

Chapter 14

Configure LDP

To configure LDP, you include statements at the [edit protocols ldp] hierarchy level of the configuration.

```
protocols {
  ldp {
    import [policy-names];
    interface interface-name {
      disable;
      hello-interval seconds;
      hold-time seconds;
    }
    keepalive-interval seconds;
    keepalive-timeout seconds;
    preference preference;
    traceoptions {
      file filename <replace> <size size> <files number> <no-stamp>
        <(world-readable | no-world-readable)>;
      flag flag <flag-modifier> <disable>;
    }
  }
}
```

By default, LDP is disabled.

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

- Minimum LDP Configuration on page 144
- Enable LDP on page 144
- Configure the LDP Hello Interval on page 144
- Configure the Hold Time on page 145
- Configure the Keepalive Interval on page 145
- Configure the Keepalive Timeout on page 145
- Configure Route Preferences on page 145
- Configure Received Label Filtering on page 146

Enable LDP over RSVP-Established LSPs on page 148

Trace LDP Protocol Traffic on page 148

Minimum LDP Configuration

To enable LDP on all interfaces, include the following statement in the configuration file. All other LDP configuration statements are optional.

```
[edit]
protocols {
  ldp {
    interface all;
  }
}
```

Enable LDP

To enable LDP on a specific interface, include the following statements at the [edit] hierarchy level:

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

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 `disable` statement within the LDP interface statement:

```
[edit]
protocols {
  ldp {
    interface interface-name {
      disable;
    }
  }
}
```

Configure 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. To modify how often LDP sends hello packets, include the `hello-interval` statement at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]
hello-interval seconds;
```

Configure the 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 be at least three times the hello interval. The default is 15 seconds. To modify the hold time, include the hold-time statement at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]  
hold-time seconds;
```

Configure the 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 at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]  
keepalive-interval seconds;
```

Configure the 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 at the [edit protocols ldp interface *interface-name*] hierarchy level:

```
[edit protocols ldp interface interface-name]  
keepalive-timeout seconds;
```

Configure 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 at the [edit protocols ldp] hierarchy level:

```
[edit protocols ldp]  
preference preference;
```

Configure Received Label Filtering

You can filter received LDP label bindings, applying policies to accept or deny bindings advertised by neighboring routers. To configure label filtering, include the `import` statement at the `[edit protocols ldp interface interface-name]` hierarchy level.

```
[edit protocols ldp interface interface-name]  
import policy-name;
```

The named policy (defined in the `[edit policy-options]` hierarchy level) is applied to all label bindings received from all LDP neighbors. All selection is done using `from` statements. Table 2 lists the only `from` operators that apply to LDP.

Table 2: from Operators that Apply to LDP

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.
nexthop	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.

Generally, applying policies in LDP can only be used 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 will choose one of the possible next hops at random.)

LDP sessions are not bound to interfaces or interface addresses. JUNOS software 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. Care must be taken when a router has multiple adjacencies to the same neighbor to ensure that the filter will do what is expected. (In most cases, the use of `nexthop` and `interface` is not appropriate in this case.)

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

```
root@ospf6> show ldp database detail

Input label database, 10.10.255.6:0-10.10.255.5:0
Label      Prefix
100020     10.10.255.2/32
State: Active, Filtered
          3     10.10.255.5/32
State: Active
Output label database, 10.10.255.6:0-10.10.255.5:0
Label      Prefix
3          10.10.255.6/32
State: Active
```

Examples: Configure 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 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;
  }
}
```

Enable 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, you must enable LDP on the lo0.0 interface (see “Enable LDP” on page 144). Additionally, you must configure the LSPs over which you want LDP to operate to include the `ldp-tunneling` statement:

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

Trace LDP Protocol Traffic

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 at the `[edit protocols ldp]` hierarchy level:

```
[edit protocols ldp]
traceoptions {
  file filename <replace> <size size> <files number> <no-stamp>
  <(world-readable | no-world-readable)>;
  flag flag <flag-modifier> <disable>;
}
```

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.

`packet-dump`—Display the contents of the messages selected with the message operation flags.

`path`—Trace label-switched path operations.

periodic—Trace the operation of hello and keepalive messages.

state —Trace protocol state transitions.

Examples: Trace 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;
    }
  }
}
```

Example: LDP Configuration

The following shows an example LDP configuration:

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

.....