

# Chapter 1

## Routing Protocols Concepts

The JUNOS routing protocol process supports a wide variety of routing protocols, including IS-IS, OSPF, RIP, RIPng, and BGP. This chapter explains the general terminology and concepts related to configuring and using the routing protocol process and the routing protocols. For information about configuring the individual routing protocols, see the individual chapter about that protocol.

This chapter discusses the following topics:

- Routing Databases on page 3
- Configure Interfaces on page 6
- Route Preferences on page 6
- Equal-Cost Paths and Load-Sharing on page 10

### Routing Databases

The JUNOS software maintains two databases for routing information:

- Routing table—Contains all the routing information learned by all routing protocols.
- Forwarding table—Contains the routes actually used to forward packets through the router.

In addition, the interior gateway protocols (IGPs), IS-IS and OSPF maintain link-state databases.

This section includes the following topics:

- Routing Protocol Databases on page 4
- JUNOS Routing Tables on page 4
- Forwarding Tables on page 5
- How the Routing and Forwarding Tables Are Synchronized on page 5

## Routing Protocol Databases

Each IGP routing protocol maintains a database of the routing information it has learned from other routers running the same protocol and uses this information as defined and required by the protocol. IS-IS and OSPF use the routing information they received to maintain link-state databases, which they use to determine which adjacent neighbors are operational and to construct network topology maps.

IS-IS and OSPF use the Dijkstra algorithm, and RIP and RIPng use the Bellman-Ford algorithm to determine the best route or routes (if there are multiple equal-cost routes) to reach each destination and installs these routes into the JUNOS software routing table.

## JUNOS Routing Tables

The JUNOS software routing table is used by the routing protocol process to maintain its database of routing information. In this table, the routing protocol process stores statically configured routes, directly connected interfaces (also called *direct routes* or *interface routes*), and all routing information learned from all routing protocols. The routing protocol process uses this collected routing information to select the *active route* to each destination, which is the route that actually is used to forward packets to that destination.

By default, the JUNOS software maintains three routing tables: one for unicast routes, another for multicast routes, and a third for Multiprotocol Label Switching (MPLS). You can configure additional routing tables to support situations where you need to separate a particular group of routes or where you need greater flexibility in manipulating routing information. In general, most operations can be performed without resorting to the complexity of additional routing tables. However, creating additional routing tables has several specific uses, including importing interface routes into more than one routing table, applying different routing policies when exporting the same route to different peers, and providing greater flexibility with incongruent multicast topologies.

Each routing table is identified by a name, which consists of the protocol family followed by a period and small, nonnegative integer. The protocol family can be *inet* (Internet), *iso* (ISO), or *mpls* (MPLS). The following names are reserved for the default routing tables maintained by the JUNOS software:

`inet.0`—Default Internet Protocol Version 4 (IPv4) unicast routing table

`inet6.0`—Default Internet Protocol Version 6 (IPv6) unicast routing table

`instance-name.inet.0`—Unicast routing table for a particular routing instance

`inet.1`—Multicast forwarding cache

`inet.2`—Unicast routes used for multicast reverse path forwarding (RPF) lookup

inet.3—MPLS routing table for path information

mpls.0—MPLS routing table for label-switched path (LSP) next hops



**Note**

For clarity, this manual contains general discussions of routing tables as if there were only one table. However, when it is necessary to distinguish among the routing tables, their names are explicitly used.

## ***Forwarding Tables***

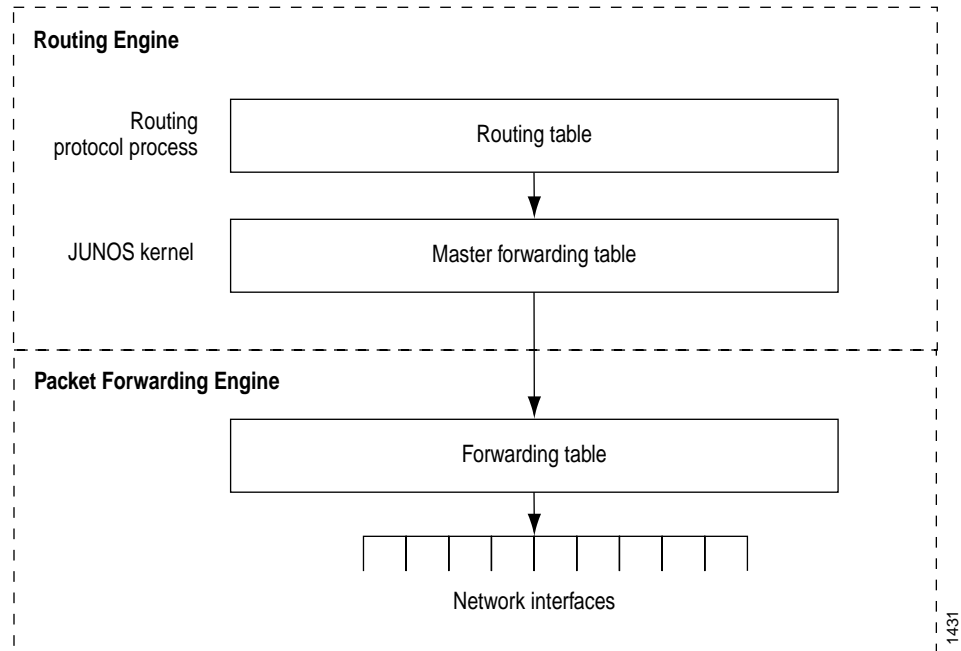
The JUNOS software installs all active routes from the routing table into the forwarding table. The active routes are routes that are used to forward packets to their destinations.

The JUNOS kernel maintains a master copy of the forwarding table. It copies the forwarding table to the Packet Forwarding Engine, which is the part of the router responsible for forwarding packets.

## ***How the Routing and Forwarding Tables Are Synchronized***

The JUNOS routing protocol process is responsible for synchronizing the routing information between the routing and forwarding tables. To do this, the routing protocol process calculates the active routes from all the routes in the routing table and installs them into the forwarding table. The routing protocol process then copies the forwarding table to the router's Packet Forwarding Engine, the part of the router that forwards packets. Figure 1 illustrates how the routing tables are synchronized.

Figure 1: Synchronizing Routing Exchange between the Routing and Forwarding Tables



## Configure Interfaces

When you configure a protocol on an interface, you must also configure a protocol family on that interface. For information about configuring interfaces, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*. For information about configuring protocol families, see the individual protocol configuration chapters in this book.

## Route Preferences

For unicast routes, the JUNOS routing protocol process uses the information in its routing table, along with the properties set in the configuration file, to choose an *active route* for each destination. While the JUNOS software might know of many routes to a destination, the active route is the preferred route to that destination and is the one that is installed in the forwarding table and used when actually routing packets.

The routing protocol process generally determines the active route by selecting the route with the lowest preference value. The preference is an arbitrary value in the range 0 through 255 that the software uses to rank routes received from different protocols, interfaces, or remote systems.

When routes in the routing table are nearly identical, the routing protocol process prefers the route whose next hop has the lowest IP address.

The preference value is used to select routes to destinations in external ASs or routing domains; it has no effect on the selection of routes within an AS (that is, within an IGP). Routes within an AS are selected by the IGP and are based on that protocol's metric or cost value.

This section includes the following topics:

Alternate and Tiebreaker Preferences on page 7

How the Active Route Is Determined on page 7

Multiple Active Routes on page 9

Default Route Preference Values on page 9

## ***Alternate and Tiebreaker Preferences***

The JUNOS software provides support for alternate and tiebreaker preferences, and some of the routing protocols, including BGP and label switching, use these additional preferences. With these protocols, you can specify a primary route preference, preference, and a secondary preference, preference2, that is used as a tiebreaker. You can also mark route preferences with additional route tiebreaker information by specifying a color, color, and a tiebreaker color, color2.

The software uses a four-byte value to represent the route preference value. When using the preference value to select an active route, the software first compares the primary route preference values, choosing the route with the lowest value. If there is a tie and if a secondary preference has been configured, the software compares the secondary preference values, choosing the route with the lowest value.

## ***How the Active Route Is Determined***

For each prefix in the routing table, the routing protocol process selects a single best path, called the active route. The algorithm for determining the active route is as follows:

1. Choose the path with the lowest preference value (routing protocol process preference). Routes that are not eligible to be used for forwarding (for example, because they were rejected by routing policy or because a next hop is inaccessible) have a preference of -1 and are never chosen.
2. For BGP, prefer the path with higher local preference. For non-BGP paths, choose the path with the lowest preference2 value.

3. If the path includes an AS path:

- a. Prefer the route with a shorter AS path.

Confederation sequences are considered to have a path length of 0, and AS and confederation sets are considered to have a path length of 1.

- b. Prefer the route with the lower origin code. Routes learned from an IGP have a lower origin code than those learned from an EGP, and both have lower origin codes than incomplete routes (routes whose origin is unknown).

- c. Depending on whether nondeterministic routing table path selection behavior is configured, there are two possible cases:

If nondeterministic routing table path selection behavior is not configured (that is, if the path-selection `cisco-nondeterministic` statement is not included in the BGP configuration), for paths with the same neighboring AS numbers at the front of the AS path, prefer the path with the lowest multiple exit discriminator (MED) metric. Confederation AS numbers are not considered when deciding what the neighbor AS number is. When you display the routes in the routing table using the `show route` command, they generally appear in order from most preferred to least preferred. Routes that share the same neighbor AS are grouped together in the command output. Within a group, the best route is listed first and the other routes are marked with the `NotBest` flag in the `State` field of the `show route detail` command.

To always compare MEDs whether or not the peer ASs of the compared routes are the same, use the `path-selection (always-compare-med)` statement. For an example, see “Configure Routing Table Path Selection” on page 394.

If nondeterministic routing table path selection behavior is configured (that is, the `path-selection cisco-nondeterministic` statement is included in the BGP configuration), prefer the path with the lowest MED metric. When you display the routes in the routing table using the `show route` command, they generally appear in order from most preferred to least preferred and are ordered with the best route first, followed by all other routes in order from newest to oldest.

In both cases, confederations are not considered when determining neighboring ASs. Also, in both cases, a missing metric is treated as if a MED were present but zero.

4. Prefer strictly internal paths, which include IGP routes and locally generated routes (static, direct, local, and so forth).

5. Prefer strictly external (EBGP) paths over external paths learned through interior sessions (IBGP).

6. For BGP, prefer the path whose next hop is resolved through the IGP route with the lowest metric.

7. For BGP, prefer the path whose BGP next hop is resolved through the IGP route with the largest number of next hops.

8. For BGP, prefer the route with the shortest route reflection cluster list. Routes without a cluster list are considered to have a cluster list of length 0.
9. For BGP, prefer the route with the lowest IP address value for the BGP router ID.
10. Prefer the path that was learned from the neighbor with the lowest peer IP address.

## Multiple Active Routes

The interior gateway protocols (IGPs) compute equal-cost multipath next hops, and internal BGP (IBGP) picks up these next hops. When there are multiple, equal-cost next hops associated with a route, the routing protocol process installs only one of the next hops in the forwarding path with each route, randomly selecting which next hop to install. For example, if there are 3 equal-cost paths to an exit router and 900 routes leaving through that router, each of the paths ends up with about 300 routes pointing at it. This mechanism provides load distribution among the paths while maintaining packet ordering per destination.

## Default Route Preference Values

The JUNOS software routing protocol process assigns a default preference value to each route that the routing table receives. The default value depends on the source of the route. The preference is a value from 0 through 255, with a lower value indicating a more preferred route. Table 2 lists the default preference values.

Table 2: Default Route Preference Values

How Route Is Learned	Default Preference	Statement to Modify Default Preference
Directly connected network	0	—
System routes	4	—
Static	5	static on page 125
MPLS	7	MPLS preference in the <i>JUNOS Internet Software Configuration Guide: MPLS Applications</i>
LDF	8	LDF preference in the <i>JUNOS Internet Software Configuration Guide: MPLS Applications</i>
LDP	9	LDP preference in the <i>JUNOS Internet Software Configuration Guide: MPLS Applications</i>
OSPF internal route	10	OSPF export on page 265
IS-IS Level 1 internal route	15	IS-IS external-preference on page 210, preference on page 222
IS-IS Level 2 internal route	18	IS-IS external-preference on page 210, preference on page 222
Default	20	—
Redirects	30	—
Kernel	40	—
SNMP	50	—
Router Discovery	55	—
RIP	100	RIP preference on page 303
RIPng	100	RIPng preference on page 322
PIM	105	<i>JUNOS Internet Software Configuration Guide: Multicast</i>
DVMRP	110	<i>JUNOS Internet Software Configuration Guide: Multicast</i>

How Route Is Learned	Default Preference	Statement to Modify Default Preference
Routes to interfaces that are down	120	—
Aggregate	130	aggregate on page 94
OSPF AS external routes	150	OSPF external-preference on page 266, preference on page 275
IS-IS Level 1 external route	160	IS-IS external-preference on page 210, preference on page 222
IS-IS Level 2 external route	165	IS-IS external-preference on page 210, preference on page 222
BGP	170	BGP preference on page 439, export on page 422, import on page 427
MSDP	175	<i>JUNOS Internet Software Configuration Guide: Multicast</i>

In general, the narrower the scope of the statement, the higher precedence its preference value is given, but the smaller the set of routes it affects. To modify the default preference value for routes learned by routing protocols, you generally apply routing policy when configuring the individual routing protocols. You also can modify some preferences with other configuration statements, which are indicated in the table. For information about defining and applying routing policies, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

## Equal-Cost Paths and Load-Sharing

For equal-cost paths, load-sharing is based on the BGP next hop. For example, if four prefixes all point to a next hop and there is more than one equal-cost path to that next hop, the routing protocol process randomizes the path chosen among the four prefixes. Also, for each prefix, the routing protocol process installs a single forwarding entry pointing along one of the paths. The routing software does not rerandomize the path taken as prefixes pointing to the next hop come and go, but it does rerandomize if the number of paths to the next hop changes. Because a prefix is tied to a particular path, packet reordering should not happen. The degree of load sharing improves as the number of prefixes increases.

## IPv6

Internet Protocol version 6 (IPv6) is the new version of the Internet Protocol (IP). The Internet Protocol allows numerous nodes on different networks to interoperate seamlessly. Internet Protocol version 4 (IPv4) is currently used in intranets and private networks, as well as the Internet. IPv6 is the successor to IPv4, and is based for the most part on IPv4.

This section discusses the following topics:

- IPv6 Standards on page 11
- IPv6 Introduction on page 11
- IPv6 Packet Headers on page 11
- IPv6 Addressing on page 12

## IPv6 Standards

IPv6 is defined in the following document:

RFC 2460, *Internet Protocol, Version 6 (IPv6)*

RFC 2373, *IP Version 6 Addressing Architecture*

To access Internet Requests for Comments (RFCs) and drafts, see <http://www.ietf.org>.

## IPv6 Introduction

IPv4 has been widely deployed and used to network the Internet today. With the rapid growth of the Internet, enhancements to IPv4 are needed to support the influx of new subscribers, Internet-enabled devices, and applications. IPv6 is designed to enable the global expansion of the Internet.

IPv6 builds upon the functionality of IPv4, providing improvements to addressing, configuration and maintenance, and security.

IPv6 offers the following benefits:

Expanded addressing capabilities—IPv6 provides a larger address space. IPv6 addresses consist of 128 bits, while IPv4 addresses consist of 32 bits. 128-bit addressing increases the address space by approximately  $10^{29}$  unique addresses, enough to last for the foreseeable future.

Header format simplification—IPv6 packet header format is designed to be efficient. IPv6 standardizes the size of the packet header to 40 bytes, divided into 8 fields.

Improved support for extensions and options—Extension headers carry Internet-layer information and have a standard size and structure.

Flow labeling capability—Flow labels provide consistent handling of packets belonging to the same flow.

Improved privacy and security—IPv6 supports extensions for authentication and data integrity, which enhances privacy and security.

## IPv6 Packet Headers

IPv6 introduces a new packet header structure.

This section discusses the following topics that provide background information about IPv6 headers:

Header Structure on page 12

Extension Headers on page 12

## Header Structure

IPv6 packet headers contain many of the fields found in IPv4 packet headers; some of these fields have been modified from IPv4. The 40-byte IPv6 header consists of the following 8 fields:

**Traffic Class**—Class-of-service (CoS) priority of the packet. Previously the type-of-service (ToS) field in IPv4. However, the semantics of this field (for example, diffserv code points) are identical to IPv4.

**Destination address**—Final destination node address for the packet.

**Flow label**—Packet flows requiring a specific CoS. The flow label identifies all packets belonging to a specific flow, and routers can identify these packets and handle them in a similar fashion.

**Hop limit**—Maximum number of hops allowed. Previously the time-to-live (TTL) field in IPv4.

**Next header**—Next extension header to examine. Previously the protocol field in IPv4.

**Payload length**—Length of the IPv6 payload. Previously the total length field in IPv4.

**Source address**—Address of the source node sending the packet.

**Version**—Version of the Internet Protocol.

## Extension Headers

In IPv6, *extension header s* are used to encode optional Internet-layer information.

Extension headers are placed between the IPv6 header and the upper layer header in a packet.

Extension headers are chained together using the next header field in the IPv6 header. The next header field indicates to the router which extension header to expect next. If there are no more extension headers, the next header field indicates the upper layer header (TCP header, User Datagram Protocol [UDP] header, ICMPv6 header, an encapsulated IP packet, or other items).

## IPv6 Addressing

IPv6 introduces a new 128-bit addressing model. This creates a much larger address space than IPv4 addresses, which are made up of 32 bits. IPv6 addresses also contain a scope field that categorizes what types of applications are suitable for the address. IPv6 does not support broadcast addresses, but instead uses multicast addresses to serve this role. In addition, IPv6 also defines a new type of address called *anycast*.

This section discusses the following topics that provide background information about IPv6 addressing:

Address Representation on page 13

Address Types on page 13

Address Scope on page 13

Address Structure on page 14

## Address Representation

IPv6 addresses consist of 8 groups of 16-bit hexadecimal values separated by colons (:). The IPv6 address format is as follows:

```
aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa
```

*aaaa* is a 16-bit hexadecimal value, and *a* is a 4-bit hexadecimal value. Following is an example of an actual IPv6 address:

```
3FFE:0000:0000:0001:0200:F8FF:FE75:50DF
```

You can omit the leading zeros, as shown:

```
3FFE:0:0:1:200:F8FF:FE75:50DF
```

You can compress 16-bit groups of zeros to “::”, as shown here, but only once per address:

```
3FFE::1:200:F8FF:FE75:50DF
```

## Address Types

There are three types of IPv6 addresses:

Unicast—For a single interface.

Multicast—For a set of interfaces on the same physical medium. A packet is sent to all of the interfaces associated with the address.

Anycast—For a set of interfaces on different physical mediums. A packet is sent to only one of the interfaces associated with this address, not to all the interfaces.

## Address Scope

IPv6 addresses have *scope*, which identifies the application suitable for the address. Unicast and multicast addresses support scoping.

Unicast addresses support two types of scope: *global scope* and *local scope*. There are two types of local scope: *link-local* addresses and *site-local* addresses. Link-local unicast addresses are used within a single network link. The first 10 bits of the prefix identify the address as a link-local address. Link-local addresses cannot be used outside a network link. Site-local unicast addresses are used within a site or intranet. A site consists of multiple network links, and site-local addresses identify nodes inside the intranet. Site-local addresses cannot be used outside the site.

Multicast addresses support 16 different types of scope, including node, link, site, organization, and global scope. A four-bit field in the prefix identifies the scope.

## Address Structure

Unicast addresses identify a single interface. The address consists of  $n$  bits for the prefix, and  $128-n$  bits for the interface ID.

Multicast addresses identify a set of interfaces. The address is made up of the first 8 bits of all ones, a 4-bit flags field, a 4-bit scope field, and a 112-bit group ID:

11111111 | *flgs* | *scop* | *group ID*

The first octet of ones identifies the address as a multicast address. The flags field identifies whether the multicast address is a well-known address or a transient multicast address. The scope field identifies the scope of the multicast address. The 112-bit group ID identifies the multicast group.

Similar to multicast addresses, anycast addresses identify a set of interfaces. However, packets are sent to only one of the interfaces, not to all interfaces. Anycast addresses are allocated from the normal unicast address space and cannot be distinguished from a unicast address in format. Therefore, each member of an anycast group must be configured to recognize certain addresses as anycast addresses.