change the equipment default password.

Please read the **Users' Authentication** chapter to check how to proceed to change user password.

Using the CLI

The easiest manner to use the command line is simply writing the command and pressing [Enter].

```
# command [Enter]
```

If the command includes also a parameter, the keyword and its arguments should be inserted. The argument specifies how the parameter is changed. Values include numbers, strings or addresses, depending on the keyword. After input of the command [Enter] should be pressed.

```
# command keyword argument [Enter]
```



In the DmOS interfaces are used **chassis/slot/port** naming conventions.



The **!** character is used in CLI commands for indentation and can also be used to add comments in DmOS configuration scripts.

1.2 Firmware Management

The DmOS has two memory locations for firmware storage. After downloading the firmware the image is saved in the inactive or empty position.



Contact DATACOM Technical Support to check the firmware images available for download and installation according to your product and requirements.

1.2.1 DmOS Software Update

To update the equipment using CLI, PC with TFTP, SCP or HTTP server is required in order to send the firmware image to the equipment. The examples below demonstrate how to update the equipment firmware with file named **build.swu** through server with IPv4 address **192.168.0.1**.

To receive the firmware image from a **TFTP server**, use the following command:

```
request firmware add tftp://192.168.0.1/build.swu
```

To receive the firmware image using **SCP**, use the following command:

```
request firmware add scp://192.168.0.1/build.swu username user password "pass"
```

To receive the firmware image using **HTTP**, use the following command:

```
request firmware add http://192.168.0.1/build.swu
```

The received firmware image will be in the *Inactive* position. It is possible to check the download progress of firmware and the newly received firmware using the following command:



As the firmware is written directly on the flash, the equipment removes the firmware that was in the *Inactive* position during the update process, so in case of failure it will be *Empty*.

```
show firmware
```

To activate the firmware that is in the *Inactive* position, use the command below. The equipment will reboot automatically after the firmware activation is completed.

```
request firmware activate
Warning: Firmware downgrade may not be totally supported. Please, refer to
the Hardware and Software Compatibility section in the DmOS Release Notes.
Warning: The system will reboot automatically in order to complete the
activation process. Once initiated this process cannot be interrupted.
Proceed with activation? [no,yes] yes
```



An automatic reboot will occur after the activation process is completed.

After the equipment is rebooted, check if the new firmware is in the **Active** position using the command below.

```
show firmware
```

1.2.2 Firmware downgrade

The firmware downgrade process, should be performed in a controlled manner and some precautions are required to avoid incompatibility issues.

DmOS does not preserve the current configuration during the firmware downgrade process, the last configuration used in the previous version of the installed firmware will be loaded.



If the equipment never had the older firmware installed, it does will not have a configuration backup of that version and will boot with factory configuration.

For details on DmOS version compatibility, the document **DmOS Release Notes** should be consulted.

Below are the steps to take in the DmOS firmware downgrade process.

1- Save current setting to a text file.

```
configure  
save <FILE_NAME>
```

2 - Request firmware for equipment.

```
request firmware add <protocol://ipaddress/path/fw_name>
```

3 - Verify Firmware Appears with Inactive Status.

```
show firmware
```

4 - Activate the firmware.

```
request firmware activate
```

1.2.3 ONUs Software Update

For hardware platforms that support the GPON technology, the ONU firmware image can be sent to the equipment using the CLI. Before executing the next steps, it is important to ensure that all the ONUs to be updated are in the UP operational status.

Firmware Download

To download the ONU firmware to the OLT execute the following procedure:

```
request firmware onu add tftp://192.168.0.1/fw_onu.bin
```



Wait for the message “ONU firmware file download has succeeded” to proceed with the next steps.

ONU Updating

To update only the ONU 1 in the GPON 1/1/1 interface, the user should proceed with the following command:

```
request firmware onu install fw_onu.bin interface gpon 1/1/1 onu 1
```

To check the update progress, use the following command:

```
show interface gpon 1/1/1 onu
```

ID	Serial Number	Oper State	Software Download State	Name
0	DACM00001533	Down	None	CLIENT-01
1	DACM000001E0	Up	Download in progress (60%)	CLIENT-02
126	DACM00001C7B	Up	None	CLIENT-03
127	DTCM10000006	Up	None	CLIENT-04



During the download status, the ONU status will be in **Download in progress**. After a few minutes the ONU will reset automatically with the new firmware, changing the status to Complete.

Alternatively, it is possible to update an ONU firmware via an L3 interface using the following command:

```
request firmware onu install fw_onu.bin in-band-upgrade ip-address 172.24.1.158 model dm984-42x
username support password support
```



Note that the ONU is referred by its IP address configured in the IP host interface, in this case, it is necessary to configure an IP address to the correspondent VLAN in the OLT via L3 interface, as described in [IP Addresses Configuration](#). This connectivity can be validated using ping to the ONU IP host address from the OLT. During copy firmware process the CLI is unavailable to the user until the procedure finishes.

Updating all ONUs of a PON link

To update all the ONUs of a PON link the user should execute the following procedure. The example shows the update of all the ONUs of the 1/1/7 PON link.

```
request firmware onu install fw_onu.bin interface gpon 1/1/7 all
```



Update will be executed in groups of 8 ONUs.

To check the update progress of all the ONUs, use the following command:

```
show interface gpon 1/1/7 onu
```

ID	Serial Number	Oper State	Software Download State	Name
0	DACM00000B4F	Up	Download in progress (97%)	CLIENT-22
1	DACM00000B7C	Up	Download in progress (97%)	CLIENT-23
2	DACM00000B7B	Up	Download in progress (97%)	CLIENT-24
3	DACM00000B92	Up	Download in progress (97%)	CLIENT-25
4	DACM00000B73	Up	Download in progress (97%)	CLIENT-26
5	DACM00000B8A	Up	Download in progress (97%)	CLIENT-31
6	DACM00000B8E	Up	Download in progress (97%)	CLIENT-32
7	DACM00000B78	Up	Download in progress (92%)	CLIENT-33
8	DACM00000B8D	Up	None	CLIENT-34
9	DACM00000B7A	Up	None	CLIENT-35
10	DACM00000B8B	Up	None	CLIENT-36
11	DACM00000B90	Up	None	CLIENT-37
12	DACM00000B96	Up	None	CLIENT-38
13	DACM00000B74	Up	None	CLIENT-39
14	DACM00000B49	Up	None	CLIENT-40
15	DACM00000B58	Up	None	CLIENT-41
16	DACM00000B15	Up	None	CLIENT-42



During the download, the ONUs status will be in **Download in progress**. After a few minutes the ONUs will reset automatically with the new firmware, changing the status to Complete.

1.3 CLI Overview

The equipment can be managed through the CLI using the equipment console port or using TELNET and SSH sessions.

DmOS' CLI supports the **config** and **operational** modes that provide commands related to config, monitoring of software, hardware and network connectivity with other equipment.

1.3.1 Operational Mode

Upon logging in the equipment, the user will enter automatically in the operational mode. In this mode, it is possible to check the equipment information, execute network connectivity test and other items. However, in this mode, it is not possible to execute modifications in the equipment config.



To visualize the list of commands available in this mode, enter the command ?

It is possible to check some information related to the equipment in the operational mode through the following commands:

Command	Description
show platform	Presents the equipment model, modules and firmware in use
show inventory	Presents the equipment inventory, modules and interfaces in use
show environment	Presents the temperature sensor values
show firmware	Presents the firmware version
show running-config	Presents the equipment current config
show system cpu	Presents the equipment values of the CPU in use
show system memory	Presents the values of equipment memory
show system uptime	Presents the equipment time of activity
who	Present the users connected in the equipment

It is possible to execute any command of the operational mode within the config mode adding the keyword **do** before the command. Below is an example:

```
do show running-config
```

1.3.2 Configuration Mode

To modify the config, entering in the config mode through the following command is required:

```
config
```

If the user would like to exit the config mode, it may use the command below at any config hierarchical level or also enter only **[Ctrl]+[Z]**.

```
end
```

If the user would like to return to the first level of config, it is possible to use the command below at any config hierarchical level.

```
top
```

Two options of config mode are available: **terminal** and **exclusive**. If the command config is not completed, with the preferred mode, by default, the terminal mode will be used.

Terminal Mode

In this mode, any config in the equipment changed by other session will conflict with the config of the current session. In the attempt to save a config, a message will the instructions to solve the conflict will be exhibited. The command below is used to enter in this config mode:

```
config terminal
```

By default, if the user enters in the config mode without specifying any specific mode, the mode to be used will be the terminal mode.

```
config
```

Exclusive Mode

When the user enters in the **exclusive** mode, any other simultaneous session will be unable to apply its configs. The command below is used to enter in this config mode:

```
config exclusive
```

1.3.3 Configuration Types

The DmOS uses the **NETCONF** protocol define by the **RFC4741**. The NETCONF defines the existence of one or more saved data configs allowing operation of the config in each one. The DmOS uses two configs, however, only one is executing in fact in the equipment, as follows:

- **Candidate-config:** While the user changes the config and does not execute the commit, the config is saved temporarily in a candidate-config. If the device resets or exits the session, the candidate-config will be lost.
- **Running-config:** After the user executes the commit command the candidate-config is applied to the running-config becoming active in the equipment and in all the software components.

When the user enters in the config mode and starts executing configs, the config in fact is still not being applied in the equipment. In this case, the user is writing the config in the candidate-config. The command below shall exhibit the candidate-config of the hierarchical level in which the user is located:

```
show
```

The next command shall exhibit only the changes introduced in the candidate-config:

```
show configuration
```


To activate and save the candidate-config it is important to copy it to the **running-config**. The command below shall save the candidate-config in the **running-config**.

```
commit
```

However, if the user would like only to check the candidate-config but does not wish to copy it to the **running-config** the following command should be used:

```
commit check
```

The user can also confirm temporarily a candidate-config and wait for a confirmation within a given period of time (standard 10 minutes). If the time expires and the user does not confirm, the config will be reversed to the previous. This option is available only in the **exclusive** config mode.

```
commit confirmed
```

The user may abort the config still to be confirmed and before the time limit using the following command:

```
commit abort
```

To delete all the config changes introduced after the last saved config, the user should use the following command:

```
clear
```

1.3.4 Create Alias Command

The DmOS allows the user to create a custom command, making it possible to return the result of one or more commands as a result of only one command.

Suppose the user often runs a script to check system information.

The steps below show how to set up an alias to return the output of the **show environment**, **show platform**, and **show firmware** commands by running only the **show-system** command.

```
config
alias show-system
expansion "show environment; show platform; show firmware"
commit
```



The alias command does not allow auto-complete.

Create Alias using parameters

The alias supports the use of parameters to run shows, in this way it is possible to pass the ID of an interface, VLAN or any other dynamic information.

Suppose the user wants to view more than one information referring to ONU 5 in PON 1/1/1 in only one command, for example, configuration of ONU and service-port.

The steps below show how to set up an alias to return the output of the **show running-config interface gpon 1/1/1 onu 5** and **show running-config service-port gpon 1/1/1 | context-match onu.5[^0-9]** commands by running the **show-config-onu(<pon>,<onu_id>)** command and passing the interface and ONU IDs as parameters.

```
config
alias show-config-onu(gpon,onu)
  expansion "show running-config interface gpon 1/1/$(gpon) onu $(onu); show running-config
  service-port gpon 1/1/$(gpon) | context-match onu.$(onu)[^0-9]"
commit
```

Below is an example of running show-config-onu (<pon>,<onu_id>).

```
DM4615# show-config-onu(1,5)
DM4615# show running-config interface gpon 1/1/1 onu 5; show running-config service-port gpon
1/1/1 | context-match onu.5[^0-9]
interface gpon 1/1/1
  onu 5
    name DM985-102 108.35
    serial-number DACM911E59C4
    line-profile DM985-102
    tr069-acis-profile DMVIEWBASE-HTTP
    auto-provisioned true
    snmp profile DEFAULT-SNMP
    ipv4 vlan vlan-id 108
    ipv4 static address 172.22.108.35/24 default-gateway 172.22.108.254
    ethernet 1
      negotiation
      no shutdown
    !
    veip 1
  !
!
service-port 18
  gpon 1/1/1 onu 5 gem 1 match vlan vlan-id 108 action vlan replace vlan-id 108 description MGMT
service-port 1062
  gpon 1/1/1 onu 5 gem 3 match vlan vlan-id 1062 action vlan replace vlan-id 1062 description
DHCP-Router-DM985-102
DM4615#
```

1.4 Configuration Management

1.4.1 Stored Configurations

When the user saves a config, a file containing the config changes is generated and stored. To check this list of files, the user should use the following command:

```
show configuration commit list
```



The last 64 committed settings are saved.

1.4.2 Restoring Configuration

If the user would like to reverse the last saved config, it should use the following procedure:

```
rollback configuration  
commit
```

The user may restore the more recently saved configs. To do this, it should use the following procedure:

```
rollback configuration FILE-NAME  
commit
```

However, if the user would like to select only one specific file that is saved without returning to the most recent changes, it should use the following procedure:

```
rollback selective FILE-NAME  
commit
```

1.4.3 Restoring Factory Config



The procedure below shall delete the entire config and load the default config in its position. Route and IP address configs will be lost.

To load a default config in the candidate-config, the user should execute the command:

```
load factory-config
```



It is possible to execute any config prior to execution of the **commit**. Thus, it is possible to maintain the management config if required.

```
commit
```

1.5 Files Management

1.5.1 Saving the Configuration in File

The user may save the candidate-config in a file (including the standard configs) without applying it in the document. The command below shall save the candidate-config in a file called **CANDIDATE-CONFIG**:

```
save CANDIDATE-CONFIG
```

The user may also save configs made in a specific path using the path filter. For example, if the user would like to save only the config of a MGMT interface (including the standard configs) in a file called **INTF-MGMT-CONFIG**, it should use the following command:

```
save INTF-MGMT-CONFIG interface mgmt
```



Care should be taken not to load a saved file that has no full config using the override option.

1.5.2 Exporting the Files

After saving a file, the user may export such file to a TFTP or SCP server. The command below shall forward the file via TFTP protocol saved as **CANDIDATE-CONFIG** to the 172.1.1.1 server.

```
copy file CANDIDATE-CONFIG tftp://172.1.1.1
```

1.5.3 Importing the Files

After exporting a file, the user may import such file from a TFTP or SCP server. The command below shall download the **CANDIDATE-CONFIG** file from 172.1.1.1 server via TFTP protocol.

```
copy file tftp://172.1.1.1 CANDIDATE-CONFIG
```

1.5.4 File Handling

To exhibit all the saved files, the user should use the following command. As it is an operational mode command, the keyword “**do**” should be added in front of the command when in the config mode.

```
file list
```

It is possible to check the content of a saved file using the following command:

```
file show FILE-NAME
```

To delete a file, the following command should be used: :

```
file delete FILE-NAME
```

1.5.5 Updating config with a saved file

It is possible to merge the candidate-config with a saved file using the **merge** option. Thus, if new commands exist in the file, they will be loaded for the candidate-config. If the commands in the file are in conflict with those in the candidate-config, these will replace the commands in the candidate-config.

```
load merge FILE-NAME  
commit
```

Through the **override** command, the user may delete the entire candidate-config and load a new full config of a file:

```
load override FILE-NAME  
commit
```

1.5.6 File edit

Is is possible to edit an existing file or create a new one, if it does not already exist. The file name is limited to 255 characters and must not start with "." ou "-", nor contain directory path.

To edit a file, the following command should be used:

```
file edit FILE-NAME
```

The file editor will be opened. Use "CTRL + s" to save the file and "CTRL + x" to exit the editor and return to DmOS CLI.

1.5.7 Exporting the SNMP MIBs

The user can export a file with all SNMP MIBs supported by equipment to a TFTP or SCP server.

The command below shall forward the MIBs file named **datacom-mibs.tar.gz** via TFTP protocol to the 172.1.1.1 server.

```
copy mibs tftp://172.1.1.1
```

2 Equipment Management

The network administrator may set an equipment with DmOS in two manners:

- **CLI (Comand-Line Interface):** Provides a set of commands to manage the equipment through a Telnet, SSH connection or through the console port.
- **DmView:** It is the NMS (Network Management System) of DATACOM based on SNMP and NETCONF. The DmView is a network management integrated system and elements, designed to supervise and set DATACOM equipment, offering monitoring, config, provisioning, audit, performance, security, discovery, maps and inventory functionalities.

This chapter will guide the user in how to proceed with the equipment management config through CLI. It contains the following sections:

- Password Configuration
- Password Reset
- Licenses Configuration
- Product Model Configuration
- Forwarding Resources Configuration
- Management Configuration
- CLI Access Configuration
- Hostname Configuration
- Banner Configuration
- System Clock and Date Configuration

2.1 Password Configuration



It is recommended to set the protocol passwords always between double quotations marks "password". Thus, it is possible to set passwords with no problems related to use of special characters.

2.2 Password Reset

To perform a password reset in DmOS, it is necessary to access the equipment through the console port and restart it. During the boot process, when the message **Press CTRL-C to stop booting** is displayed, press **CTRL+C** to access *U-Boot*. A password will be requested.

To generate the U-Boot password, it is necessary to contact Datacom Support and inform the equipment serial number and MAC address, shown during boot as below.

```

MGMT PHY: reset OK
Board ID: 0x27270100
Serial Number: 1234567
MAC Address: 00:04:DF:00:00:00
Net: Initializing Fman
SF: Detected n25q64 with page size 256 Bytes, erase size 4 KiB, total 8 MiB
Fman1: Uploading microcode version 106.4.17
FM1@DTSEC4 [PRIME]
SF: Detected n25q64 with page size 256 Bytes, erase size 4 KiB, total 8 MiB
Press CTRL-C to stop booting: (3 seconds to boot)
Password:

```

There are two ways to reset the admin password. In the first case, the configuration is reset to factory default. In the second case, the configuration is kept and only the admin password is reset to the *default* value.

2.2.1 Configuration reset

After accessing U-Boot, insert the commands below. The equipment will boot with the factory configuration and the admin user default password.

```

setenv load-factory-config 1
save
boot

```

2.2.2 Admin user password reset

After accessing U-Boot, insert the commands below. The equipment will boot without changes in the configuration, but with the admin user default password.

```

setenv reset-admin-password 1
save
boot

```

2.3 Licenses Configuration

A license is required for some equipment operations. To check which licenses your equipment support, use the **show license** command. To get the license contact the DATACOM Support team informing the serial number and the MAC address of the equipment. These informations can be obtained in the **show inventory** command as below:

```

show inventory
...
Chassis/Slot      : 1/1
Product model     : 24GX+4XS+2QX
Part number       : 800.5184.01
Serial number     : 4461034
Product revision  : 1
PCB revision      : 1
Hardware version  : 0
Manufacture date  : 01/08/2018
Manufacture hour  : 12:00:00
Operat. temp.     : 0 - 55 Celsius degrees
Base MAC address  : 00:04:df:5c:0c:77
...

```

2.3.1 Enabling MPLS license

The next steps will demonstrate how to activate the MPLS license.



Activation is done after the commit, there is no need to restart the equipment.

```
config
license mpls enabled key
(<string>): *****
commit
```



The available commands for troubleshooting can be found in the topic [Verifying License](#).

2.3.2 Verifying Licensing

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show license
```

2.4 Product Model Configuration

DmOS supports Product Model Configuration to change port configuration if supported by equipment. To check if equipment supports this feature use the **card-model** command.



Changing the Product Model will reboot the equipment and return to factory configuration.



This feature is only supported in DM4270 24XS and the default card-model is **24XS+3CX**.

2.4.1 DM4270 24XS Product Model Configuration

Product Model	Supported Configuration
24XS+3CX	24 ten-gigabit-ethernet + 3 hundred-gigabit-ethernet
24XS+4QX	24 ten-gigabit-ethernet + 4 forty-gigabit-ethernet

DmOS has the below port mapping:



In the firmware version 5.12.0, the equipment supports up to 3 operations 100G (3CX) ports, thus operating with the card-models below. In previous versions the model supports up to 2 100G (2CX) ports.

- **24XS+3CX:** DmOS use the hundred-gigabit-ethernet 1/1/1, hundred-gigabit-ethernet 1/1/2 and hundred-gigabit-ethernet 1/1/3 for 40 or 100 Gigabit ports.
- **24XS+4QX:** DmOS use the hundred-gigabit-ethernet 1/1/1, hundred-gigabit-ethernet 1/1/2, forty-gigabit-ethernet 1/1/1 and forty-gigabit-ethernet 1/1/2 for 40 Gigabit ports.

Example:

```
DmOS# card-model 24XS+4QX
Warning: The system will automatically reboot and load the factory configuration. Once initiated,
this process cannot be interrupted.
Proceed with this action? [yes,N0]
```

2.5 Forwarding Resources Configuration

DmOS supports forwarding resources configuration to define a usage profile to extend MAC table or extend routing table supported by the equipment.

To check if equipment supports this feature use the **forwarding-resources profile** command.



This feature is supported only in DM4270, DM4775 and DM4380.

User can choose between three forwarding resources profiles.

- **default:** Default profile configured on each platform.
- **extended-ip:** Profile used for increased routing table.
- **extended-mac:** Profile used for increased MAC address table.

In the example below it is possible to check the three types of profiles available for the DM4270 platform.

```
DM4270# forwarding-resources profile
Possible completions:
default      128000 L2 MAC / 128000 IPv4 / 32000 IPv6/64 / 4000 IPv6/128
extended-ip   32000 L2 MAC / 168000 IPv4 / 42000 IPv6/64 / 10000 IPv6/128
extended-mac  288000 L2 MAC / 16000 IPv4 / 4000 IPv6/64 / 1000 IPv6/128
```

Below the procedure to configure the profile:



It is necessary to restart the equipment after performing this configuration.

```
DM4270# forwarding-resources profile extended-mac
This change will take effect on next reboot.
DM4270# reboot
```

To check the applied profile:

```
DM4270# show forwarding-resources
Profile Name  MAC      IPv4      IPv6/64   IPv6/128  Running  Startup
-----
default      128000    128000    32000     4000      false   false
extended-ip   32000    168000    42000     10000     false   false
extended-mac  288000    16000     4000      1000      true    true
```

The table displays in the **Running** column the flag **true**. The **Startup** column displays the **true** flag to inform which setting will be active after the next reboot.

Internally, the selection of profiles manages the allocation of the UFT (Unified Forwarding Table), which shares Hardware resources for MACs, IPv4, IPv6/64 and IPv6/128. It is necessary to know its basics in order to understand how the values assigned to each profile relate to each other and how the indicated limits can be reached.

The partition reserved for MACs is fixed for each profile, and it is able to be filled completely without changing the other capacities.

There are also two partitions for allocating IP addresses:

- **ip-a partition:** Stores IPv4 and IPv6/64 addresses
 - Supports to be filled with the maximum capacity for IPv4 routes or the maximum capacity for IPv6/64. A combination of the two can also be established, but it is not possible to achieve the maximum limits simultaneously. IPv6/64 addresses occupy twice the space of an IPv4 address. In some products, when this partition gets full, the ip-b partition will store IPv4 or IPv6/64 addresses, thereby reducing IPv6/128 capacity.
- **ip-b partition:** Stores IPv6/128 addresses
 - Always reaches the maximum capacity indicated in the profile. However, in products where this partition can share IPv4 or IPv6/64 addresses, the limit may not be available if the restrictions depicted for the ip-a partition are not respected.

2.6 Management Configuration

It is possible to set the out-of-band management to maintain the access to the equipment even when the data network is deactivated. If the user is connected by the **MGMT interface**, the session will be disconnected after confirmation. To continue the config of the equipment using the **MGMT interface**, the user should set an IP address in its PC within the same network or connect using the console.



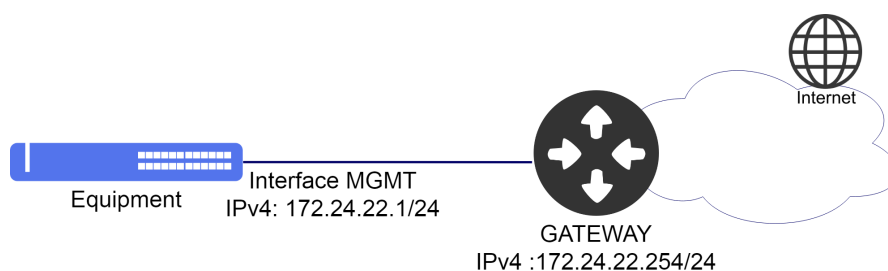
It is possible to set the equipment management using IPv4 or IPv6 address.



It is possible to set the equipment management with a **VRF mgmt**. In this application, only basic services such as SSH, Telnet, Local authentication and firmware updates are supported. See how to set the VRF to proceed with this config.

2.6.1 Configuring Out-of-Band Management

The topology below indicates an example of how to manage the equipment using the **MGMT interface**.



Example of Out-of-Band Management

Considering that the user would like to use the Interface MGMT Interface with the **172.24.22.1/24** IPv4 address and with the **172.24.22.254/24** as standard gateway. The procedure below shall indicate how to execute this config as from the config mode:

```
config
interface mgmt 1/1/1
  ipv4 address 172.24.22.1/24
!
!
router static
  address-family ipv4
    0.0.0.0/0 next-hop 172.24.22.254
commit
```

2.6.2 Configuring In-Band Management

It is possible to set the In-band management to manage the equipment through an interface also used by the data traffic in the network.

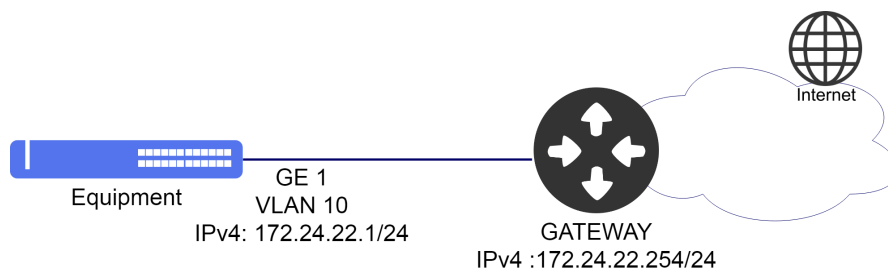


It is possible to set the equipment management using IPv4 or IPv6 address.



It is possible to set the equipment management using secondary IPv4 address. Secondary IPv6 address is not supported.

The diagram below shows an example of how to manage the equipment using an In-Band interface.



Example of In-Band Management

Considering that the user would like to use a **VLAN 10** for In-Band management through the **gigabit-ethernet 1/1/1** interface with **172.24.22.1/24** IPv4 address and **172.24.22.254** as standard gateway. The procedure below shall indicate how to carry out this config:

```

config
dot1q vlan 10
name In_Band-Management
interface gigabit-ethernet-1/1/1
!
interface l3 in-band
ipv4 address 172.24.22.1/24
lower-layer-if vlan 10
!
router static
address-family ipv4
0.0.0.0/0 next-hop 172.24.22.254
commit
  
```

2.7 CLI Access Configuration

The SSH (Secure Shell) and Telnet are protocols used to provide access to the equipment terminal. For security purposes, the DmOS default configuration is the enabled SSH server protocol and the Telnet server deactivated.



The DmOS supports the SSHv2 with encryption of RSA (Rivest, Shamir and Adelman) public key and DAS (Digital System Algorithm).

2.7.1 Generating SSH Keys

The next steps will indicate how to generate the RSA key.

```
ssh-server generate-key rsa size <1024-2048>
Really want to do this? [yes,no] yes
Generated keys
```

2.7.2 Enabling SSH Legacy Support

For security purposes, the SSH clients executing the OpenSSH with versions above version 7.0 are supported. To obtain the compatibility with previous versions, the user should execute the following procedure.

```
config
ssh-server legacy-support
```

2.7.3 Configuring Maximum Connections of SSH and Telnet

By default, 8 SSH connections and 8 Telnet connections with maximum of 16 connections for each protocol are supported. To change the maximum number of connections to value 10, the user should execute the following procedure.

```
config
ssh-server max-connections 10
telnet-server max-connections 10
commit
```

2.7.4 Enabling Telnet Service

For security, Telnet server is deactivated. If the user would like to activate the Telnet service, it should execute the following procedure:

```
config
telnet-server enabled
commit
```

It is possible to change the Telnet server port, the configuration below changes the port on the Telnet server to 2323.

```
config
telnet-server port 2323
commit
```



After executing the commit to apply the port change configuration, if there is any open Telnet session it will be closed automatically, being necessary to access the equipment again using the new configured port.

2.8 Hostname Configuration

2.8.1 Configuring Hostname

Considering that the user would like to use the **DATAKOM-ROUTER-R1** name to identify the equipment. The procedure below shall indicate how to carry out this config:

```
config
hostname DATAKOM-ROUTER-R1
commit
```

2.9 Banner Configuration

The login banner is displayed after the login.

2.9.1 Configuring Banner in Single Line

It can be configured in a one line command, as shown below.

```
config
banner login "\nRestricted Access\n"
commit
```



The "\" is a scape character. To display a "\", it should be entered "\\"



The available commands for troubleshooting can be found in the topic [Verifying Banner](#).

2.9.2 Configuring Banner in Multiple Line

It is also possible to configure it in multiple lines.

```
config
banner login
(<Hit <cr> to enter in multi-line mode. Alternatively, enter a text between
double quotes. Remember to insert a line break at the end. See command
```

```
reference for examples. Maximum length of 3240 characters.>)
(\nRestricted Access\n):
[Multiline mode, exit with ctrl-D.]
>
> Restricted Access
> <CTRL-D>
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Banner](#).

2.9.3 Verifying Banner

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show banner login
```

2.10 System Clock and Date Configuration

2.10.1 Configuring System Clock

The config below adjusts the system's clock in a forced manner, this means, without any synchronization. The clock and date config is important to visualize logs and events in equipment.



It is recommended to use a centralized synchronization through the SNTP protocol.

Considering that the user would like to set the date to **January 20, 2017** and the time to **10 o'clock, 5 minutes and 30 seconds**. The procedure below shall indicate how to carry out this config:

```
set system clock 20170120 10:05:30
```

2.10.2 Configuring Timezone

Considering that the user would like to set the **timezone** to -3.

```
config
clock timezone BRA -3
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Clock](#).

2.10.3 Verifying Clock

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show system clock
```


3 Network Management

DmOS offers some tools and protocols for network management.

This chapter contains the following sections:

- LLDP Configuration
- SNTP Configuration
- Syslog Configuration
- SNMP Configuration
- Ping
- Traceroute
- SSH Client
- Telnet Client
- Tcpdump

3.1 LLDP Configuration

The Link Layer Discovery Protocol (LLDP) is used to announce interface and management information to directly connected neighbors.

3.1.1 Configuring LLDP between two Neighbors

The configuration below shows how to enable LLDP in interface gigabit-ethernet 1/1/1.

```
lldp
interface gigabit-ethernet-1/1/1
  admin-status tx-and-rx
  notification
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying LLDP](#).

3.1.2 Verifying LLDP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

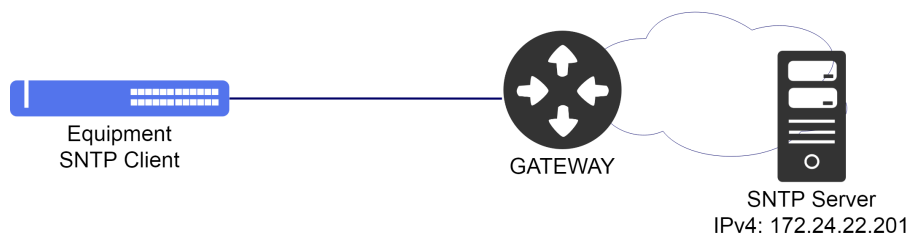
```
show lldp brief
show lldp statistics
show lldp local
debug enable proto-lldp
```

3.2 SNTP Configuration

The SNTP (Simple Network Time Protocol) is a simplified version of the NTP (Network Time Protocol) which is used to synchronize the system's clock with a server. This config is important to visualize logs and events in the equipment.

3.2.1 Configuring SNTP

The scenario below will be used to illustrate the config of the SNTP.



Example of SNTP config

Considering that the user would like to set the equipment as SNTP customer and use the SNTP server that has the **172.24.22.201** IPv4 address. The procedure below shall indicate how to carry out this config:

```
config
 sntp client
 sntp server 172.24.22.201
commit
```

It is possible to bind the SNTP client to a l3 interface to allow SNTP to work in user VRFs. The following configuration binds SNTP server to interface CUST-A-VLAN20, which is assigned to VRF cust-a. SNTP client requests will be sent in that VRF using the specified l3 interface IP address as source.

```
config
 sntp source interface l3-VRF-CUST-A
 sntp client
 sntp server 192.168.10.200
 sntp vrf cust-a
 !
 vrf cust-a
 !
 interface l3 CUST-A-VLAN20
```

```
vrf cust-a
lower-layer-if vlan 20
address 192.168.20.1/24
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying SNTP](#).

3.2.2 Configuring SNTP with Authentication

It is also possible to set the MD5 authentication with the SNTP server. The procedure below shall indicate how to proceed with this config:

```
config
sntp authenticate
sntp client
sntp authentication-key 1 md5 "SERVER-KEY"
sntp server 172.24.22.201 key 1
commit
```



The available commands for troubleshooting can be found in the topic [Verifying SNTP](#).

3.2.3 Verifying SNTP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

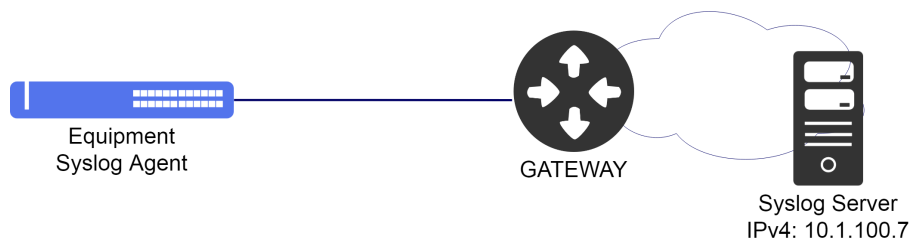
```
show sntp
```

3.3 Syslog Configuration

According to RFC5424, the Syslog protocol is used to transport event notification messages. The syslog is used by network devices to send event messages to an external server usually called Syslog Server. For example, if an Ethernet interface is deactivated, a message will be sent to the external server set to alert this change. This config is important to visualize the logs and events of the equipment in the network in a centralized manner.

3.3.1 Configuring Remote Syslog

The scenario below will be used to illustrate config of the Remote Syslog server.



Example of Remote Syslog config

Considering that the user would like to use a **remote syslog server** that has the IPv4 **10.1.100.7** address. The procedure below shall indicate how to perform this configuration:

```
config
log syslog 10.1.100.7
commit
```

It is possible to change the port of Syslog service, the configuration below changes the port on the service Syslog on server 10.1.100.7 from 514 (default port) to 9000.

```
config
log syslog 10.1.100.7 port 9000
commit
```

It is possible to change the source IP address of Syslog service, the configuration below changes the source IP address on the service Syslog on server 10.1.100.7 to IP address of loopback 1 interface.

```
config
log syslog 10.1.100.7 source interface loopback-1
commit
```

It is possible to use Syslog in VRFs. The following configuration binds Syslog server 10.1.100.7 to client VRF *vrf_client_a*.

```
config
log syslog 10.1.100.7 vrf vrf_client_a
commit
```

It is possible to change the source IP address of Syslog service in VRFs, the configuration below changes the source IP address on the service Syslog on server 10.1.100.7 to IP address of loopback 1 interface to client VRF *vrf_client_a*.

```
config
log syslog 10.1.100.7 vrf vrf_client_a source interface loopback-1
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Syslog](#).

3.3.2 Verifying Syslog

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show log
```

3.4 SNMP Configuration

SNMP is a protocol that helps the network administrators to manage the network devices and troubleshoot network problems. The network management system is based on two main elements: a manager and agents. The SNMP protocol has three versions:

Version	Description
SNMPv1	Original version of the SNMP, strings of communities sent in simple text with weak security.
SNMPv2c	Version developed to correct some problems of v1. However, several versions were developed, not truly approaching the problems with v1. The v2c version is the mostly used version and improved handling of protocols as compared to version v1, resulting in operations slightly enhanced. However, security is still a problem because it uses strings of community in simple text.
SNMPv3	Most recent version of the SNMP, supporting security and SHA authentication and full MD5. Should be used, if possible, specifically in non reliable networks.

By default, DmOS has in the factory configuration several commands necessary to use SNMP. The topics below will demonstrate how to enable the features required for some of the scenarios served with SNMP in DmOS.

3.4.1 Configuring SNMPv2

The scenario below will be used to demonstrate the SNMP settings.



Example of SNMP scenario

The configuration below shows how to enable the SNMPv2 agent to respond to requests. The **datacom** community will be used to request the agent.



The double quotes "" present in access command is the name of global VRF in DmOS, so even if user does not use VRF it is necessary use the double quotes to reference the global VRF.

```
config
snmp agent enabled
snmp agent version v2c
snmp community datacom
    sec-name datacom
!
snmp vacm group datacom
    member datacom
    sec-model [ v2c ]
!
access "" v2c no-auth-no-priv
    read-view root
    write-view root
!
snmp vacm view root
    subtree 1.3
    included
!
commit
```

After applying the above configuration, the agent will respond to SNMPv2 requests through the *community datacom*.

If necessary it is possible assign a VRF using SNMP context instead of global VRF. In this case the information returned is the same as global VRF, except in the part of BGP that will be specific of VRF named *vrf-client-datacom* assigned in context configuration.

```
config
snmp agent enabled
snmp agent version v2c
snmp agent context vrf-client-datacom
snmp community datacom
    sec-name datacom
    context-map vrf-client-datacom
!
snmp vacm group datacom
    member datacom
```

```

    sec-model [ v2c ]
    !
    access vrf-client-datacom v2c no-auth-no-priv
    read-view root
    write-view root
    !
    !
    snmp vacm view root
    subtree 1.3
    included
    !
    !
    commit

```

After applying the above configuration, the agent will respond to SNMPv2 requests through the *community datacom* in VRF *vrf-client-datacom*.



The available commands for troubleshooting can be found in the topic [Verifying SNMP](#).

3.4.2 Configuring SNMPv3

To enable the SNMPv3 agent to respond to requests securely via **user dmview** with authentication password and privacy password using the encryption mode below:

- SHA authentication with password **dmview123-sha**.
- AES password privacy **dmview123-aes**.



The double quotes "" present in access command is the name of global VRF in DmOS, so even if user does not use VRF it is necessary use the double quotes to reference the global VRF.

Proceed in the following manner:

```

config
snmp agent enabled
snmp agent version v3
!
snmp vacm group VACM-SNMPv3
member dmview
sec-model [ usm ]
!
access "" usm auth-priv
read-view root
write-view root
!
!
snmp vacm view root
subtree 1.3
included
!
!
snmp usm local user dmview
auth sha password "dmview123-sha"
priv aes password "dmview123-aes"
!
commit

```

If necessary it is possible assign a VRF using SNMP context instead of global VRF. In this case the information returned is the same as global VRF, except in the part of BGP that will be specific of VRF named *vrf-client-datacom* assigned in context

configuration.



To access objects using context in SNMPv3 it is necessary to specify the context in the tool or command used on servers.

```
config
snmp agent enabled
snmp agent version v3
snmp agent context vrf-client-datacom
!
snmp vacm group VACM-SNMPv3
member dmview
sec-model [ usm ]
!
access vrf-client-datacom usm auth-priv
read-view root
write-view root
!
snmp vacm view root
subtree 1.3
included
!
snmp usm local user dmview
auth sha password "dmview123-sha"
priv aes password "dmview123-aes"
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying SNMP](#).

3.4.3 Configuring the sending of SNMP notifications

It is possible to configure SNMP to send notification messages about events occurred in the system. These messages can be of two types:

- **Traps:** Messages are sent but do not receive confirmation.
- **Informs:** Messages are sent and receive confirmation.

Traps

For the equipment to send **traps** to server **172.22.1.152**, the configuration below can be used.



All traps are enabled by default and can be disabled as needed.

```
config
!
snmp notify std_v2_trap
tag std_v2_trap
```



```
type trap
!
snmp target SNMP-Trap-Server
ip 172.22.1.252
tag [ std_v2_trap ]
v2c sec-name public
!
commit
```

The SNMP target can also be assigned to a VRF.

```
config
snmp target SNMP-Trap-Server
vrf myvrf
ip 172.22.1.252
tag [ std_v2_trap ]
v2c sec-name public
!
commit
```

It is possible to change the source IP address of Traps in SNMP target, the configuration below changes the source IP address to loopback 1 IP address. This configuration can be also used in target associated to a VRF.

```
config
!
snmp target SNMP-Trap-Server
source interface loopback-1
!
commit
```

The user can disable some type of trap according to the example below.

```
config
!
no snmp traps login-success
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying SNMP](#).

Informs

For the equipment to send **informs** to SNMPv2 server, the configuration below can be used.

```
config
!
snmp notify std_v2_inform
tag std_v2_inform
type inform
!
snmp target SNMP-Notify-Server
ip 172.22.1.252
tag [ std_v2_inform ]
v2c sec-name public
!
commit
```

For the equipment to send **informs** to SNMPv3 server, the configuration below can be used.

```
config
!
snmp notify std_v3_inform
tag std_v3_inform
type inform
!
snmp target SNMP-Notify-Server
ip 172.22.1.252
tag [ std_v3_inform ]
engine-id 12:34:00:00:00:00:00:00:00:00:00
usm user-name dmview
usm sec-level auth-priv
!
snmp usm remote 12:34:00:00:00:00:00:00:00:00:00
user dmview
auth sha password "dmview123-sha"
priv aes password "dmview123-aes"
!
commit
```



In order for the SNMPv3 server to receive notifications, it is necessary to have the name of **user** and **engineID** configured in the server database. Otherwise the server will reject notification messages.

The SNMP target can also be assigned to a VRF.

```
config
snmp target SNMP-Trap-Server
vrf myvrf
ip 172.22.1.252
tag [ std_v3_inform ]
engine-id 12:34:00:00:00:00:00:00:00:00:00
usm user-name dmview
usm sec-level auth-priv
!
commit
```

It is possible to change the source IP address of Informs in SNMP target, the configuration below changes the source IP address to loopback 1 IP address. This configuration can be also used in target associated to a VRF.

```
config
!
snmp target SNMP-Trap-Server
source interface loopback-1
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying SNMP](#).

3.4.4 Configuring SNMP parameters

This topic will demonstrate how to configure or change parameters that are optional for the user, such as:

- SNMP agent version.
- SNMP agent IP address.
- SNMP agent interface.

- SNMP community.
- System information for *MIB-2 System*.
- Users.
- Access control.

Changing the agent version

In default SNMP configuration, the agent in versions v2 and v3 are already configured. If the user wants to change the agent configuration, removing the v3 agent, for example, the configuration below must be applied.

```
config
no snmp agent version v3
commit
```

From that moment on, SNMPv3 queries will no longer be answered by the agent.

Specifying the SNMP listening IP of SNMP agent



The SNMP agent IP default configuration is 0.0.0.0 which accepts SNMP packets from all IP address configured in the equipment. It is not recommended to change this configuration.

It is possible to specify which IPv4 or IPV6 address the SNMP agent will listening the SNMP packets by configuring **agent ip**. With this configuration will be accepted only SNMP packets to this IP.



The IP address used must be configured on the device.

```
config
snmp agent ip <IP>
commit
```

Specifying the SNMP agent interface

It is possible to specify on which L3 interfaces SNMP can receive requests through the **listen interface** configuration. In this way, it is possible to use SNMP on L3 interfaces of VRFs.

```
config
snmp agent listen interface <l3-myintf1>
snmp agent listen interface <l3-myintf2>
commit
```

Configuring community

In the standard SNMP configuration, the **public** community is already configured with permission from **read, write and notification** for SNMPv2. For security reasons it is recommended change the default community. If the user wishes to remove or add a new community, the following steps will demonstrate how to perform these operations.



The double quotes "" present in access command is the name of global VRF in DmOS, so even if user does not use VRF it is necessary use the double quotes to reference the global VRF.

Removing the default community **public**):

```
config
no snmp community public
no snmp vacm group public
commit
```

Configuring a community **datacom-ro** with read-only permission:

```
config
snmp community datacom-ro
sec-name datacom-ro
!
snmp vacm group datacom-ro
member datacom-ro
sec-model [ v2c ]
!
access "" v2c no-auth-no-priv
read-view root
!
commit
```

Configuring a community **datacom-rw** with read and write permission:

```
config
snmp community datacom-rw
sec-name datacom-rw
!
snmp vacm group datacom-rw
member datacom-rw
sec-model [ v2c ]
!
access "" v2c no-auth-no-priv
read-view root
write-view root
!
commit
```

Configuring system information

Some system information can be configured to be reported through the SNMP agent using the **MIB-2 System**. If the user wishes to include contact and location information, for example, the configuration below must be applied:

```
config
snmp system contact datacom@datacom.com.br
snmp system location Eldorado-RS
commit
```

From that moment on, the MIB-2 System **sysContact** and **sysLocation** objects will return the configured information.

Configuring users

When an SNMPv3 agent is used, it is necessary to configure the users who will respond to requests. To configure the user **datacom** with authentication and privacy, the setting below should apply:

```
config
snmp usm local user datacom
  auth sha password "datacom-sha"
  priv aes password "datacom-aes"
commit
```

Configuring access control

In the standard SNMP configuration, the access control group **snmp vacm view root** allows all objects (OIDs) from the local base to respond from branch 1.3 of the MIB. If the user wants to create a new access level to allow queries from branch 1.3.6.1.4.1 (private enterprises) for example, the configuration below must be applied:



The double quotes "" present in access command is the name of global VRF in DmOS, so even if user does not use VRF it is necessary use the double quotes to reference the global VRF.

```
config
snmp vacm view ENTERPRISE
  subtree 1.3.6.1.4.1
  included
commit
```

From that moment on, the ENTERPRISE view can be applied to the access group.

```
config
snmp vacm group datacom
  member datacom
  sec-model [ v2c ]
  !
  access "" v2c no-auth-no-priv
  read-view ENTERPRISE
  !
snmp vacm view ENTERPRISE
  subtree 1.3.6.1.4.1
  included
commit
```



The available commands for troubleshooting can be found in the topic [Verifying SNMP](#).

3.4.5 Verifying SNMP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show running-config snmp
```

3.5 Ping

The ping command is a common method to check the connectivity of the equipment with the other equipment or to test a specific protocol.



The user should use the keyword **do** before the command if it is in the config mode.

To execute a ping with **IPv4 address**, follow the procedure below:

```
ping 5.178.41.1
PING 5.178.41.1 (5.178.41.1) 56(84) bytes of data.
64 bytes from 5.178.41.1: icmp_seq=1 ttl=61 time=2.15 ms
64 bytes from 5.178.41.1: icmp_seq=2 ttl=61 time=2.06 ms
64 bytes from 5.178.41.1: icmp_seq=3 ttl=61 time=2.12 ms
64 bytes from 5.178.41.1: icmp_seq=4 ttl=61 time=2.27 ms
64 bytes from 5.178.41.1: icmp_seq=5 ttl=61 time=2.07 ms
--- 5.178.41.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 2.065/2.139/2.272/0.074 ms
```

To execute a ping with **IPv6 address**, follow the procedure below:

```
ping6 2002:c0a8:fe05::6
PING 2002:c0a8:fe05::6(2002:c0a8:fe05::6) 56 data bytes
64 bytes from 2002:c0a8:fe05::6: icmp_seq=1 ttl=62 time=7.94 ms
64 bytes from 2002:c0a8:fe05::6: icmp_seq=2 ttl=62 time=4.66 ms
64 bytes from 2002:c0a8:fe05::6: icmp_seq=3 ttl=62 time=5.05 ms
64 bytes from 2002:c0a8:fe05::6: icmp_seq=4 ttl=62 time=5.00 ms
64 bytes from 2002:c0a8:fe05::6: icmp_seq=5 ttl=62 time=6.84 ms
--- 2002:c0a8:fe05::6 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 4.664/5.903/7.948/1.274 ms
```

Using the **source** parameter, it is possible to specify the ICMP packets source IP address. It is also possible to specify a source interface.

To run ping in a VRF (IPv4 only), the **vrf** parameter can be used or an interface assigned to the VRF can be specified in the **source** parameter.

3.6 Traceroute

The traceroute command is a method to execute the network diagnosis informing the hop-by-hop connectivity through which the pack is passing until the final destination.



The user should use the keyword **do** before the command if it is in the config mode.

To execute a traceroute with **IPv4 address**, follow the procedure below:

```
traceroute 5.178.41.1
traceroute to 5.178.41.1 (5.178.41.1), 30 hops max, 38 byte packets
 1  192.168.48.3 (192.168.48.3)  2.029 ms  4.345 ms  1.751 ms
 2  192.168.48.1 (192.168.48.1)  2.842 ms  2.488 ms  3.226 ms
 3  192.168.254.22 (192.168.254.22)  3.582 ms  3.403 ms  3.622 ms
 4  192.168.84.22 (192.168.84.22)  2.306 ms  1.802 ms  2.264 ms
 5  5.178.41.1 (5.178.41.1)  2.219 ms  1.651 ms  54.376 ms
```

The source address used by the traceroute command can be specified by the **source** command.

The traceroute command can be execute in VRFs specifying the VRF through the **vrf** parameter.

To execute a traceroute with **IPv6 address**, follow the procedure below:

```
traceroute6 2002:c0a8:fe05::6
traceroute to 2002:c0a8:fe05::6 (2002:c0a8:fe05::6) from 1997::c0a8:3002,
30 hops max, 16 byte packets
 1  1997::c0a8:3001 (1997::c0a8:3001)  13.877 ms  2.298 ms  2.249 ms
 2  2001::c0a8:3001 (2001::c0a8:3001)  3.64 ms  2.969 ms  2.869 ms
 3  2002:c0a8:fe05::6 (2002:c0a8:fe05::6)  4.444 ms  3.624 ms  5.787 ms
```

3.7 SSH Client

It is possible to access other equipment using SSH protocol as from an equipment with DmOS.



The user should use the keyword **do** before the command if it is in the config mode.

To access an equipment with IPv4 **192.168.1.254** address through the **SSH**, the user should use the command below, specifying the user to be authenticated, in this example, the **admin** user:

```
ssh admin@192.168.1.254
```

It is also possible to access other equipment that is in a VRF using the SSH protocol.

To access an equipment with IPv4 **192.168.1.254** address through the **SSH**, the user should use the command below, specifying the user to be authenticated, in this example, the **admin** user and the VRF where the host is located:

```
ssh admin@192.168.1.254 vrf <VRF-NAME>
```

3.8 Telnet Client

It is possible to access other equipment using Telnet protocol as from an equipment with DmOS.



The user should use the keyword **do** before the command if it is in the config mode.

To access an equipment with IPv4 **192.168.1.254** address through the **TELNET** the user should use the command mentioned below:

```
telnet 192.168.1.254
```

3.9 Tcpcdump

It is possible to capture the control plane packets received and sent by equipment using the **tpcdump** feature from DmOS equipment. The capture output is text mode in CLI or pcap file that allows user to export this file to a remote server.



SSH and Telnet packets are not captured in this feature.



tcpdump in DmOS uses the same filters available in Linux tcpdump.

To capture received packets:

```
tcpdump rx
```

To capture sent packets:

```
tcpdump tx
```

To capture sent and received packets:

```
tcpdump rx-and-tx
```

To limit to 10 the number of captured packets:


```
tcpdump rx-and-tx count 10
```

3.9.1 Example of using filters

To capture sent and received packets in VLAN 10:

```
tcpdump rx-and-tx filter "vlan 10"
```

To capture sent and received packets of IP 192.168.0.1:

```
tcpdump rx-and-tx filter "vlan and host 192.168.0.1"
```

To capture sent packets to IP 192.168.0.1:

```
tcpdump rx-and-tx filter "vlan and dst 192.168.0.1"
```

To capture received packets from IP 192.168.0.1:

```
tcpdump rx-and-tx filter "vlan and src 192.168.0.1"
```

To capture ICMP packets received and sent:

```
tcpdump rx-and-tx filter "vlan and icmp"
```

To capture Syslog packets received and sent:

```
tcpdump rx-and-tx filter "vlan and port 514"
```

To capture Syslog or ICMP packets received and sent:

```
tcpdump rx-and-tx filter "vlan and (icmp or port 514)"
```

3.9.2 Generate and export pcap file

To generate capture file with packets received and sent:

```
tcpdump rx-and-tx save-pcap
```

To export capture file by TFTP protocol:

```
copy pcap tftp://<server_address>
```

4 OAM

This chapter shows a group of Operations, Administration and Maintenance (OAM) functionalities that provide indication of network failure, fault location, performance information, data functions and diagnostic. It contains the following sections:

- CFM Configuration
- EFM Configuration
- RDM Configuration
- TWAMP Configuration
- sFlow Configuration
- Traffic Loop Configuration
- Task scheduling configuration
- User Defined Counters Configuration

4.1 CFM Configuration

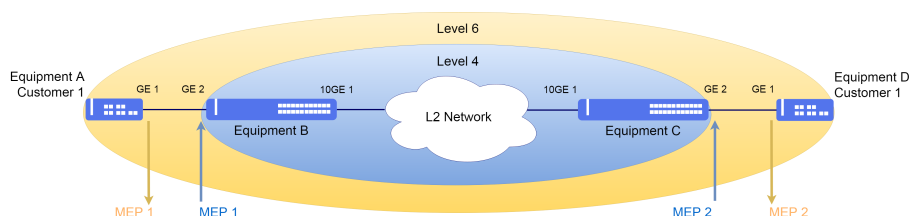
The CFM (Connectivity Fault Management) protocol is defined in the **IEEE 802.1ag** standard and provides an assurance of end-to-end full path, point-to-point or in a LAN made up by several equipment. In the CFM, network entities formed by network operators, service providers and end clients are part of different network domains managed by different individuals. In the CFM, the domains are the MD (Maintenance Domain) that have levels that in turn have one or more MAs (Maintenance Association) that are responsible for protection of a list of VLANs where the MEP will communicate. The MEPs (Maintenance End Point) are active entities responsible for sending of CFM PDUs.



DmOS does not support MIPs.

4.1.1 Configuring CFM

The scenario below will be used to illustrate the config of CFM between Customer and Service Provider.



Implementation of the CFM

Considering that the user would like to execute the following configs:

- **Equipment A:** VLAN 2000 for the CFM with a gigabit-ethernet-1/1/1 interface as MEP 1 – Down in level 6.
- **Equipment B:** VLAN 2000 for CFM with gigabit-ethernet-1/1/2 interface as MEP 1 – Up in level 4 and the ten-gigabit-ethernet-1/1/1 interface as Uplink of VLAN 2000.
- **Equipment C:** VLAN 2000 for CFM with gigabit-ethernet-1/1/2 interface as MEP 2 – Up in level 4 and the ten-gigabit-ethernet-1/1/1 interface as Uplink of VLAN 2000.
- **Equipment D:** VLAN 2000 for the CFM with a gigabit-ethernet-1/1/1 interface as MEP 2 – Down in level 6.
- All MEPs enable fault alarm notification for all errors.

```

!Equipment A
config
dot1q
vlan 2000
interface gigabit-ethernet-1/1/1 tagged
!
!
oam
cfm
md Client
level 6
ma Client
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 2
mep 1
interface gigabit-ethernet-1/1/1
direction down
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
commit

```

```

!Equipment B
config
dot1q
vlan 2000
interface ten-gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
!
!
oam
cfm
md ServiceProvider
level 4
ma ServiceProvider
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 2
mep 1
interface gigabit-ethernet-1/1/2
direction up
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
commit

```

```

!Equipment C
config
dot1q
vlan 2000
interface ten-gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
!
!
oam
cfm
md ServiceProvider
level 4
ma ServiceProvider
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s

```

```

remote-meps 1
mep 2
interface gigabit-ethernet-1/1/2
direction up
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
commit

```

```

!Equipment D
config
dot1q
vlan 2000
interface gigabit-ethernet-1/1/1 tagged
!
!
oam
cfm
md Client
level 6
ma Client
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 1
mep 2
interface gigabit-ethernet-1/1/1
direction down
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
commit

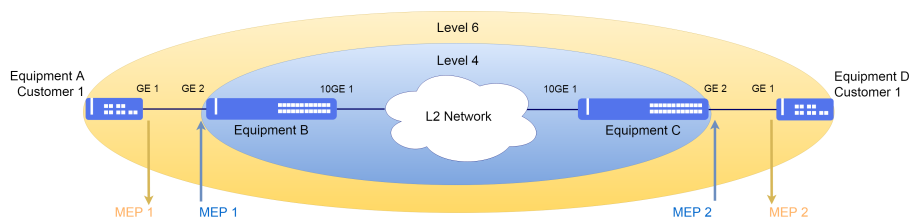
```



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

4.1.2 Configuring CFM with QinQ

The scenario below will be used to illustrate the config of CFM between Customer and Service Provider with QinQ to support some clients in Service VLAN.



Implementation of the CFM

Considering that the user would like to execute the following configs:

- **Equipment A:** VLAN 2000 for the CFM with a gigabit-ethernet-1/1/1 interface as MEP 1 – Down in level 6.
- **Equipment B:** QinQ in VLAN 3000 for CFM with gigabit-ethernet-1/1/2 interface as MEP 1 – Up in level 4 and the ten-gigabit-ethernet-1/1/1 interface as Uplink of VLAN 3000.

- **Equipment C:** QinQ in VLAN 3000 for CFM with gigabit-ethernet-1/1/2 interface as MEP 2 – Up in level 4 and the ten-gigabit-ethernet-1/1/1 interface as Uplink of VLAN 3000.
- **Equipment D:** VLAN 2000 for the CFM with a gigabit-ethernet-1/1/1 interface as MEP 2 – Down in level 6.
- All MEPs enable fault alarm notification for all errors.

```
!Equipment A
config
dot1q
vlan 2000
interface gigabit-ethernet-1/1/1 tagged
!
!
oam
cfm
md Client
level 6
ma Client
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 2
mep 1
interface gigabit-ethernet-1/1/1
direction down
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
commit
```

```
!Equipment B
config
dot1q
vlan 3000
interface ten-gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 untagged
!
!
switchport
interface gigabit-ethernet-1/1/2
native-vlan
vlan-id 3000
!
qinq
!
oam
cfm
md ServiceProvider
level 4
ma ServiceProvider
primary-vlan-id 3000
vlan-list 3000
ccm-interval 1s
remote-meps 2
mep 1
interface gigabit-ethernet-1/1/2
direction up
primary-vlan-id 3000
inner-vlan-id 2000
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
commit
```

```
!Equipment C
config
dot1q
vlan 3000
interface ten-gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 untagged
!
!
switchport
interface gigabit-ethernet-1/1/2
native-vlan
vlan-id 3000
!
qinq
```

```

!
oam
 cfm
  md ServiceProvider
    level 4
    ma ServiceProvider
      primary-vlan-id 3000
      vlan-list 3000
      ccm-interval 1s
      remote-meps 1
    mep 2
      interface gigabit-ethernet-1/1/2
      direction up
      primary-vlan-id 3000
      inner-vlan-id 2000
      continuity-check
      cci-enabled
      lowest-fault-priority-defect remote-rdi
commit

```

```

!Equipment D
config
dot1q
vlan 2000
 interface gigabit-ethernet-1/1/1 tagged
!
!
oam
 cfm
  md Client
    level 6
    ma Client
      primary-vlan-id 2000
      vlan-list 2000
      ccm-interval 1s
      remote-meps 1
    mep 2
      interface gigabit-ethernet-1/1/1
      direction down
      continuity-check
      cci-enabled
      lowest-fault-priority-defect remote-rdi
commit

```



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

4.1.3 Enabling Alarm Indication Signal (ETH-AIS)

The Ethernet Alarm Indication Signal ETH-AIS has been proposed by ITU-Y.1731 to avoid flagging the same fault repeatedly in a scenario with more than one domain when the internal domain fails.

For AIS messages to be issued, the configuration must be enabled in the MA. AIS messages will be sent to the domain with a level immediately above yours, that is, an external domain.

When transmission is enabled, AIS frames are transmitted when a fault is detected, regardless of any alarm configuration and report. When AIS alarm suppression is enabled, alarms are not reported if AIS frames are received.



AIS reception takes the alarm-suppression parameter only.

Below is an example configuration for transmitting and receiving AIS for a particular MA.

```
oam
 cfm
  md Client
  level 6
  ma Client
  primary-vlan-id 2000
  vlan-list 2000
  ccm-interval 1s
  ais
  transmission
  level 7
  interval 1min
  !
  reception
  alarm-suppression
  !
commit
```



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

4.1.4 Enabling Action Block

CFM supports interface blocking configuration with MEPs that are on failure. When the interfaces are in the blocking state, the layer 2 protocols (RSTP, EAPS, ERPS, LLDP) are signaled and changed to failure status. This feature helps the convergence of protocols. Scenarios in which the equipment are directly connected are supported.



The feature is only supported in MEP Down.

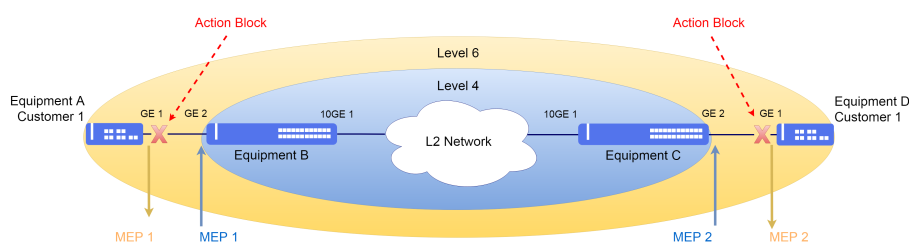


LACP, Link-flap, and loopback-detection are not triggered by CFM.



All MEPs should support action block so that the interface is not blocked in only one of the equipment.

The figure illustrates the point where the topology block occurs if the blocking action is configured the MEP interface.



Scenario with CFM action block

The configuration of equipment A and D with blocking action in MEP is shown below.

```
!Equipment A
config
dot1q
vlan 2000
interface gigabit-ethernet-1/1/1 tagged
!
!
oam
cfm
md Client
level 6
ma Client
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 2
mep 1
interface gigabit-ethernet-1/1/1
direction down
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
fault-action block-port
commit
```

```
!Equipment D
config
dot1q
vlan 2000
interface gigabit-ethernet-1/1/1 tagged
!
!
oam
cfm
md Client
level 6
ma Client
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 1
mep 2
interface gigabit-ethernet-1/1/1
direction down
continuity-check
cci-enabled
lowest-fault-priority-defect remote-rdi
fault-action block-port
commit
```



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

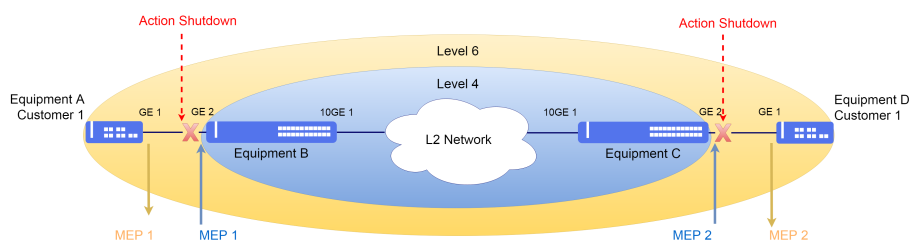
4.1.5 Enabling Action Shutdown

CFM supports the shutdown configuration of the interfaces when the MEPs on failure. When the interfaces are in the shutdown state, the other protocols configured on this interface are signaled by changing to failure status.



The feature is only supported in MEP Up.

The figure illustrates the point where shutdown occurs in the topology if the shutdown action is configured in the MEP interface.



Scenario with CFM action shutdown

The configuration of equipment B and C with shutdown action in MEP is shown below.

```
!Equipment B
config
dot1q
vlan 2000
interface ten-gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
!
!
oam
cfm
md ServiceProvider
level 4
ma ServiceProvider
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 2
mep 1
interface gigabit-ethernet-1/1/2
direction up
continuity-check
cci-enabled
lowest-fault-priority-defect remote-mac-error
fault-action shutdown-port
commit
```

```
!Equipment C
config
dot1q
vlan 2000
interface ten-gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
!
!
oam
cfm
md ServiceProvider
level 4
ma ServiceProvider
```

```
primary-vlan-id 2000
vlan-list 2000
ccm-interval 1s
remote-meps 1
mep 2
interface gigabit-ethernet-1/1/2
direction up
continuity-check
cci-enabled
lowest-fault-priority-defect remote-mac-error
fault-action shutdown-port
commit
```



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

4.1.6 Fault management



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

Continuity Check Protocol

The CFM Continuity Check Protocol is used to fault detection, notification and recovery. Below are the commands to check the communication exchanged between MEPS.

Loopback Protocol

The CFM Loopback Protocol is similar to Ping, but it is in Ethernet layer, being possible to check faults between MEPS.

```
oam cfm loopback md <md_name> ma <ma_name> mep <local_mep_id> remote-mep <remote_mep_id>
```

LinkTrace Protocol

The CFM LinkTrace Protocol is similar to Traceroute, but it is in Ethernet layer, being possible to discover the path and isolate faults.

```
oam cfm linktrace md <md_name> ma <ma_name> mep <local_mep_id> remote-mep <remote_mep_id>
```

4.1.7 Ethernet Delay Measurement (ETH-DM)

The CFM's Ethernet Delay Measurement (ETH-DM) allows measuring delay and jitter between two MEPS. These MEPS might monitor customer circuits and service provider transports networks, for instance. There are two types of delay-measurement: one-way and two-way, which are explained next.

One-way delay-measurement is based on unidirectional communication between two MEPs using Layer-2 frames. This delay is computed based on two timestamps values added by the originating MEP and the target MEP. For accurate statistics, precise time synchronization between both network devices is required. Please note that, in this scenario, statistics are consolidated by the target MEP, which is not convenient when the involved network devices are not under the same administration.

In turn, a two-way delay-measurement does not require any time synchronization among the involved devices. This is possible because each network device uses two timestamps, and the statistics are computed based on the difference between a pair of timestamps added by the same device. Considering an example with MEP_A and MEP_B, the network delay between them is computed as follows:

- MEP_A: sends a DMM frame, adding the first timestamp (TX_A)
- MEP_B: receives the DMM frame, adding the second timestamp (RX_B)
- MEP_B: creates a DMR frame, based on the received DMM, adding a third timestamp (TX_B)
- MEP_A: receives the DMR frame, adding the last timestamp (RX_A)

The network device that contains MEP_A computes the network delay as follows:

$$Delay = RX_A - TX_A - (TX_B - RX_B)$$



DmOS supports only the ETH-DM two-way mode.

To measure delay on demand use the below command:

```
oam cfm delay-measurement md <md_name> ma <ma_name> mep <local_mep_id> remote-mep <remote_mep_id>
```



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

Enabling Ethernet Delay Measurement probe

CFM supports the Ethernet Delay Measurement (ETH-DM) probe to periodically runs the delay measurement command storing the results.

Below shows the probe configuration based on premise that CFM is already configured in the equipment.

```
config
oam
 cfm
  delay-measurement probe 1
    md <md_name>
    ma <ma_name>
    mep <local_mep_id>
    remote-mep <remote_mep_id>
  session 1
```

```

      pcp 7
    !
  !
!
commit

```

The probes values can be collected through SNMP. To obtain the DmOS MIBs use the procedure described in chapter [Exporting the SNMP MIBs](#).



The available commands for troubleshooting can be found in the topic [Verifying CFM](#).

4.1.8 Verifying CFM

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



If some Ethernet interface is blocked by CFM, the **CFM** acronym will appear in the field **Blocked by** of **show interface link** command.



For more details about commands output, check the **Command Reference**.

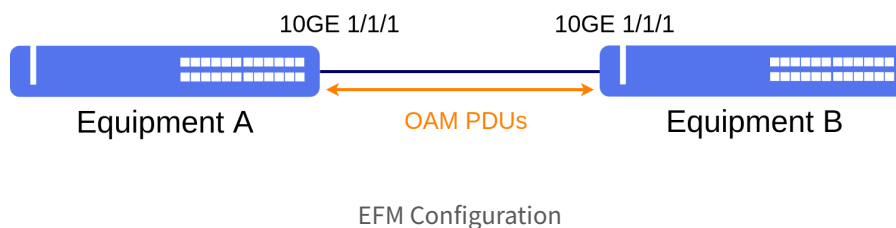
```
show interface link
show oam cfm
show oam cfm brief
show oam cfm detail
show oam cfm local
show oam cfm remote
show oam cfm statistics
show oam cfm delay-measurement
show oam cfm delay-measurement detail
show oam cfm delay-measurement probe <probe_id>
show oam cfm delay-measurement probe <probe_id> detail
show oam cfm linktrace
show oam cfm linktrace md <md_name> ma <ma_name> mep <local_mep_id>
show alarm
debug enable cfm-ais-rx
debug enable cfm-ais-tx
debug enable cfm-discard
debug enable cfm-dm
debug enable cfm-loopback
debug enable cfm-linktrace
```

4.2 EFM Configuration

EFM (Ethernet in the First Mile) is an OAM (Operations, Administration and Maintenance) protocol defined in the **IEEE 802.3ah** standard for link monitoring, blocking the interface when communication is lost.

4.2.1 Enabling EFM

The scenario below will be used to illustrate the config of EFM link monitoring between equipment A and equipment B.



```
config
oam
 efm
  interface ten-gigabit-ethernet-1/1/1
  !
!
commit
```



The available commands for Troubleshooting can be found in the topic [Verifying EFM](#).

4.2.2 Verifying EFM

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



If some Ethernet interface is blocked by EFM, the **EFM** acronym will appear in the field **Blocked by** of **show interface link** command.



For more details about commands output, check the **Command Reference**.

```
show interface link
show oam efm
show oam efm interface <interface>
debug enable proto-efm
```

4.3 RDM Configuration

The RDM (Remote Devices Management) protocol is a Datacom proprietary protocol. The purpose of this feature is to provide a means of managing remote equipment from the DmOS line. In the remainder of this chapter, the DM4000 line

equipment will be referenced generically by intermediate equipment, or simply **IE**.

The RDM solution architecture has the following logical components:

- **Master/slave devices:** Remote management is possible only when one of the devices is a master and the other is a slave. The IE is the master device, the remote is the slave device.
- **Communication between IE and remote:** IE has an exclusive VLAN for remote management. This VLAN includes only those ports where a remote is connected; adding ports to this VLAN can be done only dynamically, as remotes are detected. This Remote Management VLAN has an IP of the A.B.255.254/16 network, with A and B configurable in the CLI by the user. Each remote device has an IP of this same network in a VLAN (the remote connected to port P of unit U would have the IP A.B.U.P). In this way, there is an IP communication channel between the IE and the remote. It is worth remembering that the addresses of the A.B.0.0/16 network are not visible externally to the IE, since the remote's VLAN has members only ports on which there are remote ones connected.

Remote management is always done by the master's IP, that is, access to the remote U/P port is done by the master's management IP using an alternative port, which can be consulted in the master's CLI.

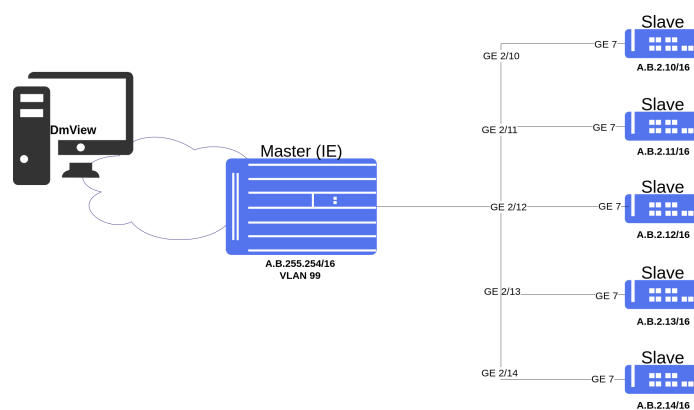


DmOS is not supported to act as a master device in this release.

4.3.1 Configuring RDM as a slave

In the DmOS factory default, RDM is enabled on all ports on devices that support the protocol. If the device is not configured to the factory default configuration, it may be necessary to enable RDM on the interface connected to the master (IE).

The scenario below considers that the IE is already configured to operate as a master RDM.



RDM Scenario

Suppose the user wants to enable RDM to act as a remote device to be managed by a central device on the network. The following procedure will show how to perform this configuration:



The EFM protocol must be enabled for RDM to work.

```
config
remote-devices interface gigabit-ethernet-1/1/1
oam efm interface gigabit-ethernet-1/1/1 mode passive
commit
```

After enabling the above settings, the device will be automatically configured, creating the VLAN for management and a static route to the master device.

From that moment on, the slave device can be accessed via the master device or through one of the services configured on the master such as Telnet, SSH or Netconf, for example.

To use the SSH (22) and Netconf (830) protocols on the slave, it is necessary to enable the services on the master device.



Contact DATACOM Technical Support to check the documentation available for configuring the master device.



The available commands for troubleshooting can be found in the topic [Verifying RDM](#).

4.3.2 Verifying RDM

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



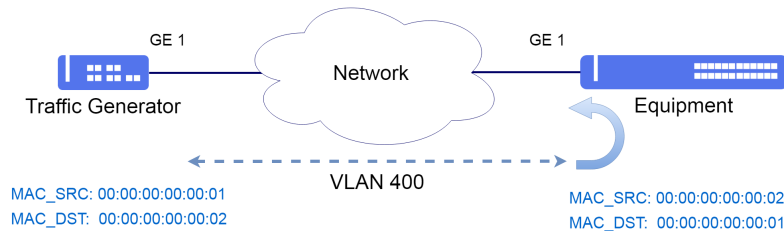
For more details about commands output, check the **Command Reference**.

```
debug enable remote-devices
show log component rdm_proto
show remote-devices
show running-config dot1q
show running-config router static
```

4.4 Traffic Loop Configuration

4.4.1 Configuring Traffic Loop for L2 Traffic Validation

The DmOS lets you loop L2 flows to meet RFC 2544 tests or other traffic testing to validate circuit delivery to the client. The following is an example of Feature Configuration.



Traffic loop scenario

Suppose the user wants to loop traffic using the VLAN 400 with gigabit-ethernet-1/1/1 interface as uplink interface. The configured MAC addresses must respect the configured data stream on the generator.



To avoid the risk of loss of access to equipment management, it is recommended to use the Traffic Loop functionality in the exclusive management mode.

```
config exclusive
dot1q
vlan 400
interface gigabit-ethernet-1/1/1
!
traffic-loop 1
interface gigabit-ethernet-1/1/1
source-mac-address 00:00:00:00:00:01
destination-mac-address 00:00:00:00:00:02
vlan 400
!
```

The user must use the **commit confirmed** command to save and apply the configuration. In the example below the commit will temporarily apply the configuration for 10 minutes. The user can change the commit time if necessary.

```
commit confirmed 10
```



There are no troubleshooting commands for this functionality.

4.5 TWAMP Configuration

The Two-Way Active Measurement Protocol (TWAMP) measures network performance parameters such as latency, latency variation (jitter), and packet loss. The implementation of the TWAMP server is based on the specifications described in RFC

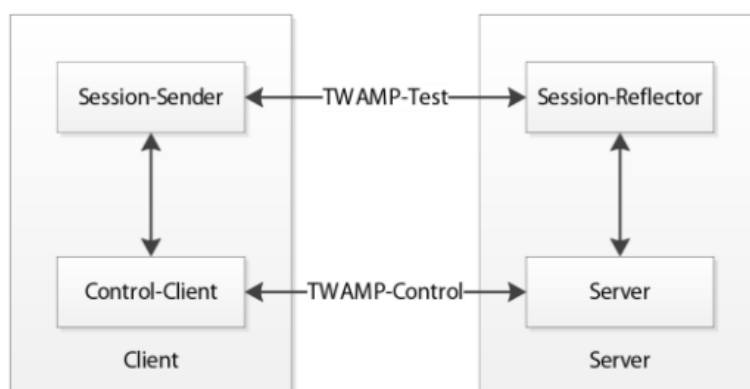
5357.

The TWAMP server solution architecture defines the following logical components:

- **Session Reflector:** Adds information to the received test packets and sends them back.
- **Server:** Manages multiple TWAMP sessions.

The TWAMP client solution architecture defines the following logical components:

- **Session Sender:** Creates and sends TWAMP test packets to the Session Reflector.
- **Control Client:** Sends requests to the TWAMP server to establish new sessions.



TWAMP Architecture



Authentication/Encryption is not supported



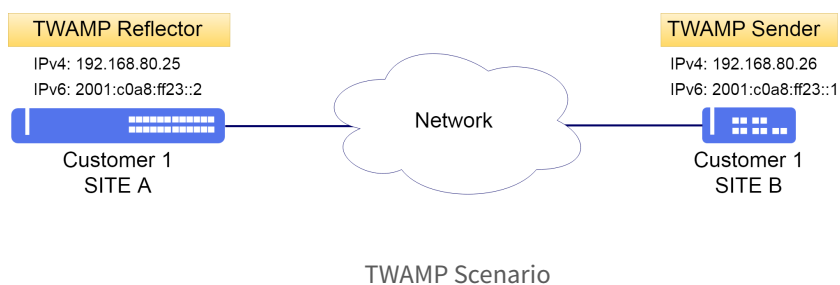
TWAMP listens to the port 862 by default, but that port can be changed.



The test-session supports ID from 1 to 65535.

4.5.1 Configuring a TWAMP session

The scenario below will be used to demonstrate the TWAMP configuration.



Suppose the user wants to configure a TWAMP session through of port 4000 to monitoring IPv4 and IPv6 services.

- **Reflector:** IPv4: 192.168.80.25 and IPv6: target-address 2001:c0a8:ff23::2
- **Sender:** IPv4: 192.168.80.26 and IPv6: target-address 2001:c0a8:ff23::1

The following procedure will show you how to perform this configuration:

```
!Equipment REFLECTOR - SITE A
config
 oam twamp reflector port 4000
commit
```

```
!Equipment SENDER- SITE B
config
 oam
 twamp
  sender connection 1
   server-port 4000
   description TWAMP-IPv4
   ipv4 source-address 192.168.80.26
   !
   ipv4 target-address 192.168.80.25
   !
  test-session 1
   description SITE B-SITE A
   ipv4 source-address 192.168.80.26
   !
   ipv4 target-address 192.168.80.25
   !
  !
  sender connection 2
   server-port 4000
   description TWAMP-IPv6
   ipv6 source-address 2001:c0a8:ff23::1
   !
   ipv6 target-address 2001:c0a8:ff23::2
   !
  test-session 2
   description SITE B-SITE A
   ipv6 source-address 2001:c0a8:ff23::1
   !
   ipv6 target-address 2001:c0a8:ff23::2
   !
 commit
```



The available commands for troubleshooting can be found in the topic [Verifying TWAMP](#).

4.5.2 Configuring ACLs in TWAMP Reflector

It is possible to restrict which clients can communicate with the reflector. The configuration below restricts access to network 10.1.15.0/24, IPv4 address 192.168.80.26 and IPv6 address 2001:c0a8:ff23::1 only.

If the user does not specify an IP address on TWAMP Reflector, all addresses will be accepted.

```
config
oam
twamp
  reflector
    ipv4
      client-address 192.168.80.26
      !
      client-network 10.1.15.0/24
      !
    ipv6
      client-address 2001:c0a8:ff23::1
      !
  commit
```



The available commands for troubleshooting can be found in the topic [Verifying TWAMP](#).

4.5.3 Configuring TWAMP in VRF

It is also possible to configure TWAMP (Reflector/Sender) on VRF.

Below the configuration of TWAMP Reflector in VRF TWAMP. For VRF configuration, see topic [VRF Configuration](#).

```
config
oam
twamp
  reflector
    vrf TWAMP
  commit
```



DmOS supports only one instance of TWAMP Reflector on device. TWAMP Reflector does not support configuring more than one instance, even on separate VRFs.

Below the configuration of TWAMP Sender in VRF TWAMP. For VRF configuration, see topic [VRF Configuration](#).

```
config
oam
twamp
  sender connection 1
    vrf TWAMP
```

4.5.4 Calculating the maximum number of sessions in TWAMP Reflector

TWAMP Reflector supports a maximum number of *simultaneous* test sessions. Test sessions above this limit are rejected and the TWAMP Sender will perform retries until the session is established.

To determine the maximum number of *non simultaneous* sessions supported by the reflector, the following formula can be used: *simultaneous test-sessions x (interval between tests / test duration)*.

As example, for a maximum number of simultaneous sessions of **8**, using the value of test duration of 20s and the default value of interval between tests of 300s, the maximum theoretical number of non simultaneous test sessions is $8 \times (300/20) = 120$.



The Product Datasheet should be consulted to obtain the values of maximum simultaneous sessions for each platform.

4.5.5 Verifying TWAMP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show oam twamp reflector
show oam twamp reflector connection brief
show oam twamp reflector connection detail
show oam twamp reflector test-session brief
show oam twamp reflector test-session detail
show oam twamp sender connection all brief
show oam twamp sender connection all test-session
show oam twamp sender connection all test-session-statistics
debug enable proto-twamp
```

4.6 sFLOW Configuration

The sFlow is a technology for monitoring data that travels on the network. A great advantage of sFlow is forward not all collected traffic to the sFlow collector, instead sFlow only forwards traffic samples to the collector at a configurable rate, reducing the computational load.

For sFlow operation, two components are required:

- **sFlow Agent:** Function assigned to switches, routers, access points that samples transmitted and/or received packets and forward them to a sFlow collector.
- **sFlow Collector:** Function assigned to analyze the information received from each sFlow Agent.

Having as sampling technique:

- **Flow Sampling:** Based on the packet sample, used to obtain information from the package contents such as protocols and etc.
- **Counter sampling:** Based on the time sample, used to obtain statistics of interfaces.



Only Flow Sampling is supported.



Only one sFlow collector is allowed.



In the egress traffic only unicast packages will be sampled.

4.6.1 Configuring sFLOW

Suppose the user wishes to monitor the data streams forwarded through the interface gigabit-ethernet-1/1/10.

Below is an example of how to configure sFlow agent on the 1/1/10 gigabit-ethernet interface by sending samples to COLLECTOR-1 on port 1555. By default the collector port is 6343.



In the egress traffic only unicast packages will be sampled.



Only one sFlow collector is allowed.

```
config
oam
sflow
  collector COLLECTOR-1
  ipv4 172.22.107.14
  port 1555
!
interface gigabit-ethernet-1/1/10
  flow-sampling-collector COLLECTOR-1
!
commit
```



There are no troubleshooting commands for this functionality.

4.7 Task scheduling configuration

It is possible to schedule task execution using the **assistant-task** feature.

The first step is to create a file with the commands to be executed. To create a command file, consult the topic **File Edit** in chapter [Files Management](#).

It is possible to create the file on another equipment and later import to run.

The file must be copied to the device via TFTP or SCP. See topic **Importing the Files** in the chapter [Files Management](#).



DmOS only supports ASCII format files and use Unix format (LF).



If script was created in Windows which use CRLF format, even DmOS converts to Unix format when import the file, the last command can not run. As workaround it is possible adding ! character in the end of file.

4.7.1 Automatic reboot configuration

In the example below, an automatic reboot has been schedule to be executed on **30-Sep-2019 as 02:00AM**.

A file **reboot.cli** has been created with the commands below. Content can be viewed with the command **file show**.

```
reboot
```

Finally, the reboot can be scheduled. This task will be executed only once, so the parameter **once** was inserted.

```
config
 assistant-task reboot
  action cli-file reboot.cli
  schedule once day 30 month 9 hour 2 minute 0
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Assistant Task](#).

4.7.2 Automatic backup configuration

Another possible use of assistant-task is configuration backup.

In the example below, there is a script that saves the current configuration to file **config.txt** and sends it to TFTP server **192.168.0.1**. Below the contents of the **config-backup.cli** script.

```
show running-config | save overwrite config.txt
copy file config.txt tftp://192.168.0.1/
```



The **overwrite** parameter was used to save the file. In case the file already exists, it will be automatically overwritten without confirmation.

The script execution is scheduled for every day at 06:00AM

```
config
assistant-task config-backup
  action cli-file config-backup.cli
  schedule recursive hour 6
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Assistant Task](#).

4.7.3 Executing a task manually

To execute the task manually once, the following command can be used.

```
assistant-task config-backup run-now
```



The available commands for troubleshooting can be found in the topic [Verifying Assistant Task](#).

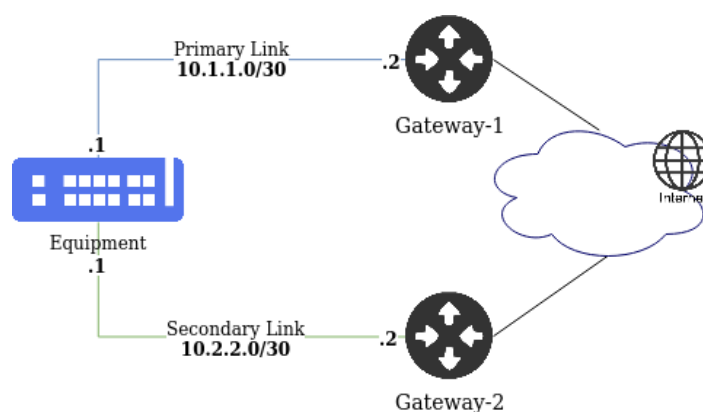
4.7.4 Running a task from a pattern

The assistant-task allows to perform actions based on the output of a command. To use this feature must be used the **watch** parameter.

Below is an example of a task to be run every minute (no date/time specified), checking link connectivity. When connectivity fails on the primary link, it changes the default route to the secondary link and vice versa. For this task, the script **connectGw1.cli** will be used to enable the primary link and **connectGw2.cli** script to enable the secondary link.



Do not use "|" repeat" or other commands that have no return, otherwise the assistant task will not be able to log its output and it will run indefinitely, until it is interrupted by the "logout" command.



Scenario Assistant Task with watch.

Below are the contents of the **connectGw1.cli** script.

```
config
no router static address-family ipv4 0.0.0.0/0 next-hop 10.2.2.2
router static address-family ipv4 0.0.0.0/0 next-hop 10.1.1.2
commit
```

Below are the contents of the **connectGw2.cli** script.

```
config
no router static address-family ipv4 0.0.0.0/0 next-hop 10.1.1.2
router static address-family ipv4 0.0.0.0/0 next-hop 10.2.2.2
commit
```

Creating a action to change the default route gateway when connectivity fails on the primary link. Note that the **regex** parameter sets a pattern to match the return of the command set in **watch cli-cmd**.

All **watch match** rules is checked against **watch cli-cmd** and all rules that match is run.

```
config
assistant-task ChangeGW
schedule recursive hour *
schedule recursive minute 0-59
action watch cli-cmd "ping 10.1.1.2 | include loss"
action watch match M0
cli-file connectGw2.cli
regex "100%\% packet loss"
!
action watch match M1
cli-file connectGw1.cli
regex "received, 0%\% packet loss"
commit
```



It is necessary use the escape character "\" before special characters for regex to work properly.

4.7.5 Verifying Assistant Task

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show assistant-task
show assistant-task <task-name> last-success output
show assistant-task <task-name> last-failure output
```

4.8 User Defined Counters Configuration

User defined counters can be used to measure VLAN or interface traffic.

4.8.1 VLAN counters configuration

In the example below, a counter to configure the VLAN 13 ingress is configured.

```
counters
  ingress id 1
    description "VLAN 13 ingress counter"
    type octets
    vlan 13
  !
!
```

For egress traffic, it is configured similarly.

```
counters
  egress id 1
    description "VLAN 13 egress counter"
    type octets
    vlan 13
  !
!
```



When specifying a VLAN, untagged interfaces will not have their traffic accounted.



The available commands for troubleshooting can be found in the topic [Verifying Counters](#).

4.8.2 Interface counters configuration

In the previous example, packets from all interfaces assigned to VLAN 13 are accounted. To limit the counter to a specific interface, it can be specified as following.

```
counters
  egress id 1
  description "VLAN 13 egress counter"
  type octets
  vlan 13
  interface ten-gigabit-ethernet-1/1/1
!
```

Counters can be used to measure traffic of two or more interfaces simultaneously. In the following configuration, a counter measures egress traffic of both interfaces TenGigabitEthernet 1/1/1 and 1/1/2.

```
counters
  egress id 1
  description "Interface egress counter"
  type octets
  interface ten-gigabit-ethernet-1/1/1 ten-gigabit-ethernet-1/1/2
!
```

The counters values can be collected through SNMP. To obtain the DmOS MIBs use the procedure described in chapter [Exporting the SNMP MIBs](#).



The available commands for troubleshooting can be found in the topic [Verifying Counters](#).

4.8.3 Verifying Counters

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show counters
show counters ingress
show counters ingress id <id>
show counters egress
show counters egress id <id>
```

5 Users Authentication

DmOS uses levels of privileges to define which information is available to user account in the equipment. Three levels of management access of users are supported: admin, config and audit.

Level	Description
admin	Allows exhibition and change of all the device parameters. It is a full access of reading and writing for the entire device.
config	Allows some functions more than only reading, however, less than the admin level. Allows the user to visualize all the device parameters. Allows all the config commands, except those for device administration purposes, such as: hostname, SNMP, monitor, RADIUS, SNTP, TACACS+ and Local users.
audit	Allows only reading functions.

Only one user account is set by default in the DmOS. The user is the **admin** with **admin** password and has admin level of privileges.



For security purposes, it is highly recommended to change the equipment standard password.



It is recommended to set the protocol passwords always between double quotations marks "password". Thus, it is possible to set passwords with no problems related to use of special characters.

To change the admin user standard password, follow the steps below:

```
config
aaa user admin password "new-password"
commit
```

This chapter contains the following sections:

- [Local Users Configuration](#)
- [TACACS+ Configuration](#)
- [RADIUS Configuration](#)
- [Authentication Order Configuration](#)

5.1 Local Users Configuration

5.1.1 Creating a new Local User

The next steps shall indicate how to set a new user called “**joao**” with “**joao1234**” password and with “**admin**” administrator privileges.

```
config
aaa user joao password "joao1234"
group admin
commit
```

5.1.2 Deleting a Local User

The next steps will indicate how to delete the user “joao”.

```
config
no aaa user joao
commit
```



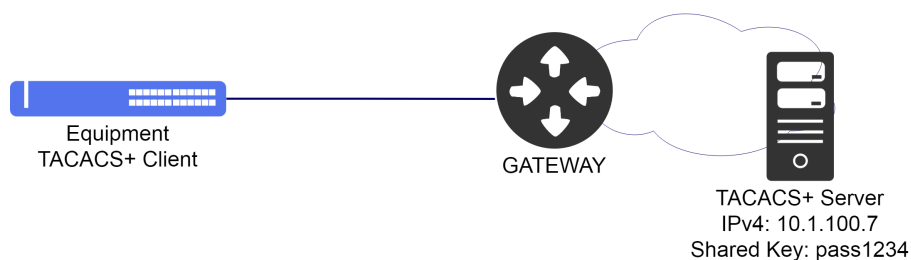
There are no troubleshooting commands for this functionality.

5.2 TACACS+ Configuration

The TACACS+ (Terminal Access Controller Access-Control System) is a protocol based on model AAA that provides the authentication, authorization and accounting services in a safe manner with encryption of the entire packet. This encryption depends on a shared secret key configured in the equipment.

5.2.1 Configuring a TACACS+ Server

The scenario below will be used to illustrate config of the TACACS+.



Example of TACACS+



For the accounting service to work it is mandatory that the user be authenticated also through the TACACS+ server.

The configuration below shows how to configure TACACS+ using a server with IPv4 address **10.1.100.7** and authentication password equal to **“pass1234”**. The procedure indicates how to enable authentication, authorization and accounting:

```
config
aaa server tacacs TACACS-SERVER host 10.1.100.7
shared-secret "pass1234"
authentication
authorization
accounting
commit
```

It is possible to specify a interface in the TACACS+ configuration, which will be used as the source address for the packets generated by the TACACS+ client. The specified interface can be in a VRF.

```
config
aaa server tacacs TACACS-SERVER host 10.1.100.7
shared-secret "pass1234"
authentication
authorization
accounting
source interface l3-myintf
commit
```

It is possible to specify a loopback interface in the TACACS+ configuration, which will be used as the source address for the packets generated by the TACACS+ client.



For the *source interface* parameter to be functional, it is necessary to set the **vrf** parameter in the TACACS+ configuration.

```
config
interface loopback 7
vrf VRF_CLI1
description Services-MGMT
ipv4 address 10.1.200.254/32
!
aaa server tacacs TACACS-SERVER host 10.1.100.7
shared-secret "pass1234"
authentication
authorization
accounting
source interface loopback-7
vrf VRF_CLI1
commit
```

It is possible to change the *authentication-type* to ASCII in the TACACS+ configuration, the default configuration is PAP. This configuration is valid for all TACACS+ server configured.

```
config
aaa authentication-type tacacs ascii
commit
```



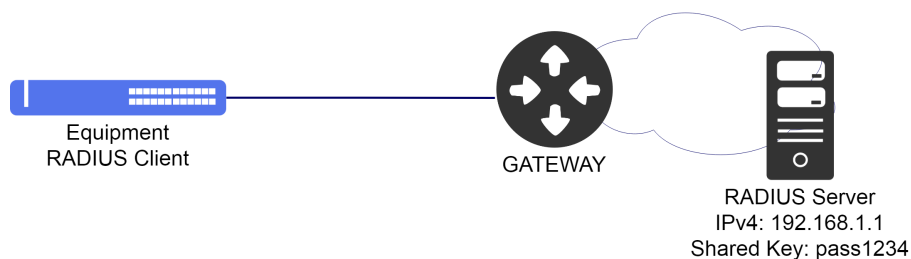
There are no troubleshooting commands for this functionality.

5.3 RADIUS Configuration

The RADIUS (Remote Authentication Dial In User Service) is a protocol based on model AAA that provides the authentication, authorization and accounting services. Communication between the RADIUS customer and the RADIUS server is safe and an exclusive key-word in both systems is required.

5.3.1 Configuring a RADIUS Server

The scenario below will be used to illustrate config of the RADIUS.



Example of RADIUS

Considering that a user would like to set a **RADIUS** server that has the IPv4 **192.168.1.1** address and authentication password equal to “**pass1234**”. The procedure below shall indicate how to execute this config enabling the authentication, authorization and accounting:

```
config
aaa server radius RADIUS-SERVER host 192.168.1.1
  shared-secret "pass1234"
  authentication
  accounting
commit
```

It is possible to specify a interface in the RADIUS configuration, which will be used as the source address for the packets generated by the RADIUS client. The specified interface can be in a VRF.

```
config
aaa server radius RADIUS-SERVER host 192.168.1.1
  shared-secret "pass1234"
  authentication
  accounting
  source interface l3-myintf
commit
```



There are no troubleshooting commands for this functionality.

5.4 Authentication Order Configuration

The user may define the authentication-order between: **local**, **RADIUS** and **TACACS+**. When a user tries to login in the system, the DmOS will try to authenticate it following the order defined by the CLI command “**authentication-order**”.

5.4.1 Configuring RADIUS with higher priority

Considering that a user has configured a RADIUS server to be used as an authentication method and would like to use it as a preferential method, however it would like to use the authentication in the local base in case of communication failure with the RADIUS server. The procedure to execute this config is indicated below:

```
config
aaa authentication order [radius local]
commit
```

5.4.2 Configuring TACACS+ with higher priority

Considering that a user has configured a TACACS+ server to be used as an authentication method and would like to use it as a preferential method. The procedure to execute this config is indicated below:

```
config
aaa authentication order [tacacs local]
commit
```

6 Interfaces

This chapter will provide examples of how to configure Ethernet interfaces. For GPON interfaces, please check GPON chapter.

This chapter contains the following sections:

- Ethernet Interfaces Configuration
- Link Aggregation Configuration
- Backup-Link Configuration
- Port Mirroring Configuration
- Link Flap Detection Configuration
- Hold Time Configuration
- DWDM Interface Configuration

6.1 Ethernet Interfaces Configuration

6.1.1 Configuring Ethernet Interfaces

To configure an Ethernet interface, the user should enter in the interface config level.

To configure a 1G interface located in Chassis 1, Slot 1 and Port 1 (1/1/1), the user should use the following command:

```
config
interface gigabit-ethernet 1/1/1
```

To configure a 10G interface located in Chassis 1, Slot 1 and Port 1 (1/1/1), the user should use the following command:

```
config
interface ten-gigabit-ethernet 1/1/1
```

To configure a 40G interface located in Chassis 1, Slot 1 and Port 1 (1/1/1), the user should use the following command:

```
config
interface forty-gigabit-ethernet 1/1/1
```

To configure a 100G interface located in Chassis 1, Slot 1 and Port 1 (1/1/1), the user should use the following command:

```
config
interface hundred-gigabit-ethernet 1/1/1
```



Numbering scheme of the port of the chassis/slot/port was designed for standardization with equipment of several slots and chassis. Thus, it is always necessary to enter the full location, even if the equipment has no several slots or chassis.

To administratively disable a 1G interface, the user should use the following procedure. The same procedure is used if the user would like to disable interfaces of other capacities, such as 10G, 40G or 100G.

```
config
interface gigabit-ethernet 1/1/1
shutdown
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Ethernet Interfaces](#).

To reactivate an 1G interface, the user should use the command “no shutdown”. The same procedure is used if the user would like to reactivate interfaces of other capacities, such as 10G or 40G.

```
config
interface gigabit-ethernet 1/1/1
no shutdown
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Ethernet Interfaces](#).

6.1.2 Configuring Ethernet Interfaces Range

It is possible to configure several interfaces at the same time using the range of interfaces. The procedure below gives an example of how to deactivate the gigabit-ethernet 1/1/1, 1/1/2, 1/1/3 and 1/1/4 interfaces using the range.

```
config
interface gigabit-ethernet 1/1/1-4
shutdown
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Ethernet Interfaces](#).

6.1.3 Configuring Ethernet Interfaces Description

The user can configure descriptions in Ethernet interfaces, as shown below. To display all descriptions, the command `show interface link` can be used.

```
config
interface gigabit-ethernet 1/1/1
description "Link to switch 2"
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Ethernet Interfaces](#).

6.1.4 Configuring Ethernet Interfaces MTU

It is possible to change an Ethernet interface MTU using the configuration shown below.

```
config
interface gigabit-ethernet 1/1/1
mtu 1500
commit
```



The default MTU value is different for each platform. Refer to the **DmOS Datasheet** to check the maximum values for each hardware platform.



The interface configured MTU value is not used by protocols.



The available commands for troubleshooting can be found in the topic [Verifying Ethernet Interfaces](#).

6.1.5 Configuring Ethernet Interfaces TPID

The interface TPID can be changed using the configuration shown below.

```
config
switchport
interface gigabit-ethernet 1/1/1
tpid <tpid>
commit
```



The default TPID value is 0x8100.

The allowed TPID values are:

- **0x88a8**: TPID for 802.1ad bridges
- **0x8100**: default TPID for the 802.1Q VLANs

- **0x9100:** Alternative TPID



PDU's originated by protocols like EAPS, ERPS and CFM are sent with the TPID value configured in the interface.



When tagged frames are received with TPID value different from the configured in the interface, the frames are forwarded using native VLAN tag.



There are no troubleshooting commands for this functionality.

6.1.6 Configuring a 10Gbps Interface to operate in 1Gbps

The DmOS allows the use of 1G optical modules in 10G interfaces in two ways:

- **Forced Mode or Non-Negotiated Mode**
- **Negotiated Mode**



The auto-negotiation configuration is disabled by default.



DmOS does not support SFP+ operating at 1G.

To use a 10G interface operating in forced 1G (non-negotiated mode), it is necessary to make the following configurations:

```
config
interface ten-gigabit-ethernet 1/1/1
speed 1G
no negotiation
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Ethernet Interfaces](#).

To use a 10G interface operating in 1G in the negotiated mode, it is necessary to make the following configurations:

```
config
interface ten-gigabit-ethernet 1/1/1
  advertising-abilities lgfull
  negotiation
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Ethernet Interfaces](#).

6.1.7 Verifying Ethernet Interfaces

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show interface <interface-type> <chassis/slot/port>
show interface link
```

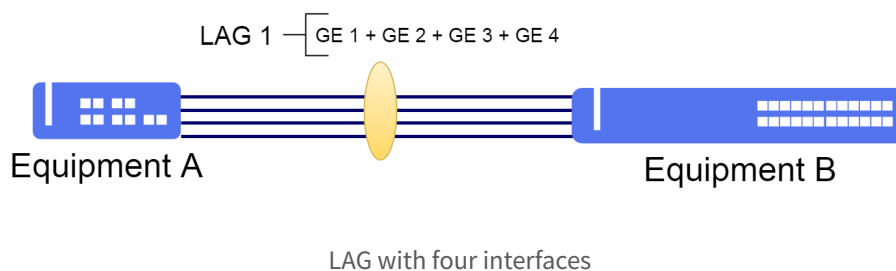
6.2 Link Aggregation Configuration

6.2.1 Configuring a LAG in static mode

Link aggregation defined by IEEE 802.3ad allows grouping of Ethernet interfaces to form a single logical interface (LAG). The LAG balances the traffic between the physical interfaces and effectively increases the bandwidth. Another advantage of link aggregation is the increase of link availability between the two equipment. If one of the physical interfaces fails, the LAG will continue to transport traffic using the remaining interfaces.



It is not supported Link Aggregation using different interfaces configuration, for example: VLANs, duplex and speed.



The next steps will indicate how to configure the link-aggregation in a static manner using four (4) Gigabit Ethernet interfaces, to increase link bandwidth up to 4 Gbps.

```
config
link-aggregation interface lag 1
interface gigabit-ethernet-1/1/1
interface gigabit-ethernet-1/1/2
interface gigabit-ethernet-1/1/3
interface gigabit-ethernet-1/1/4
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Link Aggregation](#).

6.2.2 Configuring a LAG in dynamic mode (LACP)

The LACP (Link Aggregation Control Protocol) is a protocol used to assure the end-to-end connectivity of aggregated interfaces (LAG). It detects and protects the network against incorrect configs, assuring the links are aggregated only if they are configured and cabled correctly. The LACP can be configured in two modes:

- **Active mode:** The device sends immediately LACP (PDUs LACP) messages when the interface is activated.
- **Passive mode:** Places an interface in a passive negotiation status, in which the interface only responds to the PDUs of the LACP that receives, but not initiates negotiation of the protocol.

If at least one of the (endpoints) is set as active, the LAG can be made up assuming a successful negotiation of the other parameters.



Aggregation between interfaces with different VLANs, duplex or speed is not supported.

The next steps will indicate how to configure the dynamic aggregation in active mode using two (2) Gigabit Ethernet, interfaces, totaling a band of 2Gbps to the aggregated link.

```
config
link-aggregation interface lag 1
mode active
interface gigabit-ethernet-1/1/1
interface gigabit-ethernet-1/1/2
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Link Aggregation](#).

6.2.3 Configuring the load balancing hash

It is possible change the load balancing hash algorithm used in the calculation to modify traffic forwarding between interfaces in the LAG.



DmOS supports different load balancing hash modes, with **crc16xor8** mode the default.



This feature is global, being valid for all LAGs used in the equipment.

The next steps will indicate how to configure the hash to **xor16**.

```
config
link-aggregation load-balance hash-function xor16
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Link Aggregation](#).

6.2.4 Configuring the load balancing

It is possible change the load balancing algorithm to modify traffic forwarding between interfaces in the LAG.



DmOS supports different load balancing modes, with **enhanced** mode the default.

Below is the list of supported balancing modes:

- enhanced
- dst-ip
- dst-mac
- src-dst-ip
- src-dst-mac
- src-ip
- src-mac

- dynamic

The **dynamic** balance type provides an evenly load distribution across the LAG members. According with the instant traffic load of each link, flows can be moved from members with heavier load to members with lower load dynamically.

The other types of balancing are hash-based. Fields of the packet are analysed with xor computations to select the output LAG member. In these modes packet order is always maintained.

The balance performance will depend on the variability of the content of these fields. Packets with the same information will be directed to the same output LAG member.

The next steps will indicate how to configure the link-aggregation in a static manner using four (4) Gigabit Ethernet interfaces, to increase link bandwidth up to 4 Gbps. The balancing method used will be **dynamic**.

```
config
link-aggregation interface lag 1
load-balance dynamic
interface gigabit-ethernet-1/1/1
interface gigabit-ethernet-1/1/2
interface gigabit-ethernet-1/1/3
interface gigabit-ethernet-1/1/4
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Link Aggregation](#).

6.2.5 Configuring maximum and minimum number of active links in a LAG

- **Maximum Links:** When configuring up a maximum number of active links it is possible to keep redundant links inactive, in case some active link fails, the redundant link takes over as active.
- **Minimum Links:** When configuring a minimum number of active links, if the number of active links is less than the minimum number of configured links, all Link-Aggregation interfaces will be disabled.



The maximum number of active links by default is 16.



The minimum number of active links by default is 1.

The next steps will demonstrate how to configure Link-Aggregation using two (2) Gigabit Ethernet interfaces with a maximum number of one (1) active link:

```
config
link-aggregation interface lag 1
maximum-active links 1
interface gigabit-ethernet-1/1/1
interface gigabit-ethernet-1/1/2
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Link Aggregation](#).

The next steps will demonstrate how to configure Link-Aggregation using two (2) Gigabit Ethernet interfaces with a minimum of two (2) active links:

```
config
link-aggregation interface lag 1
minimum-active links 2
interface gigabit-ethernet-1/1/1
interface gigabit-ethernet-1/1/2
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Link Aggregation](#).

6.2.6 Verifying Link Aggregation

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

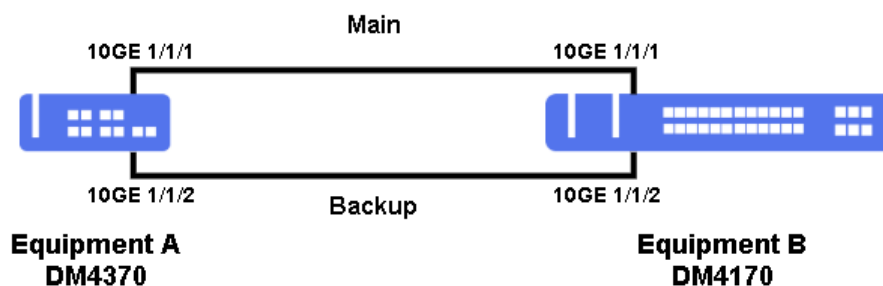
```
show link-aggregation
show link-aggregation brief
show link-aggregation interfaces
show link-aggregation lacp brief
show link-aggregation lacp extensive
show link-aggregation lacp statistics
```

6.3 Backup-Link Configuration

6.3.1 Backup-Link Configuration

It is possible to create a dual homed application using Backup-Link. That is, in case of link interface failed (link down, link-flap detection, EFM, loopback detection, among others.), the backup interface takes its place, preventing packet loss.

The main interface takes over when there is a problem with the backup interface (and the link state of the main interface is up) or when revertive mode is configured. Backup interface can be configured with **block** or **shutdown** action, default is **block**. Backup-Link can be configured on Ethernet interfaces and LAGs.



Backup-Link scenario.

```
config
dot1q
vlan 10
name CLIENT-A
interface ten-gigabit-ethernet-1/1/1
!
interface ten-gigabit-ethernet-1/1/2
!
!
!
backup-link 1
main ten-gigabit-ethernet-1/1/1
backup ten-gigabit-ethernet-1/1/2
!
commit
```

To configure Backup-Link with action *shutdown*, the user should use the following command:

```
config
backup-link 1
action shutdown
!
commit
```

To configure Backup-Link in revertive mode, the user should use the following command:

```
config
backup-link 1
reversion mode revertive
!
commit
```

To configure delay time in revertive mode in Backup-Link, in this case the backup link wait the configured time to return to main interface when backup interface is active, the user should use the following command:



The default time in revertive mode is 35 seconds.

```
config
 backup-link 1
 reversion delay 10
!
commit
```



Revertive mode is not supported with action *shutdown*.



It is possible to configure 64 Backup-Link instances, limited by the number of equipment interfaces.



Cannot configure two backup interfaces for one main interface.



Redundancy protocols that block ports are incompatible with this feature.



The available commands for troubleshooting can be found in the topic [Verifying Backup-Link](#).

6.3.2 Verifying Backup-Link

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show running-config backup-link
show interface <interface-type> <chassis/slot/port>
show interface link
debug enable backup-link
```

6.4 Port Mirroring Configuration

The Port Mirroring allows the Switch to copy network packets from one port to another port on a Switch. This functionality is typically used to mirror traffic, allowing the administrator to monitor the performance of the Switch, and can troubleshoot the network by placing a network analyzer or protocol analyzer on the port that is receiving the mirrored data.

6.4.1 Configuring Port Mirroring for received traffic

The next steps will demonstrate how to configure port mirroring to mirror the inbound traffic of the gigabit-ethernet-1/1/1 interface to the gigabit-ethernet-1/1/2 interface.

```
config
monitor
session 1
destination
interface gigabit-ethernet-1/1/2
|
source
interface gigabit-ethernet-1/1/1
rx
!
!
!
commit
```



There are no troubleshooting commands for this functionality.

6.4.2 Configuring Port Mirroring for transmitted traffic

The next steps will demonstrate how to configure port mirroring to mirror the outbound traffic of the gigabit-ethernet-1/1/1 interface to the gigabit-ethernet-1/1/2 interface.

```
config
monitor
session 1
destination
interface gigabit-ethernet-1/1/2
|
source
interface gigabit-ethernet-1/1/1
tx
!
!
!
commit
```



There are no troubleshooting commands for this functionality.

6.4.3 Configuring Port Mirroring for transmitted and received traffic

The next steps will demonstrate how to configure port mirroring to mirror the inbound and outbound traffic of the gigabit-ethernet-1/1/1 interface to the gigabit-ethernet-1/1/2 interface.

```
config
monitor
session 1
destination
interface gigabit-ethernet-1/1/2
source
interface gigabit-ethernet-1/1/1
all
!
commit
```



There are no troubleshooting commands for this functionality.

6.5 Link Flap Detection Configuration

Link-Flap Detection is a feature that aims to eliminate the side effects of the interminable variation of the link state of a port. This functionality is activated by a certain number of link state changes in a given time interval.

The Link-Flap Detection feature acts by blocking a link when two or more link state change events occur within a configurable time interval. When the port is blocked, state changes are ignored bringing stability to the network. The port will be unlocked after an configurable interval of event free time.

6.5.1 Configuring Link Flap Detection on Ethernet Interface

The next steps will demonstrate how to configure link flap detection of the gigabit-ethernet-1/1/1 interface. In the configuration below, the interface block will occur if 20 state transitions are detected within 60 seconds. If the state transitions stop, after 90 seconds the interface will be unblocked.

```
config
link-flap
interface gigabit-ethernet-1/1/1
detection transitions 20
detection interval 60
detection restore-timeout 90
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Link Flap](#).

6.5.2 Verifying Link Flap

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



If some Ethernet interface is blocked by Link Flap, the **LFD** acronym will appear in the field **Blocked by** of **show interface link** command.



For more details about commands output, check the **Command Reference**.

```
show interface link
show link-flap
show link-flap config-interval
show link-flap config-restore-timeout
show link-flap config-transitions
show link-flap detected-transitions
show link-flap detection-timeout
show link-flap restore-timeout
show link-flap link-flap
```

6.6 Hold Time Configuration

Hold Time allows to configure a timer for the transition announcement of the link state until the interface has remained in the new state for the configured time.

When *hold-down* is configured and the state transition from *Up* to *Down* occurs, the timer for *down-hold-time* is triggered. If the transition from interface status from *Down* to *Up*, the timer is reset.

If the interface remains inactive for the configured time, the state *Down* is announced for the protocols.



DmOS only supports the hold-down configuration.



The hold time value in interfaces which have OSPFv2 and BFD configured must be lower than 200 ms.

6.6.1 Hold Time Configuration

The next steps will demonstrate how to configure hold time on the gigabit-ethernet-1/1/1 interface. In the configuration below, the interface down announcement will occur after 2 seconds (2000 milliseconds). After 2 seconds the device will announce that the interface is inactive.

```
config
hold time interface gigabit-ethernet-1/1/1
down 2000
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Hold Time](#).

6.6.2 Verifying Hold Time

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show log component lim_l2 | include HOLD_TIME
```

6.7 DWDM Interface Configuration

6.7.1 DWDM Interface Configuration

DWDM interfaces allow the configuration of parameters for data transmission, using or not MUX/DEMUX to assist in transmission. According to the module, it is possible to carry out transmissions of up to 120km using only the QSFP-DD. Using the command **show dwdm channels** can check the configurable channels, frequency and wavelength.

- Grid: Set the transmission grid, it can be 100GHz or 50GHz.
- Laser Frequency: Defines the transmission frequency, that is, the DWDM channel that will be occupied and the adjustable range is between 191.3 and 196.1THz, being the step of 0.05THz for 50GHz grid and 0.1THz for 100GHz grid.
- Transmission power: Defines the transmission power in dBm, it can be configured between -20dBm and 0dBm. By default the tx-power value is 0, so if not configured, the module will define the TX power based on the internal saturation, which can vary between -6dBm and -2dBm. This feature is required to perform power leveling when used in combination with MUX/DEMUX DWDM and optical amplifier.



These features are available on the DM4770 16CX equipment and only on the hundred-gigabit-ethernet 10, 12, 14 and 16 interfaces when equipped with QSFP-DD transceivers.



These settings do not affect other QSFP modules.



Laser wavelength information will be available via the command **show dwdm channels**.

```
config
dwdm
interface hundred-gigabit-ethernet-1/1/10
  grid-50ghz frequency 196.1
  tx-power 0.0
!
interface hundred-gigabit-ethernet-1/1/12
  grid-50ghz frequency 194.5
  tx-power -20.0
!
interface hundred-gigabit-ethernet-1/1/14
  grid-100ghz frequency 194.6
  tx-power 0.0
!
interface hundred-gigabit-ethernet-1/1/16
  grid-50ghz frequency 195.35
  tx-power -10.0
!
!
commit
```



Reconfiguring the DWDM parameters causes the link to drop, after a few seconds the link is reestablished.



The available commands for troubleshooting can be found in the topic [Verifying DWDM Interface](#).

6.7.2 Verifying DWDM Interface

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show dwdm channels
show running-config dwdm
show running-config interface hundred-gigabit-ethernet <chassis/slot/port>
show interface link
show inventory chassis 1 transceivers detail
show interface transceivers status
show interface transceivers digital-diagnostics
show log component tcv
```


7 GPON

The GPON uses a WDM (Wavelength Division Multiplexing) technology, allowing the bidirectional transmission on a single fiber (wave length different for downstream and upstream). To segregate traffic of several users, the GPON uses broadcast in the downstream direction (OLT for ONU) and TDMA (Time Division Multiple Access), in the upstream direction (ONU for OLT).

As the data is transmitted from OLT to ONU, the ONUs (Optical network units) should filter the user data traffic as well as coordinate, multiplexing the signals, the client output in order not to create a conflict with the data of other users.

Since the data packs are transmitted in a broadcast manner for all the ONUs, the GPON standard uses AES (Advanced Encryption Standard) to encrypt the flow of data in the downstream direction (OLT for ONU). The encryption is a safe mean to avoid interception and assure that only the authorized user accesses the information.



Read the Datasheet of the equipment to check if these functionalities are available in your hardware platform.

This chapter contains the following sections:

- [Basic Operation of GPON](#)
- [GPON Profiles](#)
- [GPON Service Type](#)
- [Mapping the Service Port](#)
- [Setting GPON Application](#)
- [Automatic Provisioning of ONUs](#)

7.1 Basic Operation of GPON

7.1.1 Setting the GPON interface

To set a GPON interface, the user should enter in the interface config level. To set a GPON interface located in chassis 1, Slot 1 and Port 1 (1/1/1), the user should use the following command:

```
config
interface gpon 1/1/1
```



By default, all the GPON interfaces are deactivated.

To activate a GPON interface, the user should use the following procedure.

```
config
interface gpon 1/1/1
no shutdown
commit
```

To administratively disable a GPON interface, the user should use the following procedure.

```
config
interface gpon 1/1/1
shutdown
commit
```

By default, the FEC is enabled in the GPON interfaces for flows in the downstream and upstream directions. The user may deactivate it using the following configs:

```
interface gpon 1/1/1
no upstream-fec
no downstream-fec
commit
```



The available commands for troubleshooting can be found in the topic [GPON basic verification](#).

7.1.2 Setting the ONUs Authentication Method

The ONU authentication method is a global config of the GPON. Thus, it is applied in all the GPON interfaces.

The default method of authentication is **serial-number**.

The available authentication methods are:

- **serial-number**: Authentication via ONU serial number.
- **password-only**: Authentication with password. The DATACOM ONU password is composed of the ONU serial number without the letters "DA".
Example: If the serial number is DACM12345678 the password for authentication will be CM123456789.
- **serial-number-and-password**: Authentication via serial number plus password combination.

The procedure below indicates how to set the password authentication method.

```
config
gpon 1/1
onu-auth-method password
commit
```



The available commands for troubleshooting can be found in the topic [GPON basic verification](#).

7.1.3 Discovering the ONUS

To discover the ONUs that are linked to some GPON ports of the OLT, the user should perform the procedure described below:

```
show interface gpon discovered-onus
```



The SN (Serial Number) of all the ONUs that are not provisioned in OLT shall be informed

```
# show interface gpon discovered-onus
Chassis / Slot / Port  Serial Number
-----
1/1/1                 DACM00001B30
```

7.1.4 ONU Provisioning

Before provisioning a ONU, it must be included in the list of discovered ONUs, as verified in the previous topic. With the ONU serial number, the user can perform the procedure described below:



Before performing this procedure, it is necessary to have the line-profile configured. For more details on configuring the line-profile, see topic [Profiles GPON](#).

```
config
interface gpon 1/1/1
no shutdown
onu 1
serial-number DACM00001B30
line-profile <LINE_PROFILE_NAME>
!
commit
```

7.1.5 ONU Removing

It is possible to remove a configured ONU. This process is used for cases where the user needs to use the ONU in another PONLINK or just remove a configuration that is not in use. For this process the user can perform the procedure described below:



Before performing this procedure, it is necessary to remove all service-ports linked to the ONU that will be removed.

```
config
interface gpon 1/1/1
no onu 1
!
commit
```

To remove the service-port the user can perform the procedure described below:

```
config
no service-port <SERVICE_PORT_ID>
commit
```



The available commands for troubleshooting can be found in the topic [GPON basic verification](#).

7.1.6 GPON basic verification

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

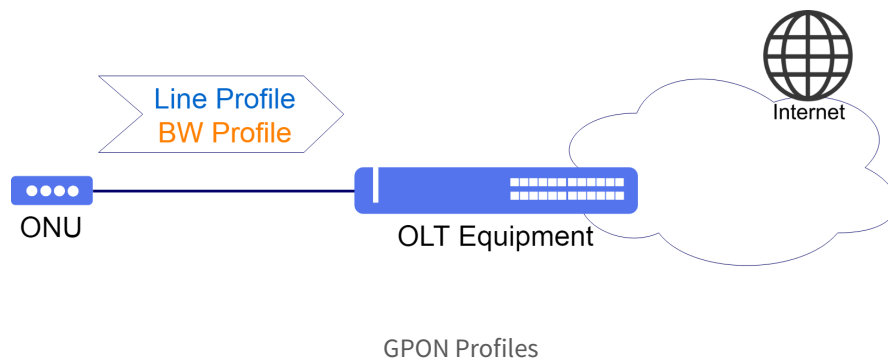
```
show running-config gpon chassis/slot
show interface gpon chassis/slot/port
show interface gpon chassis/slot/port brief
show interface gpon discovered-onus
```

7.2 GPON Profiles

In a typical PON network, there are several end users, but few types of ONU models and services.

Thus, to avoid repetitive provisioning tasks, the GPON profiles allow definition of common attributes that can be reused several times and applied in several service ports.

The figure below is intended to facilitate visualization of where each profile is applied.



After configuring the GPON profiles, they must be associated with the configuration of the ONUs in order for the configured service to operate.

7.2.1 Loading the Default Profiles

It is possible to load the GPON profiles that allow a quick config in the GPON services. It is possible to load the default profiles for ONUs Bridge, for ONUs Router or for both types of ONUs.

To load the default profiles, the user should execute the procedure below:

```
config
load default-gpon-profiles
commit
```

To check the profiles that were loaded, it is possible to execute the show command within the config mode as indicated below:

```
(config)# show profile gpon
profile gpon bandwidth-profile DEFAULT-BANDWIDTH
traffic type-4 max-bw 1106944
!
profile gpon line-profile DEFAULT-LINE
upstream-fec
tcont 1 bandwidth-profile DEFAULT-BANDWIDTH
gem 1
  tcont 1 priority 1
  map any-ethernet
    ethernet any vlan any cos any
  !
  map any-veip
    veip 1 vlan any cos any
  !
gem 2
  tcont 1 priority 0
  map any-iphost
    iphost vlan any cos any
  !
!
profile gpon media-profile DEFAULT-MEDIA
no oob-dtmf
jitter target dynamic-buffer
jitter maximum onu-internal-buffer
codec-order 1
  type pcma
  no silence-suppression
!
codec-order 2
  type pcmu
  no silence-suppression
!
codec-order 3
```

```

type g723
no silence-suppression
!
codec-order 4
type g729
no silence-suppression
!
profile gpon snmp-profile DEFAULT-SNMP
if-type
if-descr
if-oper-status
if-onu-power-rx
statistics-in-bw-usage
statistics-out-bw-usage
!

```

7.2.2 Bandwidth Profile

The bandwidth profile defines the upstream bandwidth allocation characteristics, such as type of T-CONT, fixed bandwidth, bandwidth assured and maximum bandwidth, according to the table below.

BW Type	Delay Affected	Types of applicable T-CONT				
		Type1	Type2	Type3	Type4	Type5
Fixed	Yes	X				X
Assured	No		X	X		X
Non-Assured	No			X		X
Best-Effort	No				X	X
Max	No			X	X	X

Types of Bandwidth vs Types of applicable T-CONT

The commands below provide an example covering creation of a profile that sets a type 3 T-CONT, with 2 Mbit/s of assured band and 10 Mbit/s of maximum band. Only bands multiple of 64 Kbit/s are allowed.

```

config
profile gpon bandwidth-profile <BANDWIDTH_PROFILE_NAME>
traffic type-3 assured-bw 2048 max-bw 9984
commit

```

7.2.3 Line Profile

This profile is used to associate GEM ports to a T-CONT and map a GEM port with ONU services. The GEM port represents a flow of data, which should be associated to a band profile.

Configuration for bridge ONU

The following commands provide an example covering definition of a band profile for a traffic arriving at the Ethernet 1 interface with VLAN ID 100:

```
config
profile gpon line-profile <LINE_PROFILE_NAME>
tcont 1 bandwidth-profile <BANDWIDTH_PROFILE_NAME>
gem 1
tcont 1
map <MAPPING_NAME>
    ethernet 1 vlan 100 cos any
!
!
commit
```

Configuration for router ONU

The following commands provide an example covering definition of a band profile for a traffic arriving at the VEIP 1 interface with VLAN ID 100:

```
config
profile gpon line-profile <LINE_PROFILE_NAME>
tcont 1 bandwidth-profile <BANDWIDTH_PROFILE_NAME>
gem 1
tcont 1
map <MAPPING_NAME>
    veip 1 vlan 100 cos any
!
!
commit
```

7.2.4 Media Profile

The media-profile is used to configure media parameters for VoIP services, allowing the user to set a priority ordered codec list, where is set the codec type, packet-period and silence-suppression for each entry on the list. Media-profile also allows to enable/disable out-of-band DTMF, configure the target of the jitter buffer, and the maximum depth of the jitter buffer and also configure the PSTN protocol variant (country codes) to define the POTS signaling that will be used.



It is mandatory that the ONU supports this configuration via OMCI protocol.

The following commands exemplify the creation of a media-profile that configures the out-of-band DTMF disabled, jitter target in dynamic mode and maximum jitter set by the ONU. POTS signaling will be configured for use in Brazil.

The order of the audio codecs that will be configured:

- **1 PCMA:** Uncompressed, 64 Kbps bandwidth. Used in Europe.
- **2 PCMU:** Uncompressed, 64 Kbps bandwidth. Used in the United States.
- **3 G723:** Compression, bandwidth of 5.3 / 6.3 Kbps.

- **4 G729:** Compression, 8 Kbps bandwidth.



It is necessary to configure at least 4 types of codecs in the media-profile.

```
config
profile gpon media-profile <MEDIA_PROFILE_NAME>
no oob-dtmf
jitter target dynamic-buffer
jitter maximum onu-internal-buffer
pstn-protocol-variant BRA
codec-order 1
type pcma
no silence-suppression
!
codec-order 2
type pcmu
no silence-suppression
!
codec-order 3
type g723
no silence-suppression
!
codec-order 4
type g729
no silence-suppression
!
commit
```

7.2.5 SIP Agent Profile

The SIP Agent profile defines the IP addresses of the servers for the POTS service that will record an analogical line and will control the call process. There are three services to set.

- **Register Server:** This is the server that accepts REGISTRATION requests and places the information received in these requests in the location service for the domain with which it handles.
- **Proxy Server:** It is an intermediary entity that actuates as a server and a client with the objective to submit requests in name of other clients. A proxy server basically performs the role of routing, which means that its job is to assure that a request be sent to another entity "nearest" to the targeted user.
- **Outbond Proxy:** The outbound proxy receives the request of a client, even if it is not the server resolved by the request URI.



The SIP Agent Profile is only valid for ONUs that have POTS interface.

If the user wishes to define a SIP Agent profile in a POTS interface, it should use the following commands:

```
config
profile gpon sip-agent-profile <SIP_AGENT_PROFILE_NAME>
registrar <REGISTRAR_IP_ADDRESS>
proxy-server <PROXY_SERVER_IP_ADDRESS>
outbound-proxy <OUTBOUND-PROXY-IP-ADDRESS>
!
```



```
commit
```

7.2.6 SNMP Profile

The SNMP profile defines which objects (OIDs) will be available for consultation at the ONU through the SNMP manager. After configuring the SNMP profile, it should be added in the ONU configuration in the GPON interface.

Below is a list of some objects that can be enabled for SNMP query through the ONU.



For the other objects that can be configured, check the **Command Reference**.

- **if-admin-status**: Administrative status of the interface.
- **if-alias**: Name associated with the UNI interface.
- **if-descr**: Interface description.
- **if-name**: Name associated with the ONU.
- **if-onu-power-rx**: Optical power received.
- **if-onu-power-tx**: Transmitted optical power.
- **if-onu-sysuptime**: Uptime of the ONU.
- **if-oper-status**: Operational status of the interface.
- **if-type**: interface type (IANA).
- **statistics-in-broadcast-pkts**: Broadcast packet statistics on Ethernet UNI (input).
- **statistics-in-bw-usage**: Bandwidth statistics used on Ethernet UNI (input).
- **statistics-in-discards**: Statistics of discarded packets on Ethernet UNI (input).
- **statistics-in-errors**: Ethernet error packet statistics UNI (input).
- **statistics-in-multicast-pkts**: Multicast packet statistics on Ethernet UNI (input).
- **statistics-in-octets**: Statistics in bytes on Ethernet UNI (input).
- **statistics-in-ucast-pkts**: Statistics of unicast packets on Ethernet UNI (input).
- **statistics-in-unknown-protos**: Unknown protocol packet statistics on the UNI Ethernet (input).
- **statistics-out-broadcast-pkts**: Broadcast packet statistics on UNI Ethernet (output).
- **statistics-out-bw-usage**: Bandwidth statistics used in the UNI Ethernet (output).
- **statistics-out-discards**: Statistics of discarded packets in the UNI Ethernet (output).
- **statistics-out-errors**: Statistics of packets with error on the UNI Ethernet (output).

- **statistics-out-multicast-pkts:** Multicast packet statistics on UNI Ethernet (output).
- **statistics-out-octets:** Statistics in bytes on Ethernet UNI (output).
- **statistics-out-ucast-pkts:** Statistics of unicast packets in UNI Ethernet (output).
- **statistics-out-unknown-protos:** Unknown protocol packet statistics on the UNI Ethernet (output).



Enable the SNMP agent for queries in OLT. Refer to chapter [SNMP Configuration](#)

If user want to define an SNMP profile for the ONUs, it should use the following commands:

```
config
profile gpon snmp-profile <GPON-SNMP-PROFILE>
if-type
if-descr
if-oper-status
if-onu-power-rx
statistics-in-bw-usage
statistics-out-bw-usage
!
commit
```

7.2.7 GEM Traffic Agent Profile

This service is used to apply a limit of Downstream rate in the ONU. It is important for the ISP (Internet Service Provider) to allow the DHCP authentication with the limit of traffic of the network according to the subscription. The GEM traffic profile defines the CIR and EIR band for an ONU.

- **CIR - Committed Information Rate:** This is the rate in Kbps assured to pass through the interface.
- **EIR - Excess Information Rate:** This is the additional rate in Kbps, in case of bandwidth availability, the ONU traffic will be allowed to reach CIR + EIR.

If the user would like to set a GEM traffic profile, which is set together with the T-CONT, which in turn was declared in the line profile, such user should use the following commands:

```
config
profile gpon gem-traffic-profile <GEM_TRAFFIC_PROFILE_NAME>
cir <COMMITTED-RATE>
eir <EXCESS-RATE>
!
profile gpon line-profile <LINE_PROFILE_NAME>
tcont 1 bandwidth-profile <BANDWIDTH_PROFILE_NAME>
gem 1
tcont 1 gem-traffic-profile <GEM_TRAFFIC_PROFILE_NAME>
!
commit
```

7.2.8 Residential Gateway Profile (RG-Profile)

The RG profile defines the router characteristics that should be set in the ONUs. This profile implements a DATACOM proprietary solution and should be used only with DATACOM ONU models DM984-42x with router function.

It is possible to set three main types of connections:

- **wan-pppoe-connection:** PPPoE WAN connection setup. Each connection represents a WAN connection at the ONU. Used for ONU PPPoE authentication.
- **wan-ip-connection:** Configures a default gateway address for the related IP connection in the RG Profile. Used for ONU IP communication.
- **wan-bridge-connection:** WAN bridge connection configuration. Each connection represents a ONU WAN bridge connection. Used for LAN-to-LAN scenarios.

It is allowed to configure **ip-filtering** within the wan-pppoe-connection and wan-ip-connection connections. This setting is **optional**, but will be available in the configuration examples that follow.

If it is necessary to change the password for users **admin** and/or **support**, the the command **user-mgmt priv-lvl-support password custom support2** can be added to the rg-profile. In the following example, the password of the user **support** is changed to **support2**:

```
config
profile gpon rg-profile PPPoE
user-mgmt priv-lvl-support password custom support2
!
commit
```

wan-pppoe-connection

If the user would like to set a Residential Gateway profile for application with **PPPoE authentication** in ONU's WAN using the **VLAN 2000**, it should set a **wan-pppoe-connection** in the RG-Profile. The procedure below shall indicate how to carry out this config:

```
config
profile gpon rg-profile PPPoE
wan-pppoe-connection PPPoE
vlan-mux vlan 2000
nat
no fullcone-nat
firewall
no multicast-proxy igmp
no multicast-source igmp
auth-type auto
ip-filtering 0
incoming
match protocol tcp
action permit
!
!
commit
```

wan-ip-connection

If the user would like to set a Residential Gateway profile for IP application with **DHCP authentication** in ONU's WAN using the **VLAN 2100**, it should set a **wan-ip-connection** in the RG-Profile. The procedure below shall indicate how to perform this config:

```
config
profile gpon rg-profile DHCP
wan-ip-connection DHCP
vlan-mux vlan 2100
nat
no fullcone-nat
firewall
no multicast-proxy igmp
no multicast-source igmp
ipv4 dhcp
primary-dns 10.0.1.1
secondary-dns 10.0.1.2
ip-filtering 1
incoming
match protocol tcp-udp
action permit
!
!
commit
```

wan-bridge-connection

If the user would like to set a Residential Gateway profile for **LAN-to-LAN** application using the **VLAN 520** in the eth1 interface of the ONU, it should set a **wan-bridge-connection** in the RG-Profile. The procedure below shall indicate how to perform this config:

```
config
profile gpon rg-profile RG-ROUTER-520
wan-bridge-connection VLAN-520
vlan-mux vlan 520
no multicast-source igmp
itf-grouping
igmp-snooping
ports eth1 vlan 520
!
!
commit
```

one-shot-provisioning

The routing parameters that can not be configured through Residential Gateway profiles can be configured via the ONU WEB interface. However, after every ONU reboot, these alterations are lost due to the characteristic of reprovisioning of Residential Gateway profiles after every ONU startup, overwriting the WEB interface configurations. To make these configurations permanent, it is possible to enable the option of **rg-one-shot-prov**. The procedure below shall indicate how to carry out this config:

```
config
gpon 1/1
rg-one-shot-prov
!
commit
```

Activating the configuration of **rg-one-shot-prov**, all of the new ONUs activated will have the configuration made via WEB interface persisted, preventing the OLT from applying the Residential Gateway profile after every startup of the ONUs. The ONUs that were configured before the activation of the command **rg-one-shot-prov** only will have the configuration persistent after a reboot or a reprovisioning of the Residential Gateway profile. This operation can be done using the following command:

```
config
interface gpon 1/1/12 onu 4
rg-reprovision
!
commit
```

The command **rg-reprovision** is applied in the ONU configuration level and it is not necessary to commit.



The command **rg-reprovision** must be executed when it is necessary to make a Residential Gateway profile reapplication in the ONU. When the customer realizes a wrong configuration via WEB interface or performs a reset in the ONU configurations are examples of when this command needs to be executed to recover the ONU default functionalities.

The ONUs configured before the activation of the command **rg-one-shot-prov** presents the status field **RG One Shot Provision** with the information **Not provisioned**, indicating that the configurations done via WEB interface can be lost when the ONU restarts. The following example shows this information:

```
show interface gpon 1/1/12 onu 4
Last updated      : 2019-09-27 12:05:54 UTC-3
ID                : 4
Serial Number     : DACM00009C5C
Password          :
Uptime           : 4 min
Last Seen Online  : N/A
Vendor ID         : DACM
Equipment ID      : DM984-422
Name              :
Operational state : Up
Primary status    : Active
Distance          : 0 [km]
IPv4 mode         : DHCP
IPv4 address      : 172.24.1.157/24
IPv4 default gateway : 172.24.1.12
IPv4 VLAN         : 1505
IPv4 CoS          : 0
Line Profile      : Triple-Play-42X-veip
Service Profile   : DM984-42X
RG Profile        : 1-1-12-onu-4
RG One Shot Provision : Not provisioned
SNMP              : Disabled
Allocated bandwidth : 2048 fixed, 22144 assured+fixed [kbit/s]
Upstream-FEC      : Enabled
Anti Rogue ONU isolate : Disabled
Version           : 800.5156.12
Active FW          : v4.1.6-8-g943a valid, committed
Standby FW         : v4.1.6-7-g5f87 valid, not committed
Software Download State : None
Rx Optical Power [dBm] : -8.00
Tx Optical Power [dBm] : Not supported
```

Once applied the command **rg-reprovision** or realized a reset in the ONU, the field **RG One Shot Provision** is filled with the timestamp of when the provisioning of the ONU via Residential Gateway profile was realized, as shown in the following example:

```

show interface gpon 1/1/12 onu 4
Last updated       : 2019-09-27 12:07:27 UTC-3
ID                 : 4
Serial Number      : DACM00009C5C
Password           :
Uptime             : 1 min
Last Seen Online   : N/A
Vendor ID          : DACM
Equipment ID       : DM984-422
Name               :
Operational state   : Up
Primary status      : Active
Distance           : 0 [km]
IPv4 mode           : DHCP
IPv4 address        : 172.24.1.157/24
IPv4 default gateway : 172.24.1.12
IPv4 VLAN           : 1505
IPv4 CoS            : 0
Line Profile        : Triple-Play-42X-veip
Service Profile     : DM984-42X
RG Profile          : 1-1-12-onu-4
RG One Shot Provision : Provisioned on 2019-09-27 12:07:21 UTC-3
SNMP                : Disabled
Allocated bandwidth : 2048 fixed, 22144 assured+fixed [kbit/s]
Upstream-FEC        : Enabled
Anti Rogue ONU isolate : Disabled
Version             : 800.5156.12
Active FW            : v4.1.6-8-g943a valid, committed
Standby FW           : v4.1.6-7-g5f87 valid, not committed
Software Download State : None
Rx Optical Power [dBm] : -7.99
Tx Optical Power [dBm] : Not supported

```



The available commands for troubleshooting can be found in the topic [Verifying GPON profiles](#).

7.2.9 TR-069 ACS Profile

The Auto Configuration Server (ACS) provides a feature of automatic provisioning of services using the TR-069 protocol. To provide a configuration of an ONU via ACS, it is necessary that the ONU supports the TR-069 protocol. Once the ACS server is configured with parameters to be provisioned in ONU, the ONU must know how to reach the ACS server. This information is provided by OLT via OMCI through the **tr069-acs-profile** where is contained a configuration of ACS server URL, username and password.

- **URL:** TR-069 server address.
- **Username:** TR-069 server username.
- **Password:** TR-069 server password.

To perform the creation of an ACS profile, use the following commands:

```

config
profile gpon tr069-acs-profile <ACS_PROFILE_NAME>
url <ACS-URL>
username <ACS-USERNAME>
password <ACS-PASSWORD>
!
commit

```

Once defined, the ACS profile must be applied to the ONUs that need to use automatic provisioning services. The following commands show the process of creation of an ACS profile and their application in a specific ONU:

```
config
profile gpon tr069-acis-profile TR-069
url http://tr-069-server.internal:17000
username datacom
password datacom1234
!
interface gpon 1/1/1 onu 0
tr069-acis-profile TR-069
!
commit
```



ONU must support provisioning via the TR-069 protocol. Before using, verify the product documentation to make sure if the ONU supports configuration via TR-069 protocol.



The available commands for troubleshooting can be found in the topic [Verifying GPON profiles](#).

7.2.10 Verifying GPON profiles

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show running-config profile gpon
```

7.3 GPON Service Type

It is possible to set several GPON applications between the OLT and the ONUs. The DmOS supports three main types of services:

7.3.1 Service VLAN N:1

- **N:1:** In general, this type of service is implemented to provide access to Internet to residential clients, since only one VLAN is used to transport the internet service in the entire network.

The following command will set a VLAN for the N:1 service. This means that the clients (N) in the same VLAN (1) shall not be able to communicate one with the other.

```
config
service vlan 100 type n:1
commit
```

7.3.2 Service VLAN 1:1

- **1:1:** In general, this type of service is implemented to provide corporate applications or access to residential internet, since a different VLAN is used to transport the service of each client through the network. Each class of traffic of the same subscriber should have a same VLAN.

The following command will set a VLAN for the 1:1 service. This means that the clients in the same VLAN shall not be able to communicate one with the other.

```
config
service vlan 100 type 1:1
commit
```

7.3.3 Service VLAN TLS

- **TLS:** In general, this type of service is implemented to provide corporate applications, since a different VLAN is used to transport the service of each client through the network. Each class of traffic of the same subscriber can have a same or a different VLAN. This service when used together with the Hairpin enables offer of LAN-to-LAN services and no additional equipment will be required (for example routers).

The following command will set a VLAN for the TLS service. This means that the clients in the same VLAN shall be able to communicate one with the other.

```
config
service vlan 100 type tls
commit
```

7.4 Mapping the Service Port

A service port is used to establish a relationship between the traffic of GEM port and the service VLAN.

Examples of applications with Service Port:

7.4.1 Service Port - Transparent

For example, the traffic originating from VLAN ID 100 at the ONUs will be routed transparently through the following command:

```
config
service-port 1 gpon 1/1/1 onu 1 gem 1 match vlan vlan-id 100 action vlan replace vlan-id 100
commit
```


7.4.2 Service Port - Replace

For example, the traffic resulting from VLAN ID 100 in the ONUs will be mapped for the Service VLAN 200 using the following commands:

```
config
service-port 1 gpon 1/1/1 onu 1 gem 1 match vlan vlan-id 100 action vlan replace vlan-id 200
commit
```

7.4.3 Service Port - Add

For example, the traffic originating from VLAN ID 100 at ONUs will receive a new tag VLAN 1000 through the following command:

```
config
service-port 1 gpon 1/1/1 onu 1 gem 1 match vlan vlan-id 100 action vlan add vlan-id 1000
commit
```



A service-port that uses an N:1 VLAN service only supports the replace operation.

7.5 Setting GPON Application

To configure a GPON application with ONU (bridge/router), follow the steps below:

- **Configure VLAN**
- **Configure Service VLAN**
- **Enable the GPON interface**
- **Verify ONU Discovery**
- **Configure the bandwidth-profile**
- **Configure the line-profile**
- **Configure the RG Profile (only for ONU router)**
- **Provisionar a ONU**
- **Configure the Service Port**

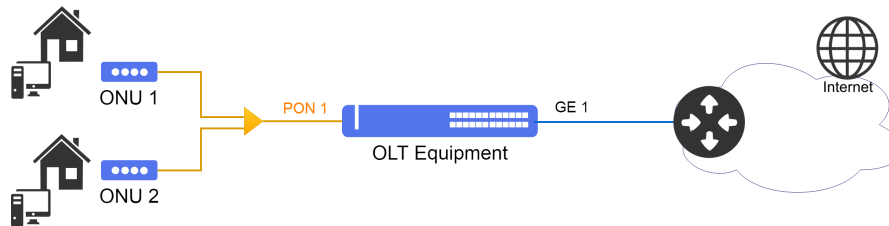


The configurations below are compatible with ONUs DATACOM. For ONUs from other manufacturers, it may be necessary to contact DATACOM Technical Support to check compatibility.

7.5.1 Configuring a N:1 Application with ONU bridge

Considering that the user would like to set two customers with the same band profile. For this example, the N:1 type of service will be set exemplifying the supply of Internet access to the residential clients. The VLAN 100 will be used as a service VLAN. The **native vlan** configuration at the ONU will be used to deliver the service without a VLAN tag over the ONU Ethernet.

The scenario below will be used to indicate the config of the service using the Ethernet interface of the ONUs.



Implementing an Internet Access Service using the Service-VLAN N:1

Below is the complete configuration for the N:1 bridge ONU application:

```

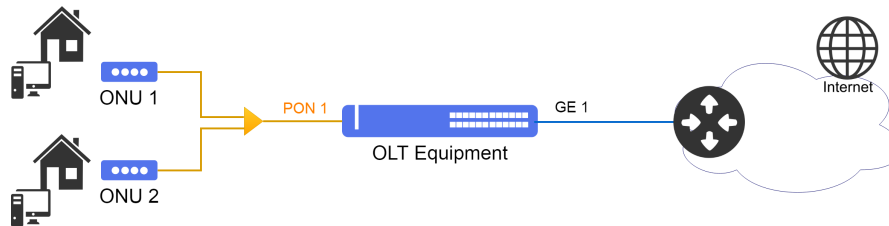
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1
!
service vlan 100 type n:1
!
profile gpon bandwidth-profile DEFAULT-BANDWIDTH
traffic type-4 max-bw 1106944
!
profile gpon line-profile DEFAULT-LINE
upstream-fec
tcont 1 bandwidth-profile DEFAULT-BANDWIDTH
gem 1
tcont 1 priority 0
map ethernet
ethernet 1 vlan 100 cos any
native vlan vlan-id 100
!
interface gpon 1/1/1
no shutdown
onu 1
name CLIENTE-1
serial-number DACM00000001
line-profile DEFAULT-LINE
ethernet 1
negotiation
no shutdown
!
onu 2
name CLIENTE-2
serial-number DACM00000002
line-profile DEFAULT-LINE
ethernet 1
negotiation
no shutdown
native vlan vlan-id 100
!
!
service-port 1 gpon 1/1/1 onu 1 gem 1 match vlan vlan-id 100 action vlan replace vlan-id 100
service-port 2 gpon 1/1/1 onu 2 gem 1 match vlan vlan-id 100 action vlan replace vlan-id 100
!
commit

```

7.5.2 Configuring a 1:1 Application with ONU bridge

Considering that the user would like to set two customers with the same band profile. For this example, the 1:1 type of service will be set exemplifying the supply of Internet access for corporate clients. The VLAN 100 will be used as the ONU 1 service VLAN and the VLAN 200 as the ONU 2 service VLAN. Service delivery takes place with a VLAN tag on the ONU Ethernet.

The scenario below will be used to indicate the config of the service using the Ethernet interface of the ONUs.



Implementing an Internet Access Service using the Service-VLAN 1:1

Below is the complete configuration for the 1:1 bridge ONU application:

```

config
dot1q
vlan 100,200
interface gigabit-ethernet-1/1/1
!
!
service vlan 100 type 1:1
!
service vlan 200 type 1:1
!
profile gpon bandwidth-profile DEFAULT-BANDWIDTH
traffic type-4 max-bw 1106944
!
profile gpon line-profile LINE-1
upstream-fec
tcont 1 bandwidth-profile DEFAULT-BANDWIDTH
gem 1
tcont 1 priority 0
map ethernet
ethernet 1 vlan 100 cos any
!
!
profile gpon line-profile LINE-2
upstream-fec
tcont 1 bandwidth-profile DEFAULT-BANDWIDTH
gem 1
tcont 1 priority 0
map ethernet
ethernet 1 vlan 200 cos any
!
!
interface gpon 1/1/1
no shutdown
onu 1
name CLIENTE-1
serial-number DACM00000001
line-profile LINE-1
ethernet 1
negotiation
no shutdown
!
!
onu 2
name CLIENTE-2
serial-number DACM00000002
line-profile LINE-2
ethernet 1

```

```

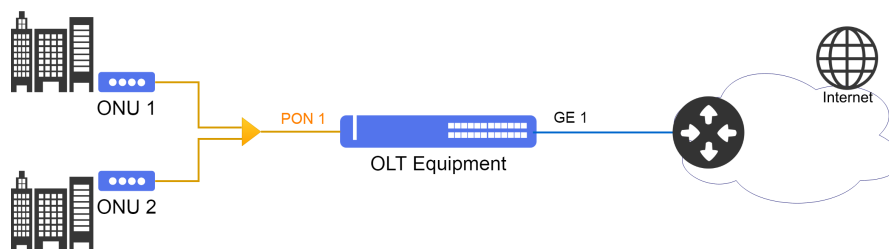
negotiation
no shutdown
!
!
service-port 1 gpon 1/1/1 onu 1 gem 1 match vlan vlan-id 100 action vlan replace vlan-id 100
service-port 2 gpon 1/1/1 onu 2 gem 1 match vlan vlan-id 200 action vlan replace vlan-id 200
!
commit

```

7.5.3 Configuring a TLS Application with ONU router

Considering that the user wishes to configure a LAN-to-LAN application for protocol transparency and the possibility of the clients communicating with each other. For this example, the TLS service type will be configured by exemplifying the delivery of enterprise applications. The VLAN 100 will be used as a service VLAN.

The scenario below will be used to indicate the config of the service using the VEIP interface of the ONUs DATACOM model DM98x.



Implementing an Corporate Service using the Service-VLAN TLS



it is necessary to configure an RG Profile so that the settings are sent to ONU DM98x.

The main change over the for ONU bridge is in the configured interface type which goes from **ethernet** to **veip**.

Below the complete configuration for the application TLS with ONU router model DM98x.

```

config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1
!
!
service vlan 100 type tls
!
profile gpon bandwidth-profile DEFAULT-BANDWIDTH
traffic type-4 max-bw 1106944
!
profile gpon line-profile DEFAULT-LINE
upstream-fec
tcont 1 bandwidth-profile DEFAULT-BANDWIDTH
gem 1
tcont 1 priority 0
map veip
veip 1 vlan 100 cos any
!
!
!

```

```

profile gpon rg-profile RG-DM98x
wan-bridge-connection VLAN-100
vlan-mux vlan 100
no multicast-source igmp
itf-grouping
igmp-snooping
ports eth1 vlan 100
!
!
interface gpon 1/1/1
no shutdown
onu 1
name CLIENTE-1
serial-number DACM00000001
rg-profile RG-DM98x
line-profile DEFAULT-LINE
veip 1
!
onu 2
name CLIENTE-2
serial-number DACM00000002
rg-profile RG-DM98x
line-profile DEFAULT-LINE
veip 1
!
!
service-port 1 gpon 1/1/1 onu 1 gem 1 match vlan vlan-id 100 action vlan replace vlan-id 100
service-port 2 gpon 1/1/1 onu 2 gem 1 match vlan vlan-id 100 action vlan replace vlan-id 100
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying GPON applications](#).

7.5.4 Configuring a GPON application with MPLS

To configure this application, consult the topic [MPLS with GPON access](#).

7.5.5 Verifying GPON applications

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```

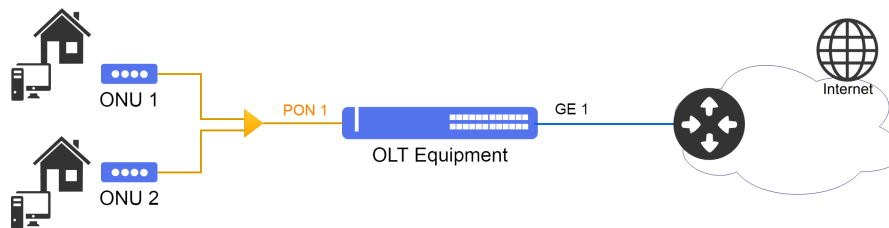
show interface gpon <chassis/slot/port> statistics
show interface gpon onu
show interface gpon <chassis/slot/port> onu <id>
show interface gpon <chassis/slot/port> onu <id> gem <id> statistics
show mac-address-table

```

7.6 Automatic Provisioning of ONUs

The automatic provisioning tool is used to set in an automatic manner all the ONUs discovered in the OLT based on a set of pre-defined profiles. The config is performed in a global manner and will apply the config defined in the automatic provisioning to all the discovered ONUs.

The GPON created profiles should be included in the automatic provisioning and also it is possible to use the loaded default profiles.



Implementing an Internet Access Service using the Service-VLAN N:1

Assume that the user wants all ONU that are discovered in the PON branch to be automatically configured with the profiles previously configured.

The following procedure presents the configuration required to enable automatic provisioning.

```

config
gpon 1/1
onu-auto-provisioning
enable
line-profile <DEFAULT-LINE>
ethernet 1
veip 1
service-port 1 gem 1 match vlan vlan-id <VLAND-ID> action vlan add vlan-id <VLAND-ID>
!
commit

```

Alternatively, if it is necessary to configure the automatic provisioning functionality only to one GPON interface, the following commands can be used:

```

config
interface gpon 1/1/1
onu-auto-provisioning
enable
line-profile <DEFAULT-LINE>
ethernet 1
veip 1
service-port 1 gem 1 match vlan vlan-id <VLAND-ID> action vlan add vlan-id <VLAND-ID>
!
!
commit

```



The configuration of automatic provisioning functionality per PONLINK has precedence over the Global configuration. Both configurations can coexist in the same equipment, but the Global one will be activated only when PONLINK configuration is not present.



The service-port should be created for each GEM that the user wishes to set. It is possible to set up to 16 service-ports in the automatic provisioning, which will be applied in all the discovered ONUs.

Below the complete configuration for the application with automatic provisioning in the Global format:

```
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1
!
!
service vlan 100 type n:1
!
profile gpon bandwidth-profile DEFAULT-BANDWIDTH
traffic type-4 max-bw 1106944
!
profile gpon line-profile DEFAULT-LINE
upstream-fec
tcont 1 bandwidth-profile DEFAULT-BANDWIDTH
gem 1
tcont 1 priority 0
map any-ethernet
ethernet any vlan any cos any
!
map any-veip
veip 1 vlan 100 cos any
!
!
gem 2
tcont 1 priority 0
map any-iphost
iphost vlan any cos any
!
!
!
profile gpon snmp-profile DEFAULT-SNMP
if-type
if-descr
if-oper-status
if-onu-power-rx
statistics-in-bw-usage
statistics-out-bw-usage
!
interface gpon 1/1/1
no shutdown
!
gpon 1/1
onu-auto-provisioning
enable
line-profile DEFAULT-LINE
snmp-profile DEFAULT-SNMP
ethernet 1
veip 1
service-port 1 gem 1 match vlan vlan-id 100 action vlan add vlan-id 100
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying GPON](#).

7.6.1 Verifying GPON

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show interface gpon onu
show interface gpon <chassis/slot/port> onu <id>
show mac-address-table
```

7.6.2 Verifying automatic provisioning

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show running gpon
show interface gpon onu
show interface gpon <chassis/slot/port> onu <id>
```


8 Switching

In a network of Layer 2, each network segment has its own collision domain and all the segments are in the same transmission domain. The entire transmission is seen by all the network devices. The 802.1Q standard allows creation of VLANs that are used to segment a single broadcast domain into several broadcast domains.



The 802.1Q standard supports frames tagged from identifier 1 to 4094.

This chapter contains the following sections:

- [MAC Table Configuration](#)
- [VLAN Configuration](#)
- [RSTP Configuration](#)
- [MSTP Configuration](#)
- [EAPS Configuration](#)
- [ERPS Configuration](#)
- [L2CP Configuration](#)
- [Loopback Detection Configuration](#)
- [DHCP Relay L2 Configuration](#)

8.1 MAC Table Configuration

8.1.1 Configuring Aging Time

The switching equipment operate in L2 layer and execute forwarding of frames through MAC addresses. The MAC address table stores the MAC addresses caught by the device, associating them to an interface port.

The MAC addresses are caught dynamically or statically by the device. In static mode, the user saves an input with MAC address and port. This input shall persist in the table until it is removed by the user. In the dynamic mode, the switch receives a frame and saves the MAC address of origin and the interface port in the table. This address will continue saved while a traffic exist or will wait for the aging time to clean this input in the table. The standard value of the aging time is 600 seconds.

The next steps will indicate how to set the aging time for the value of **300** seconds.

```
config
mac-address-table aging-time 300
commit
```



The available commands for troubleshooting can be found in the topic [Verifying MAC Address Table](#).

8.1.2 Disabling MAC learning

MAC learning in interfaces is enabled by default. If that is not desirable, it can be disabled per interface.



Disabling MAC learning in an interface can result in frame flooding, causing frames to be forwarded unnecessarily.

The commands below show how to disable MAC learning in the interface gigabit-ethernet 1/1/6.

```
config
mac-address-table interface gigabit-ethernet-1/1/6 learning disabled
commit
```



The available commands for troubleshooting can be found in the topic [Verifying MAC Address Table](#).

8.1.3 Verifying MAC Address Table

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show mac-address-table
show mac-address-table interface <INTERFACE>
show mac-address-table mac-address <MAC_ADDRESS>
show mac-address-table type <STATIC/DYNAMIC>
show mac-address-table vlan <VLAN_ID>
show running-config mac-address-table interface learning
```

8.2 VLAN Configuration

8.2.1 Configuring VLAN with Tagged Interfaces

The **tagged** mode is used in the interfaces that perform the output and input of traffic with VLAN ID (802.1Q) tag.

The next steps will define how to set the VLAN 200 for output of data traffic between the Gigabit Ethernet 1/1/1 and Gigabit Ethernet 1/1/2 interfaces using the tagged mode.

```
config
dot1q
vlan 200
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
```



By default, if the user does not specify the interface mode in the VLAN, the mode used will be **tagged**.

It is also possible for the user to set several VLANs using a range and inserting the desired interfaces. The procedure below exemplifies config of a range of VLANs from ID 1500 through ID 2000 with the ten-gigabit-ethernet 1/1/1 interface in tagged mode.

```
config
dot1q
vlan 1500-2000
name TRAFEGO
interface ten-gigabit-ethernet-1/1/1 tagged
commit
```



The available commands for troubleshooting can be found in the topic [Verifying VLAN Configuration](#).

8.2.2 Configuring VLAN with Untagged Interfaces

The untagged mode is used in the interfaces that perform the output and input of traffic and that do not have the VLAN ID (802.1Q) tag. This mode is used mainly in the interfaces connected to computers, servers, printers, etc.



For the untagged traffic, setting of a native-vlan in the interfaces using the switchport config is required

The next steps will define how to set the VLAN 200 for traffic between the Gigabit Ethernet 1/1/1 and Gigabit Ethernet 1/1/2 interfaces using the untagged mode.

```
config
dot1q
vlan 200
interface gigabit-ethernet-1/1/1 untagged
interface gigabit-ethernet-1/1/2 untagged
!
switchport
interface gigabit-ethernet-1/1/1
native-vlan
vlan-id 200
!
```

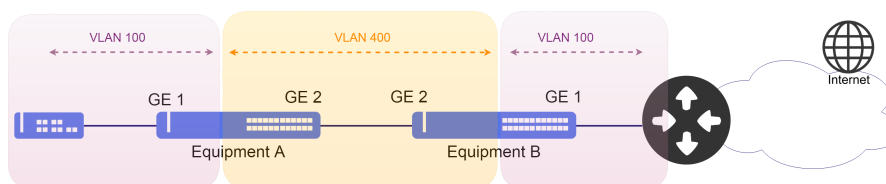
```
interface gigabit-ethernet-1/1/2
 native-vlan
 vlan-id 200
 commit
```



The available commands for troubleshooting can be found in the topic [Verifying VLAN Configuration](#).

8.2.3 Configuring VLAN Translate

The VLAN-Translate performs replacement of a given VLAN by another VLAN in the out direction or in the in direction of the traffic.



Implementation of VLAN Translate

The next steps will indicate how to set the VLAN Translate to translate the VLAN 100 into the VLAN 400 in the input (in) of the gigabit ethernet 1/1/1 interface and execute the opposite operation in the output (out).



VLAN-Translate configuration is done through the VLAN mapping functionality, using the action **replace**.

```
config
dot1q
vlan 400
interface gigabit-ethernet-1/1/1
!
interface gigabit-ethernet-1/1/2
!
!
vlan-mapping
interface gigabit-ethernet-1/1/1
ingress
rule TRANSLATE-ingress-rule1
match vlan vlan-id 100
action replace vlan vlan-id 400
egress
rule TRANSLATE-egress-rule1
match vlan vlan-id 400
action replace vlan vlan-id 100
commit
```

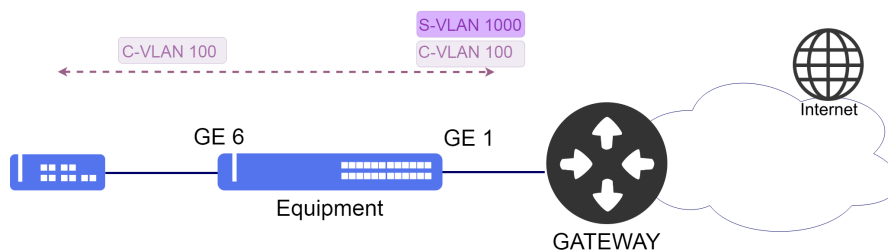


There are no troubleshooting commands for this functionality.

8.2.4 Configuring QinQ

The QinQ is a L2 functionality also known as tunneling QinQ, 802.1Q tunnel, VLAN Stacking or double-tag. With this functionality, a service provider can assign different service VLANs (S-VLANs) to a given type of different clients traffic, or even a single VLAN to all the clients. This allows a separation between the traffic of each client in the service provider's network. The client's VLANs are then transported in a transparent manner within the service provider's network.

The scenario below will be used to illustrate the config of the QinQ.



Implementation of QinQ

The next steps will indicate how to set the QinQ to transport one client connected to the Gigabit-Ethernet-1/1/6 interface. The client have a **VLAN (C-VLAN) 100** and will be transported through the service provider's network with the **VLAN (S-VLAN) 1000**.



To set the QinQ it is necessary to set the interface in an untagged manner and activating the option QinQ using the switchport config.

```
config
dot1q
vlan 1000
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/6 untagged
!
switchport
interface gigabit-ethernet-1/1/6
qinq
native-vlan vland-id 1000
commit
```



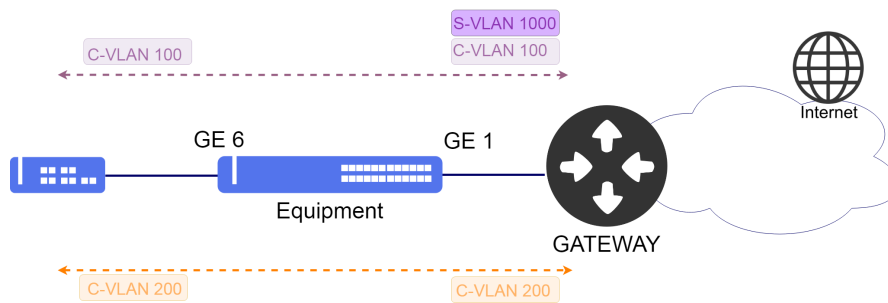
There are no troubleshooting commands for this functionality.

8.2.5 Configuring Selective QinQ

The selective QinQ has the same logic of the standard QinQ, however it adds a new VLAN in the traffic that enters in an interface only for the specified VLAN (C-VLANs) of clients. This functionality has as objective to create service VLANs

(S-VLANs) to separate clients (C-VLANs) that diverge in the type of contracted service, such as for example the QoS.

The scenario below will be used to illustrate config of the selective QinQ.



Implementation of Selective QinQ

Considering that the user would like to set two different customers. Both connected to the gigabit-ethernet-1/1/6 interface, however, the client with a VLAN 100 (C-VLAN) will be transported in a transparent manner within the service provider's network through the VLAN 1000 (S-VLAN) and the second client will have the VLAN 200 (C-VLAN) maintained. The next steps will indicate how to set the selective QinQ.



To set the selective QinQ, config of a rule in the vlan-mapping using the option **add** is required.

```
config
dot1q
vlan 200
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/6 tagged
!
vlan 1000
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/6 untagged
!
vlan-mapping
interface gigabit-ethernet-1/1/6
ingress
rule qinq-seletivo-vlan-100
match vlan vlan-id 100
action add vlan vlan-id 1000
commit
```



There are no troubleshooting commands for this functionality.

8.2.6 Verifying VLAN Configuration

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show vlan brief
show vlan detail
show vlan membership detail
```

8.3 RSTP Configuration

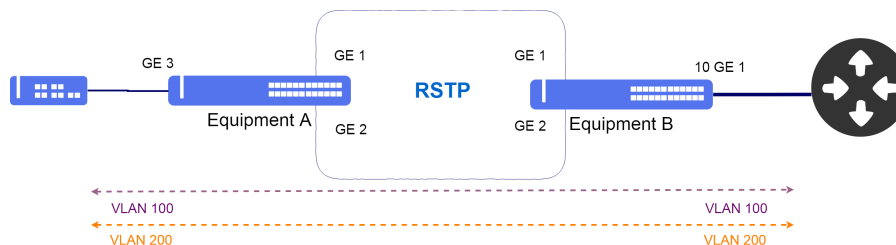
The RSTP (Rapid Spanning Tree Protocol) protocol defined by IEEE 802.1w is used to provide a single path in the network, eliminating loops among equipment.



The BPDU of the RSTP are forwarded without presence of VLAN (untagged).

8.3.1 Configuring a Basic RSTP

The scenario below will be used to illustrate config of the RSTP.



Implementation of RSTP

Considering that the user would like to execute the following configs:

- **Equipment A:** VLAN ID 100 and 200 for traffic with a gigabit-ethernet-1/1/3 interface as uplink interface.
- **Equipment B:** VLAN ID 100 and 200 for traffic with a ten-gigabit-ethernet-1/1/1 interface as uplink interface.

```
!Equipment A
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
interface gigabit-ethernet-1/1/3 tagged
!
vlan 200
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
interface gigabit-ethernet-1/1/3 tagged
!
```

```
!
!
spanning-tree
interface gigabit-ethernet-1/1/1
interface gigabit-ethernet-1/1/2
commit
```

```
!Equipment B
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
interface ten-gigabit-ethernet-1/1/1 tagged
!
vlan 200
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
interface ten-gigabit-ethernet-1/1/1 tagged
!
!
spanning-tree
interface gigabit-ethernet-1/1/1
interface gigabit-ethernet-1/1/2
commit
```

8.3.2 Applying RSTP parameters

If necessary, the user can change Spanning Tree default parameters or enable parameters that are not available in the Spanning Tree default configuration.

Below the list of parameters that can be modified at the Spanning Tree interfaces:

- **auto-edge:** If do not receive BPDUs, the interface goes automatically to edge port state and will not transmit BPDUs.
- **edge-port:** When configured, the interface will not transmit BPDUs. The default value is auto-edge.
- **bpdu-guard:** The interface goes to forwarding state but will not transmit BPDUs, unless a BPDU is received by interface. This command only applies to interfaces that have already been configured as edge-port.
- **cost:** Allows to change the cost of the interface path for STP calculations. By default, this value is associated with the link speed.
- **link-type:** Configure the interface to inform whether the LAN segment is point-to-point or point-multipoint. The default value is auto.
- **port-priority:** Configure the port priority to change the probability of becoming a root port. The default value is 128.
- **restricted-role:** Settings the interface to not be selected as the Root Port of an STP topology.
- **restricted-tcn:** Settings the interface to not propagate the notifications received from changing the STP topology.

The next steps will demonstrate how to configure edge-port on interface gigabit-ethernet-1/1/1.



To configure the other parameters listed above, the procedure is the same as in the example below.


```
config
spanning-tree
interface gigabit-ethernet-1/1/1 edge-port
commit
```



The available commands for troubleshooting can be found in the topic [Verifying RSTP](#).

8.3.3 Verifying RSTP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show spanning-tree
show spanning-tree brief
show spanning-tree detail
show spanning-tree extensive
```

8.4 MSTP Configuration

The MSTP (Multiple Spanning Tree Protocol) protocol defined by IEEE 802.1s is used to provide a single path in the network, eliminating loops among equipment. The protocol has advantage over RSTP of providing a load balancing through of different MSTI instances by tuning the ports costs for efficient load balancing.



The protected VLAN group in a MSTI can only overlap with the VLAN group protected by the EAPS and ERPS protocols if the VLANs are exactly the same.



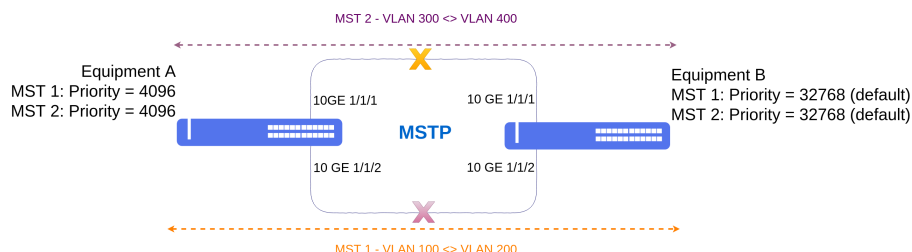
There can be no overlap of VLANs between MSTIs.



The Control VLANs of ERPS and EAPS can not be part of a MSTI.

8.4.1 Configuring MSTP for load balancing

The scenario below will be used to illustrate config of the RSTP for efficient load balancing.



Implementation of MSTP

Considering that the user would like to execute the following configs:

Equipment A

- **MST 1:** Priority 4096 and VLAN ID 100 to 200. Port-priority = 32 in ten-gigabit-ethernet-1/1/2.
- **MST 2:** Priority 4096 and VLAN ID 300 to 400.

Equipment B

- **MST 1:** Priority 32768 (default) and VLAN ID 100 to 200.
- **MST 2:** Priority 32768 (default) and VLAN ID 300 to 400.

The both equipment use the ten-gigabit-ethernet 1/1/1 and ten-gigabit-ethernet 1/1/2 interfaces to form MSTP topology with the following parameters: **name = datacom** and **revision = 12345**.



The parameters **name** and **revision** MUST be the same on all equipment that participate in MSTP topology.



The interfaces present in the CIST are required in MSTI instances only if need to set specific parameters such as **cost** and **priority**.

```
!Equipment A
config
dot1q
vlan 100-200,300-400
  interface ten-gigabit-ethernet-1/1/1
  !
  interface ten-gigabit-ethernet-1/1/2
  !
!
spanning-tree
mode mstp
name datacom
revision 12345
```

```
interface ten-gigabit-ethernet-1/1/1
  auto-edge
!
interface ten-gigabit-ethernet-1/1/2
  auto-edge
!
mst 1
  priority 4096
  vlan 100-200
  interface ten-gigabit-ethernet-1/1/2
    port-priority 32
!
mst 2
  priority 4096
  vlan 300-400
commit
```

```
!Equipment B
config
dot1q
vlan 100-200,300-400
  interface ten-gigabit-ethernet-1/1/1
  !
  interface ten-gigabit-ethernet-1/1/2
  !
!
spanning-tree
  mode mstp
  name datacom
  revision 12345
  interface ten-gigabit-ethernet-1/1/1
    auto-edge
  !
  interface ten-gigabit-ethernet-1/1/2
    auto-edge
  !
  mst 1
    vlan 100-200
  !
  mst 2
    vlan 300-400
commit
```



The available commands for troubleshooting can be found in the topic [Verifying MSTP](#).

8.4.2 Verifying MSTP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show spanning-tree
show spanning-tree brief
show spanning-tree detail
show spanning-tree extensive
```

8.5 EAPS Configuration

The EAPS (Ethernet Automatic Protection Switching) protocol is used to provide a single path in the network and eliminating loops among equipment. It also provides a quicker convergence in relation to the RSTP protocol.



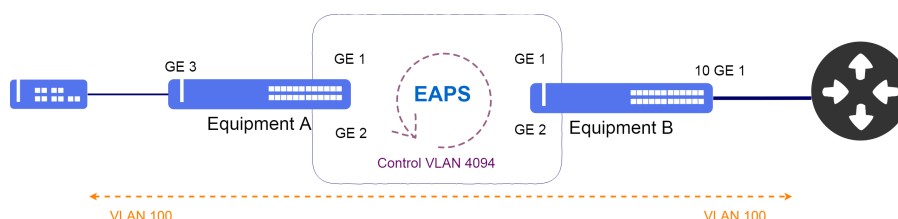
The EAPS protocol operated adequately only in ring type topologies.



The protected VLAN group in a EAPS instance can only overlap with the VLAN group protected by the MSTP and ERPS protocols if the VLANs are exactly the same.

8.5.1 Configuring a Basic Ring EAPS

The scenario below will be used to illustrate the config of the EAPS.



Implementation of EAPS

Considering that the user would like to execute the following configs:

- **Equipment A:** VLAN 100 for traffic with a gigabit-ethernet-1/1/3 interface as access interface and the VLAN 4094 for VLAN of EAPS control in transit mode using the gigabit-ethernet-1/1/1 and 1/1/2 interfaces.
- **Equipment B:** VLAN 100 for traffic with a ten-gigabit-ethernet-1/1/1 interface as uplink interface and the VLAN 4094 for VLAN of EAPS control in master mode using the gigabit-ethernet-1/1/1 and 1/1/2 interfaces.

```
!Equipment A
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
interface gigabit-ethernet-1/1/3 tagged
!
vlan 4094
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
!
!
eaps 0
control-vlan 4094
protected-vlans 100
port
primary gigabit-ethernet-1/1/1
```

```
secondary gigabit-ethernet-1/1/2
!  
!  
mode transit  
commit
```

```
!Equipment B  
config  
dot1q  
vlan 100  
interface gigabit-ethernet-1/1/1 tagged  
interface gigabit-ethernet-1/1/2 tagged  
interface ten-gigabit-ethernet-1/1/1 tagged  
!  
vlan 4094  
interface gigabit-ethernet-1/1/1 tagged  
interface gigabit-ethernet-1/1/2 tagged  
!  
!  
eaps 0  
control-vlan 4094  
protected-vlans 100  
port  
primary gigabit-ethernet-1/1/1  
secondary gigabit-ethernet-1/1/2  
!  
!  
mode master  
commit
```



The available commands for troubleshooting can be found in the topic [Verifying EAPS](#).

8.5.2 Verifying EAPS

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show eaps  
show eaps brief  
show eaps detail
```

8.6 ERPS Configuration

The ERPS (Ethernet Ring Protection Switching) protocol defined by rule ITU-U G.8032 is used to provide a single path in the network avoiding and eliminating loops among equipment.

Inhibition of loop in an Ethernet ring is carried out assuring that a segment remains without passing of traffic, this means, barred. The ERPS protocol uses a port called RPL Owner responsible for barring the entire traffic, except the control packs of the protocol.



It is recommended to use the adjacent interface connected to the RPL Owner as RPL Neighbor. This configuration is optional, however, it assists the RPL Owner in blocking link traffic.



It is mandatory to configure "ring-id 1" if it is necessary to interoperate with other DATACOM products and other vendors that have ERPSv1.



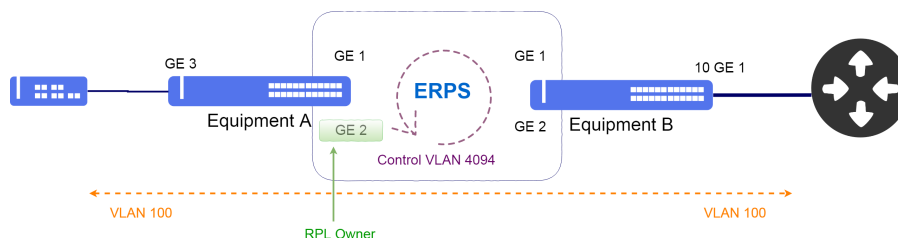
The protected VLAN group in a ERPS instance can only overlap with the VLAN group protected by the EAPS and MSTP protocols if the VLANs are exactly the same.



In scenarios that have DWDM, SDH or other data transport technology that does not propagate the link down, it is necessary to use Ethernet OAM or CFM with MEP DOWN for ERPS convergence in case of failure of the technology used.

8.6.1 Configuring an ERPS Single-ring

The scenario below will be used to illustrate the config of the ERPS.



Implementation of ERPS Single-ring

Considering that the user would like to execute the following configs:

- **Equipment A:** VLAN 100 for traffic with a gigabit-ethernet-1/1/3 interface as access interface and the VLAN 4094 for control VLAN of ERPS and a gigabit-ethernet-1/1/2 interface as RPL Owner.
- **Equipment B:** VLAN 100 for traffic with a ten-gigabit-ethernet-1/1/1 interface as uplink interface and the VLAN 4094 for control VLAN of the ERPS.

```
!Equipment A
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
```

```

interface gigabit-ethernet-1/1/3 tagged
!
vlan 4094
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
!
!
erps
ring ERPS
ring-id 1
control-vlan 4094
protected-vlans 100
port0
interface gigabit-ethernet-1/1/1
!
port1
interface gigabit-ethernet-1/1/2 rpl-role owner
!
!
!
commit

```

```

!Equipment B
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
interface ten-gigabit-ethernet-1/1/1 tagged
!
vlan 4094
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 tagged
!
!
erps
ring ERPS
ring-id 1
control-vlan 4094
protected-vlans 100
port0
interface gigabit-ethernet-1/1/1
!
port1
interface gigabit-ethernet-1/1/2 rpl neighbor
!
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying ERPS](#).

8.6.2 Configuring an ERPS Multi-ring

The ERPSv2 provides support for multi-rings topologies using sub-rings. The sub-ring uses a virtual channel through of a link of the major ring or a link of sub-ring to "close" the topology. It is possible to create complex scenarios with multiple sub-rings interconnected.



A RPL Owner on each ring must be configured to correctly block the major ring and the created subrings.



The ring interconnection equipment must be configured in the sub-ring instance as an interconnection node using the **node interconnection** command and must indicate which ERPS ring it is major ring using the **parent-ring** command.



The virtual channel must have its own control VLAN.

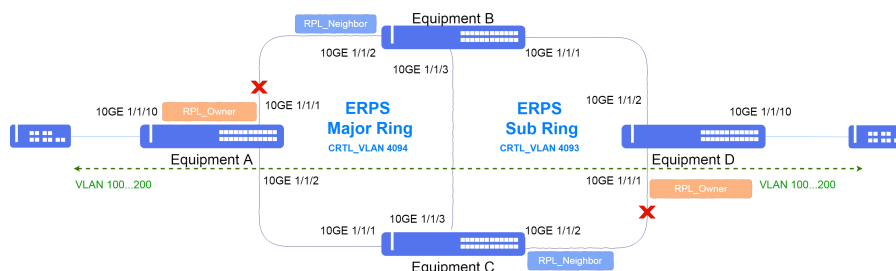


In this DmOS version, the control VLAN of virtual channel can not be the same as the control VLAN of sub-ring.



In this DmOS version, it is mandatory to map the control VLAN of sub-rings on the interconnection equipment that creates the virtual channel by using the **vlan-mapping** command.

The scenario below will be used to illustrate ERPS configuration using multi-ring topology.



Implementation of ERPS Multi-Ring

Suppose the user wants to protect the VLAN group 100 to 200 using a Major ring through of control VLAN 4094, and a sub-ring using the control VLAN 4093. A virtual channel will be configured using the control VLAN 4000 and it will be protected in the major ring.

- **Equipment A:** The VLAN group 100 to 200 will be protected for data traffic. The ten-gigabit-ethernet-1/1/10 interface will be used as access interface. The ten-gigabit-ethernet-1/1/1 and ten-gigabit-ethernet-1/1/2 interfaces will be used in the major ring and the ten-gigabit-ethernet-1/1/1 interface will be used as RPL Owner.
- **Equipment B:** The VLAN group 100 to 200 will be protected for data traffic. The ten-gigabit-ethernet-1/1/2 and ten-gigabit-ethernet-1/1/3 will be used in the major ring and ten-gigabit-ethernet-1/1/1 interface will be used in sub-ring. The virtual channel will be a protected VLAN in major ring, that is, between A, B and C equipment.

- Equipment C:** The VLAN group 100 to 200 will be protected for data traffic. The ten-gigabit-ethernet-1/1/1 and ten-gigabit-ethernet-1/1/3 will be used in the major ring and ten-gigabit-ethernet-1/1/2 interface will be used in sub-ring. The virtual channel will be a protected VLAN in major ring, that is, between A, B and C equipment.
- Equipment D:** The VLAN group 100 to 200 will be protected for data traffic. The ten-gigabit-ethernet-1/1/10 interface will be used as access interface. The ten-gigabit-ethernet-1/1/1 will be used on the major ring and ten-gigabit-ethernet-1/1/2 interface will be used on the sub-ring. The ten-gigabit-ethernet-1/1/3 interface will be used as a virtual channel connecting the two rings. The ten-gigabit-ethernet-1/1/1 and ten-gigabit-ethernet-1/1/2 interfaces will be used on the sub-ring and the ten-gigabit-ethernet-1/1/1 interface will be used as RPL Owner.

```

!Equipment A
config
dot1q
vlan 100-200
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/10 tagged
    !
vlan 4000
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    !
vlan 4094
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    !
!
!
erps
ring ERPS-MAJOR
    ring-id 1
    control-vlan 4094
    protected-vlans 100-200,4000
    port0
        interface ten-gigabit-ethernet-1/1/1 rpl-role owner
        !
    port1
        interface ten-gigabit-ethernet-1/1/2
        !
    !
!
commit

```

```
!Equipment B
config
dot1q
vlan 100-200
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
vlan 4000
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
vlan 4093
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
vlan 4094
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
!
vlan-mapping
    interface ten-gigabit-ethernet-1/1/1
    egress
        rule ERPS-rule-1
        match vlan vlan-id 4000
        action replace vlan vlan-id 4093 pcp copy
    !
    !
    interface ten-gigabit-ethernet-1/1/2
    egress
```

```

    rule ERPS-rule-2
    match vlan vlan-id 4093
    action replace vlan vlan-id 4000 pcp copy
    !
    !
    !
interface ten-gigabit-ethernet-1/1/3
egress
    rule ERPS-rule-3
    match vlan vlan-id 4093
    action replace vlan vlan-id 4000 pcp copy
    !
    !
    !
!
!
erps
ring ERPS-MAJOR
ring-id 1
control-vlan 4094
protected-vlans 100-200,4000
port0
    interface ten-gigabit-ethernet-1/1/2 rpl-role neighbor
    !
port1
    interface ten-gigabit-ethernet-1/1/3
    !
!
ring ERPS-SUBRING
ring-id 1
control-vlan 4093
protected-vlans 100-200,4000
type sub-ring
node interconnection
parent-ring ERPS-MAJOR
port0
    interface ten-gigabit-ethernet-1/1/1
    !
port1
    virtual-channel control-vlan 4000
    !
commit

```

```

!Equipment C
config
dot1q
vlan 100-200
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
vlan 4000
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
vlan 4093
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
vlan 4094
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/3 tagged
    !
!
!
vlan-mapping
interface ten-gigabit-ethernet-1/1/2
egress
    rule ERPS-rule-1
    match vlan vlan-id 4000
    action replace vlan vlan-id 4093 pcp copy
    !
    !
!
!
interface ten-gigabit-ethernet-1/1/1
egress
    rule ERPS-rule-2
    match vlan vlan-id 4093
    action replace vlan vlan-id 4000 pcp copy
    !
    !
!
!
interface ten-gigabit-ethernet-1/1/3
egress
    rule ERPS-rule-3
    match vlan vlan-id 4093
    action replace vlan vlan-id 4000 pcp copy
    !
    !
!

```

```
!
erps
ring ERPS-MAJOR
  ring-id 1
  control-vlan 4094
  protected-vlans 100-200,4000
  port0
    interface ten-gigabit-ethernet-1/1/1
  !
  port1
    interface ten-gigabit-ethernet-1/1/3
  !
!
ring ERPS-SUBRING
  ring-id 1
  control-vlan 4093
  protected-vlans 100-200,4000
  type sub-ring
  node interconnection
  parent-ring ERPS-MAJOR
  port0
    interface ten-gigabit-ethernet-1/1/2 neighbor
  !
  port1
    virtual-channel control-vlan 4000
  !
!
!
commit
```

```

!Equipment D
config
dot1q
vlan 100-200
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
    interface ten-gigabit-ethernet-1/1/10 tagged
!
vlan 4093
    interface ten-gigabit-ethernet-1/1/1 tagged
    interface ten-gigabit-ethernet-1/1/2 tagged
!
!
!
erps
ring ERPS-SUBRING
    ring-id 1
    control-vlan 4093
    protected-vlans 100-200
    type sub-ring
    port0
        interface ten-gigabit-ethernet-1/1/2
    !
    port1
        interface ten-gigabit-ethernet-1/1/1 rpl owner
    !
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying ERPS](#).

8.6.3 Verifying ERPS

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show erps  
show erps brief
```

8.7 L2CP Configuration

The L2CP (Layer 2 Control Protocol) protocol is used to provide LAN-to-LAN service in a transparent manner through a network in such a way that the central equipment of the network do not process the PDUs.



The L2CP is supported only in Switch platforms.



The DmOS supports the **L2CP** in the **extended** mode and for **CDP, Dot1x, EAPS, ERPS, GVRP, LACP, LLDP, Marker, OAM, PAgP, PVST, STP, UDLD and VTP** protocols.



The L2CP extended mode is proprietary, for interoperability use the L2CP by protocol.



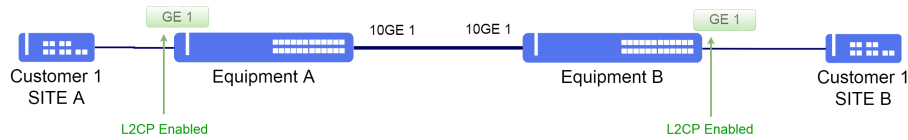
The L2CP configured by protocol has precedence over extended mode.

8.7.1 Configuring L2CP in extended mode

The L2CP extended mode can tunnel the following MAC address:

PDU Type	MAC Address
IEEE	01:80:C2:00:00:0X e 01:80:C2:00:00:2X
EAPS	00:E0:2B:00:00:04
RRPP	00:0F:E2:07:82:XX
Cisco Protocols	01:00:0C:CC:XX:XX e 01:00:0C:CD:XX:XX

The scenario below will be used to illustrate the config of the L2CP protocol in extended mode. This configuration is used when two private networks of the same customer are connected by ISP network and this customer requires that L2 protocols run like one network between the private networks.



Implementation of L2CP

Considering that the user would like to use the following configs in both equipment:

- VLAN ID 100 for Customer 1 with gigabit-ethernet-1/1/1 interface as access interface and ten-gigabit-ethernet-1/1/1 interface as uplink interface. The L2CP is activated in the access interface.

```
config
dot1q
vlan 100
interface ten-gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/1 untagged
!
!
switchport
interface gigabit-ethernet-1/1/1
native-vlan
vlan-id 100
!
!
layer2-control-protocol
interface gigabit-ethernet-1/1/1
extended action tunnel
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2CP](#).

8.7.2 Configuring L2CP by Specific Protocol

Using the [Configuring L2CP in extended mode](#) scenario it is possible to change L2CP configuration to use L2CP by Protocol instead L2CP extended mode in both equipment.

```
config
layer2-control-protocol interface gigabit-ethernet-1/1/1 cdp action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 dot1x action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 eaps action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 erps action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 gvrp action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 lacp action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 lldp action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 marker action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 oam action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 pagp action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 pvst action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 stp action tunnel
layer2-control-protocol interface gigabit-ethernet-1/1/1 udld action tunnel
```

```
layer2-control-protocol interface gigabit-ethernet-1/1/1 vtp action tunnel
commit
```

In case of interoperability with other vendor which use destination MAC (01:00:0C:CD:CD:D0) in L2CP feature use the command below.

```
config
layer2-control-protocol tunnel-mac interop
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2CP](#).

8.7.3 Configuring L2CP Extended PDU transparency

It is possible enable L2CP Extended PDU transparency using the command below. This feature is used to forward PDUs from one port to another without change any PDU information, because of this all equipment in path needs support PDU transparency.



It is necessary configure the PDU transparency in all equipment in the path.

Using the [Configuring L2CP in extended mode](#) scenario it is possible to change L2CP configuration to use PDU transparency instead L2CP tunneling.

```
config
layer2-control-protocol interface gigabit-ethernet-1/1/1 extended action forward
layer2-control-protocol interface ten-gigabit-ethernet-1/1/1 extended action forward
```



The available commands for troubleshooting can be found in the topic [Verifying L2CP](#).

8.7.4 Verifying L2CP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show running-config layer2-control-protocol
debug enable l2cp-tunneling
```

8.7.5 PDUs default behavior in OLT platforms



On DM461x platforms that support GPON technology the transparency of L2 PDUs for TLS service (service vlan type TLS) is enabled without the possibility of changing this behavior. For services 1: 1 and N: 1 (service vlan type 1: 1 or n: 1), the transparency of L2 PDUs is disabled without the possibility of changing this behavior.

The table below summarizes the transparency behavior of L2 PDUs on DM461x platforms for each type of GPON service.

PDU Type	MAC Address	TLS Service	Service N:1 or 1:1
IEEE	01:80:C2:00:00:0X and 01:80:C2:00:00:2X	Forward	Drop
EAPS	00:E0:2B:00:00:04	Forward	Forward
ERPS	01:19:A7:<ring-id>	Forward	Forward
RRPP	00:0F:E2:07:82:XX	Forward	Forward
Cisco Protocols	01:00:0C:CC:XX:XX and 01:00:0C:CD:XX:XX	Forward	Forward

8.7.6 PDUs default behavior in Switch platforms

In Switch platforms actions **tunnel** and **forward** are supported. The table below summarizes the default tunneling behavior of PDUs if L2CP is not configured.

PDU Type	MAC Address	Default Action
IEEE	01:80:C2:00:00:0X and 01:80:C2:00:00:2X	Drop
EAPS	00:E0:2B:00:00:04	Forward
ERPS	01:19:A7:<ring-id>	Forward
RRPP	00:0F:E2:07:82:XX	Forward
Cisco Protocols	01:00:0C:CC:XX:XX and 01:00:0C:CD:XX:XX	Forward

8.8 Loopback Detection Configuration

Loopback Detection (LBD) detects a loop sending periodically PDUs in an interface and check if PDUs are received in the same interface. If the loop is detected, the following actions will occur:

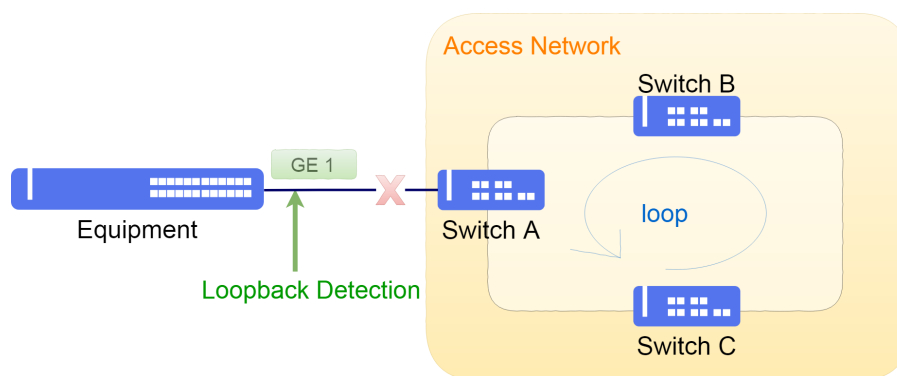
- The interface is blocked.
- An alarm is activated.
- A log is recorded.
- A SNMP trap is sent to the configured SNMP server.



Loopback Detection configuration is not supported on LAG member ports.

8.8.1 Configuring Loopback Detection for access network

In the scenario below, as shown, a loop occurs in access network connected to the equipment. Packets sent from an interface are sent back to this interface.



Scenario with loop

The next steps will demonstrate how to enable loopback detection on an interface.

```
config
loopback-detection
 interface gigabit-ethernet-1/1/1
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Loopback Detection](#).

8.8.2 Verifying Loopback Detection

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
debug enable loopback-detection
show alarm
```

8.9 DHCP Relay L2 Configuration

The DHCP Relay L2 executes snooping of DHCP packs for security and subscribers' management purposes, maintaining control of the IP assigned by a reliable DHCP server to the non-reliable network devices. The DHCP option 82 attached by the retransmission agent can be used to maintain tracking of the user and provide a network config based on the network clients' locations.



The standard config has the DHCP as deactivated.



Currently, the DHCP Relay functionality is available only in the DM461x platforms with support to the GPON technology.

To enable the DHCP Relay in VLAN 20 the user would perform the following procedure:

```
config
dhcp l2-relay vlan 20
commit
```

8.9.1 Configuring Circuit ID

It is possible to configure the Circuit ID information in option 82 of the DHCP relay, changing the circuit pattern information to another pattern that facilitates the network management.

The default Circuit ID content in OLT DATACOM has the following format:

```
<hostname> eth <oltSlot>/<panelPort>/<onuId>/<onuSlot>/<gemPort>:<svlan>
```

Example: DM4615 eth 1/5/1/0/5:1024



The Circuit ID setting acts globally and will be applied to all DHCPv4 packets from the VLANs configured in DHCP Relay L2. DHCPv6 packets are forwarded transparently without modification to the Circuit ID.

In the following example, the DHCP frames will have the Circuit ID configured to send the ONU serial information, fixed text (DATACOM), GEM Port and SVLAN generating the content in the pattern **DTCM12345678:DATACOM:2:1234**.

```
config
dhcp l2-relay circuit-id format <onuSerial>:DATACOM:<gemPort>:<svlan>
commit
```

Some fields are required to form the Circuit ID such as **<gemPort>** and **<onuSerial>**, with the field **<onuSerial>** being replaced by **<oltSlot> <panelPort> <onuld>**.

In the following example, setting the Circuit ID by changing the mandatory **<onuSerial>** field to **<oltSlot> <panelPort> <onuld>**:

```
config
dhcp l2-relay circuit-id format <oltSlot>/<panelPort>/<onuId>:DATACOM:<gemPort>:<svlan>
commit
```

The above example will generate the content in the pattern **1/2/3:DATACOM:2:1234**.



The Circuit ID generated cannot exceed 63 characters. If the package information exceeds this size, the content will be truncated.



The available commands for troubleshooting can be found in the topic [Verifying DHCP Relay](#).

8.9.2 Configuring filter by MAC/IP

OLT DATACOM has a limit of entries in the anti-ip-spoofing feature to control data traffic based on the IP address, when using dhcp l2-relay and service vlan of type n:1. When the anti-ip-spoofing entries limit is reached, data traffic is blocked for excess entries.



Refer to the DmOS Datasheet to check maximum values by platform.

To have no limit in this scenario, it is possible to set the filter mode to **filter-by mac**. In this way the DHCP entries will not be limited by the anti-ip-spoofing feature. In MAC mode filtering, traffic is allowed via MAC address.



DmOS uses IP filtering by default.

In the following example, the filter setting is changed to MAC on dhcp l2-relay for VLAN 1004:

```
config
dhcp l2-relay vlan 1004 filter-by mac
commit
```



It is recommended to set a low limit of packets per second in the CPU queue that handles DHCP packets in scenarios with a high number of concurrent DHCP requests, through the command **cpu-dos-protect protocols dhcp max-pps <PPS>**.

For more details about *CPU DoS Protect* feature, please consult topic [Configuring CPU DoS Protect](#).



The available commands for troubleshooting can be found in the topic [Verifying DHCP Relay](#).

8.9.3 Verifying DHCP Relay

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show allowed-ip entry-type dhcp
show log component ip_spoofing
show mac-address-table vlan <VLAN_ID>
```

9 IP Services

In this chapter, it is shown basic L3 interface configuration, with IPv4 and IPv6 addressing. It also contains IP services configuration.

This chapter contains the following sections:

- IP Addresses Configuration
- IPv6 SLAAC Configuration
- L3 DHCP Relay Configuration

9.1 IP Addresses Configuration

The user can configure IPv4 and IPv6 addresses manually in management, loopback and L3 interfaces.

9.1.1 Configuring IPv4 addresses

The following steps show how to configure the IPv4 address **10.10.0.1/30** in L3 interface (VLAN 2).

```
config
dot1q
vlan 2
interface gigabit-ethernet-1/1/1 untagged
!
!
switchport
interface gigabit-ethernet-1/1/1
native-vlan
vlan-id 2
!
!
interface l3 VLAN2
ipv4 address 10.10.0.1/30
lower-layer-if vlan 2
commit
```



The available commands for troubleshooting can be found in the topic [Verifying IP Address](#).

9.1.2 Configuring IPv6 addresses

The following steps show how to configure the IPv4 address **2001::a0a:1/126** in L3 interface (VLAN 2).

```
config
dot1q
vlan 2
interface gigabit-ethernet-1/1/1 untagged
!
!
switchport
interface gigabit-ethernet-1/1/1
native-vlan
vlan-id 2
!
!
```

```
!
interface l3 VLAN2
  lower-layer-if vlan 2
  ipv6 enable
  ipv6 address 2001::a0a:1/126
commit
```



The available commands for troubleshooting can be found in the topic [Verifying IP Address](#).

9.1.3 Verifying IP Address

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show ip interface brief
show ipv6 interface brief
```

9.1.4 MTU configuration in L3 interfaces

It is also possible to configure the MTU in L3 interfaces. In this case, the MTU restriction will only be applied to packets in the control plane, or packets directed to the equipment CPU.

```
config
interface l3 VLAN2
  ip-mtu 2000
commit
```



The default MTU value is 1500 bytes and MTU can be configured between 68 bytes and 9198 bytes.



There are no troubleshooting commands for this functionality.

9.2 IPv6 SLAAC Configuration

SLAAC (IPv6 Stateless Address Autoconfiguration) allow equipment interfaces to provide IPv6 prefix information to hosts connected to these interfaces without using a DHCPv6 server or manually configuring a IPv6 address.

The SLAAC operation is based on the exchange of messages **RA (Router Advertisement)** and **RS (Router Solicitation)** allowing clients to use the information of those messages to auto-configure their own IPv6 addresses.



SLAAC is available for L3 interfaces and the MGMT interface.



By default, the **lifetime** of RA messages is **1800** seconds. If the interface should not be used as default route, the lifetime must be set to 0. The **lifetime** is not available for MGMT interface because it can not be used as default route.

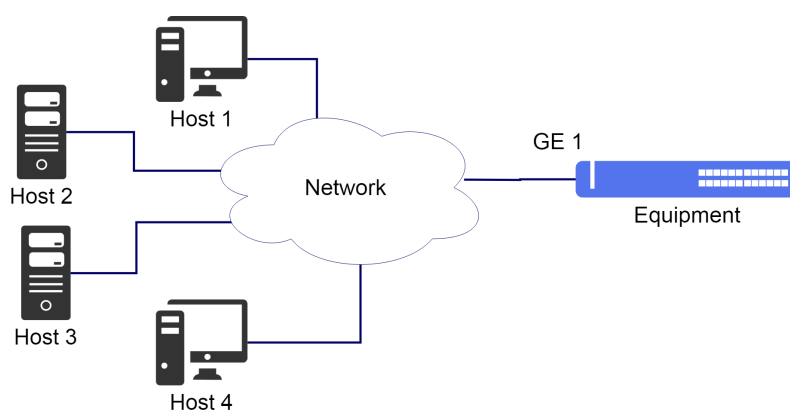


By default, once the command "ipv6 enable" is configured into an interface, RA messages will start to be sent to all clients in the broadcast domain. The prefix information will only be added into an RA message after the configuration of "ipv6 address <x:x:x::y/64>" or "ipv6 nd ra prefix <x:x:x::/64>".



For SLAAC to work, it is necessário that the IPv6 has /64 prefix.

The scenario below will be used to illustrate the configuration of SLAAC.



Implementation of SLAAC

Considering that the user would like to use the network prefix **2222::/64** for a specific network. The SLAAC can be enabled

in order to propagate this network prefix via **SLAAC-1** L3 interface to all hosts connected to the broadcast domain. The next steps will indicate how to execute this configuration.

```
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1 untagged
!
!
switchport
interface gigabit-ethernet-1/1/1
native-vlan
vlan-id 10
!
!
interface l3 SLAAC-1
lower-layer-if vlan 10
ipv6 enable
ipv6 nd ra prefix 2222::/64
commit
```



The available commands for troubleshooting can be found in the topic [Verifying IPv6 SLAAC](#).

9.2.1 Verifying IPv6 SLAAC

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
debug enable ipv6-nd-rx
debug enable ipv6-nd-tx
```

9.3 L3 DHCP Relay Configuration

It is possible to perform DHCP messages forwarding between two distinct broadcast domains using the DHCP Relay agent. The broadcast messages generated by the client (DHCP Discovery and DHCP request) are received by the DHCP relay switch, those messages are converted from broadcast to unicast messages and then forwarded to the DHCP server. The operation of DHCP relay agent depends on the local network that was configured with a VLAN and IP address, as the network where the DHCP server is located. This configuration can be performed as indicated in [IP Addresses Configuration](#).

DHCP has some additional configurations that provide more security in the allocation of addresses. These options can be configured globally on the DHCP relay or separately by interface.

- **information option:** Inserts an Option 82 to the frame referred to the inbound network interface;
- **information trust-all:** Accepts frames that arrive with Option 82 already inserted;

- **information policy keep:** Keeps the Option 82 in the frames when those arrive with this field.



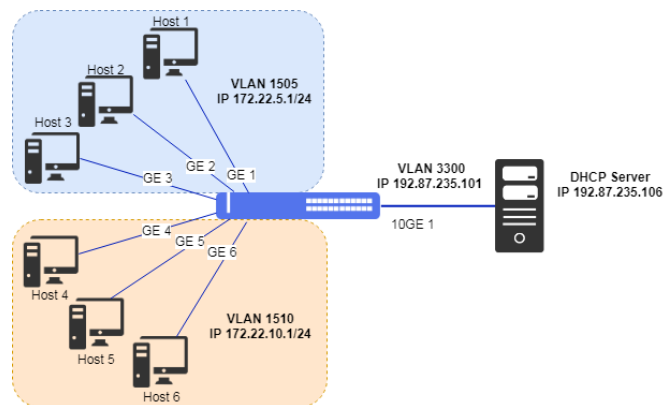
If Option 82 is configured globally and by interface, the configuration by interface has preference.



For more details about other Option 82 parameters, check the **Command Reference**.

9.3.1 Configuring the L3 DHCP Relay

The following topology presents an application of the DHCP relay service:



L3 DHCP relay scenario

The following steps show how to configure the L3 DHCP relay service to the clients of VLAN 1505 and VLAN 1510 destined to the DHCP server in VLAN 3300:

```
config
dot1q
vlan 1505
interface gigabit-ethernet-1/1/1
untagged
interface gigabit-ethernet-1/1/2
untagged
interface gigabit-ethernet-1/1/3
untagged
!
vlan 1510
interface gigabit-ethernet-1/1/4
untagged
interface gigabit-ethernet-1/1/5
untagged
interface gigabit-ethernet-1/1/6
untagged
!
vlan 3300
interface ten-gigabit-ethernet-1/1/1
!
!
switchport interface gigabit-ethernet-1/1/1 native-vlan vlan-id 1505
```



```

!
switchport interface gigabit-ethernet-1/1/2 native-vlan vlan-id 1505
!
switchport interface gigabit-ethernet-1/1/3 native-vlan vlan-id 1505
!
switchport interface gigabit-ethernet-1/1/4 native-vlan vlan-id 1510
!
switchport interface gigabit-ethernet-1/1/5 native-vlan vlan-id 1510
!
switchport interface gigabit-ethernet-1/1/6 native-vlan vlan-id 1510
!
interface l3 vl-1505
 lower-layer-if vlan 1505
 ipv4 address 172.22.5.1/24
!
interface l3 vl-1510
 lower-layer-if vlan 1510
 ipv4 address 172.22.10.1/24
!
interface l3 vl-3300
 lower-layer-if vlan 3300
 ipv4 address 192.87.235.101/24
!
!
dhcp relay DHCP-RELAY-L3
 server ipv4 192.87.235.106
 interface l3-vl-1505
!
 interface l3-vl-1510
!
!
commit

```

9.3.2 Configuring the DHCP Option globally

Based on the previous example, the DHCP Option will be added globally.

In this example, the DHCP frames that arrive with no Option 82 will be marked before forwarding to the server. The frames that arrive with Option 82 already marked will be forwarded without changes in this field. The Option 82 inserted by the switch is composed by Circuit ID and Remote ID fields, example: **10ge- 1/1/1:1505**.

```

config
 dhcp relay DHCP-RELAY-L3
  information option
  information trust-all
  information policy keep
  server ipv4 192.87.235.106
!
 interface l3-vl-1505
!
 interface l3-vl-1510
!
!
commit

```

9.3.3 Configuring the DHCP Option by interface

Based on the initial example, DHCP Option will be added on a specific interface.

DHCP relay service configuration by interface allows different configuration of Option 82 in an interface instead of global Option 82 configuration for the DHCP relay instance.

The following steps show how to configure the DHCP relay service to insert Option 82 only for client in interface gigabit-ethernet-1/1/1, the others interfaces will not have Option 82 inserted. The DHCP client packets are forwarded to the DHCP server with IPv4 address 192.87.235.106:

```
config
dhcp relay DHCP-RELAY-L3
server ipv4 192.87.235.106
interface l3-vl-1505
!
interface l3-vl-1510
!
if-option gigabit-ethernet-1/1/1
information option
!
commit
```

10 Routing

Routing is the process to forward packs to their destination using network addresses. Routing is executed by devices capable to exchange information required to create tables containing path information to reach a destination, using specific protocols or inputs assigned manually.

The dynamic routing protocols, such as the OSPF, gather the required information from the neighboring devices in order to create its routing table, used to define to which location the traffic will be sent.

As alternatives to the dynamic methods, static routes exist. The static routes are recommended in routers that have few networks and less paths for destination.

The information received through the routing protocols are added in a table called RIB (Routing Information Base) that is the base for calculation of definition of best path. The result of the route calculation is the FIB (Forwarding Information Base) that contains information that the devices use to route the traffic.

This chapter contains the following sections:

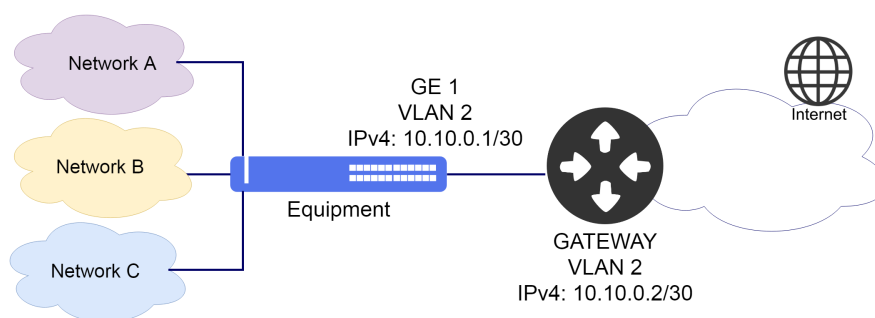
- [Static Routing Configuration](#)
- [Black Hole Route Configuration](#)
- [VLAN Routing Configuration](#)
- [VRF Configuration](#)
- [PBR Configuration](#)
- [OSPFv2 Configuration](#)
- [OSPFv3 Configuration](#)
- [BGP Configuration](#)
- [VRRP Configuration](#)
- [BFD Configuration](#)

10.1 Static Routing Configuration

The static routing has as objective to output packs between different networks with the config of the routes in a manual manner by the network administrators.

10.1.1 Configuring a Default Static Route

The scenario below will be used to illustrate the config of the static routing.



Implementation of static routing

Considering that the user would like that the entire traffic be forwarded through the L3 (VLAN 2) interface with IPv4 **10.10.0.1/30** address. In his case, a default route should be set. The next steps will indicate how to execute these configs.

```
config
dot1q
vlan 2
  interface gigabit-ethernet-1/1/1 tagged
  !
  !
!
interface l3 DEFAULT_ROUTE-VLAN2
  ipv4 address 10.10.0.1/30
  lower-layer-if vlan 2
  !
!
router static address-family ipv4 0.0.0.0/0 next-hop 10.10.0.2
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Static Routes](#).

10.1.2 Verifying Static Routes

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

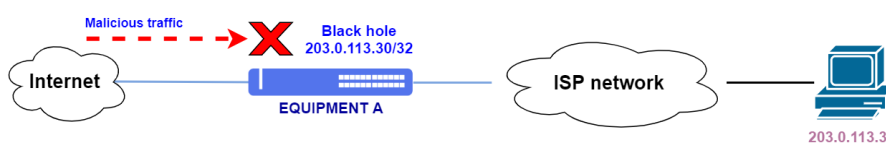
```
show ip route
show ip route static
show ip rib
show ip rib static
show ip interface brief
```

10.2 Black hole route configuration

All traffic forwarded to a *black hole* route is discarded. In an attack situation where a large volume of traffic is being forwarded to a specific destination, a black hole route can be configured to discard that traffic and avoid links in the ISP (Internet Service Provider) network to be saturated.

10.2.1 IPv4 black hole route configuration

In the following example, address **203.0.113.30/32** is being attacked, which can lead to ISP links being saturated and affect other customers. To avoid that, a black hole route has been configured and all traffic to 203.0.113.30 is discarded as soon as it gets in the ISP network.

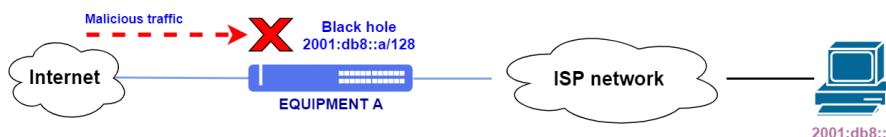


IPv4 black hole route

```
config
router static
address-family ipv4
  203.0.113.30/32 blackhole
!
commit
```

10.2.2 IPv6 black hole route configuration

In the following example, address **2001:db8::a/128** is being attacked, which can lead to ISP links being saturated and affect other customers. To avoid that, a black hole route has been configured and all traffic to 2001:db8::a is discarded as soon as it gets in the ISP network.



IPv6 black hole route

```
config
router static
address-family ipv6
  2001:db8::a/128 blackhole
!
commit
```

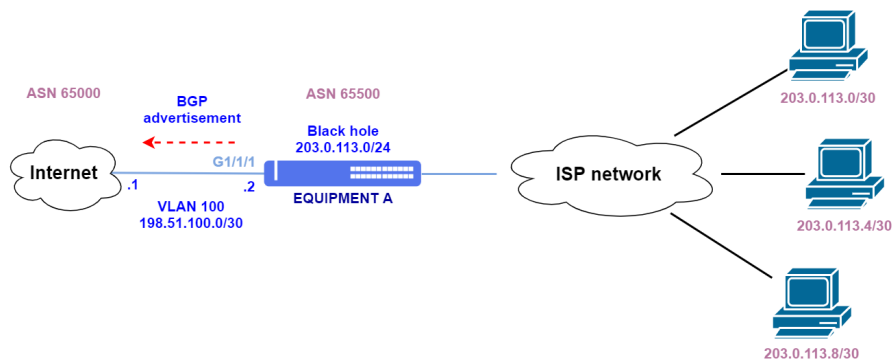
10.2.3 Summarization with black hole routes

In the following scenario, the ISP has a 203.0.113.0/24 network distributed in several /30 routes for its customers. The /30 routes can not be advertised via BGP to the Internet transit link because only /24 or less specific routes are accepted by the neighbor.

It is possible to summarize those routes and advertise only the /24 supernet containing the /30 networks. In the following example, a black hole static route is configured and redistributed to the BGP neighbor.

When the route is advertised through eBGP, its next-hop is changed to the address in the **update-source-address** parameter in the BGP neighbor configuration. The black hole route becomes a route with a valid next hop when it is advertised.

Packets destined to customer addresses will not be discarded by the black hole route. The more specific /30 routes will be used instead, since they have priority over less specific routes.



Summarization with black hole routes

```
!Equipment A
config
dot1q
vlan 100
  interface gigabit-ethernet-1/1/1
  !
!
interface l3 VLAN100
  lower-layer-if vlan 100
  ipv4 address 198.51.100.2/30
!
interface loopback 0
  ipv4 address 203.0.113.254/32
!
router bgp 65500
  router-id 203.0.113.254
  address-family ipv4 unicast
  !
  redistribute static address-family ipv4
    match-address 203.0.113.0/24
  !
  neighbor 198.51.100.1
    update-source-address 198.51.100.2
    remote-as 65000
    address-family ipv4 unicast
  !
!
router static
  address-family ipv4
    203.0.113.0/24 blackhole
  !
!
```

```
commit
```



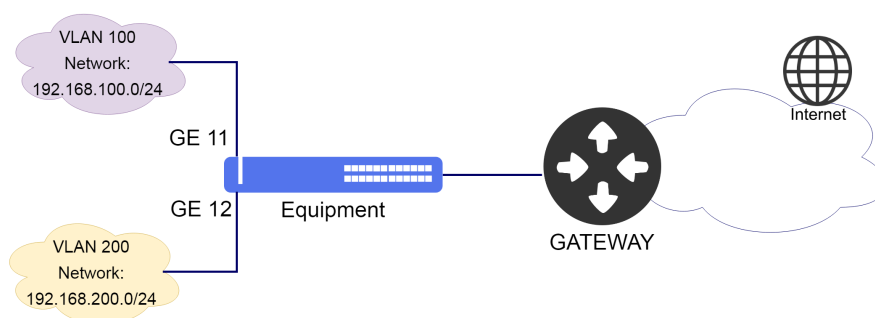
Route summarization through black hole static routes redistribution can also be done with IPv6 routes.

10.3 VLAN Routing Configuration

By default, different VLANs do not communicate one with the other, since they are in exclusive broadcast domains. In order to have a communication between two VLANs, it is necessary to use a router or a way of routing in the equipment itself. Routing between VLANs allows this communication through config of L3 interfaces associated to the required VLANs. The network associated to L3 interface is inserted in the routing table and can be accessed by other networks.

10.3.1 Configuring a Basic Routing between VLANs

The scenario below will be used to illustrate the config of routing between VLANs.



Implementation of routing between VLANs

Considering that the user would like to set the routing between VLAN 100 that has a L3 interface with 192.168.100.1/24 address and VLAN 200 that has a L3 interface with 192.168.200.1/24 address. The next steps will indicate how to perform these configs.



It is possible configure secondary IPv4 address on L3 interfaces. Secondary IPv6 address is not supported.

```
config
dot1q
vlan 100
  interface gigabit-ethernet-1/1/11 tagged
!
vlan 200
  interface gigabit-ethernet-1/1/12 tagged
```

```
!
!
interface l3 L3-VLAN100
  ipv4 address 192.168.100.1/24
  lower-layer-if vlan 100
!
!
interface l3 L3-VLAN200
  ipv4 address 192.168.200.1/24
  lower-layer-if vlan 200
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Routes](#).

10.3.2 Verifying Routes

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show ip route
show ip route connected
show ip interface brief
```

10.4 VRF Configuration

VRF (Virtual Routing and Forwarding) is a feature that allows different routing instances in the same equipment.

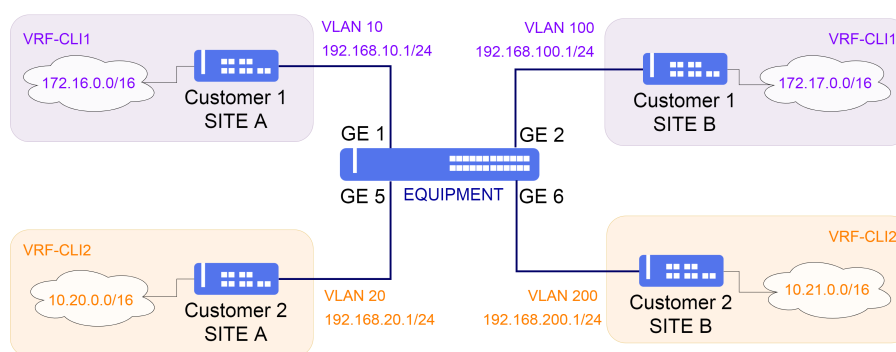
By default, DmOS has two routing tables, global and management. It is possible to create other tables configuring other VRF instances.



To configure the **VRF mgmt** (VRF used exclusively for equipment management), it is necessary to configure the mgmt interface and a default route in the mgmt VRF. By default, the mgmt VRF is already created in DmOS.

10.4.1 Configuring a IPV4 VRF Lite

VRF lite is a basic version of VRF and does not support MPLS signaling. The scenario below will be used to illustrate the config of the VRF Lite.



Implementation of VRF Lite

There must not be communication between clients 1 and 2. Therefore, two VRFs will be configured to isolate the routing tables and traffic between them. The following specifications will be used:

- **VRF-CLI1:**
Interface in VLAN 10 with IPv4 address 192.168.10.1/24
Interface in VLAN 100 with IPv4 address 192.168.100.1/24
- **VRF-CLI2:**
Interface in VLAN 20 with IPv4 address 192.168.20.1/24
Interface in VLAN 200 with IPv4 address 192.168.200.1/24

To create VRFs VRF-CLI1 and VRF-CLI2, just follow the configurations below.

```
config
vrf VRF-CLI1
!
description CLIENT1
!
vrf VRF-CLI2
!
description CLIENT2
!
commit
```

Then, interfaces and routes are configured and assigned to these VRFs.

```
config
dot1q
vlan 10
!
interface gigabit-ethernet-1/1/1 tagged
!
vlan 20
!
interface gigabit-ethernet-1/1/5 tagged
!
vlan 100
!
interface gigabit-ethernet-1/1/2 tagged
!
vlan 200
!
interface gigabit-ethernet-1/1/6 tagged
!
!
vrf VRF-CLI1
!
vrf VRF-CLI2
!
interface l3 CLI1-VLAN10
vrf VRF-CLI1
ipv4 address 192.168.10.1/24
lower-layer-if vlan 10
```

```

!
interface l3 CLI1-VLAN100
 vrf VRF-CLI1
 ipv4 address 192.168.100.1/24
 lower-layer-if vlan 100
!
interface l3 CLI2-VLAN20
 vrf VRF-CLI2
 ipv4 address 192.168.20.1/24
 lower-layer-if vlan 20
!
interface l3 CLI2-VLAN200
 vrf VRF-CLI2
 ipv4 address 192.168.200.1/24
 lower-layer-if vlan 200
!
router static
 vrf VRF-CLI1
  address-family ipv4
    172.16.0.0/16 next-hop 192.168.10.2
    172.17.0.0/16 next-hop 192.168.100.2
  !
!
 vrf VRF-CLI2
  address-family ipv4
    10.20.0.0/16 next-hop 192.168.20.2
    10.21.0.0/16 next-hop 192.168.200.2
  !
!
commit

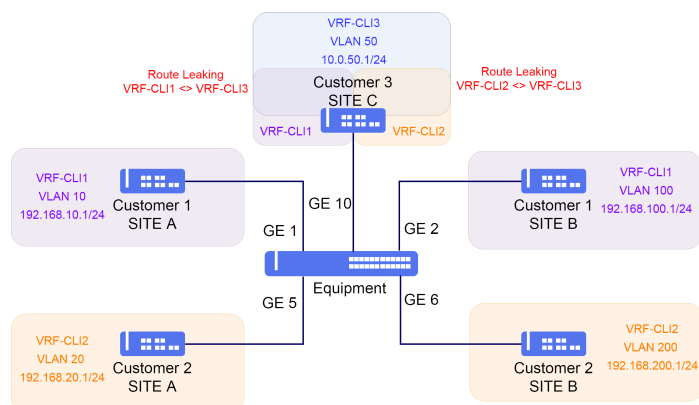
```



The available commands for troubleshooting can be found in the topic [Verifying VRFs](#).

10.4.2 Enabling Route Leaking between IPv4 VRFs

The previous scenario will be used as base. A third client is included.



Route leaking between VRFs

Client 3 has the following specifications:

- **VRF-CLI3:**
Interface in VLAN 50 with IPv4 address 10.1.50.1/24

Since all the clients are in different VRFs, there is no communication between them. Considering there is a requirement that client 1 and 2 must access client 3, route leaking between VRFs must be used.

Each VRF must have a unique identifier called Route Distinguisher (RD). The RD is used to identify to which VRF each route belongs, allowing as well IP address overlapping between VRFs. The following RD's will be used:

- **VRF-CLI1:** rd 1:10
- **VRF-CLI2:** rd 2:20
- **VRF-CLI3:** rd 3:50

To enable the leaking, route-targets (RT) are used. Just like RD, RTs are identifiers added to the routes so the router knows which routes to be inserted in which VRFs. RTs can have the same format as the RD. Routes exported with a specific RT will be imported in VRFs that have the same RT configured as import.

It will be configured leaking between VRF-CLI1 and VRF-CLI3 and between VRF-CLI2 and VRF-CLI3, so there will be possible to Client 3 communicate with Client 1 and Client 2 as well. There will not be possible to Client 1 to communicate to Client 2 because they are not configured to import routes between them.



The route leaking between VRFs occurs with the redistribution of static routes and with the redistribution of directly connected routes.

```
config
dot1q
vlan 50
interface gigabit-ethernet-1/1/10 tagged
!
!
interface l3 CLI3-VLAN50
vrf VRF-CLI3
ipv4 address 10.1.50.1/24
lower-layer-if vlan 50
!
vrf VRF-CLI1
rd 1:10
route-target import 3:50
!
route-target import 1:10
!
route-target export 1:10
!
!
vrf VRF-CLI2
rd 2:20
route-target import 3:50
!
route-target import 2:20
!
route-target export 2:20
!
!
vrf VRF-CLI3
rd 3:50
route-target import 1:10
!
route-target import 2:20
!
route-target import 3:50
!
route-target export 3:50
!
!
router static
vrf VRF-CLI3
address-family ipv4
0.0.0.0/0 next-hop 10.1.50.2
!
!
```

```

!
router bgp 65500
 address-family ipv4 unicast
!
vrf VRF-CLI1
 address-family ipv4 unicast
 redistribute static
!
 exit-address-family
!
vrf VRF-CLI2
 address-family ipv4 unicast
 redistribute static
!
 exit-address-family
!
vrf VRF-CLI3
 address-family ipv4 unicast
 redistribute static
!
 exit-address-family
!
!
commit

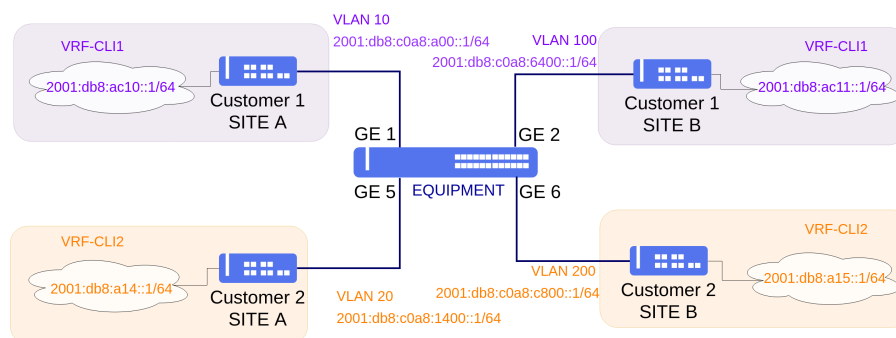
```



The available commands for troubleshooting can be found in the topic [Verifying VRFs](#).

10.4.3 Configuring a IPV6 VRF Lite

VRF lite is a basic version of VRF and does not support MPLS signaling. The scenario below will be used to illustrate the config of the VRF Lite.



Implementation of VRF Lite

There must not be communication between clients 1 and 2. Therefore, two VRFs will be configured to isolate the routing tables and traffic between them. The following specifications will be used:

- **VRF-CLI1:**
Interface in VLAN 10 with IPv6 address 2001:db8:c0a8:a00::1/64
Interface in VLAN 100 with IPv6 address 2001:db8:c0a8:6400::1/64
- **VRF-CLI2:**

Interface in VLAN 20 with IPv6 address 2001:db8:c0a8:1400::1/64

Interface in VLAN 200 with IPv6 address 2001:db8:c0a8:c800::1/64

To create VRFs VRF-CLI1 and VRF-CLI2, just follow the configurations below.

```
config
vrf VRF-CLI1
  description CLIENT1
!
vrf VRF-CLI2
  description CLIENT2
!
commit
```

Then, interfaces and routes are configured and assigned to these VRFs.

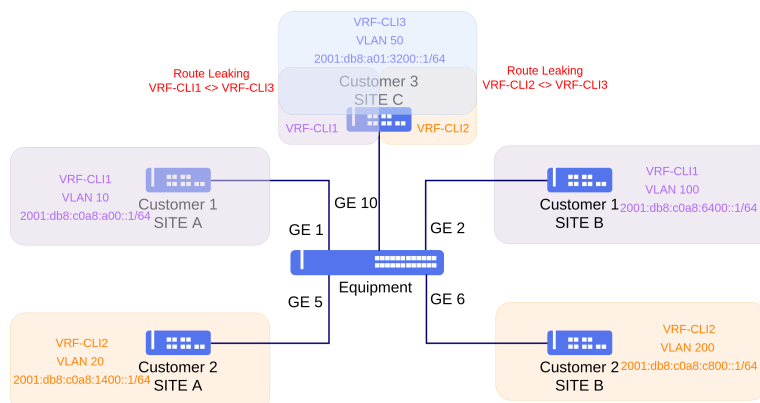
```
config
dot1q
  vlan 10
    interface gigabit-ethernet-1/1/1 tagged
  !
  !
  vlan 20
    interface gigabit-ethernet-1/1/5 tagged
  !
  !
  vlan 100
    interface gigabit-ethernet-1/1/2 tagged
  !
  !
  vlan 200
    interface gigabit-ethernet-1/1/6 tagged
  !
  !
!
interface l3 CLI1-VLAN10
vrf VRF-CLI1
ipv6 enable
ipv6 address 2001:db8:c0a8:a00::1/64
lower-layer-if vlan 10
!
interface l3 CLI1-VLAN100
vrf VRF-CLI1
ipv6 enable
ipv6 address 2001:db8:c0a8:6400::1/64
lower-layer-if vlan 100
!
interface l3 CLI2-VLAN20
vrf VRF-CLI2
ipv6 enable
ipv6 address 2001:db8:c0a8:1400::1/64
lower-layer-if vlan 20
!
interface l3 CLI2-VLAN200
vrf VRF-CLI2
ipv6 enable
ipv6 address 2001:db8:c0a8:c800::1/64
lower-layer-if vlan 200
!
router static
vrf VRF-CLI1
  address-family ipv6
    2001:db8:ac10::/64 next-hop 2001:db8:c0a8:a00::2
    2001:db8:ac11::/64 next-hop 2001:db8:c0a8:6400::2
  !
!
vrf VRF-CLI2
  address-family ipv6
    2001:db8:a14::/64 next-hop 2001:db8:c0a8:1400::2
    2001:db8:a15::/64 next-hop 2001:db8:c0a8:c800::2
  !
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying VRFs](#).

10.4.4 Enabling Route Leaking between IPv6 VRFs

The previous scenario will be used as base. A third client is included.



Route leaking between VRFs

Client 3 has the following specifications:

- **VRF-CLI3:**
Interface in VLAN 50 with IPv6 address 2001:db8:a01:3200::1/64

Since all the clients are in different VRFs, there is no communication between them. Considering there is a requirement that client 1 and 2 must access client 3, route leaking between VRFs must be used.

Each VRF must have a unique identifier called Route Distinguisher (RD). The RD is used to identify to which VRF each route belongs, allowing as well IP address overlapping between VRFs. The following RD's will be used:

- **VRF-CLI1:** rd 1:10
- **VRF-CLI2:** rd 2:20
- **VRF-CLI3:** rd 3:50

To enable the leaking, route-targets (RT) are used. Just like RD, RTs are identifiers added to the routes so the router knows which routes to be inserted in which VRFs. RTs can have the same format as the RD. Routes exported with a specific RT will be imported in VRFs that have the same RT configured as import.

It will be configured leaking between VRF-CLI1 and VRF-CLI3 and between VRF-CLI2 and VRF-CLI3, so there will be possible to Client 3 communicate with Client 1 and Client 2 as well. There will not be possible to Client 1 to communicate to Client 2 because they are not configured to import routes between them.



The route leaking between VRFs occurs with the redistribution of static routes and with the redistribution of directly connected routes.

```

config
dot1q
vlan 50
interface gigabit-ethernet-1/1/10 tagged
!
!
interface l3 CLI3-VLAN50
vrf VRF-CLI3
ipv6 enable
ipv6 address 2001:db8:a01:3200::1/64
lower-layer-if vlan 50
!
vrf VRF-CLI1
rd 1:10
route-target import 3:50
!
route-target import 1:10
!
route-target export 1:10
!
!
vrf VRF-CLI2
rd 2:20
route-target import 3:50
!
route-target import 2:20
!
route-target export 2:20
!
!
vrf VRF-CLI3
rd 3:50
route-target import 1:10
!
route-target import 2:20
!
route-target import 3:50
!
route-target export 3:50
!
!
router static
vrf VRF-CLI3
address-family ipv6
::/0 next-hop 2001:db8:a01:3200::2
!
!
!
!
router bgp 65500
address-family ipv6 unicast
!
vrf VRF-CLI1
address-family ipv6 unicast
redistribute static
!
exit-address-family
!
!
vrf VRF-CLI2
address-family ipv6 unicast
redistribute static
!
exit-address-family
!
!
vrf VRF-CLI3
address-family ipv6 unicast
redistribute static
!
exit-address-family
!
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying VRFs](#).

10.4.5 Verifying VRFs

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



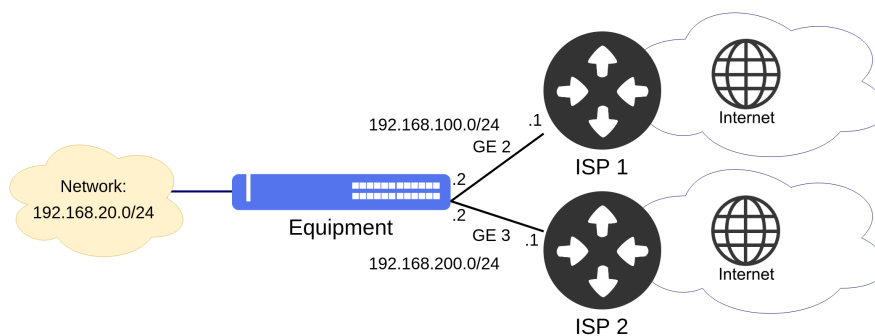
For more details about commands output, check the **Command Reference**.

```
show ip route vrf <VRF_NAME>
show ip fib vrf <VRF_NAME> brief
show ip host-table vrf <VRF_NAME> brief
show ip interface vrf <VRF_NAME> brief
show ipv6 route vrf <VRF_NAME>
show ipv6 fib vrf <VRF_NAME> brief
show ipv6 host-table vrf <VRF_NAME> brief
show ipv6 interface vrf <VRF_NAME> brief
```

10.5 PBR Configuration

PBR (Policy-based routing) allows the user to use rules to classify the traffic based on its attributes and forwarding selectively the packets for an alternative next hop. All the packets received which match a PBR rule are considered for policy based routing. When no routing policy is applied, the routing decisions are taken using the main routing table of the system. The PBR policies can be applied only to Ethernet interfaces of the data plane for incoming traffic, however the user cannot apply the PBR policies to packets locally generated.

The scenario below will be used to show configuration of the PBR.



PBR scenario

Considering that the user would like to configure a PBR policy so that the traffic originated in 192.168.20.1 host could be forwarded by the 192.168.200.1 next hop which is reachable through the Gigabit Ethernet 3 interface, and also so that the traffic originated in others hosts from 192.168.20.0/24 network could be forwarded by the main routing table.

The next steps will indicate how to execute these configurations.


```
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/2
untagged
!
vlan 200
interface gigabit-ethernet-1/1/3
untagged
!
!
switchport
interface gigabit-ethernet-1/1/2
native-vlan
vlan-id 100
!
!
interface gigabit-ethernet-1/1/3
native-vlan
vlan-id 200
!
!
interface l3 ISP_1
lower-layer-if vlan 100
ipv4 address 192.168.100.2/24
!
interface l3 ISP_2
lower-layer-if vlan 200
ipv4 address 192.168.200.2/24
!
router static
address-family ipv4
0.0.0.0/0 next-hop 192.168.100.1
!
!
router pbr 1
priority 7
match source ipv4-address 192.168.20.1/32
action next-hop 192.168.200.1
!
router pbr 2
priority 10
match source ipv4-address 192.168.20.0/24
action l3-routing
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying PBR](#).

10.5.1 Verifying PBR

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show router pbr
show router pbr <rule_id>
```

10.6 OSPFv2 Configuration

OSPFv2 (Open Shortest Path First version 2) is the Interior Gateway Protocol (IGP) described by the RFC 2328 (version 2) for IPv4 routing. As it is an IGP, it is used for routing inside an AS (Autonomous System). There is no routing exchange between ASs. It is based on Dijkstra algorithm, which calculates the shortest path to each destination based on costs each link cost.

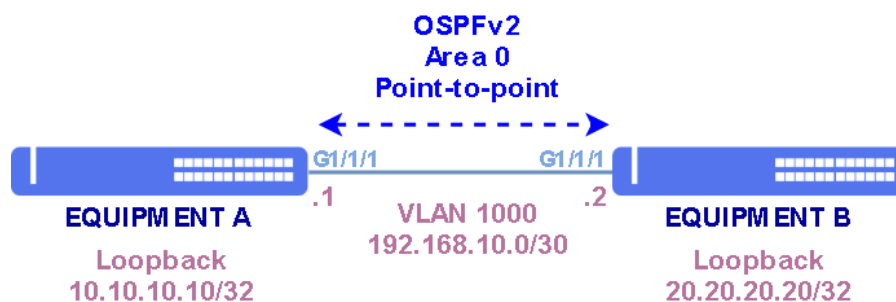


Currently, the OSPFv2 supports networks of the point-to-point and broadcast type.

- **Point to Point:** Network type that only a single adjacency can be formed over the link. There is no election of DR (designated router) or BDR (Backup Designated Router).
- **Broadcast:** A router is elected as DR (Designated Router) and another is elected as BDR (Backup Designated Router). This election is based on OSPF priorities and it is used to limit the amount of adjacencies formed in the network. Therefore, each router on the OSPF network will only form adjacencies with these two routers, the DR and BDR.

10.6.1 Configuring OSPFv2 in Point to Point Network

The scenario below will be used to illustrate the config of the OSPFv2.



Point to Point implementation of OSPFv2 protocol

To configure an OSPF session in area 0 with network-type point-to-point, follow the configuration steps below:

- **Equipment A:** L3 Interface in VLAN 1000 with IPv4 192.168.10.1/30 address and loopback interface with IPv4 10.10.10.10/32 being used as router-id in OSPFv2 in area 0.
- **Equipment B:** L3 Interface in VLAN 1000 with IPv4 192.168.10.2/30 address and loopback interface with IPv4 20.20.20.20/32 being used as router-id in OSPFv2 in area 0.

```
!Equipment A
config
dot1q
vlan 1000
interface gigabit-ethernet-1/1/1
untagged
!!
```

```

switchport
interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 1000
!
!
!
interface l3 OSPF
  lower-layer-if vlan 1000
  ipv4 address 192.168.10.1/30
!
interface loopback 0
  ipv4 address 10.10.10.10/32
!
router ospf 1
  router-id 10.10.10.10
  area 0
    interface l3-OSPF
      network-type point-to-point
!
!
!
interface loopback-0
!
!
!
commit

```

```

!Equipment B
config
dot1q
vlan 1000
  interface gigabit-ethernet-1/1/1
    untagged
!
!
!
switchport
interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 1000
!
!
!
interface l3 OSPF
  lower-layer-if vlan 1000
  ipv4 address 192.168.10.2/30
!
interface loopback 0
  ipv4 address 20.20.20.20/32
!
router ospf 1
  router-id 20.20.20.20
  area 0
    interface l3-OSPF
      network-type point-to-point
!
!
!
interface loopback-0
!
!
!
commit

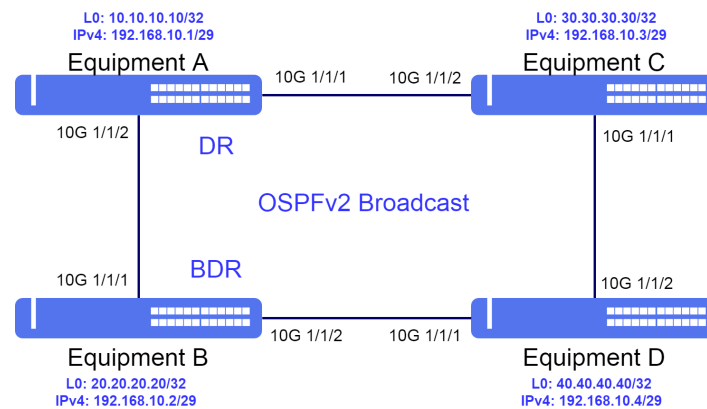
```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv2](#).

10.6.2 Configuring OSPFv2 in Broadcast Network

The scenario below will be used to demonstrate the configuration of OSPFv2 in Broadcast networks.



Broadcast implementation of OSPFv2 protocol

To configure an OSPF session in Area 0 with Broadcast network type, follow the configuration steps below:

- **Equipment A:** L3 Interface in VLAN 1000 with IPv4 address 192.168.10.1/29 and loopback interface with IPv4 address 10.10.10.10/32 being used as router-id in OSPFv2 in area 0. The OSPF priority configured will be 250.
- **Equipment B:** L3 Interface in VLAN 1000 with IPv4 address 192.168.10.2/29 and loopback interface with IPv4 address 20.20.20.20/32 being used as router-id in OSPFv2 in area 0. The OSPF priority configured will be 200.
- **Equipment C:** L3 Interface in VLAN 1000 with IPv4 address 192.168.10.3/29 and loopback interface with IPv4 address 30.30.30.30/32 being used as router-id in OSPFv2 in area 0. The OSPF priority configured will be 150.
- **Equipment D:** L3 Interface in VLAN 1000 with IPv4 address 192.168.10.4/29 and loopback interface with IPv4 address 40.40.40.40/32 being used as router-id in OSPFv2 in area 0. The OSPF priority configured will be 100.



Because OSPF's priorities are higher in EQUIPMENT A and B, they will be elected DR and BDR, respectively.

```
!Equipment A
config
dot1q
vlan 1000
interface ten-gigabit-ethernet-1/1/1
untagged
interface ten-gigabit-ethernet-1/1/2
untagged
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 1000
interface ten-gigabit-ethernet-1/1/2
native-vlan
vlan-id 1000
!
!
interface l3 OSPF
lower-layer-if vlan 1000
ipv4 address 192.168.10.1/29
!
interface loopback 0
ipv4 address 10.10.10.10/32
```

```

!
router ospf 1
router-id 10.10.10.10
area 0
interface l3-OSPF
network-type broadcast
router-priority 250
!
interface loopback-0
!
!
!
commit

```

```

!Equipment B
config
dot1q
vlan 1000
interface ten-gigabit-ethernet-1/1/1
untagged
interface ten-gigabit-ethernet-1/1/2
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 1000
interface ten-gigabit-ethernet-1/1/2
native-vlan
vlan-id 1000
!
!
!
interface l3 OSPF
lower-layer-if vlan 1000
ipv4 address 192.168.10.2/29
!
interface loopback 0
ipv4 address 20.20.20.20/32
!
router ospf 1
router-id 20.20.20.20
area 0
interface l3-OSPF
network-type broadcast
router-priority 200
!
interface loopback-0
!
!
!
commit

```

```

!Equipment C
config
dot1q
vlan 1000
interface ten-gigabit-ethernet-1/1/1
untagged
interface ten-gigabit-ethernet-1/1/2
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 1000
interface ten-gigabit-ethernet-1/1/2
native-vlan
vlan-id 1000
!
!
!
interface l3 OSPF
lower-layer-if vlan 1000
ipv4 address 192.168.10.3/29
!
interface loopback 0
ipv4 address 30.30.30.30/32
!
router ospf 1
router-id 30.30.30.30
area 0
interface l3-OSPF

```

```

    network-type broadcast
    router-priority 150
    !
  interface loopback-0
  !
  !
!
commit

```

```

!Equipment D
config
dot1q
vlan 1000
interface ten-gigabit-ethernet-1/1/1
  untagged
interface ten-gigabit-ethernet-1/1/2
  untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 1000
interface ten-gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 1000
!
!
!
interface l3 OSPF
  lower-layer-if vlan 1000
  ipv4 address 192.168.10.4/29
!
interface loopback 0
  ipv4 address 40.40.40.40/32
!
router ospf 1
  router-id 40.40.40.40
  area 0
    interface l3-OSPF
      network-type broadcast
      router-priority 100
    !
  interface loopback-0
  !
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv2](#).

10.6.3 Configuring the area in OSPFv2

It is possible to configure different area types in OSPFv2. The areas available are **normal**, **stub**, **stub no-summary**, **nssa**, **nssa no-summary** and **nssa suppress-external**.



It is possible to configure NSSA area with no-summary and suppress-external together.



It is not possible to configure the area type for area 0. This area always will be of NORMAL type, because it is considered the backbone area of OSPF network.

In the example below, the area 10 will be configured as stub area.

```
config
router ospf 1
 area 10
  stub
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv2](#).

10.6.4 Filtering received OSPFv2 routes

It may be required to limit the number of installed routes in some equipment due to hardware capacity restrictions.

In the example below, a prefix-list named *allowed-prefixes* is configured to allow only routes 203.0.113.0 with prefix length between /24 and /32. When that prefix-list is applied to OSPF protocol configuration, only the specified routes are installed. Other received prefixes are kept in the OSPF database, but are not installed in the RIB.



There is an implicit deny statement at the end of each prefix list.



A router tests for prefix list matches from the lowest sequence number to the highest.

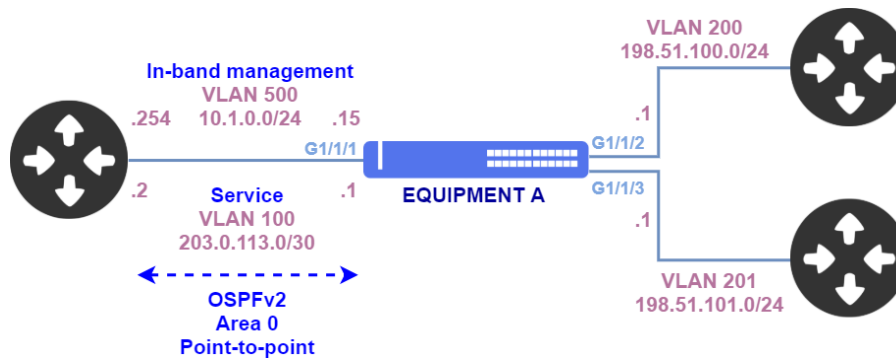
```
prefix-list allowed-prefixes
 seq 10
  action permit
  address 203.0.113.0/24
  le 32
!
!
router ospf 1
 import-prefix-list allowed-prefixes
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv2](#).

10.6.5 Filtering redistributed routes into OSPFv2

In the topology below, **EQUIPMENT A** is a CPE with *in-band* management (VLAN 500) and OSPF adjacency with the ISP in the service network (VLAN 100). This equipment has two L3 interfaces with IP addresses **198.51.100.1/24** and **198.51.101.1/24**. These routes have to be redistributed in OSPF. When using the **redistribute connected** configuration in OSPF, the management route will also be redistributed to the service network, which is not desired.



Filtering redistributed routes to OSPFv2

To make only VLANs 200 and 201 networks to be redistributed, an **export-prefix-list** rule can be configured in OSPF. The **mgmt-network** prefix is assigned to that rule, which filters the network 10.1.0.0/24, preventing it to be advertised to the OSPF neighbors. Routes 198.51.100.0/24 and 198.51.101.0/24 are still redistributed.

```
!Equipment A
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1
!
vlan 200
interface gigabit-ethernet-1/1/2
!
vlan 201
interface gigabit-ethernet-1/1/3
!
vlan 500
interface gigabit-ethernet-1/1/1
!
!
interface l3 VLAN100
lower-layer-if vlan 100
ipv4 address 203.0.113.1/30
!
interface l3 VLAN200
lower-layer-if vlan 200
ipv4 address 198.51.100.1/24
!
interface l3 VLAN201
lower-layer-if vlan 201
ipv4 address 198.51.101.1/24
!
interface l3 MGMT
lower-layer-if vlan 500
ipv4 address 10.1.0.15/24
!
router static
address-family ipv4
0.0.0.0/0 next-hop 10.1.0.254
!
router ospf 1
```



```

export-prefix-list mgmt-network
redistribute connected
area 0
interface l3-VLAN100
network-type point-to-point
!
!
!
prefix-list mgmt-network
seq 10
action deny
address 10.1.0.0/24
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv2](#).

10.6.6 Enabling ECMP in OSPFv2

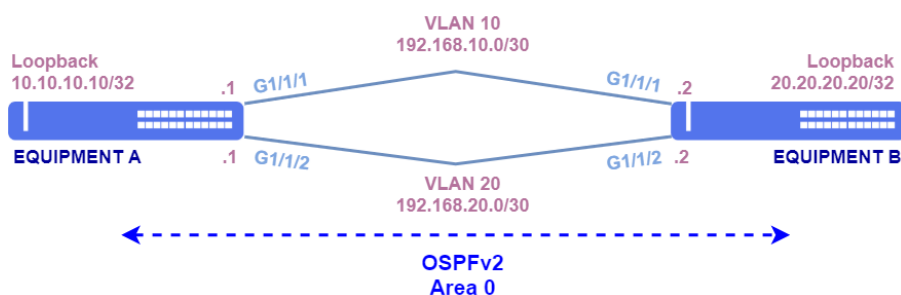
The OSPFv2 default behavior is the installation of only one route for the same destination. If there is more than one valid route with the same cost, only one is selected. Enabling ECMP (Equal-Cost Multi-Path), it makes OSPFv2 install more than one route to the same destination and loads balance between the available paths.

The load-balance algorithm works in a very similar way LAG load-balancing. A hash is generated based on packet parameters (IP addresses, UDP/TCP ports and VLAN). The hash value defines through which link the packets belonging to a specific flow is forwarded.

In the scenario below, there are two paths with the same cost between equipment A and equipment B. When the command **maximum paths 2** is inserted in OSPF, both paths are installed in the routing table.



It is possible to enable ECMP to use up to 16 different paths if they are available.



```
!Equipment A
config
dot1q
vlan 10
    interface gigabit-ethernet-1/1/1
    !
vlan 20
    interface gigabit-ethernet-1/1/2
    !
!
interface l3 VLAN10
    lower-layer-if vlan 10
    ipv4 address 192.168.10.1/30
!
interface l3 VLAN20
    lower-layer-if vlan 20
    ipv4 address 192.168.20.1/30
!
interface loopback 0
    ipv4 address 10.10.10.10/32
!
router ospf 1
    maximum paths 2
    area 0
        interface l3-VLAN10
            network-type point-to-point
        !
        interface l3-VLAN20
            network-type point-to-point
        !
        interface loopback-0
        !
    !
!
commit
```

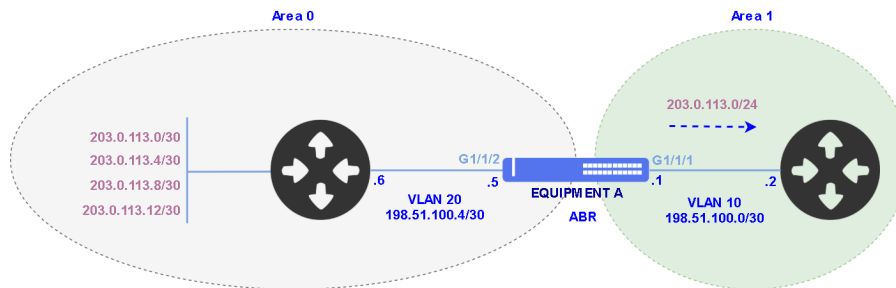
```
!Equipment B
config
dot1q
vlan 10
    interface gigabit-ethernet-1/1/1
    !
vlan 20
    interface gigabit-ethernet-1/1/2
    !
!
interface l3 VLAN10
    lower-layer-if vlan 10
    ipv4 address 192.168.10.2/30
!
interface l3 VLAN20
    lower-layer-if vlan 20
    ipv4 address 192.168.20.2/30
!
interface loopback 0
    ipv4 address 20.20.20.20/32
!
router ospf 1
    maximum paths 2
    area 0
        interface l3-VLAN10
            network-type point-to-point
        !
        interface l3-VLAN20
            network-type point-to-point
        !
        interface loopback-0
        !
    !
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv2](#).

10.6.7 Route summarization in OSPFv2

The topology below will be used to demonstrate OSPFv2 route summarization.



Route summarization in OSPFv2

In area 0, network 203.0.113.0/24 is divided in several subnets. **EQUIPMENT A** will advertise all routes to area 1. To decrease the number of routes in area 1, OSPF route summarization can be configured so only the network 203.0.113.0/24 is advertised.

In OSPF, route summarization happens only between areas. In this example, it is performed in EQUIPMENT A, which is an **ABR (Area Border Router)**, since it has connections in area 0 and area 1.

As shown in the configuration below, command **range 203.0.113.0 255.255.255.0**, configured in area 0, instructs OSPF to advertise to other areas only the network 203.0.113.0/24 and not its subnetworks.

It is recommended the use of a *black hole* route in EQUIPMENT A, avoiding that packets with destination to subnets that do not exist in area 0 are forwarded. For instance, in the presented topology, if a packet with destination IP address 203.0.113.100 is routed to area 0, it will not find a more specific route and could get in a routing loop. The *black hole* route discards these packets, avoiding that they are forwarded unnecessarily.



For more details on *black hole* routes configuration, please consult topic [Black hole route configuration](#)

```
!Equipment A
config
dot1q
vlan 10
    interface gigabit-ethernet-1/1/1
        untagged
    !
!
vlan 20
    interface gigabit-ethernet-1/1/2
        untagged
    !
!
!
switchport
interface gigabit-ethernet-1/1/1
    native-vlan
        vlan-id 10
    !
!
interface gigabit-ethernet-1/1/2
```

```
native-vlan
vlan-id 20
!
!
interface l3 VLAN10
lower-layer-if vlan 10
ipv4 address 198.51.100.1/30
!
interface l3 VLAN20
lower-layer-if vlan 20
ipv4 address 198.51.100.5/30
!
router static
address-family ipv4
203.0.113.0/24 black-hole
!
router ospf 1
area 0
interface l3-VLAN20
network-type point-to-point
!
range 203.0.113.0 255.255.255.0 advertise
!
area 1
interface l3-VLAN10
network-type point-to-point
!
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv2](#).

10.6.8 Verifying OSPFv2

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show ip ospf
show ip ospf neighbor
show ip ospf database
show ip ospf interface
show ip ospf detail
show ip ospf extensive
show ip ospf brief
show ip ospf database external
show ip route ospf
show ip rib ospf
```

10.7 OSPFv3 Configuration

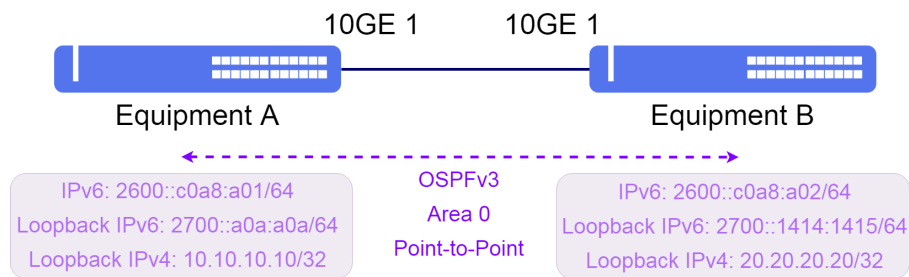
OSPFv3 (Open Shortest Path First version 3) is the Interior Gateway Protocol (IGP) described by the RFC 2740 for IPv6 routing. As this protocol is an IGP, it is used within an AS (Autonomous System) only. It is based on Dijkstra algorithm, which calculates the shortest path to each destination based on link costs.



Currently, the OSPFv3 supports only networks of the point-to-point type.

10.7.1 Configuring OSPFv3 Point to Point

The scenario below will be used to illustrate the config of the OSPFv3.



Basic implementation of OSPFv3 protocol

The parameters below are used for the following OSPFv3 configuration.

- **Equipment A:** L3 interface in VLAN 1000 with IPv6 2600::c0a8:a01/64 address and loopback interface with IPv6 2700::a0a:a0a/64 and IPv4 10.10.10.10/32, being used as router-id in the OSPFv3 in area 0.
- **Equipment B:** L3 interface in VLAN 1000 with IPv6 2600::c0a8:a02/64 address and loopback interface with IPv6 2700::1414:1415/64 and IPv4 20.20.20.20/32 being used as router-id in the OSPFv3 in area 0.

```
!Equipment A
config
dot1q
vlan 1000
interface ten-gigabit-ethernet-1/1/1
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 1000
!
!
interface l3 OSPFv3
lower-layer-if vlan 1000
ipv6 enable
ipv6 address 2600::c0a8:a01/64
!
interface loopback 0
ipv4 address 10.10.10.10/32
ipv6 enable
ipv6 address 2700::a0a:a0a/64
!
router ospfv3 1
router-id 10.10.10.10
area 0
interface l3-OSPFv3
network-type point-to-point
!
interface loopback-0
!
!
!
```

```
commit
```

```
!Equipment B
config
dot1q
vlan 1000
interface ten-gigabit-ethernet-1/1/1
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 1000
!
!
!
interface l3 OSPFv3
lower-layer-if vlan 1000
ipv6 enable
ipv6 address 2600::c0a8:a02/64
!
interface loopback 0
ipv4 address 20.20.20.20/32
ipv6 enable
ipv6 address 2700::1414:1415/64
!
router ospfv3 1
router-id 20.20.20.20
area 0
interface l3-OSPFv3
network-type point-to-point
!
interface loopback-0
!
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv3](#).

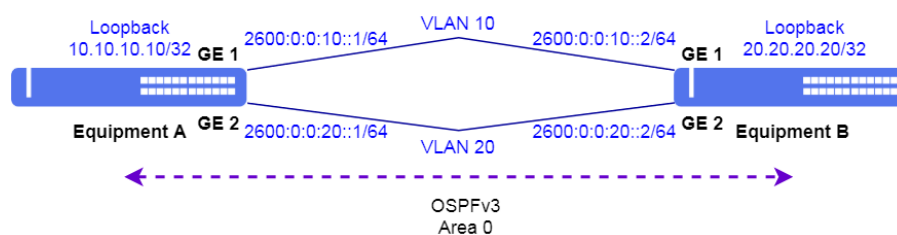
10.7.2 Enabling ECMP in OSPFv3

The OSPFv3 default behavior is the installation of only one route for the same destination. If there is more than one valid route with the same cost, only one will be selected. Enabling ECMP (Equal-Cost Multi-Path), it will make OSPFv3 install more than one route to the same destination and will load balance between the available paths.

In the scenario below, there are two paths with the same cost between equipment A and equipment B. Inserting the configuration command **maximum paths 2** in OSPF, both path will be installed in the routing table.



It is possible to enable ECMP to use up to 16 different paths if they are available.



OSPFv3 with ECMP

```
!Equipment A
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
!
vlan 20
interface gigabit-ethernet-1/1/2
!
!
interface l3 VLAN10
lower-layer-if vlan 10
ipv6 enable
ipv6 address 2600:0:0:10::1/64
!
interface l3 VLAN20
lower-layer-if vlan 10
ipv6 enable
ipv6 address 2600:0:0:20::1/64
!
interface loopback 0
ipv4 address 10.10.10.10/32
!
!
router ospfv3 1
maximum paths 2
area 0
interface l3-VLAN10
network-type point-to-point
!
interface l3-VLAN20
network-type point-to-point
!
interface loopback-0
!
!
commit
```

```
!Equipment B
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
!
vlan 20
interface gigabit-ethernet-1/1/2
!
!
interface l3 VLAN10
lower-layer-if vlan 10
ipv6 enable
ipv6 address 2600:0:0:10::2/64
!
interface l3 VLAN20
lower-layer-if vlan 10
ipv6 enable
ipv6 address 2600:0:0:20::2/64
!
interface loopback 0
ipv4 address 20.20.20.20/32
!
!
```

```
router ospfv3 1
maximum paths 2
area 0
interface l3-VLAN10
network-type point-to-point
!
interface l3-VLAN20
network-type point-to-point
!
interface loopback-0
!
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying OSPFv3](#).

10.7.3 Verifying OSPFv3

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

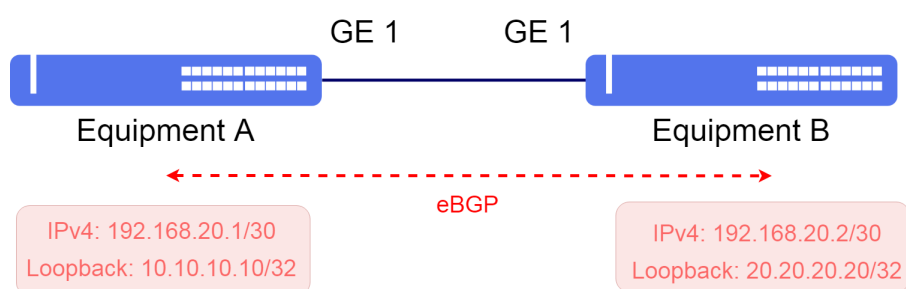
```
show ipv6 ospf
show ipv6 ospf neighbor
show ipv6 ospf database
show ipv6 ospf brief
show ipv6 ospf database external
show ipv6 route ospf
show ipv6 rib ospf
```

10.8 BGP Configuration

The BGP (Border Gateway Protocol) protocol is the protocol used for exchange of routing information between the AS (autonomous-system) in Internet. Upon establishing a neighborhood with a different AS, the BGP is called **eBGP** (external BGP) while when the neighborhood is established between routers of the same AS, the BGP is called **iBGP** (internal BGP).

10.8.1 Configuring a eBGP IPv4 Single Homed

The scenario below will be used to indicate the config of BGP protocol with IPv4 addresses in different AS, this means, eBGP.



Basic implementation of BGP IPv4 protocol

Considering that the user would like to execute the following configs:

- **Equipment A:** L3 Interface in VLAN 2000 with IPv4 192.168.20.1/30 address and loopback interface with IPv4 10.10.10.10/32 being used as router-id in BGP with local AS 20000 and remote AS 40000.
- **Equipment B:** L3 Interface in VLAN 2000 with IPv4 192.168.20.2/30 address and loopback interface with IPv4 20.20.20.20/32 being used as router-id in BGP with local AS 40000 and remote AS 20000.



It is recommended to use the address of the loopback interface instead of the physical interfaces in config of the iBGP neighborhood. On the other side the eBGP is recommended to use the addresses of the physical interfaces instead of the loopback.

```

!Equipment A
config
dot1q
    vlan 2000
        interface gigabit-ethernet-1/1/1
            untagged
    !
!
!
!
switchport
    interface gigabit-ethernet-1/1/1
        native-vlan
        vlan-id 2000
    !
!
!
!
interface l3 BGP
    lower-layer-if vlan 2000
    ipv4 address 192.168.20.1/30
!
!
interface loopback 0
    ipv4 address 10.10.10.10/32
!
!
!
router bgp 20000
    router-id 10.10.10.10
    address-family ipv4 unicast
!
    neighbor 192.168.20.2
        update-source-address 192.168.20.1
        remote-as 40000
        ebgp-multihop 1
        address-family ipv4 unicast
    !
!
!
!
commit

```

```
!Equipment B
config
dot1q
vlan 2000
  interface gigabit-ethernet-1/1/1
    untagged
  !
  !
!
!
switchport
interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 2000
  !
  !
!
interface l3 BGP
  lower-layer-if vlan 2000
  ipv4 address 192.168.20.2/30
  !
!
interface loopback 0
  ipv4 address 20.20.20.2/32
  !
!
!
router bgp 40000
  router-id 20.20.20.20
  address-family ipv4 unicast
  !
  neighbor 192.168.20.1
    update-source-address 192.168.20.2
    remote-as 20000
    ebgp-multihop 1
    address-family ipv4 unicast
  !
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying BGP](#).

10.8.2 Configuring route-maps and IPv4 prefix-lists

The configuration above will advertise every valid route present in the BGP RIB and accept any valid route received from the neighbor. In some cases, it may be required to filter the prefixes to avoid sending or receiving unwanted routes.

The export configuration (advertise-filter) will:

- Do not advertise private routes
- Do not advertise routes with prefix length greater than /24
- Advertise routes with prefix length between /16 and /18 with ASN prepend
- Advertise the remaining routes which do not match any of the previous rules

The import configuration (receive-filter) will:

- Do not accept private routes
- Do not accept routes with prefix length greater than /24
- Do not accept routes originated in AS 5678

- Accept remaining routes which do not match any of the previous rules



The **match-as-path** parameter accepts regular expressions. In following configuration, to match AS paths starting with ASN 5678, the used regular expression is `[^0-9]5678$`. `[^0-9]` will not match any number on that specific position, so a match will happen only with 5678 and not 15678, for example.

```
!Equipment A
config
prefix-list ipv4-private-networks
seq 10
  action permit
  address 10.0.0.0/8
  ge 8
seq 20
  action permit
  address 192.168.0.0/16
  ge 16
seq 30
  action permit
  address 172.16.0.0/12
  ge 12
!
prefix-list ipv4-more-specific-24
seq 10
  action permit
  address 0.0.0.0/0
  ge 25
!
prefix-list ipv4-between-16-18
seq 10
  action permit
  address 0.0.0.0/0
  ge 16
  le 18
!
router bgp 20000
route-map advertise-filter 10
  action deny
  match-ip nlri prefix-list ipv4-private-networks
!
route-map advertise-filter 30
  action deny
  match-ip nlri prefix-list ipv4-more-specific-24
!
route-map advertise-filter 40
  action permit
  match-ip nlri prefix-list ipv4-between-16-18
  set-prepend-local-as 1
!
route-map advertise-filter 50
  action permit
!
route-map receive-filter 10
  action deny
  match-ip nlri prefix-list ipv4-private-networks
!
route-map receive-filter 30
  action deny
  match-ip nlri prefix-list ipv4-more-specific-24
!
route-map receive-filter 40
  action deny
  match-as-path [^0-9]5678$
!
route-map receive-filter 50
  action permit
!
route-policy neighbor-as4000-policy
  import-route-map receive-filter
  export-route-map advertise-filter
!
neighbor 192.168.20.2
  route-policy neighbor-as4000-policy
!
commit
```

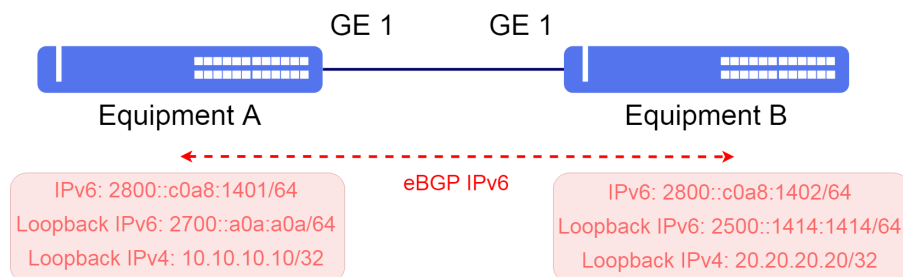


The available commands for troubleshooting can be found in the topic [Verifying BGP](#).

10.8.3 Configuring a iBGP IPv6 Single Homed

The BGP (Border Gateway Protocol) protocol is the protocol used for exchange of routing information between the AS (autonomous-system) in Internet. Upon establishing a neighborhood with a different AS, the BGP is called eBGP (external BGP) while when the neighborhood is established between routers of the same AS, the BGP is called iBGP (internal BGP).

The scenario below will be used to indicate the config of BGP protocol with IPv6 addresses in the same AS, this means, iBGP.



Basic implementation of BGP IPv6 protocol

Considering that the user would like to execute the following configs:

- **Equipment A:** L3 interface in VLAN 2000 with IPv6 2800::c0a8:1401/64 address and loopback interface with IPv6 2700::a0a:a0a/64 and IPv4 10.10.10.10/32 being used as router-id in the BGP with local AS 20000 and remote AS 20000.
- **Equipment B:** L3 interface in VLAN 2000 with IPv6 2800::c0a8:1402/64 address and loopback interface with IPv6 2500::1414:1414/64 and IPv4 20.20.20.20/32 being used as router-id in the BGP with local AS 20000 and remote AS 20000.



It is recommended to use the address of the loopback interface instead of the physical interfaces in config of the iBGP neighborhood. On the other side the eBGP is recommended to use the addresses of the physical interfaces instead of the loopback.

```
!Equipment A
config
dot1q
vlan 2000
interface gigabit-ethernet-1/1/1
untagged
!
!
!
!
!
switchport
```

```

interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 2000
  !
  !
  !
interface l3 BGP
  lower-layer-if vlan 2000
  ipv6 enable
  ipv6 address 2800::c0a8:1401/64
  !
  !
interface loopback 0
  ipv4 address 10.10.10.10/32
  ipv6 enable
  ipv6 address 2700::a0a:a0a/64
  !
  !
router bgp 20000
  router-id 10.10.10.10
  address-family ipv6 unicast
  !
  neighbor 2500::1414:1414
  update-source-address 2700::a0a:a0a
  remote-as 20000
  ebgp-multihop 255
  address-family ipv6 unicast
  !
  !
router static
  address-family ipv6
  2500::/64 next-hop 2800::c0a8:1402
commit

```

```

!Equipment B
config
dot1q
  vlan 2000
  interface gigabit-ethernet-1/1/1
    untagged
    !
    !
    !
  !
  !
switchport
  interface gigabit-ethernet-1/1/1
    native-vlan
    vlan-id 2000
    !
    !
  !
  !
interface l3 BGP
  lower-layer-if vlan 2000
  ipv6 enable
  ipv6 address 2800::c0a8:1402/64
  !
  !
interface loopback 0
  ipv4 address 20.20.20.20/32
  ipv6 enable
  ipv6 address 2500::1414:1414/64
  !
  !
router bgp 20000
  router-id 20.20.20.20
  address-family ipv6 unicast
  !
  neighbor 2700::a0a:a0a
  update-source-address 2500::1414:1414
  remote-as 20000
  ebgp-multihop 255
  address-family ipv6 unicast
  !
  !
router static
  address-family ipv6
  2700::/64 next-hop 2800::c0a8:1401
commit

```



The available commands for troubleshooting can be found in the topic [Verifying BGP](#).

10.8.4 Configuring route-maps and IPv6 prefix-lists

The configuration above will advertise every valid route present in the BGP RIB and accept any valid route received from the neighbor. In some cases, it may be required to filter the prefixes to avoid sending or receiving unwanted routes.

The following configuration will prevent EQUIPMENT A from sending or receiving private routes and routes with mask length greater than 48. Also, EQUIPMENT A will reject routes originated from ASN 5678.

```
!Equipment A
config
prefix-list ipv6-private-networks
seq 10
  action permit
  address fc00::/7
  ge 8
!
prefix-list ipv6-more-specific-48
seq 10
  action permit
  address ::/0
  ge 49
!
!
router bgp 20000
route-map advertise-filter 10
  action deny
  match-ip nlri prefix-list ipv6-private-networks
!
route-map advertise-filter 30
  action deny
  match-ip nlri prefix-list ipv6-more-specific-48
!
route-map advertise-filter 40
  action permit
!
route-map receive-filter 10
  action deny
  match-ip nlri prefix-list ipv6-private-networks
!
route-map receive-filter 30
  action deny
  match-ip nlri prefix-list ipv6-more-specific-48
!
route-map receive-filter 40
  match-as-path [^0-9]5678$
  action deny
!
route-map receive-filter 50
  action permit
!
route-policy neighbor-as2000-policy
  import-route-map receive-filter
  export-route-map advertise-filter
!
neighbor 2500::1414:1414
  route-policy neighbor-as2000-policy
!
commit
```

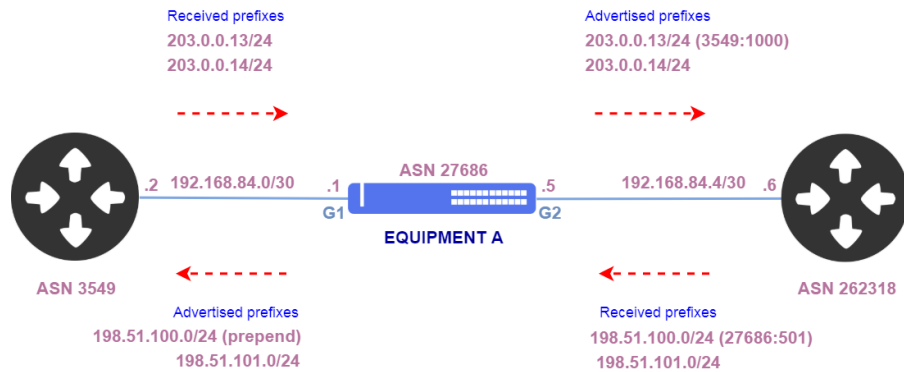


The available commands for troubleshooting can be found in the topic [Verifying BGP](#).

10.8.5 Configuring BGP Communities

BGP communities are parameters that can be added to prefixes advertised by BGP. These parameters can be used to perform actions over the prefixes, as reject them or change their characteristics.

The topology below will be used to show how to configure BGP communities.



Communities configuration

```
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1
!
vlan 200
interface gigabit-ethernet-1/1/2
!
!
interface l3 VLAN100
lower-layer-if vlan 100
ipv4 address 192.168.84.1/30
!
interface l3 VLAN200
lower-layer-if vlan 200
ipv4 address 192.168.84.5/30
!
!
router bgp 27686
router-id 192.168.0.1
address-family ipv4 unicast
!
neighbor 192.168.84.2
update-source-address 192.168.84.1
remote-as 3549
ebgp-multihop 1
address-family ipv4 unicast
!
neighbor 192.168.84.6
update-source-address 192.168.84.5
remote-as 262318
ebgp-multihop 1
address-family ipv4 unicast
!
!
```

Among the received prefixes from ASN 3549, prefix 203.0.0.13/24 must be marked with community 3549:1000 before being readvertised.

Among prefixes received from ASN 262318, prefix 198.51.100.0/24 is received with community 27686:501. Prefixes received

marked with this community must be readvertised with 2 ASNs prepend.

```

config
prefix-list prefix-203_0_0
seq 10
  action permit
  address 203.0.0.0/24
!
router bgp 27686
route-map import-from-asn3549 10
  action permit
  set-community 3549:1000
  set-community-action set-specific
  match-ip nlri prefix-list prefix-203_0_0
route-map import-from-asn3549 20
  action permit
!
route-map export-to-asn3549 10
  action permit
  match-community 27686:501
  set-prepend-local-as 2
route-map export-to-asn3549 20
  action permit
!
route-policy asn3549-policy
import-route-map import-from-asn3549
export-route-map export-to-asn3549
!
neighbor 192.168.84.2
route-policy asn3549-policy
!
!

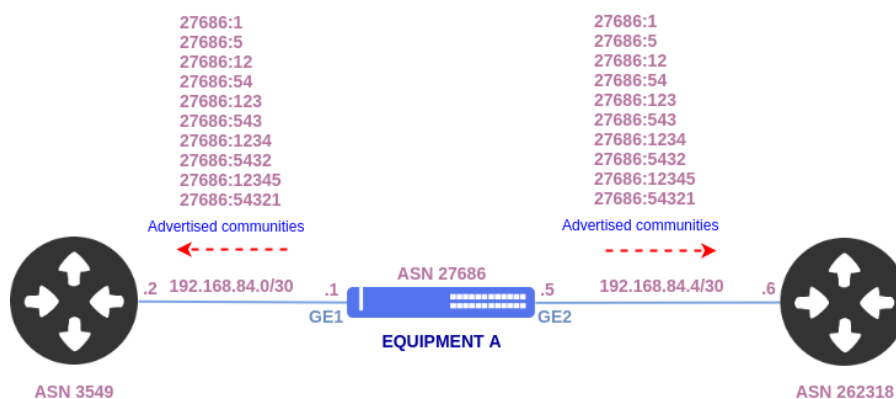
```



The available commands for troubleshooting can be found in the topic [Verifying BGP](#).

Configuration of communities list

As previously mentioned, communities are used to help control which prefixes will be accepted, rejected, distributed, advertised or altered in any other characteristics of the prefixes. In this way, it is possible to create a list of communities to apply these actions. In the example below, ASN 27686 is sending your list of communities to neighbors ASNs according to the contracting of IP transit.



Configuration of communities list



Using communities, in case there is no match by IP version, they will be applied to the 2 address families if the route-policy is configured in the IPv4 and IPv6 neighbors.

```
!Equipment A
config
dot1q
vlan 100
interface gigabit-ethernet-1/1/1
!
vlan 200
interface gigabit-ethernet-1/1/2
!
!
interface l3 VLAN100
lower-layer-if vlan 100
ipv4 address 192.168.84.1/30
!
interface l3 VLAN200
lower-layer-if vlan 200
ipv4 address 192.168.84.5/30
!
router bgp 27686
route-map export-community-list 100
set-community 27686:1
set-community-action set-specific
continue 101
!
route-map export-community-list 101
set-community 27686:5
set-community-action set-specific
continue 102
!
route-map export-community-list 102
set-community 27686:12
set-community-action set-specific
continue 103
!
route-map export-community-list 103
set-community 27686:54
set-community-action set-specific
continue 104
!
route-map export-community-list 104
set-community 27686:123
set-community-action set-specific
continue 105
!
route-map export-community-list 105
set-community 27686:543
set-community-action set-specific
continue 106
!
route-map export-community-list 106
set-community 27686:1234
set-community-action set-specific
continue 107
!
route-map export-community-list 107
set-community 27686:5432
set-community-action set-specific
continue 108
!
route-map export-community-list 108
set-community 27686:12345
set-community-action set-specific
continue 109
!
route-map export-community-list 109
set-community 27686:54321
set-community-action set-specific
!
route-policy export-community-list
export-route-map export-community-list
!
router-id 192.168.0.1
address-family ipv4 unicast
!
```

```

neighbor 192.168.84.2
 update-source-address 192.168.84.1
 route-policy export-community-list
 remote-as 3549
 ebgp-multihop 1
 address-family ipv4 unicast
!
neighbor 192.168.84.6
 update-source-address 192.168.84.5
 route-policy export-community-list
 remote-as 262318
 ebgp-multihop 1
 address-family ipv4 unicast
!
!
!

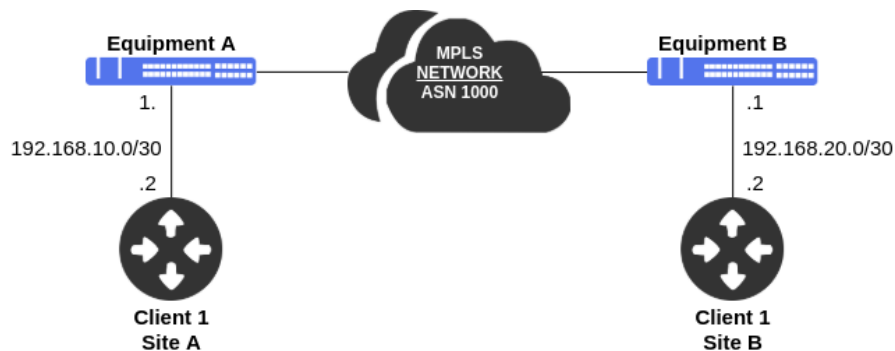
```



The available commands for troubleshooting can be found in the topic [Verifying BGP](#).

Communities configuration between PE and CPE L3VPN/6VPE.

Communities can also be used when there is BGP between PEs and CPEs in L3VPN/6VPE.



Communities configuration between PE and CPE.

In the example below there is a BGP connection between provider PE and Client1 CPE.



The commands available for complete performing iBGP configuration between PEs can be found in the topics [L3VPN Configuration](#) e [6VPE Configuration](#).

```

!Equipment A
config
!
router bgp 1000
 router-id 1.0.0.1
 address-family ipv4 unicast
!
 address-family ipv6 unicast
!
 address-family vpnv4 unicast
!
 address-family vpnv6 unicast
!

```

```

vrf cli1
router-id 192.168.10.1
address-family ipv4 unicast
 redistribute connected
!
exit-address-family
!
address-family ipv6 unicast
 redistribute connected
!
exit-address-family
!
neighbor 192.168.10.2
 update-source-address 192.168.10.1
 remote-as 65000
 next-hop-self
 address-family ipv4 unicast
  as-override
  exit-address-family
!
neighbor 2001:db8:1000::192:168:10:2
 update-source-address 2001:db8:1000::192:168:10:1
 route-policy asn-65000
 remote-as 65000
 next-hop-self
 address-family ipv6 unicast
  as-override
  exit-address-family
!
!
!
commit

```

```

!Equipment B
config
!
router bgp 1000
router-id 2.0.0.2
address-family ipv4 unicast
!
address-family ipv6 unicast
!
address-family vpnv4 unicast
!
address-family vpnv6 unicast
!
vrf cli1
router-id 192.168.20.1
address-family ipv4 unicast
 redistribute connected
!
exit-address-family
!
address-family ipv6 unicast
 redistribute connected
!
exit-address-family
!
neighbor 192.168.20.2
 update-source-address 192.168.20.1
 remote-as 65000
 next-hop-self
 address-family ipv4 unicast
  as-override
  exit-address-family
!
neighbor 2001:db8:1000::192:168:20:2
 update-source-address 2001:db8:1000::192:168:20:1
 route-policy asn-65000
 remote-as 65000
 next-hop-self
 address-family ipv6 unicast
  as-override
  exit-address-family
!
!
!
commit

```

In this case, different actions will be performed on each of the PEs. In Equipment A, according to the community received from the CPE, prepend 2 and metric 10 will be added in the prefixes received from the CPE. On Equipment B, the prefixes received from the CPE will be marked with the community 65000:20 and advertise to the customer's CPE that the prefixes

with the community 65000:10 will receive prepend 2 and metric 10.

```
!Equipment A
config
!
router bgp 1000
vrf cli1
  route-map import-from-asn-65000-site-A 10
    match-community 65000:10
    set-local-preference 10
  !
  route-map export-to-asn-65000-site-A 10
    match-community 65000:20
    set-prepend-local-as 2
  !
  route-map export-to-asn-65000-site-A 20
    match-community 65000:20
    set-med 20
  !
  route-policy asn-65000-policy
    import-route-map import-from-asn-65000-site-A
    export-route-map export-to-asn-65000-site-A
  !
  neighbor 192.168.10.2
    route-policy asn-65000-policy-site-A
  !
  neighbor 2001:db8:1000::192:168:10:2
    route-policy asn-65000-policy-site-A
  !
!
!
commit
```

```
!Equipment B
config
!
router bgp 1000
vrf cli1
  route-map import-from-asn-65000-site-B 10
    set-community 65000:20
    set-community-action set-specific
  !
  route-map export-to-asn-65000-site-B 10
    match-community 65000:10
    set-prepend-local-as 2
  !
  route-map export-to-asn-65000-site-B 20
    match-community 65000:10
    set-med 10
  !
  route-policy asn-65000-policy
    import-route-map import-from-asn-65000-site-B
    export-route-map export-to-asn-65000-site-B
  !
  neighbor 192.168.20.2
    route-policy asn-65000-policy-site-B
  !
  neighbor 2001:db8:1000::192:168:20:2
    route-policy asn-65000-policy-site-B
  !
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying BGP](#).

10.8.6 Verifying BGP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show ip bgp
show ip bgp neighbor
show ip bgp prefixes
show ip bgp summary
show ip bgp community
show ip route bgp
show ip rib bgp
show ipv6 route bgp
show ipv6 rib bgp
show ip bgp vrf <vrf-name>
show ip bgp vrf <vrf-name> neighbor
show ip bgp vrf <vrf-name> prefixes
show ip bgp vrf <vrf-name> summary
show ip bgp vrf <vrf-name> community
show ip rib vrf <vrf-name>
show ipv6 route vrf <vrf-name>
show ipv6 rib vrf <vrf-name>
```

10.9 VRRP Configuration

The VRRP (Virtual Router Redundancy Protocol) has as objective to eliminate the single point of failure, making available one or more equipment to be gateways of a LAN if the main gateway becomes unavailable. The protocol controls the IP addresses associated to a virtual router in which one of the equipment is elected Master and the others are elected Backup.



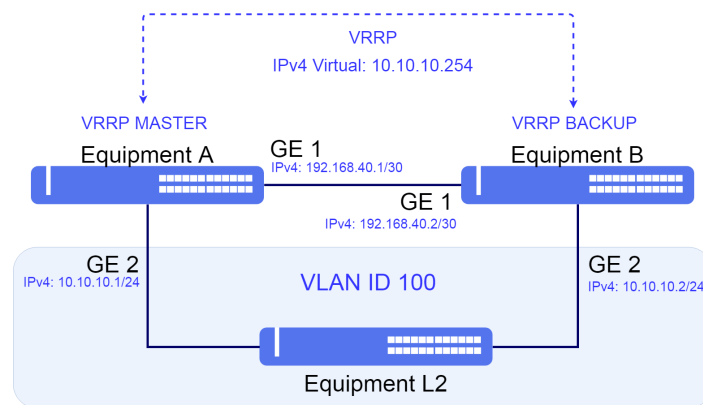
The **VRRPv2** versions are supported (with support to IPv4 addresses, described by the RFC 3768) and **VRRPv3** (with support to IPv4 and IPv6 addresses, described by RFC 5798).



A direct connection between the VRRP routers is recommended to increase the resiliency in case of individual failures of the links. In these direct connections use of the RSTP or of other L2 control protocols should be avoided.

10.9.1 Configuring a VRRPv2 to provide High-Availability

The scenario below will be used to illustrate the config of the VRRPv2 protocol to provide High-Availability.



Basic implementation of VRRP protocol

Considering that the user would like to execute the following configs:

- **Equipment A:** L3 interface for gateway of L2 network in VLAN 100 with IPv4 10.10.10.1/24 address. VRRP in version 2 with IP of Virtual Router 10.10.10.254, priority 250 and authentication with "password" password. Direct connection between routers (A and B) through L3 Interface in VLAN 4000 with IPv4 192.168.40.1/30 address.
- **Equipment B:** L3 interface for gateway of L2 network in VLAN 100 with IPv4 10.10.10.2/24 address. VRRP in version 2 with IP of Virtual Router 10.10.10.254, priority 200 and authentication with "password" password. Direct connection between routers (A and B) through L3 Interface in VLAN 4000 with IPv4 192.168.40.2/30 address

```
!Equipment A
config
dot1q
vlan 100
  interface gigabit-ethernet-1/1/2
  untagged
  !
  !
vlan 4000
  interface gigabit-ethernet-1/1/1
  !
!
switchport
  interface gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 100
  !
  !
!
interface l3 EQUIP-A-to-EQUIP-B
  lower-layer-if vlan 4000
  ipv4 address 192.168.40.1/30
  !
!
interface l3 VRRP
  lower-layer-if vlan 100
  ipv4 address 10.10.10.1/24
  !
!
router vrrp
  interface l3-VRRP
  address-family ipv4
  vr-id 1
  version v2
  priority 250
  authentication simple-text "password"
  address 10.10.10.254
commit
```

```
!Equipment B
config
dot1q
vlan 100
    interface gigabit-ethernet-1/1/2
        untagged
    !
    !
vlan 4000
    interface gigabit-ethernet-1/1/1
    !
!
switchport
interface gigabit-ethernet-1/1/2
    native-vlan
        vlan-id 100
    !
    !
!
interface l3 EQUIP-B-to-EQUIP-A
    lower-layer-if vlan 4000
    ipv4 address 192.168.40.2/30
    !
!
interface l3 VRRP
    lower-layer-if vlan 100
    ipv4 address 10.10.10.2/24
    !
!
router vrrp
    interface l3-VRRP
        address-family ipv4
            vr-id 1
                version v2
                priority 200
                authentication simple-text "password"
                address 10.10.10.254
        !
    !
commit
```



The available commands for troubleshooting can be found in the topic [Verifying VRRP](#).

10.9.2 Verifying VRRP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show router vrrp brief
```

10.10 BFD Configuration

The BFD protocol (Bidirectional Forwarding Detection) is defined by RFC 5880. It is used for fast link failure detection between two equipment.



BFD protocol is supported only in OSPFv2.



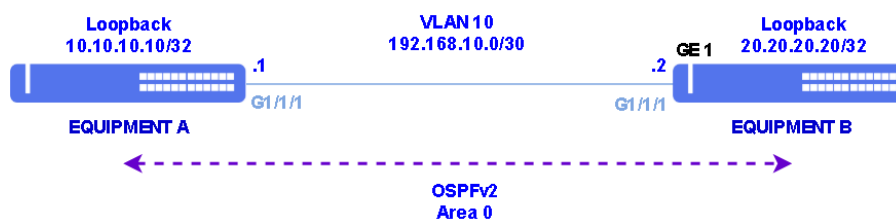
TX and RX intervals are fixed in 300ms and the multiplier is fixed in 3. It is not possible to change them through configuration, but they can be changed upon negotiation with the BFD neighbor.

If there is a communication failure in the link between two equipment, but the interfaces stay up, the routing protocol does not go down immediately. Only after a timeout occurs, the protocol goes down. The timeout can take several seconds, depending on the routing protocol configurations. While it is still up, traffic is sent through that link, and it is discarded.

When BFD is enabled in that routing protocol interface, it detects the link failure very fast, signaling to the routing protocol that the session is down, making it to converge immediately, avoiding that packets keep being forwarded to that the link during the failure.

10.10.1 Configuring BFD in OSPFv2

In the scenario below, there is an OSPF session between equipment A and B. The following configuration shows how to enable BFD in this OSPF interface.



BFD with OSPFv2

```
!Equipment A
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
untagged
!
!
switchport
interface gigabit-ethernet-1/1/1
native vlan
vlan-id 10
!
!
interface l3 VLAN10
lower-layer-if vlan 10
ipv4 address 192.168.10.1/30
!
interface loopback 0
ipv4 address 10.10.10.10/32
!
```



```
router ospf 1
 area 0
  interface loopback 0
  !
  interface l3-VLAN10
   bfd
   session-type desired
  !
  network-type point-to-point
 !
 !
 !
commit
```

```
!Equipment B
config
dot1q
vlan 10
 interface gigabit-ethernet-1/1/1
  untagged
 !
 !
switchport
 interface gigabit-ethernet-1/1/1
  native vlan
  vlan-id 10
 !
 !
interface l3 VLAN10
 lower-layer-if vlan 10
 ipv4 address 192.168.10.2/30
 !
interface loopback 0
 ipv4 address 20.20.20.20/32
 !
router ospf 1
 area 0
  interface loopback 0
  !
  interface l3-VLAN10
   bfd
   session-type desired
  !
  network-type point-to-point
 !
 !
 !
commit
```

10.10.2 Verifying BFD

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show bfd session
```

11 MPLS

MPLS (Multi-Protocol Label Switching), defined by RFC 3031, based on forwarding of packets by labels. It provides allows traffic engineering and VPNs (Virtual Private Networks). In this chapter, L2VPNs and L3VPNs configuration will be covered.



To work the PDUs encapsulation in L2VPNs it is necessary to remove the remote-devices feature that is enabled in the default configuration for DM4360 and DM4370 platforms. *configure; no remote-devices; commit.*

This chapter contains the following sections:

- LDP Configuration
- RSVP Configuration
- VPWS Configuration
- VPLS Configuration
- Verifying L2VPNs
- L3VPN Configuration
- 6VPE Configuration
- BGP Free Core Configuration

The signaling in MPLS is responsible for defining the LSP which is the path that the L2VPN will use and can be performed via LDP or RSVP protocols.

11.1 LDP Configuration

LDP (Label Distribution Protocol) is used for label distribution in the MPLS network equipment.



A license is required for MPLS operation. For more details on how to activate it, check the topic [Licenses Configuration](#).

The following topology will be used to exemplify LDP configuration.



LDP Configuration

```
!Equipment A
config
dot1q
vlan 10
  interface ten-gigabit-ethernet-1/1/1
  untagged
  !
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 10
  !
!
!
interface l3 VLAN10
  ipv4 address 203.0.113.1/30
  lower-layer-if vlan 10
  !
!
interface loopback 0
  ipv4 address 198.51.100.1/32
  !
!
router ospf 1
  router-id 198.51.100.1
  area 0
    interface l3-VLAN10
    network-type point-to-point
  !
  interface loopback-0
  !
!
!
mpls ldp
  lsr-id loopback-0
  interface l3-VLAN10
  !
!
!
commit
```

```
!Equipment B
config
dot1q
vlan 10
  interface ten-gigabit-ethernet-1/1/1
  untagged
  !
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 10
  !
!
!
interface l3 VLAN10
  ipv4 address 203.0.113.2/30
  lower-layer-if vlan 10
  !
!
interface loopback 0
  ipv4 address 198.51.100.2/32
  !
!
router ospf 1
  router-id 198.51.100.2
  area 0
    interface l3-VLAN10
    network-type point-to-point
  !
  interface loopback-0
  !
!
!
mpls ldp
  lsr-id loopback-0
  interface l3-VLAN10
  !
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying LDP](#).

11.1.1 Verifying LDP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show mpls ldp neighbor brief
show mpls ldp neighbor detail
show mpls ldp database
show mpls forwarding-table
```

11.2 RSVP Configuration

The **RSVP** protocol is used for tunnel establishment between equipment. LDP uses the best path chosen by the IGP to reach the destination while RSVP uses a path configured by the operator that does not necessarily follow the IGP.



A license is required for MPLS operation. For more details on how to activate it, check the topic [Licenses Configuration](#).

It is possible to configure RSVP in the DmOS using [affinity](#) or [explicit path](#).

11.2.1 Configuring RSVP affinity

To define which path a tunnel should use, DmOS uses interface attributes called *affinity*. Each bit configured in that attribute can correspond to a particular characteristic of the the link.

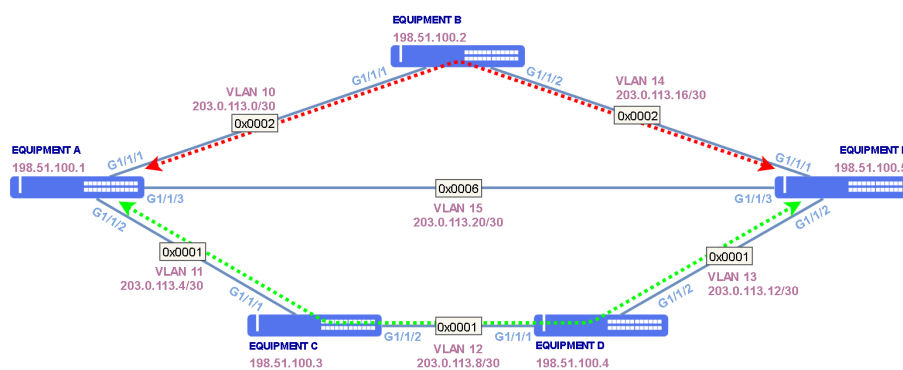
In the following example, links have three possible characteristics. For each characteristic, a specific bit is used. That value is represented in hexadecimal in DmOS.

Characteristic	Binary	Hexadecimal
High throughput link	00000001	0x0001
Low throughput link	00000010	0x0002

Characteristic	Binary	Hexadecimal
High latency link	00000100	0x0004

A link can have two or more characteristics simultaneously. For instance, a link can be of low throughput and high latency at the same time. In that case, its attribute will be the sum of 0x2 and 0x4, resulting in 0x6.

Tunnels are unidirectional. It is necessary to configure one tunnel originating in EQUIPMENT A destined to EQUIPMENT E and other tunnel originating in EQUIPMENT E destined to EQUIPMENT A. The tunnels will be assigned to two paths. The primary path will pass only through links with attribute 0x1 (high throughput). If a failure occurs in the primary path, the tunnel will be reestablished through the secondary path, which can pass through links with attribute 0x2 (low throughput). Both paths can not pass through links with attribute 0x4 (high latency), so both paths are configured to exclude those links. In the topology, the lowest cost path is through VLAN 15, but this link has the high latency attribute (0x4), so it has been excluded.



RSVP with affinity

The first step is to configure interfaces and OSPF.

```
!Equipment A
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
untagged
!
vlan 11
interface gigabit-ethernet-1/1/2
untagged
!
vlan 15
interface gigabit-ethernet-1/1/3
untagged
!
!
switchport
interface gigabit-ethernet-1/1/1
native-vlan
vlan-id 10
!
interface gigabit-ethernet-1/1/2
native-vlan
vlan-id 11
!
```

```

!
interface gigabit-ethernet-1/1/3
 native-vlan
   vlan-id 15
!
!
interface loopback 0
 ipv4 address 198.51.100.1/32
!
interface l3 VLAN10
 lower-layer-if vlan 10
 ipv4 address 203.0.113.1/30
!
interface l3 VLAN11
 lower-layer-if vlan 11
 ipv4 address 203.0.113.5/30
!
interface l3 VLAN15
 lower-layer-if vlan 15
 ipv4 address 203.0.113.21/30
!
router ospf 1 vrf global
 area 0
   interface l3-VLAN10
     network-type point-to-point
   !
   interface l3-VLAN11
     network-type point-to-point
   !
   interface l3-VLAN15
     network-type point-to-point
   !
   interface loopback-0
   !
!
!
commit

```

```

!Equipment B
config
dot1q
 vlan 10
   interface gigabit-ethernet-1/1/1
     untagged
   !
 vlan 14
   interface gigabit-ethernet-1/1/2
     untagged
   !
!
switchport
 interface gigabit-ethernet-1/1/1
   native-vlan
     vlan-id 10
   !
 interface gigabit-ethernet-1/1/2
   native-vlan
     vlan-id 14
   !
!
!
interface loopback 0
 ipv4 address 198.51.100.2/32
!
interface l3 VLAN10
 lower-layer-if vlan 10
 ipv4 address 203.0.113.2/30
!
interface l3 VLAN14
 lower-layer-if vlan 14
 ipv4 address 203.0.113.17/30
!
router ospf 1 vrf global
 area 0
   interface l3-VLAN10
     network-type point-to-point
   !
   interface l3-VLAN14
     network-type point-to-point
   !
   interface loopback-0
   !
!
!
commit

```

```

!Equipment C
config
dot1q
vlan 11
    interface gigabit-ethernet-1/1/1
        untagged
    !
!
vlan 12
    interface gigabit-ethernet-1/1/2
        untagged
    !
!
switchport
interface gigabit-ethernet-1/1/1
    native-vlan
    vlan-id 11
    !
!
interface gigabit-ethernet-1/1/2
    native-vlan
    vlan-id 12
    !
!
!
interface loopback 0
    ipv4 address 198.51.100.3/32
!
interface l3 VLAN11
    lower-layer-if vlan 11
    ipv4 address 203.0.113.6/30
!
interface l3 VLAN12
    lower-layer-if vlan 12
    ipv4 address 203.0.113.9/30
!
router ospf 1 vrf global
    area 0
        interface l3-VLAN11
            network-type point-to-point
        !
        interface l3-VLAN12
            network-type point-to-point
        !
        interface loopback-0
        !
    !
!
commit

```

```

!Equipment D
config
dot1q
vlan 12
    interface gigabit-ethernet-1/1/1
        untagged
    !
!
vlan 13
    interface gigabit-ethernet-1/1/2
        untagged
    !
!
switchport
interface gigabit-ethernet-1/1/1
    native-vlan
    vlan-id 12
    !
!
interface gigabit-ethernet-1/1/2
    native-vlan
    vlan-id 13
    !
!
!
interface loopback 0
    ipv4 address 198.51.100.4/32
!
interface l3 VLAN12
    lower-layer-if vlan 12
    ipv4 address 203.0.113.10/30
!
interface l3 VLAN13
    lower-layer-if vlan 13
    ipv4 address 203.0.113.13/30
!
router ospf 1 vrf global
    area 0

```

```

interface l3-VLAN12
 network-type point-to-point
!
interface l3-VLAN13
 network-type point-to-point
!
interface loopback-0
!
!
!
commit

```

```

!Equipment E
config
dot1q
vlan 13
 interface gigabit-ethernet-1/1/2
  untagged
!
!
vlan 14
 interface gigabit-ethernet-1/1/1
  untagged
!
!
vlan 15
 interface gigabit-ethernet-1/1/3
  untagged
!
!
!
switchport
 interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 14
!
!
 interface gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 13
!
!
 interface gigabit-ethernet-1/1/3
  native-vlan
  vlan-id 15
!
!
!
 interface loopback 0
  ipv4 address 198.51.100.5/32
!
 interface l3 VLAN13
  lower-layer-if vlan 13
  ipv4 address 203.0.113.14/30
!
 interface l3 VLAN14
  lower-layer-if vlan 14
  ipv4 address 203.0.113.18/30
!
 interface l3 VLAN15
  lower-layer-if vlan 15
  ipv4 address 203.0.113.22/30
!
 router ospf 1 vrf global
  area 0
   interface l3-VLAN13
    network-type point-to-point
   !
   interface l3-VLAN14
    network-type point-to-point
   !
   interface l3-VLAN15
    network-type point-to-point
   !
   interface loopback-0
   !
!
!
commit

```

It is necessary to enable RSVP in the interfaces and in OSPF.

Informations of each link are advertised by OSPF. RSVP protocol defines a path and signals the tunnel based on the OSPF information.


```
!Equipment A
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN10
!
interface l3-VLAN11
!
interface l3-VLAN15
!
!
commit
```

```
!Equipment B
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN10
!
interface l3-VLAN14
!
!
commit
```

```
!Equipment C
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN11
!
interface l3-VLAN12
!
!
commit
```

```
!Equipment D
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN12
!
interface l3-VLAN13
!
!
commit
```

```
!Equipment E
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN13
!
interface l3-VLAN14
!
interface l3-VLAN15
!
!
commit
```

The next step is to configure the interface *affinity* attributes.

```
!Equipment A
config
mpls traffic-eng
interface l3-VLAN10
affinity-flags 0x2
!
interface l3-VLAN11
```

```

    affinity-flags 0x1
!
interface l3-VLAN15
    affinity-flags 0x6
!
!
commit

```

```

!Equipment B
config
mpls traffic-eng
interface l3-VLAN10
    affinity-flags 0x2
!
interface l3-VLAN14
    affinity-flags 0x2
!
!
commit

```

```

!Equipment C
config
mpls traffic-eng
interface l3-VLAN11
    affinity-flags 0x1
!
interface l3-VLAN12
    affinity-flags 0x1
!
!
commit

```

```

!Equipment D
config
mpls traffic-eng
interface l3-VLAN12
    affinity-flags 0x1
!
interface l3-VLAN13
    affinity-flags 0x1
!
!
commit

```

```

!Equipment E
config
mpls traffic-eng
interface l3-VLAN13
    affinity-flags 0x1
!
interface l3-VLAN14
    affinity-flags 0x2
!
interface l3-VLAN15
    affinity-flags 0x6
!
!
commit

```

After RSVP has been enabled and *affinity* attributes have been configured, it is possible to configure the tunnels. The tunnel configuration is performed only in the originating equipment.

There are three ways to select the path using *affinity* attributes. Links that do not comply with the rules configured in the path definition (*path-option*) are not considered for tunnel establishment.

- include-any - At least one bit of the link affinity attribute must match with the bits configured in this parameter. If the include-any parameter value is 0, a link is valid despite its affinity attribute value.
- include-all - All bits configured in this parameter must match with the bits configured in the link affinity attribute. If the include-all parameter value is 0, a link is valid despite its affinity attribute value.

- **exclude-any** - Paths that have any of the specified bits are excluded.

Two paths have been defined. The primary path **HIGH-CAPACITY-LOW-DELAY** is established only through links that have the attribute related to high throughput (include-any 0x1) and do not have the attributed related to high latency (exclude-any 0x4).

Path **LOW-CAPACITY-LOW-DELAY** can be established only through links with attribute related to low throughput (include-any 0x2) and do not have the attributed related to high latency (exclude-any 0x4).

The path priority is defined by a number when assigned the path to the tunnel. In the configuration below, *path-option 10* has higher priority over *path-option 20*.

```
!Equipment A
config
mpls traffic-eng
attribute-set
  path-option HIGH-CAPACITY-LOW-DELAY
    affinity-flags exclude-any 0x4
    affinity-flags include-any 0x1
  !
  path-option LOW-CAPACITY-LOW-DELAY
    affinity-flags exclude-any 0x4
    affinity-flags include-any 0x2
  !
!
!
interface tunnel-te 1
destination 198.51.100.5
  path-option 10 dynamic attribute-set HIGH-CAPACITY-LOW-DELAY
  path-option 20 dynamic attribute-set LOW-CAPACITY-LOW-DELAY
!
commit
```

```
!Equipment E
config
mpls traffic-eng
attribute-set
  path-option HIGH-CAPACITY-LOW-DELAY
    affinity-flags exclude-any 0x4
    affinity-flags include-any 0x1
  !
  path-option LOW-CAPACITY-LOW-DELAY
    affinity-flags exclude-any 0x4
    affinity-flags include-any 0x2
  !
!
!
interface tunnel-te 1
destination 198.51.100.1
  path-option 10 dynamic attribute-set HIGH-CAPACITY-LOW-DELAY
  path-option 20 dynamic attribute-set LOW-CAPACITY-LOW-DELAY
!
commit
```

For examples of tunnel association to L2VPNs, consult sessions [VPWS with RSVP](#) or [VPLS with RSVP](#).



The available commands for troubleshooting can be found in the topic [Verifying RSVP](#).

11.2.2 Configuring RSVP explicit path

Another way to configure RSVP tunnel is using *explicit path* which consists in a series of hops. These hops can be *loose* or *strict* types. Loose type permit that hop is not directly connected to previous node of list and strict needs hop is directly

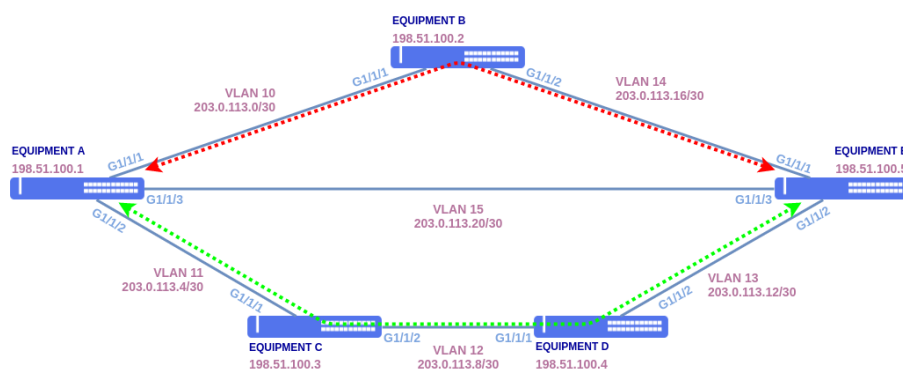
connected to previous node of list. It is possible combine loose and strict hops in the same path.

Tunnels are unidirectional. It is necessary to configure one tunnel originating in EQUIPMENT A destined to EQUIPMENT E and other tunnel originating in EQUIPMENT E destined to EQUIPMENT A.



If no type is specified, by default the hop is assumed to be strict.

The tunnels will be assigned to two paths. The primary path will pass only through links with equipment B. If a failure occurs in the primary path, the tunnel will be reestablished through the secondary path, which can pass through links of equipments C and D. The middle way is not used for any tunnel.



RSVP with explicit path

The first step is to configure interfaces and OSPF.

```
!Equipment A
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
untagged
!
vlan 11
interface gigabit-ethernet-1/1/2
untagged
!
vlan 15
interface gigabit-ethernet-1/1/3
untagged
!
!
switchport
interface gigabit-ethernet-1/1/1
native-vlan
vlan-id 10
!
interface gigabit-ethernet-1/1/2
native-vlan
vlan-id 11
!
interface gigabit-ethernet-1/1/3
native-vlan
vlan-id 15
!!
!
```

```

interface loopback 0
  ipv4 address 198.51.100.1/32
!
interface l3 VLAN10
  lower-layer-if vlan 10
  ipv4 address 203.0.113.1/30
!
interface l3 VLAN11
  lower-layer-if vlan 11
  ipv4 address 203.0.113.5/30
!
interface l3 VLAN15
  lower-layer-if vlan 15
  ipv4 address 203.0.113.21/30
!
router ospf 1 vrf global
  area 0
    interface l3-VLAN10
      network-type point-to-point
    !
    interface l3-VLAN11
      network-type point-to-point
    !
    interface l3-VLAN15
      network-type point-to-point
    !
    interface loopback-0
    !
  !
!
commit

```

```

!Equipment B
config
dot1q
vlan 10
  interface gigabit-ethernet-1/1/1
    untagged
  !
!
vlan 14
  interface gigabit-ethernet-1/1/2
    untagged
  !
!
!
switchport
interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 10
!
!
interface gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 14
!
!
!
interface loopback 0
  ipv4 address 198.51.100.2/32
!
!
interface l3 VLAN10
  lower-layer-if vlan 10
  ipv4 address 203.0.113.2/30
!
interface l3 VLAN14
  lower-layer-if vlan 14
  ipv4 address 203.0.113.17/30
!
router ospf 1 vrf global
  area 0
    interface l3-VLAN10
      network-type point-to-point
    !
    interface l3-VLAN14
      network-type point-to-point
    !
    interface loopback-0
    !
  !
!
commit

```

```

!Equipment C
config
dot1q
vlan 11

```

```

interface gigabit-ethernet-1/1/1
  untagged
!
!
vlan 12
interface gigabit-ethernet-1/1/2
  untagged
!
!
!
switchport
interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 11
!
!
interface gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 12
!
!
!
interface loopback 0
  ipv4 address 198.51.100.3/32
!
interface l3 VLAN11
  lower-layer-if vlan 11
  ipv4 address 203.0.113.6/30
!
interface l3 VLAN12
  lower-layer-if vlan 12
  ipv4 address 203.0.113.9/30
!
router ospf 1 vrf global
  area 0
    interface l3-VLAN11
      network-type point-to-point
    !
    interface l3-VLAN12
      network-type point-to-point
    !
    interface loopback-0
    !
  !
!
commit

```

```

!Equipment D
config
dot1q
vlan 12
  interface gigabit-ethernet-1/1/1
    untagged
  !
!
vlan 13
  interface gigabit-ethernet-1/1/2
    untagged
  !
!
!
switchport
interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 12
!
!
interface gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 13
!
!
!
interface loopback 0
  ipv4 address 198.51.100.4/32
!
interface l3 VLAN12
  lower-layer-if vlan 12
  ipv4 address 203.0.113.10/30
!
interface l3 VLAN13
  lower-layer-if vlan 13
  ipv4 address 203.0.113.13/30
!
router ospf 1 vrf global
  area 0
    interface l3-VLAN12
      network-type point-to-point
    !
    interface l3-VLAN13

```

```

    network-type point-to-point
    !
  interface loopback-0
  !
!
!
commit

```

```

!Equipment E
config
dot1q
vlan 13
  interface gigabit-ethernet-1/1/2
  untagged
  !
!
vlan 14
  interface gigabit-ethernet-1/1/1
  untagged
  !
!
vlan 15
  interface gigabit-ethernet-1/1/3
  untagged
  !
!
!
switchport
  interface gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 14
  !
  interface gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 13
  !
  interface gigabit-ethernet-1/1/3
  native-vlan
  vlan-id 15
  !
!
!
interface loopback 0
  ipv4 address 198.51.100.5/32
  !
interface l3 VLAN13
  lower-layer-if vlan 13
  ipv4 address 203.0.113.14/30
  !
interface l3 VLAN14
  lower-layer-if vlan 14
  ipv4 address 203.0.113.18/30
  !
interface l3 VLAN15
  lower-layer-if vlan 15
  ipv4 address 203.0.113.22/30
  !
router ospf 1 vrf global
  area 0
  interface l3-VLAN13
  network-type point-to-point
  !
  interface l3-VLAN14
  network-type point-to-point
  !
  interface l3-VLAN15
  network-type point-to-point
  !
  interface loopback-0
  !
!
!
commit

```

It is necessary to enable RSVP in the interfaces and in OSPF.

Informations of each link are advertised by OSPF. RSVP protocol defines a path and signals the tunnel based on the OSPF information.

```
!Equipment A
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN10
!
interface l3-VLAN11
!
interface l3-VLAN15
!
!
commit
```

```
!Equipment B
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN10
!
interface l3-VLAN14
!
!
commit
```

```
!Equipment C
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN11
!
interface l3-VLAN12
!
!
commit
```

```
!Equipment D
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN12
!
interface l3-VLAN13
!
!
commit
```

```
!Equipment E
config
router ospf 1 vrf global
mpls-te router-id loopback-0
!
mpls rsvp
interface l3-VLAN13
!
interface l3-VLAN14
!
interface l3-VLAN15
!
!
commit
```

After RSVP has been enabled, it is possible to configure the tunnels. The tunnel configuration is performed only in the originating equipment.

Two paths have been defined. The primary path is established only through links of equipment B and secondary path can be established only through links of C and D equipments. The middle way is not used.

The path priority is defined by a number when assigned the path to the tunnel. In the configuration below, *path-option 10* has higher priority over *path-option 20*.

```
!Equipment A
config
mpls traffic-eng
explicit-path primary
  hop 10 ipv4 next-address 203.0.113.2 strict
  hop 20 ipv4 next-address 203.0.113.18 strict
!
explicit-path secondary
  hop 10 ipv4 next-address 203.0.113.6 strict
  hop 20 ipv4 next-address 203.0.113.10 strict
  hop 30 ipv4 next-address 203.0.113.14 strict
!
!
interface tunnel-te 1
name RouterA-RouterE
destination 198.51.100.5
path-option 10 explicit name primary
path-option 20 explicit name secondary
!
commit
```

```
!Equipment E
config
mpls traffic-eng
explicit-path primary
  hop 10 ipv4 next-address 203.0.113.17 strict
  hop 20 ipv4 next-address 203.0.113.1 strict
!
explicit-path secondary
  hop 10 ipv4 next-address 203.0.113.13 strict
  hop 20 ipv4 next-address 203.0.113.9 strict
  hop 30 ipv4 next-address 203.0.113.5 strict
!
!
interface tunnel-te 1
name RouterE-RouterA
destination 198.51.100.1
path-option 10 explicit name primary
path-option 20 explicit name secondary
!
commit
```

For examples of tunnel association to L2VPNs, consult sessions [VPWS with RSVP](#) or [VPLS with RSVP](#).



The available commands for troubleshooting can be found in the topic [Verifying RSVP](#).

11.2.3 Verifying RSVP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```

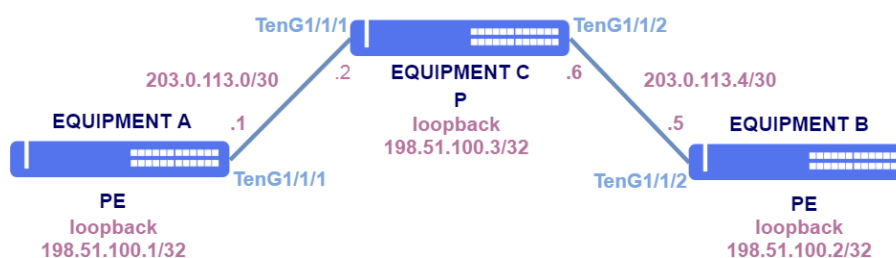
show mpls traffic-eng tunnel-te brief
show mpls traffic-eng tunnel-te id <tunnel-te-id>
show mpls traffic-eng tunnel-te name <lsp_name>
show mpls forwarding-table
show ip ospf database opaque-area

```

11.3 VPWS Configuration

A VPWS (Virtual Private Wire Service) provides the emulation of point-to-point Ethernet services over a MPLS network.

The following topology will be used as base for the examples in this section.



Base topology for VPWS

Loopbacks:

Equipment	Loopback
EQUIPMENT A	198.51.100.1/32
EQUIPMENT B	198.51.100.2/32
EQUIPMENT C	198.51.100.3/32

Addressing between PEs and P:

PE	PE Intf	PE Address	P	P Intf	P Address	VLAN
EQUIP A	TenG1/1/1	203.0.113.1/30	EQUIP C	TenG1/1/2	203.0.113.2/30	10
EQUIP B	TenG1/1/2	203.0.113.5/30	EQUIP C	TenG1/1/1	203.0.113.6/30	20

```

!Equipment A
config
dot1q
vlan 10
interface ten-gigabit-ethernet-1/1/1
untagged
!!
!
switchport
interface ten-gigabit-ethernet-1/1/1

```

```

native-vlan
vlan-id 10
!
!
interface l3 VLAN10
ipv4 address 203.0.113.1/30
lower-layer-if vlan 10
!
interface loopback 0
ipv4 address 198.51.100.1/32
!
!
router ospf 1
router-id 198.51.100.1
area 0
interface l3-VLAN10
network-type point-to-point
!
interface loopback-0
!
!
mpls ldp
lsr-id loopback-0
interface l3-VLAN10
!
!
commit

```

```

!Equipment B
config
dot1q
vlan 20
interface ten-gigabit-ethernet-1/1/2
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/2
native-vlan
vlan-id 20
!
!
!
interface l3 VLAN20
ipv4 address 203.0.113.5/30
lower-layer-if vlan 20
!
interface loopback 0
ipv4 address 198.51.100.2/32
!
!
router ospf 1
router-id 198.51.100.2
area 0
interface l3-VLAN20
network-type point-to-point
!
interface loopback-0
!
!
!
mpls ldp
lsr-id loopback-0
interface l3-VLAN20
!
!
commit

```

```

!Equipment C
config
dot1q
vlan 10
interface ten-gigabit-ethernet-1/1/1
untagged
!
!
!
vlan 20
interface ten-gigabit-ethernet-1/1/2

```

```

    untagged
    !
    !
    !
switchport
interface ten-gigabit-ethernet-1/1/1
    native-vlan
    vlan-id 10
    !
interface ten-gigabit-ethernet-1/1/2
    native-vlan
    vlan-id 20
    !
    !
    !
interface l3 VLAN10
    ipv4 address 203.0.113.2/30
    lower-layer-if vlan 10
    !
interface l3 VLAN20
    ipv4 address 203.0.113.6/30
    lower-layer-if vlan 20
    !
interface loopback 0
    ipv4 address 198.51.100.3/32
    !
    !
router ospf 1
    router-id 198.51.100.3
    area 0
        interface l3-VLAN10
            network-type point-to-point
        !
        interface l3-VLAN20
            network-type point-to-point
        !
        interface loopback-0
    !
    !
mpls ldp
    lsr-id loopback-0
    !
    interface l3-VLAN10
    !
    interface l3-VLAN20
    !
    !
    !
commit

```

VPN signaling happens through LDP protocol. It is necessary to establish a targeted LDP session between PEs, as shown below.

```

!Equipment A
config
mpls ldp
    lsr-id loopback-0
    neighbor targeted 198.51.100.2
    !
    !
commit

```

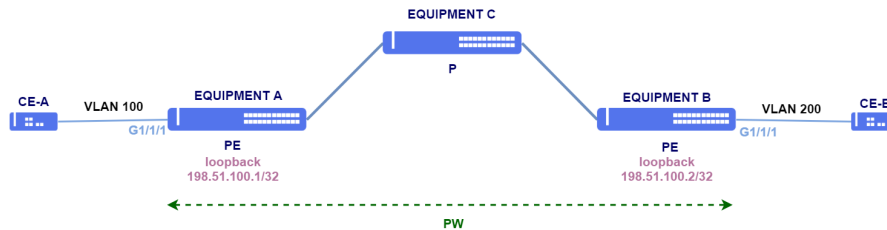
```

!Equipment B
config
mpls ldp
    lsr-id loopback-0
    neighbor targeted 198.51.100.1
    !
    !
commit

```

11.3.1 Configuring a VPWS with PW type VLAN - Case 1

In the following topology, there is a VPNs with tagged access interfaces. The VLAN tags used in both VPN sites are different.



PW type VLAN configuration

Both access interfaces are *tagged*. It means only frames with the specified VLAN tag will be encapsulated and transported.

In this scenario, site A receives frames with VLAN tag 100 and site B received frames with VLAN tag 200. The received frames in site A (EQUIPMENT A) from CE-A contain tag 100. The frames are forwarded to site B (EQUIPMENT B) and the tags are replaced by tag 200 before forwarding them to CE-B. The received frames with tag 200 from CE-B are forwarded to site A (EQUIPMENT) and have their targets replaced with tag 100 before being forwarded to CE-A. In this way, the communication between different VLANs in a VPN is possible.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
!
!
mpls l2vpn
  vpws-group CUSTOMER1
  vpn VPN1
    neighbor 198.51.100.2
    pw-type vlan
    pw-id 100
  !
  access-interface gigabit-ethernet-1/1/1
  dot1q 100
  !
!
!
commit
```

```
!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
!
!
```

```

!
mpls l2vpn
vpws-group CUSTOMER1
vpn VPN1
neighbor 198.51.100.1
pw-type vlan
pw-id 100
!
!
access-interface gigabit-ethernet-1/1/1
dot1q 200
!
!
!
commit

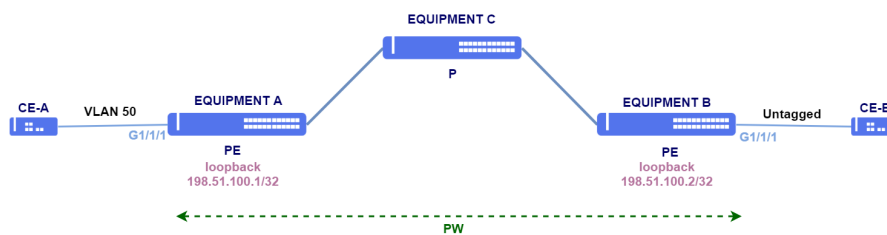
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.2 Configuring a VPWS with VLAN PW type - Case 2

In the scenario below, there is a L2VPN with VLAN pw type and dot1q access interface in site A and untagged access interface in site B.



VPWS configuration with VLAN PW type

Site A access interface is *tagged* and allows only frames with tag 50 to be encapsulated and transported. In site B, the access interface is untagged.

The received frames with tag 50 from CE-A are forwarded to site B (EQUIPMENT B) and have their tags removed before being forwarded to CE-B. Before being forwarded to site A (EQUIPMENT A), tag 200 is added to the untagged frames received from CE-B. Tag 200 is replaced with tag 50 in EQUIPMENT A before being forwarded to CE-A.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.

```

!Equipment A
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.2
!

```

```

!
mpls l2vpn
vpws-group CUSTOMER2
vpn VPN2
neighbor 198.51.100.2
pw-type vlan
pw-id 200
!
!
access-interface gigabit-ethernet-1/1/1
dot1q 50
!
!
!
!
commit

```

```

!Equipment B
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
!
mpls l2vpn
vpws-group CUSTOMER2
vpn VPN2
neighbor 198.51.100.1
pw-type vlan 200
pw-id 200
!
!
access-interface gigabit-ethernet-1/1/1
!
!
!
!
commit

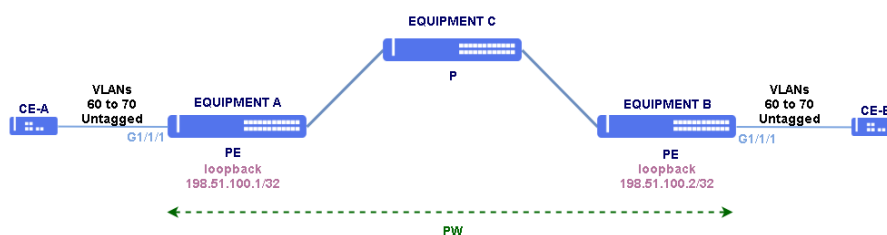
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.3 Configuring a QinQ VPWS with PW type VLAN

In the following topology, there is a QinQ VPN with tagged access interfaces.



QinQ VPWS with VLAN PW type

Both access interfaces are *tagged*. It means only frames with the VLANs tag 60 to 70 will be encapsulated and transported. If necessary, untagged configuration can be added to encapsulate data traffic without a VLAN tag.



Untagged traffic or the PDUs from untagged protocols will be encapsulated in the VPWS only if the untagged configuration is present on the VPN.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
  !
!
mpls l2vpn
  vpws-group CUSTOMER1
  vpn VPN1
    qinq
    neighbor 198.51.100.2
    pw-type vlan 100
    pw-id 100
    !
    !
  access-interface gigabit-ethernet-1/1/1
    encapsulation
    dot1q 60-70
    untagged
  !
  !
!
!
commit
```

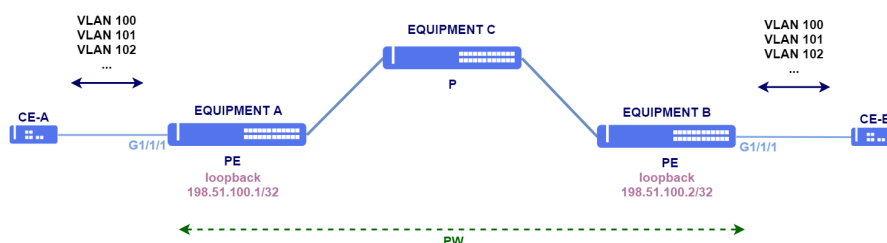
```
!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
  !
!
mpls l2vpn
  vpws-group CUSTOMER1
  vpn VPN1
    qinq
    neighbor 198.51.100.1
    pw-type vlan 100
    pw-id 100
    !
    !
  access-interface gigabit-ethernet-1/1/1
    encapsulation
    dot1q 60-70
    untagged
  !
  !
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.4 Configuring a VPWS with Ethernet PW type - Case 1

In the following scenario, there is a PW type Ethernet VPN. Interfaces do not have dot1q configured. In this case, the VPN becomes *port-based*. Any frame, with any VLAN tag, will be encapsulated and forwarded through the VPN.



PW type Ethernet VPN configuration



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of `pw-mtu` is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
  !
!
mpls l2vpn
vpws-group CUSTOMER1
  vpn VPN3
    neighbor 198.51.100.2
    pw-type ethernet
    pw-id 102
    !
  !
  access-interface gigabit-ethernet-1/1/1
  !
!
commit
```

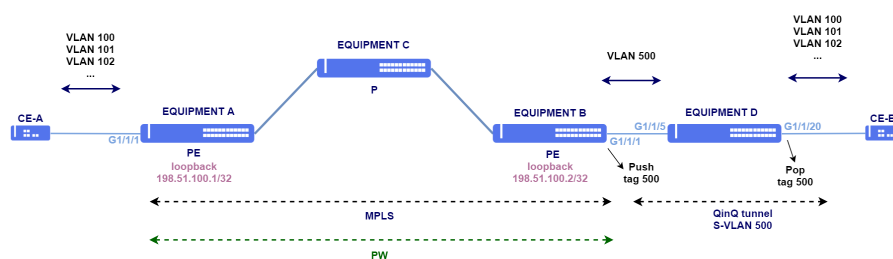
```
!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
  !
!
mpls l2vpn
  vpws-group CUSTOMER1
  vpn VPN3
    neighbor 198.51.100.1
    pw-type ethernet
    pw-id 102
    !
  !
  access-interface gigabit-ethernet-1/1/1
  !
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.5 Configuring a VPWS with Ethernet PW type - Case 2

In the following topology, there is a Ethernet PW VPN. In site A, frames with several VLAN tags are expected. Site B is connected to EQUIPMENT D, a switch configured to work only as layer 2 and encapsulate frames in S-VLAN 500. All transported VLANs are encapsulated in VLAN 50 in site B.



Configuring PW type Ethernet VPN and QinQ

Site A must be configured with Ethernet PW type and as *port-based* mode. Site B must be configured with Ethernet PW type as well and VLAN-based mode (*tagged*), so the 500 tag is pushed to the frame in the egress.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of `pw-mtu` is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
!
!
mpls l2vpn
  vpws-group CUSTOMER1
  vpn VPN4
  neighbor 198.51.100.2
  pw-type ethernet
  pw-id 103
!
!
access-interface gigabit-ethernet-1/1/1
!
!
commit
```

```

!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
  !
!
mpls l2vpn
  vpws-group CUSTOMER1
  vpn VPN4
  neighbor 198.51.100.1
  pw-type ethernet
  pw-id 103
  !
  access-interface gigabit-ethernet-1/1/1
  dot1q 500
  !
!
!
commit

```

```

!Equipment D
config
dot1q
  vlan 500
  interface gigabit-ethernet-1/1/5
  !
  interface gigabit-ethernet-1/1/20
  untagged
  !
!
switchport
  interface gigabit-ethernet-1/1/20
  native-vlan
  vlan-id 500
  !
  qinq
  !
!
commit

```

11.3.6 Configuring a VPWS with RSVP

VPWS can be configured to have its traffic forwarded over RSVP tunnels. The VPN configuration is similar to the examples already demonstrated. The only difference is that the VPWS neighbor is assigned to a tunnel.



L2VPN signaling is still performed through a targeted LDP session. Only L2VPN traffic is forwarded through tunnels.

In the following example, tunnel 10 has been assigned to neighbor 198.51.100.2. Traffic destined to that neighbor will be forwarded over this tunnel.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.

```

config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
  !
!
mpls l2vpn
  vpws-group CUSTOMER1
  vpn VPN1
    neighbor 198.51.100.2
    pw-type vlan
    pw-id 100
    tunnel-interface tunnel-te-10
  !
  access-interface gigabit-ethernet-1/1/1
  dot1q 100
  !
!
!
!
commit

```

For more details in RSVP tunnel configuration, please consult session [RSVP Configuration](#).



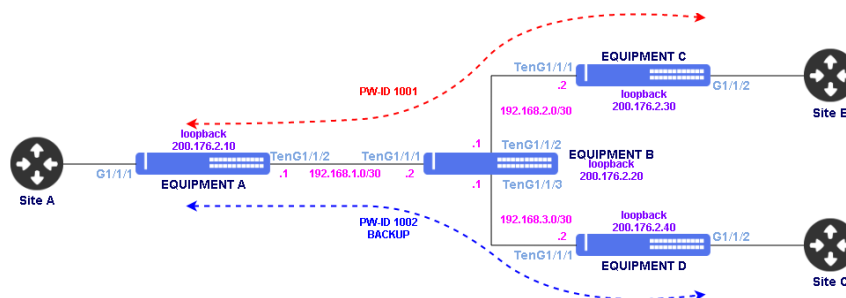
The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.7 Configuring a VPWS with PW Backup

PW Backup is a feature that allows you to configure another Provider Edge (PE) router as the redundancy of an L2VPN circuit. This way, when the main PE is inaccessible, the backup PE will forward the traffic. When the primary PE recovers from the failure, traffic will be routed through it again. The following topology will be used as base for the examples in this section.



After recovering from a failure on the main PE, when the PW is back to the up state, there will be a 30 second delay before returning to the main PE.



VPN configuration with PW Backup

```
!Equipment A
config
dot1q
vlan 10
  interface ten-gigabit-ethernet-1/1/2
  untagged
  !
  !
!
switchport
interface ten-gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 10
  !
  !
!
interface l3 VLAN10
  ipv4 address 192.168.1.1/30
  lower-layer-if vlan 10
  !
!
interface loopback 0
  ipv4 address 200.176.2.10/32
  !
!
router ospf 1
  router-id 200.176.2.10
  area 0
    interface l3-VLAN10
    network-type point-to-point
    !
    interface loopback-0
    !
  !
!
mpls ldp
  lsr-id loopback-0
  interface l3-VLAN10
  !
!
!
commit
```

```
!Equipment C
config
dot1q
vlan 20
  interface ten-gigabit-ethernet-1/1/1
  untagged
  !
  !
!
switchport
interface ten-gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 20
  !
  !
!
interface l3 VLAN20
  ipv4 address 192.168.2.2/30
  lower-layer-if vlan 20
  !
!
interface loopback 0
  ipv4 address 200.176.2.30/32
  !
!
router ospf 1
  router-id 200.176.2.30
  area 0
    interface l3-VLAN20
    network-type point-to-point
    !
    interface loopback-0
    !
  !
!
mpls ldp
  lsr-id loopback-0
  interface l3-VLAN20
  !
!
!
commit
```

```

!Equipment D
config
dot1q
vlan 30
interface ten-gigabit-ethernet-1/1/1
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 30
!
!
!
interface l3 VLAN30
ipv4 address 192.168.3.2/30
lower-layer-if vlan 30
!
!
interface loopback 0
ipv4 address 200.176.2.40/32
!
!
router ospf 1
router-id 200.176.2.40
area 0
interface l3-VLAN30
network-type point-to-point
!
interface loopback-0
!
!
!
mpls ldp
lsr-id loopback-0
interface l3-VLAN30
!
!
!
commit

```

In the scenario above, there is a PW type VLAN VPN. In this case, the VPN will only encapsulate the traffic on the S-VLAN 500. It is possible to combine VLAN-based and port-based interfaces to achieved the desired behavior. Below the VPN configuration of the scenario above.



The VPN configured with PW Backup does not support MPLS-TE with RSVP.



The Backup PW feature uses the PW Status TLV to signal the main and backup PW information, for this reason it is necessary that neighbors have support for PW Status TLV and that the use of this TLV is enabled in L2VPN.



The Backup PW feature does not support the backup configuration on both PEs, so only one PE can have backup configured.



After recovering from a failure on the main PE, when the PW is back to the up state, there will be a 30 second delay before returning to the main PE.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 200.176.2.30
  !
  neighbor targeted 200.176.2.40
  !
!
mpls l2vpn
  vpws-group pw-backup
  vpn pw-backup-vlan-2012
  neighbor 200.176.2.30
  pw-type vlan
  pw-id 1001
  !
  access-interface gigabit-ethernet-1/1/1
  dot1q 500
  !
  backup-neighbor 200.176.2.40
  pw-id 1002
  !
!
!
commit
```

```
!Equipment C
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 200.176.2.10
  !
!
mpls l2vpn
  vpws-group pw-backup
  vpn pw-backup-vlan-2012
  neighbor 200.176.2.10
  pw-type vlan
  pw-id 1001
  !
  access-interface gigabit-ethernet-1/1/1
  dot1q 500
  !
!
!
commit
```

```
!Equipment D
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 200.176.2.10
  !
!
mpls l2vpn
  vpws-group pw-backup
  vpn pw-backup-vlan-2012
  neighbor 200.176.2.10
  pw-type vlan
  pw-id 1002
  !
  access-interface gigabit-ethernet-1/1/1
  dot1q 500
  !
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.8 Enabling FAT in a L2VPN

L2VPNs are usually considered as a unique flow by LAG (Link Aggregation) hashing mechanisms, which will result in the traffic not being balanced satisfactorily. The *Flow-Aware Transport* (FAT) has as objective the increase of traffic entropy by adding a new label called *Flow Label*. The new label will help LAG hashing algorithm to make a more efficient load balance.

FAT can be enabled on packet reception, packet transmission or both. If the neighbor is not configured accordingly, the VPN will be established but the FAT feature will be disabled.



Changing FAT configuration will cause a brief interruption in L2VPN traffic due label change in L2VPN.

In the example below, FAT is configured in a VPWS. Also it is possible to set FAT in VPLS.

```
config
mpls l2vpn
mpls ldp
  lsr-id loopback-0
  neighbor targeted 20.20.20.20
  !
!
!
vpws-group VPWS-DATACOM
vpn VPN1
  neighbor 20.20.20.20
  pw-type vlan
  pw-load-balance
  flow-label both
  !
  pw-id 10
  !
  access-interface gigabit-ethernet-1/1/5
  dot1q 100
  !
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.9 Configuring a VPWS with GPON access

Using an MPLS solution directly on the OLTs allows you to transport data for all customers without having to configure their VLANs on all equipment in the L2 network. Another significant gain is the implementation of VPN based access services for site-to-site solutions. In this way it is possible to provide GPON access with protocol transparency and traffic isolation between clients using OLT's MPLS feature for the establishment of these VPNs.

The following pages will present some scenarios with examples of configurations that can be adopted when using MPLS services with Datacom OLTs. The choice of IP addresses, VPN indexes, VLANs, among others, were used as an example in this document, so they must be adapted to the scenario where they will be applied.

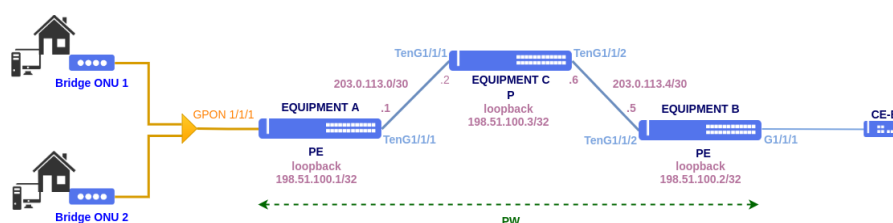


This feature is only supported on the DM4615 and DM4610-HW2 OLTs.



The equipment must have an MPLS license enabled. To configure MPLS license, see the topic [License configuration](#).

The topology below will be used as a basis for the examples in this session.



VPWS with GPON access

The configuration for the GPON service that will be used as the basis for MPLS is available below:

Profile GPON

```
!Equipment A
config
!
profile gpon bandwidth-profile 100Mbps
  traffic type-4 max-bw 100032
!
profile gpon line-profile MPLS
  upstream-fec
  tcont 1 bandwidth-profile 100Mbps
  gem 1
    tcont 1 priority 0
    map 1
    ethernet any vlan 2282 cos any
  !
  gem 2
    tcont 1 priority 0
    map 2
    ethernet any vlan any cos any
  !
!
commit
```

ONU Provisioning

```
!Equipment A
config
!
interface gpon 1/1/1
  onu 1
    serial-number DACM000BBB01
```

```

line-profile MPLS
 ethernet 1
 negotiation
 no shutdown
!
onu 2
 serial-number DACM000BBB02
 line-profile MPLS
 ethernet 1
 negotiation
 no shutdown
!
commit

```

For more details on the GPON configuration, see the section [Configuring GPON Applications](#).

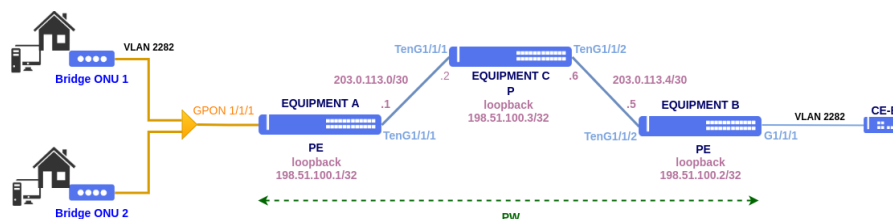
MPLS infrastructure



For more details on configuring the MPLS infrastructure with VPWS, see the topic [VPWS configuration](#).

11.3.10 Configuring a VPWS with PW type VLAN and Service Port access

For this use case, the service-port will be the access interface for a VPWS that is established with the MPLS network. These service-ports can coexist with other service-ports used for other services, such as Internet access, Multicast/IGMP traffic for video traffic or VoIP telephony. It is worth mentioning that the service-port used in VPN can be created in the activation of a new ONU or be used in addition to the services in operation. To this end, the profiles used at the ONU must be edited in order to include all services.



VPWS with Service Port access

To perform this configuration, L2VPN must be configured with PW type VLAN and VLAN Based mode using the service-port as the access interface. Only traffic from VLAN 2282 mapped on the service-port through gem 1 is encapsulated in L2VPN.

Service port

```

!Equipment A
config
service-port 2282 gpon 1/1/1 onu 1 gem 1
commit

```

It is important to note that in cases where service-ports will be used as an access interface for MPLS, the indication of VLAN is not necessary, as shown in the example for service port 2282.

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
!
mpls l2vpn
  vpws-group GPON_VPWS
  vpn 2282
  description VPWS_VLAN_BASED_SP
  neighbor 198.51.100.2
  pw-type vlan
  pw-id 2282
  pw-mtu 2000
!
  access-interface service-port 2282
  dot1q 2282
  mtu 2000
!
commit
```

In the example, **2000 bytes MTU** is configured both on the PW and on the access due to the maximum MTU limit supported by the ONU. This value can be changed as necessary, respecting the limit of each equipment.

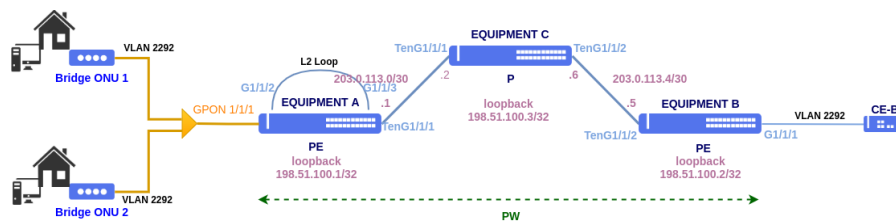
```
!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
!
mpls l2vpn
  vpws-group GPON_VPWS
  vpn 2282
  description VPWS_VLAN_BASED_SP
  neighbor 198.51.100.1
  pw-type vlan
  pw-id 2282
  pw-mtu 2000
!
  access-interface gigabit-ethernet 1/1/1
  dot1q 2282
  mtu 2000
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.3.11 Configuring a VPWS with PW type VLAN and Ethernet Loop access

In the topology below, there is a VPN with tagged access interface using a Ethernet loop. Ethernet loop will be used to group services from different service-ports in L2VPN access, since VPWS only supports one interface in access.



VPN GPON access with Ethernet loop

The GbE-1/1/2 interface will be used as OLT Uplink for GPON services, with the configuration of GPON services carried out in a conventional manner, as if the GbE-1/1/2 interface was in fact OLT's Uplink interface, without taking into account the MPLS part.

After configuring the OLT, the GbE-1/1/2 interface will be physically connected to the GbE-1/1/3 interface creating a physical loop between them, the GbE-1/1/3 interface will be the access interface for a VPWS which will be interconnected with the MPLS network.

For this scenario, it is necessary to map on the OLT the **gigabit-ethernet-1/1/2 interface on VLAN 2292** and create the service-port for the mapping of VLAN 2292 at the ONU.

Service VLAN

```
!Equipment A
config
!
dot1q
  vlan 2292
  interface gigabit-ethernet-1/1/2
!
service vlan 2292
type tls
!
!
commit
```

Service port

```
!Equipment A
config
!
service-port 1
  gpon 1/1/1 onu 1 gem 2 match vlan vlan-id 2292 action vlan replace vlan-id 2292
!
service-port 2
  gpon 1/1/1 onu 2 gem 2 match vlan vlan-id 2292 action vlan replace vlan-id 2292
!
commit
```

In the example above, for cases where the MPLS access uses an Ethernet interface the service-port used for the MPLS performs VLAN match.

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
!
!
mpls l2vpn
vpn 2292
```

```
description VPWS_VLAN_BASED_ETH
neighbor 198.51.100.2
pw-type vlan
pw-id 2292
pw-mtu 2000
!
access-interface gigabit-ethernet-1/1/3
dot1q 2292
mtu 2000
!
!
commit
```

In the example, **2000 bytes MTU** is configured both on the PW and on the access due to the maximum MTU limit supported by the ONU. This value can be changed as necessary, respecting the limit of each equipment.

```
!Equipment B
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
mpls l2vpn
vpws-group GPON_VPWS
vpn 2292
description VPWS_VLAN_BASED_ETH
neighbor 198.51.100.1
pw-type vlan
pw-id 2292
pw-mtu 2000
!
access-interface gigabit-ethernet 1/1/1
dot1q 2292
mtu 2000
!
!
commit
```

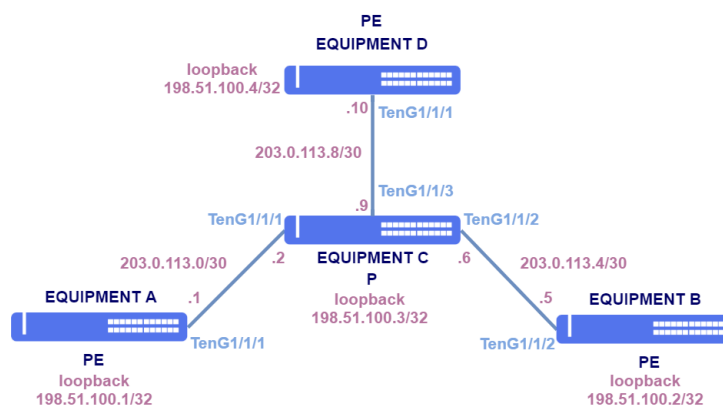


The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4 VPLS Configuration

VPLS (Virtual Private LAN Service) is a L2VPN service that interconnects networks in different sites through an IP/MPLS network, making them in the same broadcast domain. VPLS emulates a point-to-multipoint Ethernet service.

The following topology will be used as base for the examples in this section.



VPLS topology

Loopbacks:

Equipment	Loopback
EQUIPMENT A	198.51.100.1/32
EQUIPMENT B	198.51.100.2/32
EQUIPMENT C	198.51.100.3/32
EQUIPMENT D	198.51.100.4/32

Addressing between PEs and P:

PE	PE Intf	PE Address	P	P Intf	P Address	VLAN
EQUIP A	TenG1/1/1	203.0.113.1/30	EQUIP C	TenG1/1/2	203.0.113.2/30	10
EQUIP B	TenG1/1/2	203.0.113.5/30	EQUIP C	TenG1/1/1	203.0.113.6/30	20
EQUIP D	TenG1/1/1	203.0.113.9/30	EQUIP C	TenG1/1/3	203.0.113.10/30	30

```

!Equipment A
config
dot1q
vlan 10
interface ten-gigabit-ethernet-1/1/1
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 10
!
!
!
interface l3 VLAN10
ipv4 address 203.0.113.1/30
lower-layer-if vlan 10
!

```

```

!
interface loopback 0
  ipv4 address 198.51.100.1/32
!
!
router ospf 1
  router-id 198.51.100.1
  area 0
    interface l3-VLAN10
      network-type point-to-point
    !
    interface loopback-0
    !
  !
!
mpls ldp
  lsr-id loopback-0
  interface l3-VLAN10
  !
!
!
commit

```

```

!Equipment B
config
dot1q
  vlan 20
    interface ten-gigabit-ethernet-1/1/2
      untagged
    !
    !
  !
!
switchport
  interface ten-gigabit-ethernet-1/1/2
    native-vlan
    vlan-id 20
  !
  !
!
interface l3 VLAN20
  ipv4 address 203.0.113.5/30
  lower-layer-if vlan 20
  !
!
!
interface loopback 0
  ipv4 address 198.51.100.2/32
  !
!
router ospf 1
  router-id 198.51.100.2
  area 0
    interface l3-VLAN20
      network-type point-to-point
    !
    interface loopback-0
    !
  !
!
mpls ldp
  lsr-id loopback-0
  interface l3-VLAN20
  !
!
!
commit

```

```

!Equipment C
config
dot1q
  vlan 10
    interface ten-gigabit-ethernet-1/1/1
      untagged
    !
    !
  !
  vlan 20
    interface ten-gigabit-ethernet-1/1/2
      untagged
    !
    !
  !
  vlan 30
    interface ten-gigabit-ethernet-1/1/3
      untagged
    !
    !
  !
!

```

```

!
switchport
interface ten-gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 10
!
!
interface ten-gigabit-ethernet-1/1/2
  native-vlan
  vlan-id 20
!
!
interface ten-gigabit-ethernet-1/1/3
  native-vlan
  vlan-id 30
!
!
!
interface l3 VLAN10
  ipv4 address 203.0.113.2/30
  lower-layer-if vlan 10
!
interface l3 VLAN20
  ipv4 address 203.0.113.6/30
  lower-layer-if vlan 20
!
interface l3 VLAN30
  ipv4 address 203.0.113.10/30
  lower-layer-if vlan 30
!
interface loopback 0
  ipv4 address 198.51.100.3/32
!
!
router ospf 1
  router-id 198.51.100.3
  area 0
    interface l3-VLAN10
      network-type point-to-point
    !
    interface l3-VLAN20
      network-type point-to-point
    !
    interface l3-VLAN30
      network-type point-to-point
    !
    interface loopback-0
!
!
!
mpls ldp
  lsr-id loopback-0
  interface l3-VLAN10
  interface l3-VLAN20
  interface l3-VLAN30
!
!
commit

```

```

!Equipment D
config
dot1q
vlan 30
  interface ten-gigabit-ethernet-1/1/1
    untagged
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
  native-vlan
  vlan-id 30
!
!
!
interface l3 VLAN30
  ipv4 address 203.0.113.9/30
  lower-layer-if vlan 30
!
!
interface loopback 0

```



```

ipv4 address 198.51.100.4/32
!
router ospf 1
router-id 198.51.100.4
area 0
interface l3-VLAN30
network-type point-to-point
!
interface loopback-0
!
!
mpls ldp
lsr-id loopback-0
interface l3-VLAN30
!
!
commit

```

VPN signaling happens through LDP protocol. It is necessary to establish a targeted LDP session between PEs, as shown below.

```

!Equipment A
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.2
!
neighbor targeted 198.51.100.4
!
!
commit

```

```

!Equipment B
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
neighbor targeted 198.51.100.4
!
!
commit

```

```

!Equipment D
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
neighbor targeted 198.51.100.2
!
!
commit

```

11.4.1 Configuring a VPLS with Ethernet PW type

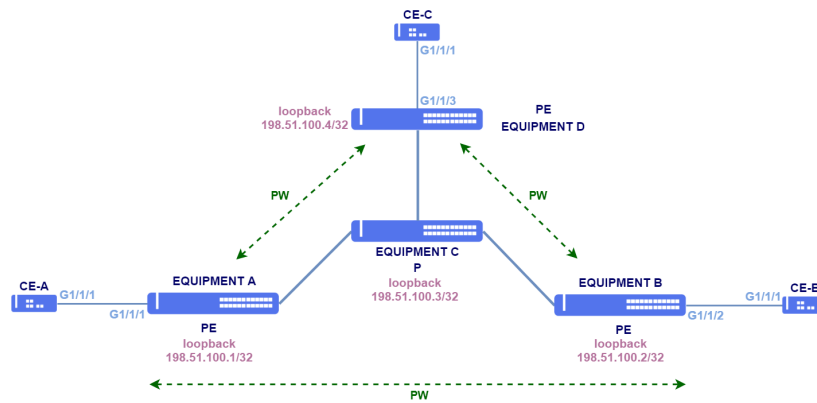
A PW type Ethernet VPN is shown in the following topology. It has port-based interfaces. Any frame arriving in the access interfaces will be transported.



As shown in section [VPWS Configuration](#), it is possible to combine VLAN-based and port-based interfaces to achieve the desired behavior of the VPLS.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.



VPLS with Ethernet PW type

```
!Equipment A
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.2
!
neighbor targeted 198.51.100.4
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type ethernet
neighbor 198.51.100.2
pw-id 100
!
neighbor 198.51.100.4
pw-id 101
!
bridge-domain
access-interface gigabit-ethernet-1/1/1
!
!
!
commit
```

```
!Equipment B
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
neighbor targeted 198.51.100.4
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type ethernet
neighbor 198.51.100.1
pw-id 100
```

```

!
neighbor 198.51.100.4
pw-id 103
!
!
bridge-domain
access-interface gigabit-ethernet-1/1/2
!
!
!
!
!
commit

```

```

!Equipment D
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
neighbor targeted 198.51.100.2
!
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type ethernet
neighbor 198.51.100.1
pw-id 101
!
neighbor 198.51.100.2
pw-id 103
!
!
bridge-domain
access-interface gigabit-ethernet-1/1/3
!
!
!
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4.2 Configuring a VPLS with VLAN PW type - Case 1

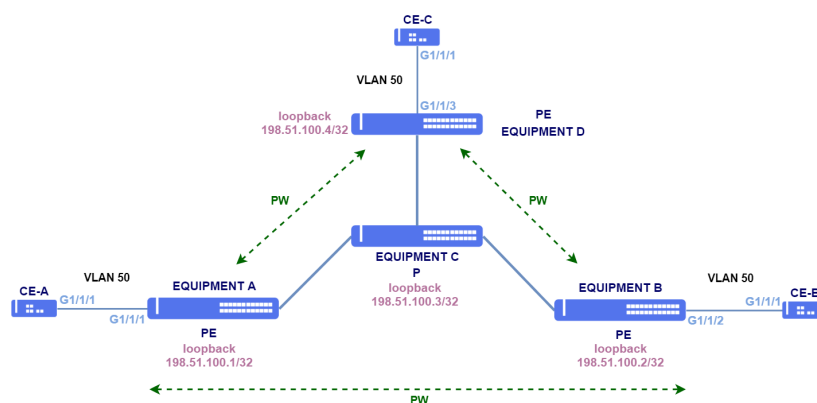
The following topology has a VLAN PW type VPLS with tagged interface access (VLAN based). Only frame with the specified VLAN tag will be transported.



As shown in section [VPWS Configuration](#), it is possible to combine VLAN-based and port-based interfaces to achieved the desired behavior of the VPLS.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.



VPLS with VLAN PW type

```

!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
  !
  neighbor targeted 198.51.100.4
  !
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
  vfi
  pw-type vlan
  neighbor 198.51.100.2
  pw-id 100
  !
  neighbor 198.51.100.4
  pw-id 101
  !
  bridge-domain
  dot1q 50
  access-interface gigabit-ethernet-1/1/1
  !
!
!
commit

```

```

!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
  !
  neighbor targeted 198.51.100.4
  !
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
  vfi
  pw-type vlan
  neighbor 198.51.100.1
  pw-id 100
  !
  neighbor 198.51.100.4
  pw-id 103
  !
  bridge-domain
  dot1q 50
  access-interface gigabit-ethernet-1/1/2
  !
!
!

```

```
!
commit
```

```
!Equipment D
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
  !
  neighbor targeted 198.51.100.2
  !
!
mpls l2vpn
  vpls-group CUSTOMER1
  vpn VPN1
    vfi
      pw-type vlan
      neighbor 198.51.100.1
      pw-id 101
      !
      neighbor 198.51.100.2
      pw-id 103
      !
    !
  !
  bridge-domain
  dot1q 50
  access-interface gigabit-ethernet-1/1/3
  !
!
!
commit
```

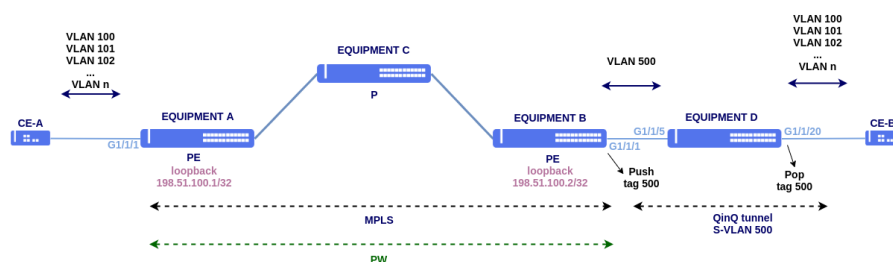


The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4.3 Configuring a VPLS with VLAN PW type - Case 2

The following topology has a VPLS with PW type VLAN, port-based VPN at site A and VLAN-based VPN at site B.

Any frame received at CE-A is forwarded to PE-B (EQUIPMENT B), but will have the VLAN 500 tag added before being sent to EQUIPMENT B. EQUIPMENT B delivers to EQUIPMENT D all traffic sent by CE-A with SVLAN 500 and CVLANs. EQUIPMENT D removes the VLAN 500 and delivers traffic with the CVLANs to CE-B.



VPLS with VLAN PW type and SVLAN

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
```

```
neighbor targeted 198.51.100.2
!!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type vlan 500
neighbor 198.51.100.2
pw-id 100
!!
bridge-domain
access-interface gigabit-ethernet-1/1/1
!!
!!
commit
```

```
!Equipment B
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type vlan
neighbor 198.51.100.1
pw-id 100
!!
bridge-domain
dot1q 500
access-interface gigabit-ethernet-1/1/1
!!
!!
commit
```

```
!Equipment D
config
dot1q
vlan 500
interface gigabit-ethernet-1/1/5
!!
interface gigabit-ethernet-1/1/20
untagged
!!
!!
switchport
interface gigabit-ethernet-1/1/20
native-vlan
vlan-id 500
!!
qinq
!!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4.4 Configuring a QinQ VPLS with VLAN PW type - Case 1

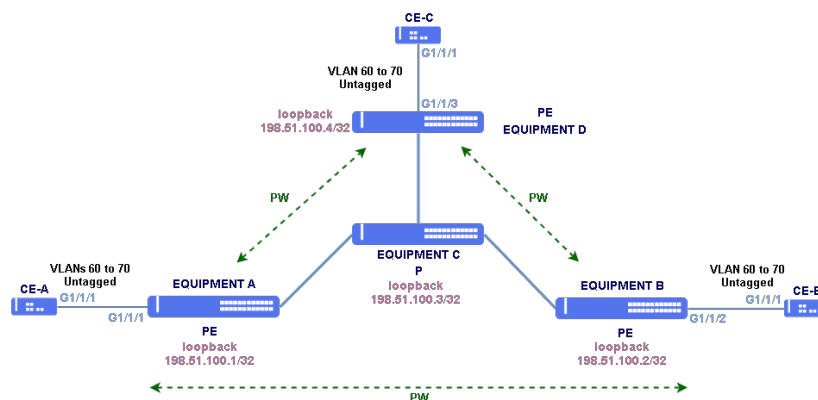
The following topology has a VLAN QinQ PW type VPLS with tagged interface access (VLAN based). Only frames with the specified VLANs tags will be transported. If necessary, **untagged** configuration can be added to encapsulate data traffic without a VLAN tag.



As shown in section [VPWS Configuration](#), it is possible to combine VLAN-based and port-based interfaces to achieved the desired behavior of the VPLS.



The MTU value configured in the PW is used exclusively for signaling and must be equal among both neighbors in the VPN. If the value of pw-mtu is not specified, the value that will be considered will be the one specified in the AC (access-interface) that by default uses 9198 Bytes.



QinQ VPLS with VLAN PW type

```
!Equipment A
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.2
!
neighbor targeted 198.51.100.4
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type vlan 50
neighbor 198.51.100.2
pw-id 100
!
neighbor 198.51.100.4
pw-id 101
!
!
bridge-domain
qinq
access-interface gigabit-ethernet-1/1/1
encapsulation
dot1q 60-70
```

```
commit
!
!
!
!
untagged
```

```

!Equipment B
config
mpls ldp
  lsr-id loopback-0
    neighbor targeted 198.51.100.1
    !
  neighbor targeted 198.51.100.4
  !
!
!
mpls l2vpn
vpls-group CUSTOMER1
  vpn VPN1
    vfi
      pw-type vlan 50
      neighbor 198.51.100.1
      pw-id 100
      !
      neighbor 198.51.100.4
      pw-id 103
      !
    !
  !
  bridge-domain
    qinq
      access-interface gigabit-ethernet-1/1/2
      encapsulation
        dot1q 60-70
        untagged
      !
    !
  !
!
!
commit

```

```

!Equipment D
config
mpls ldp
  lsr-id loopback-0
    neighbor targeted 198.51.100.1
    !
  neighbor targeted 198.51.100.2
  !
!
!
mpls l2vpn
vpls-group CUSTOMER1
  vpn VPN1
    vfi
      pw-type vlan 50
      neighbor 198.51.100.1
      pw-id 101
      !
      neighbor 198.51.100.2
      pw-id 103
      !
    !
  !
  bridge-domain
    qinq
      access-interface gigabit-ethernet-1/1/3
      encapsulation
        dot1q 60-70
        untagged
      !
    !
  !
!
!
commit

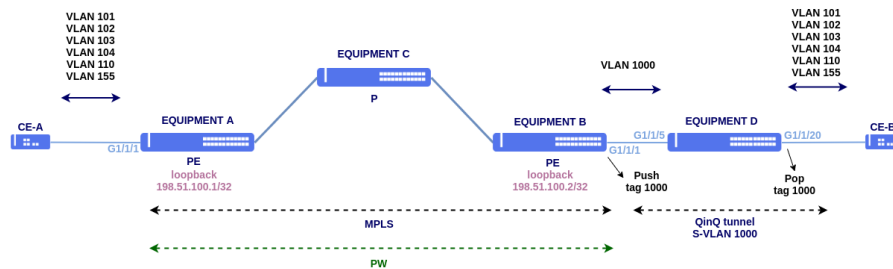
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4.5 Configuring a QinQ VPLS with VLAN PW type - Case 2

The following topology contains QinQ VPLS with PW type VLAN and VLAN-based type access interfaces. Only frames with the specified VLAN tags will be transported by VPLS. When EQUIPMENT B delivers traffic to EQUIPMENT D, it will keep SVLAN 1000. EQUIPMENT D will remove the SVLAN and deliver traffic to CE-B only with the CVLAN received on EQUIPMENT A.



QinQ VPLS with VLAN PW type and SVLAN

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
  vfi
  pw-type vlan 1000
  neighbor 198.51.100.2
  pw-id 100
!
!
bridge-domain
qinq
access-interface gigabit-ethernet-1/1/1
encapsulation
  dot1q 101-104,110,155
!
!
!
commit
```

```
!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
  vfi
  pw-type vlan
  neighbor 198.51.100.1
  pw-id 100
!
!
bridge-domain
dot1q 1000
access-interface gigabit-ethernet-1/1/1
!
!
!
```

```
!
commit
```

```
!Equipment D
config
dot1q
vlan 1000
interface gigabit-ethernet-1/1/5
!
interface gigabit-ethernet-1/1/20
untagged
!
!
!
switchport
interface gigabit-ethernet-1/1/20
native-vlan
vlan-id 1000
!
qinq
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

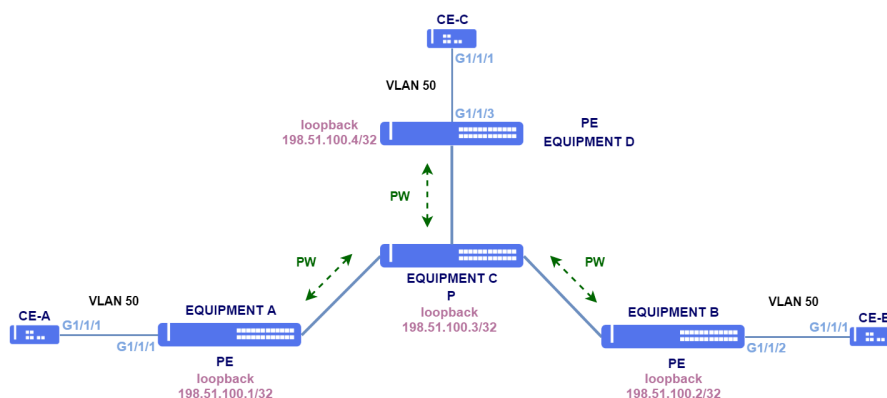
11.4.6 Configuring a H-VPLS

Hierarchical VPLS (H-VPLS) divides a VPLS in a backbone domain and edge domains, reducing the number of PWs.

The following topology exemplifies a VPLS in which PEs establish one PW with a single main equipment, simplifying the configuration and decreasing the number of signaled PWs.



As shown in section [VPWS Configuration](#), it is possible to combine VLAN-based and port-based interfaces to achieved the desired behavior of the VPLS.



H-VPLS

```
!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.3
  !
!
mpls l2vpn
vpls-group CUSTOMER1
  vpn VPN1
  vfi
  pw-type vlan
  neighbor 198.51.100.3
  pw-id 100
  !
  !
  bridge-domain
  dot1q 50
  access-interface gigabit-ethernet-1/1/1
  !
  !
!
!
commit
```

```
!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.3
  !
!
mpls l2vpn
vpls-group CUSTOMER1
  vpn VPN1
  vfi
  pw-type vlan
  neighbor 198.51.100.3
  pw-id 100
  !
  !
  bridge-domain
  dot1q 50
  access-interface gigabit-ethernet-1/1/2
  !
  !
!
!
commit
```

```
!Equipment C
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
  !
  neighbor targeted 198.51.100.2
  !
  neighbor targeted 198.51.100.4
  !
!
mpls l2vpn
vpls-group CUSTOMER1
  vpn VPN1
  vfi
  pw-type vlan
  neighbor 198.51.100.1
  pw-id 100
  split-horizon disable
  !
  neighbor 198.51.100.2
  pw-id 100
  split-horizon disable
  !
  neighbor 198.51.100.4
  pw-id 100
  split-horizon disable
  !
  !
!
!
commit
```

```
!Equipment D
config
mpls ldp
    lsr-id loopback-0
    neighbor targeted 198.51.100.3
!
mpls l2vpn
vpls-group CUSTOMER1
    vpn VPN1
        vfi
            pw-type vlan
            neighbor 198.51.100.3
            pw-id 100
        !
    !
    bridge-domain
        dot1q 50
        access-interface gigabit-ethernet-1/1/3
    !
!
!
commit
```

11.4.7 VPLS with RSVP

VPLS can be configured to have their traffic forwarded over RSVP tunnels. The VPN configuration is similar to the examples already demonstrated. The only difference is that the VPLS neighbor is assigned to a tunnel.



L2VPN signaling is still performed through a targeted LDP session. Only L2VPN traffic is forwarded through tunnels.

In the following example, tunnel 10 has been assigned to neighbor 198.51.100.2. Traffic destined to that neighbor will be forwarded over this tunnel.

```

config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
!
mpls l2vpn
vpls-group CUSTOMER1
  vpn VPN1
    vfi
      pw-type ethernet
      neighbor 192.51.100.2
        pw-id 100
      tunnel-interface tunnel-te-10
    !
    neighbor 192.51.100.4
      pw-id 101
    !
  !
bridge-domain
  access-interface gigabit-ethernet-1/1/1
!
!
!
commit

```

For more details in RSVP tunnel configuration, please consult session [RSVP Configuration](#)



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4.8 Enabling TLS in a VPLS

TLS (Transparent Lan Service) is used to transport L2 protocols PDUs in a VPLS. To encapsulate the PDUs in both directions is necessary to configure the TLS in all PEs used by L2VPN.

```
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.3
  !
!
mpls l2vpn
  vpls-group CUSTOMER1
  vpn VPN1
    vfi
      pw-type vlan
      neighbor 198.51.100.3
      pw-id 100
    !
    bridge-domain
      dot1q 50
      transparent-lan-service
      access-interface gigabit-ethernet-1/1/3
    !
  !
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4.9 Limiting MAC addresses in a VPLS

VPLS has MAC address learning and as needed it is possible to limit the number of MAC addresses for each VPLS.



MAC address table is shared between L2, L3 and MPLS. This feature it is not supported in all platforms. On these platforms the limit is the value of MAC address supported by the platform. For more details of feature support, see **Datasheet**.

Platforms that support the limit have the default value of 1024 MAC addresses per VPLS, being possible to change as needed.

The action for the MAC address limit in VPLS is drop, that is, if the value is exceeded the traffic will be dropped except in the case where MAC is learned on access or uplink. In this case the flow is forwarded by flood.

In VPLS below, the limit has been set at 5000 MAC addresses in VPLS.

```
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.3
  !
  !
mpls l2vpn
  vpls-group CUSTOMER1
  vpn VPN1
    vfi
      pw-type vlan
      neighbor 198.51.100.3
      pw-id 100
      !
    bridge-domain
      dot1q 50
      transparent-lan-service
      mac-limit 5000
      access-interface gigabit-ethernet-1/1/3
      !
    !
  !
  !
commit
```

11.4.10 Configuring VPLS with GPON access

Using an MPLS solution directly on the OLTs allows you to transport data for all customers without having to configure their VLANs on all equipment in the L2 network. Another significant gain is the implementation of VPN based access services for site-to-site solutions. In this way it is possible to provide GPON access with protocol transparency and traffic isolation between clients using OLT's MPLS feature for the establishment of these VPNs.

The following pages will present some scenarios with examples of configurations that can be adopted when using MPLS services with Datacom OLTs. The choice of IP addresses, VPN indexes, VLANs, among others, were used as an example in this document, so they must be adapted to the scenario where they will be applied.

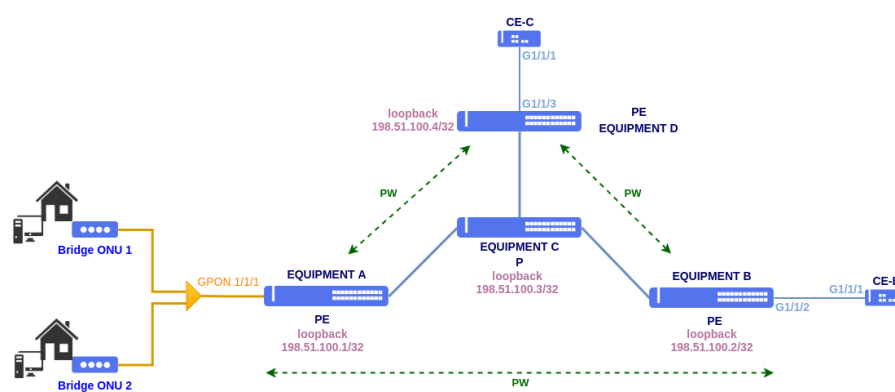


This feature is only supported on the DM4615 and DM4610-HW2 OLTs.



The equipment must have an MPLS license enabled. To configure MPLS license, see the topic [License configuration](#).

The topology below will be used as a basis for the examples in this session.



MPLS with GPON access

The configuration for the GPON service that will be used as the basis for MPLS is available below:

Profile GPON

```
!Equipment A
config
!
profile gpon bandwidth-profile 100Mbps
  traffic type-4 max-bw 100032
!
profile gpon line-profile MPLS
  upstream-fec
  tcont 1 bandwidth-profile 100Mbps
  gem 1
    tcont 1 priority 0
    map 1
    ethernet any vlan 50 cos any
  !
  gem 2
    tcont 1 priority 0
    map 2
    ethernet any vlan 222 cos any
  !
!
commit
```

ONU Provisioning

```
!Equipment A
config
!
interface gpon 1/1/1
  onu 1
    serial-number DACM000BBB01
    line-profile MPLS
    ethernet 1
    negotiation
    no shutdown
  !
  onu 2
    serial-number DACM000BBB02
    line-profile MPLS
    ethernet 1
    negotiation
    no shutdown
  !
!
commit
```

For more details on the GPON configuration, see the section [Configuring GPON Applications](#).

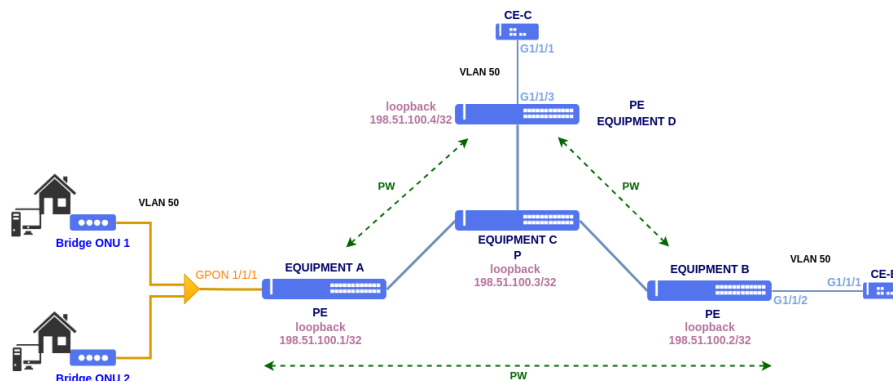
MPLS infrastructure



For more details on configuring the MPLS infrastructure with VPLS, see the topic [VPLS configuration](#).

11.4.11 Configuring VPLS with PW type VLAN and Service Port access

The topology below contains a VPLS with PW type VLAN and VLAN-based type service-port access interface. Only frames with the specified VLAN tag and mapped to the service-port will be forwarded over VPLS.



VPLS with Service Port access



PPPoE and DHCP circuits information will be not included in L2VPN encapsulated packets. If you want to include the circuit information, the recommended scenario is: [Configuring a VPWS with PW type VLAN and Ethernet Loop access](#).

To perform this configuration, L2VPN must be configured with PW type VLAN and VLAN Based mode using the service-port as access interface. Only traffic from VLAN 50 mapped to the service-port through gem 1 is encapsulated in L2VPN.

Service port

```
!Equipment A
config
service-port 50 gpon 1/1/1 onu 1 gem 1
commit
```

It is important to note that in cases where the service-ports will be used as an access interface for MPLS, it will not be necessary to indicate a VLAN, as shown in the example for service-port 50.

```
!Equipment A
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.2
!
neighbor targeted 198.51.100.4
!
```



```

!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type vlan
neighbor 198.51.100.2
pw-id 100
!
neighbor 198.51.100.4
pw-id 101
!
!
bridge-domain
dot1q 50
transparent lan-service
access-interface service-port-50
!
!
!
!
!
commit

```

```

!Equipment B
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
neighbor targeted 198.51.100.4
!
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type vlan
neighbor 198.51.100.1
pw-id 100
!
neighbor 198.51.100.4
pw-id 103
!
!
bridge-domain
dot1q 50
access-interface gigabit-ethernet-1/1/2
!
!
!
!
!
commit

```

```

!Equipment D
config
mpls ldp
lsr-id loopback-0
neighbor targeted 198.51.100.1
!
neighbor targeted 198.51.100.2
!
!
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
vfi
pw-type vlan
neighbor 198.51.100.1
pw-id 101
!
neighbor 198.51.100.2
pw-id 103
!
!
bridge-domain
dot1q 50
access-interface gigabit-ethernet-1/1/3
!
!
!
!
!
commit

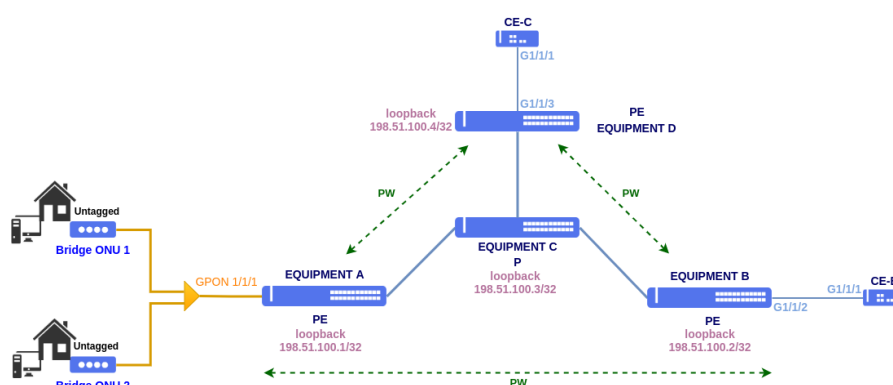
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.4.12 Configuring VPLS with PW type Ethernet and Service Port access

The topology below contains a VPLS with PW type Ethernet and service-port access interface. Any frame arriving at the service-port access interfaces will be transported over the VPLS.



VPLS with Service Port access



PPPoE and DHCP circuits information will be not included in L2VPN encapsulated packets. If you want to include the circuit information, the recommended scenario is: [Configuring a VPWS with PW type VLAN and Ethernet Loop access](#).

To perform this configuration, L2VPN must be configured with PW type Ethernet and Port Based mode using the service-port as the access interface. All traffic coming from the service-port through gem is encapsulated in L2VPN.

Service port

```
!Equipment A
config
service-port 221 gpon 1/1/1 onu 1 gem 2
!
service-port 222 gpon 1/1/1 onu 2 gem 2
commit
```

The native vlan configuration at the ONU will be used to deliver the service without a VLAN tag over the ONU Ethernet.

ONU Provisioning

```

!Equipment A
config
!
interface gpon 1/1/1
onu 1
  serial-number DACM000BBB01
  line-profile MPLS
  ethernet 1
  negotiation
  native vlan vlan-id 10
!
onu 2
  serial-number DACM000BBB02
  line-profile MPLS
  ethernet 1
  negotiation
  native vlan vlan-id 10
!
commit

```

LDP e L2VPN

```

!Equipment A
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.2
  !
  neighbor targeted 198.51.100.4
  !
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
  vfi
  pw-type ethernet
  neighbor 198.51.100.2
  pw-id 100
  !
  neighbor 198.51.100.4
  pw-id 101
  !
  !
  bridge-domain
  transparent lan-service
  access-interface service-port-221
  !
  access-interface service-port-222
  !
!
!
commit

```

```

!Equipment B
config
mpls ldp
  lsr-id loopback-0
  neighbor targeted 198.51.100.1
  !
  neighbor targeted 198.51.100.4
  !
!
mpls l2vpn
vpls-group CUSTOMER1
vpn VPN1
  vfi
  pw-type ethernet
  neighbor 198.51.100.1
  pw-id 100
  !
  neighbor 198.51.100.4
  pw-id 103
  !
  !
  bridge-domain
  access-interface gigabit-ethernet-1/1/2
  !
!
!
commit

```

```
!Equipment D
config
mpls ldp
  lsr-id loopback-0
    neighbor targeted 198.51.100.1
    !
    neighbor targeted 198.51.100.2
    !
  !
!
mpls l2vpn
vpls-group CUSTOMER1
  vpn VPN1
    vfi
      pw-type ethernet
      neighbor 198.51.100.1
      pw-id 101
      !
      neighbor 198.51.100.2
      pw-id 103
      !
    !
  !
  bridge-domain
    access-interface gigabit-ethernet-1/1/3
    !
  !
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L2VPN](#).

11.5 Verifying L2VPN

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show mpls l2vpn hardware
show mpls l2vpn vpw-group brief
show mpls l2vpn vpw-group detail
show mpls l2vpn vpls-group brief
show mpls l2vpn vpls-group detail
show mpls l2vpn vpls-group mac-address-table
show mpls ldp database
show mpls ldp neighbor
show mpls ldp parameters
show mpls forwarding-table
```

11.6 L3VPN Configuration

11.6.1 Configuring a L3VPN Site-to-Site

While L2VPN provides transparent layer 2 services, in L3VPN the routing is performed by ISP equipment. Packets are forwarded using MPLS labels. BGP is the protocol used to exchange route and label information between PEs.

Each route is identified by a route-distinguisher (RD), which must be unique for each client, allowing IP addresses overlapping between different customers. Routes are marked with BGP communities called route-targets, which are used to define in which VPN each route will be installed.



It is mandatory to have LDP protocol configured and running in the network to be able to use L3VPN service.

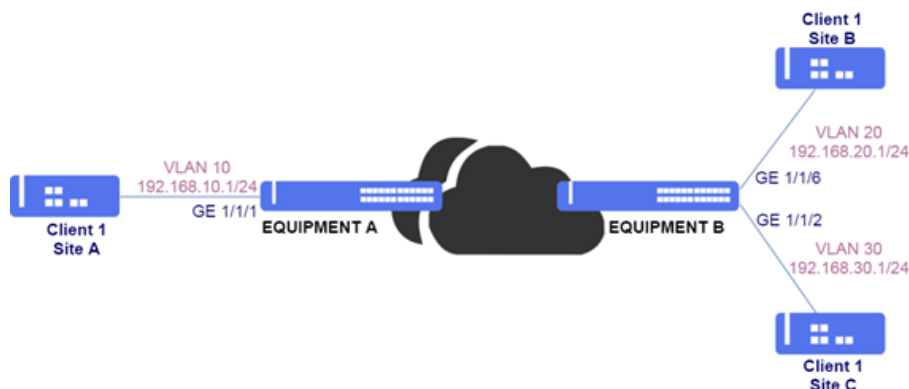


Despite the route-distinguisher format being similar to the route-target format, both are independent and have different functions.



BGP is responsible for label and route exchange between PEs. It is necessary to enable the VPNv4 family in BGP protocol.

In the following topology, two PEs (EQUIPMENT A and EQUIPMENT B) will be configured. In EQUIPMENT A, there is an interface with IP address 192.168.10.1/24 connected to a CE (Site A). In EQUIPMENT B, there are two interfaces with IP addresses 192.168.20.1/24 and 192.168.30.1/24, connected to another two CEs (Site B and Site C). The equipment A and B have loopback addresses 1.1.1.1/32 and 2.2.2.2/32 respectively. The directly connected networks will be redistributed between the PEs. The PEs are connected by gigabit-ethernet-1/1/5 interface using OSPF and LDP protocols to provide the infrastructure for L3VPN through of AS1000.



L3VPN Site-to-Site Scenario

```
!Equipment A
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
!
vlan 1000
interface gigabit-ethernet-1/1/5
untagged
```

```

!
!
switchport
interface gigabit-ethernet-1/1/5
  native-vlan
  vlan-id 1000
!
!
vrf cli1
rd 1000:10
route-target import 1000:10
!
route-target export 1000:10
!
!
interface l3 OSPF
lower-layer-if vlan 1000
ipv4 address 10.10.10.1/30
!
interface l3 VRF-CLI1-VLAN10
vrf cli1
lower-layer-if vlan 10
ipv4 address 192.168.10.1/24
!
interface loopback 0
ipv4 address 1.1.1.1/32
!
router ospf 1
router-id 1.1.1.1
area 0
interface l3-OSPF
network-type point-to-point
!
interface loopback-0
!
!
router bgp 1000
router-id 1.1.1.1
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
neighbor 2.2.2.2
update-source-address 1.1.1.1
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
!
vrf cli1
address-family ipv4 unicast
redistribute connected
!
exit-address-family
!
!
mpls ldp
lsr-id loopback-0
interface l3-OSPF
!
neighbor targeted 2.2.2.2
!
!
commit

```

```

!Equipment B
config
dot1q
vlan 20
interface gigabit-ethernet-1/1/6
!
vlan 30
interface gigabit-ethernet-1/1/2
!
vlan 1000
interface gigabit-ethernet-1/1/5
  untagged
!
!
switchport
interface gigabit-ethernet-1/1/5
  native-vlan
  vlan-id 1000
!

```

```

!
vrf cli1
rd 1000:10
route-target import 1000:10
!
route-target export 1000:10
!
!
interface l3 OSPF
lower-layer-if vlan 1000
ipv4 address 10.10.10.2/30
!
interface l3 VRF-CLI1-VLAN20
vrf cli1
lower-layer-if vlan 20
ipv4 address 192.168.20.1/24
!
interface l3 VRF-CLI1-VLAN30
vrf cli1
lower-layer-if vlan 30
ipv4 address 192.168.30.1/24
!
interface loopback 0
ipv4 address 2.2.2.2/32
!
router ospf 1
router-id 2.2.2.2
area 0
interface l3-OSPF
network-type point-to-point
!
interface loopback-0
!
!
router bgp 1000
router-id 2.2.2.2
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
neighbor 1.1.1.1
update-source-address 2.2.2.2
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
!
vrf cli1
address-family ipv4 unicast
redistribute connected
!
exit-address-family
!
!
mpls ldp
lsr-id loopback-0
interface l3-OSPF
!
neighbor targeted 1.1.1.1
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying L3VPNs](#).

11.6.2 Configuring a L3VPN Hub and Spoke

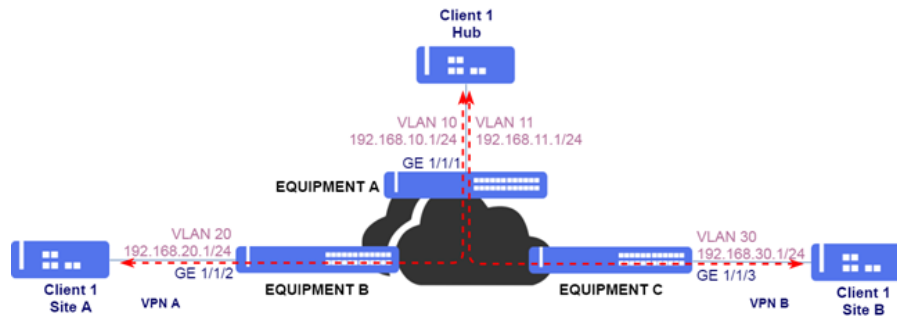
In a hub-and-spoke topology, different CEs (Hub, Site A and Site B) can access a central site but cannot communicate between themselves directly.

In the topology below, sites A e B should have connectivity to the central site, but they should not be able to communicate

between them. Sites A e B traffic will be Always forwarded to the hub. For that to happen, it is necessary to have two VPNs, one between the hub and site A and another between site B and the hub.

As all traffic will be forwarded to the hub, it will be able to control the routing between sites.

In the following topology, the PEs (EQUIPMENT A and EQUIPMENT B) are connected by gigabit-ethernet-1/1/5 interface and PEs (EQUIPMENT A and EQUIPMENT C) are connected by gigabit-ethernet-1/1/6 interface using OSPF and LDP protocols to provide the infrastructure for L3VPN through of AS1000.



L3VPN in a Hub-and-spoke Scenario

```
!Equipment A
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
!
vlan 11
interface gigabit-ethernet-1/1/1
!
vlan 1000
interface gigabit-ethernet-1/1/5
untagged
!
!
vlan 2000
interface gigabit-ethernet-1/1/6
untagged
!
!
switchport
interface gigabit-ethernet-1/1/5
native-vlan
vlan-id 1000
!
interface gigabit-ethernet-1/1/6
native-vlan
vlan-id 2000
!
!
!
!
vrf cli1-A
rd 1000:20
route-target import 1000:20
!
route-target export 1000:20
!
!
vrf cli1-B
rd 1000:30
route-target import 1000:30
!
route-target export 1000:30
!
!
interface l3 OSPF A-B
lower-layer-if vLan 1000
ipv4 address 10.10.10.1/30
!
interface l3 OSPF_A-C
```



```

lower-layer-if vlan 2000
ipv4 address 20.20.20.1/30
!
interface l3 VRF-CLI1-A-VLAN10
vrf cli1-A
lower-layer-if vlan 10
ipv4 address 192.168.10.1/24
!
interface l3 VRF-CLI1-B-VLAN11
vrf cli1-B
lower-layer-if vlan 11
ipv4 address 192.168.11.1/24
!
interface loopback 0
ipv4 address 1.1.1.1/32
!
router static
vrf cli1-A
address-family ipv4
0.0.0.0/0 next-hop 192.168.10.2
!
!
vrf cli1-B
address-family ipv4
0.0.0.0/0 next-hop 192.168.11.2
!
!
!
router ospf 1
router-id 1.1.1.1
area 0
interface l3-OSPF_A-B
network-type point-to-point
!
interface l3-OSPF_A-C
network-type point-to-point
!
interface loopback-0
!
!
!
router bgp 1000
router-id 1.1.1.1
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
neighbor 2.2.2.2
update-source-address 1.1.1.1
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
!
neighbor 3.3.3.3
update-source-address 1.1.1.1
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
!
vrf cli1-A
address-family ipv4 unicast
redistribute connected
redistribute static
!
exit-address-family
!
vrf cli1-B
address-family ipv4 unicast
redistribute connected
redistribute static
!
exit-address-family
!
!
mpls ldp
lsr-id loopback-0
interface l3-OSPF_A-B
!
interface l3-OSPF_A-C
!
neighbor targeted 2.2.2.2
!
neighbor targeted 3.3.3.3

```

```
!
commit
```

```
!Equipment B
config
dot1q
vlan 20
    interface gigabit-ethernet-1/1/2
    !
vlan 1000
    interface gigabit-ethernet-1/1/5
        untagged
    !
!
switchport
interface gigabit-ethernet-1/1/5
    native-vlan
    vlan-id 1000
    !
!
vrf cli1
rd 1000:20
route-target import 1000:20
!
route-target export 1000:20
!
!
interface l3 OSPF_A-B
lower-layer-if vlan 1000
ipv4 address 10.10.10.2/30
!
interface l3 VRF-CLI1-VLAN20
vrf cli1
lower-layer-if vlan 20
ipv4 address 192.168.20.1/24
!
interface loopback 0
ipv4 address 2.2.2.2/32
!
router ospf 1
router-id 2.2.2.2
area 0
    interface l3-OSPF_A-B
        network-type point-to-point
    !
    interface loopback-0
    !
!
!
router bgp 1000
router-id 2.2.2.2
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
neighbor 1.1.1.1
update-source-address 2.2.2.2
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
!
vrf cli1
address-family ipv4 unicast
redistribute connected
!
exit-address-family
!
!
mpls ldp
lsr-id loopback-0
    interface l3-OSPF_A-B
    !
    neighbor targeted 1.1.1.1
    !
commit
```

```
!Equipment C
config
dot1q
vlan 30
    interface gigabit-ethernet-1/1/3
    !
```

```

vlan 2000
  interface gigabit-ethernet-1/1/6
    untagged
  !
!
switchport
interface gigabit-ethernet-1/1/6
  native-vlan
  vlan-id 2000
!
!
vrf cli1
rd 1000:30
route-target import 1000:30
!
route-target export 1000:30
!
!
interface l3 OSPF A-C
lower-layer-if vlan 2000
ipv4 address 20.20.20.2/30
!
interface l3 VRF-CLI1-VLAN30
vrf cli1
lower-layer-if vlan 30
ipv4 address 192.168.30.1/24
!
interface loopback 0
ipv4 address 3.3.3.3/32
!
router ospf 1
router-id 3.3.3.3
area 0
interface l3-OSPF A-C
network-type point-to-point
!
interface loopback-0
!
!
!
router bgp 1000
router-id 3.3.3.3
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
neighbor 1.1.1.1
update-source-address 3.3.3.3
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
!
vrf cli1
address-family ipv4 unicast
redistribute connected
!
exit-address-family
!
!
!
mpls ldp
lsr-id loopback-0
interface l3-OSPF A-C
!
neighbor targeted 1.1.1.1
!
commit

```



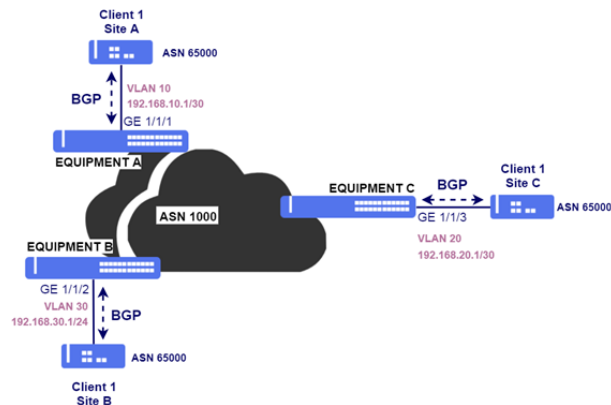
The available commands for troubleshooting can be found in the topic [Verifying L3VPNs](#).

11.6.3 Configuring BGP between PEs and CEs

In order to avoid the configuration of static routes in PEs (EQUIPMENT A, B and C), it is recommended to use a routing protocol between PEs and CEs (Hub, Site A and Site B) for route distribution. In the following topology, BGP will be used.

CEs are in AS 65000 and the PEs are in AS 1000. Because we are using iBGP in the CEs, it is necessary to configure the *as-override* parameter in BGP neighbor to avoid that the BGP loop detection mechanism discard prefixes exchanged for CEs.

OSPF protocol can also be used between CE and PE for route redistribution, as shown in [Configuring OSPF between PEs and CEs](#).



eBGP Session between PE and CE

```
!Equipment A
config
router bgp 1000
router-id 1.1.1.1
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
vrf cli1
address-family ipv4 unicast
redistribute connected
!
exit-address-family
!
neighbor 192.168.10.2
update-source-address 192.168.10.1
remote-as 65000
next-hop-self
address-family ipv4 unicast
as-override
exit-address-family
commit
```

```
!Equipment B
config
router bgp 1000
router-id 2.2.2.2
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
vrf cli1
address-family ipv4 unicast
redistribute connected
!
exit-address-family
!
neighbor 192.168.20.2
update-source-address 192.168.20.1
remote-as 65000
next-hop-self
address-family ipv4 unicast
as-override
exit-address-family
commit
```

```
!Equipment C
config
router bgp 1000
router-id 3.3.3.3
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
vrf cli1
address-family ipv4 unicast
redistribute connected
!
exit-address-family
!
neighbor 192.168.30.2
update-source-address 192.168.30.1
remote-as 65000
next-hop-self
address-family ipv4 unicast
as-override
exit-address-family
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L3VPNs](#).

11.6.4 Enabling AS Override

In some scenarios, it could be necessary to change the AS PATH to avoid that the BGP loop detection mechanism discard received prefixes. AS Override can be used for that.

```
router bgp 1000
vrf cli1
neighbor 192.168.30.2
address-family ipv4 unicast
as-override
exit-address-family
!
commit
```

11.6.5 Enabling Allow AS In

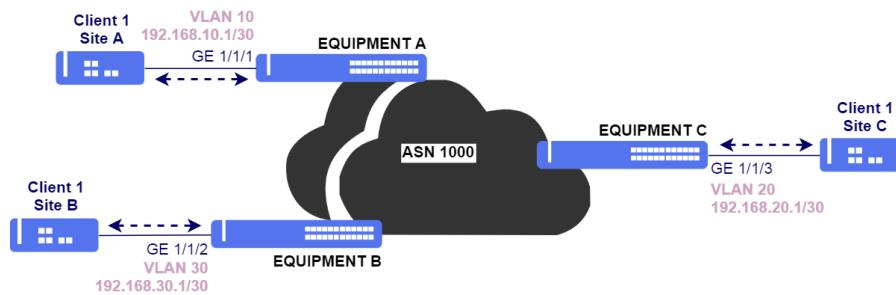
Allow AS In can also be used to allow loops in AS PATH in the PE.

```
router bgp 1000
vrf cli1
neighbor 192.168.30.2
address-family ipv4 unicast
allow-as-in 1
exit-address-family
!
commit
```

11.6.6 Configuring OSPF between PEs and CEs

In order to avoid the configuration of static routes in PEs (EQUIPMENT A, B and C), it is recommended to use a routing protocol between PEs and CEs (Hub, Site A and Site B) for route distribution. In the following topology, OSPF will be used.

BGP protocol can also be used between PEs and CEs for route redistribution, as shown in [Configuring BGP between PEs and CEs](#).



OSPF between PEs and CEs in L3VPN

To advertise routes received via MP-BGP from other PEs to CEs, **redistribute bgp** command must be configured in the OSPF configuration.

To advertise routes received from CEs via OSPF to PEs via MP-BGP, **redistribute ospf** command must be configured in BGP configuration.

```
!Equipment A
config
router bgp 1000
vrf cli1
  address-family ipv4 unicast
    redistribute ospf
    !
  exit-address-family
!
!
router ospf 10 vrf cli1
  redistribute bgp
  !
  area 0
    interface l3-VRF-CLI1-VLAN10
      network-type point-to-point
    !
  !
!
commit
```

```
!Equipment B
config
router bgp 1000
vrf cli1
  address-family ipv4 unicast
    redistribute ospf
    !
  exit-address-family
!
!
router ospf 10 vrf cli1
  redistribute bgp
  !
  area 0
    interface l3-VRF-CLI1-VLAN30
      network-type point-to-point
    !
  !
!
commit
```

```
!Equipment C
config
router bgp 1000
vrf cli1
  address-family ipv4 unicast
  redistribute ospf
  !
  exit-address-family
!
!
router ospf 10 vrf cli1
redistribute bgp
!
area 0
interface l3-VRF-CLI1-VLAN20
network-type point-to-point
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying L3VPNs](#).

11.6.7 Verifying L3VPNs

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show ip ospf neighbor brief
show mpls ldp neighbor
show mpls l3vpn vpnv4 vrf <vrf-name> brief
show ip bgp vpnv4 labels
show ip route vrf <vrn-name>
show ip fib vrf <vrf-name> brief
show ip host-table vrf <vrf-name> brief
show ip interface vrf <vrf-name> brief
```

11.7 6VPE Configuration

6VPE (IPv6 VPN Provider Edge Router) use the the IPv4 backbone to provide VPN IPv6 services, eliminating the need for dual-stacking within the MPLS core. It is similar to an IPv4 L3VPN, but with IPv6 support. It provides logically separate routing table entries for VPN member devices using VRFs.

11.7.1 Configuring a 6VPE L3VPN Site-to-Site

While L2VPN provides transparent layer 2 services, in L3VPN the routing is performed by ISP equipment. Packets are forwarded using MPLS labels. BGP is the protocol used to exchange route and label information between PEs.

Each route is identified by a route-distinguisher (RD), which must be unique for each client, allowing IP addresses overlapping between different customers. Routes are marked with BGP communities called route-targets, which are used to define in which VPN each route will be installed.



It is mandatory to have LDP protocol configured and running in the network to be able to use L3VPN service.

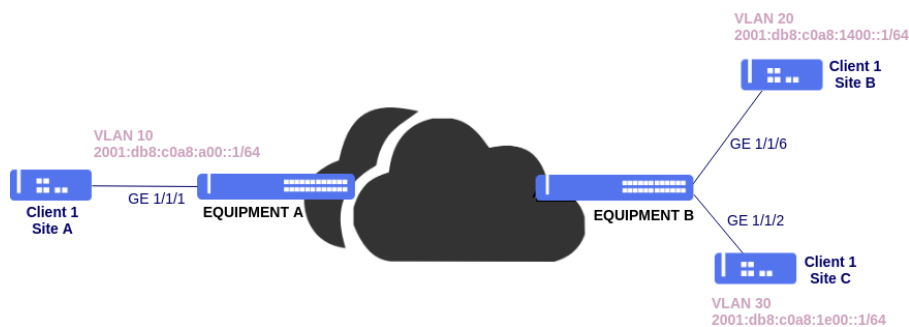


Despite the route-distinguisher format being similar to the route-target format, both are independent and have different functions.



BGP is responsible for label and route exchange between PEs. It is necessary to enable the VPNv4 family in BGP protocol.

In the following topology, two PEs (EQUIPMENT A and EQUIPMENT B) will be configured. In EQUIPMENT A, there is an interface with IP address 2001:db8:c0a8:a00::1/64 connected to a CE (Site A). In EQUIPMENT B, there are two interfaces with IP addresses 2001:db8:c0a8:1400::1/64 and 2001:db8:c0a8:1e00::1/64, connected to another two CEs (Site B and Site C). The equipment A and B have loopback addresses 1.1.1.1/32 and 2.2.2.2/32 respectively. The directly connected networks will be redistributed between the PEs. The PEs are connected by gigabit-ethernet-1/1/5 interface using OSPF and LDP protocols to provide the infrastructure for L3VPN through of AS1000.



6VPE Site-to-Site Scenario

```
!Equipment A
config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1
!
vlan 1000
interface gigabit-ethernet-1/1/5
untagged
!
!
switchport
```



```

interface gigabit-ethernet-1/1/5
  native-vlan
  vlan-id 1000
!
!
vrf cli1
  rd 1000:10
  route-target import 1000:10
  route-target export 1000:10
!
!
interface l3 OSPF
  lower-layer-if vlan 1000
  ipv4 address 10.10.10.1/30
!
interface l3 VRF-CLI1-VLAN10
  vrf cli1
  lower-layer-if vlan 10
  ipv6 enable
  ipv6 address 2001:db8:c0a8:a00::1/64
!
interface loopback 0
  ipv4 address 1.1.1.1/32
!
router ospf 1
  router-id 1.1.1.1
  area 0
    interface l3-OSPF
      network-type point-to-point
    interface loopback-0
!
!
router bgp 1000
  router-id 1.1.1.1
  address-family ipv4 unicast
  address-family vpnv4 unicast
  address-family vpnv6 unicast
  neighbor 2.2.2.2
  update-source-address 1.1.1.1
  remote-as 1000
  next-hop-self
  address-family ipv4 unicast
  address-family vpnv6 unicast
!
vrf cli1
  address-family ipv6 unicast
  redistribute connected
  exit-address-family
!
!
mpls ldp
  lsr-id loopback-0
  interface l3-OSPF
  neighbor targeted 2.2.2.2
!
!
commit

```

```

!Equipment B
config
dot1q
  vlan 20
    interface gigabit-ethernet-1/1/6
  !
  vlan 30
    interface gigabit-ethernet-1/1/2
  !
  vlan 1000
    interface gigabit-ethernet-1/1/5
    untagged
  !
!
switchport
  interface gigabit-ethernet-1/1/5
    native-vlan
    vlan-id 1000

```

```

!
!
vrf cli1
rd 1000:10
route-target import 1000:10
!
route-target export 1000:10
!
!
interface l3 OSPF
lower-layer-if vlan 1000
ipv4 address 10.10.10.2/30
!
interface l3 VRF-CLI1-VLAN20
vrf cli1
lower-layer-if vlan 20
ipv6 enable
ipv6 address 2001:db8:c0a8:1400::1/64
!
interface l3 VRF-CLI1-VLAN30
vrf cli1
lower-layer-if vlan 30
ipv6 enable
ipv6 address 2001:db8:c0a8:1e00::1/64
!
interface loopback 0
ipv4 address 2.2.2.2/32
!
router ospf 1
router-id 2.2.2.2
area 0
interface l3-OSPF
network-type point-to-point
!
interface loopback-0
!
!
!
router bgp 1000
router-id 2.2.2.2
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
address-family vpnv6 unicast
!
neighbor 1.1.1.1
update-source-address 2.2.2.2
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv6 unicast
!
!
vrf cli1
address-family ipv6 unicast
redistribute connected
!
exit-address-family
!
!
!
mpls ldp
lsr-id loopback-0
interface l3-OSPF
!
neighbor targeted 1.1.1.1
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying 6VPE](#).

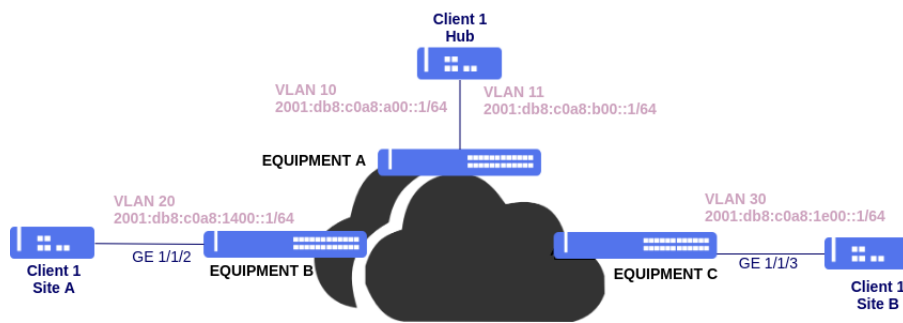
11.7.2 Configuring a 6VPE L3VPN Hub and Spoke

In a hub-and-spoke topology, different CEs (Hub, Site A and Site B) can access a central site but cannot communicate between themselves directly.

In the topology below, sites A e B should have connectivity to the central site, but they should not be able to communicate between them. Sites A e B traffic will be Always forwarded to the hub. For that to happen, it is necessary to have two VPNs, one between the hub and site A and another between site B and the hub.

As all traffic will be forwarded to the hub, it will be able to control the routing between sites.

In the following topology, the PEs (EQUIPMENT A and EQUIPMENT B) are connected by gigabit-ethernet-1/1/5 interface and PEs (EQUIPMENT A and EQUIPMENT C) are connected by gigabit-ethernet-1/1/6 interface using OSPF and LDP protocols to provide the infrastructure for L3VPN through of AS1000.



6VPE in a Hub-and-spoke Scenario

```
!Equipment A
config
dot1q
vlan 10
!
interface gigabit-ethernet-1/1/1
!
vlan 11
!
interface gigabit-ethernet-1/1/1
!
vlan 1000
!
interface gigabit-ethernet-1/1/5
!
untagged
!
!
vlan 2000
!
interface gigabit-ethernet-1/1/6
!
untagged
!
!
!
switchport
!
interface gigabit-ethernet-1/1/5
!
native-vlan
!
vlan-id 1000
!
!
interface gigabit-ethernet-1/1/6
!
native-vlan
!
vlan-id 2000
!
!
!
!
!
vrf cli1-A
!
rd 1000:20
!
route-target import 1000:20
!
route-target export 1000:20
!
!
```

```

!
vrf cli1-B
rd 1000:30
route-target import 1000:30
!
route-target export 1000:30
!
!
interface l3 OSPF_A-B
lower-layer-if vlan 1000
ipv4 address 10.10.10.1/30
!
interface l3 OSPF_A-C
lower-layer-if vlan 2000
ipv4 address 20.20.20.1/30
!
interface l3 VRF-CLI1-A-VLAN10
vrf cli1-A
lower-layer-if vlan 10
ipv6 enable
ipv6 address 2001:db8:c0a8:a00::1/64
!
interface l3 VRF-CLI1-B-VLAN11
vrf cli1-B
lower-layer-if vlan 11
ipv6 enable
ipv6 address 2001:db8:c0a8:b00::1/64
!
interface loopback 0
ipv4 address 1.1.1.1/32
!
router static
vrf cli1-A
address-family ipv6
::/0 next-hop 2001:db8:c0a8:a00::2
!
!
vrf cli1-B
address-family ipv6
::/0 next-hop 2001:db8:c0a8:b00::2
!
!
!
router ospf 1
router-id 1.1.1.1
area 0
interface l3-OSPF_A-B
network-type point-to-point
!
interface l3-OSPF_A-C
network-type point-to-point
!
interface loopback-0
!
!
router bgp 1000
router-id 1.1.1.1
address-family ipv4 unicast
!
address-family vpnv4 unicast
!
address-family vpnv6 unicast
!
neighbor 2.2.2.2
update-source-address 1.1.1.1
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv6 unicast
!
!
neighbor 3.3.3.3
update-source-address 1.1.1.1
remote-as 1000
next-hop-self
address-family ipv4 unicast
!
address-family vpnv6 unicast
!
!
vrf cli1-A
address-family ipv6 unicast
redistribute connected
redistribute static
!
exit-address-family
!
vrf cli1-B

```

```

    address-family ipv6 unicast
    redistribute connected
    redistribute static
    !
  exit-address-family
!
!
mpls ldp
  lsr-id loopback-0
  interface l3-OSPF_A-B
  !
  interface l3-OSPF_A-C
  !
  neighbor targeted 2.2.2.2
  !
  neighbor targeted 3.3.3.3
!
!
commit

```

```

!Equipment B
config
dot1q
  vlan 20
  interface gigabit-ethernet-1/1/2
  !
  vlan 1000
  interface gigabit-ethernet-1/1/5
  untagged
  !
!
switchport
  interface gigabit-ethernet-1/1/5
  native-vlan
  vlan-id 1000
  !
!
vrf cli1
  rd 1000:20
  route-target import 1000:20
  !
  route-target export 1000:20
  !
!
interface l3 OSPF A-B
  lower-layer-if v1an 1000
  ipv4 address 10.10.10.2/30
  !
interface l3 VRF-CLI1-VLAN20
  vrf cli1
  lower-layer-if v1an 20
  ipv6 enable
  ipv6 address 2001:db8:c0a8:1400::1/64
  !
interface loopback 0
  ipv4 address 2.2.2.2/32
  !
router ospf 1
  router-id 2.2.2.2
  area 0
  interface l3-OSPF A-B
  network-type point-to-point
  !
  interface loopback-0
  !
!
router bgp 1000
  router-id 2.2.2.2
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
  neighbor 1.1.1.1
  update-source-address 2.2.2.2
  remote-as 1000
  next-hop-self
  address-family ipv4 unicast
  !
  address-family vpnv6 unicast
  !
!
vrf cli1
  address-family ipv6 unicast
  redistribute connected

```

```

    !
    exit-address-family
  !
  !
mpls ldp
  lsr-id loopback-0
  interface l3-OSPF_A-B
  !
  neighbor targeted 1.1.1.1
  !
  !
commit

```

```

!Equipment C
config
dot1q
  vlan 30
  interface gigabit-ethernet-1/1/3
  !
  vlan 2000
  interface gigabit-ethernet-1/1/6
  untagged
  !
  !
switchport
  interface gigabit-ethernet-1/1/6
  native-vlan
  vlan-id 2000
  !
  !
  !
vrf cli1
  rd 1000:30
  route-target import 1000:30
  !
  route-target export 1000:30
  !
  !
interface l3 OSPF_A-C
  lower-layer-if vlan 2000
  ipv4 address 20.20.20.2/30
  !
interface l3 VRF-CLI1-VLAN30
  vrf cli1
  lower-layer-if vlan 30
  ipv6 enable
  ipv6 address 2001:db8:c0a8:1e00::1/64
  !
interface loopback 0
  ipv4 address 3.3.3.3/32
  !
router ospf 1
  router-id 3.3.3.3
  area 0
  interface l3-OSPF_A-C
  network-type point-to-point
  !
  interface loopback-0
  !
  !
  !
router bgp 1000
  router-id 3.3.3.3
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family vpnv6 unicast
  !
  neighbor 1.1.1.1
  update-source-address 3.3.3.3
  remote-as 1000
  next-hop-self
  address-family ipv4 unicast
  !
  address-family vpnv6 unicast
  !
  !
  !
vrf cli1
  address-family ipv6 unicast
  redistribute connected
  !
  exit-address-family
  !
  !
mpls ldp
  lsr-id loopback-0

```

```

interface l3-OSPF_A-C
!
neighbor targeted 1.1.1.1
!
commit

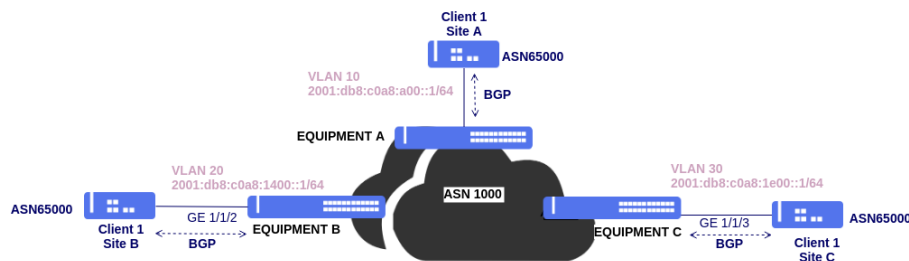
```



The available commands for troubleshooting can be found in the topic [Verifying 6VPE](#).

11.7.3 Configuring BGP between PEs and CEs

In order to avoid the configuration of static routes in PEs (EQUIPMENT A, B and C), it is recommended to use a routing protocol between PEs and CEs (Hub, Site A and Site B) for route distribution. In the following topology, BGP will be used. CEs are in AS 65000 and the PEs are in AS 1000. Because we are using iBGP in the CEs, it is necessary to configure the *as-override* parameter in BGP neighbor to avoid that the BGP loop detection mechanism discard prefixes exchanged for CEs.



eBGP Session between PE and CE

```

!Equipment A
config
router bgp 1000
router-id 1.1.1.1
address-family ipv4 unicast
!
address-family ipv6 unicast
!
address-family vpnv4 unicast
!
address-family vpnv6 unicast
!
vrf cli1
address-family ipv6 unicast
redistribute connected
!
exit-address-family
!
neighbor 2001:db8:c0a8:a00::2
update-source-address 2001:db8:c0a8:a00::1
remote-as 65000
next-hop-self
address-family ipv6 unicast
as-override
exit-address-family
commit

```

```
!Equipment B
config
router bgp 1000
router-id 2.2.2.2
address-family ipv4 unicast
!
address-family ipv6 unicast
!
address-family vpnv4 unicast
!
address-family vpnv6 unicast
!
vrf cli1
address-family ipv6 unicast
redistribute connected
!
exit-address-family
!
neighbor 2001:db8:c0a8:1400::2
update-source-address 2001:db8:c0a8:1400::2
remote-as 65000
next-hop-self
address-family ipv6 unicast
as-override
exit-address-family
commit
```

```
!Equipment C
config
router bgp 1000
router-id 3.3.3.3
address-family ipv4 unicast
!
address-family ipv6 unicast
!
address-family vpnv4 unicast
!
address-family vpnv6 unicast
!
vrf cli1
address-family ipv6 unicast
redistribute connected
!
exit-address-family
!
neighbor 2001:db8:c0a8:1e00::2
update-source-address 2001:db8:c0a8:1e00::1
remote-as 65000
next-hop-self
address-family ipv6 unicast
as-override
exit-address-family
commit
```



The available commands for troubleshooting can be found in the topic [Verifying 6VPE](#).

11.7.4 Enabling AS Override

In some scenarios, it could be necessary to change the AS PATH to avoid that the BGP loop detection mechanism discard received prefixes. AS Override can be used for that.

```
router bgp 1000
vrf cli1
neighbor 2001:db8:c0a8:1400::2
address-family ipv6 unicast
as-override
exit-address-family
!
commit
```


11.7.5 Enabling Allow AS In

Allow AS In can also be used to allow loops in AS PATH in the PE.

```
router bgp 1000
vrf cli1
neighbor 2001:db8:c0a8:1400::2
address-family ipv6 unicast
allow-as-in 1
exit-address-family
!
commit
```

11.7.6 Verifying 6VPE

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show ip ospf neighbor brief
show mpls ldp neighbor
show mpls l3vpn vpnv6 vrf <vrf-name> brief
show ip bgp vpnv6 labels
show ipv6 route vrf <vrn-name>
show ipv6 fib vrf <vrf-name> brief
show ipv6 host-table vrf <vrf-name> brief
show ipv6 interface vrf <vrf-name> brief
```

11.8 BGP Free Core Configuration

BGP Free Core as the name suggests is a approach where service providers Core does not have BGP configured. This approach requires some tunneling technology, the most common being use MPLS. This approach has positive and negative points, one of the benefits is to eliminate the need for Core equipment to have a large number of prefixes.



DmOS only acts on the P function, due only supports data encapsulation from a L2VPN or L3VPN.

The following topology will be used as a demonstration to configure the feature.



BGP Free Core scenario

PE equipment must have the following features configured:

- OSPF
- LDP
- iBGP between PEs loopbacks with next-hop-self
- It is also necessary that the PE equipment encapsulate the data in MPLS, depending on the vendor this is done by default and in other cases additional configuration is required.

From the point of view of P equipment, data is received and forwarded by MPLS label, for this reason it is only necessary to have OSPF and LDP configured in these devices.

```

!Equipment A
config
dot1q
vlan 10
interface ten-gigabit-ethernet-1/1/1
untagged
!!
!
vlan 20
interface ten-gigabit-ethernet-1/1/2
untagged
!!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 10
!
!
interface ten-gigabit-ethernet-1/1/2
native-vlan
vlan-id 20
!!
!
interface l3 VLAN10
ipv4 address 203.0.113.1/30
lower-layer-if vlan 10
!
!
interface l3 VLAN20
ipv4 address 172.30.0.1/30
lower-layer-if vlan 20
!
!
interface loopback 0
ipv4 address 198.51.100.1/32
!
!
router ospf 1
router-id 198.51.100.1
area 0
interface l3-VLAN10
network-type point-to-point
!
interface l3-VLAN20
network-type point-to-point
!
interface loopback-0
!
!
mpls ldp
lsr-id loopback-0
interface l3-VLAN10
!
interface l3-VLAN20
!
!
commit

```

```

!Equipment B
config
dot1q
vlan 10
interface ten-gigabit-ethernet-1/1/1
untagged
!
!
vlan 20
interface ten-gigabit-ethernet-1/1/2
untagged
!
!
!
switchport
interface ten-gigabit-ethernet-1/1/1
native-vlan
vlan-id 10
!
!
interface ten-gigabit-ethernet-1/1/2
native-vlan
vlan-id 20
!
!
!
interface l3 VLAN10
ipv4 address 203.0.113.2/30
lower-layer-if vlan 10
!
!
interface l3 VLAN20
ipv4 address 172.31.0.1/30
lower-layer-if vlan 20
!
!
interface loopback 0
ipv4 address 198.51.100.2/32
!
!
router ospf 1
router-id 198.51.100.2
area 0
interface l3-VLAN10
network-type point-to-point
!
interface l3-VLAN20
network-type point-to-point
!
interface loopback-0
!
!
!
mpls ldp
lsr-id loopback-0
interface l3-VLAN10
!
interface l3-VLAN20
!
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying LDP](#).

12 Multicast

This chapter describes the multicast protocols configuration. It contains the following sections:

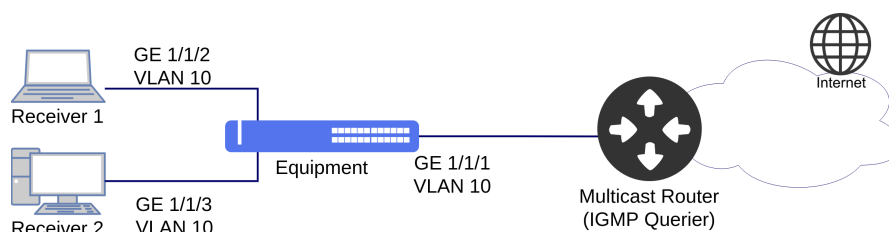
- IGMP Snooping Configuration

12.1 IGMP Snooping Configuration

The IGMP Snooping (Internet Group Management Protocol) protocol analyses the packets of IGMP protocol within a VLAN in order to discover which interfaces have interest in receiving the multicast stream. Using the information obtained by the protocol, the IGMP Snooping reduces consumption of bandwidth in a LAN, avoiding flooding multicast traffic to devices that do not wish to receive multicast flows.

12.1.1 Configuring IGMP Snooping in Ethernet Application

The scenario below will be used to describe a multicast application with IGMP Snooping.



Implementation of IGMP Snooping



IGMP Querier is not supported in DmOS.



In DmOS, the IGMP Snooping version must be the same IGMP version configured in IGMP Querier.

The next steps will indicate how to configure the IGMP Snooping version 2 in **VLAN 10** for inspection of multicast stream in the gigabit 1/1/1 uplink interface and in gigabit 1/1/2 and gigabit 1/1/3 access interfaces where receivers 1 and 2 are connected.



If the IGMP version is not specified in an interface, by default IGMP Snooping version 3 is configured.

```

config
dot1q
vlan 10
interface gigabit-ethernet-1/1/1 tagged
interface gigabit-ethernet-1/1/2 untagged
interface gigabit-ethernet-1/1/3 untagged
!
!
switchport
interface gigabit-ethernet-1/1/2
native-vlan
vlan-id 10
!
interface gigabit-ethernet-1/1/3
native-vlan
vlan-id 10
!
!
multicast igmp snooping 1
bridge-domain id 10
interface gigabit-ethernet-1/1/1 version 2
interface gigabit-ethernet-1/1/2 version 2
interface gigabit-ethernet-1/1/3 version 2
commit

```



The available commands for troubleshooting can be found in the topic [Verifying IGMP](#).

12.1.2 Configuring IGMP Snooping in GPON Application

The scenario below will be used to describe a multicast application with IGMP Snooping in GPON scenario.



Implementation of IGMP Snooping in GPON scenario



Setting of a GPON service prior to application of the following configs is required. And, it is possible to execute the config using an Ethernet interface as access interface instead of a service-port of GPON interface.

The next steps will indicate how to configure the IGMP Snooping in **VLAN 3000** for inspection of multicast traffic in the gigabit 1/1/1 interface and in ONU 1 that is set in service-port 1.

```
config
dot1q
vlan 3000
interface gigabit-ethernet-1/1/1 tagged
interface service-port-1
!
!
multicast igmp snooping 1
bridge-domain id 3000
interface gigabit-ethernet-1/1/1
interface service-port 1
commit
```



The available commands for troubleshooting can be found in the topic [Verifying IGMP](#).

12.1.3 Verifying IGMP

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show multicast igmp snooping groups
show multicast igmp snooping groups brief
show multicast igmp snooping groups detail
show multicast igmp snooping groups extensive
show multicast igmp snooping mrouter
show multicast igmp snooping statistics
```

13 QoS

The QoS (Quality of Service) is a set of mechanisms and algorithms used to classify and organize the traffic in the network. The main objective is to assure that the services that need transmission quality in the network (latency, jitter and bandwidth), for example: voip or multicast operate in an adequate manner.

This chapter contains the following sections:

- Congestion Control Configuration
- Traffic Shapping Configuration
- Traffic Policing Configuration



In MPLS scenarios, the customers flows with VLAN has automatically the PCP copied to EXP field of labels, acting according to the configuration of scheduler queues, rate-limit or policer queues.

13.1 Congestion Control Configuration

13.1.1 Configuring WFQ Scheduler

The WFQ (Weighted Fair Queuing) is a scheduler that defines weights for the queues providing a bandwidth for each queue in congestion conditions. The queue when set as SP shall consume the entire bandwidth available and only the excess will be divided between the other queues with calculation based on the weight of each one.

The next steps will define how to set the WFQ in the gigabit 1/1/1 interface with the following specifications:

- **Queue 0:** weight 5
- **Queue 1 and 2:** weight 10
- **Queue 3 and 4:** weight 15
- **Queue 5:** weight 20
- **Queue 6:** weight 25
- **Queue 7:** SP (Strict Priority)

```
config
qos scheduler-profile WFQ-Profile-1
mode wfq
queue 0 weight 5
queue 1 weight 10
queue 2 weight 10
queue 3 weight 15
queue 4 weight 15
queue 5 weight 20
queue 6 weight 25
queue 7 weight SP
!
qos interface gigabit-ethernet-1/1/1 scheduler-profile WFQ-Profile-1
commit
```



There are no troubleshooting commands for this functionality.

13.2 Traffic Shapping Configuration

Traffic Shapping adjusts the traffic rate through the use of a buffer, which will hold packets when they are over the allowed bandwidth, introducing some delay.

13.2.1 Configuring Rate Limit on Interface

The Rate limit is a functionality that limits the maximum rate of traffic and the burst that an interface may output or input.



The rate inserted should be in kbps unit and the burst in KB.

The next steps will indicate how to set the Rate limit in the input with the value of 30 Mbps (30000 kbps) with burst of 2 MB (2000 kB) and in the output with the value of 100 Mbps (100000 kbps) with burst of 2 MB (2000 kB) in the gigabit 1/1/1 interface.

```
config
qos interface gigabit-ethernet-1/1/1
rate-limit
  ingress
    bandwidth 30000
    burst 2000
  egress
    bandwidth 100000
    burst 2000
commit
```



There are no troubleshooting commands for this functionality.

13.3 Traffic Policing Configuration

Policer is one of the functionalities that allow the control of traffic used over an available but finite bandwidth. It is a mechanism of classification and control of flows according to the levels of services desired. The Policer classifies the flows in colors (green, yellow and red) depending on policer mode configured with respective rates and burts allowing to take different actions according to the classification made.



The burst parameters use **bytes** unit and the rate parameters in **kbits/s** unit.



Policer supports flow, srTCM (RFC 2697), trTCM (RFC 2698) e Differentiated Service trTCM (RFC 4115) modes. Depending on platform used some modes are not supported.



Traffic Policer can be performed based on interface, VLAN, inner-VLAN, PCP, inner-PCP and DSCP.

13.3.1 Configuring Traffic Policing based on VLANs

The next steps will demonstrate how to configure Traffic Policer by limiting the client bandwidth using **VLAN 10** for **downloading 15 Mbps** (15000 kbits/s) and **uploading 5 Mbps** (5000 kbits/s) using **burst 1 MB** (1000000 bytes) by performing the discarding of excess traffic.

```
config
qos policer
profile download
mode flow
parameters
  cir 15000
  cbs 1000000
!
stage egress
actions
  red drop
!
profile upload
mode flow
parameters
  cir 5000
  cbs 1000000
!
stage ingress
actions
  red drop
!
instance 1
interface ten-gigabit-ethernet-1/1/3
profile download
vlan 10
!
instance 2
interface ten-gigabit-ethernet-1/1/3
profile upload
vlan 10
!
commit
```

In case of client use the **VLANs 10 and 20** it is possible to configure a list of VLANs using the command below:

```
config
qos policer
instance 1
  vlan 10,20
!
instance 2
```

```
vlan 10,20
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying QoS policers](#).

13.3.2 Configuring Traffic Policing based on inner VLAN

The next steps will demonstrate how to configure Traffic Policer by limiting the bandwidth of clients or services associated to different *inner-vlans* in a service VLAN.

The Policer limits traffic using **inner VLANs 300 to 500** and **service vlan 10** for **downloading 150 Mbps** (150000 kbits/s) and **uploading 50 Mbps** (50000 kbits/s) using **burst 1 MB** (1000000 bytes) by performing the discarding of excess traffic.

Access interface is ten-gigabit-ethernet-1/1/3 and uplink interface is ten-gigabit-ethernet-1/1/5.

```
config
qos policer
profile download
mode flow
parameters
  cir 150000
  cbs 1000000
!
stage ingress
actions
  red drop
!
profile upload
mode flow
parameters
  cir 50000
  cbs 1000000
!
stage ingress
actions
  red drop
!
instance 1
interface ten-gigabit-ethernet-1/1/5
name uplink-interface
profile download
inner-vlan 300-500
vlan 10
!
instance 2
interface ten-gigabit-ethernet-1/1/3
name access-interface
profile upload
inner-vlan 300-500
vlan 10
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying QoS policers](#).

13.3.3 Configuring Traffic Policing based on PCP

The next steps will demonstrate how to configure Traffic Policer by limiting the client bandwidth using **PCP 5** for **downloading 15 Mbps** (15000 kbits/s) and **uploading 5 Mbps** (5000 kbits/s) using **burst 1 MB** (1000000 bytes) by performing the discarding of excess traffic.

```
config
qos policer
profile download
mode flow
parameters
  cir 15000
  cbs 1000000
!
stage egress
actions
  red drop
!
!
profile upload
mode flow
parameters
  cir 5000
  cbs 1000000
!
stage ingress
actions
  red drop
!
!
instance 1
interface ten-gigabit-ethernet-1/1/3
profile download
pcp 5
!
instance 2
interface ten-gigabit-ethernet-1/1/3
profile upload
pcp 5
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying QoS policers](#).

13.3.4 Configuring Traffic Policing based on DSCP

The next steps will demonstrate how to configure Traffic Policer by limiting the client bandwidth using **DSCP cs1** for **downloading 15 Mbps** (15000 kbits/s) and **uploading 5 Mbps** (5000 kbits/s) using **burst 1 MB** (1000000 bytes) by performing the discarding of excess traffic.

Access interface is ten-gigabit-ethernet-1/1/3 and uplink interface is ten-gigabit-ethernet-1/1/5.

```
config
qos policer
profile download
mode flow
parameters
  cir 15000
  cbs 1000000
!
stage ingress
actions
  red drop
!
!
profile upload
mode flow
```

```

parameters
  cir 5000
  cbs 1000000
!
stage ingress
actions
  red drop
!
instance 1
interface ten-gigabit-ethernet-1/1/5
profile download
dscp cs1
!
instance 2
interface ten-gigabit-ethernet-1/1/3
profile upload
dscp cs1
!
commit

```



The available commands for troubleshooting can be found in the topic [Verifying QoS policers](#).

13.3.5 Configuring Hierarchical Traffic Policing based on PCP

The Hierarchical Quality of Service (HQoS) is a feature used to limit a specific traffic profile within a bandwidth limit.



The *flow* mode is mandatory in hierarchial instance and it is recommended to use *trtcmds* mode in profiles included in the children instances.



Hierarchical Quality of Service (HQoS) support available for ingress (RX).

The next steps will demonstrate how to configure the Hierarchical Traffic Policer by limiting the customer's total upload bandwidth by ensuring a specific band according to the service. Upload traffic from the **ten-gigabit-ethernet-1/1/1** interface will be limited to **10 Mbps** (10000 kbits/s) using a **1 MB** burst (1000000 bytes) discarding excess traffic.

Traffic limits will be applied according to each service.

- **PCP 5:** Traffic limit of **3 Mbps**, excess traffic will be marked as best effort (PCP 0)
- **PCP 7:** Traffic limit of **3 Mbps**, excess traffic will be marked as best effort (PCP 0)
- The rest of the traffic will pass up to the 10Mbps limit.

```

config
dot1q
vlan 1005
interface ten-gigabit-ethernet-1/1/1
!
interface hundred-gigabit-ethernet-1/1/1
!
qos policer

```

```

profile upload-client
mode flow
parameters
  cir 10000
  cbs 1000000
!
stage ingress
!
profile upload-client-best-effort
mode trtcms
parameters
  cir 0
  cbs 2048
  eir 10000
  ebs 2048
!
stage ingress
actions
  red drop
!
!
profile upload-client-pcp-5-voip
mode trtcms
parameters
  cir 3000
  cbs 2048
  eir 50000
  ebs 2048
!
stage ingress
actions
  yellow set-pcp 0
  red drop
!
!
profile upload-client-pcp-7-network-control
mode trtcms
parameters
  cir 3000
  cbs 2048
  eir 10000
  ebs 2048
!
stage ingress
actions
  yellow set-pcp 0
  red drop
!
!
instance 1
interface ten-gigabit-ethernet-1/1/1
profile upload-client-best-effort
vlan 1005
!
instance 2
interface ten-gigabit-ethernet-1/1/1
profile upload-client-pcp-5-voip
vlan 1005
pcp 5
!
instance 3
interface ten-gigabit-ethernet-1/1/1
profile upload-client-pcp-7-network-control
vlan 1005
pcp 7
!
hierarchical 1
profile upload-client
instance 1-3
!
!
commit
!

```



The available commands for troubleshooting can be found in the topic [Verifying QoS policers](#).

13.3.6 Verifying QoS policers

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.



To view QoS policers counters, it is necessary to configure **counters enabled** in policer instances.

```
show qos policer  
show qos policer resources
```

14 Security

Maintain security in the network consists in adopting access policies, monitoring of resources and protection of the equipment to avoid undesirable attacks.

This chapter describes how to set some security functionalities and resources available in the DmOS. It contains the following sections:

- Storm Control Configuration
- ACL Configuration
- Anti IP Spoofing Configuration
- MAC Limit Configuration
- CPU DoS Protect Configuration

14.1 Storm Control Configuration

The Storm Control is a traffic control resource that avoids that physical interfaces become impacted by a broadcast, unknown multicast or unknown unicast traffic (DLF). This flood can occur with traffic attack or during a network convergence until learning MAC addresses by switches.



Unknown multicast is considered when equipment does not have IGMP snooping enabled in bridge-domain and interface which have multicast storm control enabled.



The value specified for traffic control is a percentage of the rated speed of the interface that can be specified from 0 to 100 in steps of 0.01. The configuration in 100% will allow the entire set type traffic configured which is similar when not having a feature enabled in the interface.

14.1.1 Configuring Storm Control

The next steps will indicate how to set the Storm Control the interface gigabit 1/1/1 to suppress the broadcast traffic in **5%**, approximately 50 Mbps of the interface, the unknown multicast traffic in **5%**, approximately 50 Mbps and the unknown unicast traffic (DLF) in **5%**, approximately 50 Mbps. These approximate values take into account the 1 Gbps interface.

```
config
switchport
interface gigabit-ethernet-1/1/1
storm-control
broadcast 5
multicast 5
unicast 5
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Storm Control](#).

14.1.2 Verifying Storm Control

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show interface utilization
show interface <interface> statistics
```

14.2 ACL Configuration

ACLs (Access Control Lists) have the function of permitting or denying packets, protecting the equipment against attacks or avoiding non authorized access in network resources.

DmOS supports ingress ACLs e also ACLs specific for traffic destined to equipment CPU. It is possible to deny or permit packets, or change packets properties.



ACLs do not act on traffic received or sent by MGMT interface.



Each hardware platform supports a maximum number of ACLs. Refer to the **DmOS Datasheet** to check maximum values.

- L2 Filters: Destination and Source MAC, Ethertype, PCP, Inner-PCP, VLAN e Inner-VLAN.
- L3 Filters: L2 Filters, Destination and Source IPv4, TCP/UDP Destination Port, DSCP, IP Protocol e ToS.

14.2.1 Configuring ACL L2 to deny traffic of a VLAN

The next steps will indicate how to set a L2 ACL with priority 0 in the gigabit 1/1/1 interface denying traffic of VLAN 20 in this interface.


```
config
access-list
acl-profile ingress l2 ACL-L2 priority 0
access-list-entry 0 match vlan 20
action deny
!
!
access-list interface gigabit-ethernet-1/1/1 ingress ACL-L2
commit
```



The available commands for troubleshooting can be found in the topic [Verifying ACLs](#).

14.2.2 Configuring ACL L3 to deny traffic of IPv4 Address

The next steps will indicate how to set a L3 ACL with priority 256 in the gigabit 1/1/1 interface denying traffic with 192.168.5.10 address of origin in this interface.

```
config
access-list
acl-profile ingress l3 ACL-L3 priority 256
access-list-entry 0 match source-ipv4-address 192.168.5.10
action deny
!
access-list interface gigabit-ethernet-1/1/1 ingress ACL-L3
commit
```



The available commands for troubleshooting can be found in the topic [Verifying ACLs](#).

14.2.3 Configuring an ACL for CPU protection

Packets destined to interfaces configured in the equipment (loopbacks as well l3 interfaces) are forwarded for processing in CPU, which can lead to high processing, affecting protocol, or even allowing vulnerabilities to be exploited. For those reasons, it is recommended the use of ACLs for CPU (control plane) protection, allowing only the required packets for troubleshooting (i.e. ICMP, SNMP), management (i.e. SSH) and protocols establishment.

In the example below, a ACL is configured for CPU protecting based on the following criteria:

- Allow ARP packets
- Allow ICMP IPv4 packets
- Allow ICMP IPv6 packets
- Allow SSH access only for packets with source in network 10.0.0.0/8
- Allow OSPF packets (IP protocol 89)
- Allow BGP packets (port 179)

- Allow LDP packets (port 646)
- Allow Slow Protocols - LACP, EFM, among others (ethertype 0x8809)
- Allow STP (ethertype 0x4242)
- Allow TFTP* and any other protocols only for configured servers
- Drop the remaining packets



*TFTP protocol uses port 69 on the server only for an initial negotiation, after this step the source and destination ports are random, for this reason the option we have to allow this protocol in the CPU using source IP address to permit only IP addresses for reliable servers.

The ACL was configured using name **control-plane-protection** and it was applied using the configuration command **configura  o access-list protection cpu control-plane-protection**.



Even having the desired ports permitted in the ACL, it is important that ARP and ICMP IPv6 protocol are permitted as well, as shown in entries 10 and 30 of the following configuration.

```
config
access-list
protection
  cpu control-plane-protection
!
acl-profile cpu l3 control-plane-protection
priority 0
access-list-entry 10
  match ethertype arp
  action permit
!
access-list-entry 20
  match ip-protocol icmp
  action permit
!
access-list-entry 30
  match ip-protocol ipv6-icmp
  action permit
!
access-list-entry 40
  match source-ipv4-address 10.0.0.0/8
  match destination-port ssh
  action permit
!
access-list-entry 50
  match ip-protocol 89
  action permit
!
access-list-entry 60
  match destination-port 179
  action permit
!
access-list-entry 61
  match source-port 179
  action permit
!
access-list-entry 70
  match destination-port 646
  action permit
!
access-list-entry 71
  match source-port 646
  action permit
!
access-list-entry 80
  match ethertype 0x8809
  action permit
```

```
!
access-list-entry 90
  match ethertype 0x4242
  action permit
!
access-list-entry 91
  match source-ipv4-address 10.0.0.1/32
  action permit
!
access-list-entry 100
  action deny
!
!
commit
```



The available commands for troubleshooting can be found in the topic [Verifying ACLs](#).

14.2.4 Configuring an ACL for CPU based packets

Packets originating in the CPU, such as CFM, L3, TWAMP, MPLS, can go through ACL rules before being forwarded. These rules can alter some field in the packet or even restrict it.



It has no action on packets forwarded directly to interfaces. Example of L2 protocols: EAPS/ERPS/xSTP/LACP/OAM.

In the example below an ACL is configured to change the priority (PCP) of TWAMP packets that are sent by CPU:

The ACL was configured with the name **originated-cpu** and priority 5 (set pcpc 5) was applied to packets designated port 9999 of the TWAMP reflector.



TWAMP must be configured with port 9999.

```
config
access-list
  interface cpu-port-1/1/1 ingress originated-cpu
  !
  acl-profile ingress l3 originated-cpu
  priority 256
  access-list-entry 0
    match destination-port 9999
    action set pcpc 5
  !
  !
commit
```



The available commands for troubleshooting can be found in the topic [Verifying ACLs](#).

14.2.5 Verifying ACLs

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show acl-resources
show acl-resources brief
show acl-resources detail
show acl-resources extensive
```

14.3 Anti IP Spoofing Configuration

The anti-ip-spoofing functionality is a technique that consists in protecting the spoofing interfaces in the packs, avoiding attacks of the SYN flood type, routing redirect among others.

It is possible to set rules to allow traffic of a specific IP address, all the IPV4 addresses, all the IPV6 addresses or all the IPv4 and IPv6 addresses.



This security resource is available only in OLT platform with support to the GPON technology.



For the service-port that uses the DHCP or PPPoE as authentication of GPON clients, the IP addresses will be released automatically. This config is not necessary.



It is not possible to deactivate rules in the GPON interfaces. The rules can be applied in Ethernet interfaces or in Service-ports of the GPON.

14.3.1 Configuring Anti IP Spoofing for specific IPv4 and MAC address

The next steps will indicate how to set the anti-ip-spoofing in the gigabit 1/1/3 interface, releasing the IP traffic for the 1.1.1.1 address in service-port 2 with MAC 00:AA:10:20:30:41.

```
config
anti-ip-spoofing
interface service-port-2
allowed-ip ipv4 1.1.1.1 vlan 10 mac 00:AA:10:20:30:41
```



The available commands for troubleshooting can be found in the topic [Verifying Anti IP Spoofing](#).

14.3.2 Configuring Anti IP Spoofing for specific IPv4 address

The next steps will indicate how to set the anti-ip-spoofing in service-port-2 releasing only the IP traffic for the IPv4 192.10.20.1 address.

```
config
anti-ip-spoofing
interface service-port-2
  allowed-ip ipv4 address 192.10.20.1
```



The available commands for troubleshooting can be found in the topic [Verifying Anti IP Spoofing](#).

14.3.3 Configuring Anti IP Spoofing for all IPv6 addresses

The next steps will indicate how to set the anti-ip-spoofing in service-port-2 releasing only the IP traffic for all the IPv6 addresses.

```
config
anti-ip-spoofing
interface service-port-2
  allowed-ip ipv6-all
commit
```



The available commands for troubleshooting can be found in the topic [Verifying Anti IP Spoofing](#).

14.3.4 Configuring Anti IP Spoofing for all IPv4 and IPv6 addresses

The next steps will indicate how to set the anti-ip-spoofing in service-port-2 releasing the IP traffic for all the IPv4 and IPv6 addresses.

```
config
anti-ip-spoofing
interface service-port-2
  allowed-ip all
```



The available commands for troubleshooting can be found in the topic [Verifying Anti IP Spoofing](#).

14.3.5 Verifying Anti IP Spoofing

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show allowed-ip
show allowed-ip address <IP_address>
show allowed-ip entry-type type
show allowed-ip mac <MAC>
show allowed-ip status status
show allowed-ip vlan <VLAN-ID>
```

14.4 MAC Limit Configuration

The MAC limit is the quantity of MAC addresses that an Ethernet interface can learn. It is possible to configure MAC Limit in interfaces and VLANs.

14.4.1 Configuring MAC Limit on Interface

The next steps will indicate how to set MAC limit for 100 MACs addresses in the gigabit 1/1/3 interface.

```
config
mac-address-table
interface gigabit-ethernet-1/1/3
limit maximum 100
```



The available commands for troubleshooting can be found in the topic [Verifying MAC Limit](#).

14.4.2 Configuring MAC Limit on VLAN

The next steps will indicate how to set MAC limit for 20 MACs addresses in VLAN 3000.

```
config
mac-address-table
vlan 3000
limit maximum 20
commit
```



The available commands for troubleshooting can be found in the topic [Verifying MAC Limit](#).

14.4.3 Verifying MAC Limit

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show mac-address-table interface <interface>
show mac-address-table interface <interface> | linnum | begin 3 | count
show mac-address-table vlan <vlan>
show mac-address-table vlan <vlan> | linnum | begin 3 | count
```

14.5 CPU DoS Protect Configuration

It is important to control the number of packets sent to the CPU in order to guarantee the quality, availability of services and to avoid attacks that may cause the loss of management or a wrong state change of a protocol.

DmOS allows the user to control the maximum number of packets sent per second (pps) to the equipment's CPU through global configuration or per protocol.



The configuration of the number of packets in the CPU DoS Protect must be planned according to the network. Setting a value too low can cause malfunction in some protocol, just as setting a value too high can leave the equipment susceptible to attacks directed at the CPU.



On the DM4770 platform the default value for the CPU DoS Protect is 3000 pps, on the other platforms the default value is 900 pps.

14.5.1 Configuring the CPU DoS Protect

The next steps will demonstrate how to configure the CPU DoS Protect to limit the maximum number of packets that the CPU must accept to 1000 packets per second (pps).

```
config
cpu-dos-protect global
max-pps 1000
commit
```



There are no troubleshooting commands for this functionality.

14.5.2 Configuring the CPU DoS Protect per Protocol

It is possible to configure the maximum number of packets for different protocols, such as **ARP**, **LLDP**, **PPPoE**, **VRRP**, among others.

The next steps will demonstrate how to update the maximum number of packets per second that the CPU can accept for the ARP and PPPoE protocols. Excess packets from each queue will be discarded.

```
config
cpu-dos-protect protocols arp max-pps 90
cpu-dos-protect protocols pppoe max-pps 800
commit
```



The available commands for troubleshooting can be found in the topic [Verifying the CPU DoS Protect](#).

14.5.3 Verifying the CPU DoS Protect

Below are the main commands available to check the feature. If the user is at the config level, the usage of the keyword **do** before the command is required.



For more details about commands output, check the **Command Reference**.

```
show cpu-dos-protect protocols
```


Legal Note

In spite the fact that all the precautions were taken in development of the present document, DATACOM shall not be held responsible for eventual errors or omissions as well as no obligation is assumed due to damages resulting from the use of the information included in this guide. The specifications provided in this manual shall be subject to changes with no prior notification and are not acknowledged as any type of contract.

© 2021 DATACOM - All rights reserved.

Warranty

DATACOM's products are covered by a warranty against manufacturing defects during a minimum period of 12 (twelve) months including the legal term of 90 days, as from the date of issue of the supply Nota Fiscal (Invoice).

Our warranty is standard counter warranty, this means, for exercise of the warranty, the customer should send the product to DATACOM Authorized Technical Assistance with paid freight. The return freight of the equipment will be DATACOM responsibility.

To obtain additional information, see our warranty policy in <https://www.datacom.com.br/en>.

Telephone Number: **+55 51 3933-3094**