

Chapter 8

Miscellaneous MPLS Properties Configuration Guidelines

This chapter discusses the following topics:

- Configuring MPLS to Pop the Label on the Ultimate-Hop Router on page 170
- Configuring Traffic Engineering for LSPs on page 171
- Enabling Interarea Traffic Engineering on page 174
- Enabling Inter-AS Traffic Engineering for LSPs on page 174
- Configuring MPLS to Gather Statistics on page 178
- Controlling MPLS System Log Messages and SNMP Traps on page 179
- Configuring MPLS Firewall Filters and Policers on page 180
- Configuring MPLS Rewrite Rules on page 189
- Configuring BFD for MPLS LSPs on page 191
- Pinging LSPs on page 193
- Tracing MPLS and LSP Packets and Operations on page 194

Configuring MPLS to Pop the Label on the Ultimate-Hop Router

You can control the label value advertised on the egress router of a label-switched path (LSP). The default advertised label is label 3 (Implicit Null Label). If label 3 is advertised, the penultimate-hop router removes the label and sends the packet to the egress router. By enabling ultimate-hop popping, label 0 (IPv4 Explicit Null Label) is advertised. Ultimate-hop popping ensures that any packets traversing an MPLS network include a label.

To configure MPLS to pop the label on the ultimate-hop router, include the `explicit-null` statement:

```
explicit-null;
```

You can configure these statements at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]



NOTE: Juniper Networks routers queue packets based on the incoming label. Routers from other vendors might queue packets differently. Keep this in mind when working with networks containing routers from multiple vendors.

For more information about labels, see “Label Description” on page 26 and “Label Allocation” on page 28.

Configuring Traffic Engineering for LSPs

When you configure an LSP, a host route (a 32-bit mask) is installed in the ingress router toward the egress router; the address of the host route is the destination address of the LSP. By default, only BGP can use LSPs in its route calculations (**traffic-engineering bgp**). By configuring the other **traffic-engineering** statement options, you can alter this behavior in the master instance. This functionality is not available for specific routing instances. Also, you can enable only one of the **traffic-engineering** statement options (**bgp**, **bgp-igp**, **bgp-igp-both-ribs**, or **mpls-forwarding**) at a time.



NOTE: Enabling or disabling any of the **traffic-engineering** statement options causes all the MPLS routes to be removed and then reinserted into the routing tables.

You can configure traffic engineering for LSPs as follows:

- Using RSVP and LDP Routes for Traffic Forwarding on page 171
- Using RSVP and LDP Routes for Forwarding in VPNs on page 172
- Using RSVP and LDP Routes for Forwarding But Not Route Selection on page 172

You can also configure Open Shortest Path First (OSPF) and traffic engineering to advertise the LSP metric in summary link-state advertisements (LSAs), as described in the section “Advertising the LSP Metric in Summary LSAs” on page 173.

Using RSVP and LDP Routes for Traffic Forwarding

Configure the **bgp-igp** option of the **traffic-engineering** statement to cause BGP and the interior gateway protocols (IGPs) to use LSPs for forwarding traffic destined for egress routers. The **bgp-igp** option causes all **inet.3** routes to be moved to the **inet.0** routing table.

On the ingress router, include the **traffic-engineering bgp-igp** statement:

```
traffic-engineering bgp-igp;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]



NOTE: The **bgp-igp** option of the **traffic-engineering** statement cannot be configured for VPNs. VPN routing instances require that routes be in the **inet.3** routing table.

Using RSVP and LDP Routes for Forwarding in VPNs

VPNs rely on the routes in the `inet.3` routing table to function properly. For VPNs, configure the `bgp-igp-both-ribs` option of the `traffic-engineering` statement to cause BGP and the IGP to use LSPs for forwarding traffic destined for egress routers. The `bgp-igp-both-ribs` option installs the ingress routes in both the `inet.0` routing table (for IPv4 unicast routes) and the `inet.3` routing table (for MPLS path information).

On the ingress router, include the `traffic-engineering bgp-igp-both-ribs` statement:

```
traffic-engineering bgp-igp-both-ribs;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]

Using RSVP and LDP Routes for Forwarding But Not Route Selection

If you configure the `traffic-engineering bgp-igp` statement or the `traffic-engineering bgp-igp-both-ribs` statement, high-priority RSVP and LDP routes can supersede IGP routes in the `inet.0` routing table. IGP routes might no longer be redistributed since they are no longer the active routes.

When you configure the `mpls-forwarding` option at either the [edit logical-routers *logical-router-name* protocols mpls traffic-engineering] hierarchy level or the [edit protocols mpls traffic-engineering] hierarchy level, RSVP and LDP routes are used for forwarding but are excluded from route selection. These routes are added to both the `inet.0` and `inet.3` routing tables. The RSVP and LDP routes in the `inet.0` routing table are given a low preference when the active route is selected. However, the RSVP and LDP routes in the `inet.3` routing table are given a normal preference and are therefore used for selecting forwarding next hops.

When you activate the `mpls-forwarding` option, routes whose state is `ForwardingOnly` are preferred for forwarding even if their preference is lower than that of the currently active route. To examine the state of a route, execute a `show route detail` command.

To configure, include the `traffic-engineering mpls-forwarding` statement:

```
traffic-engineering mpls-forwarding;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]

When you configure the `mpls-forwarding` option, IGP shortcut routes are copied to the `inet.0` routing table only.

Advertising the LSP Metric in Summary LSAs

You can configure MPLS and OSPF to treat an LSP as a link. This configuration allows other routers in the network to use this LSP. To accomplish this goal, you need to configure MPLS and OSPF traffic engineering to advertise the LSP metric in summary LSAs.

For MPLS, include the `traffic-engineering bgp-igp` and `label-switched-path` statements:

```
traffic-engineering bgp-igp;
label-switched-path lsp-path-name {
    to address;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]

For OSPF, include the `lsp-metric-into-summary` statement:

```
lsp-metric-into-summary;
```

You can include this statement at the following hierarchy levels:

- [edit protocols ospf traffic-engineering shortcuts]
- [edit logical-routers *logical-router-name* protocols ospf traffic-engineering shortcuts]

For more information about MPLS traffic engineering, see “Configuring Traffic Engineering for LSPs” on page 171. For more information about OSPF traffic engineering, see the *JUNOS Routing Protocols Configuration Guide*.

Enabling Interarea Traffic Engineering

The JUNOS software can signal a contiguous traffic-engineered LSP across multiple OSPF areas. The LSP signaling must be done using either nesting or contiguous signaling, as described in RFC 4206, *Label-Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)*. However, contiguous signaling support is limited to just basic signaling. Reoptimization is not supported with contiguous signaling.

The following describes some of the interarea traffic engineering features:

- Interarea traffic engineering can be enabled when the loose-hop area border routers (ABRs) are configured on the ingress router using CSPF for the Explicit Route Object (ERO) calculation within an OSPF area. ERO expansion is completed on the ABRs.
- interarea traffic engineering can be enabled when CSPF is enabled, but without ABRs specified in the LSP configuration on the ingress router (ABRs can be automatically designated).
- Differentiated Services (DiffServ) traffic engineering is supported as long as the class type mappings are uniform across multiple areas.

To enable interarea traffic engineering, include the `expand-loose-hop` statement in the configurations for each of LSP transit router:

```
expand-loose-hop;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols mpls]`
- `[edit logical-routers logical-router-name protocols mpls]`

Enabling Inter-AS Traffic Engineering for LSPs

Generally, traffic engineering is possible for LSPs that meet the following conditions:

- Both ends of the LSP are in the same OSPF area or at the same IS-IS level.
- The two ends of the LSP are in different OSPF areas within the same autonomous system (AS). LSPs that end in different IS-IS levels are not supported.
- The two ends of an explicit-path LSP are in different OSPF ASs and the autonomous system border routers (ASBRs) are configured statically as the loose hops supported on the explicit-path LSP. For more information about explicit-path LSPs, see “Static and Explicit-Path LSP Configuration Guidelines” on page 159.

Without statically defined ASBRs on LSPs, traffic engineering is not possible between one routing domain, or AS, and another. However, when the ASs are under the control of single service provider, it is possible in some cases to have traffic engineered LSPs span the ASs and dynamically discover the OSPF ASBRs linking them (IS-IS is not supported with this feature).

Inter-AS traffic engineered LSPs are possible as long as certain network requirements are met, none of the limiting conditions apply, and OSPF passive mode is configured with EBGP. Details are provided in the following sections:

- Inter-AS Traffic Engineering Requirements on page 175
- Inter-AS Traffic Engineering Limitations on page 176
- Configuring OSPF Passive TE Mode on page 177

Inter-AS Traffic Engineering Requirements

The proper establishment and functioning of inter-AS traffic engineered LSPs depend on the following network requirements, all of which must be met:

- All ASs are under control of a single service provider.
- OSPF is used as the routing protocol with each AS, and EBGP is used as the routing protocol between the ASs.
- ASBR information is available inside each AS.
- EBGP routing information is distributed by OSPF, and an IBGP full mesh is in place within each AS.
- Transit LSPs are *not* configured on the inter-AS links, but *are* configured between entry and exit point ASBRs on each AS.
- The EBGP link between ASBRs in different ASs is a direct link and must be configured as a passive traffic engineering link under OSPF. The remote link address itself, not the loopback or any other link address, is used as the remote node identifier for this passive link. For more information about OSPF passive traffic engineering mode configuration, see “Configuring OSPF Passive TE Mode” on page 177.

In addition, the address used for the remote node of the OSPF passive traffic engineering link must be the same as the address used for the EBGP link. For more information about OSPF and BGP in general, see the *JUNOS Routing Protocols Configuration Guide*.

Inter-AS Traffic Engineering Limitations

Only LSP hierarchical, or nested, signaling is supported for inter-AS traffic engineered LSPs. Only point-to-point LSPs are supported (there is no point-to-multipoint support).

In addition, the following limitations apply. Any one of these conditions is sufficient to render inter-AS traffic engineered LSPs impossible, even if the above requirements are met.

- The use of multihop BGP is not supported.
- The use of policers or topologies that prevent BGP routes from being known inside the AS is not supported.
- Multiple ASBRs on a LAN between EBGPs are not supported. Only one ASBR on a LAN between EBGPs is supported (others ASBRs can exist on the LAN, but cannot be advertised).
- Route reflectors or policies that hide ASBR information or prevent ASBR information from being distributed inside the ASs are not supported.
- Bidirectional LSPs are not supported (LSPs are unidirectional from the traffic engineering perspective).
- Topologies with both inter-AS and intra-AS paths to the same destination are not supported.

In addition, several features that are routine with all LSPs are not supported with inter-AS traffic engineering:

- Admin group link colors are not supported.
- Secondary standby is not supported.
- Reoptimization is not supported.
- Crankback on transit routers is not supported.
- Diverse path calculation is not supported.
- Graceful restart is not supported.

These lists of limitations or unsupported features with inter-AS traffic engineered LSPs are not exhaustive.

Configuring OSPF Passive TE Mode

Ordinarily, interior routing protocols such as OSPF are not run on links between ASs. However, for inter-AS traffic engineering to function properly, information about the inter-AS link, in particular, the address on the remote interface, must be made available inside the AS. This information is not normally included either in EBGp reachability messages or in OSPF routing advertisements.

To flood this link address information within the AS and make it available for traffic engineering calculations, you must configure OSPF passive mode for traffic engineering on each inter-AS interface. You must also supply the remote address for OSPF to distribute and include in the traffic engineering database (TED).

To configure OSPF passive mode for traffic engineering on an inter-AS interface, include the `passive` statement for the link at the `[edit protocols ospf area area-id interface interface-name]` hierarchy level:

```
passive {
  traffic-engineering {
    remote-node-id ip-address;    /* IP address at far end of inter-AS link */
  }
}
```

OSPF must be properly configured on the router. The following example configures the inter-AS link `so-1/1/0` to distribute traffic engineering information with OSPF within the AS. The local IP address on the link is `192.168.207.1` and the remote address is `192.168.207.2`.

```
[edit protocols ospf area 0.0.0.0]
interface so-1/1/0 {
  unit 0 {
    passive {
      traffic-engineering {
        remote-node-id 192.168.207.2;
      }
    }
  }
  family inet {
    address 192.168.207.1;
  }
}
```

Configuring MPLS to Gather Statistics

You can configure MPLS so that it periodically gathers traffic statistics about all MPLS sessions, including transit sessions by configuring the **statistics** statement. You must configure the **statistics** statement if you want to collect MPLS traffic statistics using Simple Network Management Protocol (SNMP) polling of MPLS Management Information Bases (MIBs).

To enable MPLS statistics collection, include the **statistics** statement:

```
statistics {
  auto-bandwidth;
  file filename <size size> <files number> <no-stamp> <replace>
    (world-readable | <no-world-readable>);
  interval seconds;
}
```

You can configure these statements at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]

The default interval is 300 seconds.

The statistics are placed in a file, with one entry per LSP. During the specified interval, the following information is recorded in this file:

- The number of packets, number of bytes, packets per second, and bytes per second transmitted by each LSP.
- The percent of bandwidth transmitted over a given LSP in relation to the bandwidth percentage configured for that LSP. If no bandwidth is configured for an LSP, 0 percent is recorded in the percentage column.



NOTE: If you configure the **replace** option, an existing trace file is replaced if there is one. However, the change does not take effect after you commit the configuration. The change takes effect only after the routing protocol process restarts.

At the end of each periodic report, a summary shows the current time, total number of sessions, number of sessions read, number of sessions ignored, and read errors, if any. Ignored sessions are typically those not in the up state or those with a reserved (0 through 15) incoming label (typically the egress point of an LSP). The reason for a read error appears on the same line as the entry for the LSP on which the error occurred. Gathering statistics is an unreliable process; occasional read errors might affect their accuracy. The following is a sample of the information included in the output file:

```
1sp6                0 pkt                0 Byte          0 pps          0 Bps          0
1sp5                0 pkt                0 Byte          0 pps          0 Bps          0
1sp6.1             34845 pkt           2926980 Byte    1049 pps       88179 Bps     132
1sp5.1             0 pkt                0 Byte          0 pps          0 Bps          0
1sp4                0 pkt                0 Byte          0 pps          0 Bps          0
Dec 7 17:28:38 Total 6 sessions: 5 success, 0 fail, 1 ignored
```

Controlling MPLS System Log Messages and SNMP Traps

Whenever an LSP makes a transition from up to down, or down to up, and whenever an LSP switches from one active path to another, the ingress router generates a system log message and sends an SNMP trap. The following shows a sample system log message:

```
MPLS lsp sheep1 up on primary(any) Route 192.168.1.1 192.168.1.2 192.168.1.3
MPLS lsp sheep1 change on primary(any) Route 192.168.1.1 192.168.1.2
192.168.1.3
MPLS lsp sheep1 down on primary(any)
MPLS lsp sheep1 up on secondary(any) Route 192.168.1.1 192.168.1.2 192.168.1.3
MPLS lsp sheep1 change on secondary(any) to primary(any), Route 192.168.1.1
192.168.1.2 192.168.1.3
```

For information about the MPLS SNMP traps and the proprietary MPLS MIBs, see the *JUNOS Network Management Configuration Guide*.

To generate system log messages for LSPs, include the `syslog` option to the `log-updown` statement:

```
log-updown {
    syslog;
}
```

To generate SNMP traps for LSPs, include the `trap` option to the `log-updown` statement:

```
log-updown {
    trap;
}
```

To generate SNMP traps whenever an LSP path goes down, include the `trap-path-down` option to the `log-updown` statement:

```
log-updown {
    trap-path-down;
}
```

To generate SNMP traps whenever an LSP path comes up, include the `trap-path-up` option to the `log-updown` statement:

```
log-updown {
    trap-path-up;
}
```

To disable the generation of system log messages, include the `no-syslog` option to the `log-updown` statement:

```
log-updown {
    no-syslog;
}
```

To disable the generation of SNMP traps, include the `no-trap` option to the `log-updown` statement:

```
log-updown {  
    no-trap;  
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]

For scalability reasons, only the ingress router generates SNMP traps. By default, MPLS issues traps for all configured LSPs. If you have many LSPs, the number of traps can become quite large. To disable the generation of SNMP traps, configure the `no-trap` option.

Configuring MPLS Firewall Filters and Policers

You can configure an MPLS firewall filter to count packets based on the EXP bits for the top-level MPLS label in a packet. You can also configure policers for MPLS LSPs.

The following sections discuss MPLS firewall filters and policers:

- Configuring MPLS Firewall Filters on page 181
- Examples: Configuring MPLS Firewall Filters on page 182
- Configuring Policers for LSPs on page 183
- Example: Configuring an LSP Policer on page 185
- Configuring Automatic Policers on page 186
- Writing Different DSCP and EXP Values in MPLS-Tagged IP Packets on page 189

Configuring MPLS Firewall Filters

You can configure an MPLS firewall filter to count packets based on the experimental (EXP) bits for the top-level MPLS label in a packet. You can then apply this filter to a specific interface. You can also configure a policer for the MPLS filter to police (that is, rate-limit) the traffic on the interface to which the filter is attached. You cannot apply MPLS firewall filters to Ethernet (fxp0) or loopback (lo0) interfaces.

You can configure an MPLS firewall filter on the M-series and the T-series platforms.

You can configure the following match criteria attributes for MPLS filters at the [edit firewall family mpls filter *filter-name* term *term-name* from] hierarchy level:

- exp
- exp-except

These attributes can accept EXP bits in the range 0 through 7. You can configure:

- A single EXP bit—for example, `exp 3`;
- Several EXP bits—for example, `exp 0, 4`;
- A range of EXP bits—for example, `exp [0-5]`;

If you do not specify a match criterion (that is, you do not configure the **from** statement and use only the **then** statement with the **count** action keyword), all the MPLS packets passing through the interface on which the filter is applied will be counted.

You also can configure any of the following action keywords at the [edit firewall family mpls filter *filter-name* term *term-name* then] hierarchy level:

- count
- accept
- discard
- next
- policer

For more information about how to configure firewall filters, see the *JUNOS Policy Framework Configuration Guide*. For more information about how to configure interfaces, see the *JUNOS Network Interfaces Configuration Guide* and the *JUNOS Services Interfaces Configuration Guide*.

Examples: Configuring MPLS Firewall Filters

The following examples illustrate how you might configure an MPLS firewall filter and then apply the filter to an interface. This filter is configured to count MPLS packets with EXP bits set to either 0 or 4.

The following shows a configuration for an MPLS firewall filter:

```
[edit firewall]
family mpls {
  filter expf {
    term expt0 {
      from {
        exp 0,4;
      }
      then {
        count counter0;
        accept;
      }
    }
  }
}
```

The following shows how to apply the MPLS firewall filter to an interface:

```
[edit interfaces]
so-0/0/0 {
  mtu 4474;
  encapsulation ppp;
  sonet-options {
    fcs 32;
  }
  unit 0 {
    point-to-point;
    family mpls {
      filter {
        input expf;
        output expf;
      }
    }
  }
}
```

The MPLS firewall filter is applied to the input and output of an interface (see the **input** and **output** statements in the preceding example).

Configuring Policers for LSPs

MPLS LSP policing allows you to control the amount of traffic forwarded through a particular LSP. Policing helps to ensure that the amount of traffic forwarded through an LSP never exceeds the requested bandwidth allocation. LSP policing is supported on regular LSPs, LSPs configured with DiffServ-aware traffic engineering, and multiclass LSPs. You can configure multiple policers for each multiclass LSP. For regular LSPs, each LSP policer is applied to all of the traffic traversing the LSP. The policer's bandwidth limitations become effective as soon as the total sum of traffic traversing the LSP exceeds the configured limit.

You configure the multiclass LSP and DiffServ-aware traffic engineering LSP policers in a filter. The filter can be configured to distinguish between the different class types and apply the relevant policer to each class type. The policers distinguish between class types based on the EXP bits.

You configure LSP policers under the **family any** filter. The **family any** filter is used because the policer is applied to traffic entering the LSP. This traffic might be from different families: IPv6, MPLS, and so on. You do not need to know what sort of traffic is entering the LSP, as long as the match conditions apply to all types of traffic.

You can configure only those match conditions that apply across all types of traffic. The following are the supported match conditions for LSP policers:

- forwarding-class
- packet-length
- interface
- interface-set

To enable a policer on an LSP, first you need to configure a policing filter and then include it in the LSP configuration. For information on how to configure policers, see the *JUNOS Policy Framework Configuration Guide*.

To configure a policer for an LSP, specify a filter by including the `filter` option to the `policing` statement:

```
policing {
    filter filter-name;
}
```

You can include the `policing` statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-path-name*]
- [edit logical-routers *logical-router-name* protocols mpls label-switched-path *lsp-path-name*]

LSP Policer Limitations

When configuring MPLS LSP policers, be aware of the following limitations:

- LSP policers are supported for packet LSPs only.
- LSP policers are supported for unicast next hops only. Multicast next hops are not supported.
- LSP policers are not supported on aggregated interfaces.
- The LSP policer runs before any output filters.
- Traffic sourced from the Routing Engine (for example, ping traffic) does not take the same forwarding path as transit traffic. This type of traffic cannot be policed.
- LSP policers work on all T-series routing platforms and on M-series routers that have the Internet Processor II application-specific integrated circuit (ASIC).

Example: Configuring an LSP Policer

The following example shows how you can configure a policing filter for an LSP:

```
[edit firewall]
policer police-ct1 {
  if-exceeding {
    bandwidth-limit 50m;
    burst-size-limit 1500;
  }
  then {
    discard;
  }
}
policer police-ct0 {
  if-exceeding {
    bandwidth-limit 200m;
    burst-size-limit 1500;
  }
  then {
    discard;
  }
}
family any {
  filter bar {
    term discard-ct0 {
      then {
        policer police-ct0;
        accept;
      }
    }
  }
  term discard-ct1 {
    then {
      policer police-ct1;
      accept;
    }
  }
}
```

Configuring Automatic Policers

Automatic policing of LSPs allows you to provide strict service guarantees for network traffic. Such guarantees are especially useful in the context of Differentiated Services for traffic engineered LSPs, providing better emulation for ATM wires over an MPLS network. For more information about Differentiated Services for LSPs, see “DiffServ-Aware Traffic Engineering Configuration Guidelines” on page 135.

Differentiated Services for traffic engineered LSPs allow you to provide differential treatment to MPLS traffic based on the EXP bits. To ensure these traffic guarantees, it is insufficient to simply mark the traffic appropriately. If traffic follows a congested path, the requirements might not be met.

LSPs are guaranteed to be established along paths where enough resources are available to meet the requirements. However, even if the LSPs are established along such paths and are marked properly, these requirements cannot be guaranteed unless you ensure that no more traffic is sent to an LSP than there is bandwidth available.

It is possible to police LSP traffic by manually configuring an appropriate filter and applying it to the LSP in the configuration. However, for large deployments it is cumbersome to configure thousands of different filters. Configuration groups cannot solve this problem either, since different LSPs might have different bandwidth requirements, requiring different filters. To police traffic for numerous LSPs, it is best to configure automatic policers.

When you configure automatic policers for LSPs, a policer is applied to all of the LSPs configured on the router. However, you can disable automatic policing on specific LSPs.



NOTE: You cannot configure automatic policing for LSPs carrying CCC traffic.

The following sections describe how to configure automatic policers for LSPs:

- Configuring Automatic Policers for LSPs on page 187
- Configuring Automatic Policers for DiffServ-Traffic Engineering LSPs on page 188
- Disabling Automatic Policing on an LSP on page 188
- Example: Configuring Automatic Policers for LSPs on page 189

Configuring Automatic Policers for LSPs

To configure automatic policers for standard LSPs (neither DiffServ-aware traffic engineered LSPs nor multiclass LSPs), include the **auto-policing** statement with either the class all *policer-action* option or the class ct0 *policer-action* option:

```
auto-policing {
    class all policer-action;
    class ct0 policer-action;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]

You can configure the following policer actions for automatic policers:

- **drop**—Drop all packets.
- **loss-priority-high**—Set the packet loss priority (PLP) to high.
- **loss-priority-low**—Set the PLP to low.

These policer actions are applicable to all types of LSPs. The default policer action is to do nothing.



NOTE: You cannot configure automatic policers for LSPs that traverse aggregated interfaces or Multilink Point-to-Point Protocol (MLPPP) interfaces.

Configuring Automatic Policers for DiffServ-Traffic Engineering LSPs

To configure automatic policers for DiffServ-traffic engineering LSPs and for multiclass LSPs, include the `auto-policing` statement with either the `class all` *policer-action* option or the `class ctnumber` *policer-action* option. You can configure a different policer action for each class type.

To configure automatic policers for DiffServ-aware LSPs, include the `auto-policing` statement:

```
auto-policing {
  class all policer-action;
  class ctnumber policer-action;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls]
- [edit logical-routers *logical-router-name* protocols mpls]

For a list of the actions available for automatic policers, see “Configuring Automatic Policers for LSPs” on page 187. The default policer action is to do nothing.



NOTE: You cannot configure automatic policers for LSPs that traverse aggregated interfaces or MLPPP interfaces.

Disabling Automatic Policing on an LSP

When you enable automatic policing, all of the LSPs on the router or logical router are affected. To disable automatic policing on a specific LSP on a router where you have enabled automatic policing, include the `policing` statement with the `no-automatic-policing` option:

```
policing no-automatic-policing;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mpls label-switched-path *lsp-path-name*]
- [edit logical-routers *logical-router-name* protocols mpls label-switched-path *lsp-path-name*]

Example: Configuring Automatic Policers for LSPs

This example shows how you could configure automatic policing for LSPs. It shows automatic policing configured for a multiclass LSP with class types `ct0`, `ct1`, `ct2`, and `ct3` configured.

```
[edit protocols mpls]
diffserv-te {
    bandwidth-model extended-mam;
}
auto-policing {
    class ct1 loss-priority-low;
    class ct0 loss-priority-high;
    class ct2 drop;
    class ct3 loss-priority-low;
}
traffic-engineering bgp-igp;
label-switched-path sample-lsp {
    to 3.3.3.3;
    bandwidth {
        ct0 11;
        ct1 1;
        ct2 1;
        ct3 1;
    }
}
interface fxp0.0 {
    disable;
}
interface t1-0/5/3.0;
interface t1-0/5/4.0;
```

Writing Different DSCP and EXP Values in MPLS-Tagged IP Packets

You can selectively set the DiffServ code point (DSCP) field of MPLS-tagged IPv4 and IPv6 packets to 0 without affecting output queue assignment, and continue to set the MPLS EXP field according to the configured rewrite table, which is based on forwarding classes. You can accomplish this by configuring a firewall filter for the MPLS-tagged packets.

For instructions on how to write different DSCP and EXP values in MPLS-tagged IP packets, see the *JUNOS Class of Service Configuration Guide*. For instructions on how to configure firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

Configuring MPLS Rewrite Rules

You can apply a number of different rewrite rules to MPLS packets.

The following sections describe how you can apply rewrite rules to MPLS packets:

- Rewriting the EXP Bits of All Three Labels of an Outgoing Packet on page 190
- Rewriting MPLS and IPv4 Packet Headers on page 190

For more information about how to configure statements at the [edit class-of-service] hierarchy level, see the *JUNOS Class of Service Configuration Guide*.

Rewriting the EXP Bits of All Three Labels of an Outgoing Packet

In interprovider, carrier-of-carrier, and complex traffic engineering scenarios, it is sometimes necessary to push three labels on the next hop, using a swap-push-push or triple-push operation.

By default, on M-series routing platforms except the M320, the top MPLS EXP label of an outgoing packet is not rewritten when you configure swap-push-push and triple-push operations. You can rewrite the EXP bits of all three labels of an outgoing packet, thereby maintaining the class of service (CoS) of an incoming MPLS or non-MPLS packet.

To push three labels on incoming MPLS packets, include the `exp-swap-push-push default` statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules]
exp-swap-push-push default;
```

To push three labels on incoming non-MPLS packets, include the `exp-push-push-push default` statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
rewrite-rules]
exp-push-push-push default;
```

For more information about how to configure statements at the [edit class-of-service] hierarchy level, see the *JUNOS Class of Service Configuration Guide*.

Rewriting MPLS and IPv4 Packet Headers

You can apply a rewrite rule to MPLS and IPv4 packet headers simultaneously. This allows you to initialize MPLS EXP and IP precedence bits at LSP ingress. You can configure different rewrite rules depending on whether the traffic is VPN or non-VPN.

To rewrite MPLS and IPv4 packet headers, include the `protocol` statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* rewrite-rules *rewrite-rule-name*] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules
rewrite-rule-name]
protocol types;
```

Use the `protocol` statement to specify the types of MPLS packets and packet headers to which to apply the rewrite rule. The MPLS packet can be a standard MPLS packet or an MPLS packet with an IPv4 payload. Specify the type of MPLS packet by using the following options:

- `mpls-any`—Applies the rewrite rule to MPLS packets and writes the code point value to MPLS headers.
- `mpls-inet-both`—Applies the rewrite rule to VPN MPLS packets with IPv4 payloads. Writes the code point value to the MPLS and IPv4 headers in T-series and M320 routers. On M-series routing platforms, except the M320, the `mpls-inet-both` option causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with 000 code points for IP precedence and MPLS EXP values.
- `mpls-inet-both-non-vpn`—Applies the rewrite rule to any non-VPN MPLS packets with IPv4 payloads. Writes the code point value to the MPLS and IPv4 headers in T-series and M320 routers. On M-series routing platforms, except the M320, the `mpls-inet-both-non-vpn` option causes all ingress MPLS LSP packets with IPv4 payloads to be initialized with 000 code points for IP precedence and MPLS EXP values.

For a detailed example on how to configure rewrite rules for MPLS and IPv4 packets and for more information about how to configure class of service, see the *JUNOS Class of Service Configuration Guide*.

Configuring BFD for MPLS LSPs

You can configure Bidirectional Forwarding Detection (BFD) on MPLS IPv4 LSPs as outlined in the Internet draft `draft-ietf-bfd-mpls-02.txt`, *BFD for MPLS LSPs*. You can configure BFD for LSPs using either LDP or RSVP as the signaling protocol. BFD is used as a periodic Operation, Administration, and Maintenance (OAM) feature for LSPs to detect LSP data plane faults.

You can also use the LSP ping commands to detect LSP data plane faults. However, there are a couple of benefits to using BFD. It requires less computer processing than LSP ping. It can also quickly detect faults in large numbers of LSPs (LSP ping commands must be issued manually for each LSP).

Although you can use BFD to detect LSP data plane faults, it does not have the equivalent functionality of LSP ping. Specifically, you cannot use it to check the control plane against the data plane. An LSP ping echo request can be associated with a forwarding equivalence class (FEC).

An LSP ping echo request can be associated with an FEC. This makes it possible to verify the control plane against the data plane at the egress LSR.

To configure BFD for MPLS LSPs, complete the procedures in the following sections:

- Configuring BFD for RSVP LSPs on page 192
- Configuring BFD for LDP LSPs on page 375

Configuring BFD for RSVP LSPs

BFD for RSVP supports unicast IPv4 LSPs. When BFD is configured for an RSVP LSP on the ingress router, it is enabled on the primary path and on all standby secondary paths for that LSP. You can enable BFD for all LSPs on a router or for specific LSPs. If you configure BFD for a specific LSP, whatever values configured globally for BFD are overridden. The BFD sessions originate only at the ingress router and terminate at the egress router.

An error is logged whenever a BFD session for a path fails. The following shows how BFD for RSVP LSP log messages might appear:

```
RPD_MPLS_PATH_BFD_UP: MPLS BFD session for path path1 up on LSP R0_to_R3
RPD_MPLS_PATH_BFD_DOWN: MPLS BFD session for path path1 down on LSP R0_to_R3
```

To configure BFD for RSVP LSPs, include the `oam` and `bfd-liveness-detection` statements:

```
oam {
  bfd-liveness-detection {
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    minimum-transmit-interval milliseconds;
    multiplier detection-time-multiplier;
  }
}
```

For a list of hierarchy levels at which you can include these statements, see the statement summary sections for these statements.

The `bfd-liveness-detection` statement includes the following options:

- `minimum-interval`—Specifies the minimum transmit and receive interval.
- `minimum-receive-interval`—Specifies the minimum receive interval. The range is from 1 through 255,000 milliseconds.
- `minimum-transmit-interval`—Specifies the minimum transmit interval. The range is from 1 through 255,000 milliseconds.
- `multiplier`—Specifies the detection time multiplier. The range is from 1 through 255.



NOTE: To avoid triggering false negatives, configure a BFD fault detection time that is longer than the fast reroute time.

Pinging LSPs

The following sections describe the various functions of `ping mpls` command:

- Pinging an MPLS LSP on page 193
- Pinging an MPLS LSP Endpoint on page 193
- Pinging a CCC LSP on page 194
- Pinging a Layer 3 VPN on page 194



NOTE: The `ping mpls` command is not supported within logical routers or routing instances.

Pinging an MPLS LSP

You can ping a specific LSP. Echo requests are sent over the LSP as MPLS packets. The payload is a User Datagram Protocol (UDP) packet forwarded to the address 127.0.0.1. The label and interface information for building and sending this information as an MPLS packet is the same as for standard LSP traffic.

When the echo request arrives at the egress node, the receiver checks the contents of the packet and sends a reply containing the correct return value, by using UDP. The router sending the echo request waits to receive an echo reply after a timeout of 2 seconds (you cannot configure this value).

You must configure MPLS at the `[edit protocols mpls]` hierarchy level on the remote router to be able to ping an LSP terminating there. You must configure MPLS even if you intend to ping only LDP forwarding equivalence classes (FECs).

On the egress router (the router receiving the MPLS echo packets), you must configure the address 127.0.0.1/32 on its lo0 interface. If this is not configured, the egress router does not have this forwarding entry and therefore simply drops the incoming MPLS pings and replies with “ICMP host unreachable” messages.

To ping an MPLS LSP use the `ping mpls <count count> <ldp <fec>> <rsvp <exp forwarding-class> <lsp-name>>` command. To ping a secondary MPLS LSP, use the `ping mpls <count count> <rsvp <lsp-name>> standby path-name` command. For a detailed description of this command, see the *JUNOS Routing Protocols and Policies Command Reference*.

Pinging an MPLS LSP Endpoint

To determine whether an LSP between two provider edge (PE) routers is up and running, you can ping the endpoint address of the LSP. To ping an MPLS LSP endpoint, use the `ping mpls lsp-end-point address` command. This command tells you what type of LSP (RSVP or LDP) terminates at the address specified and whether that LSP is up or down.

For a detailed description of this command, see the *JUNOS Routing Protocols and Policies Command Reference*.

Pinging a CCC LSP

You can ping a specific CCC LSP. The CCC LSP ping command is identical to the one used for MPLS LSPs. The command you use is `ping mpls <count count> <rsvp <lsp-name>>`. You can also ping a secondary standby CCC LSP by using the `ping mpls <count count> <rsvp <lsp-name>> standby path-name` command.

For a detailed description of this command, see the *JUNOS Routing Protocols and Policies Command Reference*.

Pinging a Layer 3 VPN

You can use a similar command, `ping mpls l3vpn vpn-name prefix prefix <count count>`, to ping a Layer 3 VPN. For more information about this command, see the *JUNOS VPNs Configuration Guide* and the *JUNOS Routing Protocols and Policies Command Reference*.

LSP Ping and Traceroute Based on RFC 4379

The JUNOS software now partially supports LSP ping and traceroute commands based on RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*. However, the JUNOS software only supports this functionality on LSP transit routers. If a ping or traceroute command is issued from a router that fully supports RFC 4379, it can propagate correctly on routing platforms running the JUNOS software.

LSP ping and traceroute commands based on RFC 4379 attempt to trace the path taken by an LSP by relying on MPLS TTL expiration. An LSP can take multiple paths from ingress to egress. This occurs in particular with Equal Cost Multipath (ECMP). The LSP traceroute command can trace all possible paths to an LSP egress node.

Tracing MPLS and LSP Packets and Operations

To trace MPLS and LSP packets and operations, include the `traceoptions` statement:

```
traceoptions {
  file filename <replace> <size size> <files number> <no-stamp>
  <world-readable | no-world-readable>;
  flag flag;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

You can specify the following MPLS-specific flags in the MPLS `traceoptions` statement:

- `all`—Trace all operations.
- `connection`—Trace all circuit cross-connect (CCC) activity.
- `connection-detail`—Trace detailed CCC activity.
- `cspf`—Trace CSPF computations.

- `cspf-link`—Trace links visited during CSPF computations.
- `cspf-node`—Trace nodes visited during CSPF computations.
- `error`—Trace MPLS error conditions.
- `graceful-restart`—Trace MPLS graceful restart events.
- `lsping`—Trace LSP ping packets and return codes.
- `state`—Trace all LSP state transitions.

When you configure trace options to track an MPLS LSP using the `cspf` option, the CSPF log displays information about the MPLS LSP using the term “generalized MPLS” (GMPLS). For example, a message in the CSPF log might state that the “link passes GMPLS constraints”. Generalized MPLS (GMPLS) is a superset of MPLS, so this message is normal and does not affect proper MPLS LSP operation.

For general information about tracing and global tracing options, see the *JUNOS Routing Protocols Configuration Guide*.

