Configure Segment Routing for BGP

Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP) that allows you to create loop-free inter-domain routing between autonomous systems. An autonomous system is a set of routers under a single technical administration. Routers in an autonomous system can use multiple Interior Gateway Protocols (IGPs) to exchange routing information inside the autonomous system and an EGP to route packets outside the autonomous system.

This module provides the configuration information used to enable segment routing for BGP.

For additional information on implementing BGP on your Cisco ASR 9000 Series Router, see the Implementing BGP module in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

- Segment Routing for BGP, on page 1
- Configure BGP Prefix Segment Identifiers, on page 2
- Configure Segment Routing Egress Peer Engineering, on page 3
- Configure BGP Link-State, on page 4
- Example: Configuring SR-EPE and BGP-LS, on page 5
- Configure BGP Proxy Prefix SID, on page 7

Segment Routing for BGP

In a traditional BGP-based data center (DC) fabric, packets are forwarded hop-by-hop to each node in the autonomous system. Traffic is directed only along the external BPG (eBGP) multipath ECMP. No traffic engineering is possible.

In an MPLS-based DC fabric, the eBGP sessions between the nodes exchange BGP labeled unicast (BGP-LU) network layer reachability information (NLRI). An MPLS-based DC fabric allows any leaf (top-of-rack or border router) in the fabric to communicate with any other leaf using a single label, which results in higher packet forwarding performance and lower encapsulation overhead than traditional BGP-based DC fabric. However, since each label value might be different for each hop, an MPLS-based DC fabric is more difficult to troubleshoot and more complex to configure.

BGP has been extended to carry segment routing prefix-SID index. BGP-LU helps each node learn BGP prefix SIDs of other leaf nodes and can use ECMP between source and destination. Segment routing for BGP simplifies the configuration, operation, and troubleshooting of the fabric. With segment routing for BGP, you can enable traffic steering capabilities in the data center using a BGP prefix SID.
Configure BGP Prefix Segment Identifiers

Segments associated with a BGP prefix are known as BGP prefix SIDs. The BGP prefix SID is global within a segment routing or BGP domain. It identifies an instruction to forward the packet over the ECMP-aware best-path computed by BGP to the related prefix. The BGP prefix SID is manually configured from the segment routing global block (SRGB) range of labels.

Each BGP speaker must be configured with an SRGB using the `segment-routing global-block` command. See the About the Segment Routing Global Block section for information about the SRGB.

**Note**
Because the values assigned from the range have domain-wide significance, we recommend that all routers within the domain be configured with the same range of values.

To assign a BGP prefix SID, first create a routing policy using the `set label-index index` attribute, then associate the index to the node.

**Note**
A routing policy with the `set label-index` attribute can be attached to a network configuration or redistribute configuration. Other routing policy language (RPL) configurations are possible. For more information on routing policies, refer to the "Implementing Routing Policy" chapter in the Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide.

**Example**
The following example shows how to configure the SRGB, create a BGP route policy using a $SID parameter and `set label-index` attribute, and then associate the prefix-SID index to the node.

```
RP/0/RSP0/CPU0:router(config)# segment-routing global-block 16000 23999
RP/0/RSP0/CPU0:router(config)# route-policy SID($SID)
RP/0/RSP0/CPU0:router(config-rpl)# set label-index $SID
RP/0/RSP0/CPU0:router(config-rpl)# end policy
RP/0/RSP0/CPU0:router(config)# router bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1
RP/0/RSP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-bgp-af)# network 1.1.1.3/32 route-policy SID(3)
RP/0/RSP0/CPU0:router(config-bgp-af)# allocate-label all
RP/0/RSP0/CPU0:router(config-bgp-af)# commit
RP/0/RSP0/CPU0:router(config-bgp-af)# end
```

```
RP/0/RSP0/CPU0:router# show bgp 1.1.1.3/32
BGP routing table entry for 1.1.1.3/32
Versions:
Process bRIB/RIB SendTblVer
Speaker 74 74
Local Label=16003
Last Modified: Sep 29 19:52:18.155 for 00:07:22
Paths: (1 available, best #1)
   Advertised to update-groups (with more than one peer):
       0.2
```
Configure Segment Routing Egress Peer Engineering

Segment routing egress peer engineering (EPE) uses a controller to instruct an ingress provider edge, or a content source (node) within the segment routing domain, to use a specific egress provider edge (node) and a specific external interface to reach a destination. BGP peer SIDs are used to express source-routed inter-domain paths.

The controller learns the BGP peer SIDs and the external topology of the egress border router through BGP-LS EPE routes. The controller can program an ingress node to steer traffic to a destination through the egress node and peer node using BGP labeled unicast (BGP-LU).

EPE functionality is only required at the EPE egress border router and the EPE controller.

This task explains how to configure segment routing EPE on the EPE egress node.

**SUMMARY STEPS**

1. `router bgp as-number`
2. `neighbor ip-address`
3. `remote-as as-number`
4. `egress-engineering`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>router bgp as-number</code></td>
</tr>
<tr>
<td>Example:</td>
<td>Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.</td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config)# router bgp 1</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>neighbor ip-address</code></td>
</tr>
<tr>
<td>Example:</td>
<td>Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.</td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp)# neighbor 192.168.1.3</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>remote-as as-number</code></td>
</tr>
<tr>
<td>Example:</td>
<td>Creates a neighbor and assigns a remote autonomous system number to it.</td>
</tr>
<tr>
<td></td>
<td><code>RP/0/RSP0/CPU0:router(config-bgp)# remote-as 192.168.1.3</code></td>
</tr>
</tbody>
</table>
Configure BGP Link-State

BGP Link-State (LS) is an Address Family Identifier (AFI) and Sub-address Family Identifier (SAFI) defined to carry interior gateway protocol (IGP) link-state database through BGP. BGP LS delivers network topology information to topology servers and Application Layer Traffic Optimization (ALTO) servers. BGP LS allows policy-based control to aggregation, information-hiding, and abstraction. BGP LS supports IS-IS and OSPFv2.

For segment routing, the following attributes have been added to BGP LS:

- **Node**—Segment routing capability (including SRGB range) and algorithm
- **Link**—Adjacency SID and LAN adjacency SID
- **Prefix**—Prefix SID and segment routing mapping server (SRMS) prefix range

The following example shows how to exchange link-state information with a BGP neighbor:

```bash
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.0.0.2
RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 1
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family link-state link-state
RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# exit
```

IGP Extensions

A given BGP node may have connections to multiple, independent routing domains; IGP link state distribution into BGP has been added for both OSPF and ISIS protocols to enable that node to pass this information, in a similar fashion, on to applications that desire to build paths spanning or including these multiple domains.

To distribute ISIS link-state data using BGP LS, use the **distribute bgp-ls** command in router configuration mode:

```bash
RP/0/RSP0/CPU0:router# configure
```
To distribute OSPFv2 and OSPFv3 link-state data using BGP LS, use the `distribute bgp-ls` command in router configuration mode.

```bash
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router ospf 100
RP/0/RSP0/CPU0:router(config-ospf)# distribute bgp-ls instance-id 32 throttle 10
```

**Example: Configuring SR-EPE and BGP-LS**

In the following figure, segment routing is enabled on autonomous system AS1 with ingress node A and egress nodes B and C. In this example, we configure EPE on egress node C.

**Figure 1: Topology**

![Topology Diagram]

#### Step 1
Configure node C with EPE for eBGP peers D and E.

**Example:**

```bash
RP/0/RSP0/CPU0:router_C(config)# router bgp 1
RP/0/RSP0/CPU0:router_C(config-bgp)# neighbor 192.168.1.3
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# remote-as 3
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# description to E
egress-engineering
RP/0/RSP0/CPU0:router_C(config-bgp-nbr)# address-family ipv4 unicast
route-policy bgp_in in
route-policy bgp_out out
exit
exit
```

**Step 2**
Configure node C to advertise peer node SIDs to the controller using BGP-LS.
Example: Configuring SR-EPE and BGP-LS

Step 3  Commit the configuration.

Example:

```
RP/0/RSP0/CPU0:router_C(config)# commit
```

Step 4  Verify the configuration.

Example:

```
RP/0/RSP0/CPU0:router_C(config)# show bgp egress-engineering
```

The output shows that node C has allocated peer SIDs for each eBGP peer.

Example:

```
RP/0/RSP0/CPU0:router_C# show mpls forwarding labels 24002 24003
```

<table>
<thead>
<tr>
<th>Label</th>
<th>Outgoing Prefix</th>
<th>Outgoing</th>
<th>Next Hop</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>24002</td>
<td>Unlabelled No ID</td>
<td>Te0/3/0/0</td>
<td>192.168.1.2</td>
<td>0</td>
</tr>
<tr>
<td>24003</td>
<td>Unlabelled No ID</td>
<td>Te0/1/0/0</td>
<td>192.168.1.3</td>
<td>0</td>
</tr>
</tbody>
</table>
The output shows that node C installed peer node SIDs in the Forwarding Information Base (FIB).

---

**Configure BGP Proxy Prefix SID**

To support segment routing, Border Gateway Protocol (BGP) requires the ability to advertise a segment identifier (SID) for a BGP prefix. A BGP-Prefix-SID is the segment identifier of the BGP prefix segment in a segment routing network. BGP prefix SID attribute is a BGP extension to signal BGP prefix-SIDs. However, there may be routers which do not support BGP extension for segment routing. Hence, those routers also do not support BGP prefix SID attribute and an alternate approach is required.

BGP proxy prefix SID feature allows you to attach BGP prefix SID attributes for remote prefixes learnt from BGP labeled unicast (LU) neighbours which are not SR-capable and propagate them as SR prefixes. This allows an LSP towards non SR endpoints to use segment routing global block in a SR domain. Since BGP proxy prefix SID uses global label values it minimizes the use of limited resources such as ECMP-FEC and provides more scalability for the networks.

BGP proxy prefix SID feature is implemented using the segment routing mapping server (SRMS). SRMS allows the user to configure SID mapping entries to specify the prefix-SIDs for the prefixes. The mapping server advertises the local SID-mapping policy to the mapping clients. BGP acts as a client of the SRMS and uses the mapping policy to calculate the prefix-SIDs.

**Configuration Example:**

This example shows how to configure the BGP proxy prefix SID feature for the segment routing mapping server.

```
RP/0/RSP0/CPU0:router(config)# segment-routing
RP/0/RSP0/CPU0:router(config-sr)# mapping-server
RP/0/RSP0/CPU0:router(config-sr-ms)# prefix-sid-map
RP/0/RSP0/CPU0:router(config-sr-ms-map)# address-family ipv4
RP/0/RSP0/CPU0:router(config-sr-ms-map-af)# 1.1.1.1/32 10 range 200
RP/0/RSP0/CPU0:router(config-sr-ms-map-af)# 192.168.64.1/32 400 range 300
```

This example shows how to configure the BGP proxy prefix SID feature for the segment-routing mapping client.

```
RP/0/RSP0/CPU0:router(config)# router bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# address-family ip4 unicast
RP/0/RSP0/CPU0:router(config-bgp-af)# segment-routing prefix-sid-map
```

**Verification**

These examples show how to verify the BGP proxy prefix SID feature.

```
RP/0/RSP0/CPU0:router# show segment-routing mapping-server prefix-sid-map ipv4 detail
Prefix 1.1.1.1/32
  SID Index: 10
  Range: 200
  Last Prefix: 1.1.1.200/32
  Last SID Index: 209
  Flags:
```
Number of mapping entries: 1

RP/0/RSP0/CPU0:router# **show bgp ipv4 labeled-unicast 192.168.64.1/32**

BGP routing table entry for 192.168.64.1/32

Versions:
- Process: bRIB/RIB: SendTblVer
  - Speaker: 117
  - 117
- Local Label: 16400
- Last Modified: Oct 25 01:02:28.562 for 00:11:45
- Paths: (2 available, best #1)
  - 201.1.1.1
  - 201.1.1.1
  - Advertised to peers (in unique update groups):
    - 20.0.101.1 from 20.0.101.1 (20.0.101.1)
      - Received Label 61
      - Origin IGP, localpref 100, valid, internal, best, group-best, multipath, labeled-unicast
      - Received Path ID 0, Local Path ID 0, version 117
      - Prefix SID Attribute Size: 7
      - Label Index: 1

RP/0/RSP0/CPU0:router# **show route ipv4 unicast 192.68.64.1/32 detail**

Routing entry for 192.68.64.1/32

Known via "bgp 65000", distance 200, metric 0, [ei]-bgp, labeled SR, type internal

Installed Oct 25 01:02:28.583 for 00:20:09

Routing Descriptor Blocks
- 20.0.101.1, from 20.0.101.1, BGP multi path
  - Route metric is 0
  - Label: 0x3d (61)
  - Tunnel ID: None
  - Binding Label: None
  - Extended communities count: 0
  - NHID:0x0 (Ref:0)
  - Route version is 0x6 (6)
  - Local Label: 0x3e81 (16400)
  - IP Precedence: Not Set
  - QoS Group ID: Not Set
  - Flow-tag: Not Set
  - Fwd-class: Not Set
  - Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_LOCAL
  - Download Priority 4, Download Version 242
  - No advertising protos.

RP/0/RSP0/CPU0:router# **show cef ipv4 192.168.64.1/32 detail**

192.168.64.1/32, version 476, labeled SR, drop adjacency, internal 0x500001 0x80 (ptr 0x71ae7e78) reference count 3, flags 0x7a, source rib (7), 0 backups
- 2 type 5 flags 0x88401 (0x722eb450) ext 0x0 (0x0)
- LW-LDI[type=5, refc=3, ptr=0x71c11590, sh-ldi=0x722eb450]
- via 20.0.101.1/32, 0 dependencies, recursive, bgp-ext [flags 0x6020]
- path-idx 0 NHID 0x0 [0x7129a294 0x0]
- recursion-via=-/32
- unresolved
- local label 16400
RP/0/RSP0/CPU0:router# **show bgp labels**
BGP router identifier 2.1.1.1, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000  RD version: 245
BGP main routing table version 245
BGP NSR Initial initsync version 16 (Reached)
BGP NSR/ISSU Sync-Group versions 245/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Rcvd Label</th>
<th>Local Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt;1.1.1.1/32</td>
<td>1.1.1.1</td>
<td>3</td>
<td>16100</td>
</tr>
<tr>
<td>*&gt; 2.1.1.1/32</td>
<td>0.0.0.0</td>
<td>nolabel</td>
<td>3</td>
</tr>
<tr>
<td>*&gt; 192.68.64.1/32</td>
<td>20.0.101.1</td>
<td>2</td>
<td>16400</td>
</tr>
<tr>
<td>*&gt; 192.68.64.2/32</td>
<td>20.0.101.1</td>
<td>2</td>
<td>16401</td>
</tr>
</tbody>
</table>
Configure BGP Proxy Prefix SID