Chapter 10

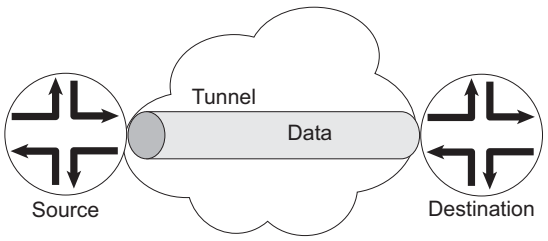
Configuring IP Tunnels

IP tunnels provide a way of transporting datagrams between routers separated by networks that do not support all the protocols that those routers support. This chapter describes how to configure IP tunnels on E-series routers; it contains the following sections:

Overview

E-series routers support static IP tunnels. An IP tunnel is a virtual point-to-point connection between two routers. See Figure 19. To establish an IP tunnel, you specify a tunnel type and name, and then configure an interface on each router to act as an endpoint for the tunnel.

Figure 19: IP Tunneling



E-series routers support the following types of IP tunnels:

- Generic Routing Encapsulation (GRE) tunnels
- Distance Vector Multicast Routing Protocol (DVMRP) tunnels, also known as IP-in-IP tunnels

GRE Tunnels

GRE encapsulates IP packets to enable data transmission through an IP tunnel. The resulting encapsulated packet contains a GRE header and a delivery header. Consequently, the packet requires more processing than an IP packet, and GRE can be slower than native routing protocols.

GRE tunnels can be secured with IPSec. See *Chapter 13, Securing L2TP and IP Tunnels with IPSec*.

DVMRP Tunnels

DVMRP tunnels allow the exchange of IP multicast traffic between routers separated by networks that do not support multicast routing. For information about DVMRP, see *JUNOSe Multicast Routing Configuration Guide, Chapter 1, Configuring IPv4 Multicast*.

DVMRP tunnels can be secured with IPSec. See *Chapter 13, Securing L2TP and IP Tunnels with IPSec*.

Platform Considerations

For information about modules that support IP tunnels on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support IP tunnels.

For information about modules that support IP tunnels on the E120 router and the E320 router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support IP tunnels.

Module Requirements

The supported modules for creating IP tunnels depends on the type of E-series router that you have.

ERX-7xx Models, ERX-14xx Models, and the ERX-310 Router

To create IP tunnels on an ERX-7xx model, ERX-14xx model, or an ERX-310 router, you must install a Service line module (SM) or a module that supports the use of shared tunnel-server ports. For information about installing modules in the ERX routers, see the *ERX Hardware Guide*.

SMs provide dedicated tunnel-server ports that are always configured on the module. Unlike other line modules, SMs do not pair with corresponding I/O modules that contain ingress and egress ports. Instead, they receive data from and transmit data to other line modules with access to ingress and egress ports on their own associated I/O modules. However, you must assign interfaces on other line modules or loopback interfaces to act as source endpoints for the tunnel.

You can also create IP tunnels on router modules that support shared tunnel-server ports. You can configure (provision) a shared tunnel-server port to use a portion of the module's bandwidth to provide tunnel services. For a list of the modules that support shared tunnel-server ports, see the *ERX Module Guide*.

For information about configuring tunnel services on dedicated and shared tunnel-server ports, see *JUNOSe Physical Layer Configuration Guide, Chapter 6, Managing Tunnel-Service and IPSec-Service Interfaces*.

All line modules forward traffic to IP tunnels. For information about which line modules accept traffic for IP tunnels, see the *ERX Module Guide*.

E120 Router and E320 Router

To create IP tunnels on an E120 router or an E320 router, you must install an ES2 4G line module (LM) with an ES2-S1 Service I/O adapter (IOA), or an IOA that supports the use of shared tunnel-server ports. For information about installing modules in these routers, see the *E120 and E320 Hardware Guide*.

The ES2 4G LM and ES2-S1 Service IOA combination provides a dedicated tunnel-server port that are always configured on the IOA. Unlike SMs, the ES2 4G LM requires the ES2-S1 Service IOA to condition it to receive and transmit data to other line modules. The ES2-S1 Service IOA also does not have ingress or egress ports.

You can also create IP tunnels on IOAs that support shared tunnel-server ports. You can configure (provision) a shared tunnel-server port to use a portion of the IOA's bandwidth to provide tunnel services. For a list of the IOAs that support shared tunnel-server ports, see the *E120 and E320 Module Guide*.

All line modules forward traffic to tunnels. For information about which IOAs accept traffic for tunnels, see the *E120 and E320 Module Guide*.

Redundancy and Tunnel Distribution

For information about the redundancy and tunnel distribution mechanisms supported for SMs, the ES2-S1 Service IOA, and shared tunnel-server ports, see *Tunnel-Service Interface Considerations* in *JUNOSe Physical Layer Configuration Guide, Chapter 6, Managing Tunnel-Service and IPSec-Service Interfaces*.

References

For more information about IP tunnels, see the following documents:

- RFC 791—Internet Protocol DARPA Internet Program Protocol Specification (September 1981)
- RFC 1700—Assigned Numbers (October 1994)
- RFC 1701—Generic Routing Encapsulation (October 1994)
- RFC 1702—Generic Routing Encapsulation over IPv4 Networks (October 1994)
- RFC 2003—IP Encapsulation within IP (October 1996)
- RFC 2784—Generic Routing Encapsulation (GRE) (March 2000)

Configuration Tasks

To configure an IP tunnel:

1. Create or select a physical or loopback interface.

   This interface acts as an anchor for the source of the tunnel.
2. Assign an IP address to the physical or loopback interface.
3. Create a tunnel interface.
4. Set the source address for the tunnel.
5. Set the destination address for the tunnel.
6. (Optional) Enable error checking across a GRE tunnel.
7. Set the maximum transmission unit (MTU) size for the tunnel.

NOTE: On SM interfaces, issue only the commands listed below. Do not configure protocols such as Multilink PPP or Multilink Frame Relay on SM interfaces.

interface tunnel

- Use to create an IP tunnel interface.
- Specify the type and name of the tunnel you want to create.
- You can use the **transport-virtual-router** keyword to establish the tunnel on a virtual router other than the current virtual router.
- Example

  host1(config)#**interface tunnel dvmrp:boston-tunnel-1 transport-virtual-router boston**

- Use the **no** version to remove the tunnel.

NOTE: When you delete a virtual router that has been configured as a transport virtual router for a DVMRP tunnel, the **show configuration** output displays No Router for the transport virtual router. To remove the DVMRP tunnel interface, simply omit any reference to the transport virtual router. For example, to delete **interface tunnel dvmrp:boston-tunnel-1 transport-virtual-router No Router** from the configuration, issue the command, **no interface tunnel dvmrp:boston-tunnel-1**.

tunnel checksum

- Use to enable checksum computation across a GRE tunnel.
- Checksum computation is not supported for DVMRP tunnels.
- Selecting this feature causes the E-series router to drop corrupted packets it receives on the tunnel interface.
- Example

  host1(config)#**interface tunnel gre:tunnel2**
  host1(config-if)#**tunnel checksum**

- Use the **no** version to disable the checksum option.

tunnel destination

- Use to configure the remote end of the tunnel.
- Specify either the IP address of an interface on the remote router or the hostname of the remote router.
  - The IP address is the address for the destination interface.
  - The hostname is the name of the destination interface.
- Example 1

  host1(config)#**interface tunnel dvmrp:tunnel2**
  host1(config-if)#**tunnel destination 192.13.7.1**

- Example 2

  host1(config)#**interface tunnel dvmrp:tunnel2**
  host1(config-if)#**tunnel destination remoteHost**

- Use the **no** version to remove the destination of a tunnel.

tunnel mtu

- Use to set the MTU for the tunnel.
- Specify a value in the range 1024–10240 bytes.
- Example

  host1(config-if)#**tunnel mtu 7500**

- Use the **no** version to restore the default, 10240 bytes.

tunnel source

- Use to configure the source of the tunnel.
- Specify either the primary IP address or the type and specifier of an interface.
- Do not specify an unnumbered interface.
- Example 1—Primary IP address

  host1(config)#**interface tunnel dvmrp:boston-tunnel-1**
  host1(config-if)#**tunnel source 192.10.2.1**

- Example 2—ATM interface on an ERX-7xx model, ERX-14xx model, or the ERX-310 router that uses the *slot/port* format

  host1(config)#**interface tunnel dvmrp:boston-tunnel-1**
  host1(config-if)#**tunnel source atm 5/0.12**

- Example 3—ATM interface on an E320 router that uses the *slot/adapter/port* format

  host1(config)#**interface tunnel dvmrp:boston-tunnel-1**
  host1(config-if)#**tunnel source atm 5/1/0.12**

- Use the **no** version to remove the source of a tunnel.

Configuration Example

In this example, two GRE tunnel interfaces are configured on different virtual routers of an E-series router. The source of the first tunnel interface matches the destination of the second tunnel interface and vice versa.

NOTE: This example contains an ATM interface configuration for an ERX-7xx model, ERX-14xx model, or ERX-310 router that uses the *slot/port* format.

1. Configure a virtual router called boston that supports one end of the tunnel.

   host1#**virtual-router boston**

2. Configure a physical or loopback interface for the end of the tunnel on virtual router boston.

   The IP address of this interface appears in the header of tunneled frames and is used for forwarding traffic.

   host1:boston#**interface atm 12/0.5**
   host1:boston(config-if)#**ip address 10.5.5.5 255.255.255.0**

3. Configure the tunnel interface on virtual router boston.

   a. Create the tunnel interface.

   host1:boston(config)#**interface tunnel gre:ChicagoTunnel**

   b. Configure the source and destination points of the tunnel interface.

   host1:boston(config-if)#**tunnel source 10.5.5.5**
   host1:boston(config-if)#**tunnel destination 10.6.6.6**

   c. Set the MTU for the tunnel.

   host1:boston(config-if)#**tunnel mtu 8000**

   d. Configure the IP address of the tunnel interface.

   host1:boston(config-if)**#ip address 10.7.7.7 255.255.255.0**

4. Configure a virtual router called chicago that supports the other end of the tunnel.

   host1(config)#**virtual-router chicago**

5. Configure a physical or loopback interface for the end of the tunnel on virtual router chicago.

   host1:chicago(config)#**interface atm 12/1.5**
   host1:chicago(config-if)#**ip address 10.6.6.6 255.255.255.0**

6. Configure the tunnel interface on virtual router chicago.

   a. Create the tunnel interface.

      The name of the tunnel interface can differ from the tunnel interface configured in Step 3.

      host1:chicago(config-if)#**interface tunnel gre:BostonTunnel**

   b. Configure the source and destination points of the tunnel interface.

      The destination of this tunnel interface matches the source of the tunnel interface configured in Step 3 and vice versa.

      host1:chicago(config-if)#**tunnel source 10.6.6.6**
      host1:chicago(config-if)#**tunnel destination 10.5.5.5**

c. Set the MTU for the tunnel.

The MTU must match the MTU configured in Step 3.

host1:chicago(config-if)#**tunnel mtu 8000**

d. Configure the IP address of the tunnel interface.

host1:chicago(config-if)#**ip address 10.9.9.9 255.255.255.0**

Configuring IP Tunnels to Forward IP Frames

When a line module receives IP frames destined for a tunnel, the module forwards the frames to a tunnel-service module. Tunnel-service modules include SMs and modules that support the use of shared tunnel-server ports.

The tunnel-service module encapsulates the frames and forwards them to the tunnel through an interface determined by a route lookup of an IP frame. The source and destination addresses in the IP frame are the source and destination addresses of the tunnel.

Similarly, when a line module receives traffic from a tunnel, the module forwards the traffic to the tunnel-service module for deencapsulation. After deencapsulation, the tunnel-service module forwards the resulting IP frames to an interface determined by a route lookup.

When you have configured a tunnel interface, treat it in the same way as any IP interface on the router. For example, you can configure static IP routes or enable routing protocols on the tunnel interface. The IP configurations you apply to the tunnels control how traffic travels through the network.

Preventing Recursive Tunnels

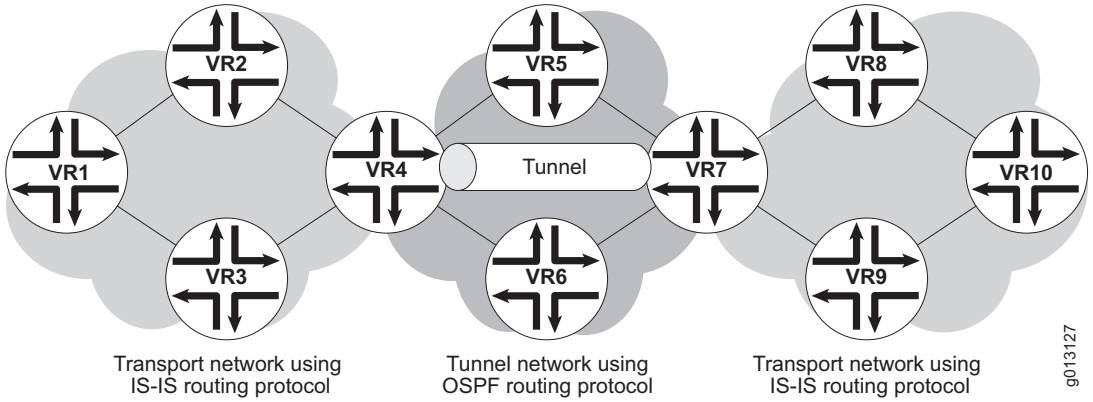
If routing information about the tunnel network combines with routing information about the transport networks (the networks that the tunnel services), a *recursive* tunnel can occur. In this case, the routing table defines the tunnel itself as the best path to a tunnel destination. To prevent recursive tunnels, differentiate routing information for the tunnel network and the transport networks with one or both of the following techniques:

- Use different routing protocols for the tunnel network and the transport networks.
- Define a static route to the tunnel destination.

NOTE: If you define a static route to a tunnel destination, be careful not to create routing loops.

Figure 20 illustrates how to prevent recursive tunnels by using different routing protocols for the tunnel network and the transport networks.

Figure 20: Transport and Tunnel Networks Using Different Routing Protocols



Creating Multicast VPNs Using GRE Tunnels

For information about configuring multicast VPNs using GRE tunnels, see *JUNOSe Multicast Routing Configuration Guide, Chapter 3, Configuring PIM for IPv4 Multicast*.

Monitoring IP Tunnels

You can monitor DVMRP and GRE tunnels by using the following commands.

show dvmrp tunnel

- Use to display information about DVMRP tunnels.
- To view detailed information about tunnels, specify the **detail** keyword.
- To view the number of tunnels in a specific state, specify the **state** keyword and the state of the tunnel (disabled, down, enabled, lower-down, not-present, up).
- To view the state of a specific tunnel, specify a tunnel name.
- To view the number of tunnels associated with that IP address, specify an IP address.
- To view the number of tunnels associated with an IP address on the virtual router, specify an IP address with the **virtual-router** keyword and the name of the virtual router.

- Field descriptions
  - Tunnel status
    - up—Tunnel is operational
    - down—Tunnel is not operational
    - not-present—Tunnel is not operational, because the hardware (such as a line module) supporting the tunnel is inaccessible
  - Tunnel name—Name of the tunnel
  - Tunnel mtu—Value of the maximum transmission unit for the tunnel
  - Tunnel source address—IP address of the source of the tunnel
  - Tunnel destination address—IP address of the destination of the tunnel
  - Tunnel transport virtual router—Name of the virtual router associated with the tunnel
  - Tunnel up/down trap—Indicates whether or not the E-series router sends traps to SNMP when the operational state of the tunnels changes, enabled or disabled
  - Tunnel server location—Location of the tunnel server in *slot/port* format (ERX-7xx models, ERX-14xx models, and the ERX-310 router) or *slot/adapter/port* format (E120 and E320 routers).
  - Tunnel secured by ipsec transport interface—IPSec interface that secures the tunnel.
  - Tunnel administrative state—Configured state of the tunnel: up or down
  - Statistics—Details of packets received or transmitted by the tunnel
  - Packets—Number of packets received or transmitted by the tunnel
  - Octets—Number of octets received or transmitted by the tunnel
  - Discards—Number of packets not accepted by the tunnel
  - Errors—Number of packets with errors received or transmitted by the tunnel
  - Data rx—Received data
  - Data tx—Transmitted data
  - Number of tunnels found—Total number of DVMRP tunnels found
  - Number of static tunnels—Number of tunnels created statically
- Example 1

```
host1#show dvmrp tunnel
DVMRP tunnel boston1 is up
1 DVMRP tunnel found
1 tunnel was created static
```

- Example 2

```
host1#show dvmrp tunnel detail
DVMRP tunnel boston1 is up (tunnel is static)
Tunnel operational configuration
  Tunnel name is 'boston1'
  Tunnel mtu is '10240'
  Tunnel source address is '0.0.0.0'
```

```
  Tunnel destination address is '0.0.0.0'
  Tunnel transport virtual router is vr1
  Tunnel up/down trap is enabled
  Tunnel server location is 4/0
  Tunnel secured by ipsec transport interface 1
  Tunnel administrative state is up
Statistics    packets         octets          discards      errors
  Data rx     0               0               0             0
  Data tx     0               0               0             0
1 DVMRP tunnel found
1 tunnel was created static
```

- Example 3

```
host1#show dvmrp tunnel state enabled
DVMRP tunnel boston1 is up
1 DVMRP tunnel found
1 tunnel was created static
```

- Example 4

```
host1#show dvmrp tunnel virtual-router vr1 ip 0.0.0.0
DVMRP tunnel boston1 is up
1 DVMRP tunnel found
1 tunnel was created static
```

- Example 5—Displays a DVMRP tunnel on an E320 router

```
host1#show dvmrp tunnel detail
DVMRP tunnel v1Tunnel1 is Up (tunnel is static)
Tunnel operational configuration
  Tunnel mtu is '10240'
  Tunnel source address is '50.1.1.1'
  Tunnel destination address is '50.1.1.2'
  Tunnel transport virtual router is v1
  Tunnel up/down trap is enabled
  Tunnel-server location is 13/0/0
  Tunnel administrative state is Up
Statistics    packets             octets              discards   errors
  Data rx     5                   740                 0          0
  Data tx     5                   740                 0          0
```

show dvmrp tunnel summary

- Use to display a summary of information about DVMRP tunnels.
- Field descriptions
  - Administrative status
    - enabled—Tunnel is available for use
    - disabled—Tunnel is not available for use
  - Operational status
    - up—Tunnel is operational
    - down—Tunnel is not operational
    - not-present—Tunnel is not operational, because the hardware (such as a line module) supporting the tunnel is inaccessible
- Example

```
host1#show dvmrp tunnel summary
Administrative status   enabled   disabled
                        1         0
Operational status      up        down     not-present
                        1         0        0
```

show gre tunnel

- Use to display information about a GRE tunnel or a list of GRE tunnels.
- To view detailed information about tunnels, specify the **detail** keyword.
- To view the number of tunnels in a specific state, specify the **state** keyword and the state of the tunnel (disabled, down, enabled, lower-down, not-present, up).
- To view the state of a specific tunnel, specify a tunnel name.
- To view the number of tunnels associated with an IP address, specify an IP address.
- To view the number of tunnels associated with an IP address on the virtual router, specify an IP address with the **virtual-router** keyword and the name of the virtual router.
- Field descriptions
  - Tunnel name—Name of the tunnel
  - Tunnel mtu—Value of the maximum transmission unit for the tunnel
  - Tunnel source address—IP address of the source of the tunnel
  - Tunnel destination address—IP address of the destination of the tunnel
  - Tunnel transport virtual router—Name of the virtual router associated with the tunnel
  - Tunnel mdt—State of the tunnel MDT
  - Tunnel checksum option—State of the checksum feature: enabled or disabled
  - Tunnel up/down trap—Indicates whether or not the E-series router sends traps to SNMP when the operational state of the tunnels changes, enabled or disabled

- Tunnel server location—Location of the tunnel server in *slot/port* format (ERX-7xx models, ERX-14xx models, and the ERX-310 router) or *slot/adapter/port* format (E120 and E320 routers).
- Tunnel is secured by ipsec transport interface—IPSec interface that secures the tunnel.
- Tunnel administrative state—Configured state of the tunnel: up or down
- Statistics—Details of packets received or transmitted by the tunnel
- Packets—Number of packets received or transmitted by the tunnel
- Octets—Number of octets received or transmitted by the tunnel
- Discards—Number of packets not accepted by the tunnel
- Errors—Number of packets with errors received or transmitted by the tunnel
- Data rx—Received data
- Data tx—Transmitted data
- Number of tunnels found—Total number of GRE tunnels found
- Number of static tunnels—Number of tunnels created statically

- Example 1

```
host1#show gre tunnel
3 GRE tunnels found
3 tunnels were created static
```

- Example 2

```
host1#show gre tunnel detail
Tunnel operational configuration
  Tunnel name is 'vr1
  Tunnel mtu is '10240'
  Tunnel source address is '10.0.0.2'
  Tunnel destination address is '10.0.0.1'
  Tunnel transport virtual router is vr1
  Tunnel mdt is disabled
  Tunnel checksum option is disabled
  Tunnel up/down trap is enabled
  Tunnel server location is 4/0
  Tunnel administrative state is up

Statistics    packets             octets              discards   errors
  Data rx     0                   0                   0          0
  Data tx     0                   0                   0          0
Tunnel operational configuration
  Tunnel name is 'default'
  Tunnel mtu is '10240'
  Tunnel source address is '10.0.0.1'
  Tunnel destination address is '10.0.0.2'
  Tunnel transport virtual router is default
  Tunnel checksum option is disabled
  Tunnel up/down trap is enabled
  Tunnel server location is 4/0
  Tunnel secured by ipsec transport interface 1
  Tunnel administrative state is up
```

```
Statistics    packets              octets               discards   errors
    Data rx     0                    0                    0          0
    Data tx     0                    0                    0          0
Tunnel operational configuration
  Tunnel name is 'london2'
  Tunnel mtu is '10240'
  Tunnel source address is '0.0.0.0'
  Tunnel destination address is '0.0.0.0'
  Tunnel transport virtual router is vr1
  Tunnel checksum option is disabled
  Tunnel up/down trap is enabled
  Tunnel server location is 4/0
  Tunnel administrative state is up

Statistics    packets              octets               discards   errors
  Data rx     0                    0                    0          0
  Data tx     0                    0                    0          0
3 GRE tunnels found
3 tunnels were created static
```

- Example 3

```
host1#show gre tunnel state up
GRE tunnel VR1 is up
GRE tunnel default is up
GRE tunnel london2 is down
3 GRE tunnels found
3 tunnels were created static
```

- Example 4

```
host1#show gre tunnel virtual-router vr1 ip 10.0.0.1
GRE tunnel VR1 is up
1 GRE tunnel found
1 tunnel was created static
```

- Example 5—Displays a GRE tunnel on an E320 router

```
host1#show gre tunnel detail
GRE tunnel start is Up (tunnel is static)
Tunnel operational configuration
  Tunnel mtu is '10240'
  Tunnel source address is '15.0.0.1'
  Tunnel destination address is '15.0.0.2'
  Tunnel transport virtual router is default
  Tunnel checksum option is disabled
  Tunnel sequence number option is disabled
  Tunnel up/down trap is enabled
  Tunnel-server location is 1/0/0
  Tunnel administrative state is Up
Statistics    packets              octets               discards   errors
  Data rx     0                    0                    0          0
  Data tx     0                    0                    0          0
GRE tunnel end is Up
Tunnel operational configuration
  Tunnel mtu is '10240'
  Tunnel source address is '15.0.0.2'
  Tunnel destination address is '15.0.0.1'
  Tunnel transport virtual router is vpnA
  Tunnel checksum option is disabled
  Tunnel sequence number option is disabled
  Tunnel up/down trap is enabled
  Tunnel-server location is 1/0/0
  Tunnel administrative state is Up
```

```
Statistics    packets             octets              discards   errors
  Data rx     0                   0                   0          0
  Data tx     0                   0                   0          0
2 GRE tunnels found
2 tunnels were created static
```

show gre tunnel summary

- Use to display a summary of information about GRE tunnels.
- Field descriptions
  - Administrative status
    - enabled—Tunnel is available for use
    - disabled—Tunnel is not available for use
  - Operational status
    - up—Tunnel is operational
    - down—Tunnel is not operational
    - not-present—Tunnel is not operational, because the hardware (such as a line module) supporting the tunnel is inaccessible
- Example

```
host1#show gre tunnel summary
Administrative status    enabled    disabled
                         3          0
Operational status       up         down       not-present
                         3          0          0
```