

Chapter 6

Extended Actions Configuration

This chapter describes how to configure extended actions for a routing policy. These extended actions include criteria that manipulate the route characteristics and are more complex than standard actions. You should have an in-depth understanding of extended actions before configuring them. For a complete list of the actions that manipulate route characteristics, see Table 11 on page 49.

This chapter provides information about the following routing policy actions configuration tasks:

Configuring AS Path Prepend Action on page 132

Configuring AS Path Expand Action on page 132

Configuring Class Action on page 133

Configuring Damping Action on page 133

Configuring Load-Balance Per-Packet Action on page 138

Configuring AS Path Prepend Action

You can *prepend* or add one or more autonomous system (AS) numbers at the beginning of an AS path. The AS numbers are added after the local AS number has been added to the path. Prepending an AS path makes a shorter AS path look longer and therefore less preferable to the Border Gateway Protocol (BGP).

For example, from AS 1, there are two equal paths (through AS 2 and AS 3) to reach AS 4. You might want packets from certain sources to use the path through AS 2. Therefore, you must make the path through AS 3 look less preferable so that BGP chooses the path through AS 2. In AS 1, you can prepend multiple AS numbers:

```
[edit]
policy-options {
  policy-statement as-path-prepend {
    term prepend {
      from {
        route-filter 192.168.0.0/16 orlonger;
        route-filter 172.16.0.0/12 orlonger;
        route-filter 10.0.0.0/8 orlonger;
      }
      then as-path-prepend "1 1 1 1";
    }
  }
}
```

Configuring AS Path Expand Action

You can *expand* or add one or more AS numbers to an AS sequence. The AS numbers are added before the local AS number has been added to the path. Expanding an AS path makes a shorter AS path look longer and therefore less preferable to BGP. The last AS number in the existing path is extracted and prepended *n* times, where *n* is a number from 1 through 32. This is similar to the AS path prepend action, except that the AS path expand action adds an arbitrary sequence of AS numbers.

For example, from AS 1, there are two equal paths (through AS 2 and AS 3) to reach AS 4. You might want packets from certain sources to use the path through AS 2. Therefore, you must make the path through AS 3 look less preferable so that BGP chooses the path through AS 2. In AS 1, you can expand multiple AS numbers.

```
[edit]
policy-options {
  policy-statement as-path-expand {
    term expand {
      from {
        route-filter 192.168.0.0/16 orlonger;
        route-filter 172.16.0.0/12 orlonger;
        route-filter 10.0.0.0/8 orlonger;
      }
      then as-path-expand last-as count 4;
    }
  }
}
```

For routes from AS 2, this makes the route look like 1 2 2 2 2 when advertised, where 1 is from AS 1, the 2 from AS 2 is prepended 4 times, and the final 2 is the original 2 received from the neighbor router.

Configuring Class Action

For information about class of service (CoS), see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

Configuring Damping Action

BGP *route flapping* describes the situation in which BGP systems send an excessive number of update messages to advertise network reachability information. BGP *flap damping* is a way to reduce the number of update messages sent between BGP peers, thereby reducing the load on these peers without adversely affecting the route convergence time.

Flap damping reduces the number of update messages by marking routes as ineligible for selection as the active or preferable route. Doing this leads to some delay, or *suppression*, in the propagation of route information, but the result is increased network stability. You typically apply flap damping to external BGP (EBGP) routes (that is, to routes in different ASs). You can also apply it within a confederation, between confederation member ASs. Because routing consistency within an AS is important, do not apply flap damping to internal BGP (IBGP) routes. (If you do, it is ignored.)

BGP flap damping is defined in RFC 2439, *BGP Route Flap Damping*.

To effect changes to the default BGP flap damping values, you define actions by creating a named set of damping parameters and including it in a routing policy with the damping action (described in Table 11 on page 49). For the damping routing policy to work, you also must enable BGP route flap damping.

The following sections discuss the following topics:

Configuring Flap Damping Parameters on page 134

Defining Damping Action on page 136

Enabling BGP Route Flap Damping on page 136

Disabling Damping by Prefix on page 136

Example: Configuring BGP Flap Damping on page 137

Configuring Flap Damping Parameters

To define damping parameters, include the damping statement:

```
damping name {
  disable;
  half-life minutes;
  max-suppress minutes;
  reuse number;
  suppress number;
}
```

You can include this statement at the following hierarchy levels:

[edit policy-options]

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

The name identifies the group of damping parameters. It can contain letters, numbers, and hyphens (-) and can be up to 255 characters. To include spaces in the name, enclose the entire name in quotation marks (double quotes).

You can specify one or more of the damping parameters described in Table 21.

Table 21: Damping Parameters

| Damping Parameter | Description | Default | Possible Values |
|-----------------------------|------------------------------------|-----------------|-----------------------------|
| half-life <i>minutes</i> | Decay half-life, in minutes | 15 minutes | 1 through 45 minutes |
| max-suppress <i>minutes</i> | Maximum hold-down time, in minutes | 60 minutes | 1 through 720 minutes |
| reuse | Reuse threshold | 750 (unitless) | 1 through 20,000 (unitless) |
| suppress | Cutoff (suppression) threshold | 3000 (unitless) | 1 through 20,000 (unitless) |

If you do not specify one or more of the damping parameters, the default value of the parameter is used.

To understand how to configure these parameters, you need to understand how damping suppresses routes. How long a route can be suppressed is based on a *figure of merit*, which is a value that correlates to the probability of future instability of a route. Routes with higher figure-of-merit values are suppressed for longer periods of time. The figure-of-merit value decays exponentially over time.

A figure-of-merit value of zero is assigned to each new route. The value is increased each time the route is withdrawn or readvertised, or when one of its path attributes changes. With each incident of instability, the value increases as follows:

Route is withdrawn—1000

Route is readvertised—1000

Route's path attributes change—500

When a route's figure-of-merit value reaches a particular level, called the *cutoff* or *suppression threshold*, the route is suppressed. If a route is suppressed, the routing table no longer installs the route into the forwarding table and no longer exports this route to any of the routing protocols. By default, a route is suppressed when its figure-of-merit value reaches 3000. To modify this default, include the `suppress` option at the [edit policy-options damping *name*] hierarchy level.

If a route has flapped, but then becomes stable so that none of the incidents listed above occur within a configurable amount of time, the figure-of-merit value for the route decays exponentially. The default half-life is 15 minutes. For example, for a route with a figure-of-merit value of 1500, if no incidents occur, its figure-of-merit value is reduced to 750 after 15 minutes and to 375 after another 15 minutes. To modify the default half-life, include the `half-life` option at the [edit policy-options damping *name*] hierarchy level.

A suppressed route becomes reusable when its figure-of-merit value decays to a value below a *reuse threshold*, thus allowing routes that experience transient instability to once again be considered valid. The default reuse threshold is 750. When the figure-of-merit value passes below the reuse threshold, the route once again is considered usable and can be installed in the forwarding table and exported from the routing table. To modify the default reuse threshold, include the `reuse` option at the [edit policy-options damping *name*] hierarchy level.

The maximum suppression time provides an upper bound on the time that a route can remain suppressed. The default maximum suppression time is 60 minutes. To modify the default, include the `max-suppress` option at the [edit policy-options damping *name*] hierarchy level.

A route's figure-of-merit value stops increasing when it reaches a maximum suppression threshold, which is determined based on the route's suppression threshold level, half-life, reuse threshold, and maximum hold-down time.

The merit ceiling, ϵ_c , which is the maximum merit that a flapping route can collect, is calculated using the following formula:

$$\epsilon_c \leq \epsilon_r 2^{(t/\lambda) (\ln 2)}$$

ϵ_r is the figure-of-merit reuse threshold, t is the maximum hold-down time in minutes, and λ is the half-life in minutes. For example, if you use the default figure-of-merit values in this formula, but use a half-life of 30 minutes, the calculation is as follows:

$$\begin{aligned} \epsilon_c &\leq 750 2^{(60/30) (\ln 2)} \\ \epsilon_c &\leq 3000 \end{aligned}$$



NOTE: The cutoff threshold, which you configure using the `suppress` option, must be less than or equal to the merit ceiling, ϵ_c . If the configured cutoff threshold or the default cutoff threshold is greater than the merit ceiling, the route is never suppressed and damping never occurs.

To display figure-of-merit information, use the `show policy damping` command.

A route that has been assigned a figure of merit is considered to have a damping state. To display the current damping information on the router, use the `show route detail` command.

Defining Damping Action

To define the damping action, include the damping action and the name of the configured damping parameters either in a `then` statement or in a `route-filter` option in a `from` statement.

Enabling BGP Route Flap Damping

For information about enabling BGP route flap damping, see the *JUNOS Routing Protocols Configuration Guide*.

Disabling Damping by Prefix

Normally, you enable or disable damping on a per-peer basis. However, you can disable damping for a specific prefix received from a peer by including the `disable` option:

```
disable;
```

You can include this statement at the following hierarchy levels:

```
[edit policy-options damping name]
```

```
[edit logical-routers logical-router-name policy-options damping name]
```

Example: Disabling by Prefix

In this routing policy example, although damping is enabled for the peer, the `damping none` statement specifies that damping be disabled for prefix `3.0.0.0/8` in Policy-A. This route is not damped because the routing policy statement named Policy-A filters on the prefix `3.0.0.0/8` and the action points to the damping statement named `none`. The remaining prefixes are damped using the default parameters.

```
[edit]
policy-options {
  policy-statement Policy-A {
    from {
      route-filter 10.0.0.0/8 exact;
    }
    then damping none;
  }
  damping none {
    disable;
  }
}
```

Example: Configuring BGP Flap Damping

Enable BGP flap damping and configure damping parameters:

```
[edit]
routing-options {
  autonomous-system 666;
}
protocols {
  bgp {
    damping;
    group group1 {
      traceoptions {
        file bgp-log size 1m files 10;
        flag damping;
      }
      import damp;
      type external;
      peer-as 10458;
      neighbor 192.168.2.30;
    }
  }
}
policy-options {
  policy-statement damp {
    from {
      route-filter 192.168.0.0/32 exact {
        damping high;
        accept;
      }
      route-filter 172.16.0.0/32 exact {
        damping medium;
        accept;
      }
      route-filter 10.0.0.0/8 exact {
        damping none;
        accept;
      }
    }
  }
  damping high {
    half-life 30;
    suppress 3000;
    reuse 750;
    max-suppress 60;
  }
  damping medium {
    half-life 15;
    suppress 3000;
    reuse 750;
    max-suppress 45;
  }
  damping none {
    disable;
  }
}
```

To display damping parameters for this configuration, use the `show policy damping` command:

```
user@host> show policy damping
Damping information for "high":
  Halflife: 30 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 60 minutes
  Computed values:
    Merit ceiling: 3008
    Maximum decay: 24933
Damping information for "medium":
  Halflife: 15 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 45 minutes
  Computed values:
    Merit ceiling: 6024
    Maximum decay: 12449
Damping information for "none":
  Damping disabled
```

Configuring Load-Balance Per-Packet Action

By default, when there are multiple equal-cost paths to the same destination for the active route, the JUNOS software uses a hash algorithm to choose one of the next-hop addresses to install into the forwarding table. Whenever the set of next hops for a destination changes in any way, the next-hop address is rechosen, also at using the hash algorithm.

You can configure the JUNOS software so that, for the active route, all next-hop addresses for a destination are installed in the forwarding table. This feature is called per-packet load balancing. You can use load balancing to spread traffic across multiple paths between routers. The behavior of the load-balance per-packet function depends on the version of the Internet Processor application-specific integrated circuit (ASIC) in your routing platform.

On routing platforms with an Internet Processor ASIC, when per-packet load balancing is configured, traffic between routers with multiple paths is spread using the hash algorithm across the available interfaces. The forwarding table balances the traffic headed to a destination, transmitting it in round-robin fashion among the multiple next hops (up to a maximum of eight equal-cost load-balanced paths). The traffic is load-balanced on a per-packet basis.

On routing platforms with the Internet Processor II ASIC, when per-packet load balancing is configured, traffic between routers with multiple paths is divided into individual traffic flows (up to a maximum of 16 equal-cost load-balanced paths). Packets for each individual flow are kept on a single interface.

To configure the load-balance per-packet action, include the `load-balance per-packet` action in a `then` statement or a `route-filter` option in a `from` statement in a routing policy.

You must apply the routing policy to routes exported from the routing table to the forwarding table to complete the configuration. To do this, include the `export` statement.

You must apply the routing policy to routes exported from the routing table to the forwarding table, include the export statement:

```
[edit routing-options forwarding-table]
export;
```

You can include this statement at the following hierarchy levels:

```
[edit routing-options forwarding-table]

[edit routing-instances routing-instance-name routing-options forwarding-table]

[edit logical-routers logical-router-name routing-options forwarding-table]

[edit logical-routers logical-router-name routing-instances routing-instance-name
routing-options forwarding-table]
```

You can also configure additional information about flows, such as source and destination port number and ingress interface information, to further identify them. By default, the software ignores port data when determining flows. To enable per-flow load balancing, you must set the `load-balance per-packet` action in the routing policy configuration; for more information about this action, see “Routing Policy Configuration” on page 41.

To include port data in the flow determination, include the `family inet` statement at the `[edit forwarding-options hash-key]` hierarchy level:

```
[edit forwarding-options hash-key]
family inet {
    layer-3;
    layer-4;
}
```

You must include the `layer-3` statement. If you omit the `layer-3` statement, the management process removes the `hash-key` statement from the configuration and the router behaves as if you specified `layer-3`.

If you include both the `layer-3` and `layer-4` statements, the router uses the following Layer 3 and Layer 4 information to load-balance:

- Source IP address
- Destination IP address
- Protocol
- Source port number
- Destination port number
- Incoming interface index

The router recognizes packets in which all of these layer-3 and layer-4 parameters are identical, and ensures that these packets are sent out through the same interface. This prevents problems that might otherwise occur with packets arriving at their destination out of their original sequence.

This is appropriate behavior for Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) packets. For Internet Control Message Protocol (ICMP) packets, the field location offset is the checksum field, which makes each ping packet a separate “flow.” There are other protocols that can be encapsulated in IP that may have a varying value in the 32-bit offset. This may also be problematic because they are seen as a separate flow.

By default, or if you specify only the layer-3 statement, the router uses the incoming interface index as well as the following Layer 3 information in the packet header to load balance:

- Source IP address
- Destination IP address
- Protocol

The following sections discuss the following topics:

- Load Balancing Based on the MPLS Label Information on page 140

- Examples: Configuring Per-Packet Load Balancing on page 141

Load Balancing Based on the MPLS Label Information

To load-balance based on the Multiprotocol Label Switching (MPLS) label information, include the family mpls statement at the [edit forwarding-options hash-key] hierarchy level:

```
family mpls {
  label-1;
  label-2;
  payload {
    ip;
  }
}
```

This is for aggregated Ethernet and aggregated SONET/SDH interfaces, as well as for multiple equal-cost MPLS next hops.

To include the first label in the hash key, include the label-1 option. This is used for a one-label packet.

To include the first and second label in the hash key, include both the label-1 and label-2 options. This is used for a two-label packet. The router provides hashing on the first and second labels by default. If both labels are specified on an M-series router, the entire first label and the first 16 bits of the second label are hashed. If both labels are specified on a T-series routing platform, the first 16 bits of the first label and the first 16 bits of the second label are hashed.

Hashing may include IP addresses to provide better distribution of traffic to aggregated interfaces.

To include the bits in the IP address of the IPv4 or IPv6 payload as well as the first label in the hash key, include the family mpls statement at the [edit forwarding-options hash-key] hierarchy level:

```
[edit forwarding-options hash-key]
family mpls {
  label-1;
  payload {
    ip;
  }
}
```

(For T-series routing platforms only) To include the bits of the IP address of the IPv4 or IPv6 payload as well as both the first label and the second label in the hash key, include the family mpls statement at the [edit forwarding-options hash-key family mpls] hierarchy level:

```
[edit forwarding-options hash-key family mpls]
family mpls {
  label-1;
  label-2;
  payload {
    ip;
  }
}
```

Examples: Configuring Per-Packet Load Balancing

Perform per-packet load balancing for all routes:

```
[edit]
policy-options {
  policy-statement load-balancing-policy {
    then {
      load-balance per-packet;
    }
  }
}
routing-options {
  forwarding-table {
    export load-balancing-policy;
  }
}
```

Perform per-packet load balancing only for a limited set of routes:

```
[edit]
policy-options {
  policy-statement load-balancing-policy {
    from {
      route-filter 192.168.10/24 orlonger;
      route-filter 10.114/16 orlonger;
    }
    then {
      load-balance per-packet;
    }
  }
}
routing-options {
  forwarding-table {
    export load-balancing-policy;
  }
}
```