

## Chapter 14

# LDP Configuration Guidelines

The Label Distribution Protocol (LDP) is instance-aware. To configure the LDP protocol, include the `ldp` statement:

```
ldp {
  explicit-null;
  (deaggregate | no-deaggregate);
  egress-policy policy-name;
  export [ policy-names ];
  import [ policy-names ];
  graceful-restart {
    disable;
    helper-disable;
    maximum-recovery-time value;
    recovery-time value;
  }
  interface interface-name {
    apply-groups;
    disable;
    hello-interval seconds;
    hold-time seconds;
    transport-address (interface | router-id);
  }
  keepalive-interval seconds;
  keepalive-timeout seconds;
  log-updown {
    trap disable;
  }
  no-forwarding;
  policing {
    fec fec-address {
      ingress-traffic filter-name;
      transit-traffic filter-name;
    }
  }
  preference preference;
  session address {
    authentication-key authentication-key;
  }
  strict-targeted-hellos;
```

```

tracoptions {
  apply-groups;
  file filename <replace> <size size> <files number> <no-stamp>
    <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
track-igp-metric;
traffic-statistics {
  file filename <replace> <size size> <files number>
    <world-readable | no-world-readable>;
  interval interval;
}
transport-address (interface | router-id);
}

```

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

By default, LDP is disabled.

This chapter describes the minimum required LDP configuration and discusses the following configuration tasks:

- Minimum LDP Configuration on page 361
- Enabling and Disabling LDP on page 361
- Configuring the LDP Hello Interval on page 362
- Configuring the LDP Hold Time on page 362
- Configuring the LDP Keepalive Interval on page 363
- Configuring the LDP Keepalive Timeout on page 363
- Configuring LDP Route Preferences on page 363
- Configuring LDP Graceful Restart on page 364
- Configuring LDP Received-Label Filtering on page 367
- Configuring LDP Outbound-Label Filtering on page 369
- Configuring LDP Transport Address Control on page 371
- Configuring the LDP Egress Policy on page 372
- Configuring FEC Deaggregation on page 373
- Configuring Policers for LDP FECs on page 374
- Configuring BFD for LDP LSPs on page 375
- Collecting LDP Statistics on page 376

- Tracing LDP Protocol Traffic on page 379
- Configuring Miscellaneous LDP Properties on page 382

## Minimum LDP Configuration

---

To enable LDP on a single interface, include the `ldp` statement and specify the interface using the `interface` statement. This is the minimum LDP configuration. All other LDP configuration statements are optional.

```
ldp {
  interface interface-name;
}
```

To enable LDP on all interfaces, specify `all` for *interface-name*.

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

## Enabling and Disabling LDP

---

LDP is routing-instance-aware. To enable LDP on a specific interface, include the following statements:

```
ldp {
  interface interface-name;
}
```

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

To enable LDP on all interfaces, specify `all` for *interface-name*.

If you have configured interface properties on a group of interfaces and want to disable LDP on one of the interfaces, include the `interface` statement with the `disable` option:

```
interface interface-name {
  disable;
}
```

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

## Configuring the LDP Hello Interval

---

LDP hello messages enable LDP nodes to discover one another and to detect the failure of a neighbor or of the link to the neighbor. Hello messages are sent periodically on all interfaces where LDP is enabled.

By default, LDP sends hello messages every 5 seconds for link hello messages and every 15 seconds for targeted hello messages. You cannot configure a time for the LDP hello interval that is greater than the LDP hold time. For more information, see “Configuring the LDP Hold Time” on page 362.

To modify how often LDP sends hello packets, include the `hello-interval` statement:

```
hello-interval seconds;
```

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

## Configuring the LDP Hold Time

---

The hold time determines how long an LDP node should wait for a hello message before declaring a neighbor to be down. This value is sent as part of a hello message so that each LDP node tells its neighbors how long to wait. The values sent by each neighbor do not have to match.

The hold time should normally be at least three times the hello interval. The default is 15 seconds for link hello messages and 45 seconds for targeted hello messages. However, it is possible to configure an LDP hold time that is close to the value for the hello interval.



**NOTE:** Configuring an LDP hold time that is close to the hello interval can cause the router to declare an LDP neighbor down that is still functioning normally. For more information, see “Configuring the LDP Hello Interval” on page 362.

---

To modify the hold time, include the `hold-time` statement:

```
hold-time seconds;
```

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

## Configuring the LDP Keepalive Interval

---

The keepalive interval determines how often a message is sent over the session to ensure that the keepalive timeout is not exceeded. If no other LDP traffic is sent over the session in this much time, a keepalive message is sent. The default is 10 seconds.

To modify the keepalive interval, include the `keepalive-interval` statement:

```
keepalive-interval seconds;
```

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

## Configuring the LDP Keepalive Timeout

---

After an LDP session is established, messages must be exchanged periodically to ensure that the session is still working. The keepalive timeout defines the amount of time that the neighbor LDP node waits before deciding that the session has failed. This value is usually set to at least three times the keepalive interval. The default is 30 seconds.

To modify the keepalive interval, include the `keepalive-timeout` statement:

```
keepalive-timeout seconds;
```

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

The value configured for the `keepalive-timeout` statement is displayed as the hold time when you issue the `show ldp session detail` command.

## Configuring LDP Route Preferences

---

When several protocols calculate routes to the same destination, route preferences are used to select which route is installed in the forwarding table. The route with the lowest preference value is selected. The preference value can be a number in the range 0 through 255. By default, LDP routes have a preference value of 9.

To modify the route preferences, include the `preference` statement:

```
preference preference;
```

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

## Configuring LDP Graceful Restart

---

By default, graceful restart helper mode is enabled, but graceful restart is disabled. Thus, the default behavior of a router is to assist neighboring routers attempting a graceful restart, but not to attempt a graceful restart itself.

To configure LDP graceful restart, see the following sections:

- Enabling Graceful Restart on page 364
- Disabling LDP Graceful Restart or Helper Mode on page 365
- Configuring Recovery Time and Maximum Recovery Time on page 366

### Enabling Graceful Restart

To enable LDP graceful restart, you also need to enable graceful restart on the router. To enable graceful restart, include the `graceful-restart` statement:

```
graceful-restart;
```

You can include this statement at the following hierarchy levels:

- [edit routing-options]
- [edit logical-routers *logical-router-name* routing-options]

The `graceful-restart` statement enables graceful restart for all protocols supporting this feature on the router. For more information about graceful restart, see the *JUNOS Routing Protocols Configuration Guide*.

By default, LDP graceful restart is enabled when you enable graceful restart at both the LDP protocol level and on all the routing instances. However, you can disable both LDP graceful restart and LDP graceful restart helper mode.

### **Disabling LDP Graceful Restart or Helper Mode**

To disable LDP graceful restart and recovery, include the **disable** statement:

```
ldp {
    graceful-restart {
        disable;
    }
}
```

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

You can disable helper mode at the LDP protocols level only. You cannot disable helper mode for a specific routing instance. To disable LDP helper mode, include the **helper-disable** statement:

```
ldp {
    graceful-restart {
        helper-disable;
    }
}
```

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

The following LDP graceful restart configurations are possible:

- LDP graceful restart and helper mode are both enabled.
- LDP graceful restart is disabled but helper mode is enabled. A router configured in this way cannot restart gracefully but can help a restarting neighbor.
- LDP graceful restart and helper mode are both disabled. The router does not use LDP graceful restart or the graceful restart TLV sent in the initialization message. The router behaves as a router that cannot support LDP graceful restart.

A configuration error is issued if you attempt to enable graceful restart and disable helper mode.

## Configuring Recovery Time and Maximum Recovery Time

The recovery time is the amount of time a router waits for LDP to restart gracefully. The recovery time period begins when an initialization message is sent or received. This period is also typically the amount of time that a neighboring router maintains its information about the restarting router, allowing it to continue to forward traffic.

To prevent a neighboring router from being adversely affected if it receives a false value for the recovery time from the restarting router, you can configure the maximum recovery time on the neighboring router. A neighboring router maintains its state for the shorter of the two times. For example, router A is performing an LDP graceful restart. It has sent a recovery time of 900 seconds to neighboring router B. However, router B has its maximum recovery time configured at 400 seconds. Router B will only wait for 400 seconds before it purges its LDP information from router A.

To configure recovery time, include the `recovery-time` statement and the `maximum-recovery-time` statement:

```
graceful-restart {  
    maximum-recovery-time seconds;  
    recovery-time seconds;  
}
```

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

## Configuring LDP Received-Label Filtering

You can filter received LDP label bindings, applying policies to accept or deny bindings advertised by neighboring routers. To configure received-label filtering, include the `import` statement:

```
import [policy-name];
```

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

The named policy (configured at the `[edit policy-options]` hierarchy level) is applied to all label bindings received from all LDP neighbors. All filtering is done with `from` statements. Table 9 lists the only `from` operators that apply to LDP received-label filtering.

**Table 9: from Operators That Apply to LDP Received-Label Filtering**

from Operator	Description
interface	Matches on bindings received from a neighbor that is adjacent over the specified interface
neighbor	Matches on bindings received from the specified LDP router ID
next-hop	Matches on bindings received from a neighbor advertising the specified interface address
route-filter	Matches on bindings with the specified prefix

If a binding is filtered, it still appears in the LDP database, but is not considered for installation as part of a label-switched path (LSP).

Generally, applying policies in LDP can be used only to block the establishment of LSPs, not to control their routing. This is because the path that an LSP follows is determined by unicast routing, and not by LDP. However, when there are multiple equal-cost paths to the destination through different neighbors, you can use LDP filtering to exclude some of the possible next hops from consideration. (Otherwise, LDP chooses one of the possible next hops at random.)

LDP sessions are not bound to interfaces or interface addresses. LDP advertises only per-router (not per-interface) labels; so if multiple parallel links exist between two routers, only one LDP session is established, and it is not bound to a single interface. When a router has multiple adjacencies to the same neighbor, take care to ensure that the filter does what is expected. (Generally, using `next-hop` and `interface` is not appropriate in this case.)

If a label has been filtered (meaning that it has been rejected by the policy and is not used to construct an LSP), it is marked as filtered in the database:

```
user@host> show ldp database
Input label database, 10.10.255.1:0-10.10.255.6:0
Label Prefix
3 10.10.255.6/32 (Filtered)
Output label database, 10.10.255.1:0-10.10.255.6:0
Label Prefix
3 10.10.255.1/32 (Filtered)
```

For more information about how to configure policies for LDP, see the *JUNOS Policy Framework Configuration Guide*.

### Examples: Configuring Received-Label Filtering

Accept only /32 prefixes from all neighbors:

```
[edit]
protocols {
  ldp {
    import only-32;
    ...
  }
}
policy-options {
  policy-statement only-32 {
    term first {
      from {
        route-filter 0.0.0.0/0 upto /31;
      }
      then reject;
    }
    then accept;
  }
}
```

Accept 131.108/16 or longer from router ID 10.10.255.2 and accept all prefixes from all other neighbors:

```
[edit]
protocols {
  ldp {
    import nosy-neighbor;
    ...
  }
}
policy-options {
  policy-statement nosy-neighbor {
    term first {
      from {
        neighbor 10.10.255.2;
        route-filter 131.108.0.0/16 orlonger accept;
        route-filter 0.0.0.0/0 orlonger reject;
      }
    }
    then accept;
  }
}
```

## Configuring LDP Outbound-Label Filtering

You can configure export policies to filter LDP outbound labels. You can filter outbound label bindings by applying routing policies to block bindings from being advertised to neighboring routers. To configure outbound label filtering, include the **export** statement:

```
export [ policy-names ];
```

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

The named export policy (configured at the [edit **policy-options**] hierarchy level) is applied to all label bindings transmitted to all LDP neighbors. The only **from** operator that applies to LDP outbound label filtering is **route-filter**, which matches bindings with the specified prefix. The only **to** operators that apply to outbound label filtering are the operators in Table 10.

**Table 10: to Operators for LDP Outbound-Label Filtering**

to Operator	Description
interface	Matches on bindings sent to a neighbor that is adjacent over the specified interface
neighbor	Matches on bindings sent to the specified LDP router ID
next-hop	Matches on bindings sent to a neighbor advertising the specified interface address

If a binding is filtered, the binding is not advertised to the neighboring router, but it can be installed as part of an LSP on the local router. You can apply policies in LDP to block the establishment of LSPs, but not to control their routing. The path an LSP follows is determined by unicast routing, not by LDP.

LDP sessions are not bound to interfaces or interface addresses. LDP advertises only per-router (not per-interface) labels. If multiple parallel links exist between two routers, only one LDP session is established, and it is not bound to a single interface.

Do not use the `next-hop` and `interface` operators when a router has multiple adjacencies to the same neighbor.

Filtered labels are marked in the database:

```
user@host> show ldp database
Input label database, 10.10.255.1:0-10.10.255.3:0
Label Prefix
100007 10.10.255.2/32
3 10.10.255.3/32
Output label database, 10.10.255.1:0-10.10.255.3:0
Label Prefix
3 10.10.255.1/32
100001 10.10.255.6/32 (Filtered)
```

For more information about how to configure policies for LDP, see the *JUNOS Policy Framework Configuration Guide*.

### **Examples: Configuring Outbound-Label Filtering**

Block transmission of 10.10.255.6/32 to all neighbors:

```
[edit protocols]
ldp {
  export block-one;
}
policy-options {
  policy-statement block-one {
    term first {
      from {
        route-filter 10.10.255.6/32 exact;
      }
      then reject;
    }
    then accept;
  }
}
```

Send only 131.108/16 or longer to router ID 10.10.255.2, and send all prefixes to all other routers:

```
[edit protocols]
ldp {
  export limit-lsps;
}
policy-options {
  policy-statement limit-lsps {
    term allow-one {
      from {
        route-filter 131.108.0.0/16 orlonger;
      }
      to {
        neighbor 10.10.255.2;
      }
      then accept;
    }
    term block-the-rest {
      to {
        neighbor 10.10.255.2;
      }
      then reject;
    }
    then accept;
  }
}
```

## Configuring LDP Transport Address Control

---

You can control the transport address used by LDP. The transport address is the address used for the TCP session over which LDP is running. To configure transport address control, include the `transport-address` statement:

```
transport-address ( router-id | interface );
```

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

If you select `router-id`, the address of the router identifier is used as the transport address (unless otherwise configured, the router identifier is typically the same as the loopback address). If you select `interface`, the interface address is used as the transport address for any LDP sessions to neighbors that can be reached over that interface. Note that the router identifier is used as the transport address by default.

You cannot use `transport-address interface` when there are multiple parallel links to the same LDP neighbor, because the LDP specification requires that the same transport address be advertised on all interfaces to the same neighbor. If LDP detects multiple parallel links to the same neighbor, it disables interfaces to that neighbor one by one until the condition is cleared, either by disconnecting the neighbor on an interface or by configuring `transport-address router-id`.

## Configuring the LDP Egress Policy

You can control the set of prefixes that are advertised into LDP and cause the router to be the egress router for those prefixes. By default, only the loopback address is advertised into LDP. To configure the set of prefixes from the routing table to be advertised into LDP, include the `egress-policy` statement:

```
egress-policy policy-name;
```

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



**NOTE:** If you configure an egress policy for LDP that does not include the loopback address, it is no longer advertised in LDP. To continue to advertise the loopback address, you need to explicitly configure it as a part of the LDP egress policy.

The named policy (configured at the [edit policy-options] or the [edit logical-routers *logical-router-name* policy-options] hierarchy level) is applied to all routes in the routing table. Those routes that match the policy are advertised into LDP. You can control the set of neighbors to which those prefixes are advertised by using the `export` statement. Only `from` operators are considered; you can use any valid `from` operator. For more information, see the *JUNOS Routing Protocols Configuration Guide*.

### Example: Configuring the LDP Egress Policy

Advertise all connected routes into LDP:

```
[edit protocols]
ldp {
  egress-policy connected-only;
}
policy-options {
  policy-statement connected-only {
    from {
      protocol direct;
    }
    then accept;
  }
}
```

## Configuring FEC Deaggregation

---

When an LDP egress router advertises multiple prefixes, the prefixes are bound to a single label and aggregated into a single forwarding equivalence class (FEC). By default, LDP maintains this aggregation as the advertisement traverses the network.

By default, because an LSP cannot be split across multiple next hops and all the prefixes are bound into a single LSP, you cannot load-balance across equal-cost paths.

To change the default to load-balance across equal-cost paths, you need to deaggregate the FECs. Deaggregating the FECs causes each prefix to be bound to a separate label and become a separate LSP.

To configure deaggregated FECs, include the **deaggregate** statement:

```
deaggregate;
```

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

For all LDP sessions, you can configure deaggregated FECs only globally.

Deaggregating a FEC allows the resulting multiple LSPs to be distributed across multiple equal-cost paths and distributes LSPs across the multiple next hops on the egress segments but installs only one next hop per LSP.

To aggregate FECs, include the **no-deaggregate** statement:

```
no-deaggregate;
```

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

For all LDP sessions, you can configure aggregated FECs only globally.

## Configuring Policers for LDP FECs

You can configure the JUNOS software to track and police traffic for LDP FECs. LDP FEC policers can be used to do any of the following:

- Track or police the ingress traffic for an LDP FEC.
- Track or police the transit traffic for an LDP FEC.
- Track or police LDP FEC traffic originating from a specific forwarding class.
- Track or police LDP FEC traffic originating from a specific virtual routing and forwarding (VRF) site.
- Discard false traffic bound for a specific LDP FEC.

To police traffic for an LDP FEC, you must first configure a filter. Specifically, you need to configure either the `interface` statement or the `interface-set` statement at the `[edit firewall family protocol-family filter filter-name term term-name from]` hierarchy level. The `interface` statement allows you to match the filter to a single interface. The `interface-set` statement allows you to match the filter to multiple interfaces.

For more information on how to configure the `interface` statement, the `interface-set` statement, and policers for LDP FECs, see the *JUNOS Policy Framework Configuration Guide*.

Once you have configured the filters, you need to include them in the `policing` statement configuration for LDP. To configure policers for LDP FECs, include the `policing` statement:

```
policing {
  fec fec-address {
    ingress-traffic filter-name;
    transit-traffic filter-name;
  }
}
```

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

The `policing` statement includes the following options:

- `fec`—Specify the FEC address for the LDP FEC you want to police.
- `ingress-filter`—Specify the name of the ingress traffic filter.
- `transit-traffic`—Specify the name of the transit traffic filter.

## Configuring BFD for LDP LSPs

You can configure Bidirectional Forwarding Detection (BFD) for LDP LSPs. BFD can be turned on for all or a subset of LDP IPv4 FECs.

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

```
RPD_LDP_BFD_UP: LDP BFD session for FEC 10.255.16.14/32 is up
RPD_LDP_BFD_DOWN: LDP BFD session for FEC 10.255.16.14/32 is down
```

You can also configure BFD for RSVP LSPs, as described in “Configuring BFD for RSVP LSPs” on page 192.

To enable BFD for LDP LSPs, include the `oam` and `bfd-liveness-detection` statements. BFD is enabled for all LDP LSPs on the router.

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

You can configure this statement at the following hierarchy levels:

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

The `oam` statement includes the following option:

- `fec`—Specify the FEC address.

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

- `minimum-interval`—Specifies the minimum transmit and receive interval. If you configure the `minimum-interval` option, you do not need to configure the `minimum-receive-interval` option or the `minimum-transmit-interval` option.
- `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.

## Collecting LDP Statistics

---

LDP traffic statistics show the volume of traffic that has passed through a particular FEC on a router.

When you configure the `traffic-statistics` statement at the `[edit protocols ldp]` hierarchy level, the LDP traffic statistics are gathered periodically and written to a file. You can configure how often statistics are collected (in seconds) by using the `interval` option. The default collection interval is 5 minutes. You must configure an LDP statistics file; otherwise LDP traffic statistics are not gathered. If the LSP goes down, the LDP statistics are reset.

To collect LDP traffic statistics, include the `traffic-statistics` statement:

```
traffic-statistics {  
  file filename <files number> <replace> <size size>  
    <world-readable | no-world-readable>;  
  interval interval;  
  no-penultimate-hop;  
}
```

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

This section includes the following topics:

- LDP Statistics Output on page 377
- Disabling LDP Statistics on the Penultimate-Hop Router on page 378
- LDP Statistics Limitations on page 378

## LDP Statistics Output

The following is sample output from an LDP statistics file:

FEC	Type	Packets	Bytes	Shared
10.255.350.448/32	Transit	0	0	No
	Ingress	0	0	No
10.255.350.450/32	Transit	0	0	Yes
	Ingress	0	0	No
10.255.350.451/32	Transit	0	0	No
	Ingress	0	0	No
220.220.220.1/32	Transit	0	0	Yes
	Ingress	0	0	No
220.220.220.2/32	Transit	0	0	Yes
	Ingress	0	0	No
220.220.220.3/32	Transit	0	0	Yes
	Ingress	0	0	No

May 28 15:02:05, read 12 statistics in 00:00:00 seconds

The following describes each column of data collected in the LDP statistics file:

- **Bytes**—Number of bytes of data passed by the FEC since its LSP came up.
- **FEC**—FEC for which LDP traffic statistics are collected.
- **Packets**—Number of packets passed by the FEC since its LSP came up.
- **read**—This number (which appears next to the date and time) might differ from the actual number of the statistics displayed. Some of the statistics are summarized before being displayed.
- **Shared**—A **Yes** value indicates that several prefixes are bound to the same label (for example, when several prefixes are advertised with an egress policy). The LDP traffic statistics for this case apply to all the prefixes and should be treated as such.
- **Type**—Type of traffic originating from a router, either **Ingress** (originating from this router) or **Transit** (forwarded through this router).

## Disabling LDP Statistics on the Penultimate-Hop Router

Gathering LDP traffic statistics at the penultimate-hop router can consume excessive system resources, next-hop routes in particular. This problem is exacerbated if you have configured the `deaggregate` statement in addition to the `traffic-statistics` statement. For routers reaching their limit of next-hop route usage, we recommend configuring the `no-penultimate-hop` option for the `traffic-statistics` statement:

```
traffic-statistics {
  no-penultimate-hop;
}
```

For a list of hierarchy levels at which you can configure the `traffic-statistics` statement, see the statement summary section for this statement.



**NOTE:** When you configure the `no-penultimate-hop` option, no statistics are available for the FECs that are the penultimate hop for this router.

Whenever you include or remove this option from the configuration, the LDP sessions are taken down and then restarted.

The following is sample output from an LDP statistics file showing routers on which the `no-penultimate-hop` option is configured:

FEC	Type	Packets	Bytes	Shared
10.255.245.218/32	Transit	0	0	No
	Ingress	4	246	No
10.255.245.221/32	Transit	statistics disabled		
	Ingress	statistics disabled		
13.1.1.0/24	Transit	statistics disabled		
	Ingress	statistics disabled		
13.1.3.0/24	Transit	statistics disabled		
	Ingress	statistics disabled		

## LDP Statistics Limitations

The following are issues related to collecting LDP statistics by configuring the `traffic-statistics` statement:

- You cannot clear the LDP statistics.
- If you shorten the specified interval, a new LDP statistics request is issued only if the statistics timer expires later than the new interval.
- A new LDP statistics collection operation cannot start until the previous one has finished. If the interval is short or if the number of LDP statistics is large, the time gap between the two statistics collections might be longer than the interval.

When an LSP goes down, the LDP statistics are reset.

## Tracing LDP Protocol Traffic

---

The following sections describe how to configure the trace options to examine LDP protocol traffic:

- Tracing LDP Protocol Traffic at the Protocol and Routing Instance Levels on page 379
- Tracing LDP Protocol Traffic Within FEC on page 380
- Examples: Tracing LDP Protocol Traffic on page 381

### **Tracing LDP Protocol Traffic at the Protocol and Routing Instance Levels**

To trace LDP protocol traffic, you can specify options in the global `traceoptions` statement at the `[edit routing-options]` hierarchy level, and you can specify LDP-specific options by including the `traceoptions` statement:

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

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

Use the `file` statement to specify the name of the file that receives the output of the tracing operation. All files are placed in the directory `/var/log`. We recommend that you place LDP-tracing output in the file `ldp-log`.

The following trace flags display the operations associated with the sending and receiving of various LDP messages. Each can carry one or more of the following modifiers:

- **address**—Trace the operation of address and address withdrawal messages.
- **binding**—Trace label-binding operations.
- **error**—Trace error conditions.
- **event**—Trace protocol events.
- **initialization**—Trace the operation of initialization messages.
- **label**—Trace the operation of label request, label map, label withdrawal, and label release messages.
- **notification**—Trace the operation of notification messages.
- **packet**—Trace the operation of address, address withdrawal, initialization, label request, label map, label withdrawal, label release, notification, and periodic messages. This modifier is equivalent to setting the **address**, **initialization**, **label**, **notification**, and **periodic** modifiers.
- **path**—Trace label-switched path operations.
- **periodic**—Trace the operation of hello and keepalive messages.
- **route**—Trace the operation of route messages.
- **state**—Trace protocol state transitions.

### **Tracing LDP Protocol Traffic Within FEC**

You can trace LDP protocol traffic within a specific FEC. You can filter LDP trace statements based on a FEC. The following trace flags are available for this purpose: **route**, **path**, and **binding**.

The following example illustrates how you might configure the LDP **traceoptions** statement to filter LDP trace statements based on a FEC:

```
[edit protocols ldp traceoptions]
  flag route filter match-on fec policy "filter-policy-for-ldp-fec";
```

This feature has the following limitations:

- The filtering capability is only available for FECs composed of IP version 4 (IPv4) prefixes.
- Layer 2 circuit FECs cannot be filtered.
- When you configure both route tracing and filtering, Multiprotocol Label Switching (MPLS) routes are not displayed (they are blocked by the filter).

- Filtering is determined by the policy and the configured value for the `match-on` option. When configuring the policy, be sure that the default behavior is always `reject`.
- The only `match-on` option is `fec`. Consequently, the only type of policy you should include is a route-filter policy.

### **Examples: Tracing LDP Protocol Traffic**

Trace LDP path messages in detail:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5;
      flag path;
    }
  }
}
```

Trace all LDP outgoing messages:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5;
      flag packets;
    }
  }
}
```

Trace all LDP error conditions:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5;
      flag error;
    }
  }
}
```

Trace all LDP incoming messages and all label-binding operations:

```
[edit]
protocols {
  ldp {
    traceoptions {
      file ldp size 10m files 5 world-readable;
      flag packets receive;
      flag binding;
    }
    interface all {
    }
  }
}
```

## Configuring Miscellaneous LDP Properties

---

The following sections describe how to configure a number of miscellaneous LDP properties:

- [Configuring LDP to Use the IGP Route Metric on page 382](#)
- [Preventing Ingress Routes from Being Added to inet.0 on page 383](#)
- [Multiple-Instance LDP and Carrier-of-Carriers VPNs on page 383](#)
- [Configuring MPLS and LDP to Pop the Label on the Ultimate-Hop Router on page 383](#)
- [Enabling LDP over RSVP-Established LSPs on page 384](#)
- [Enabling LDP over RSVP-Established LSPs in Heterogeneous Networks on page 384](#)
- [Configuring the TCP MD5 Signature for an LDP Session on page 385](#)
- [Disabling SNMP Traps for LDP on page 385](#)
- [Enabling Strict Targeted Hellos on page 386](#)
- [Configuring LDP Synchronization with the IGP on page 386](#)

### Configuring LDP to Use the IGP Route Metric

Use the `track-igp-metric` statement if you want the interior gateway protocol (IGP) route metric to be used for the LDP routes instead of the default LDP route metric (the default LDP route metric is 1).

To use the IGP route metric, include the `track-igp-metric` statement:

```
track-igp-metric;
```

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

### Preventing Ingress Routes from Being Added to inet.0

By configuring the `no-forwarding` statement, you can prevent ingress routes from being added to the `inet.0` routing table instead of the `inet.3` routing table even if you enabled the `traffic-engineering bgp-igp` statement at the `[edit protocols mpls]` or the `[edit logical-routers logical-router-name protocols mpls]` hierarchy level. By default, the `no-forwarding` statement is disabled.

To omit ingress routes from the `inet.0` routing table, include the `no-forwarding` statement:

```
no-forwarding;
```

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

### Multiple-Instance LDP and Carrier-of-Carriers VPNs

By configuring multiple LDP routing instances, you can use LDP to advertise labels in a carrier-of-carriers VPN from a service provider provider edge (PE) router to a customer carrier customer edge (CE) router. This is especially useful when the carrier customer is a basic Internet service provider (ISP) and wants to restrict full Internet routes to its PE routers. By using LDP instead of BGP, the carrier customer shields its other internal routers from the Internet. Multiple-instance LDP is also useful when a carrier customer wants to provide Layer 2 or Layer 3 VPN services to its customers.

For an example of how to configure multiple LDP routing instances for carrier-of-carriers VPNs, see the *JUNOS Feature Guide*.

### Configuring MPLS and LDP to Pop the Label on the Ultimate-Hop Router

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. If ultimate-hop popping is enabled, label 0 (IPv4 Explicit Null label) is advertised. Ultimate-hop popping ensures that any packets traversing an MPLS network include a label.

To configure ultimate-hop popping, include the `explicit-null` statement:

```
explicit-null;
```

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



**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.

### Enabling LDP over RSVP-Established LSPs

You can run LDP over LSPs established by RSVP, effectively tunneling the LDP-established LSP through the one established by RSVP. To do so, enable LDP on the `lo0.0` interface (see “Enabling and Disabling LDP” on page 361). You must also configure the LSPs over which you want LDP to operate by including the `ldp-tunneling` statement at the `[edit protocols mpls label-switched-path lsp-path-name]` hierarchy level:

```
[edit]
protocols {
  mpls {
    label-switched-path lsp-path-name {
      from source;
      to destination;
      ldp-tunneling;
    }
  }
}
```

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

For more information about tunneling LDP LSPs, see “Tunneling LDP LSPs in RSVP LSPs” on page 353.

### Enabling LDP over RSVP-Established LSPs in Heterogeneous Networks

Some other vendors use an OSPF metric of 1 for the loopback address. Juniper Networks routers use an OSPF metric of 0 for the loopback address. This can cause interoperability problems when you configure LDP tunneling over RSVP LSPs in heterogeneous networks.

When a Juniper Networks router is linked to another vendor’s router through an RSVP tunnel, and LDP tunneling is also enabled, by default the Juniper Networks router might not use the RSVP tunnel to route traffic to the LDP destinations downstream of the other vendor’s egress router if the RSVP path has a metric of 1 larger than the physical OSPF path.

To ensure that LDP tunneling functions properly in heterogeneous networks, you can configure OSPF to ignore the RSVP LSP metric by including the `ignore-lsp-metrics` statement:

```
ignore-lsp-metrics;
```

You can configure this statement at the following hierarchy levels:

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

To enable LDP over RSVP LSPs, you also still need to complete the procedure in “Enabling LDP over RSVP-Established LSPs” on page 384.

## Configuring the TCP MD5 Signature for an LDP Session

You can configure an MD5 signature for an LDP TCP connection to protect against the introduction of spoofed TCP segments into LDP session connection streams.

A router using the MD5 signature option is configured with a password for each peer for which authentication is required. The password is stored encrypted.

LDP hello adjacencies can still be created even when peering interfaces are configured with different security signatures. However, the TCP session cannot be authenticated and is never established.



**NOTE:** If you apply an MD5 signature to an LDP interface with an established session, it drops the TCP connection and all the associated label bindings to the FEC entries for that session. The session regenerates the database information for that session once both interfaces agree on a common security method and password.

To configure an MD5 signature for an LDP TCP connection, include the `session` and `authentication-key` statement:

```
session address {
    authentication-key md5-authentication-key;
}
```

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

Use the `session` statement to configure the address for the remote end of the LDP session.

The `md5-authentication-key` (password) can be up to 69 characters long. Characters can include any ASCII strings. If you include spaces, enclose all characters in quotation marks.

## Disabling SNMP Traps for LDP

Whenever an LDP LSP makes a transition from up to down, or down to up, the router sends a Simple Network Management Protocol (SNMP) trap. However, it is possible to disable the LDP SNMP traps on a router, logical router, or routing instance.

For information about the LDP SNMP traps and the proprietary LDP Management Information Base (MIB), see the *JUNOS Network Management Configuration Guide*.

To disable SNMP traps for LDP, specify the `trap disable` option for the `log-updown` statement:

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

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

## Enabling Strict Targeted Hellos

This feature prevents LDP sessions from being established with remote neighbors that have not been specifically configured. If you configure the `strict-targeted-hellos` statement, an LDP peer will not respond to targeted hellos coming from a source that is not one of the configured remote neighbors. Configured remote neighbors can include:

- Endpoints of RSVP tunnels for which LDP tunneling is configured
- Layer 2 circuit neighbors

A hello from an unconfigured neighbor is ignored, and an error is logged (with the `error` traceoption flag) indicating the source. For example, if a targeted hello is received from `10.0.0.1`, and no neighbor with this address is specifically configured, the following message is printed to the LDP log file:

```
LDP: Ignoring targeted hello from 10.0.0.1
```

To enable strict targeted hellos, include the `strict-targeted-hellos` statement:

```
strict-targeted-hellos;
```

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

## Configuring LDP Synchronization with the IGP

LDP is a protocol for distributing labels in non-traffic-engineered applications. Labels are distributed along the best path determined by the IGP. If synchronization between LDP and the IGP is not maintained, the LSP goes down. When LDP is not fully operational on a given link (a session is not established and labels are not exchanged), the IGP advertises the link with the maximum cost metric. The link is not preferred but remains in the network topology.

LDP synchronization is supported only on active point-to-point interfaces and LAN interfaces configured as point-to-point under the IGP. LDP synchronization is not supported during graceful restart.

To advertise the maximum cost metric until LDP is operational for synchronization, include the `ldp-synchronization` statement:

```
ldp-synchronization {
  disable;
  hold-time seconds;
}
```

To disable synchronization, include the `disable` statement. To configure the time period to advertise the maximum cost metric for a link that is not fully operational, include the `hold-time` statement.

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