

Chapter 11

RSVP Configuration Guidelines

To configure the Resource Reservation Protocol (RSVP), you include the `rsvp` statement:

```
protocols {
  rsvp {
    disable;
    graceful-deletion-timeout seconds;
    graceful-restart {
      disable;
      helper-disable;
      maximum-helper-recovery-time seconds;
      maximum-helper-restart-time seconds;
    }
  }
  interface interface-name {
    disable;
    (aggregate | no-aggregate);
    authentication-key key;
    bandwidth bps;
    hello-interval seconds;
    link-protection {
      disable;
      bandwidth bps;
      bypass bypass-name {
        bandwidth bps;
        hop-limit number;
        path address <strict | loose>;
        priority setup-priority reservation-priority;
        to address;
      }
    }
    class-of-service cos-value;
    hop-limit number;
    max-bypasses number;
    no-node-protection;
    optimize-timer seconds;
    path address <strict | loose>;
    subscription percent;
  }
  (reliable | no-reliable);
```

```

        subscription percentage {
            ct0 percentage;
            ct1 percentage;
            ct2 percentage;
            ct3 percentage;
        }
        update-threshold percentage;
    }
    keep-multiplier number;
    load-balance {
        bandwidth;
    }
    peer-interface peer-interface-name {
        (aggregate | no-aggregate);
        authentication-key key;
        disable;
        hello-interval seconds;
        (reliable | no-reliable);
    }
    preemption {
        (aggressive | disabled | normal);
        soft-preemption {
            cleanup-timer seconds;
        }
    }
    refresh-time seconds;
    traceoptions {
        file filename <replace> <size size> <files number> <no-stamp>
            <(world-readable | no-world-readable)>;
        flag flag <flag-modifier> <disable>;
    }
}

```

You can include this statement at the following hierarchy levels:

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

By default, RSVP is disabled.

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

- Minimum RSVP Configuration on page 277
- Configuring RSVP and MPLS on page 278
- Configuring RSVP Interface Properties on page 279
- Configuring Node Protection or Link Protection on page 286
- Configuring RSVP Graceful Restart on page 293
- Configuring RSVP LSP Load Balancing on page 295
- Configuring RSVP Timers on page 296
- Preempting RSVP Sessions on page 297
- Configuring MTU Signaling in RSVP on page 298
- Configuring RSVP to Pop the Label on the Ultimate-Hop Router on page 299
- Tracing RSVP Protocol Traffic on page 300

Minimum RSVP Configuration

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

```
rsvp {
    interface interface-name;
}
```

You can include this statement at the following hierarchy levels:

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

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

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

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

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name*]

Configuring RSVP and MPLS

The primary purpose of the JUNOS RSVP software is to support dynamic signaling within label-switched paths (LSPs). When you enable both MPLS and RSVP on a router, MPLS becomes a client of RSVP. No additional configuration is required to bind MPLS and RSVP.

You can configure MPLS to set up signaled paths by using the `label-switched-path` statement at the [edit protocols mpls] hierarchy level. Each LSP translates into a request for RSVP to initiate an RSVP session. This request is passed through the internal interface between label switching and RSVP. After examining the request information, checking RSVP states, and checking the local routing tables, RSVP initiates one session for each LSP. The session is sourced from the local router and is destined to the target of the LSP.

When an RSVP session is successfully created, the LSP is set up along the paths created by the RSVP session. If the RSVP session is unsuccessful, RSVP notifies MPLS of its status. It is up to MPLS to initiate backup paths or continue retrying the initial path.

To pass label-switching signaling information, RSVP supports four additional objects: Label Request object, Label object, Explicit Route object, and Record Route object. For an LSP to be set up successfully, all routers along the path must support MPLS, RSVP, and the four objects. Of the four objects, the Record Route object is not mandatory.

To configure MPLS and make it a client of RSVP, do the following:

- Enable MPLS on all routers that will participate in label switching (that is, on all routers that might be part of a label-switching path).
- Enable RSVP on all routers and on all router interfaces that form the LSP.
- Configure the routers at the beginning of the LSP.

Example: Configuring RSVP and MPLS

The following shows a sample configuration for a router at the beginning of an LSP:

```
[edit]
protocols {
  mpls {
    label-switched-path sf-to-london {
      to 192.168.1.4;
    }
  }
  rsvp {
    interface so-0/0/0;
  }
}
```

The following shows a sample configuration for all the other routers that form the LSP:

```
[edit]
protocols {
  mpls {
    interface so-0/0/0;
  }
  rsvp {
    interface so-0/0/0;
  }
}
```

Configuring RSVP Interface Properties

The following sections describe how to configure the interface properties for RSVP:

- Configuring RSVP Refresh Reduction on page 280
- Configuring the RSVP Hello Interval on page 283
- Configuring RSVP Authentication on page 284
- Configuring the Bandwidth Subscription for Class Types on page 284
- Configuring the RSVP Update Threshold on an Interface on page 285

Configuring RSVP Refresh Reduction

You can configure RSVP refresh reduction on each interface. The following statements allow you to configure the RSVP refresh reduction features:

- **aggregate**—Enable all RSVP refresh reduction features: RSVP message bundling, RSVP message ID, reliable message delivery, and summary refresh.
- **no-aggregate**—Disable RSVP message bundling and summary refresh.
- **reliable**—Enable RSVP message ID and reliable message delivery.
- **no-reliable**—Disable RSVP message ID, reliable message delivery, and summary refresh.

For more information on RSVP refresh reduction, see “RSVP Refresh Reduction” on page 262.

Table 8 lists various combinations of the RSVP refresh reduction configuration statements and how they alter the behavior of the JUNOS software. The table describes only the expected behavior based on the configuration on the router. The actual behavior is dictated not only by the local configuration on this router, but also on the refresh reduction capabilities of its RSVP neighbors. Note that by configuring the **aggregate** statement, you enable all RSVP refresh reduction features including reliable message delivery.

Table 8: RSVP Refresh Reduction Behavior

| Configuration Statement | Send Capability | Receive Capability |
|--|---|--|
| aggregate or aggregate and reliable | RR bit = 1 Bundle Message ID (Path/Resv messages) Ack/Nack (all messages) Summary Refresh | Bundle Ack/Nack (all messages) Summary Refresh |
| aggregate and no-reliable | RR bit = 1 Bundle Ack/Nack (all messages) | Bundle Message ID (all messages) |
| reliable or reliable and no-aggregate | RR bit = 0 Message ID (Path/Resv messages) Ack/Nack (all messages) | Bundle Message ID (all messages) Ack/Nack |

The send capability shown in Table 8 lists the RSVP messages and objects related to RSVP refresh reduction that the router is capable of sending. This does not mean that all these messages are exchanged between this router and a neighbor. For example, if the router is configured with the **aggregate** statement, but RSVP refresh reduction is not enabled on its neighbor, then no Summary Refresh message is sent to this neighbor even though the router is capable of sending it.

The receive capability shown in Table 8 lists the messages and objects related to RSVP refresh reduction that the router is capable of receiving and processing without generating any errors or resulting in error conditions.

If the `no-reliable` statement is configured on the router (reliable message delivery is disabled), the router accepts RSVP messages that include the Message ID object but ignore the Message ID object and continue performing standard message processing. No error is generated in this case and RSVP operates normally.

However, not all combinations between two neighbors with different refresh reduction capabilities function correctly. For example, a router is configured with either the `aggregate` and `no-reliable` statements or with the `reliable` and `no-aggregate` statements. If an RSVP neighbor sends a Summary Refresh object to this router, no error is generated, but the Summary Refresh object cannot be processed. Consequently, RSVP states can time out on this router if the neighbor is relying only on Summary Refresh to refresh those RSVP states.

We recommend, unless there are specific requirements, that you configure RSVP refresh reduction in a similar manner on each RSVP neighbor.

To enable all RSVP refresh reduction features on an interface, include the `aggregate` statement:

```
aggregate;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name*]

To disable RSVP message bundling and summary refresh, include the `no-aggregate` statement:

```
no-aggregate;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name*]

To enable RSVP message ID and reliable message delivery on an interface, include the `reliable` statement:

```
reliable;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name*]

To disable RSVP message ID, reliable message delivery, and summary refresh, include the `no-reliable` statement:

```
no-reliable;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name*]

Determining the Refresh Reduction Capability of RSVP Neighbors

To determine the RSVP refresh reduction capability of an RSVP neighbor, you need the following information:

- The RR bit advertised by the neighbor
- The local configuration of RSVP refresh reduction
- The actual RSVP messages received from the neighbor

To obtain this information, you can issue a `show rsvp neighbor detail` command. The following is a sample of output from this command:

```
user@host> show rsvp neighbor detail
RSVP neighbor: 6 learned
  Address: 192.168.224.178 via: fxp1.0 status: Up
    Last changed time: 10:06, Idle: 5 sec, Up cnt: 1, Down cnt: 0
    Message received: 36
    Hello: sent 69, received: 69, interval: 9 sec
    Remote instance: 0x60b8feba, Local instance: 0x74bc7a8d
    Refresh reduction: not operational

  Address: 192.168.224.186 via: fxp2.0 status: Down
    Last changed time: 10:17, Idle: 40 sec, Up cnt: 0, Down cnt: 0
    Message received: 6
    Hello: sent 20, received: 0, interval: 9 sec
    Remote instance: 0x0, Local instance: 0x2ae1b339
    Refresh reduction: incomplete
    Remote end: disabled, Ack-extension: enabled

  Address: 192.168.224.188 via: fxp2.0 status: Up
    Last changed time: 4:15, Idle: 0 sec, Up cnt: 1, Down cnt: 0
    Message received: 55
    Hello: sent 47, received: 31, interval: 9 sec
    Remote instance: 0x6436a35b, Local instance: 0x663849f0
    Refresh reduction: operational
    Remote end: enabled, Ack-extension: enabled
```

For more information on the `show rsvp neighbor detail` command, see the *JUNOS Routing Protocols and Policies Command Reference*.

Configuring the RSVP Hello Interval

RSVP monitors the status of the interior gateway protocol (IGP) (Intermediate System-to-Intermediate System [IS-IS] or Open Shortest Path First [OSPF]) neighbors and relies on the IGP protocols to detect when a node fails. If an IGP protocol declares a neighbor down (because hello packets are no longer being received), RSVP also brings down that neighbor. However, the IGP protocols and RSVP still act independently when bringing a neighbor up.

Configuring a short time for the ISIS or OSPF hello timers allows these protocols to detect node failures more quickly. RSVP also benefits from early detection by the IGP protocols. It is not necessary to explicitly configure a short RSVP hello timer. If you do configure the RSVP hello timer, you can configure a longer value and can still expect the failure of a neighboring router to be quickly detected by IGP.

If all neighboring nodes support hello packets, you can reduce the refresh overhead (by increasing the value set in the **refresh-time** statement) without increasing the node or link failure detection time. The network can also scale to a larger number of sessions because the refresh operations consume less CPU time and bandwidth. For information about setting the refresh overhead, see “Configuring RSVP Timers” on page 296.

For more information about RSVP hello packets and timers, see “RSVP and IGP Hello Packets and Timers” on page 259.

By default, RSVP sends hello packets every 9 seconds. To modify how often RSVP 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.

Configuring RSVP Authentication

All RSVP protocol exchanges can be authenticated to guarantee that only trusted neighbors participate in setting up reservations. By default, RSVP authentication is disabled.

RSVP authentication uses an HMAC-MD5 message-based digest. This scheme produces a message digest based on a secret authentication key and the message contents. (The message contents also include a sequence number.) The computed digest is transmitted with RSVP messages. Once you have configured authentication, all received and transmitted RSVP messages with all neighbors are authenticated on this interface.

MD5 authentication provides protection against forgery and message modification. It also can prevent replay attacks. However, it does not provide confidentiality, because all messages are sent in clear text.

By default, authentication is disabled. To enable authentication, configure a key on each interface by including the `authentication-key` statement:

```
authentication-key key;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name*]

Configuring the Bandwidth Subscription for Class Types

By default, RSVP allows all of a class type's bandwidth (100 percent) to be used for RSVP reservations. When you oversubscribe a class type for a multiclass LSP, the aggregate demand of all RSVP sessions is allowed to exceed the actual capacity of the class type.

For detailed instructions on how to configure the bandwidth subscription for class types, see "Configuring the Bandwidth Subscription Percentage for LSPs" on page 142.

Configuring the RSVP Update Threshold on an Interface

The interior gateway protocols (IGPs) maintain the traffic engineering database (TED), but the current available bandwidth on the TED links originates from RSVP. When a link's bandwidth changes, RSVP informs the IGPs, which can then update the TED and forward the new bandwidth information to all network nodes. The network nodes then know how much bandwidth is available on the TED link (local or remote), and CSPF can correctly compute the paths.

However, IGP updates can consume excessive system resources. Depending on the number of nodes in a network, it might not be desirable to perform an IGP update for small changes in bandwidth. By configuring the `update-threshold` statement at the `[edit protocols rsvp]` hierarchy level, you can adjust the threshold at which a change in bandwidth triggers an IGP update.

You can configure a value of between 1 percent and 20 percent (the default is 10 percent) for when to trigger an IGP update. For example, if you have configured the `update-threshold` statement to be 15 percent and the router discovers that the bandwidth on a link has changed by 10 percent, RSVP does not trigger an IGP update. However, if the bandwidth on a link changes by 20 percent, RSVP does trigger an IGP update.

Include the `update-threshold` statement:

```
update-threshold percentage;
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp interface interface-name]`
- `[edit logical-routers logical-router-name protocols rsvp interface interface-name]`

Because of the update threshold, it is possible for Constrained Shortest Path First (CSPF) to compute a path using outdated TED bandwidth information on a link. If RSVP attempts to establish an LSP over that path, it might find that there is insufficient bandwidth on that link. When this happens, RSVP triggers an IGP TED update, flooding the updated bandwidth information on the network. CSPF can then recompute the path by using the updated bandwidth information, and attempt to find a different path, avoiding the congested link. Note that this functionality is the default and does not need any additional configuration.

You can configure the `rsvp-error-hold-time` statement at the `[edit protocols mpls]` hierarchy level or the `[edit logical-routers logical-router-name protocols mpls]` hierarchy level to improve the accuracy of the TED (including the accuracy of bandwidth estimates for LSPs) using information provided by PathErr messages. See “Improving TED Accuracy with RSVP PathErr Messages” on page 111.

Configuring Node Protection or Link Protection

When enabled, node protection or link protection provides bypass LSPs to the next-hop or next-next-hop routers for the LSPs traversing the configured router. To extend node protection or link protection along the entire path used by an LSP, node protection or link protection must be configured on each router that the LSP traverses.

To enable node protection or link protection, see the following sections:

- Configuring Node Protection or Link Protection on an LSP on page 286
- Configuring Link Protection on the Interfaces Used by the LSPs on page 287

Configuring Node Protection or Link Protection on an LSP

To enable node protection or link protection, configure the `node-link-protection` statement or the `link-protection` statement for each LSP that you want protected. You must also configure the `link-protection` statement on all unidirectional RSVP interfaces that the LSPs traverse.

To configure node protection for the LSP, include the `node-link-protection` statement:

```
node-link-protection;
```

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*]

To configure link protection for the LSP, include the `link-protection` statement:

```
link-protection;
```

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*]

Configuring Link Protection on the Interfaces Used by the LSPs

Whether you are configuring node protection or link protection, configure the `link-protection` statement on the RSVP interfaces used by the LSPs you configured in “Configuring Node Protection or Link Protection on an LSP” on page 286.

To configure link protection on the interfaces used by the LSPs, include the `link-protection` statement:

```
link-protection {
  disable;
  bandwidth bps;
  bypass bypass-name {
    bandwidth bps;
    hop-limit number;
    path address <strict | loose>;
    priority setup-priority reservation-priority;
    to address;
  }
  class-of-service cos-value;
  hop-limit number;
  max-bypasses number;
  no-node-protection;
  optimize-timer seconds;
  path address <strict | loose>;
  subscription percent {
    ct0 percent;
    ct1 percent;
    ct2 percent;
    ct3 percent;
  }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name*]

All the statements under `link-protection` are optional.

The following sections describe how to configure link protection:

- Configuring Bypass LSPs on page 288
- Configuring the Bandwidth for Bypass LSPs on page 289
- Configuring the Class of Service for Bypass LSPs on page 289
- Configuring the Hop Limit for Bypass LSPs on page 290
- Configuring the Maximum Number of Bypass LSPs on page 290
- Disabling Node Protection for Bypass LSPs on page 291
- Configuring the Optimization Interval for Bypass LSPs on page 291

- Configuring an Explicit Path for Bypass LSPs on page 291
- Configuring the Amount of Bandwidth Subscribed for Bypass LSPs on page 292
- Configuring Priority and Preemption for Bypass LSPs on page 293

Configuring Bypass LSPs

You can configure specific bandwidth and path constraints for a bypass LSP. You can also individually configure each bypass LSP generated when you enable multiple bypass LSPs. If you do not configure the bypass LSPs individually, they all share the same path and bandwidth constraints (if any).

If you specify the `bandwidth`, `hop-limit`, and `path` statements for the bypass LSP, these values take precedence over the values configured at the `[edit protocols rsvp interface interface-name link-protection]` hierarchy level. The other attributes (`subscription`, `no-node-protection`, and `optimize-timer`) are inherited from the general constraints.

To configure a bypass LSP, include the `bypass` statement:

```
bypass bypass-name {
  bandwidth bps;
  hop-limit number;
  path address <strict | loose>;
  priority setup-priority reservation-priority;
  to address;
}
```

You can include this statement at the following hierarchy levels:

- `[edit protocols rsvp interface interface-name link-protection]`
- `[edit logical-routers logical-router-name protocols rsvp interface interface-name link-protection]`

Configure the Next-Hop or Next-Next-Hop Node Address for Bypass LSPs

If you configure a bypass LSP, you must also configure the `to` statement. The `to` statement specifies the address for the interface of the immediate next-hop node (for link protection) or the next-next-hop node (for node-link protection). The address specified determines whether this is a link protection bypass or a node-link protection bypass. On multi-access networks (for example, a LAN), this address is also used to specify which next-hop node is being protected.

Configuring the Bandwidth for Bypass LSPs

You can specify the amount of bandwidth allocated for automatically generated bypass LSPs or you can individually specify the amount of bandwidth allocated for each LSP.

To specify the bandwidth allocation, include the `bandwidth` statement:

```
bandwidth bps;
```

For automatically generated bypass LSPs, include the `bandwidth` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

For individually configured bypass LSPs, include the `bandwidth` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]

If you have enabled multiple bypass LSPs, this statement is required. See also “Configuring the Maximum Number of Bypass LSPs” on page 290.

Configuring the Class of Service for Bypass LSPs

You can specify the class-of-service value for bypass LSPs by including the `class-of-service` statement:

```
class-of-service cos-value;
```

To apply a class-of-service value to all the automatically generated bypass LSPs, include the `class-of-service` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

To configure a class-of-service value for a specific bypass LSPs, include the `class-of-service` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]

Configuring the Hop Limit for Bypass LSPs

You can specify the maximum number of hops a bypass can traverse. By default, each bypass can traverse a maximum of 255 hops (the ingress and egress routers count as one hop each, so the minimum hop limit is 2).

To configure the hop limit for bypass LSPs, include the `hop-limit` statement:

```
hop-limit number;
```

For automatically generated bypass LSPs, include the `hop-limit` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

For individually configured bypass LSPs, include the `hop-limit` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]

Configuring the Maximum Number of Bypass LSPs

You can specify the maximum number of bypasses permitted for protecting an interface. When this statement is configured, multiple bypasses for link protection are enabled. Call admission control (CAC) is also enabled. By default, this option is disabled and only one bypass is enabled for each interface.

If you configure the `max-bypasses` statement, you must also configure the `bandwidth` statement.

To configure the maximum number of bypass LSPs for a protected interface, include the `max-bypasses` statement:

```
max-bypasses number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

Disabling Node Protection for Bypass LSPs

You can disable node protection on the RSVP interface. Link protection remains active. When this option is configured, the router can only initiate a next-hop bypass, not a next-next-hop bypass.

To disable node protection for bypass LSPs, include the `no-node-protection` statement:

```
no-node-protection;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

Configuring the Optimization Interval for Bypass LSPs

You can configure an optimization interval for bypass LSPs. At the end of this interval, an optimization process is initiated which attempts to either minimize the number of bypasses currently in use, minimize the total amount of bandwidth reserved for all of the bypasses, or both. You can configure an optimization interval from 1 through 65,535 seconds. A default value of 0 disables bypass LSP optimization.

To configure the optimization interval for bypass LSPs, include the `optimize-timer` statement:

```
optimize-timer seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

Configuring an Explicit Path for Bypass LSPs

By default, when you establish a bypass LSP to an adjacent neighbor, CSPF is used to discover the least-cost path. The `path` statement allows you to configure an explicit path (a sequence of strict or loose routes), giving you control over where and how the bypass LSP is established. To configure an explicit path, include the `path` statement:

```
path address <strict | loose>;
```

For automatically generated bypass LSPs, include the `path` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

For individually configured bypass LSPs, include the `path` statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]

Configuring the Amount of Bandwidth Subscribed for Bypass LSPs

You can configure the amount of bandwidth subscribed to bypass LSPs. You can configure the bandwidth subscription for the whole bypass LSP or for each class type that might traverse the bypass LSP. You can configure any value between 1 percent and 65,535 percent. By configuring a value less than 100 percent, you are undersubscribing the bypass LSPs. By configuring a value greater than 100 percent, you are oversubscribing the bypass LSPs.

The ability to oversubscribe the bandwidth for the bypass LSPs makes it possible to more efficiently use network resources. You can configure the bandwidth for the bypass LSPs based on the average network load as opposed to the peak load.

To configure the amount of bandwidth subscribed for bypass LSPs, include the `subscription` statement:

```
subscription percent {
  ct0 percent;
  ct1 percent;
  ct2 percent;
  ct3 percent;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection]

Configuring Priority and Preemption for Bypass LSPs

When there is insufficient bandwidth to establish a more important LSP, you might want to tear down a less important existing LSP to free up the bandwidth. You do this by preempting the existing LSP.

For more detailed information on configuring setup priority and reservation priority for LSPs, see “Configuring Priority and Preemption” on page 99.

To configure the bypass LSP’s priority and preemption properties, include the **priority** statement:

```
priority setup-priority reservation-priority;
```

You can configure this statement at the following hierarchy levels:

- [edit protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]
- [edit logical-routers *logical-router-name* protocols rsvp interface *interface-name* link-protection bypass *bypass-name*]

Configuring RSVP Graceful Restart

The following RSVP graceful restart configurations are possible:

- Graceful restart and helper mode are both enabled (the default).
- Graceful restart is enabled but helper mode is disabled. A router configured in this way can restart gracefully, but cannot help a neighbor with its restart and recovery procedures.
- Graceful restart is disabled but helper mode is enabled. A router configured in this way cannot restart gracefully, but can help a restarting neighbor.
- Graceful restart and helper mode both are disabled. This configuration completely disables RSVP graceful restart (including restart and recovery procedures and helper mode). The router behaves like a router that does not support RSVP graceful restart.

The following sections describe how to configure RSVP graceful restart:

- Enabling Graceful Restart on the Router on page 294
- Disabling Graceful Restart for RSVP on page 294
- Disabling RSVP Helper Mode on page 294
- Configuring the Maximum Helper Recovery Time on page 294
- Configuring the Maximum Helper Restart Time on page 294

Enabling Graceful Restart on the Router

To enable graceful restart for RSVP, you need to enable graceful restart for all the protocols that support graceful restart on the router. For more information about graceful restart, see the *JUNOS Routing Protocols Configuration Guide*.

To enable graceful restart on the router, 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]

Disabling Graceful Restart for RSVP

By default, RSVP graceful restart and RSVP helper mode are enabled when you enable graceful restart. However, you can disable one or both of these capabilities.

To disable RSVP graceful restart and recovery, include the `disable` statement at the [edit protocols rsvp graceful-restart] hierarchy level:

```
disable;
```

Disabling RSVP Helper Mode

To disable RSVP helper mode, include the `helper-disable` statement at the [edit protocols rsvp graceful-restart] hierarchy level:

```
helper-disable;
```

Configuring the Maximum Helper Recovery Time

To configure the amount of time the router retains the state of its RSVP neighbors while they undergo a graceful restart, include the `maximum-helper-recovery-time` statement at the [edit protocols rsvp graceful-restart] hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to recover.

```
maximum-helper-recovery-time seconds;
```

Configuring the Maximum Helper Restart Time

To configure the delay between when the router discovers that a neighboring router has gone down and when it declares the neighbor down, include the `maximum-helper-restart-time` statement at the [edit protocols rsvp graceful-restart] hierarchy level. This value is applied to all neighboring routers, so it should be based on the time required by the slowest RSVP neighbor to restart.

```
maximum-helper-restart-time seconds;
```

Configuring RSVP LSP Load Balancing

By default, when you have configured several RSVP LSPs to the same egress router, the LSP with the lowest metric is selected and carries all traffic. If all of the LSPs have the same metric, one of the LSPs is selected at random and all traffic is forwarded over it.

Alternatively, you can load balance traffic across all of the LSPs by enabling per-packet load balancing.

To enable per-packet load balancing on an ingress LSP, configure the `policy-statement` statement as follows:

```
[edit policy-options]
policy-statement policy-name {
  then {
    load-balance per-packet;
  }
  accept;
}
```

You then need to apply this statement as an export policy to the forwarding table. For more information on how to configure the `policy-statement` statement, see the *JUNOS Policy Framework Configuration Guide*.

Once per-packet load balancing is applied, traffic is distributed equally between the LSPs (by default).

You can also load-balance the traffic between the LSPs in proportion to the amount of bandwidth configured for each LSP. This capability can better distribute traffic in networks with asymmetric bandwidth capabilities across external links, since the configured bandwidth of an LSP typically reflects the traffic capacity of that LSP.

To configure RSVP LSP load balancing, include the `load-balance` statement with the `bandwidth` option:

```
load-balance {
  bandwidth;
}
```

You can configure this statement at the following hierarchy levels:

- [edit logical-routers *logical-router-name* protocols rsvp],
- [edit protocols rsvp]

The following provides further information regarding for `load-balance` statement:

- If you configure the `load-balance` statement, the behavior of currently running LSPs is not altered. To force currently running LSPs to use the new behavior, you can issue a `clear mpls lsp` command.
- The `load-balance` statement only applies to ingress LSPs which have per-packet load balancing enabled.
- For differentiated services aware traffic engineered LSPs, the bandwidth of an LSP is calculated by summing the bandwidth of all of the class types.

Configuring RSVP Timers

RSVP uses two related timing parameters:

- `refresh-time`—The refresh time controls the interval between the generation of successive refresh messages. The default value for the refresh time is 45 seconds. This number is derived from the `refresh-time` statement's default value of 30, multiplied by a fixed value of 1.5. This computation differs from RFC 2205, which states that the refresh time should be multiplied by a random value in the range of 0.5 through 1.5.

Refresh messages include path and Resv messages. Refresh messages are sent periodically so that reservation states in neighboring nodes do not time out. Each path and Resv message carries the refresh timer value, and the receiving node extracts this value from the messages.

- `keep-multiplier`—The keep multiplier is a small, locally configured integer from 1 through 255. The default value is 3. It indicates the number of messages that can be lost before a particular state is declared stale and must be deleted. The keep multiplier directly affects the lifetime of an RSVP state.

To determine the lifetime of a reservation state, use the following formula:

$$lifetime = (keep-multiplier + 0.5) \times (1.5 \times refresh-time)$$

In the worst case, $(keep-multiplier - 1)$ successive refresh messages must be lost before a reservation state is deleted.

By default, the refresh timer value is 30 seconds. To modify this value, include the `refresh-time` statement:

```
refresh-time seconds;
```

You can include this statement at the following hierarchy levels:

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

The default value of the keep multiplier is 3. To modify this value, include the `keep-multiplier` statement:

```
keep-multiplier number;
```

You can include this statement at the following hierarchy levels:

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

Preempting RSVP Sessions

Whenever bandwidth is insufficient to handle all RSVP sessions, you can control the preemption of RSVP sessions. By default, an RSVP session is preempted only by a new higher-priority session.

To always preempt a session when the bandwidth is insufficient, include the `preemption` statement with the `aggressive` option:

```
preemption aggressive;
```

You can include this statement at the following hierarchy levels:

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

To disable RSVP session preemption, include the `preemption` statement with the `disabled` option:

```
preemption disabled;
```

To return to the default (that is, preempt a session only for a new higher-priority session), include the `preemption` statement with the `normal` option:

```
preemption normal;
```

You can include this statement at the following hierarchy levels:

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

Configuring MTU Signaling in RSVP

To configure maximum transmission unit (MTU) signaling in RSVP, you need to configure MPLS to allow IP packets to be fragmented before they are encapsulated in MPLS. You also need to configure MTU signaling in RSVP. For troubleshooting purposes, you can configure MTU signaling alone without enabling packet fragmentation.

To configure MTU signaling in RSVP, include the `path-mtu` statement:

```
path-mtu {
  allow-fragmentation;
  rsvp {
    mtu-signaling;
  }
}
```

You can include this statement at the following hierarchy levels:

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

The following sections describe how to enable packet fragmentation and MTU signaling in RSVP:

- Enabling MTU Signaling in RSVP on page 298
- Enabling Packet Fragmentation on page 299

Enabling MTU Signaling in RSVP

To enable MTU signaling in RSVP, include the `rsvp mtu-signaling` statement:

```
rsvp mtu-signaling;
```

You can include this statement at the following hierarchy levels:

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

Once you have committed the configuration, changes in the MTU signaling behavior for RSVP take effect the next time the path is refreshed.

You can configure the `mtu-signaling` statement by itself at the [edit protocols mpls path-mtu rsvp] hierarchy level. This can be useful for troubleshooting. If you configure just the `mtu-signaling` statement, you can use the `show rsvp session detail` command to determine what the smallest MTU is on an LSP. The `show rsvp session detail` command displays the MTU value received and sent in the `adspec` object.

Enabling Packet Fragmentation

To allow IP packets to be fragmented before they are encapsulated in MPLS, include the `allow-fragmentation` statement:

```
allow-fragmentation;
```

You can include this statement at the following hierarchy levels:

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



NOTE: Do not configure the `allow-fragmentation` statement alone. Always configure it in conjunction with the `mtu-signaling` statement.

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

You can control the label value advertised on the egress router of an 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 (IP version 4 [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 for RSVP, include the `explicit-null` statement:

```
explicit-null;
```

You can include this statement 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.

Tracing RSVP Protocol Traffic

To trace RSVP protocol traffic, include the `traceoptions` statement:

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

You can include this statement at the following hierarchy levels:

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

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 RSVP tracing output in the file `rsvp-log`.

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

- `all`—All tracing operations.
- `error`—All detected error conditions
- `event`—RSVP-related events (helps to trace events related to RSVP graceful restart)
- `lmp`—RSVP-Link Management Protocol (LMP) interactions
- `packets`—All RSVP packets
- `path`—All path messages
- `pathtear`—PathTear messages
- `resv`—Resv messages
- `resvtear`—ResvTear messages
- `route`—Routing information
- `state`—Session state transitions

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

Examples: Tracing RSVP Protocol Traffic

Trace RSVP path messages in detail:

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

Trace all RSVP messages:

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

Trace all RSVP error conditions:

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

