Forwarding and L2X Configuration Guide

Version 2019.1.0, 24 September 2019

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

Table of Contents

1. Introduction to Forwarding . . . 4
2. Interface Commands . . . 5
   2.1. Display All Physical Interfaces . . . 5
   2.2. Display All Logical Interfaces . . . 5
   2.3. Delete a Logical Interface . . . 6
   2.4. Show Default Instance Interface Addresses . . . 7
   2.5. Assign Logical Interface IPv4 Address . . . 8
   2.6. Assign Logical Interface IPv6 Address . . . 9
   2.7. Disable (Shut Down) a Logical Interface . . . 10
   2.8. Enable a Logical Interface . . . 11
   2.9. Disable IPv4 on a Logical Interface . . . 12
   2.10. Disable IPv6 on a Logical Interface . . . 13
   2.11. Enable IPv4 on a Logical Interface . . . 14
   2.12. Enable IPv6 on a Logical Interface . . . 15
   2.13. Set IPv4 or IPv6 MTU Size . . . 16
   2.14. Display the IPv4 Unicast Routing Table . . . 17
   2.15. Display the IPv6 Unicast Routing Table . . . 18
   2.16. Delete a Logical Interface for an Instance . . . 19
   2.17. Display the IPv4 Unicast Routing Table for a VRF Instance . . . 19
   2.18. Display the IPv6 Unicast Routing Table for a VRF Instance . . . 20
   2.19. Display the IPv6 Labeled-Unicast Routing Table for the mgmt Instance . . . 21
   2.20. Display the IPv4 Unicast Routing Table for the mgmt Instance . . . 22
   2.21. Display the IPv6 Unicast Routing Table for the mgmt Instance . . . 23
   2.22. Display the MPLS Routing Table . . . 24
   2.23. Ping an IPv4 Address That is Part of a VRF Instance . . . 25
   2.24. Ping an IPv6 Address That is Part of a VRF Instance . . . 26
3. Layer 2 Cross-Connect (L2X) . . . 27
   3.1. L2X Features . . . 27
      3.1.1. Local Cross-connect (Local L2X) . . . 27
      3.1.2. Remote Cross-connect (Remote L2X) . . . 27
      3.1.3. Port and VLAN Cross-connects . . . 28
      3.1.4. VLAN Editing . . . 28
      3.1.5. VLAN Editing Examples . . . 29
      3.1.6. IEEE 802.1p Bit Handling . . . 30
      3.1.7. Multi-service Support . . . 30
   3.2. Configuration Profile . . . 30
   3.3. L2X Tables . . . 32
   3.4. L2X CLI Modeling . . . 33

3.5. Additional Configuration Examples 34
3.5.1. Port-based Local Cross-connect 34
3.5.2. VLAN-Based Remote Cross-connect 34
3.5.3. VLAN-Based Cross-connect with VLAN ID Editing 35
3.6. Limitation and Future Work 36

1. Introduction to Forwarding

FWDD interface commands fall into four major areas:

- Physical interface commands
- Logical interface commands
- MTU size command
- Routing table (IPv4, IPv6, VRF, MPLS) commands

2. Interface Commands

2.1. Display All Physical Interfaces

Command to display the status, MAC addresses, and bandwidth of all physical interfaces

rtb ffwd show interface physical

physical	Show physical interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface physical
+------------------------------+------------+------------
+--------------------+---------------+
            Interface           Admin Status Link Status       MAC
Address         Bandwidth
+------------------------------+------------+------------
+--------------------+---------------+
lo-0/0/0                            up            up
twc-0/12/11                         up            up
f4:1e:5e:12:01:02   100.000 Mbps
twc-0/12/13                         up            up
f4:1e:5e:12:01:03   100.000 Mbps
twc-0/12/14                         up            up
f4:1e:5e:12:01:04   100.000 Mbps
twc-0/12/17                         up            up
f4:1e:5e:12:01:07   100.000 Mbps
twc-0/12/18                         up            up
f4:1e:5e:12:01:08   100.000 Mbps
+------------------------------+------------+------------
+--------------------+---------------+
```

2.2. Display All Logical Interfaces

Command to display the status, MTU size, and other information about all logical interfaces

rtb ffws show interface logical

logical	Show logical interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.3. Delete a Logical Interface

Command to delete logical interface <interface> in instance <instance> (a show command will verify deletion)

rtb ffws delete interface logical <interface> **instance** <instance-name>

<interface>	Interface to delete <instance-name>
Name of instance to apply delete	<CR>

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd delete interface logical lo-0/0/0/0/20
instance default
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper Status: up.
```

2.4. Show Default Instance Interface Addresses

Command to display the IPv4 and IPv6 addresses used on interfaces in the default instance

rtb ffws show interface address

address	Interface address
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface address
+------------------------------+------------------
+----------------------------------------+
            Interface                IPv4 Address                IPv6
Address
+------------------------------+------------------
+----------------------------------------+        lo-0/0/0/0/1
192.168.11.2/32
 lo-0/0/0/0/1                                        192:168:11::2/128
 twc-0/0/12/11/1
fe80::2:f41e:5eff:fe12:102/128
 twc-0/0/12/13/1
fe80::2:f41e:5eff:fe12:103/128
 twc-0/0/12/14/1
fe80::2:f41e:5eff:fe12:104/128
+------------------------------+------------------
+---------------------------------------+
```

2.5. Assign Logical Interface IPv4 Address

Command to assign an IPv4 address (as primary) to a logical interface on the default instance and verify that the assignment is correct

rtb ffws set interface logical <interface> **address ipv6** <ipv6-address> **primary**

<interface>	Interface to act on
<ipv4-address>	IPv4 address to assign
primary	Set primary address
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd set interface logical lo-0/0/0/0/20
address ipv4 10.1.1.1/32 primary

ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface address
+------------------------------+------------------
+----------------------------------------+
            Interface               IPv4 Address                    IPv6
Address
+------------------------------+------------------
+----------------------------------------+
  lo-0/0/0/0/1                   192.168.11.2/32
  lo-0/0/0/0/20                  10.1.1.1/32
  lo-0/0/0/0/1                                       192:168:11::2/128
  twc-0/0/12/11/1
fe80::2:f41e:5eff:fe12:102/128
  twc-0/0/12/13/1
fe80::2:f41e:5eff:fe12:103/128
  twc-0/0/12/14/1
fe80::2:f41e:5eff:fe12:104/128
+------------------------------+------------------
+----------------------------------------+
```

2.6. Assign Logical Interface IPv6 Address

Command to assign an IPv6 address (as primary) to a logical interface on the default instance and verify that the assignment is correct

rtb ffws set interface logical <interface> **address ipv6** <ipv6-address> **primary**

<interface>	Interface to act on
<ipv6-address>	IPv6 address to assign
primary	Set primary address
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd set interface logical lo-0/0/0/0/20
address ipv6 10:1:1::1/128 primary

ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface address
+------------------------------+------------------
+----------------------------------------+
            Interface               IPv4 Address                IPv6
Address
+------------------------------+------------------
+----------------------------------------+
  lo-0/0/0/0/1                    192.168.11.2/32
  lo-0/0/0/0/20                   10.1.1.1/32
  lo-0/0/0/0/1                                        192:168:11::2/128
  lo-0/0/0/0/20                                       10:1:1::1/128
  twc-0/0/12/11/1
fe80::2:f41e:5eff:fe12:102/128
  twc-0/0/12/13/1
fe80::2:f41e:5eff:fe12:103/128
  twc-0/0/12/14/1
fe80::2:f41e:5eff:fe12:104/128
+------------------------------+------------------
+----------------------------------------+
```

2.7. Disable (Shut Down) a Logical Interface

Command to disable (shut down) a logical interface on the default instance

rtb ffws set interface logical <interface> **disable**

<interface>	Interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd set interface logical lo-0/0/0/0/20
disable

ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
    Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 9216, Status: up
    IPv6:MTU: 9216, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: down, Oper status: down
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.8. Enable a Logical Interface

Command to enable a logical interface that was previously shut down (essentially, delete the disable sent to the interface)

rtb ffws delete interface logical <interface> **disable**

<interface>	Interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd delete interface logical lo-0/0/0/0/20
disable
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 9216, Status: up
    IPv6:MTU: 9216, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.9. Disable IPv4 on a Logical Interface

Command to disable IPv4 on a logical interface on the default instance

rtb ffws set interface logical <interface> **ipv4-disable**

<interface>	Interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd set interface logical lo-0/0/0/0/20
ipv4-disable
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 9216, Status: down
    IPv6:MTU: 9216, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.10. Disable IPv6 on a Logical Interface

Command to disable IPv6 on a logical interface on the default instance

rtb ffws set interface logical <interface> **ipv6-disable**

<interface>	Interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd set interface logical lo-0/0/0/0/20
ipv6-disable
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 9216, Status: down
    IPv6:MTU: 9216, Status: down
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.11. Enable IPv4 on a Logical Interface

Command to enable IPv4 on a logical interface that was previously shut down (essentially, delete the disable sent to the interface)

rtb ffws delete interface logical <interface> **ipv4-disable**

<interface>	Interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd delete interface logical lo-0/0/0/0/20
ipv4-disable
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 9216, Status: up
    IPv6:MTU: 9216, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.12. Enable IPv6 on a Logical Interface

Command to enable IPv6 on a logical interface that was previously shut down (essentially, delete the disable sent to the interface)

rtb ffws delete interface logical <interface> **ipv6-disable**

<interface>	Interface
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd delete interface logical lo-0/0/0/0/20
ipv6-disable
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 9216, Status: down
    IPv6:MTU: 9216, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.13. Set IPv4 or IPv6 MTU Size

Command to assign an IPv4 or IPv6 MTU size to a logical interface

rtb ffws set interface logical <interface> (ipv4-mtu | ipv6-mtu) <mtu-size>

<interface>	Interface for MTU
ipv4-mtu or ipv6-mtu	Set MTU for IPv4 or IPv6
<mtu-size>	Set the size of the MTU (range: TBD)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd set interface logical lo-0/0/0/0/20
ipv4-mtu 1500

ubuntu@pod11_b_leaf2:~$ rtb confd set interface logical lo-0/0/0/0/20
ipv6-mtu 2000

ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: lo-0/0/0/0/20
    Tagged: false
    IPv4:MTU: 1500, Status: up
    IPv6:MTU: 2000, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.14. Display the IPv4 Unicast Routing Table

Command to display the prefix, source, preference, and next-hop for IPv4 unicast routes for the default instance

rtb ffws show ipv4 route unicast

unicast	Routing table to display
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show ipv4 route unicast

Routes for Instance: default
AFI: IPV4 SAFI: Unicast
+--------------------+----------+----------------
+-----------------------------------------+
           Prefix       Source      Preference                  Next-Hop
+--------------------+----------+----------------
+-----------------------------------------+
    10.1.1.1/32         direct           0         via lo-0/0/0/0/20
    192.168.11.2/32     direct           0         via lo-0/0/0/0/1
+--------------------+----------+----------------
+-----------------------------------------+
```

2.15. Display the IPv6 Unicast Routing Table

Command to display the prefix, source, preference, and next-hop for IPv6 unicast routes for the default instance

rtb ffws show ipv6 route unicast

unicast	Routing table to display
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show ipv6 route unicast
  Routes for Instance: default
    AFI: IPV6 SAFI: Unicast
+---------------------------------------------+----------
+----------------+-----------------+
                   Prefix                          Source     Preference
Next-Hop
+---------------------------------------------+----------
+----------------+-----------------+
10:1:1::1/128                                      direct           0
via lo-0/0/0/0/20
192:168:11::1/128                                  bgp             20
fe80::2:f41e:5eff:fe11:102 via twc-0/0/12/11/1
192:168:11::2/128                                  direct           0
via lo-0/0/0/0/1
192:168:11::5/128                                  bgp             20
fe80::2:f41e:5eff:fe13:102 via twc-0/0/12/13/1
192:168:11::6/128                                   bgp            20
fe80::2:f41e:5eff:fe13:102 via twc-0/0/12/13/1
+---------------------------------------------+----------
+----------------+-----------------+
```

2.16. Delete a Logical Interface for an Instance

Command to delete logical interface <interface> in instance <instance> (a show command will verify deletion)

rtb ffws delete interface logical <interface> **instance** <instance-name>

<interface>	Interface to act on
<instance-name>	Name of instance to act on (for example, default)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb confd delete interface logical lo-0/0/0/0/20
instance default
ubuntu@pod11_b_leaf2:~$ rtb fwdd show interface logical
Logical Interface Name: lo-0/0/0/0/1
    Tagged: false
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/11/1
    Container Interface Name: twc-0/0/12/11
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/13/1
    Container Interface Name: twc-0/0/12/13
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
Logical Interface Name: twc-0/0/12/14/1
    Container Interface Name: twc-0/0/12/14
    Outer Vlan ID: 10, Tagged: true
    IPv4:MTU: 1518, Status: up
    IPv6:MTU: 1518, Status: up
    MPLS:MTU: 9216, Status: up
    Admin Status: up, Oper status: up
```

2.17. Display the IPv4 Unicast Routing Table for a VRF Instance

Command to display the prefix, source, preference, and next-hop for IPv4 unicast routes for the subscriber instance

rtb ffws show ipv4 route unicast instance <instance-name>

<instance-name>	Name of the instance (for example, subscriber)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show ipv4 route unicast instance
subscriber
Routes for Instance: subscriber
AFI: IPV4 SAFI: Unicast
+--------------------+----------+----------
+-------------------------------------------------+
Prefix                  Source   Preference
Next-Hop
+--------------------+----------+----------
+-------------------------------------------------+
10.2.1.1/32             bgp        20       192:168:11::1 label-op push,
label 20001, bos 1
10.2.1.2/32             direct      0       via lo-0/0/0/0/2
10.2.1.5/32             bgp        20       192:168:11::5 label-op push,
label 20001, bos 1
10.2.1.6/32             bgp        20       192:168:11::6 label-op push,
label 20001, bos 1
10.2.1.0/24             static      1       via null0
10.11.1.0/24            bgp        20       8.11.1.2
20.11.1.0/24            direct      0       via twc-0/0/12/18/1
192.168.11.8/32         bgp        20       28.11.1.2
+--------------------+----------+----------
+-------------------------------------------------+
```

2.18. Display the IPv6 Unicast Routing Table for a VRF Instance

Command to display the prefix, source, preference, and next-hop for IPv6 unicast routes for the subscriber instance.

rtb ffws show ipv6 route unicast instance <instance-name>

<instance-name>	Name of the instance (for example, subscriber)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show ipv6 route unicast instance
subscriber

Routes for Instance: subscriber
AFI: IPV6 SAFI: Unicast
+--------------------------+------+----------
+-----------------------------------------------+
  Prefix                    Source Preference                  Next-Hop
+--------------------------+------+----------
+-----------------------------------------------+
 10:2:1::1/128                bgp      20       192:168:11::1 label-op
push, label 20002, bos 1
 10:2:1::2/128                direct    0       via lo-0/0/0/0/2
 10:2:1::5/128                bgp      20       192:168:11::5 label-op
push, label 20002, bos 1
 10:2:1::6/128                bgp      20       192:168:11::6 label-op
push, label 20002, bos 1
 192:168:11::8/128            bgp      20       10.11.1.2
 ::/0                         bgp      20       10.11.1.2
+--------------------------+------+----------
+-----------------------------------------------+
```

2.19. Display the IPv6 Labeled-Unicast Routing Table for the mgmt Instance

Command to display the prefix, source, preference, and next-hop for IPv6 labeled-unicast routes for the subscriber instance

rtb ffws show ipv6 route labeled-unicast instance <instance-name>

<instance-name>	Name of the instance (for example, mgmt)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show ipv6 route labeled-unicast
instance subscriber

Routes for Instance: subscriber
AFI: IPV6 SAFI: Labeled-unicast
+-----------------+-----------
+-------------------------------------------------------------+
        Prefix     Source Pref                                Next-Hop
+-----------------+------+----
+-------------------------------------------------------------+
 192:168:11::1/128 bgp    20   e80::2:f41e:5eff:fe11:102 via twc-
0/0/12/11/1 label-op push,...
 192:168:11::2/128 direct  0   via lo-0/0/0/0/1
 192:168:11::3/128 bgp    20   fe80::2:f41e:5eff:fe13:102 via twc-
0/0/12/13/ label-op push,...
 192:168:11::4/128 bgp    20   fe80::2:f41e:5eff:fe14:102 via twc-
0/0/12/14/ label-op push,...
 192:168:11::5/128 bgp    20   fe80::2:f41e:5eff:fe13:102 via twc-
0/0/12/13/ label-op push,...
 192:168:11::6/128 bgp    20   fe80::2:f41e:5eff:fe13:102 via twc-
0/0/12/13/ label-op push,...
+-----------------+------+----
+-------------------------------------------------------------+
```

2.20. Display the IPv4 Unicast Routing Table for the mgmt Instance

Command to display the prefix, source, preference, and next-hop for IPv4 unicast routes for the mgmt instance

rtb ffws show ipv4 route unicast instance <instance-name>

<instance-name>	Name of the instance (for example, mgmt)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show ipv4 route unicast instance mgmt

  Routes for Instance: mgmt
    AFI: IPV4 SAFI: Unicast
+--------------------+----------+----------
+-------------------------------------------------+
      Prefix             Source  Preference
Next-Hop
+--------------------+----------+----------
+-------------------------------------------------+
 10.2.1.1/32              bgp       20      192:168:11::1 label-op push,
label 20001, bos 1
 10.2.1.2/32            direct       0      via lo-0/0/0/0/2
 10.2.1.5/32              bgp       20      192:168:11::5 label-op push,
label 20001, bos 1
 10.2.1.6/32              bgp       20      192:168:11::6 label-op push,
label 20001, bos 1
 10.2.1.0/24            static       1      via null0
 10.11.1.0/24             bgp       20      28.11.1.2
 20.11.1.0/24           direct       0      via twc-0/0/12/18/1
 192.168.11.8/32          bgp       20      28.11.1.2
+--------------------+----------+----------
+-------------------------------------------------+
```

2.21. Display the IPv6 Unicast Routing Table for the mgmt Instance

Command to display the prefix, source, preference, and next-hop for IPv6 unicast routes for the mgmt instance

rtb ffws show ipv6 route unicast instance <instance-name>

<instance-name>	Name of the instance (for example, mgmt)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show ipv6 route unicast instance mgmt

    Routes for Instance: mgmt
      AFI: IPV6 SAFI: Unicast
+--------------------------+------+----------
+-----------------------------------------------+
          Prefix            Source Preference                  Next-Hop
+--------------------------+------+----------
+-----------------------------------------------+
 10:2:1::1/128                bgp      20     192:168:11::1 label-op
push, label 20002, bos 1
 10:2:1::2/128                direct    0     via lo-0/0/0/0/2
 10:2:1::5/128                bgp      20     192:168:11::5 label-op
push, label 20002, bos 1
 10:2:1::6/128                bgp      20     192:168:11::6 label-op
push, label 20002, bos 1
 192:168:11::8/128            bgp      20     10.11.1.2
 ::/0                         bgp      20     10.11.1.2
+--------------------------+------+----------
+-----------------------------------------------+
```

2.22. Display the MPLS Routing Table

Command to display the label, source, and next-hop for MPLS routes for the default instance

rtb ffws show mpls route

<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb fwdd show mpls route

    MPLS Routes for Instance: default
+----------------+------
+-------------------------------------------------------------------------
+
        Label     Source                                  Next-Hop
+----------------+------
+-------------------------------------------------------------------------
+
label 5111                fe80::2:f41e:5eff:fe11:102 via twc-0/0/12/11/1
label 5113                fe80::2:f41e:5eff:fe13:102 via twc-0/0/12/13/1
label 5114                fe80::2:f41e:5eff:fe14:102 via twc-0/0/12/14/1
label 5115                fe80::2:f41e:5eff:fe13:102 via twc-0/0/12/13/1
label-op swap, label 5115
label 5116                e80::2:f41e:5eff:fe13:102 via twc-0/0/12/13/1
label-op swap, label 5116
```

```
label 20001, bos 1        lookup subscriber,ipv4,unicast label-op pop-
ipv4
label 20002, bos 1        lookup subscriber,ipv6,unicast label-op pop-
ipv6
label 20003, bos 1        lookup mgmt,ipv4,unicast label-op pop-ipv4
label 20004, bos 1        lookup mgmt,ipv6,unicast label-op pop-ipv6
label 20005, bos 1        lookup radius,ipv4,unicast label-op pop-ipv4
label 20006, bos 1        lookup radius,ipv6,unicast label-op pop-ipv6
label 20007, bos 1        fe80::2:f41e:5eff:fe14:102 via twc-0/0/12/14/1
label-op swap,label 20001
label 20008, bos 1        fe80::2:f41e:5eff:fe14:102 via twc-0/0/12/14/1
label-op swap,label 20002
label 20009, bos 1        fe80::2:f41e:5eff:fe13:102 via twc-0/0/12/13/1
label-op swap,label 20001
label 20010, bos 1        fe80::2:f41e:5eff:fe13:102 via twc-0/0/12/13/1
label-op swap,label 20002
label 20011, bos 1        192:168:11::1 label-op swap, label 20001
label 20014, bos 1        192:168:11::1 label-op swap, label 20002
label 20017, bos 1        192:168:11::6 label-op swap, label 20001
label 20018, bos 1        192:168:11::5 label-op swap, label 20001
label 20019, bos 1        192:168:11::6 label-op swap, label 20002
label 20020, bos 1        192:168:11::5 label-op swap, label 20002
label 20021, bos 1        192:168:11::6 label-op swap, label 20004
label 20022, bos 1        192:168:11::5 label-op swap, label 20004
label 20023, bos 1        192:168:11::6 label-op swap, label 20003
label 20023, bos 1        192:168:11::6 label-op swap, label 20003
label 20024, bos 1        192:168:11::5 label-op swap, label 20003
label 20025, bos 1        192:168:11::6 label-op swap, label 20006
label 20026, bos 1        192:168:11::5 label-op swap, label 20006
label 20027, bos 1        192:168:11::1 label-op swap, label 20003
label 20030, bos 1        192:168:11::1 label-op swap, label 20004
label 20033, bos 1        192:168:11::6 label-op swap, label 20005
label 20034, bos 1        192:168:11::5 label-op swap, label 20005
label 20035, bos 1        192:168:11::1 label-op swap, label 20005
label 20038, bos 1        192:168:11::1 label-op swap, label 20006
+----------------+------
+-----------------------------------------------------------------------
+
```

2.23. Ping an IPv4 Address That is Part of a VRF Instance

Command to ping the IPv6 address used in the subscriber instance **rtb hald.1 ping** <ipv6-address> **instance** <instance-name>

<ipv6-address>	Address to ping
<instance-name>	Name of instance to act on (for example, subscriber)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb hald ping6 10:2:1::1 instance subscriber

Reply from 10:2:1::1 icmp_seq=1 ttl=63 time=3.13 ms
Reply from 10:2:1::1 icmp_seq=2 ttl=63 time=3.64 ms
Reply from 10:2:1::1 icmp_seq=3 ttl=63 time=0.35 ms
Reply from 10:2:1::1 icmp_seq=4 ttl=63 time=1.91 ms
Reply from 10:2:1::1 icmp_seq=5 ttl=63 time=1.41 ms
--- 10:2:1::1 ping statistics ---
```

2.24. Ping an IPv6 Address That is Part of a VRF Instance

Command to ping the IPv6 address used in the subscriber instance

rtb hald.1 ping <ipv6-address> **instance** <instance-name>

<ipv6-address>	Address to ping
<instance-name>	Name of instance to act on (for example, subscriber)
<CR>	Display information

Output of command directed to rtb on pod11_b_leaf2

```
ubuntu@pod11_b_leaf2:~$ rtb hald ping6 10:2:1::1 instance subscriber
Reply from 10:2:1::1 icmp_seq=1 ttl=63 time=3.13 ms
Reply from 10:2:1::1 icmp_seq=2 ttl=63 time=3.64 ms
Reply from 10:2:1::1 icmp_seq=3 ttl=63 time=0.35 ms
Reply from 10:2:1::1 icmp_seq=4 ttl=63 time=1.91 ms
Reply from 10:2:1::1 icmp_seq=5 ttl=63 time=1.41 ms
--- 10:2:1::1 ping statistics ---
```

3. Layer 2 Cross-Connect (L2X)

This document describes the forwarding daemon (FWDD) commands relating to Layer 2 cross-connect (L2X). L2X is a data plane feature that connects two aggregated ports (IFCs) using Layer 2 switching. At the simplest, L2X can switch all the traffic between two IFCs to provide the trunk service of an Ethernet switch. This use case and other L2X types are detailed in this section.

3.1. L2X Features

The overall L2X features are divided into several major categories:

3.1.1. Local Cross-connect (Local L2X)

In a local L2X, both IFCs are on the same router. The L2X configured on the router switches Layer 2 (frame) traffic between the ports. These cross connects are bi-directional and carry traffic in both directions.

3.1.2. Remote Cross-connect (Remote L2X)

In a remote L2X, the IFCs are located on two different routers. An MPLS tunnel transports the traffic between the two routers. The L2X configuration at the MPLS tunnel ingress (source) encapsulates the packet received on the input interface in an Ethernet frame as well as the MPLS labels (that is, the ingress performs label pushing). There are two labels added with label stacking for a remote L2X: an inner service label and an outer transport label.

This MPLS data unit is sent to the egress (output) interface through an MPLS-enabled Layer 3 core. As with any MPLS tunnel, there can be several routers between the ingress and egress.

The outer transport header label is removed (popped) at the penultimate (next-to-last) hop and the inner service label is popped at the egress node. This process is shown in Figure 1.

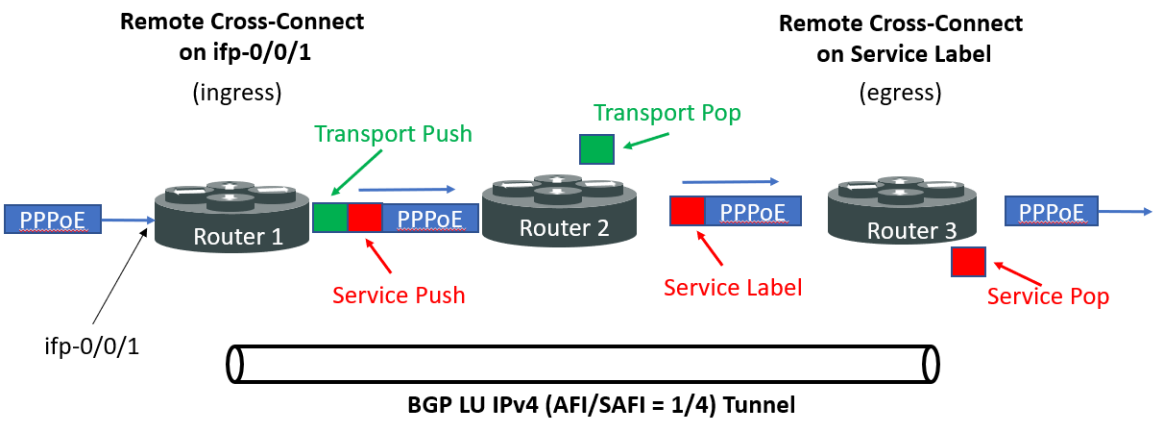

Figure 1. Remote Cross-Connect with MPLS tunnel

If there are only two routers, ingress and egress, then there is no transport label.

The L2X configuration at the MPLS tunnel egress processes the Ethernet frame and MPLS header (it performs a label pop). Then the destination sends the payload on an output interface.

Remote L2X tunnels, as all MPLS connections, are unidirectional and carry traffic only in one direction. Configure a pair of ingress and egress L2X end points on the routers to carry bi-directional traffic.

3.1.3. Port and VLAN Cross-connects

Both types of L2X switches Layer 2 traffic from input interface to output interface. The difference is that a port cross-connect switches all Layer 2 traffic arriving at an input interface, but a VLAN cross-connect only switches the Layer 2 traffic associated with a specific VLAN. A port-based L2X means a port-only configuration, so there are no VLANs involved. A VLAN-based L2X, of course has VLANs configured.

Both single-tag and double-tagged (inner and outer VLAN tags) are supported.

The port and VLAN L2X support both local and remote L2X configurations. In remote L2X connections, the VLAN cross-connects are typically configured on the MPLS tunnel ingress router.

Untagged traffic on L2X interfaces is also supported. However, there is no way to select only untagged traffic for cross-connecting. Therefore, only port cross-connects are supported for untagged traffic.

3.1.4. VLAN Editing

In a remote L2X, there is an option at the ingress router to delete the VLAN tag on arriving traffic before encapsulation.

Also, at the MPLS tunnel egress router, there is an option to add another VLAN tag to the Layer 2 payload.

It is also important to understand the way that a local L2X handles VLAN ID tags. If VLAN ID tags are configured for the arriving traffic, then the VLAN ID tags are removed at the ingress interface or added at the egress interface. If there are no VLAN ID tags configured to be removed at the ingress interface frames, them the frames are transported transparently (that is, the tags are left as there are).

Table 1 summarizes the VLAN operations and parameters required for local L2X. The table applies to both port-based and VLAN-based L2X configurations.

Table 1. L2X VLAN ID Handling

Configured Statements	Ingress Interface Action	Egress Interface Action
No VLAN ID tags configured	No operation on VLAN IDs	No operation on VLAN IDs
Only an outer VLAN ID (ifc, outer-vlan)	Remove outer VLAN ID	Add new outer VLAN ID
Both an outer and inner VLAN ID (ifc, outer-vlan, inner-vlan)	Remove both inner and outer VLAN ID	Add new inner and outer VLAN ID

A few examples help make the table clearer. All of these examples assume a local L2X with an ingress interface of ifc-0/0/0/5 and an egress interface of ifc-0/0/0/2.

3.1.5. VLAN Editing Examples

These examples cover the various possibilities for VLAN value editing in local L2X.

Example 1 (Port-based)

In this configuration example, any untagged or tagged frame is forwarded. If the frame is tagged, then the VLAN tag is forwarded transparently. Only the input and output interfaces need to be configured.

```
[forwarding-options l2x]
set input-interface ifc-0/0/0/1 outer-vlan 100
set output-interface ifc-0/0/0/2 outer-vlan 100
```

Example 2 (VLAN-based)

In this configuration example, frames tagged with the single VLAN ID tag value 100 are forwarded. The VLAN ID tag is removed at input and added at egress. Although the tag has the same value (100), it is a different tag. Note that any IEEE 802.1p bits in the arriving frame header are set to zero.

```
[forwarding-options l2x]
set input-interface ifc-0/0/0/1 outer-vlan 100 inner-vlan 7
set output-interface ifc-0/0/0/2 outer-vlan 200 inner-vlan 7
```

Example 3 (VLAN-based)

In this configuration example, the packet arrives at the ingress interface with an outer VLAN ID tag of 100 and an inner VLAN ID tag of 7. These tags are removed and new tags with the values for the outer VLAN ID tag set to 200 and the inner VLAN ID tag set to 7. Note that the value of the IEEE 802.1p bits on the arriving frame is lost with the removal of the VLAN ID tag:

```
[forwarding-options l2x]
set input-interface ifc-0/0/0/1 outer-vlan 100 inner-vlan 7
set output-interface ifc-0/0/0/2 outer-vlan 200 inner-vlan 7
```

3.1.6. IEEE 802.1p Bit Handling

Use of L2X does not change the IEEE 802.1p bits, which add traffic classes for quality of service (QoS) and dynamic multicast filtering to basic Ethernet.

However, when additional VLAN tags are added, those IEEE 802.1p bits are set to zero.

Also please note that the pushed remote L2X MPLS labels have the EXP bits set zero.

3.1.7. Multi-service Support

L2X is configured on Layer 2 (frame switching) ports and not Layer 3 (packet routing) ports. The same physical port (IFP) cannot be used for Layer 3 and Layer 2 logical interfaces (IFLs).

You can use PPP over Ethernet (PPPoE) and L2X on the same port.

3.2. Configuration Profile

The object model for the L2X configuration profile is shown in Figure 1.

```
ubuntu@omega_leaf_1:~$ rtb confd show datastore schema object table-name global.fwdd.l2x.config
File: /usr/share/rtbrick/libbds/objects/fwdd/config-l2x.json
Definition:

{
    "object": {
        "__comment__": "Layer2 cross connection configuration entry",
        "module" : "fwd",
        "attribute": [
            {
                "codepoint": 1,
                "type": "string",
                "name": "input_interface",
                "description": "The input interface which is to be cross connected"
            },
            {
                "codepoint": 2,
                "type": "uint16",
                "name": "input_inner_vlan",
                "description": "The inner vlan on the port to match"
            },
            {
                "codepoint": 4,
                "type": "uint16",
                "name": "input_outer_vlan",
                "description": "The vlan on the port to match"
            },
            {
                "codepoint": 5,
                "type": "ipv4addr",
                "name": "nexthop4",
                "description": "IPv4 nexthop address for the static route"
            },
            {
                "codepoint": 6,
                "type": "ipv6addr",
                "name": "nexthop6",
                "description": "Ipv6 nexthop address for the static route"
            },
```

```
            {
                "codepoint": 7,
                "type": "uint64",
                "name": "input_mpls_label",
                "description": "the MPLS label to match for egress side of L2X cross connection"
            },
            {
                "codepoint": 8,
                "type": "string",
                "name": "output_interface",
                "description": "The output interface to cross connect to"
            },
            {
                "codepoint": 9,
                "type": "uint16",
                "name": "output_inner_vlan",
                "description": "the inner vlan to strap the packet for L2X cross connection"
            },
            {
                "codepoint": 10,
                "type": "uint16",
                "name": "output_outer_vlan",
                "description": "The vlan to strap the packet for L2X cross connection"
            },
            {
                "codepoint": 11,
                "type": "string",
                "name": "l2x_name",
                "description": "L2x name"
            },
            {
                "codepoint": 12,
                "type": "string",
                "name": "description",
                "description": "L2x description"
            },
            {
                "codepoint": 13,
                "type": "uint32",
                "name": "service_label",
                "description": "L2x flag indicating direction of cross connect"
            },
            {
                "map"        : "vlan_operation_map",
                "codepoint"  : 14,
                "type"       : "uint8",
                "name"       : "vlan_operation",
                "description": "add or delete outer vlan at egress node output interface"
            },
            {
                "map"        : "filter_direction_map",
                "codepoint"  : 15,
                "type"       : "uint8",
                "name"       : "direction",
                "description": "L2x Direction"
            }
        ],
        "codepoint": 727,
        "name": "l2x_config_entry",
        "description": "Configuration object for L2X forwarding object"
    }
}
```


Figure 2. The L2X Configuration Profile

3.3. L2X Tables

The L2X configuration profile uses a global configuration table named global.fwdd.l2x.config. This table is also used for fwdd programming.

All of the tables listed in Table 1 can be used to verify correct L2X configuration and operation.

Table 2. L2X Tables to Verify Configuration and Operation

2X Table	L2X Table Content
global.fwdd.l2x.config	Contains the configuration for all L2X configuration profiles.

2X Table	L2X Table Content
default.fwd.fib-local.l2x.config	Contains the fib-local entry of L2X routes that need nexthop resolution. Because only the remote L2X on the ingress router has unresolved nexthops, only those routes are stored here.
default.fwd.nexthop-set.l2x.config	Contains the L2X nexthops that need resolution.
default.fwd.fib-local.l2x	Contains the fib-local L2X routes exported to the forwarding daemon (fwdd). L2X uses the same routing infrastructure as protocol routes.
default.fwd.nexthop-set.l2x	Contains the nexthop-set of L2X nexthops exported to the forwarding daemon (fwdd). L2X uses the same routing infrastructure as protocol routes.

3.4. L2X CLI Modeling

L2X has four modes of configuration under forwarding options. The basic information required for configuring one of the modes are as follows:

1. For creating port L2X on the MPLS tunnel ingress router, the l2x <input interface> configuration parameter is required
2. For creating a VLAN L2X on the MPLS tunnel ingress router, the l2x <input interface>` and outer vlan <vlan id> configuration parameters are required
3. For double-tagged packet VLAN L2X on the MPLS tunnel ingress router, the l2x <interface name>name>, outer vlan <vlan id>, and inner vlan <vlan id> configuration parameters are required
4. On the MPLS tunnel egress, router, the l2x label <input mpls label> configuration parameter is required

> The <l2x_name> parameter can follow the l2x keyword (l2x <l2x_name>), but the L2X name can only be set with HTTP, not with the CLI.

In addition to the basic information about interfaces, VLAN IDs, and MPLS labels, which determine the type of L2X, there are attributes and parameters that can be used for set or delete as listed in Table 2.

Table 3. L2X Configuration Profile Attributes

Statement	Description
set direction ["ingress", "egress"]	For remote L2X, the direction must be set to ingress on the MPLS tunnel ingress router and egress on the MPLS tunnel egress router.
set next-hop <nexthop> service-label <service_label> NOTE: Service label values outside the reserved BGP range (20000-100000) can be used: less than 20000 or greater than 100000.	The next-hop and service-label must be set on the MPLS tunnel ingress router. The <nexthop> is the IPv4 or IPv6 address of the MPLS tunnel egress router and <service_label> is the MPLS label for L2X used by the egress router. However, this statement is not used on the egress router.
set output-interface <interface-name> [outer-vlan <vlan-id>] [inner-vlan <vlan-id>]]	These values are set on the MPLS tunnel egress router. The <interface-name> is the IFC sending the Layer2 payload. The outer-vlan and inner-vlan are optional VLAN IDs. There must be an inner VLAN in order to configure an outer VLAN. If omitted, the VLAN IDs are ignored at the egress.
set vlan-operation ["Add-Outer-Vlan", "Delete-Outer-Vlan"]	To allow VLAN editing, set to "**Add-Outer-Vlan**" on the MPLS egress router and "**Delete-Outer-Vlan**" on the MPLS ingress router. An inner VLAN ID must also be configured.

3.5. Additional Configuration Examples

3.5.1. Port-based Local Cross-connect

This example creates a local cross-connect between aggregate interfaces ifc-0/0/1/1 and ifc-0/0/1/2.

```
rtb confd
edit forwarding-options
edit l2x input-interface ifc-0/0/1/1
set output-interface ifc-0/0/1/2
```

3.5.2. VLAN-Based Remote Cross-connect

This example creates a bidirectional, remote cross-connect between aggregated interface ifc-0/0/1/1, with outer VLAN ID 200 and inner VLAN ID 100, on a device named leaf and aggregated interface ifc-0/0/1/1 on a device named bleaf.

The example uses an MPLS tunnel between leaf and bleaf with IPv4 addresses 192.168.1.7 and 192.168.1.3 as logical unit (LU) addresses respectively.

The leaf device is configured with two L2X endpoints. First, an ingress end point for carrying traffic to bleaf is configured on leaf.

```
rtb confd
edit forwarding-options
edit l2x input-interface ifc-0/0/1/1 outer-vlan 200 inner-vlan 100
set direction ingress
set next-hop 192.168.1.3 service-label 2000
```

Second, an *egress* end point for carrying traffic from bleaf to leaf is configured on leaf.

```
rtb confd
edit forwarding-options
edit l2x label 3000
set direction egress
set output-interface ifc-0/0/1/1
```

Note the direction configurations and where the labels are set. Next, configure the matching two L2X endpoints on bleaf. First is the ingress interface.

```
rtb confd
edit forwarding-options
edit l2x input-interface ifc-0/0/1/1
set direction ingress
set next-hop 192.168.1.7 service-label 3000
```

Last is the egress end point on bleaf for traffic arriving from leaf.

```
rtb confd
edit forwarding-options
edit l2x label 2000
set direction egress
set output-interface ifc-0/0/1/1
```

Note how the service labels match up on the two devices.

3.5.3. VLAN-Based Cross-connect with VLAN ID Editing

This example shows how to add the VLAN ID editing feature to a bidirectional VLAN-based cross-connect such as configured in Example 2.

In this example, VLAN-ID editing adds a VLAN-ID tag value of 7 to traffic leaving bleaf (with MPLS label 2000) and deletes the outer VLAN-ID tag for traffic arriving at bleaf.

First, configure a bidirectional VLAN-based cross-connect as in Example 2.

Then, add these steps to the configuration on bleaf.

```
edit l2x label 2000
set output-interface ifc-0/0/1/1 outer-vlan 7
set vlan-operation Add-Outer-Vlan
exit
edit l2x input-interface ifc-0/0/1/1
set vlan-operation Delete-Outer-Vlan
```

3.6. Limitation and Future Work

Known limitations include the following:

- There is no support in L2X for VLAN ranges
- There is no support in L2X for VLAN lists (one L2X for each VLAN is required)
- There is no support in L2X for MPLS label lists at the egress router
- L2X compares both the inner and outer VLAN ID values of double-tagged traffic
- Traffic statistics are available if the l2x_name is set (the L2X name can only be set through HTTP, not the CLI)