NOKIA

Nokia Service Router Linux

ROUTING PROTOCOLS GUIDE
RELEASE 22.11

3HE 19047 AAAA TQZZA
Issue 01

November 2022

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

Table of contents

1 About this guide

This document describes routing protocols used with the Nokia Service Router Linux (SR Linux). Examples of commonly used commands are provided.

This document is intended for network technicians, administrators, operators, service providers, and others who need to understand how the router is configured.

Note:
This manual covers the current release and may also contain some content that will be released in later maintenance loads. See the *SR Linux Release Notes* for information on features supported in each load.

1.1 Precautionary and information messages

The following are information symbols used in the documentation.

DANGER: Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.

WARNING: Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.

Caution: Caution indicates that the described activity or situation may reduce your component or system performance.

Note: Note provides additional operational information.

Tip: Tip provides suggestions for use or best practices.

1.2 Conventions

Nokia SR Linux documentation uses the following command conventions.

- **Bold** type indicates a command that the user must enter.
- Input and output examples are displayed in `Courier` text.
- An open right-angle bracket indicates a progression of menu choices or simple command sequence (often selected from a user interface). Example: **start** > **connect to**.
- A vertical bar (|) indicates a mutually exclusive argument.
- Square brackets ([ ]) indicate optional elements.

- Braces ({ }) indicate a required choice. When braces are contained within square brackets, they indicate a required choice within an optional element.
- *Italic* type indicates a variable.

Generic IP addresses are used in examples. Replace these with the appropriate IP addresses used in the system.

2 What's new

This is the first release of this document. Information in this document was previously found in the *SR Linux Configuration Basics Guide*.

The following updates were made to content that previously existed in the *SR Linux Configuration Basics Guide*.

Topic	Location
IS-IS graceful restart	IS-IS graceful restart
IS-IS PDU authentication using directly configured keys	Configuring authentication keys

3 OSPF

Open Shortest Path First (OSPF) is a hierarchical link state protocol. OSPF is an interior gateway protocol (IGP) used within large autonomous systems (ASs). OSPF routers exchange state, cost, and other relevant interface information with neighbors. The information exchange enables all participating routers to establish a network topology map. Each router applies the Dijkstra algorithm to calculate the shortest path to each destination in the network. The resulting OSPF forwarding table is submitted to the routing table manager to calculate the routing table.

When a router is started with OSPF configured, OSPF, along with the routing-protocol data structures, is initialized and waits for indications from lower-layer protocols that its interfaces are functional.

The hierarchical design of OSPF allows a collection of networks to be grouped into a logical area. An area's topology is concealed from the rest of the AS which significantly reduces OSPF protocol traffic. With the correct network design and area route aggregation, the size of the route table can be greatly reduced, resulting in decreased OSPF route calculation time and topological database size.

Routers that belong to more than one area are called area border routers (ABRs). An ABR maintains a separate topological database for each area it is connected to. Every router that belongs to the same area has an identical topological database for that area.

Key OSPF areas are:

- Backbone Areas – The backbone distributes routing information between areas.
- Stub areas – A designated area that does not allow external route advertisements. Routers in a stub area do not maintain external routes. A single default route to an ABR replaces all external routes.
- Not-So-Stubby Areas (NSSAs) – NSSAs are similar to stub areas in that no external routes are imported into the area from other OSPF areas. External routes learned by OSPF routers in the NSSA area are advertised as type-7 LSAs within the NSSA area and are translated by ABRs into type-5 external route advertisements for distribution into other areas of the OSPF domain.

3.1 OSPF global configuration

The minimal OSPF parameters that should be configured to deploy OSPF are:

- OSPF version

  SR Linux supports OSPF version 2 and version 3. The OSPF version number must be specified in the configuration. If configuring OSPFv3, you must also specify the address family to be used, either IPv4 or IPv6.

- OSPF instance ID (when configuring multiple instances)

  An OSPF instance ID must be defined when configuring multiple instances or the instance being configured is not the base instance. If an instance ID is not configured, the default instance IDs are as follows:

  – 0 for OSPFv2

  – 0 for OSPF v3 with address family IPv6 unicast

  – 64 for OSPF v3 with address family IPv4 unicast

- Router ID

  Each router running OSPF must be configured with a unique router ID. The router ID is used by both OSPF and BGP routing protocols in the routing table manager.

  When you configure a new router ID, OSPF is automatically disabled and re-enabled to initialize the new router ID.

- An area

  At least one OSPF area must be created. An interface must be assigned to each OSPF area.

- Interfaces

  An interface is the connection between a router and one of its attached networks. An interface has state information associated with it, which is obtained from the underlying lower level protocols and the routing protocol itself. An interface to a network has associated with it a single IP address and mask (unless the network is an unnumbered point-to-point network). An interface is sometimes also referred to as a link.

3.1.1 Configuring basic OSPF parameters

Procedure

To create a basic OSPF configuration, the minimal OSPF parameters required are the following:

- OSPF version number, either v2 or v3. If configuring OSPFv3, you must specify the address family, either IPv4 or IPv6.
- A router ID
- One or more areas
- Interfaces

Example:

The following is an example of a basic OSPFv2 configuration:

```
--{ * candidate shared default }--[ network-instance default protocols ]--
# info ospf
    ospf {
        instance default {
            admin-state enable
            version ospf-v2
            router-id 1.1.1.1
            area 0.0.0.1 {
                interface ethernet-1/1.1 {
                    interface-type broadcast
                }
                interface ethernet-1/2.1 {
                    interface-type broadcast
                }
                interface ethernet-1/16.1 {
                    interface-type broadcast
                }
                interface lo0.1 {
                    interface-type broadcast
                }
            }
        }
```

```
    }
```

Example

The following is an example of a basic OSPFv3 configuration.

```
--{ * candidate shared default }--[ network-instance default protocols ]--
# info ospf
    ospf {
        instance default {
            admin-state enable
            version ospf-v3
            address-family ipv6-unicast
            router-id 1.1.1.1
            area 0.0.0.1 {
                interface ethernet-1/1.1 {
                    interface-type broadcast
                }
                interface ethernet-1/2.1 {
                    interface-type broadcast
                }
                interface ethernet-1/16.1 {
                    interface-type broadcast
                }
                interface lo0.1 {
                    interface-type broadcast
                }
            }
        }
    }
```

3.1.2 Configuring the router ID

Procedure

The router ID, expressed like an IP address, uniquely identifies the router within an AS. In OSPF, routing information is exchanged between ASs, groups of networks that share routing information. It can be set to be the same as the loopback (system interface) address. Subscriber services also use this address as far-end router identifiers when service distribution paths (SDPs) are created. The router ID is used by both OSPF and BGP routing protocols.

When you configure a new router ID, OSPF is automatically disabled and re-enabled to initialize the new router ID.

The router ID can be configured either at the network-instance level or at the OSPF protocol level. If a router ID is configured at the OSPF protocol level, OSPF uses it instead of the router ID configured at the network-instance level.

Example:

The following example configures a router ID at the network-instance level. OSPF uses this router ID unless a different router ID is configured at the OSPF protocol level.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        router-id 10.10.10.104{
        }
```

```
    }
```

Example

The following example configures a router ID for the OSPF instance at the OSPF protocol level:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            ospf {
                instance default {
                    version ospf-v2
                    router-id 2.2.2.2
                }
            }
        }
    }
```

3.1.3 Configuring an area

Procedure

An OSPF area consists of routers configured with the same area ID. To include a router in a specific area, the common area ID must be assigned and an interface identified.

If your network consists of multiple areas, you must also configure a backbone area (0.0.0.0) on at least one router. The backbone comprises the area border routers and other routers not included in other areas. The backbone distributes routing information between areas. The backbone is considered to be a participating area within the autonomous system. To maintain backbone connectivity, there must be at least one interface in the backbone area.

The minimal configuration must include an area ID and an interface. Modifying other command parameters are optional.

Example:

The following example configures an area at the OSPF level:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
    protocols {
        ospf {
            instance default {
                version ospf-v2
                area 1.1.1.1 {
                }
            }
        }
    }
```

3.1.4 Configuring a stub area

Procedure

You can configure stub areas to control external advertisement flooding and to minimize the size of the topological databases on an area's routers. A stub area cannot also be configured as an NSSA. By default, summary route advertisements are sent into stub areas.

Example:

The following example configures an OSPF stub area:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
    protocols {
        ospf {
            instance default {
                version ospf-v2
                area 1.1.1.1 {
                    stub {
                    }
                }
            }
        }
    }
```

3.1.5 Configuring a Not-So-Stubby area

Procedure

You can explicitly configure an area to be a Not-So-Stubby Area (NSSA). NSSAs are similar to stub areas in that no external routes are imported into the area from other OSPF areas. An NSSA has the capability to flood external routes it learns throughout its area and by an area border router to the entire OSPF domain. An area cannot be both a stub area and an NSSA.

Example:

The following is an OSPF NSSA configuration example:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
    protocols {
        ospf {
            instance default {
                version ospf-v2
                area 1.1.1.1 {
                    nssa {
                    }
                }
            }
        }
    }
```

3.1.6 Configuring an interface

Procedure

You can configure an interface to act as a connection between a router and one of its attached networks. An interface includes state information that was obtained from underlying lower level protocols and from the routing protocol itself. An interface to a network is associated with a single IP address and mask (unless the network is an unnumbered point-to-point network). If the address is merely changed, then the OSPF configuration is preserved.

Example:

The following is an OSPF interface configuration example:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
    protocols {
        ospf {
            instance default {
                version ospf-v2
                area 1.1.1.1 {
                    interface ethernet-1/2 {
                    }
                }
            }
        }
    }
```

4 BGP

Border Gateway Protocol (BGP) is an inter-AS routing protocol. An AS (autonomous system) is a network or a group of routers logically organized and controlled by common network administration. BGP enables routers to exchange network reachability information, including information about other ASs that traffic must traverse to reach other routers in other ASs.

ASs share routing information, such as routes to each destination and information about the route or AS path, with other ASs using BGP. Routing tables contain lists of known routers, reachable addresses, and associated path cost metrics for each router. BGP uses the information and path attributes to compile a network topology.

To set up BGP routing, participating routers must have BGP enabled, and be assigned to an AS, and the neighbor (peer) relationships must be specified. A router typically belongs to only one AS.

This section describes the minimal configuration necessary to set up BGP on SR Linux. This includes the following:

- Global BGP configuration, including specifying the autonomous system number (ASN) of the router, as well as the router ID.
- BGP peer group configuration, which specifies settings that are applied to BGP neighbor routers in the peer group.
- BGP neighbor configuration, which specifies the peer group to which each BGP neighbor belongs, as well as settings specific to the neighbor, including the AS to which the router is peered.

For information about all other BGP settings, see the SR Linux online help, as well as the *SR Linux Advanced Solutions Guide* and the *SR Linux Data Model Reference*.

4.1 BGP global configuration

Global BGP configuration includes specifying the autonomous system number (ASN) of the router and the router ID.

4.1.1 Configuring an ASN

Procedure

An autonomous system number (ASN) is a globally unique value that associates a router to a specific AS. Each router participating in BGP must have an ASN specified.

Example:

The following example configures an ASN for a router:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
```

```
                autonomous-system 65002
            }
        }
    }
```

4.1.2 Configuring the router ID

Procedure

The router ID, expressed like an IP address, uniquely identifies the router and indicates the origin of a packet for routing information exchanged between autonomous systems. The router ID is configured at the BGP level.

Example:

The following example configures a router ID:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                router-id 2.2.2.2
        }
    }
```

4.2 Configuring a BGP peer group

Procedure

A BGP peer group is a collection of related BGP neighbors. The group name should be a descriptive name for the group.

All parameters configured for a peer group are inherited by each peer (neighbor) in the peer group, but a group parameter can be overridden for specific neighbors in the configuration of that neighbor.

Example:

The following example configures the administrative state and trace options for a BGP peer group. These settings apply to all of the BGP neighbors that are members of this group, unless specifically overridden in the neighbor configuration.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                group headquarters1 {
                    admin-state enable
                    traceoptions {
                        flag events {
                        }
                        flag graceful-restart {
                        }
                    }
                }
```

```
            }
```

4.3 Configuring BGP neighbors

Procedure

After you configure a BGP group name and assign options, you can add neighbors within the same autonomous system to create internal BGP (iBGP) connections and/or neighbors in different autonomous systems to create external BGP (eBGP) peers. All parameters configured for the peer group to which the neighbor is assigned are applied to the neighbor, but a group parameter can be overridden on a specific neighbor basis.

Example:

The following example configures parameters for two BGP neighbors. The **peer-group** parameter configures both nodes to use the settings specified for the `headquarters1` group. The group settings apply unless they are specifically overridden in the neighbor configuration.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                neighbor 192.168.11.1 {
                    peer-group headquarters1
                    description "default network-instance bgp neighbor to Node A"
                    peer-as 65001
                    local-as 65002 {
                    }
                    multihop {
                        admin-state enable
                        maximum-hops 3
                    }
                    failure-detection {
                        enable-bfd true
                        fast-failover true
                    }
                }
                neighbor 192.168.13.2 {
                    peer-group headquarters1
                    description "default network-instance bgp neighbor to Node C"
                    peer-as 65003
                    local-as 65002 {
                    }
                    failure-detection {
                        enable-bfd true
                        fast-failover true
                    }
                }
            }
        }
    }
```

4.4 eBGP multihop

External BGP (eBGP) multihop can be used to form adjacencies when eBGP neighbors are not directly connected to each other; for example, when a non-BGP router is between the eBGP neighbors.

BGP TCP/IP packets sent toward an eBGP neighbor by default have a TTL value of 1. If the BGP TCP/IP packets need to pass through more than one router to reach their destination, the TTL decrements to 0, and the packets are dropped.

To prevent this, you can enable multihop for the eBGP neighbor and specify the maximum number of hops for BGP TCP/IP packets sent to the neighbor. This allows the eBGP neighbor to be indirectly connected by up to the specified number of hops.

When multihop is not enabled, the IP TTL for eBGP sessions is set to 1, and the IP TTL for iBGP sessions is set to 64. By enabling multihop and configuring the maximum number of hops to a neighbor, it allows an eBGP session to have multiple hops, and an iBGP session to have a single hop, if required.

If multihop is enabled and the maximum-hops parameter is configured for a BGP peer group, the settings are applied to the members of the group. If the multihop configuration for a neighbor is changed, the session with the neighbor must be disconnected and re-established for the change to take effect.

4.4.1 Configuring eBGP multihop

Procedure

To configure eBGP multihop, you enable it for the eBGP neighbor, and specify a value for the **maximum-hops** parameter. Additionally, the next-hop to the neighbor must be configured so that the two systems can establish a BGP session.

Example:

The following example enables multihop for an eBGP neighbor. The **maximum-hops** parameter is set to 2, which increases the TTL for BGP TCP/IP packets sent toward the eBGP neighbor, allowing the neighbor to be indirectly connected by up to 2 hops.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                neighbor 192.168.11.1 {
                    multihop {
                        admin-state enable
                        maximum-hops 2
                    }
                }
        }
    }
```

Example

The following example configures a route to the next-hop toward the eBGP neighbor:

```
--{ * candidate shared default }--[  ]--
# info network-instance default static-routes
    network-instance default {
        static-routes {
            route 192.168.11.0/24 {
```

```
                next-hop-group static-ipv4-grp
            }
        }
    }
--{ * candidate shared default }--[  ]--
# info network-instance default next-hop-groups group static-ipv4-grp
    network-instance default {
        next-hop-groups {
            group static-ipv4-grp {
                nexthop 1 {
                    ip-address 192.168.22.22
                }
            }
        }
    }
```

4.5 AS path options

You can set the following options for handling the AS_PATH in received BGP routes:

- Allow own AS – configures the router to process received routes when its own ASN appears in the AS_PATH.
- Replace peer AS – configures the router to replace the ASN of the peer router in the AS_PATH with its own ASN.
- Remove private AS path numbers – configures the router to either delete private AS numbers, shortening the AS path length, or replace private AS numbers with the local AS number used toward the peer, maintaining the AS path length.

4.5.1 Configuring allow-own-as

Procedure

Normally, when the ASN of a router appears in the AS_PATH of received routes, it is considered a loop, and the routes are discarded. The **allow-own-as** option configures the router to process the received routes when its own ASN appears in the AS_PATH. Specifically, it configures the maximum number of times the global ASN of the router can appear in any received AS_PATH before it is considered a loop and considered invalid. Default is 0.

Example:

The following example configures the router to process received routes where its own ASN appears in the AS_PATH a maximum of 1 time:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                autonomous-system 65001
                as-path-options {
                    allow-own-as 1
                    }
                }
        }
```

```
    }
```

4.5.2 Configuring replace-peer-as

Procedure

Normally, two sites having the same ASN would not be able to reach each other directly because the receiving router would see its own ASN in the AS_PATH and consider it a loop. To overcome this, you can configure the router to replace the peer ASN in the AS_PATH with its own ASN. When the **replace-peer-as** option is set to `true`, the router replaces every occurrence of the peer AS number that is present in the advertised AS_PATH with the local ASN used toward the peer.

Example:

The following example configures the router to replace the ASN of the peer with its own ASN:

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                as-path-options {
                    replace-peer-as true
                    }
                }
            }
        }
    }
```

4.5.3 Configuring remove-private-as

Procedure

You can configure how the router handles private AS numbers: either delete them, shortening the AS path length, or replace private AS numbers with the local AS number used toward the peer, which maintains the AS path length.

You can configure the router to delete or replace private AS numbers that appear before the first occurrence of a non-private ASN in the sequence of most recent ASNs in the AS path. You can also configure the router to ignore private AS numbers when they are the same as the peer ASN.

Example:

The following example configures the router to delete private AS numbers (2-byte and 4-byte) from the advertised AS path toward all peers. This shortens the AS path.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                as-path-options {
                    remove-private-as {
                        mode delete
                        }
                    }
```

```
                }
            }
        }
    }
```

Example

The following example configures the router to replace private AS numbers with the local AS number used toward the peer. This keeps the AS path the same length.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                as-path-options {
                    remove-private-as {
                        mode replace
                        }
                    }
                }
        }
    }
```

Example

The following example configures the router to replace only private AS numbers that appear before the first occurrence of a non-private ASN in the sequence of most recent ASNs in the AS path.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                as-path-options {
                    remove-private-as {
                        mode replace
                        leading-only true
                        }
                    }
                }
        }
    }
```

Example

The following example configures the router to ignore private AS numbers (neither delete nor replace them) when they are the same as the peer AS number.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                as-path-options {
                    remove-private-as {
                        mode replace
                        ignore-peer-as true
                        }
                    }
                }
```

```
        }
    }
```

4.6 Route reflection

In a standard iBGP configuration, all BGP speakers within an AS must have full BGP mesh to ensure that all externally learned routes are redistributed through the entire AS.

Configuring route reflection provides an alternative to the full BGP mesh requirement: instead of peering with all other iBGP routers in the network, each iBGP router only peers with a router configured as a route reflector.

An AS can be divided into multiple clusters, with each cluster containing at least one route reflector, which redistributes routes to the clients in the cluster. The clients within the cluster do not need to maintain a full peering mesh between each other. They only require a peering to the route reflectors in their cluster. The route reflectors must maintain a full peering mesh between all non-clients within the AS.

4.6.1 Configuring route reflection

Procedure

To configure a route reflector, you assign it a cluster ID and specify which neighbors are clients and which are non-clients. Clients receive reflected routes, and non-clients are treated as a standard iBGP peer.

Example:

The following example configures the router to be a route reflector for two clients SRL-1 and SRL-2. The router is assigned cluster ID 0.0.0.1.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                route-reflector {
                    cluster-id 0.0.0.1
                    }
                }
                neighbor SRL-1 {
                    route-reflector {
                        cluster-id 0.0.0.1
                        client true
                    }
                neighbor SRL-2 {
                    route-reflector {
                        cluster-id 0.0.0.1
                        client true
                    }
                }
            }
        }
    }
```

4.7 BGP graceful restart

BGP graceful restart allows a router whose control plane has temporarily stopped functioning because of a system failure or a software upgrade to return to service with minimal disruption to the network.

To do this, the router relies on neighbor routers, which have also been configured for graceful restart, to maintain forwarding state while the router restarts. These neighbor routers are known as helper routers. The helper routers and the restarting router continue forwarding traffic using the previously learned routing information from the restarting router. Other routers in the network are not notified about the restarting router, so network traffic is not disrupted.

When graceful restart is enabled on the SR Linux and its neighbor, the two routers exchange information about graceful restart capability, including the Address Family Identifier (AFI) and Subsequent Address Family Identifier (SAFI) of the routes supported for graceful restart.

While the router restarts, the helper router marks the routes from the restarting router as stale, but continues to use them for traffic forwarding. When the BGP session is reestablished, the restarting router indicates to the helper router that it has restarted. The helper router then sends the restarting router any BGP RIB updates, followed by an End-of-RIB (EOR) marker indicating that the updates are complete. The restarting router then makes its own updates and sends them to the helper router, followed by an EOR marker.

Graceful restart is used in conjunction with the In-Service Software Upgrade (ISSU) feature, which can be used to upgrade 7220 IXR-D2 and D3 systems while maintaining non-stop forwarding. During the ISSU, a warm reboot brings down the control and management planes while the NOS reboots, and graceful restart maintains the forwarding state in peers. You can use a **tools** command to validate that the SR Linux and its peers support warm reboot, including graceful restart configuration. See the *SR Linux Software Installation Guide* for more information.

4.7.1 Configuring graceful restart

Procedure

The following example enables graceful restart for the BGP instance. The SR Linux operates as a helper router for neighbor routers when they are restarting, assuming graceful restart is also enabled on the neighbors. Enabling graceful restart also indicates to the neighbors that they can serve as helper routers when the SR Linux itself is restarting.

Example:

When operating as a helper router, the SR Linux marks the routes from the restarting router as stale, but continues to use them for forwarding for a period of time while the neighbor router restarts. After this period expires, the SR Linux deletes the routes. The **stale-routes-time** parameter configures the amount of time in seconds the routes remain stale before they are deleted.

```
--{ * candidate shared default }--[  ]--
# info network-instance default
    network-instance default {
        protocols {
            bgp {
                graceful-restart {
                    admin-state enable
                    stale-routes-time 300
                }
            }
        }
```

```
    }
```

Following a restart, by default the system waits 600 seconds (10 minutes) to receive EOR markers from all helper routers for all address families that were up before the restart. After this time elapses, the system assumes convergence has occurred and sends its own EOR markers to its peers. You can configure the amount of time the system waits to receive EOR markers to be from 0 to 3,600 seconds.

For example, the following configures the amount of time the system waits to receive EOR markers to 270 seconds.

```
--{ * candidate shared default }--[  ]--
# info system warm-reboot
    system {
        warm-reboot {
            bgp-max-wait 270
        }
    }
```

4.8 BGP unnumbered peering

In a typical large-scale data center using BGP, leaf and spine switches are interconnected in a Clos topology, and each device establishes a single-hop eBGP session with each of its physically connected peers. The sessions come up as eBGP because of the ASN allocation scheme; it is common practice to assign a unique ASN to every leaf switch (TOR) in a cluster and a different unique ASN to the set of spine switches to which those TORs are connected. The allocated ASNs are typically private ASNs in the range 4200000000 to 4294967294, although this is not always the case.

For this type of configuration, BGP unnumbered peering can be a useful solution. BGP unnumbered peering is the dynamic setup of one or more single-hop BGP sessions over a network segment that has no globally-unique IPv4 or IPv6 addresses. Each router connected to the network segment is assumed to have an IPv6-enabled interface to the network, and these interfaces have IPv6 link-local addresses that are typically auto-generated by each router from the interface MAC addresses.

How sessions are established using BGP unnumbered peering

The set of BGP speakers configured for BGP unnumbered peering on a network segment discover each other by sending and receiving ICMPv6 router advertisement (RA) messages.

Consider an example of Router A and Router B, which are both connected to an unnumbered interface and configured for BGP unnumbered dynamic session setup. The BGP session between the two routers is established in the following sequence:

1. Router B sends an ICMPv6 RA message on its interface b1.
   Assuming the RA message is unsolicited, the source IP address of this message is the link-local address of interface b1 (fe80::7efe:90ff:fefc:7ad8), and the destination IP address is the all-nodes multicast address.
2. Asynchronously, Router A sends an ICMPv6 RA message on its interface a1.
   The source IP address is the link-local address of interface a1 (fe80::7efe:90ff:fefc:7bd8), and the destination IP address is the all-nodes multicast address.
3. Router A receives the RA message on interface a1, and the software process responsible for ICMPv6 relays the information to BGP, because in the BGP configuration, a1 is a subinterface that is configured as a dynamic neighbor interface; that is, added to the BGP dynamic-neighbors interface list.

4. BGP checks if it already has a BGP session with fe80::7efe:90ff:fefc:7ad8.
   - If BGP already has this session and it is up, or BGP is in the process of establishing this session, then the new information is a no-op. Possibly Router B started the same process moments before Router A.
   - If BGP does not have a session with this link-local address, then a new TCP connection is initiated toward fe80::7efe:90ff:fefc:7ad8.
5. When the TCP connection is established, the BGP OPEN message sent by Router A encodes a local-AS and other capabilities that come from the configuration of the peer-group associated with interface a1.
6. Router A receives a BGP OPEN message from Router B and accepts that OPEN message, proceeding to move toward the BGP established state, if the OPEN message encodes an acceptable peer AS number (in one of the `allowed-peer-as` ranges configured for interface a1). The address families supported by the session are based on the usual MP-BGP negotiation.

BGP dynamic-neighbors interface list

To enable dynamic peering, you add subinterfaces to the BGP dynamic-neighbors interface list in the SR Linux configuration.

When a subinterface is added to the dynamic-neighbors interface list:

- BGP automatically accepts incoming BGP connections to the IPv6 link-local address of that subinterface, subject to the configured `max-sessions` limit for the subinterface.
  For the connection to be accepted, the source address must be an IPv6 link-local address (that may or may not also be a defined neighbor address), and the reported ASN of the peer must match relevant configuration. If the source address does not match a configured neighbor address, the session is set up according to the peer-group associated with the subinterface, not the peer-group associated with the **dynamic-neighbors accept match-prefix** entry matching the source IPv6 link-local address if a matching entry exists.
- BGP registers for IPv6 RA messages on the subinterface. Whenever the source of one of these RA messages matches an IPv6 link-local address for which there is currently no established BGP session, the system attempts to create a BGP session to that address, as long as this does not exceed the configured `max-sessions` limit for the subinterface. The session is set up according to the configured `peer-group` associated with the subinterface.

When a BGP session is established over a subinterface in the in the dynamic-neighbors interface list:

- Changes to the `allowed-peer-as` ranges associated with the subinterface are a no-op until the next time BGP attempts to establish the sessions.
- Non-arrival of expected ICMPv6 RA messages on the subinterface are a no-op, and do not trigger teardown of associated sessions.
- Existing triggers for tearing down a session apply as normal (for example, hold-timer expiration, BFD timeout, **clear bgp neighbor** commands, and so on).
- If the link-local address of a dynamic peer is configured as a static neighbor address, the dynamic session is immediately torn down and replaced by the static session.

When a subinterface is deleted from the dynamic-neighbors interface list, all dynamic sessions associated with that subinterface (excluding sessions set up by static configuration of the neighbor) are torn down immediately.

A BGP session that was previously established on an unnumbered interface and subsequently torn down can only be re-established if the subinterface is configured in the dynamic-neighbors interface list and a recent ICMPv6 RA message is received.

Configuration overrides for dynamic peers on unnumbered interfaces

When a dynamic BGP session is initiated or accepted on an interface that is tied to a peer-group, most of the parameters relevant to that session come from the configuration of that peer-group, with the following exceptions:

- `multihop maximum-hops` is always 1 (for both eBGP and iBGP peers).
- `transport local-address` is always the link-local address of the specified interface.
- `next-hop-self` is always `true`. The neighbor is not presumed to have reachability to off-link destinations.
- `transport passive-mode` is always `false`. BGP always initiates a connection when informed by ICMPv6, unless it already has a connection.
- `ipv4-unicast receive-ipv6-next-hops` is always `enabled`.
- `ipv4-unicast advertise-ipv6-next-hops` and `evpn advertise-ipv6-next-hops` are always `enabled`.
- `graceful-restart admin-state` is always `disabled`, meaning dynamic-peers do not help during a warm reboot, and there is no helper support for dynamic peers that restart.

Peer AS Validation for dynamic peers on unnumbered interfaces

When a BGP OPEN message is received from an unnumbered dynamic neighbor, the reported AS number of the peer is checked to determine if it is acceptable to allow the peering to proceed.

For a dynamic session associated with a subinterface, the peer AS is acceptable only if it matches one of the `allowed-peer-as` elements of the dynamic-neighbors interface list entry for the subinterface, and if the peer AS is not equal to the local AS (implying an iBGP session; in the current release, iBGP is not supported for BGP unnumbered peering).

4.8.1 Configuring BGP unnumbered peering

Procedure

To configure BGP unnumbered peering, you add subinterfaces to the BGP dynamic-neighbors interface list, and specify the peer autonomous system numbers from which incoming TCP connections to the BGP well-known port are accepted.

Example

The following example adds a subinterface to the BGP dynamic-neighbors interface list.

```
--{ candidate shared default }--[  ]--
# info network-instance default protocols bgp dynamic-neighbors interface ethernet-1/1.1
    network-instance default {
        protocols {
            bgp {
                dynamic-neighbors {
                    interface ethernet-1/1.1 {
                        peer-group bgp_peer_group_0
                        allowed-peer-as [
```

```
                            4294967200
                        ]
                    }
                }
            }
        }
    }
```

In this example, subinterface `ethernet-1/1.1` is added to the BGP dynamic-neighbors interface list. This subinterface must be enabled for IPv6 and configured to accept and send IPv6 RA messages. It does not require any IPv4 addresses or global-unicast IPv6 addresses.

Incoming TCP connections to port 179 received on this subinterface that are sourced from an IPv6 link-local address and destined for the IPv6 link local address of this subinterface are automatically accepted. IPv6 RA messages received on this subinterface automatically trigger BGP session setup toward the sender of these messages, if there is not already an established BGP session.

Peer group `bgp_peer_group_0` is associated with dynamic BGP neighbors on this subinterface. Parameters configured for this peer-group are used for establishing the dynamic BGP session, with the exceptions described in Configuration overrides for dynamic peers on unnumbered interfaces.

ASN 4294967200 is configured as an allowed peer AS for dynamic BGP neighbors on this subinterface. If the BGP OPEN message from a peer on this subinterface contains a MyAS number that is not an allowed peer AS, then a NOTIFICATION is sent to the peer with the indication Bad Peer AS.

4.9 BGP configuration management

Managing the BGP configuration on SR Linux can include the following tasks:

- Modifying an AS number
- Deleting a BGP neighbor from the configuration
- Deleting a BGP group
- Resetting BGP peer connections

4.9.1 Modifying an ASN

Procedure

You can modify the ASN on the router, but the new ASN does not take effect until the BGP instance is restarted, either by administratively disabling/enabling the BGP instance, or by rebooting the system with the new configuration.

Example:

```
--{ * candidate shared default }--[ network-instance default ]--
# protocols bgp autonomous-system 95002
# protocols bgp admin-state disable
# protocols bgp admin-state enable
```

All established BGP sessions are taken down when the BGP instance is disabled.

4.9.2 Deleting a neighbor

Procedure

Use the **delete** command to delete a BGP neighbor from the configuration.

Example:

```
--{ * candidate shared default }--[ network-instance default ]--
# delete protocols bgp neighbor 192.168.11.1
```

4.9.3 Deleting a group

Procedure

Use the **delete** command to delete the settings for a BGP peer group from the configuration.

Example:

```
--{ * candidate shared default }--[ network-instance default ]--
# delete protocols bgp group headquarters1
```

4.9.4 Resetting BGP peer connections

Procedure

To refresh the connections between BGP neighbors, you can issue a hard or soft reset. A hard-reset tears down the TCP connections and returns to IDLE state. A soft-reset sends route-refresh messages to each peer. The hard or soft reset can be issued to a specific peer, to peers in a specific peer-group, or to peers with a specific ASN.

Example:

The following command hard-resets the connections to the BGP neighbors in a peer group that have a specified ASN. The hard-reset applies both to configured peers and dynamic peers.

```
# tools network-instance default protocols bgp group headquarters1 reset-peer peer-as
 95002
/network-instance[name=default]/protocols/bgp/group[group-name=headquarters1]:
    Successfully executed the tools clear command.
```

Example

The following command soft-resets the connection to BGP neighbors that have a specified ASN. The soft-reset applies both to configured peers and dynamic peers.

```
# tools network-instance default protocols bgp soft-clear peer-as 95002
/network-instance[name=default]/protocols/bgp:
    Successfully executed the tools clear command.
```

4.10 BGP shortcuts

With BGP shortcuts, SR Linux can include LDP LSPs or segment routing (SR-ISIS) tunnels in the BGP algorithm calculations. In this case, tunnels operate as logical interfaces directly connected to remote nodes in the network. Because the BGP algorithm treats the tunnels in the same way as a physical interface (being a potential output interface), the algorithm can select a destination node together with an output tunnel to resolve the next-hop, using the tunnel as a shortcut through the network to the destination.

Note: BGP shortcuts can only be used for next-hop resolution of IPv4-unicast RIB-Ins with an IPv4 next-hop address.

BGP next-hop resolution describes the procedures that BGP uses to resolve the next-hop address of each BGP RIB-In that forms part of a BGP route. The following table defines BGP RIB-In and BGP route in the context of BGP next-hop resolution.

Table 1: BGP RIB-IN and BGP route

BGP Term	Definition
BGP RIB-In	One of the following: • a received IPv4-unicast BGP route with an IPv4 next-hop address • a received IPv4-unicast BGP route with an IPv6 next-hop address (allowed as a result of sending an extended-nh-encoding capability to the peer) • a received IPv6-unicast BGP route with an IPv6 next-hop address
BGP route	A route submitted by BGP to the fib_mgr that resulted from the grouping of one or more BGP RIB-Ins. (Multiple BGP RIB-Ins per route describes a multipath scenario.)

With BGP shortcuts enabled, next-hop resolution determines whether to use a local interface or a tunnel to resolve the BGP next-hop.

Tunnel resolution mode

As part of the configuration for BGP shortcuts, you must define the tunnel-resolution mode (prefer/required/disabled). This mode determines the order of preference and fallback of using tunnels in the tunnel table to resolve the next-hop instead of using routes in the FIB, as described in the following sections.

Next-Hop Resolution of IPv4-Unicast RIB-Ins with IPv4 next-hop

The following table describes the next-hop resolution steps for IPv4-Unicast RIB-Ins with IPv4 next-hops, depending on the specified tunnel resolution mode.

Table 2: Next-hop resolution for IPv4 Unicast RIB-Ins with IPv4 next-hop address

Tunnel Resolution Mode	Next-hop resolution steps in BGP
prefer	**1.** Start with TTM lookup: **a.** Find all the tunnels in TTM with an endpoint that matches the BGP next-hop address and that have one of the types listed in the allow list. **b.** If there is a single tunnel, select that tunnel. The RIB-IN is resolved; exit.

Tunnel Resolution Mode	Next-hop resolution steps in BGP
	c. If there are multiple tunnels, select the tunnel with the numerically lowest TTM preference, and if a further tie-break is needed, select the tunnel with the lowest TTM metric. The RIB-IN is resolved; exit. **2.** If there are no tunnels, fallback to FIB lookup: **a.** Find the longest match active route in the FIB that matches the BGP next-hop address. There are presently no restrictions on this route; it can be an IGP route, a static blackhole route, a default route, or another BGP route. **b.** If there is a longest match route and it eventually resolves to a blackhole next-hop, interface or tunnel then the RIB-IN is resolved; exit. **c.** If there is no matching route the RIB-IN is unresolved.
require	Perform TTM lookup only, as described in 1 above. If there is no matching tunnel, the RIB-IN is unresolved.
disabled	Perform FIB lookup only, as described in 2 above.

Next-Hop Resolution of IPv4-Unicast and IPv6-Unicast RIB-Ins with IPv6 next-hop

If the next-hop address for the IPv4-Unicast RIB-In is an IPv6 address, the next-hop is resolved by the longest prefix match IPv6 route in the FIB. This is the only option because there are no IPv6 tunnels in the TTM. The same logic applies to BGP RIB-Ins with IPv6-unicast NLRI address family as they can only have an IPv6 next-hop address. The next-hop resolution logic is the same as the FIB lookup described in the preceding table.

4.10.1 Configuring BGP shortcuts over segment routing

About this task

This task describes how to configure BGP shortcuts.

Procedure

Step 1. In the default network-instance, define the tunnel-resolution mode for the BGP protocol.

This setting determines the order of preference and the fallback when using tunnels in the tunnel table instead of routes in the FIB. Available options are as follows:

- **require**
  requires tunnel table lookup instead of FIB lookup
- **prefer**
  prefers tunnel table lookup over FIB lookup
- **disabled (default)**
  performs FIB lookup only

Step 2. Set the allowed tunnel types for next-hop resolution.

Example: Configure IPv4 BGP shortcuts

The following example shows the BGP next-hop resolution configuration to allow IPv4 SR-ISIS tunnels, with the tunnel mode set to `prefer`.

```
--{ * candidate shared default }--[ ]--
# info network-instance default protocols bgp ipv4-unicast next-hop-resolution ipv4-next-
hops tunnel-resolution
    network-instance default {
        protocols {
            bgp {
                ipv4-unicast {
                    next-hop-resolution {
                        ipv4-next-hops {
                            tunnel-resolution {
                                mode prefer
                                allowed-tunnel-types [
                                    sr-isis
                                ]
                            }
                        }
                    }
                }
            }
        }
    }
```

Example: Configure IPv6 BGP shortcuts

The following example shows the BGP next-hop resolution configuration to allow IPv6 SR-ISIS tunnels, with the tunnel mode set to `prefer`.

```
--{ * candidate shared default }--[ ]--
# info network-instance default protocols bgp ipv4-unicast next-hop-resolution ipv4-next-
hops tunnel-resolution
    network-instance default {
        protocols {
            bgp {
                ipv6-unicast {
                    next-hop-resolution {
                        ipv6-next-hops {
                            tunnel-resolution {
                                mode prefer
                                allowed-tunnel-types [
                                    sr-isis
                                ]
                            }
                        }
                    }
                }
            }
        }
    }
```

5 IS-IS

Intermediate System to Intermediate System (IS-IS) is a link-state IGP that uses the Shortest Path First (SPF) algorithm to determine routes. Routing decisions are made using the link-state information. IS-IS evaluates topology changes and, if necessary, performs SPF recalculations.

Entities within IS-IS include networks, intermediate systems, and end systems. In IS-IS, a network is an Autonomous System (AS), or routing domain, with end systems and intermediate systems. A router is an intermediate system that sends, receives, and forwards Protocol Data Units (PDUs). End systems are network devices that send and receive PDUs.

End system and intermediate system protocols allow routers and nodes to identify each other. IS-IS sends out link-state updates periodically throughout the network, so each router can maintain current network topology information.

IS-IS supports large ASs by using a two-level hierarchy. A large AS can be administratively divided into smaller, more manageable areas. A system logically belongs to one area. Level 1 routing is performed within an area. Level 2 routing is performed between areas. You can configure routers as Level 1, Level 2, or both Level 1 and 2.

On SR Linux, you can configure a single named IS-IS instance per network instance. The following summarizes SR Linux support for IS-IS:

- Level 1, Level 2, and Level 1/2 IS types
- Configurable Network Entity Title (NET) per IS-IS instance
- support for IPv4/v6 routing
- ECMP with up to 64 next hops per destination
- IS-IS export policies (redistribution of other types of routes into IS-IS)
- authentication of LSP, CSNP, PSNP, and hello PDUs, using an authentication key or keychain specified as follows:
  - per instance or per level for all PDU types
  - per interface or per interface and level for Hello PDUs
- authentication keychains with a single key per named keychain
- Purge Originator ID TLV (RFC 6232)
- options to ignore and suppress the attached bit
- ability to set the overload bit immediately or after each subsequent restart of the IS-IS manager application and leave it on for a configurable duration each time
- control over the link-state PDU (LSP) MTU size, with range from 490 bytes to 9490 bytes
- configuration control over timers for LSP lifetime, LSP refresh interval, SPF calculation triggers, and LSP generation
- hello padding (strict, loose, and adaptive modes)
- graceful restart, acting in restarting router mode and helper router mode
- Level 1 to Level 2 route summary

- BFD for fast failure detection
- configurable hello timer with multiple per interface and level
- wide metrics (configurable per level)
- configurable route preference for each route type, Level 1-internal, Level 1-external, Level 2-internal and Level 2-external
- detailed statistics for interfaces, adjacencies, and levels

The **info detail** command displays default values for an IS-IS instance on SR Linux as shown in the following example:

```
--{ * candidate shared default }--[ network-instance default protocols isis ]--
# info detail
    instance i1 {
        admin-state disable
        level-capability L2
        max-ecmp-paths 1
        poi-tlv false
        attached-bit {
            ignore false
            suppress false
        }
        overload {
            advertise-interlevel false
            advertise-external false
            immediate {
                set-bit false
                max-metric false
            }
            on-boot {
                set-bit false
                max-metric false
            }
        }
        timers {
            lsp-lifetime 1200
            lsp-refresh {
                interval 600
                half-lifetime true
            }
            spf {
                initial-wait 1000
                second-wait 1000
                max-wait 10000
            }
            lsp-generation {
                initial-wait 1000
                second-wait 1000
                max-wait 5000
            }
        }
        transport {
            lsp-mtu-size 1492
        }
        ipv4-unicast {
            admin-state enable
        }
        ipv6-unicast {
            admin-state enable
            multi-topology false
        }
```

```
        graceful-restart {
            helper-mode false
        }
        auto-cost {
        }
        authentication {
            csnp-authentication false
            psnp-authentication false
            hello-authentication false
        }
        inter-level-propagation-policies {
            level1-to-level2 {
            }
        }
    }
```

5.1 Basic IS-IS configuration

To configure IS-IS, perform the following tasks:

- Enable an IS-IS instance
- If necessary, modify the level capability on the global IS-IS instance level
- Define area addresses
- Configure IS-IS interfaces

5.1.1 Enabling an IS-IS instance

Procedure

SR Linux supports a single IS-IS instance within a network instance. The following example enables an IS-IS instance within the default network instance.

Example

```
--{ * candidate shared default }--[ network-instance default protocols ]--
# info isis
 isis {
        instance i1 {
        }
    }
```

5.1.2 Configuring the router level

Procedure

When IS-IS is enabled, the default level-capability value is Level 1/2. This means that the router operates with both Level 1 and Level 2 routing capabilities. To change the default value in order for the router to operate as a Level 1 router or a Level 2 router, you must explicitly modify the level value.

The level-capability value can be configured on the global IS-IS instance level and also on the interface level. The level-capability value determines which level values can be assigned on the router level or on an interface-basis.

In order for the router to operate as a Level 1 only router or as a Level 2 only router, you must explicitly specify the level-number value.

- Specify Level 1 to route only within an area
- Specify Level 2 to route to destinations outside an area, toward other eligible Level 2 routers

Example

The following example configures the level capability for an IS-IS instance to Level 2.

```
--{ * candidate shared default }--[ network-instance default protocols ]--
# info isis
 isis {
        instance i1 {
            level-capability L2
        }
    }
```

5.1.3 Configuring the Network Entity Title

Procedure

SR Linux supports a configurable network entity title (NET) per IS-IS instance. The NET is 8-20 octets long and consists of 3 parts: the area address (1-13 octets), the system ID (6 octets), and the n-selector (1 octet, must be 00)

The area address portion of the NET defines the IS-IS area to which the router belongs. At least one area address should be configured on each router participating in IS-IS.

The area address portion of the NET identifies a point of connection to the network, such as a router interface. The routers in an area manage routing tables about destinations within the area. The NET value is used to identify the IS-IS area to which the router belongs.

The NET value is divided into three parts. Only the Area ID portion is configurable.

1. Area ID — A variable length field between 1 and 13 bytes. This includes the Authority and Format Identifier (AFI) as the most significant byte and the area ID.

2. System ID — A 6-byte system identifier. This value is not configurable. The system ID is derived from the system or router ID.

3. Selector ID — A 1-byte selector identifier that must contain zeros when configuring a NET. This value is not configurable. The selector ID is always 00.

Example

The following example configures a NET for an IS-IS instance:

```
--{ * candidate shared default }--[ network-instance default protocols ]--
# info isis
 isis {
        instance i1 {
            net 49.0001.1921.6800.1002.00
            }
        }
    }
```

5.1.4 Configuring global parameters

Procedure

Commands and parameters configured on the global IS-IS instance level are inherited by the interface levels. Parameters specified in the interface and interface-level configurations take precedence over global configurations.

Example

The following example shows the command usage to configure global-level IS-IS. The LSP PDU authentication setting references a keychain defined at the system level (see Protocol authentication).

```
--{ * candidate shared default }--[ network-instance default protocols ]--
# info isis
    isis {
        instance i1 {
            level-capability L2
            overload {
                on-boot {
                    timeout 90
                }
            }
            authentication {
                lsp-authentication {
                    generate true
                    check-received strict
                    keychain isisglobal
                }
            }
        }
    }
```

5.1.5 Configuring interface parameters

Procedure

There are no interfaces associated with IS-IS by default. An interface belongs to all areas configured on a router. Interfaces cannot belong to separate areas. There are no default interfaces applied to the router IS-IS instance. You must configure at least one IS-IS interface in order for IS-IS to work.

You can configure both the Level 1 parameters and the Level 2 parameters on an interface. The level-capability value determines which level values are used.

Example

The following example configures interface parameters for an IS-IS instance:

```
--{ * candidate shared default }--[ network-instance default protocols isis ]--
# info instance i1
    instance i1 {
        interface ethernet-1/2.1 {
            circuit-type point-to-point
            ipv4-unicast {
                admin-state enable
            }
            level 1 {
                authentication {
```

```
                    hello-authentication {
                        generate true
                        check-received strict
                        keychain Hello
                    }
                }
            }
        }
        level 1 {
        }
    }
```

5.1.6 Configuring authentication keys

Procedure

IS-IS supports authentication for PDUs using shared keys, which are changed at regular intervals using keys configured in a keychain. This authentication mechanism is described in Protocol authentication.

In addition to using shared keys, authentication for IS-IS Hello, CSNP, PSNP, and LSP PDUs can be done using directly configured keys. You can specify the key used for authenticating IS-IS PDUs associated with a specific IS-IS instance, received or transmitted on a specific interface, and associated with a specific level.

If a Hello PDU is received or transmitted on a specific interface, it is authenticated using the key configured for that interface. If no key exists for the interface, the key configured for the instance is used. For CSNP, PSNP. and LSP PDUs, authentication is performed using the key configured for the level. If no key exists for the level, the key configured for the instance is used.

To configure a key, you specify the secret key (`auth-password`) and cryptographic algorithm to be used for generating the key.

Example

The following example configures keys for an IS-IS instance, interface, and level.

```
--{ candidate shared default }--[  ]--
# info network-instance default protocols isis
    network-instance default {
        protocols {
            isis {
                instance i1 {
                    authentication {
                        key {
                            crypto-algorithm cleartext
                            auth-password $aes$9G3XrtckZzMg=$In9WuOvKPsTw6ehDX5YLgA==
                        }
                    }
                    interface ethernet-1/1.1 {
                        authentication {
                            key {
                                crypto-algorithm hmac-md5
                                auth-password $aes$97mfUA4Swx6I=$PfF02Mtu0gUXH5LwT/ltqQ==
                            }
                        }
                    }
                    level 1 {
                        authentication {
                            key {
                                crypto-algorithm hmac-sha-256
```

```
                                auth-password $aes$9YHCLtkxaLGw=$7HXuQHhR4wPXwifkGekFaQ==
                            }
                        }
                    }
                }
            }
        }
    }
```

5.2 IS-IS graceful restart

IS-IS graceful restart is a mechanism to prevent routing protocol re-convergence during a control plane switchover, reset, or upgrade.

Without graceful restart, when an IS-IS router restarts, its IS-IS neighbors detect that the router has gone down and remove routes that pass through that neighbor, resulting in data loss even if the restarting router is able to maintain its forwarding table state.

Configuring graceful restart can prevent this data loss. To do this, the restarting router relies on neighbor routers, which have also been configured for graceful restart, to maintain forwarding state while the router restarts. These neighbor routers are known as helper routers.

For IS-IS graceful restart, SR Linux can operate in the following modes:

- restarting router mode when its own IS-IS manager application (isis_mgr) restarts, or the application is killed
- helper router mode when it detects that an adjacent IS-IS router has restarted

Restarting router mode

When the router is configured to operate in restarting router node, if the SR Linux IS-IS manager application restarts or is killed, routes that are published by the IS-IS manager (associated with any IS-IS instance), and used for programming the FIB, remain programmed for a duration of time sufficient to allow a warm restart of the IS-IS manager application. The system starts sending periodic IIH PDUs (with the Restart Request flag set) on all IS-IS interfaces so that neighbors do not time out their adjacencies.

The helper router on each interface responds with an IIH PDU containing a Restart Acknowledgement (RA). The exchange of IIH PDUs on each interface causes adjacencies to reform. After each adjacency comes up, the helper router on the associated interface transmits a complete set of CSNPs to the restarting router.

Helper router mode

When the router is configured to operate in helper router node, SRLinux advertises TLV 211 in IIH PDUs sent on all IS-IS interfaces, which indicates that the IS-IS instance is able to help any adjacent neighbor that signals a restart.

When the router receives an IIH PDU with the RR flag set from an adjacent neighbor, the router maintains its adjacency with the neighbor, responds with an IIH PDU with the RA flag set, and sends a complete set of CSNP PDUs to the restarting router.

5.2.1 Configuring IS-IS graceful restart

Procedure

You can configure SR Linux to operate as a restarting router (informing adjacent routers when the IS-IS manager application restarts or is killed) and as a helper router (indicating to adjacent routers that it can help those signaling a restart). By default, both restarting router mode and helper router mode are disabled.

Example: Configure restarting router mode

To configure the router to operate in restarting router mode for IS-IS graceful restart, enable **non-stop-forwarding** for IS-IS. For example:

```
--{ candidate shared default }--[  ]--
# info network-instance green protocols isis
    network-instance default {
        protocols {
            isis {
                non-stop-forwarding {
                    admin-state enable
                }
            }
        }
    }
```

Example: Configure helper router mode

The following example enables the router to operate in helper router mode:

```
--{ candidate shared default }--[  ]--
# info network-instance green protocols isis instance i1 graceful-restart
    network-instance default {
        protocols {
            isis {
                instance i1 {
                    graceful-restart {
                        helper-mode true
                        acceptable-duration 120
                    }
                }
            }
        }
    }
```

The **acceptable-duration** parameter sets the amount of time in seconds that SR Linux advertises as the Remaining Time in the Restart TLV with the RA flag set when this router starts to help another router that has entered restart mode. By default, this is 60 seconds.

5.3 Displaying IS-IS information

Procedure

Use the commands shown in this section to display the following information for an IS-IS instance running in a specified network instance:

- interface information

- adjacency information
- IS-IS link state database information

Example: IS-IS summary information

To display summary information for an IS-IS instance:

```
# show network-instance green_default protocols isis summary
-------------------------------------------------------------------------------------------
-----
Network instance "green_default", isis instance "default" is enable and up
Level Capability : L1L2
Export policy    : None
-------------------------------------------------------------------------------------------
-----
System-id : 0050.0500.5005
NET       : [ 49.0001.0050.0500.5005.00 ]
Area-id   : [ 49.0001 ]
-------------------------------------------------------------------------------------------
-----
IPv4 routing is enable
IPv6 routing is enable using None
Max ECMP path : 1
-------------------------------------------------------------------------------------------
----
Ldp Synchronization is Disabled
-------------------------------------------------------------------------------------------
----
Overload
Current Status : not in overload
-------------------------------------------------------------------------------------------
----
Metric
Reference bandwidth: NA
L1 metric style: wide
L2 metric style: wide
-------------------------------------------------------------------------------------------
----
Graceful Restart
Helper Mode    : disabled
Current Status : not helping any neighbors
-------------------------------------------------------------------------------------------
-----
Timers
LSP Lifetime               : 1200
LSP Refresh                : 600
SPF initial wait           : 1000
SPF second wait            : 1000
SPF max wait               : 10000
LSP generation initial wait : 10
LSP generation second wait  : 1000
LSP generation max wait     : 5000
-------------------------------------------------------------------------------------------
-----
Route Preference
L1 internal : 15
L1 external : 160
L2 internal : 18
L2 external : 165
-------------------------------------------------------------------------------------------
-----
L1->L2 Summary Addresses Not configured
```

```
-------------------------------------------------------------------------------------------
-----
Instance Statistics
SPF run            : 29
Last SPF           : 2022-03-23T16:16:16.200Z
Partial SPF run    : 16
Last Partial SPF   : 2022-03-23T16:16:17.200Z
-------------------------------------------------------------------------------------------
-----
PDU Statistics
-------------------------------------------------------------------------------------------
----
| pdu-name |     received     |    processed     |     dropped      |       sent       |
+==========+==================+==================+==================+==================+
| LSP      | 460              | 457              | 3                | 528              |
| IIH      | 308              | 281              | 27               | 497              |
| CSNP     | 52               | 51               | 1                | 116              |
| PSNP     | 30               | 30               | 0                | 3                |
-------------------------------------------------------------------------------------------
-----
```

Example: IS-IS interface information

To display interface information for an IS-IS instance:

```
# show network-instance green_default protocols isis interface
-------------------------------------------------------------------------------------------
-----
Network Instance : green_default
Instance         : default
-------------------------------------------------------------------------------------------
-----
| Interface Name | Oper State | Level | Circuit id | Circuit type   | Ipv4 Metric L1/L2 |
 Ipv6 Metric L1/L2
===========================================================================================
=====
| ethernet-1/1.1 | up         | L1L2  | 2          | point-to-point | 10/10             |
 10/10
| ethernet-1/2.1 | up         | L1L2  | 2          | broadcast      | 10/10             |
 10/10
| ethernet-1/3.1 | up         | L1L2  | 3          | broadcast      | 10/10             |
 10/10
| ethernet-1/16. | up         | L1L2  | 4          | broadcast      | 10/10             |
 10/10
| 1              |            |       |            |                |                   |

| lo0.1          | up         | L1L2  | 5          | broadcast      | 0/0               |
 0/0
-------------------------------------------------------------------------------------------
-----
```

Example: IS-IS interface detail information

To display detail information for a specific IS-IS interface:

```
# show network-instance green_default protocols isis interface ethernet-1/1.1 detail
-------------------------------------------------------------------------------------------
Network Instance   : green_default
Instance           : default
-------------------------------------------------------------------------------------------
Interface-Name                     : ethernet-1/1.1
Status                             : IS-IS is admin enabled, oper up
Circuit                            : id 1 is broadcast and not passive
```

```
Hello Authentication Generate       : True
Hello Authentication Check Received : Strict
Hello Padding                       : disable
Csnp Interval                       : 10
Lsp Pacing                          : 100
Ldp Sync State                      : disabled
Ldp Sync Duration                   : 3274
-------------------------------------------------------------------------------------
Level                               : 1
Status                              : enabled
Adjacencies                         : 1
Hello Authentication Generate       : True
Hello Authentication Check Received : Strict
Priority                            : 64
Hello Interval                      : 9
Hello Multiplier                    : 3
Ipv4 Metric                         : 10
Ipv6 Metric                         : 10
-------------------------------------------------------------------------------------
Level                               : 2
Status                              : enabled
Adjacencies                         : 1
Hello Authentication Generate       : True
Hello Authentication Check Received : Strict
Priority                            : 64
Hello Interval                      : 9
Hello Multiplier                    : 3
Ipv4 Metric                         : 10
Ipv6 Metric                         : 10
-------------------------------------------------------------------------------------
```

Example: IS-IS adjacency information

To display IS-IS adjacency information:

```
# show network-instance default protocols isis adjacency
-------------------------------------------------------------------------------------
Network-instance    : default
IS-IS instance      : global
-------------------------------------------------------------------------------------
System-Id      Adj-Level  Interface        IPv4-Address  State  Uptime       Rem-Hold
<hostname1>    L1         ethernet-1/1.0   10.0.0.1      Up     0d 00:46:43  19s
<hostname1>    L2         ethernet-1/1.0   10.0.0.1      Up     0d 00:46:43  19s
-------------------------------------------------------------------------------------
Adjacencies: 2
-------------------------------------------------------------------------------------
```

Example: IS-IS link state database information

To display information for the IS-IS link state database:

```
# show network-instance green_default protocols isis database
-------------------------------------------------------------------------------------
Network-instance    : green_default
IS-IS instance      : default
-------------------------------------------------------------------------------------
| Level Number | Lsp Id               | Sequence | Checksum | Lifetime | Attributes |
+==============+======================+==========+==========+==========+============+
| 1            | 0010.0100.1001.00-00 | 0x33     | 0x1672   | 1167     | L1 L2      |
| 1            | 0020.0200.2002.00-00 | 0x35     | 0xd562   | 1014     | L1 L2      |
| 1            | 0030.0300.3003.00-00 | 0x38     | 0xf447   | 640      | L1 L2      |
| 1            | 0030.0300.3003.01-00 | 0x2f     | 0x4db6   | 1005     | L1 L2      |
| 1            | 0030.0300.3003.02-00 | 0x2e     | 0xd355   | 709      | L1 L2      |
```

```
| 1            | 0040.0400.4004.00-00 | 0x39     | 0x6f2a   | 638      | L1 L2      |
| 1            | 0040.0400.4004.01-00 | 0x2f     | 0xf0ef   | 822      | L1 L2      |
| 1            | 0040.0400.4004.02-00 | 0x2f     | 0xa5f8   | 999      | L1 L2      |
| 1            | 0050.0500.5005.00-00 | 0x38     | 0xfbbb   | 905      | L1 L2      |
| 1            | 0050.0500.5005.01-00 | 0x31     | 0x3937   | 745      | L1 L2      |
| 1            | 0050.0500.5005.02-00 | 0x2f     | 0xd19    | 657      | L1 L2      |
| 1            | 0060.0600.6006.00-00 | 0x37     | 0xf287   | 967      | L1 L2      |
| 1            | 0060.0600.6006.01-00 | 0x2f     | 0xadfb   | 753      | L1 L2      |
| 1            | 0060.0600.6006.02-00 | 0x2f     | 0x5f95   | 819      | L1 L2      |
| 1            | 0070.0700.7007.00-00 | 0x33     | 0x48dd   | 1058     | L1 L2      |
| 1            | 0070.0700.7007.01-00 | 0x2f     | 0xadb2   | 1164     | L1 L2      |
| 1            | 0070.0700.7007.02-00 | 0x2e     | 0xdf8e   | 852      | L1 L2      |
| 2            | 0010.0100.1001.00-00 | 0x3e     | 0xb92f   | 1150     | L1 L2      |
| 2            | 0010.0100.1001.00-01 | 0x3c     | 0x1875   | 818      | L1 L2      |
| 2            | 0020.0200.2002.00-00 | 0x41     | 0x540f   | 1177     | L1 L2      |
| 2            | 0020.0200.2002.00-01 | 0x3f     | 0x2db9   | 699      | L1 L2      |
| 2            | 0030.0300.3003.00-00 | 0x3c     | 0x302f   | 1058     | L1 L2      |
| 2            | 0030.0300.3003.00-01 | 0x3f     | 0x5150   | 668      | L1 L2      |
| 2            | 0030.0300.3003.01-00 | 0x30     | 0xb518   | 915      | L1 L2      |
| 2            | 0030.0300.3003.02-00 | 0x2f     | 0xe113   | 1035     | L1 L2      |
| 2            | 0040.0400.4004.00-00 | 0x3e     | 0xa17b   | 657      | L1 L2      |
| 2            | 0040.0400.4004.00-01 | 0x3e     | 0x260b   | 1018     | L1 L2      |
| 2            | 0040.0400.4004.01-00 | 0x30     | 0x1511   | 1066     | L1 L2      |
| 2            | 0040.0400.4004.02-00 | 0x2f     | 0x27a7   | 1035     | L1 L2      |
| 2            | 0050.0500.5005.00-00 | 0x41     | 0x59d6   | 608      | L1 L2      |
| 2            | 0050.0500.5005.00-01 | 0x44     | 0xf165   | 1110     | L1 L2      |
| 2            | 0050.0500.5005.01-00 | 0x33     | 0x7709   | 584      | L1 L2      |
| 2            | 0050.0500.5005.02-00 | 0x31     | 0xa74    | 657      | L1 L2      |
| 2            | 0060.0600.6006.00-00 | 0x3d     | 0xd5ed   | 903      | L1 L2      |
| 2            | 0060.0600.6006.00-01 | 0x44     | 0xdc97   | 666      | L1 L2      |
| 2            | 0060.0600.6006.01-00 | 0x30     | 0x9024   | 1145     | L1 L2      |
| 2            | 0060.0600.6006.02-00 | 0x30     | 0xbc66   | 1020     | L1 L2      |
| 2            | 0070.0700.7007.00-00 | 0x3a     | 0x81fd   | 862      | L1 L2      |
| 2            | 0070.0700.7007.00-01 | 0x3e     | 0xf82a   | 765      | L1 L2      |
| 2            | 0070.0700.7007.01-00 | 0x30     | 0xbad5   | 658      | L1 L2      |
| 2            | 0070.0700.7007.02-00 | 0x2f     | 0x1ecb   | 675      | L1 L2      |
+--------------+----------------------+----------+----------+----------+------------+
LSP Count: 41
-------------------------------------------------------------------------------------
```

To display information for a specific IS-IS link state database:

```
# show network-instance green_default protocols isis database 1
-------------------------------------------------------------------------------------
Network-instance    : green_default
IS-IS instance      : default
-------------------------------------------------------------------------------------
| Level Number | Lsp Id               | Sequence | Checksum | Lifetime | Attributes |
+==============+======================+==========+==========+==========+============+
| 1            | 0010.0100.1001.00-00 | 0x33     | 0x1672   | 1048     | L1 L2      |
| 1            | 0020.0200.2002.00-00 | 0x35     | 0xd562   | 894      | L1 L2      |
| 1            | 0030.0300.3003.00-00 | 0x39     | 0x7762   | 1179     | L1 L2      |
| 1            | 0030.0300.3003.01-00 | 0x2f     | 0x4db6   | 886      | L1 L2      |
| 1            | 0030.0300.3003.02-00 | 0x2f     | 0x4ace   | 1188     | L1 L2      |
| 1            | 0040.0400.4004.00-00 | 0x3a     | 0xcd9f   | 1135     | L1 L2      |
| 1            | 0040.0400.4004.01-00 | 0x2f     | 0xf0ef   | 703      | L1 L2      |
| 1            | 0040.0400.4004.02-00 | 0x2f     | 0xa5f8   | 879      | L1 L2      |
| 1            | 0050.0500.5005.00-00 | 0x38     | 0xfbbb   | 785      | L1 L2      |
| 1            | 0050.0500.5005.01-00 | 0x31     | 0x3937   | 625      | L1 L2      |
| 1            | 0050.0500.5005.02-00 | 0x30     | 0xb86e   | 1148     | L1 L2      |
| 1            | 0060.0600.6006.00-00 | 0x37     | 0xf287   | 847      | L1 L2      |
| 1            | 0060.0600.6006.01-00 | 0x2f     | 0xadfb   | 633      | L1 L2      |
| 1            | 0060.0600.6006.02-00 | 0x2f     | 0x5f95   | 700      | L1 L2      |
```

```
| 1            | 0070.0700.7007.00-00 | 0x33     | 0x48dd   | 938      | L1 L2      |
| 1            | 0070.0700.7007.01-00 | 0x2f     | 0xadb2   | 1044     | L1 L2      |
| 1            | 0070.0700.7007.02-00 | 0x2e     | 0xdf8e   | 733      | L1 L2      |
+--------------+----------------------+----------+----------+----------+------------+
LSP Count: 17
-------------------------------------------------------------------------------
```

5.4 Clearing IS-IS information

Procedure

To clear information for an IS-IS instance, use the **tools** commands below:

Example:

To clear statistics for an IS-IS instance running in a specified network instance:

```
# tools network-instance default protocols isis instance i1 statistics clear
```

Example

To clear link state database information for a level:

```
# tools network-instance default protocols isis instance i1 link-state-database clear
```

Example

To clear IS-IS adjacency information for an interface:

```
# tools network-instance default protocols isis instance i1 interface ethernet-1/1.1
 adjacencies clear
```

6 Protocol authentication

On the SR Linux, authentication of routing control messages for BGP, as well as other protocols such as LDP and IS-IS, is done using shared keys.

Message authentication between two routers involves sharing knowledge of a secret key and a cryptographic algorithm, such as MD5. This secret key, together with the message data, are used to generate a message digest. The message digest is added to each message transmitted by the sender and validated by the receiver, with the expectation that only a sender in possession of the secret key and algorithm details could generate the same message digest computed by the receiver of the message.

To limit exposure in the event a key is compromised, the secret key is changed at regular intervals using keys configured in a keychain. A keychain defines a list of one or more keys; each key is associated with a secret string, an algorithm identifier, and a start time.

When a protocol references a keychain for securing its messages with a set of peers, it uses the active key in the keychain with the most recent start time to generate the message digest in its sent messages, and it drops every received message that does not have an acceptable message digest.

6.1 Configuring protocol authentication

Procedure

To configure protocol authentication, you configure an authentication keychain at the system level and configure a protocol to use the keychain. All protocol authentication is done using keychains. If a protocol requires authentication with a single neighbor using a single key, the key is configured within a keychain, and the protocol references the keychain.

Example: Configure a keychain

The following example configures a keychain consisting of two keys.

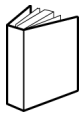
```
--{ candidate shared default }--[  ]--
# info system authentication
    system {
        authentication {
            keychain k1 {
                key 1 {
                    admin-state enable
                    algorithm md5
                    authentication-key ZcvSElJzJx/wBZ9biCt
                }
                key 2 {
                    admin-state enable
                    algorithm md5
                    authentication-key e7xdKlYO2DOm7v3IJv
                }
            }
        }
```

Example: Configure BGP to use the keychain for protocol authentication

The following example configures BGP to use the keys in the keychain above for protocol authentication:

```
--{ candidate shared default }--[  ]--
# info network-instance default protocols bgp authentication
    network-instance default {
        protocols {
            bgp {
                authentication {
                    keychain k1
                    }
                }
            }
        }
    }
```

Customer document and product support



Customer documentation

Customer documentation welcome page

Technical support

Product support portal

Documentation feedback

Customer documentation feedback