Configuring BGP

BGP Overview

Table 1: Feature History

Feature Name	Release Information	Feature Description
Configuring BGP	Cisco IOS XR Release 7.11.1	BGP routers from a TCP connection between peer routers to exchange network reachability information to open and confirm the connection parameters. BGP on NCS 1010 is enabled only on management port enabling the customer to manage the Dynamic Circuit Network(DCN) connectivity.

BGP uses TCP as its transport protocol. Two BGP routers form a TCP connection between one another (peer routers) and exchange messages to open and confirm the connection parameters.

BGP routers exchange network reachability information. This information is mainly an indication of the full paths (BGP autonomous system numbers) that a route should take to reach the destination network. This information helps construct a graph that shows which autonomous systems are loop free and where routing policies can be applied to enforce restrictions on routing behavior.

Any two routers forming a TCP connection to exchange BGP routing information are called peers or neighbors. BGP peers initially exchange their full BGP routing tables. After this exchange, incremental updates are sent as the routing table changes. BGP keeps a version number of the BGP table, which is the same for all its BGP peers. The version number changes whenever BGP updates the table due to routing information changes. Keepalive packets are sent to ensure that the connection is alive between the BGP peers and notification packets are sent in response to error or special conditions.

Note   ASN change for the BGP process is not currently supported via **commit replace**.

Prerequisites for Implementing BGP

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

BGP Router Identifier

For BGP sessions between neighbors to be established, BGP must be assigned a router ID. The router ID is sent to BGP peers in the OPEN message when a BGP session is established.

BGP attempts to obtain a router ID in the following ways (in order of preference):

- By means of the address configured using the **bgp router-id** command in router configuration mode.

- By using the highest IPv4 address on a loopback interface in the system if the router is booted with saved loopback address configuration.

- By using the primary IPv4 address of the first loopback address that gets configured if there are not any in the saved configuration.

If none of these methods for obtaining a router ID succeeds, BGP does not have a router ID and cannot establish any peering sessions with BGP neighbors. In such an instance, an error message is entered in the system log, and the **show bgp summary** command displays a router ID of 0.0.0.0.

After BGP has obtained a router ID, it continues to use it even if a better router ID becomes available. This usage avoids unnecessary flapping for all BGP sessions. However, if the router ID currently in use becomes invalid (because the interface goes down or its configuration is changed), BGP selects a new router ID (using the rules described) and all established peering sessions are reset.

Note   We strongly recommend that the **bgp router-id** command is configured to prevent unnecessary changes to the router ID (and consequent flapping of BGP sessions).

Configuring BGP

Figure 1: BGP Topology
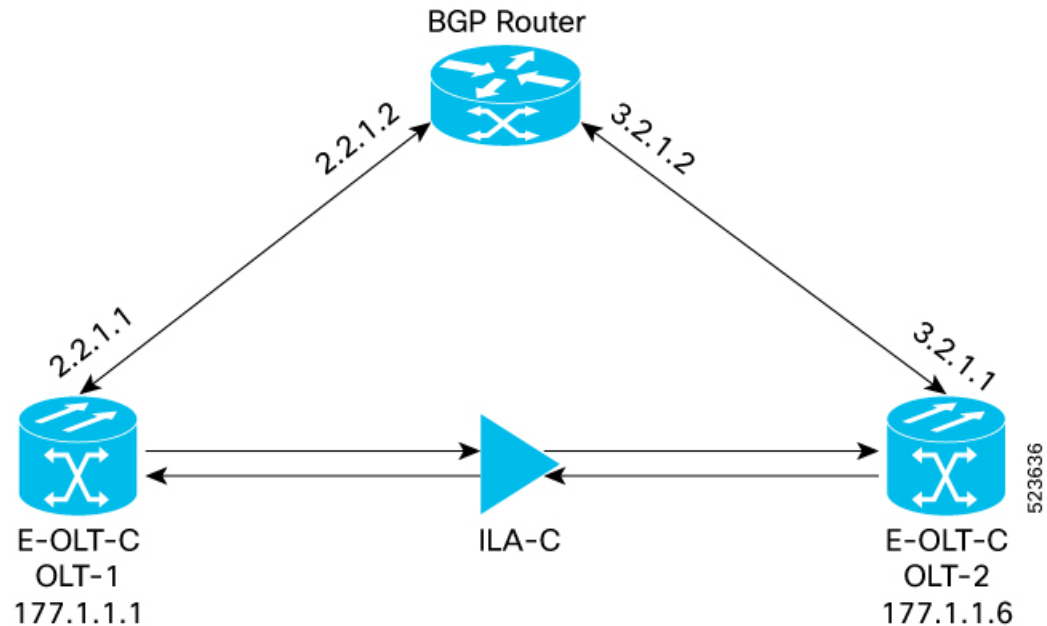

```
config
route-policy pass-all
end-policy
commit

RP/0/RP0/CPU0:OLT-1#conf
Fri Feb 23 09:32:29.216 IST
router bgp 1
bgp router-id 177.1.1.1
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 2.2.1.2
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:54:43.439 IST
router bgp 1
 bgp router-id 177.1.1.1
 address-family ipv4 unicast
  redistribute connected
```

```
 redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 2.2.1.2
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!


RP/0/RP0/CPU0:OLT-2#conf
Fri Feb 23 09:32:29.216 IST
router bgp 1
 bgp router-id 177.1.1.6
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 3.2.1.2
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:59:12.210 IST
router bgp 1
 bgp router-id 177.1.1.6
 address-family ipv4 unicast
  redistribute connected
  redistribute ospf 1 route-policy pass-all
 !
 address-family vpnv4 unicast
 !
 neighbor 3.2.1.2
  remote-as 100
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!

show route
route   router-id


RP/0/RP0/CPU0:BGP-ROUTER#config
Fri Feb 23 09:32:29.216 IST
router bgp 100
 bgp router-id 1.1.1.99
 address-family ipv4 unicast
  redistribute connected
 !
```

```
 neighbor 2.2.1.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 3.2.1.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
commit

show running-config router bgp
Thu Jan 11 15:59:58.059 IST
router bgp 100
 bgp router-id 1.1.1.99
 address-family ipv4 unicast
  redistribute connected
 !
 neighbor 2.2.1.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
 neighbor 3.2.1.1
  remote-as 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 !
!
```

Note Autonomous system numbers 2.0 standard supports 0 to 65535, and autonomous system numbers standard 4.0 supports 65536 onwards.