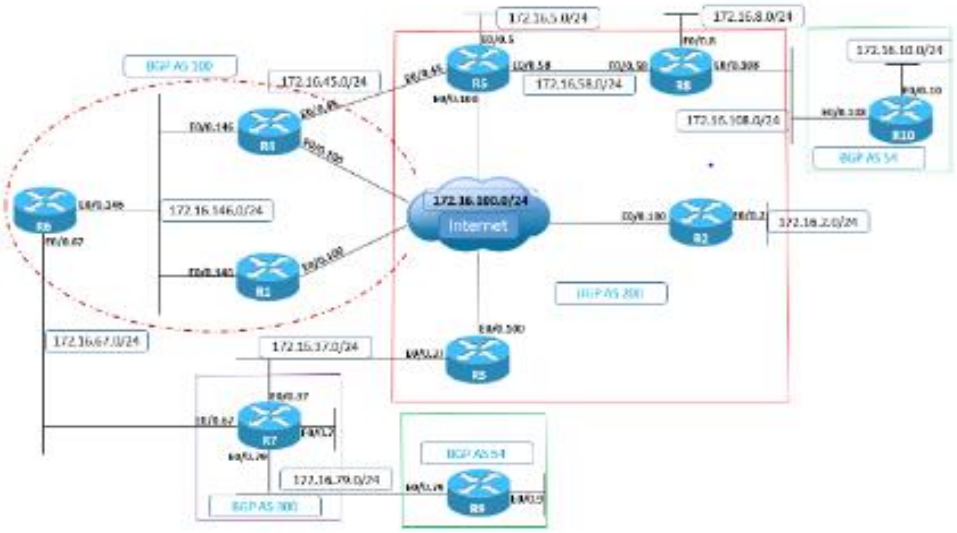
BGP

Lab 01: Establishing iBGP Peerings

Topology



Task

- Configure BGP R1 - R8 routers using AS 100.
- Create a full mesh of IBGP peerings between these devices without using their Loopback0 interfaces.
- Advertise the Loopback0 prefixes of these devices into BGP.
- Ensure full IPv4 reachability for Loopback0 prefixes of R1 - R8 routers.

Explanation

The first step in any BGP configuration is always to establish peering relationships between the BGP speaking devices. Recall that because BGP does not have its own transport protocol, underlying IGP reachability must already be established to allow the TCP port 179 sessions to be successful between neighbors. BGP is a normal TCP application, which means that a TCP client initiates the session to the TCP server with a SYN packet going to the well-known port of 179. If the BGP server is configured to accept the session, a reply with SYN/ACK comes from port 179 back to the client, going to the high port number generated by the client. If both BGP

peers attempt to establish the connection at the same time, RFC 4271 (A Border Gateway Protocol 4) defines a "BGP Connection Collision Detection" mechanism, in which essentially the session originated from the device with the higher BGP router-id is maintained, and the secondary session is dropped.The below debug output shows the step-by-step formation of the iBGP peering between R1 and R2. Note that access-list 100 is used to filter the debug output and only show the output pertinent to the BGP session between R1 and R2:

```
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 100 permit tcp any host 172.16.0.2
R1(config)#access-list 100 permit tcp host 172.16.0.2 any
R1(config)#do debug ip packet detail 100
IP packet debugging is on (detailed) for access list 100
R1(config)#router bgp 100
R1(config-router)#neighbor 172.16.0.2 remote-as 100
R1(config-router)#
IP: s=172.16.0.1 (local), d=172.16.0.2, len 44, local feature
   TCP src=11712, dst=179, seq=4030024239, ack=0, win=16384 SYN
```

R1 is configured with the neighbor statement pointing toward R2, and R2 is already configured with the neighbor statement pointing toward R1. A SYN is sent from R1 to initiate the session. Notice the source port of 11712 and the destination port of 179:

```
IP: s=172.16.0.2 (Tunnel0), d=172.16.0.1, len 40, enqueue feature
TCP src=179, dst=11712, seq=3286835183, ack=4030024297, win=16327 ACK SYN,
```

Because R2 has a neighbor statement pointing back toward the address 172.16.0.1, a reply of ACK/SYN is received, with the source port of 179 and the destination port randomly generated by R1, 11712. These two steps indicate that R1 is the client and R2 is the server:

```
IP: s=172.16.0.1 (local), d=172.16.0.2, len 40, local feature
TCP src=11712, dst=179, seq=4030024441, ack=3286835384, win=16183 ACK,
```

R1 replies with ACK, completing the 3-way TCP handshake and opening the session for BGP attribute negotiation. Only after necessary parameters are correctly negotiated, such as remote-

as numbers, authentication, and address-family support, will the BGP session actually be declared up:

```
 IP: s=172.16.0.2 (Tunnel0), d=172.16.0.1, len 142, enqueue feature
TCP src=179, dst=11712, seq=3286835282, ack=4030024339, win=16285 ACK PSH
%BGP-5-ADJCHANGE: neighbor 172.16.0.2 Up
```

The details of the peering negotiation between R1 and R2, such as the router-id and timers, can be seen below. The neighbor capability of "Address family IPv4 Unicast: advertised and received" means that by default, they can exchange IPv4 prefixes, but not other such as IPv6 Unicast or VPNv4/VPNv6. The details for IPv4 unicast address family show that one prefix has been received from the neighbor and one prefix has been advertised to the neighbor:

```
R1#show ip bgp neighbor 172.16.0.2

BGP neighbor is 172.16.0.2,  remote AS 100, internal link

BGP version 4, remote router ID 10.1.2.2

BGP state = Established, up for 00:23:43

Last read 00:00:44, last write 00:00:00, hold time is 180, keepalive interval is 60
seconds

Neighbor sessions:

1 active, is not multisession capable (disabled)

Neighbor capabilities:

Route refresh: advertised and received(new)

Four-octets ASN Capability: advertised and received

Address family IPv4 Unicast: advertised and received

Enhanced Refresh Capability: advertised and received

Multisession Capability:

Stateful switchover support enabled: NO for session 1

Message statistics:

InQ depth is 0

OutQ depth is 0
```

```
                Sent       Rcvd
Opens:             1          1
Notifications:     0          0
Updates:           2          2
Keepalives:       28         27
Route Refresh:     0          0
Total:            33         32
Default minimum time between advertisement runs is 0 seconds

For address family: IPv4 Unicast
Session: 172.16.0.2
BGP table version 15, neighbor version 15/0
Output queue size : 0
Index 1, Advertise bit 0
1 update-group member
Slow-peer detection is disabled
Slow-peer split-update-group dynamic is disabled
Interface associated: (none)
                    Sent       Rcvd
  Prefix activity:  ----       ----
  Prefixes Current:    1          1 (Consumes 120 bytes)
  Prefixes Total:      9          1
```

```
8 network entries using 1984 bytes of memory
8 path entries using 960 bytes of memory
2/2 BGP path/bestpath attribute entries using 480 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3424 total bytes of memory
BGP activity 11/3 prefixes, 11/3 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.0.2      4   100       7       7       15    0    0 00:37:24        1
172.16.0.3      4   100       7       7       15    0    0 01:14:33        1
172.16.0.4      4   100       7       7       15    0    0 01:14:28        1
172.16.0.5      4   100       7       7       15    0    0 01:14:23        1
172.16.58.8     4   100       5       6       15    0    0 01:13:43        1
172.16.67.7     4   100       5       6       15    0    0 01:14:06        1
172.16.146.6    4   100       7       7       15    0    0 01:14:20        1
```

Note that for devices running multiple address-families, such as IPv4 unicast VPNv4/VPNv6, show commands are expressed as show bgp [AFI] [SAFI] [args], such as the show bgp ipv4 unicast summary seen below. Although the output is the same as the above show ip bgp summary, it can quickly become hard to tell what output you are viewing in larger-scale BGP deployments without this logical separation.

```
R1#show bgp ipv4 unicast summary
BGP router identifier 10.1.1.1, local AS number 100
BGP table version is 15, main routing table version 15
8 network entries using 1984 bytes of memory
```

```
  Implicit Withdraw:         8        0
  Explicit Withdraw:         0        0
  Used as bestpath:        n/a        1
  Used as multipath:       n/a        0

                               Outbound    Inbound
Local Policy Denied Prefixes:  --------    -------
  Bestpath from this peer:                 8        n/a
  Bestpath from iBGP peer:                 30       n/a
  Total:                                   38       0
Number of NLRIs in the update sent: max 1, min 0
```

The TTL of the outbound session is set to 255, because this is an IBGP session. This means that the iBGP neighbors need not be directly connected, as long as IGP reachability exists between them. EBGP sessions have a TTL of 1 by default, which means that neighbors must be directly connected, unless further configuration is applied. Also note the "Local port: 11712" and "Foreign port: 179." These essentially mean the source and destination ports from R1's perspective, which again enforces the notion that R1 is the client for this session and R2 is the server:

```
Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 172.16.0.1, Local port: 11712
Foreign host: 172.16.0.2, Foreign port: 179
```

The following show ip bgp summary output shows a concise aggregation of all configured neighbors, with the important fields being the local AS, router-id, table size in both prefixes and memory, peer addresses, remote ASs, neighbor uptime, and number of prefixes learned:

```
R1#show ip bgp summary
BGP router identifier 10.1.1.1, local AS number 100
BGP table version is 15, main routing table version 15
```

```
8 path entries using 960 bytes of memory
2/2 BGP path/bestpath attribute entries using 480 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3424 total bytes of memory
BGP activity 11/3 prefixes, 11/3 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
172.16.0.2      4          100      50      51       15    0    0 00:39:56        1
172.16.0.3      4          100      88      91       15    0    0 01:17:06        1
172.16.0.4      4          100      88      89       15    0    0 01:17:00        1
172.16.0.5      4          100      88      91       15    0    0 01:16:55        1
172.16.58.8     4          100      88      91       15    0    0 01:16:15        1
172.16.67.7     4          100      88      92       15    0    0 01:16:38        1
172.16.146.6    4          100      87      90       15    0    0 01:16:52        1
```

After the peering relationships are established, the actual BGP prefixes learned can be viewed in the BGP table with show ip bgp or show bgp ipv4 unicast. This output is crucial to completely understand, especially when multiple paths to the same destination exist:

```
R1#show ip bgp
BGP table version is 15, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.1/32     0.0.0.0                  0         32768 i
 *>i 10.1.2.2/32     172.16.0.2               0    100      0 i
 *>i 10.1.3.3/32     172.16.0.3               0    100      0 i
 *>i 10.1.4.4/32     172.16.0.4               0    100      0 i
 *>i 10.1.5.5/32     172.16.0.5               0    100      0 i
 *>i 10.1.6.6/32     172.16.146.6             0    100      0 i
 *>i 10.1.7.7/32     172.16.67.7              0    100      0 i
 *>i 10.1.8.8/32     172.16.58.8              0    100      0 i
!
R1#show ip route bgp

      10.1.0.0/32 is subnetted, 8 subnets
B        10.1.2.2 [200/0] via 172.16.0.2, 00:42:08
B        10.1.3.3 [200/0] via 172.16.0.3, 01:18:11
B        10.1.4.4 [200/0] via 172.16.0.4, 01:18:07
B        10.1.5.5 [200/0] via 172.16.0.5, 01:18:01
B        10.1.6.6 [200/0] via 172.16.146.6, 01:17:54
B        10.1.7.7 [200/0] via 172.16.67.7, 01:17:43
B        10.1.8.8 [200/0] via 172.16.58.8, 01:17:43
```

```
!
R1#show ip route 172.16.146.6
Routing entry for 172.16.146.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via eigrp 100
  Routing Descriptor Blocks:
  * directly connected, via Eth0/0.146
      Route metric is 0, traffic share count is 1
```

In the above case, recursion toward 10.1.7.7/32 continues until the outgoing interface Eth0/0.146 is found. Note that unless route-recursion toward the next-hop of a BGP prefix is successful, the route cannot be considered for best path selection, which also implies that it cannot be installed in the IP routing table or advertised to any other BGP peer. This issue will be explored in detail in the coming tasks. Also note that in this task, a full-mesh of iBGP peerings is established. This is because of the design requirement that an iBGP learned route cannot be advertised to another iBGP neighbor to prevent routing loops, unless exceptions such as route-reflection or confederation are implemented. The result is that the only routes advertised to the other BGP peers are the local Loopback0 prefixes, but not any of the routes learned from the other neighbors. This also implies that in this design, if any individual peering breaks, connectivity between prefixes advertised by those peers also breaks.

```
R1#show ip bgp neighbors 172.16.0.2 advertised-routes
BGP table version is 15, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

One of the most important fields in the above outputs is the next-hop value. Note that this field is set to the peering address for the neighbor from which the route is learned. For example, the prefix 10.1.7.7/32 is via the next-hop 172.16.67.7. To reach this prefix, a recursive lookup must now be performed on the next-hop until an outgoing interface is found:

```
R1#show ip route 10.1.7.7
Routing entry for 10.1.7.7/32
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 172.16.67.7 01:18:46 ago
  Routing Descriptor Blocks:
  *172.16.67.7, from 172.16.67.7, 01:18:46 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
  MPLS label: none
!
R1#show ip route 172.16.67.7
Routing entry for 172.16.67.0/24
  Known via "eigrp 100", distance 90, metric 3072, type internal
  Redistributing via eigrp 100
  Last update from 172.16.146.6 on Eth0/0.146, 02:51:42 ago
  Routing Descriptor Blocks:
  * 172.16.146.6, from 172.16.146.6, 02:51:42 ago, via Eth0/0.146
      Route metric is 3072, traffic share count is 1
      Total delay is 20 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

```
   Network          Next Hop            Metric LocPrf Weight Path
*>  10.1.1.1/32      0.0.0.0                  0         32768 i

Total number of prefixes 1
```

Configuration

R1:

```
router bgp 100
 network 10.1.1.1 mask 255.255.255.255
 neighbor 172.16.0.2 remote-as 100
 neighbor 172.16.0.3 remote-as 100
 neighbor 172.16.0.4 remote-as 100
 neighbor 172.16.0.5 remote-as 100
 neighbor 172.16.58.8 remote-as 100
 neighbor 172.16.67.7 remote-as 100
 neighbor 172.16.146.6 remote-as 100
```

R2:

```
router bgp 100
 network 10.1.2.2 mask 255.255.255.255
 neighbor 172.16.0.1 remote-as 100
 neighbor 172.16.0.3 remote-as 100
```

```
neighbor 172.16.0.4 remote-as 100
neighbor 155.1.0.5 remote-as 100
neighbor 172.16.37.7 remote-as 100
neighbor 172.16.58.8 remote-as 100
neighbor 172.16.146.6 remote-as 100
```

R3:

```
router bgp 100
network 10.1.3.3 mask 255.255.255.255
neighbor 172.16.0.1 remote-as 100
neighbor 172.16.0.2 remote-as 100
neighbor 172.16.0.4 remote-as 100
neighbor 172.16.0.5 remote-as 100
neighbor 172.16.37.7 remote-as 100
neighbor 172.16.58.8 remote-as 100
neighbor 172.16.146.6 remote-as 100
```

R4:

```
router bgp 100
network 10.1.4.4 mask 255.255.255.255
neighbor 172.16.0.1 remote-as 100
neighbor 172.16.0.2 remote-as 100
```

```
neighbor 172.16.0.5 remote-as 100
neighbor 172.16.58.8 remote-as 100
neighbor 172.16.67.7 remote-as 100
neighbor 172.16.146.1 remote-as 100
neighbor 172.16.146.4 remote-as 100
```

R7:

```
router bgp 100
network 10.1.7.7 mask 255.255.255.255
neighbor 172.16.0.5 remote-as 100
neighbor 172.16.23.2 remote-as 100
neighbor 172.16.37.3 remote-as 100
neighbor 172.16.58.8 remote-as 100
neighbor 172.16.67.6 remote-as 100
neighbor 172.16.146.1 remote-as 100
neighbor 172.16.146.4 remote-as 100
```

R8:

```
router bgp 100
network 10.1.8.8 mask 255.255.255.255
neighbor 172.16.0.1 remote-as 100
neighbor 172.16.0.2 remote-as 100
neighbor 172.16.0.3 remote-as 100
```

```
neighbor 172.16.0.4 remote-as 100
neighbor 172.16.37.7 remote-as 100
neighbor 172.16.58.5 remote-as 100
neighbor 172.16.146.6 remote-as 100
```