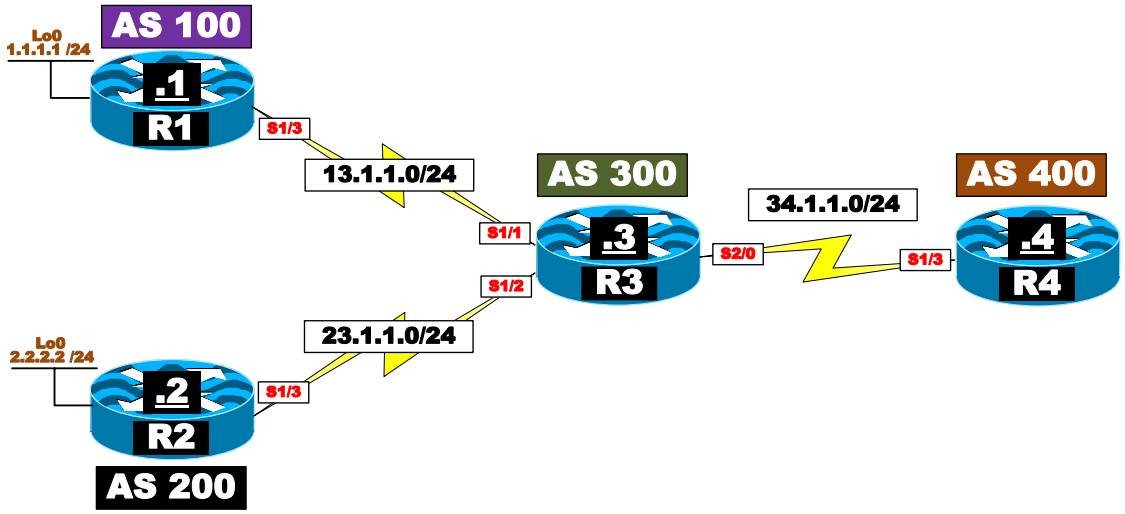
Lab 1 – BGP Automatic-tag, AS-path tag and Table-map



Task 1

Configure BGP based on the following requirements:

- Configure R1 in AS 100 to establish a BGP peer session with R3 in AS 300. R1 should be configured to advertise its loopback 0 interface in this AS.
- Configure R2 in AS 200 to establish a BGP peer session with R3 in AS 300. R2 should be configured to advertise its loopback 0 interface in this AS.
- Configure R3 to advertise a default route to R1 in AS 100 and R2 in AS 200.
- Configure R4 in AS 400. This router should NOT establish a peer session with any of the routers in this topology. The router-id of this router should be configured to be 4.4.4.4.

On R1:

```
R1(config)#router bgp 100
R1(config-router)#no au
R1(config-router)#neighbor 13.1.1.3 remote-as 200
R1(config-router)#netw 1.1.1.0 mask 255.255.255.0
```

On R2:

```
R2(config)#router bgp 200
R2(config-router)#no au
R2(config-router)#neighbor 23.1.1.3 remote-as 300
R2(config-router)#netw 2.2.2.0 mask 255.255.255.0
```

On R3:

```
R3(config)#router bgp 300
R3(config-router)#no au
R3(config-router)#neighbor 13.1.1.1 remote-as 100
R3(config-router)#neighbor 13.1.1.1 default-originate
R3(config-router)#neighbor 23.1.1.2 remote-as 200
R3(config-router)#neighbor 23.1.1.2 default-originate
```

On R4:

```
R4(config)#router bgp 400
R4(config-router)#no au
R4(config-router)#bgp router-id 4.4.4.4
```

To verify the configuration:

On R3:

```
R3#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       13.1.1.1                 0             0 100 i
*> 2.2.2.0/24       23.1.1.2                 0             0 200 i
```

On R1:

```
R1#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          13.1.1.3                 0             0 300 i
*> 1.1.1.0/24       0.0.0.0                  0         32768 i
*> 2.2.2.0/24       13.1.1.3                               0 300 200 i
```

On R2:

```
R2#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          23.1.1.3                 0             0 300 i
*> 1.1.1.0/24       23.1.1.3                               0 300 100 i
*> 2.2.2.0/24       0.0.0.0                  0         32768 i
```

Task 2

Configure OSPF based on the following requirements:

- Configure OSPF area 0 on R3's S2/0 interface. The router-id of this router should be set to 0.0.0.3.
- Configure OSPF area 0 on R4's S1/3 interface. The router-id of this router should be set to 0.0.0.4.
- R3 should be configured to redistribute BGP into OSPF.
- R4 should be configured to redistribute OSPF into BGP.
- R4 should have networks 1.1.1.0/24 and 2.2.2.0/24 in its BGP table with the AS numbers in which these two prefixes originated in.

On R3:

```
R3(config)#router ospf 1
R3(config-router)#router-id 0.0.0.3
R3(config-router)#netw 34.1.1.3 0.0.0.0 area 0
R3(config-router)#redistribute bgp 300 subnets
```

On R4:

```
R4(config)#router ospf 1
R4(config-router)#router-id 0.0.0.4
R4(config-router)#netw 34.1.1.4 0.0.0.0 area 0

R4(config)#router bgp 400
R4(config-router)#redistribute ospf 1 match internal external
```

To verify the configuration:

On R3:

```
R3#sh ip ospf database

            OSPF router with ID (0.0.0.3) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV router      Age         Seq#       Checksum Link
count
0.0.0.3         0.0.0.3         173         0x80000004 0x00AA9F 2
0.0.0.4         0.0.0.4         179         0x80000003 0x009AB0 2

                Type-5 AS External Link States

Link ID         ADV router      Age         Seq#       Checksum Tag
1.1.1.0         0.0.0.3         265         0x80000001 0x000247 100
2.2.2.0         0.0.0.3         265         0x80000001 0x00ECF4 200
```

When BGP is redistributed into an IGP, the BGP AS number is copied as a route tag in that given IGP. By looking at the external routes, we can see that netw 1.1.1.0 has a Tag of 100, and netw 2.2.2.0 is tagged with 200.

Let's verify the BGP table on R4:

On R4:

```
R4#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 ?
*> 2.2.2.0/24       34.1.1.3                 1         32768 ?
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

What happened to the AS-Path?

NOTE: When OSPF was redistributed into BGP, the AS-Path was lost and the origin code was set to unknown.

This AS-Path can be retrieved on R4 using the tag that was set when the BGP was redistributed into OSPF; this can be configured using the "AS-Path tag", let's configure this and verify:

On R4:

```
R4(config)#route-map tst permit 10
R4(config-route-map)#Set as-path tag

R4(config)#router bgp 400
R4(config-router)#redistribute ospf 1 match internal external route-map tst
```

To verify the configuration:

On R4:

```
R4#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 100 ?
*> 2.2.2.0/24       34.1.1.3                 1         32768 200 ?
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

The output of the above show command reveals that the route tags which were the BGP AS numbers are copied into the AS-Path. But the second problem that we are facing is the origin code, which is set to unknown.
R3 can be configured to fix this problem, in this case R3 can be configured to use the "Automatic-tag and the Table-map" as follows:

On R3:

The following configures a route-map and sets the "automatic-tag":

```
R3(config)#route-map tst permit 10
R3(config-route-map)#set automatic-tag
```

The route-map is referenced under router BGP using the "Table-map":

```
R3(config)#router bgp 300
R3(config-router)#table-map tst
```

To implement the changes:

```
R3#clear ip bgp *
```

To verify the configuration:

On R4:

```
R4#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 3489661028 i
*> 2.2.2.0/24       34.1.1.3                 1         32768 3489661128 i
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

So what happened to the AS number?
The AS numbers are correct but we have to go through a calculation to figure them, let's do this:

Take the number and convert it to Hex:
3489661028 → D0000064
Let's focus on the last 4 characters "0064" and convert it back to decimal → 100

Let's convert the AS number for netw 2.2.2.0/24:
3489661128 → D00000C8
Let's convert the last 4 characters "00C8" to decimal → 200

There MUST be an easier/better way to do this, how would you handle this kind of issue in the real world?

Let's remove the "route-map tst" that was configured on R4:

On R4:

```
R4(config)#no route-map tst

R4(config)#router bgp 400
R4(config-router)#no redis ospf 1 match inter exte 1 exte 2 route-map tst
R4(config-router)#redistr ospf 1 match inter exter

R4#clear ip bgp *
```

NOW, let's check the OSPF database and verify the tag for networks 1.1.1.0/24 and 2.2.2.0/24:

```
R4#sh ip ospf database


            OSPF router with ID (0.0.0.4) (Process ID 1)

                router Link States (Area 0)

Link ID         ADV router      Age         Seq#       Checksum Link
count
0.0.0.3         0.0.0.3         1607        0x80000004 0x009B0F 2
0.0.0.4         0.0.0.4         1606        0x80000002 0x008D1F 2
```

```
                Type-5 AS External Link States

Link ID         ADV router      Age         Seq#       Checksum Tag
1.1.1.0         0.0.0.3         1326        0x80000003 0x003A3C 3489661028
2.2.2.0         0.0.0.3         1326        0x80000003 0x0025E9 3489661128
```

Based on the output of the above show command we can see that the "Automatic tag" is the feature that changed the tag to the two highlighted in yellow and blue. Let's verify the tags in the BGP table:

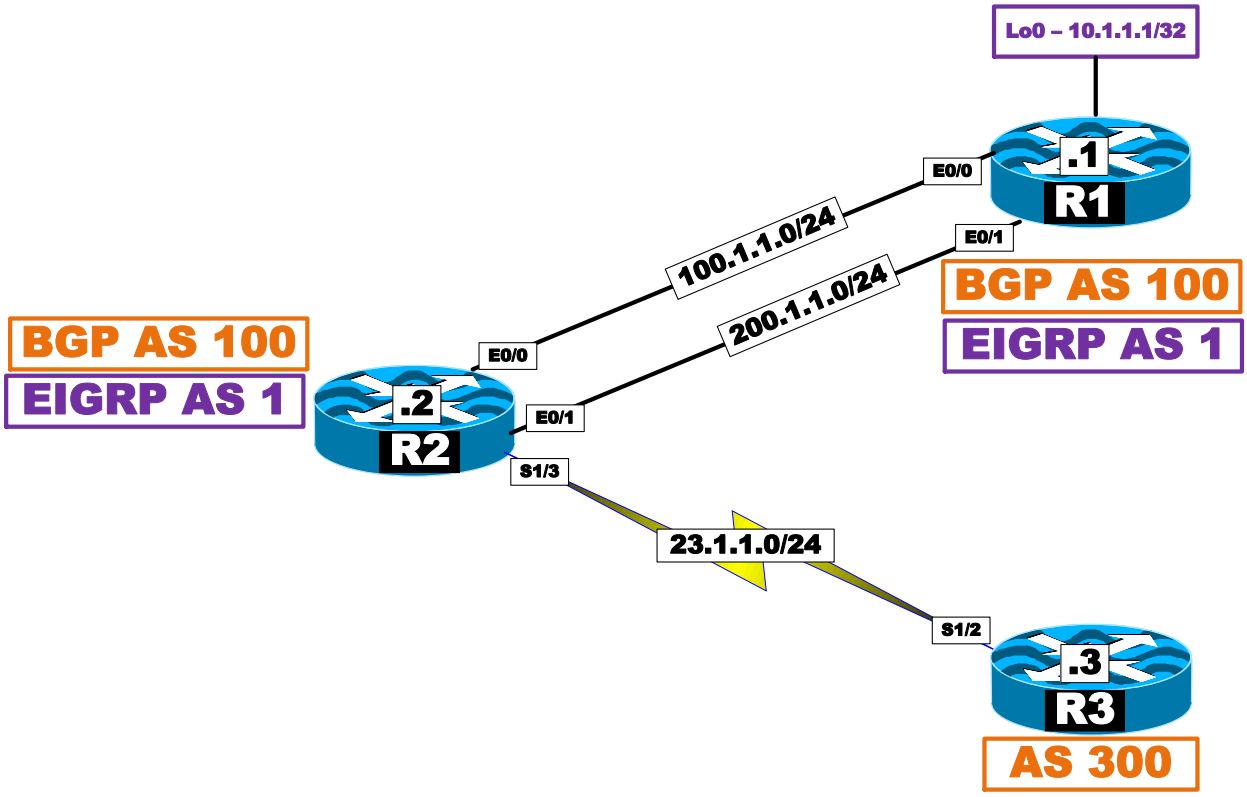
```
R4#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 100 i
*> 2.2.2.0/24       34.1.1.3                 1         32768 200 i
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

Task 3

Erase the startup configuration on all routers and switches, delete vlan.dat on all switches and reload all devices before proceeding to the next lab.

Lab 2 – BGP Suppress-inactive



Task 1

Configure the above topology.

On SW1:

```
SW1(config)#int range e0/2-3
SW1(config-if-range)#duple hal
SW1(config-if-range)#swi
SW1(config-if-range)#swi mode acc
SW1(config-if-range)#swi acc v 23
SW1(config-if-range)#spannin portf
```

```
SW1(config-if-range)#no shu
```

On R1:

```
R1(config)#int lo0
R1(config-if)#ip addr 10.1.1.1 255.255.255.255

R1(config-if)#int e0/0
R1(config-if)#ip addr 100.1.1.1 255.255.255.0
R1(config-if)#no shut

R1(config)#int e0/1
R1(config-if)#ip addr 200.1.1.1 255.255.255.0
R1(config-if)#no shu
```

On R2:

```
R2(config)#int e0/0
R2(config-if)#ip addr 100.1.1.2 255.255.255.0
R2(config-if)#no shu

R2(config)#int e0/1
R2(config-if)#ip addr 200.1.1.2 255.255.255.0
R2(config-if)#no shu

R2(config-if)#int s1/3
R2(config-if)#ip addr 23.1.1.2 255.255.255.0
R2(config-if)#no shu
```

On R3:

```
R3(config)#int s1/2
R3(config-if)#ip addr 23.1.1.3 255.255.255.0
R3(config-if)#no shut
```

Task 2

Configure R1 to establish a BGP peer session with R2 using the 200.1.1.2 IP address. R1 should advertise its lo0 to R2 using BGP.
These routers should also run EIGRP on their E0/0 interfaces. R1 should advertise its lo0 interface to R2 using EIGRP.

On R1:

```
R1(config)#router bgp 100
R1(config-router)#netw 10.1.1.1 mask 255.255.255.255
R1(config-router)#neigh 100.1.1.2 remote 100

R1(config-router)#router eigrp 1
R1(config-router)#netw 200.1.1.1 0.0.0.0
R1(config-router)#netw 10.1.1.1 0.0.0.0
```

On R2:

```
R2(config)#router bgp 100
R2(config-router)#neigh 100.1.1.1 remote 100
R2(config-router)#neigh 23.1.1.3 remote 300

R2(config)#router eigrp 1
R2(config-router)#netw 200.1.1.2 0.0.0.0
```

On R3:

```
R3(config-if)#router bgp 300
R3(config-router)#neigh 23.1.1.2 remote 100
```

To verify the configuration:

On R2:

```
R2#sh ip rou eigrp | b Gate
Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 1 subnets
D        10.1.1.1 [90/409600] via 200.1.1.1, 00:00:46, Ethernet0/1

R2#sh ip rou 10.1.1.1
Routing entry for 10.1.1.1/32
  Known via "eigrp 1", distance 90, metric 409600, type internal
  Redistributing via eigrp 1
  Last update from 200.1.1.1 on Ethernet0/1, 00:02:02 ago
  Routing Descriptor Blocks:
  * 200.1.1.1, from 200.1.1.1, 00:02:02 ago, via Ethernet0/1
      Route metric is 409600, traffic share count is 1
      Total delay is 6000 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

We can see that R2 took EIGRP over BGP because EIGRP's AD is lower. Let's check the BGP table:

```
R2#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 r>i 10.1.1.1/32      100.1.1.1                0    100      0 i
```

We can see that we have a RIB failure, let's see why:

```
R2#sh ip bgp rib-failure
  Network            Next Hop                  RIB-failure   RIB-NH Matches
10.1.1.1/32        100.1.1.1       Higher admin distance              n/a
```

Well, it is due to a higher administrative distance. Did R3 receive the route?

On R3:

```
R3#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.1/32      23.1.1.2                               0 100 i
```

Yes, R3 did receive the route, even though the route is inactive. But why?

Let's see the next hop from EIGRP and BGP's perspective:

On R2:

```
R2#sh ip rou ei | b Gate
Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 1 subnets
D        10.1.1.1 [90/409600] via 200.1.1.1, 00:06:36, Ethernet0/1


R2#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 r>i 10.1.1.1/32      100.1.1.1                0    100      0 i
```

Even though there is a mismatch in the next hop, R2 advertises 10.1.1.1/32 to R3. Because by default, the inactive routes are advertised.

To suppress the inactive routes:

```
R2(config)#router bgp 100
R2(config-router)#bgp suppress-inactive
```

To verify the configuration:

On R3:

```
R3#sh ip bgp | b Net
R3#
```

Well, this worked.
The "BGP suppress-inactive" command suppressed the rib-failed routes only if the next-hop of BGP and the IGP are different. If they are not different, meaning that they are both pointing to the same IP address then the command has no affect.

This means that if the Next hop of BGP and the RIB entry match, the "BGP suppress-inactive" command will NOT suppress the route. Let's add a static route that points to the same IP address as the BGP's next hop and test:

On R2:

```
R2(config)#ip route 10.1.1.1 255.255.255.255 100.1.1.1
```

On R2, and R3:

```
Rx#clear ip bgp *
```

To verify the configuration:

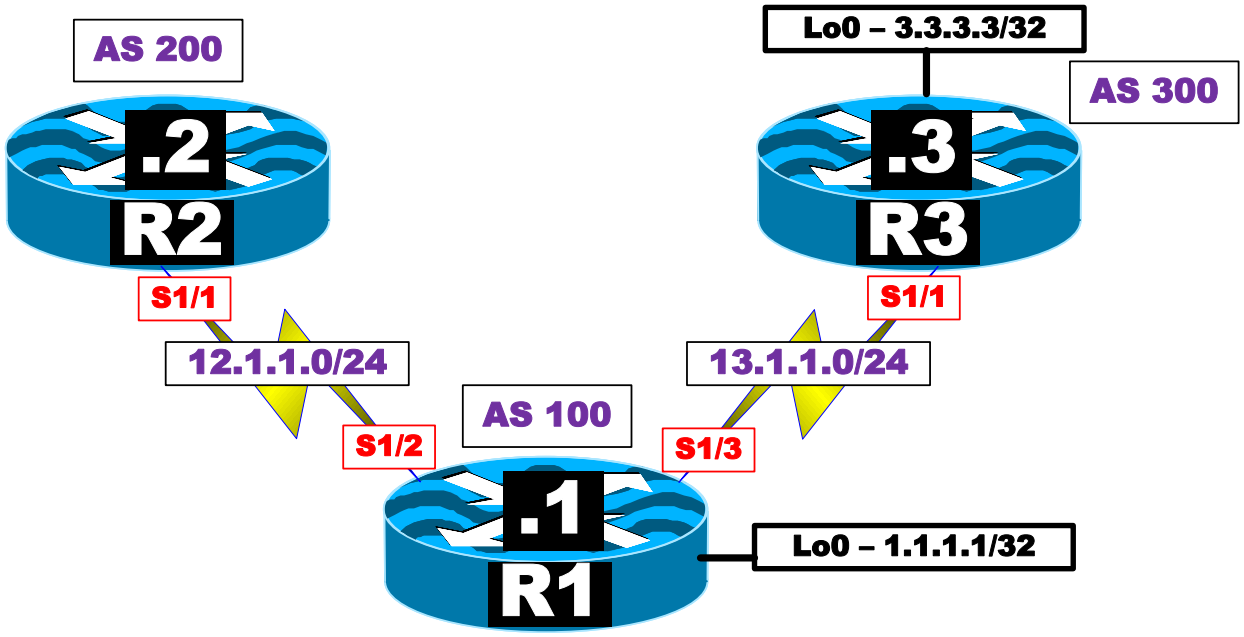
On R3:

```
R3#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.1/32      23.1.1.2                               0 100 i
```

Task 3

Erase the startup configuration on all routers and switches and reload them before proceeding to the next lab.

Lab 3 – BGP Conditional Advertisement



We configure the following two prefix-lists to identify the loopback interfaces of R1 and R3:

```
R1(config)#ip prefix-list R3 permit 3.3.3.3/32

R1(config)#ip prefix-list R1 permit 1.1.1.1/32
```

We configure the following "as-path access-list" references the AS number of the primary provider:

```
R1(config)#ip as-path access-list 1 permit ^300
```

The following route-map is configured to identify the prefix that R1 will advertise to R2 (The backup provider) if the primary provider (R3) is down:

```
R1(config)#route-map advertise
R1(config-route-map)#match ip addr prefix R1
```

The following route-map is configured to identify what needs to go down before we advertise the prefix that is referenced in the "advertise" route-map:

```
R1(config-route-map)#route-map notThere
R1(config-route-map)#match as-path 1
R1(config-route-map)#match ip addr prefix R3
```

To put them altogether:

```
R1(config-route-map)#router bgp 100
R1(config-route-map)#neighbor 12.1.1.2 advertise-map
advertise non-exist-map notThere
```

To test, we will shut down the s1/1 interface on R3 and see the results:

```
R3(config)#int s1/1
R3(config-if)#shut

R1#sh ip bgp nei 12.1.1.2 | i Condition-map
  Condition-map notThere, Advertise-map advertise, status: Advertise


R2#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.1.1.1/32       12.1.1.1                 0             0 100 i
```

Now, let's enable the S1/1 interface on R3 and see the result:

```
R3(config)#int s1/1
R3(config-if)#no shut

R1#sh ip bgp nei 12.1.1.2 | i Condition-map
  Condition-map notThere, Advertise-map advertise, status: Withdraw


R2#sh ip bgp | b Net
     Network          Next Hop          Metric LocPrf Weight Path
 *>  3.3.3.3/32       12.1.1.1                               0 100 300 i
```

We can see that prefix 1.1.1.1/32 is no longer advertised to R2.