

AfNOG MPLS Lab

May 14, 2006
Nairobi, Kenya

Table of Contents

Assignment	3
Topology	3
Connection information	3
IP Addressing	4
Initial Configuration	4
Lab Modules	5
Module 1: Basic MPLS	5
Module 2: MPLS VPN	11
Part 1: VRF configuration.....	11
Part 2: Configure VRF membership on the interfaces	12
Part 3: Configure MP-BGP between PEs and RRs	13
Part 4: Configure CE-PE routing protocols	16
Part 5: MPLS VPN verification	19
Appendix.....	24
Appendix.....	24

Assignment

You are a network Engineer at AFRONET Inc., a growing Service Provider with presence throughout Africa. AFRONET has embarked on a multi-year modernization of its network, to migrate to the next generation packet switched network.

You and others on your team have been asked to create an MPLS network connecting three POPs (Point of Presence) in Nairobi, Lagos and Johannesburg.

In the first phase of the project, you have to:

- Create an MPLS network between the three POPs.
- Provide MPLS VPN service to Customer 1 (Green Bank Inc.)
- Provide MPLS VPN service to Customer 2 (Blue Insurance Co.)
- Provide Layer 2 VPN connection from Site 1 to Site 3 for Customer 2 (Blue Insurance Co.)

Topology

The network topology is shown in figure 1. Details about the addressing, port information are shown in the appendix.

Connection information

To connect to the routers, use telnet from your PC using the following addresses.

Router	Console connection address
PE1	
P1	
PE2	
P2	
PE3	
P3	
RR1	
RR2	
IGW	
Green CE1	
Green CE2	
Green CE3	
Blue CE1	
Blue CE2	
Blue CE3	

IP Addressing

Loopback address for the routers is shown in the table below.

Router	Loopback address
PE1	10.0.0.1/32
P1	10.0.0.2/32
PE2	10.0.0.3/32
P2	10.0.0.4/32
PE3	10.0.0.5/32
P3	10.0.0.6/32
RR1	10.0.0.7/32
RR2	10.0.0.8/32
IGW	10.0.0.9/32

Links IP addresses

Link	Subnet address
PE1 – P1	41.0.0.0/24
PE1 – P2	41.0.1.0/24
PE2 – P1	41.0.2.0/24
PE2 – P2	41.0.3.0/24
PE3 – P2	41.0.4.0/24
PE3 – P3	41.0.5.0/24
P1 – P2	41.0.6.0/24
P1 – P3	41.0.7.0/24
P2 – P3	41.0.8.0/24
P1 – RR1	41.0.9.0/24
P3 – RR2	41.0.10.0/24
P3 – IGW	41.0.11.0/24

Initial Configuration

All routers have some initial configurations.

- All interfaces within the MPLS cloud are up and have IP addresses assigned to them.
- OSPF is configured in the MPLS cloud. All routers are in OSPF Area 0.
- The BGP Route Reflectors have BGP configurations already configured. BGP in the MPLS Core is using AS 1.

Lab Modules

Module 1: Basic MPLS

1. Enable MPLS globally on all AFRONET routers (PEs, Ps, RRs, and IGW).

First though, you have to enable CEF switching on all routers, as in the example shown below. On Cisco routers, MPLS requires CEF switching.

```
PE2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
PE2(config)#ip cef
```

Now enable MPLS globally.

```
PE2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
PE2(config)#mpls ip
PE2(config)#mpls label protocol ldp
PE2(config)#mpls ldp router-id loopback 0 force
```

As you can see in the example above, we executed three different commands on the router.

mpls ip enables MPLS globally on the router;

mpls label protocol ldp sets the label distribution protocol to LDP (as opposed to TDP, which is the default);

mpls ldp router-id loopback 0 force sets the router ID of the LDP session to the IP address of the loopback interface. If we do not have this command, the largest IP address on the router will be picked. However, to ensure stability and easy identification of LDP sessions, it is highly recommended to use a logical interface's IP address as the LDP router ID.

2. Check your configuration on your router. Do you see your configuration in the running-configuration?

```
PE2#sh run
Building configuration...
:
<SNIP>
:
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0 force
:
<SNIP>
```

Did you notice that your ***mpls ldp router-id loopback 0 force*** is displayed differently? On Cisco routers, ***mpls*** and ***tag-switching*** are used interchangeably.

3. To fully use MPLS, you also need to enable MPLS on the appropriate interfaces.

```
PE2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
PE2(config)#int e0/0
PE2(config-if)#mpls ip
PE2(config-if)#int e1/0
PE2(config-if)#mpls ip
```

AfNOG MPLS Lab

- Verify that the interfaces are now MPLS capable.

```
PE2#sh mpls interfaces
Interface      IP      Tunnel  Operational
Ethernet0/0    Yes (ldp) No      Yes
Ethernet1/0    Yes (ldp) No      Yes
PE2#
```

This CLI shows you that the two interfaces are running MPLS, using LDP, and are operational.

Note that we have not configured MPLS on the CE facing interfaces.

- Do you have labels assigned in your MPLS forwarding table?

```
PE2#sh mpls forwarding-table
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC or Tunnel Id   switched   interface
16     Untagged  10.0.0.8/32     0          Et1/0     41.0.3.2
      Untagged  10.0.0.8/32     0          Et0/0     41.0.2.2
17     Untagged  10.0.0.9/32     0          Et1/0     41.0.3.2
      Untagged  10.0.0.9/32     0          Et0/0     41.0.2.2
18     Untagged  10.0.0.2/32     0          Et0/0     41.0.2.2
19     Untagged  10.0.0.1/32     0          Et1/0     41.0.3.2
      Untagged  10.0.0.1/32     0          Et0/0     41.0.2.2
20     Untagged  10.0.0.6/32     0          Et1/0     41.0.3.2
      Untagged  10.0.0.6/32     0          Et0/0     41.0.2.2
21     Untagged  10.0.0.7/32     0          Et0/0     41.0.2.2
22     Untagged  10.0.0.4/32     0          Et1/0     41.0.3.2
23     Untagged  10.0.0.5/32     0          Et1/0     41.0.3.2
24     Untagged  41.0.9.0/24     0          Et0/0     41.0.2.2
25     Untagged  41.0.8.0/24     0          Et1/0     41.0.3.2
26     Untagged  41.0.11.0/24    0          Et1/0     41.0.3.2
      Untagged  41.0.11.0/24    0          Et0/0     41.0.2.2
27     Untagged  41.0.10.0/24    0          Et1/0     41.0.3.2
      Untagged  41.0.10.0/24    0          Et0/0     41.0.2.2
28     Untagged  41.0.1.0/24     0          Et1/0     41.0.3.2
29     Untagged  41.0.0.0/24     0          Et0/0     41.0.2.2
30     Untagged  41.0.5.0/24     0          Et1/0     41.0.3.2
      Untagged  41.0.5.0/24     0          Et0/0     41.0.2.2
31     Untagged  41.0.4.0/24     0          Et1/0     41.0.3.2
32     Untagged  41.0.7.0/24     0          Et0/0     41.0.2.2
33     Untagged  41.0.6.0/24     0          Et1/0     41.0.3.2
      Untagged  41.0.6.0/24     0          Et0/0     41.0.2.2
```

Notice that we have assigned a label for all prefixes in our OSPF table (verify by doing a **show ip route ospf**).

What is our lowest label value?

Why is it that for 10.0.0.8 for instance, we use label 16, regardless of whether the outgoing interface is E0/0 or E1/0? (Think FEC)

Note that the Outgoing tag is showing Untagged for all prefixes. (Remember, we currently have no LDP neighbor, hence are not getting any neighbor's label bindings).

- Is our router trying to form a neighbor relationship on E0/0 and E1/0?

```
PE2#sh mpls ldp discovery
Local LDP Identifier:
 10.0.0.3:0
Discovery Sources:
Interfaces:
 Ethernet0/0 (ldp): xmit
 Ethernet1/0 (ldp): xmit
```

AfNOG MPLS Lab

As the CLI shows, LDP is transmitting Hellos on both interfaces.

7. How often are discovery hello messages being sent from your router?

Use the CLI or enable the debug commands shown below:

```
PE2#sh mpls ldp parameters
Protocol version: 1
Downstream label generic region: min label: 16; max label: 100000
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off

PE2#sh mpls ldp discovery detail
Local LDP Identifier:
 10.0.0.3:0
Discovery Sources:
Interfaces:
 Ethernet0/0 (ldp): xmit
   Hello interval: 5000 ms; Transport IP addr: 10.0.0.3
 Ethernet1/0 (ldp): xmit
   Hello interval: 5000 ms; Transport IP addr: 10.0.0.3
```

Notice that you can also get all your other timer values from the **show mpls ldp parameters** command.

```
PE2#debug mpls ldp transport events
LDP transport events debugging is on
PE2#
PE2#
*Apr 15 22:44:07.663: ldp: Send ldp hello; Ethernet1/0, src/dst
41.0.3.1/224.0.0.2, inst_id 0
*Apr 15 22:44:08.283: ldp: Send ldp hello; Ethernet0/0, src/dst
41.0.2.1/224.0.0.2, inst_id 0
PE2#
*Apr 15 22:44:11.083: ldp: Scan listening TCBS
*Apr 15 22:44:12.703: ldp: Send ldp hello; Ethernet1/0, src/dst
41.0.3.1/224.0.0.2, inst_id 0
*Apr 15 22:44:12.923: ldp: Send ldp hello; Ethernet0/0, src/dst
41.0.2.1/224.0.0.2, inst_id 0
PE2#
*Apr 15 22:44:16.723: ldp: Send ldp hello; Ethernet0/0, src/dst
41.0.2.1/224.0.0.2, inst_id 0
*Apr 15 22:44:17.243: ldp: Send ldp hello; Ethernet1/0, src/dst
41.0.3.1/224.0.0.2, inst_id 0
```

8. Now let's go to a directly connected router and enable MPLS and LDP globally and on the interfaces. At the same time, if you monitor PE2 for the debug traces, you would see:

```
PE2#debug mpls ldp transport events
LDP transport events debugging is on
PE2#debug mpls ldp transport connections
LDP transport connection events debugging is on
*Apr 16 02:48:07.718: ldp: Send ldp hello; Ethernet0/0, src/dst
41.0.2.1/224.0.0.2, inst_id 0
*Apr 16 02:48:07.958: ldp: Rcvd ldp hello; Ethernet0/0, from 41.0.2.2
(10.0.0.2:0), intf_id 0, opt 0xC
*Apr 16 02:48:07.990: ldp: ldp Hello from 41.0.2.2 (10.0.0.2:0) to
224.0.0.2, opt 0xC
*Apr 16 02:48:07.990: ldp: New adj 0x2432748 for 10.0.0.2:0, Ethernet0/0
*Apr 16 02:48:07.990: ldp: adj_addr/xport_addr 41.0.2.2/10.0.0.2
```

AfNOG MPLS Lab

```
*Apr 16 02:48:07.990: ldp: local idb = Ethernet0/0, holdtime = 15000,
peer 41.0.2.2 holdtime = 15000
*Apr 16 02:48:07.990: ldp: Link intvl min cnt = 2, intvl = 5000, idb =
Ethernet0/0
*Apr 16 02:48:07.990: ldp: Opening ldp conn; adj 0x2432748, 10.0.0.3 <->
10.0.0.2; with normal priority
*Apr 16 02:48:08.170: ldp: ldp conn is up; adj 0x2432748, 10.0.0.3:11007
<-> 10.0.0.2:646
PE2#
*Apr 16 02:48:08.378: %LDP-5-NBRCHG: LDP Neighbor 10.0.0.2:0 is UP
PE2#
```

We received an LDP hello message, formed an adjacency, and then the LDP connection came UP!!!

Let's check the state of the session using some show commands.

```
PE2#sh mpls ldp discovery
Local LDP Identifier:
 10.0.0.3:0
Discovery Sources:
Interfaces:
  Ethernet0/0 (ldp): xmit/rcv
                   LDP Id: 10.0.0.2:0
  Ethernet1/0 (ldp): xmit
PE2#
```

Note how the LDP discovery process on Ethernet0/0 is both transmitting and receiving hello messages. However, our second interface is showing that it is not receiving any hellos. That is expected right now, since we have not yet configured the remote router Ethernet1/0 connects to.

One last thing to look at here is the LDP ids. This is a 6 bytes number, with the first four bytes being the router ID and the last two, an arbitrary number.

Next, let's look at the neighbor information.

```
PE2#sh mpls ldp neighbor
Peer LDP Ident: 10.0.0.2:0; Local LDP Ident 10.0.0.3:0
TCP connection: 10.0.0.2.646 - 10.0.0.3.11007
State: Oper; Msgs sent/rcvd: 45/46; Downstream
Up time: 00:18:53
LDP discovery sources:
  Ethernet0/0, Src IP addr: 41.0.2.2
Addresses bound to peer LDP Ident:
   41.0.0.2           41.0.2.2           41.0.6.1           41.0.7.1
   41.0.9.1           10.0.0.2
PE2#
```

This command shows us that we are using TCP port 646 on the peer and port 11007 on the local router; that the LDP session to our peer is an Operational state. At the bottom of the command output, all IP addresses of local interfaces on the peer router that have label binding are shown. (i.e., these are all local addresses to the peer router).

Next, let's see if our MPLS binding table and forwarding table look different, now that we have a peer.

```
PE2#sh mpls ldp bindings
tib entry: 10.0.0.1/32, rev 10
  local binding: tag: 19
  remote binding: tsr: 10.0.0.2:0, tag: 19
tib entry: 10.0.0.2/32, rev 6
  local binding: tag: 18
  remote binding: tsr: 10.0.0.2:0, tag: imp-null
tib entry: 10.0.0.3/32, rev 8
  local binding: tag: imp-null
  remote binding: tsr: 10.0.0.2:0, tag: 18
```


AfNOG MPLS Lab

```

tib entry: 10.0.0.4/32, rev 16
    local binding: tag: 22
    remote binding: tsr: 10.0.0.2:0, tag: 22
tib entry: 10.0.0.5/32, rev 18
    local binding: tag: 23
    remote binding: tsr: 10.0.0.2:0, tag: 23
tib entry: 10.0.0.6/32, rev 12
    local binding: tag: 20
    remote binding: tsr: 10.0.0.2:0, tag: 20
tib entry: 10.0.0.7/32, rev 14
    local binding: tag: 21
    remote binding: tsr: 10.0.0.2:0, tag: 21
tib entry: 10.0.0.8/32, rev 2
    local binding: tag: 16
    remote binding: tsr: 10.0.0.2:0, tag: 16
tib entry: 10.0.0.9/32, rev 4
    local binding: tag: 17
    remote binding: tsr: 10.0.0.2:0, tag: 17
tib entry: 41.0.0.0/24, rev 30
    local binding: tag: 29
    remote binding: tsr: 10.0.0.2:0, tag: imp-null
tib entry: 41.0.1.0/24, rev 28
    local binding: tag: 28
    remote binding: tsr: 10.0.0.2:0, tag: 27
tib entry: 41.0.2.0/24, rev 34
    local binding: tag: imp-null
    remote binding: tsr: 10.0.0.2:0, tag: imp-null
tib entry: 41.0.3.0/24, rev 32
    local binding: tag: imp-null
    remote binding: tsr: 10.0.0.2:0, tag: 28
tib entry: 41.0.4.0/24, rev 38
    local binding: tag: 31
    remote binding: tsr: 10.0.0.2:0, tag: 30
tib entry: 41.0.5.0/24, rev 36
    local binding: tag: 30
    remote binding: tsr: 10.0.0.2:0, tag: 29
tib entry: 41.0.6.0/24, rev 42
    local binding: tag: 33
    remote binding: tsr: 10.0.0.2:0, tag: imp-null
tib entry: 41.0.7.0/24, rev 40
    local binding: tag: 32
    remote binding: tsr: 10.0.0.2:0, tag: imp-null
tib entry: 41.0.8.0/24, rev 22
    local binding: tag: 25
    remote binding: tsr: 10.0.0.2:0, tag: 24
tib entry: 41.0.9.0/24, rev 20
    local binding: tag: 24
    remote binding: tsr: 10.0.0.2:0, tag: imp-null
tib entry: 41.0.10.0/24, rev 26
    local binding: tag: 27
    remote binding: tsr: 10.0.0.2:0, tag: 26
tib entry: 41.0.11.0/24, rev 24
    local binding: tag: 26
    remote binding: tsr: 10.0.0.2:0, tag: 25
PE2#

```

As you can see in the output above, we have both local and remote label bindings for all our prefixes. (Tag switching acronym TIB is LIB, TSR is LSR and tag is label) And in the LFIB below, we see the outgoing labels have changed.

PE2#sh mpls forwarding-table					
Local	Outgoing	Prefix	Bytes tag	Outgoing	Next Hop
tag	tag or VC	or Tunnel Id	switched	interface	
16	16	10.0.0.8/32	0	Et0/0	41.0.2.2
17	17	10.0.0.9/32	0	Et0/0	41.0.2.2
18	Pop tag	10.0.0.2/32	0	Et0/0	41.0.2.2
19	19	10.0.0.1/32	0	Et0/0	41.0.2.2
20	20	10.0.0.6/32	0	Et0/0	41.0.2.2
21	21	10.0.0.7/32	0	Et0/0	41.0.2.2
22	Untagged	10.0.0.4/32	0	Et1/0	41.0.3.2
23	Untagged	10.0.0.5/32	0	Et1/0	41.0.3.2

AfNOG MPLS Lab

24	Pop tag	41.0.9.0/24	0	Et0/0	41.0.2.2
25	Untagged	41.0.8.0/24	0	Et1/0	41.0.3.2
26	25	41.0.11.0/24	0	Et0/0	41.0.2.2
27	26	41.0.10.0/24	0	Et0/0	41.0.2.2
28	Untagged	41.0.1.0/24	0	Et1/0	41.0.3.2
29	Pop tag	41.0.0.0/24	0	Et0/0	41.0.2.2
30	29	41.0.5.0/24	0	Et0/0	41.0.2.2
31	Untagged	41.0.4.0/24	0	Et1/0	41.0.3.2
32	Pop tag	41.0.7.0/24	0	Et0/0	41.0.2.2
33	Pop tag	41.0.6.0/24	0	Et0/0	41.0.2.2
PE2#					

9. Optionally, if you want to take a look at the LDP state machine, you can enable **debug mpls ldp session state-machine** and you will see something similar to the example shown below.

```
PE2#debug mpls ldp session state-machine
*Apr 18 23:43:33.483: ldp: ptcl_adj:41.0.2.2(0x2D6D558): Non-existent ->
Opening Xport
*Apr 18 23:43:33.483: ldp: create ptcl_adj: tp = 0x2D6D558, ipaddr =
41.0.2.2
*Apr 18 23:43:33.483: ldp: ptcl_adj:41.0.2.2(0x2D6D558): Event: Xport
opened;
Opening Xport -> Init sent
*Apr 18 23:43:35.643: ldp: ptcl_adj:41.0.2.2(0x2D6D558): Event: Rcv Init;
Init sent -> Init rcvd actv
*Apr 18 23:43:35.643: ldp: ptcl_adj:41.0.2.2(0x2D6D558): Event: Rcv KA;
Init rcvd actv -> Oper
*Apr 18 23:43:35.663: %LDP-5-NBRCHG: LDP Neighbor 10.0.0.2:0 is UP
```

10. Go ahead and configure MPLS throughout your whole network (PEs and Ps). Once you are done, use some of the show commands that we have covered thus far to see the states on different routers.

Module 2: MPLS VPN

Part 1: VRF configuration

1. Configure the Virtual Routing Forwarding instances on the PE routers.

```
PE2(config)#ip vrf green
PE2(config-vrf)#rd 10:1
PE2(config-vrf)#route-target both 10:1
```

Blah

2. Try configuring the second VRF (Blue) with the same RD value as Green.

```
PE2(config)#ip vrf blue
PE2(config-vrf)#rd 10:1
% Cannot set RD, check if it's unique
PE2(config-vrf)#
```

As you can see, the RD value for each VRF needs to be unique.

3. How about importing and exporting the VRF using the same route target for green and blue.

```
PE2(config)#ip vrf blue
PE2(config-vrf)#rd 10:2
PE2(config-vrf)#route-target both 10:1
```

This worked; hence, you cannot have the same RD but can be importing and/or exporting routes to the same route target(s).

4. How many route-targets can we import and export?

```
PE2(config)#ip vrf green
PE2(config-vrf)#route-target import 1:1
PE2(config-vrf)#route-target import 1:2
PE2(config-vrf)#route-target import 1:3
PE2(config-vrf)#route-target export 1:3
PE2(config-vrf)#route-target export 1:10
```

As you can see, there is no restriction as to how many import or export statements you can have.

Varying mix of import and export statements can be used to provide intranet and extranet access.

5. Remove the VRFs that you were playing with, and configure the correct VRF settings.

```
PE2(config)#no ip vrf green
% IP addresses from all interfaces in VRF green have been removed
PE2(config)#no ip vrf blue
% IP addresses from all interfaces in VRF blue have been removed
```

The numbering convention we will use is as follows.

- RD: <Customer_number>:<Site_number>
- Route target: 10:<customer_number>

AfNOG MPLS Lab

This assumes that Green is Customer 1 and Blue is Customer 2. So for example, on PE2, VRF green will have RD of 1:2 and route-target of 10:1. The correct configuration is:

```
PE2(config)#ip vrf green
PE2(config-vrf)#rd 1:2
PE2(config-vrf)#route-target both 10:1
PE2(config)#ip vrf blue
PE2(config-vrf)#rd 2:2
PE2(config-vrf)#route-target both 10:2
```

Part 2: Configure VRF membership on the interfaces

The PE interfaces facing the CEs have the following addresses assigned. Note that the CEs have already been configured as part of the base configuration. Use x.x.x.1 on the PE side and assume x.x.x.2 on the CE side.

Link	Subnet
PE1 – Green CE1	11.1.1.0/24
PE1 – Blue CE1	21.1.1.0/24
PE2 – Green CE2	12.1.1.0/24
PE2 – Blue CE2	22.1.1.0/24
PE3 – Green CE3	31.1.1.0/24
PE3 – Blue CE3	32.1.1.0/24

1. Configure the CE facing interfaces with VRF membership. Follow the link assignments of network topology.

```
PE1(config)#int s2/0
PE1(config-if)#ip vrf for
PE1(config-if)#ip vrf forwarding green
PE1(config-if)#ip add 11.1.1.1 255.255.255.0
PE1(config)#int s3/0
PE1(config-if)#ip vrf for
PE1(config-if)#ip vrf forwarding blue
PE1(config-if)#ip add 21.1.1.1 255.255.255.0
```

2. Verify that the interfaces are now part of the VRFs.

```
PE1#sh ip vrf
Name                Default RD          Interfaces
blue                 2:1                Serial3/0
green                1:1                Serial2/0
```

Or use the detailed version to also see the import/export RT values, and other information.

```
PE1#sh ip vrf detail
VRF blue; default RD 2:1; default VPNID <not set>
  Interfaces:
    Serial3/0
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:10:2
  Import VPN route-target communities
    RT:10:2
  No import route-map
  No export route-map
  CSC is not configured.
VRF green; default RD 1:1; default VPNID <not set>
```

AfNOG MPLS Lab

```
Interfaces:
  Serial2/0
Connected addresses are not in global routing table
Export VPN route-target communities
  RT:10:1
Import VPN route-target communities
  RT:10:1
No import route-map
No export route-map
CSC is not configured.
PE1#
```

3. Now try pinging the CE interface from the attached PE. For example, from PE1, ping Blue CE1.

```
PE1#ping 21.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 21.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The ping failed!!! But why? (Hint: check the global routing table on the PE)

4. Remember that a VRF will create its own routing and forwarding table that no one outside that VRF will be able to see. When you tried to ping, the router looked for the address in its global routing table, and as you can see below, the address is not present there.

```
PE1#sh ip route 21.1.1.2
% Network not in table
```

Instead, if we look at the VRF blue routing table, we see that the address is directly connected via Serial3/0.

```
PE1#sh ip route vrf blue 21.1.1.2
Routing entry for 21.1.1.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
    * directly connected, via Serial3/0
      Route metric is 0, traffic share count is 1
```

So if you try pinging it from within the VRF, BINGO!! It works...

```
PE1#ping vrf blue 21.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 21.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/20/28 ms
```

For fun, you can also try pinging the address of the green VRF from within blue VRF.

Part 3: Configure MP-BGP between PEs and RRs

The next step in our MPLS VPN configuration is to enable MP-BGP for VPNv4 route distribution.

1. Enable BGP on the PE routers.

AfNOG MPLS Lab

```
PE3(config)#router bgp 1
PE3(config-router)#bgp log-neighbor-changes
PE3(config-router)#no bgp default ipv4-unicast
PE3(config-router)#neighbor 10.0.0.7 remote-as 1
PE3(config-router)#neighbor 10.0.0.7 update-source Loopback0
PE3(config-router)#neighbor 10.0.0.8 remote-as 1
PE3(config-router)#neighbor 10.0.0.8 update-source Loopback0
```

2. Let's see if we now have a BGP session between the PE and the RRs.

```
PE3#sh ip bgp summary

PE3#sh ip bgp neighbors

PE3#
```

Nothing is there as of yet. That is because we do not have any address-family configuration. And since we enabled the command **no bgp default ipv4-unicast**, the default IPv4 session is disabled. Now we explicitly enable each address-family that we want to support.

3. Enable the IPv4 and VPNv4 address-families.

```
PE3(config)#router bgp 1
PE3(config-router)#address-family ?
  all      All address families
  ipmcast  Address family
  ipv4     Address family
  ipv6     Address family
  nsap     Address family
  vpnv4    Address family
```

As you can see in the output above, you have several choices of address-family to use. Quiz: Which one of these do we really need to support MPLS VPN?

MPLS VPN requires the VPNv4 address-family. IPv4 is only needed if you want the basic BGP capabilities. (Note that the choice of address-family may be different in your Cisco IOS release. The basic ones: IPv4, VPNv4, ipv6, are usually available on most releases.)

4. Enable the IPv4 address-family

```
PE3(config)#router bgp 1
PE3(config-router)#address-family ipv4
PE3(config-router-af)#neighbor 10.0.0.7 activate
PE3(config-router-af)#neighbor 10.0.0.7 send-community both
PE3(config-router-af)#neighbor 10.0.0.8 activate
PE3(config-router-af)#neighbor 10.0.0.8 send-community both
```

There are only two commands needed for enabling IPv4 address-family. The command **neighbor x.x.x.x activate** actually enables the capability in BGP. The second command **neighbor x.x.x.x send-community both** will generate extended BGP community (on top of the standard communities).

Note that enabling IPv4 address-family is not required if you will only have VPNv4 services.

5. Enable VPNv4 address-family

```
PE3(config)#router bgp 1
PE3(config-router)#address-family vpnv4
PE3(config-router-af)#neighbor 10.0.0.7 activate
PE3(config-router-af)#neighbor 10.0.0.7 send-community both
```

AfNOG MPLS Lab

```
PE3(config-router-af)#neighbor 10.0.0.8 activate
PE3(config-router-af)#neighbor 10.0.0.8 send-community both
```

6. Let's take a look at the running configuration on the router

```
PE3#sh run | beg router bgp
router bgp 1
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 10.0.0.7 remote-as 1
  neighbor 10.0.0.7 update-source Loopback0
  neighbor 10.0.0.8 remote-as 1
  neighbor 10.0.0.8 update-source Loopback0
  !
  address-family ipv4
  neighbor 10.0.0.7 activate
  neighbor 10.0.0.7 send-community both
  neighbor 10.0.0.8 activate
  neighbor 10.0.0.8 send-community both
  no auto-summary
  no synchronization
  exit-address-family
  !
  address-family vpnv4
  neighbor 10.0.0.7 activate
  neighbor 10.0.0.7 send-community both
  neighbor 10.0.0.8 activate
  neighbor 10.0.0.8 send-community both
  exit-address-family
  !
  address-family ipv4 vrf green
  no auto-summary
  no synchronization
  exit-address-family
  !
  address-family ipv4 vrf blue
  no auto-summary
  no synchronization
  exit-address-family
  !
ip classless
no ip http server
:
:
```

We have some additional lines that we have not configured, but are still showing up in the running-config. That is because we have created some VRFs and assigned them to interfaces. Each time you have an interface which is a member of a VRF; there will be an entry in the BGP configuration in the form of **address-family ipv4 vrf <name>**.

7. Let's see if the MP-BGP session is up and we have the BGP neighborhood established for IPv4 and VPNv4 address families.

```
PE3#sh ip bgp ipv4 unicast summary
BGP router identifier 10.0.0.5, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor      V    AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down  State/PfxRcd
10.0.0.7      4    1   1660   1654     1    0    0 1d03h    0
10.0.0.8      4    1   1660   1654     1    0    0 1d03h    0

PE3#sh ip bgp vpnv4 all summary
BGP router identifier 10.0.0.5, local AS number 1
BGP table version is 23, main routing table version 23
19 network entries using 2527 bytes of memory
27 path entries using 1728 bytes of memory
10/9 BGP path/bestpath attribute entries using 1160 bytes of memory
4 BGP rrinfo entries using 96 bytes of memory
```

AfNOG MPLS Lab

```
2 BGP AS-PATH entries using 48 bytes of memory
5 BGP extended community entries using 208 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 5767 total bytes of memory
BGP activity 19/0 prefixes, 27/0 paths, scan interval 15 secs

Neighbor      V    AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down  State/PfxRcd
10.0.0.7      4    1   1660   1654     23    0    0 1d03h      0
10.0.0.8      4    1   1660   1654     23    0    0 1d03h      0
PE3#
```

The first show command shows the neighbors for IPv4 address family and the second command for VPNv4 address family. The two neighbors are the RRs. You can know that the sessions are operational because of the numbers shown in the last column. If you see an number instead of Init, Active, Down, etc..., it means that the neighbor is up. In our case, we are not receiving any prefixes from this neighbor (PfxRcd means Prefex received) so that is why it is showing zero at this point in time.

An alternative command would be the **show ip bgp vpnv4 all neighbor** command. This is a longer output, so is not shown here. (See Appendix for an example output)

Part 4: Configure CE-PE routing protocols

Next in our configuration steps is to set up the CE-PE protocol for each VRF within each CE-PE pairs. Refer to the MPLS VPN topology diagram in the Appendix section.

MPLS VPN allows the use of several protocols for CE-PE routing exchange. These are:

- eBGP
- OSPF
- IS-IS
- EIGRP
- RIPv2
- Static routes

It is important to note that the protocols used for one VRF (green for example) on different PEs may not need to be the same. For VRF green for example, we use OSPF on PE1, eBGP on PE2 and RIPv2 on PE3.

Regardless of what protocol you (or your customer) choose(s) to deploy, there are a few steps that you need to undertake for successfully configuring MPLS VPN.

- Enable the routing protocol on the CE
- Enable the routing protocol on the PE
- Redistribute between iBGP and your CE-PE protocol

If a protocol has some quirks, it will be noted.

1. Enable the routing protocol on the CE side.

Let's take the CE1-PE1 VRF green as an example. We are running OSPF here, and the CE1 is already configured as follows:

```
router ospf 10
 log-adjacency-changes
 passive-interface Serial1/0
 network 11.1.1.0 0.0.0.255 area 0
 network 11.100.1.0 0.0.0.255 area 0
```


AfNOG MPLS Lab

2. Enable the routing protocol on the PE side.

In this case, we have to create a new OSPF process (as process 1 is being used as the core routing protocol).

The configuration also requires the use the *router ospf <process> vrf <name>* nomenclature.

Finally, we also need to use the same process number on the CE and PE. This is a must for OSPF to work.

The final configuration looks like this:

```
router ospf 10 vrf green
 log-adjacency-changes
 network 10.0.0.1 0.0.0.0 area 0
 network 11.1.1.0 0.0.0.255 area 0
```

If you look at the OSPF status, you would see something like this:

```
PE1#sh ip ospf 10
 Routing Process "ospf 10" with ID 11.1.1.1
   Domain ID type 0x0005, value 0.0.0.10
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Connected to MPLS VPN Superbackbone, VRF green
 It is an area border router
 Initial SPF schedule delay 5000 msec
 Minimum hold time between two consecutive SPF's 10000 msec
 Maximum wait time between two consecutive SPF's 10000 msec
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msec
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msec
 Retransmission pacing timer 66 msec
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
   Area BACKBONE(0)
     Number of interfaces in this area is 1
     Area has no authentication
     SPF algorithm last executed 00:00:15.848 ago
     SPF algorithm executed 10 times
     Area ranges are
     Number of LSA 2. Checksum Sum 0x012D28
     Number of opaque link LSA 0. Checksum Sum 0x000000
     Number of DCbitless LSA 0
     Number of indication LSA 0
     Number of DoNotAge LSA 0
     Flood list length 0
```

And the routing table for the VRF has the OSPF route advertised from CE1.

```
PE1#sh ip route vrf green

Routing Table: green
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
       level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
```

AfNOG MPLS Lab

```
route
  o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

  11.0.0.0/24 is subnetted, 2 subnets
C    11.1.1.0 is directly connected, Serial2/0
O    11.100.1.0 [110/128] via 11.1.1.2, 00:06:22, Serial2/0
  12.0.0.0/24 is subnetted, 1 subnets
B    12.1.1.0 [200/0] via 10.0.0.3, 3d01h
  13.0.0.0/24 is subnetted, 1 subnets
B    13.1.1.0 [200/0] via 10.0.0.5, 01:54:47
```

3. Redistribute between iBGP and your CE-PE protocol

The last step left is to do a mutual redistribution between iBGP and the CE-PE protocol in use. Before doing that, let's see if the iBGP table for the VRF has any route information.

```
PE1#sh ip bgp vpv4 vrf green
```

As you can see, the table is empty. There are a few things we need to do here.

First, let's do a **redistribute connected subnets** under BGP so that the remote CE/PE routers can reach the CE-PE link.

```
PE1(config)#router bgp 1
PE1(config-router)#address-family ipv4 vrf green
PE1(config-router-af)#redistribute connected
```

Secondly, let's **redistribute OSPF into BGP**,

```
PE1(config)#router bgp 1
PE1(config-router)#address-family ipv4 vrf green
PE1(config-router-af)#redistribute ospf 10 vrf green
```

And lastly, let's **redistribute BGP into OSPF**.

```
PE1(config)#router ospf 10 vrf green
PE1(config-router)#redistribute bgp 1 subnets
```

4. Let's do some verification now, that we have completed the configuration part.

Repeating the command in step 3, we now get some routes in our iBGP table for the VRF.

```
PE1#sh ip bgp vpv4 vrf green
BGP table version is 29, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
                r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf green)
*> 11.1.1.0/24      0.0.0.0            0          32768 ?
*> 11.100.1.0/24   11.1.1.2           128         32768 ?
```

Next, let's check the label this route gets assigned by BGP.

```
PE1#sh ip bgp vpv4 vrf green labels
```

AfNOG MPLS Lab

Network	Next Hop	In label/Out label
Route Distinguisher: 1:1 (green)		
11.1.1.0/24	0.0.0.0	38/aggregate(green)
11.100.1.0/24	11.1.1.2	39/nolabel


```
PE1#sh mpls forwarding-table vrf green
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC   or Tunnel Id    switched  interface
38     Aggregate  11.1.1.0/24[V]  0         0         point2point
39     Untagged   11.100.1.0/24[V] 0         Se2/0     point2point
```

There are two command outputs shown, both show the same results, for the most part. Prefix 11.1.1.0/24 is shown as aggregate and both routes have a [V] designation, which means they are VPN routes. In this case, label 38 and 39 are assigned for these routes.

Part 5: MPLS VPN verification

Check the Appendix for complete configuration for all routers. Once the whole network is fully configured, we can do further verifications using show commands.

Command1: show ip route vrf <name> - this will show us the routes that each VRF has.

```
PE3#sh ip route vrf green
Routing Table: green
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
       level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
       route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
R       1.1.1.1 [120/1] via 13.1.1.2, 00:00:07, Serial2/0
    11.0.0.0/24 is subnetted, 2 subnets
B       11.1.1.0 [200/0] via 10.0.0.1, 00:03:09
B       11.100.1.0 [200/128] via 10.0.0.1, 00:03:09
    12.0.0.0/24 is subnetted, 2 subnets
B       12.1.1.0 [200/0] via 10.0.0.3, 00:03:09
B       12.100.1.0 [200/0] via 10.0.0.3, 00:03:09
    13.0.0.0/24 is subnetted, 2 subnets
C       13.1.1.0 is directly connected, Serial2/0
R       13.100.1.0 [120/1] via 13.1.1.2, 00:00:07, Serial2/0
PE3#

PE3#sh ip route vrf blue
Routing Table: blue
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
```

AfNOG MPLS Lab

```

level-2
  ia - IS-IS inter area, * - candidate default, U - per-user static
route
  o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    21.0.0.0/24 is subnetted, 2 subnets
B       21.1.1.0 [200/0] via 10.0.0.1, 00:03:15
B       21.100.1.0 [200/0] via 10.0.0.1, 00:03:15
    23.0.0.0/24 is subnetted, 2 subnets
C       23.1.1.0 is directly connected, Serial3/0
D       23.100.1.0 [90/2681856] via 23.1.1.2, 00:04:25, Serial3/0
    22.0.0.0/24 is subnetted, 2 subnets
B       22.1.1.0 [200/0] via 10.0.0.3, 00:03:15
B       22.100.1.0 [200/0] via 10.0.0.3, 00:03:15
  
```

Command2: show ip bgp vpnv4 vrf <green> - this will show us the routes that iBGP knows about. If we have done things right, we should see all our VRF routes from all sites here.

```

PE3#sh ip bgp vpnv4 vrf green
BGP table version is 33, local router ID is 10.0.0.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:3 (default for vrf green)
*> 1.1.1.1/32       13.1.1.2           1           32768 ?
*>i11.1.1.0/24      10.0.0.1           0           100    0 ?
*>i11.100.1.0/24    10.0.0.1           128          100    0 ?
*>i12.1.1.0/24      10.0.0.3           0           100    0 ?
*>i12.100.1.0/24    10.0.0.3           0           100    0 30 i
*> 13.1.1.0/24      0.0.0.0            0           32768 ?
*> 13.100.1.0/24    13.1.1.2           1           32768 ?
PE3#

PE3#sh ip bgp vpnv4 vrf blue
BGP table version is 33, local router ID is 10.0.0.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 2:3 (default for vrf blue)
*>i21.1.1.0/24      10.0.0.1           0           100    0 ?
*>i21.100.1.0/24    10.0.0.1           0           100    0 20 i
*>i22.1.1.0/24      10.0.0.3           0           100    0 ?
*>i22.100.1.0/24    10.0.0.3           0           100    0 ?
*> 23.1.1.0/24      0.0.0.0            0           32768 ?
*> 23.100.1.0/24    23.1.1.2           2681856      32768 ?
  
```

Command3: show ip bgp vonv4 vrf <name> <prefix> - this will show us more detailed information about the route.

```

PE3#sh ip bgp vpnv4 vrf green 11.100.1.1
BGP routing table entry for 1:3:11.100.1.0/24, version 20
Paths: (1 available, best #1, table green)
  Not advertised to any peer
  Local, imported path from 1:1:11.100.1.0/24
    10.0.0.1 (metric 21) from 10.0.0.7 (10.0.0.7)
      Origin incomplete, metric 128, localpref 100, valid, internal, best
      Extended Community: RT:10:1 OSPF DOMAIN ID:0x0005:0x0000000A0200
  OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:11.1.1.1:512
    Originator: 10.0.0.1, Cluster list: 10.0.0.7
  
```

AfNOG MPLS Lab

```
PE3#sh ip bgp vpnv4 vrf blue 22.100.1.1
BGP routing table entry for 2:3:22.100.1.0/24, version 27
Paths: (1 available, best #1, table blue)
Not advertised to any peer
Local, imported path from 2:2:22.100.1.0/24
 10.0.0.3 (metric 21) from 10.0.0.7 (10.0.0.7)
   Origin incomplete, metric 0, localpref 100, valid, internal, best
   Extended Community: RT:10:2
   Originator: 10.0.0.3, Cluster list: 10.0.0.7
```

Command4: show ip bgp vpnv4 vrf <name> labels - this will show us the VPN labels that BGP has assigned to each route.

```
PE3#sh ip bgp vpnv4 vrf green labels
Network          Next Hop      In label/Out label
Route Distinguisher: 1:3 (green)
 1.1.1.1/32      13.1.1.2     37/nolabel
11.1.1.0/24      10.0.0.1     nolabel/38
11.100.1.0/24    10.0.0.1     nolabel/39
12.1.1.0/24      10.0.0.3     nolabel/34
12.100.1.0/24    10.0.0.3     nolabel/35
13.1.1.0/24      0.0.0.0      34/aggregate(green)
13.100.1.0/24    13.1.1.2     38/nolabel

PE3#

PE3#sh ip bgp vpnv4 vrf blue labels
Network          Next Hop      In label/Out label
Route Distinguisher: 2:3 (blue)
21.1.1.0/24      10.0.0.1     nolabel/36
21.100.1.0/24    10.0.0.1     nolabel/35
22.1.1.0/24      10.0.0.3     nolabel/36
22.100.1.0/24    10.0.0.3     nolabel/37
23.1.1.0/24      0.0.0.0      35/aggregate(blue)
23.100.1.0/24    23.1.1.2     36/nolabel
```

Command5: show ip cef vrf <name> <prefix> detail - this will show us the LFIB information for the prefix, including the label stack that will be imposed onto traffic to that destination.

```
PE3#sh ip cef vrf green 11.100.1.1 detail
11.100.1.0/24, version 12, epoch 0, cached adjacency 41.0.4.2
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et0/0, 41.0.4.2, tags imposed: {20 39}
via 10.0.0.1, 0 dependencies, recursive
  next hop 41.0.4.2, Ethernet0/0 via 10.0.0.1/32
  valid cached adjacency
  tag rewrite with Et0/0, 41.0.4.2, tags imposed: {20 39}

PE3#

PE3#sh ip cef vrf blue 21.100.1.1 detail
21.100.1.0/24, version 11, epoch 0, cached adjacency 41.0.4.2
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et0/0, 41.0.4.2, tags imposed: {20 35}
via 10.0.0.1, 0 dependencies, recursive
  next hop 41.0.4.2, Ethernet0/0 via 10.0.0.1/32
  valid cached adjacency
  tag rewrite with Et0/0, 41.0.4.2, tags imposed: {20 35}

PE3#
```

Command6: (on the CEs) show ip route - this will show us the routes that the CE has learnt from the PE it is attached to.

AfNOG MPLS Lab

```
Cust1_CE3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback1
      11.0.0.0/24 is subnetted, 2 subnets
R       11.1.1.0 [120/1] via 13.1.1.1, 00:00:17, Serial0/0
R       11.100.1.0 [120/1] via 13.1.1.1, 00:00:17, Serial0/0
      12.0.0.0/24 is subnetted, 2 subnets
R       12.1.1.0 [120/1] via 13.1.1.1, 00:00:17, Serial0/0
R       12.100.1.0 [120/1] via 13.1.1.1, 00:00:17, Serial0/0
      13.0.0.0/24 is subnetted, 2 subnets
C       13.1.1.0 is directly connected, Serial0/0
C       13.100.1.0 is directly connected, Serial1/0
Cust1_CE3#

Cust2_CE3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      21.0.0.0/24 is subnetted, 2 subnets
D EX   21.1.1.0 [170/2172416] via 23.1.1.1, 00:12:35, Serial0/0
D EX   21.100.1.0 [170/2172416] via 23.1.1.1, 00:12:35, Serial0/0
      23.0.0.0/24 is subnetted, 2 subnets
C       23.1.1.0 is directly connected, Serial0/0
C       23.100.1.0 is directly connected, Serial1/0
      22.0.0.0/24 is subnetted, 2 subnets
D EX   22.1.1.0 [170/2172416] via 23.1.1.1, 00:12:35, Serial0/0
D EX   22.100.1.0 [170/2172416] via 23.1.1.1, 00:12:35, Serial0/0
Cust2_CE3#
```

And finally, the moment of truth is when you try pinging some of the addresses on the remote CEs.

```
Cust1_CE3#pi 11.100.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.100.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/52 ms
Cust1_CE3#pi 12.100.1.1

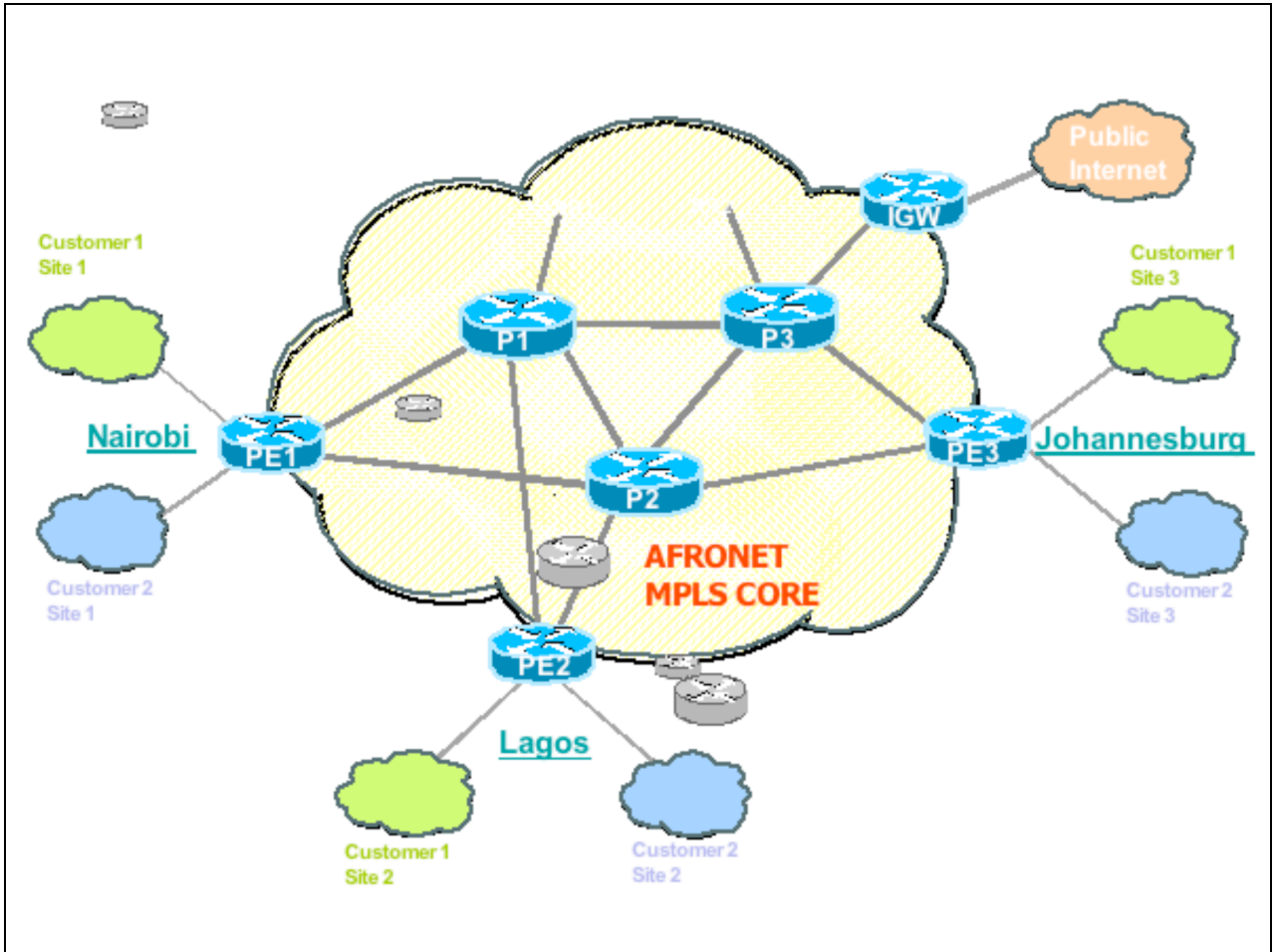
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.100.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/30/52 ms
Cust1_CE3#

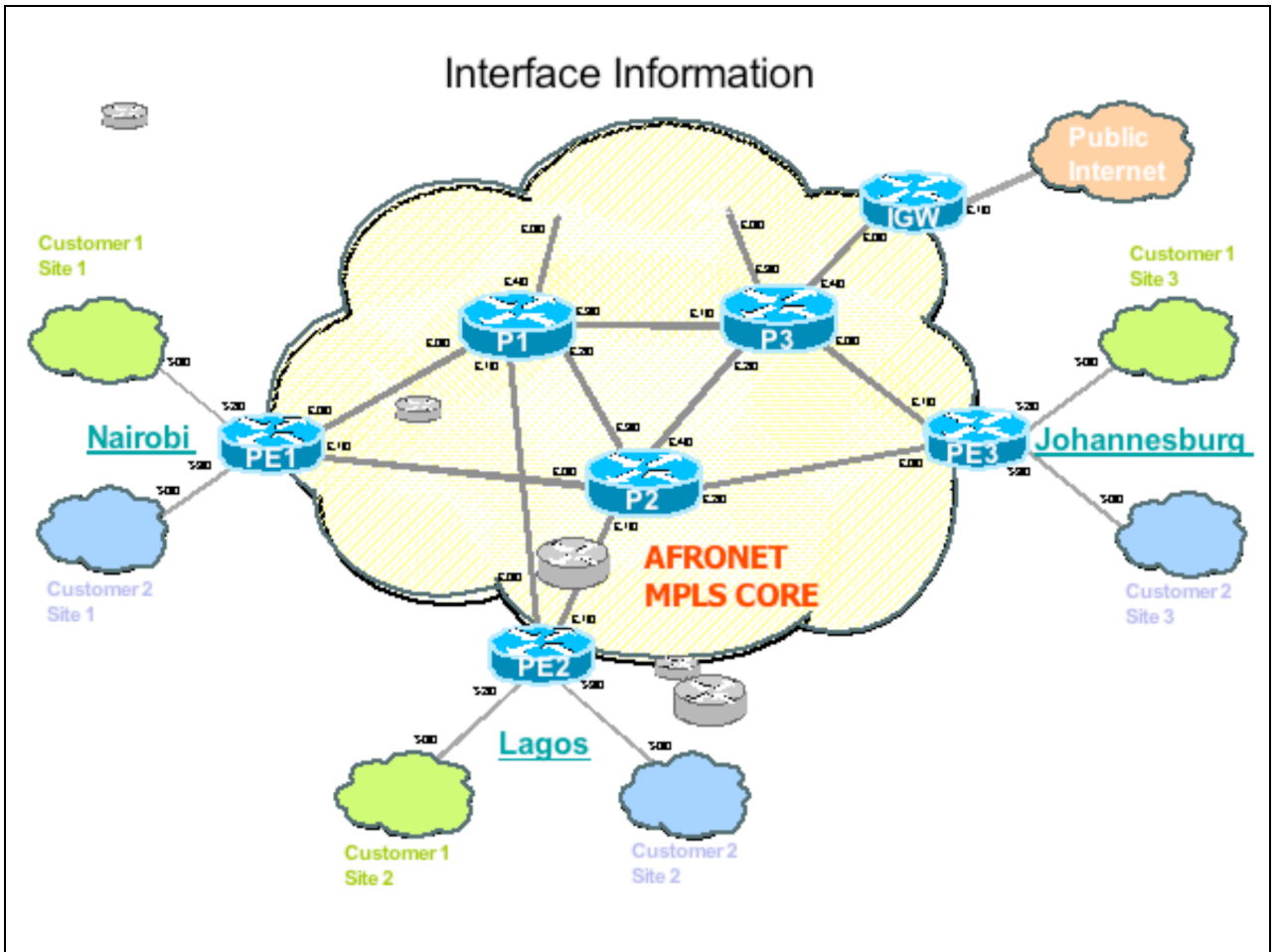
Cust1_CE3#pi 11.100.1.1
```

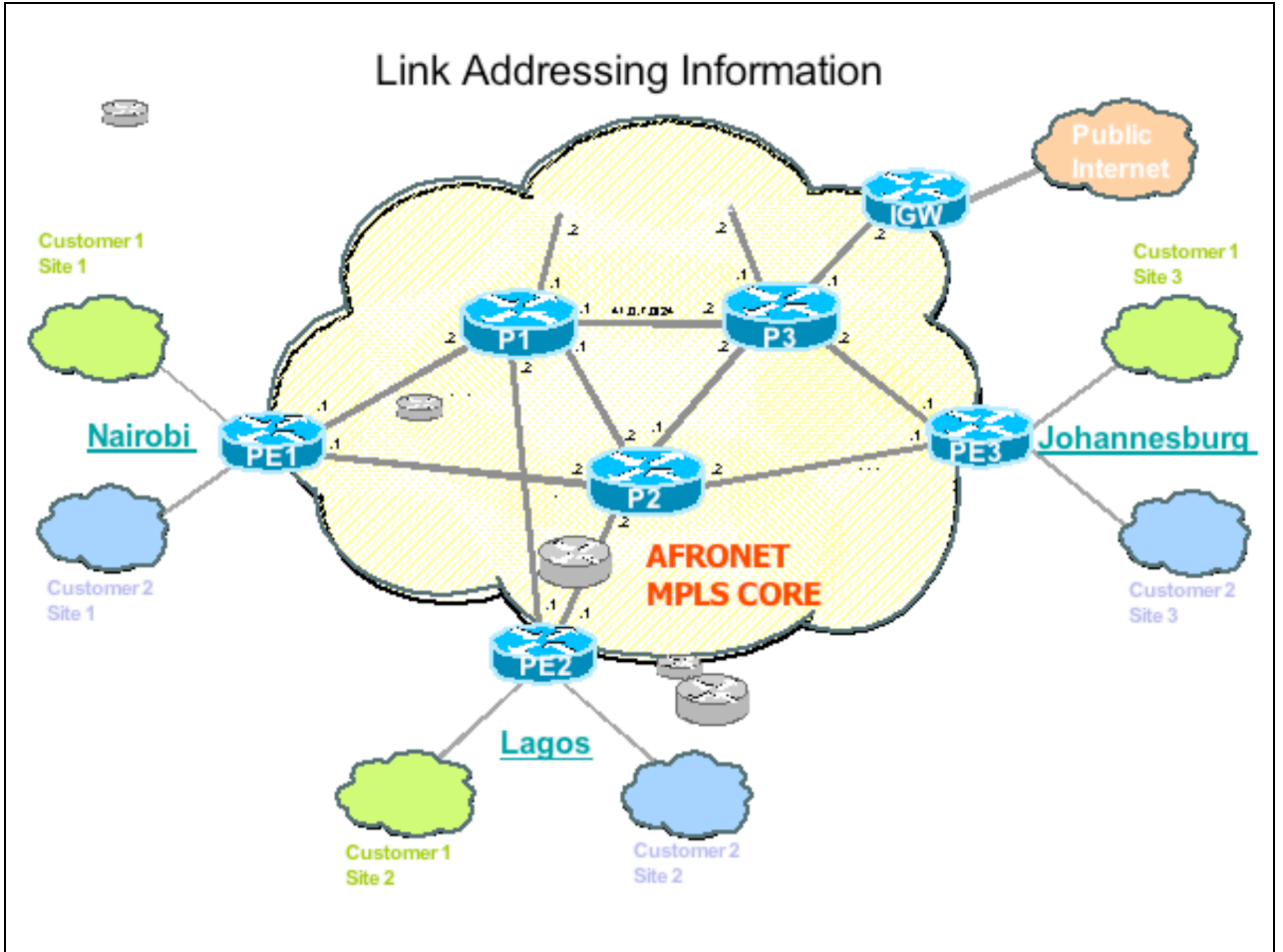
AfNOG MPLS Lab

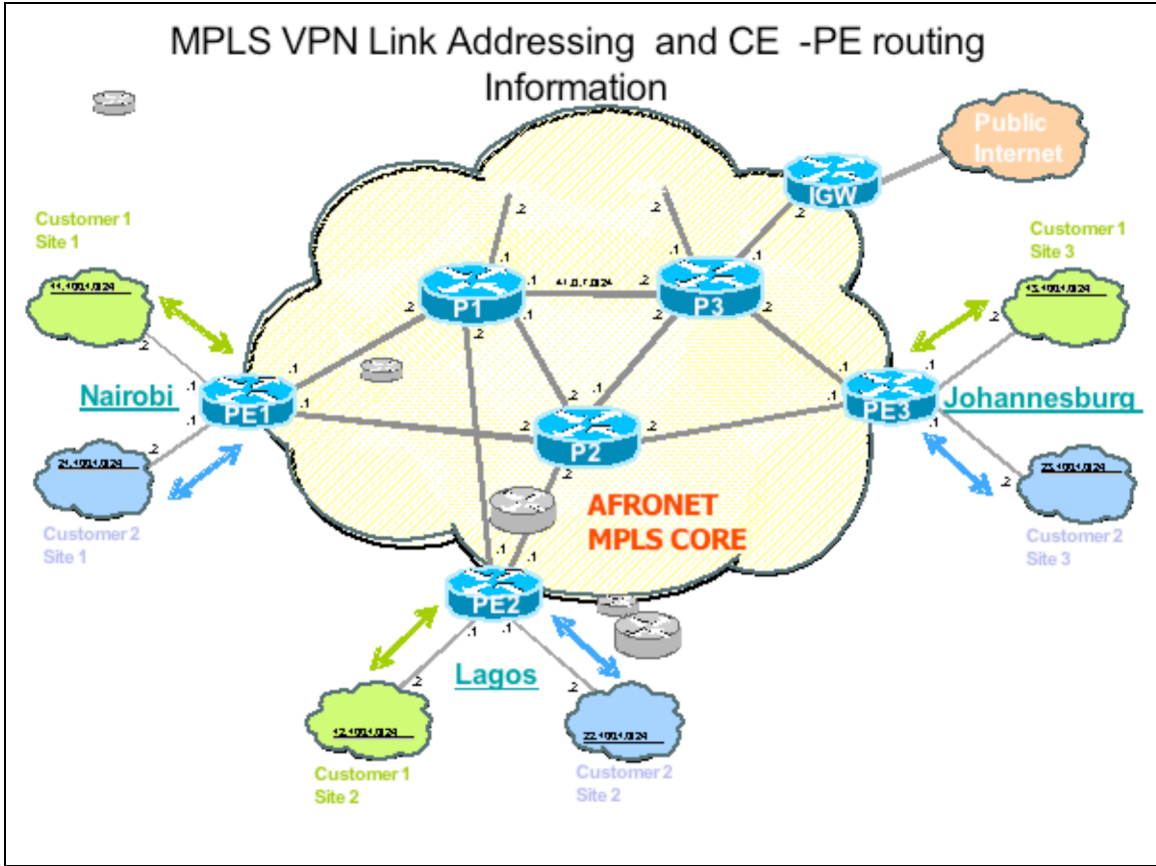
```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 11.100.1.1, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/52 ms  
Cust1_CE3#pi 12.100.1.1  
  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 12.100.1.1, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/30/52 ms  
Cust1_CE3#
```

Appendix









AfNOG MPLS Lab

Example of BGP VPNv4 neighbors with 8 prefixes advertised.

```

PE3#sh ip bgp vpnv4 all neighbors
BGP neighbor is 10.0.0.7, remote AS 1, internal link
  BGP version 4, remote router ID 10.0.0.7
  BGP state = Established, up for 1d03h
  Last read 00:00:14, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

      Sent      Rcvd
Opens:          1          1
Notifications: 0          0
Updates:        4         10
Keepalives:    1658      1658
Route Refresh: 0          0
Total:         1663      1669
  Default minimum time between advertisement runs is 5 seconds

For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue sizes : 0 self, 0 replicated
  Index 1, Offset 0, Mask 0x2
  1 update-group member
  Community attribute sent to this neighbor

    Sent      Rcvd
Prefix activity:
  Prefixes Current:      0          0
  Prefixes Total:        0          0
  Implicit Withdraw:     0          0
  Explicit Withdraw:     0          0
  Used as bestpath:      n/a         0
  Used as multipath:     n/a         0

      Outbound   Inbound
Local Policy Denied Prefixes:  -----
  Total:                      0          0
  Number of NLRIs in the update sent: max 0, min 0

For address family: VPNv4 Unicast
  BGP table version 23, neighbor version 23/0
  Output queue sizes : 0 self, 4 replicated
  Index 1, Offset 0, Mask 0x2
  1 update-group member
  Community attribute sent to this neighbor

    Sent      Rcvd
Prefix activity:
  Prefixes Current:      3          8 (Consumes 1024 bytes)
  Prefixes Total:        4          8
  Implicit Withdraw:     1          0
  Explicit Withdraw:     0          0
  Used as bestpath:      n/a        16
  Used as multipath:     n/a         0

      Outbound   Inbound
Local Policy Denied Prefixes:  -----
  ORIGINATOR loop:          n/a         3
  VPN Imported prefix:      8          n/a
  Bestpath from this peer:  8          n/a
  Total:                    16         3
  Number of NLRIs in the update sent: max 0, min 0

  Connections established 1; dropped 0
  Last reset never
  Connection state is ESTAB, I/O status: 1, unread input bytes: 0
  Connection is ECN Disabled
  Local host: 10.0.0.5, Local port: 11005
  Foreign host: 10.0.0.7, Foreign port: 179

```

AfNOG MPLS Lab

```

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x5EC009C):
Timer           Starts      Wakeups      Next
Retrans         1661         0             0x0
TimeWait        0             0             0x0
AckHold         1660         947           0x0
SendWnd         0             0             0x0
KeepAlive       0             0             0x0
GiveUp          0             0             0x0
PmtuAger        0             0             0x0
DeadWait        0             0             0x0

iss: 189968697  snduna: 190000709  sndnxt: 190000709   sndwnd: 16346
irs: 1940000973  rcvnxt: 1940033704  rcvwnd: 16213   delrcvwnd: 171

SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 12 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: active open, nagle
IP Precedence value : 6

Datagrams (max data segment is 536 bytes):
Rcvd: 2536 (out of order: 0), with data: 1669, total data bytes: 32730
Sent: 2671 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion:
0), with data: 1663, total data bytes: 32011

BGP neighbor is 10.0.0.8, remote AS 1, internal link
BGP version 4, remote router ID 10.0.0.8
BGP state = Established, up for 1d03h
Last read 00:00:15, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  Route refresh: advertised and received(old & new)
  Address family IPv4 Unicast: advertised and received
  Address family VPNv4 Unicast: advertised and received
Message statistics:
  InQ depth is 0
  OutQ depth is 0

          Sent      Rcvd
Opens:           1         1
Notifications:   0         0
Updates:         4        10
Keepalives:     1658      1658
Route Refresh:   0         0
Total:          1663      1669

Default minimum time between advertisement runs is 5 seconds

For address family: IPv4 Unicast
BGP table version 1, neighbor version 1/0
Output queue sizes : 0 self, 0 replicated
Index 1, Offset 0, Mask 0x2
1 update-group member
Community attribute sent to this neighbor

          Sent      Rcvd
Prefix activity:  ----  ----
Prefixes Current:    0         0
Prefixes Total:      0         0
Implicit Withdraw:   0         0
Explicit Withdraw:   0         0
Used as bestpath:    n/a        0
Used as multipath:   n/a         0

          Outbound   Inbound
Local Policy Denied Prefixes:  -----  -----
Total:                          0         0
Number of NLRI in the update sent: max 0, min 0

For address family: VPNv4 Unicast
BGP table version 23, neighbor version 23/0
Output queue sizes : 0 self, 4 replicated
Index 1, Offset 0, Mask 0x2
1 update-group member
Community attribute sent to this neighbor

```

AfNOG MPLS Lab

```

Prefix activity:
Prefixes Current:      Sent      Rcvd
Prefixes Total:       3        8 (Consumes 512 bytes)
Implicit Withdraw:     1        0
Explicit Withdraw:    0        0
Used as bestpath:     n/a      0
Used as multipath:    n/a      0

Local Policy Denied Prefixes:
Outbound  Inbound
-----  -----
ORIGINATOR loop:    n/a      3
VPN Imported prefix: 8        n/a
Bestpath from this peer: 8      n/a
Total:              16      3
Number of NLRIs in the update sent: max 1, min 0

Connections established 1; dropped 0
Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 10.0.0.5, Local port: 11006
Foreign host: 10.0.0.8, Foreign port: 179

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x5EC03B4):
Timer      Starts  Wakeups  Next
Retrans    1661    0        0x0
TimeWait   0        0        0x0
AckHold    1659    1049     0x0
SendWnd    0        0        0x0
KeepAlive  0        0        0x0
GiveUp     0        0        0x0
PmtuAger  0        0        0x0
DeadWait   0        0        0x0

iss: 3307919820  snduna: 3307951832  sndnxt: 3307951832  sndwnd: 16346
irs: 2466612743  rcvnxt: 2466645474  rcvwnd: 16213  delrcvwnd: 171

SRTT: 300 ms, RTTO: 303 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 12 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: active open, nagle
IP Precedence value : 6

Datagrams (max data segment is 536 bytes):
Rcvd: 2402 (out of order: 0), with data: 1669, total data bytes: 32730
Sent: 2773 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion:
0), with data: 1663, total data bytes: 32011
PE3#

```