
Let's say we need to establish an EBGP session between these two routers using their loopback interfaces, what is the hop count? And what should we set the hop count to be?
Between these two routers the hop count should be 1, it is obvious that we don't have another router between these two routers. Let's test:

| On R1: | On R2: |
|---|---|
| R1(config)#int lo0<br>R1(config-if)#ip addr 1.1.1.1 255.255.255.255<br><br>R1(config)#int s1/2<br>R1(config-if)#clock rate 64000<br>R1(config-if)#ip addr 12.1.1.1 255.255.255.0<br>R1(config-if)#No shut<br><br>R1(config)#router ospf 1<br>R1(config-router)#network 1.1.1.1 0.0.0.0 area 0<br>R1(config-router)#network 12.1.1.1 0.0.0.0 area 0<br><br>R1(config)#router bgp 100<br>R1(config-router)#neighbor 2.2.2.2 remote-as 200<br>R1(config-router)#neighbor 2.2.2.2 update-source lo0 | R2(config)#int lo0<br>R2(config-if)#ip addr 2.2.2.2 255.255.255.255<br><br>R2(config-if)#int s1/1<br>R2(config-if)#ip addr 12.1.1.2 255.255.255.0<br>R2(config-if)#no shu<br><br>R2(config)#router ospf 1<br>R2(config-router)#network 12.1.1.2 0.0.0.0 area 0<br>R2(config-router)#network 2.2.2.2 0.0.0.0 area 0<br><br>R2(config)#router bgp 200<br>R2(config-router)#neighbor 1.1.1.1 remote-as 100<br>R2(config-router)#neighbor 1.1.1.1 update-source lo0 |

We can see that these routers are not establishing an EBGP peer session. The question is why?
In order for these routers to establish an EBGP session, they will first check to see if the neighboring router is on the directly connected network, meaning that they are in our routing table as "C" or connected. The output of the following show command reveals that they are NOT directly connected.

On R2:

R2#Show ip route | b Gate
Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
O       1.1.1.1 [110/782] via 12.1.1.1, 00:00:11, Serial1/1

```
     2.0.0.0/32 is subnetted, 1 subnets
C       2.2.2.2 is directly connected, Loopback0
     12.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       12.1.1.0/24 is directly connected, Serial1/1
L       12.1.1.2/32 is directly connected, Serial1/1
```

So why are these routers not forming an adjacency?

Let's instruct these routers not to perform that check by using the "Disable-connected-check" and verify:

On R1:

```
R1(config)#router bgp 100
R1(config-router)#neighbor 2.2.2.2 disable-connected-check
```

On R2:

```
R2(config)#router bgp 200
R2(config-router)#neighbor 1.1.1.1 disable-connected-check
```

We can see that the peer session is up, so the routers do check to see if the neighboring router is on the directly connected network. But what is the TTL count? Let's check:

```
R2#Show ip bgp neighbors 1.1.1.1 | Inc TTL
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 1
```

We can see that we are going out with a TTL of one.

Let's prove this philosophy:

| On R1: | On R2: |
|---|---|
| R1(config)#int s1/2<br>R1(config-if)#ip unnumbered lo0<br>R1(config-if)#encap ppp | R2(config)#int s1/1<br>R2(config-if)#ip unnumbered lo0<br>R2(config-if)#encap ppp |

In the above configuration we are using the "ip unnumbered" and "encap ppp" so the loopback interface's IP is now assigned to our serial interface and the PPP neighbor peer route which is a host route is going to be injected in our neighbor's routing table as "C". Let's verify:

| On R1: | On R2: |
|---|---|
| R1#Show ip route \| b Gate<br>Gateway of last resort is not set | R2#Show ip route \| b Gate<br>Gateway of last resort is not set |

| | |
|---|---|
| 1.0.0.0/32 is subnetted, 1 subnets<br>C 1.1.1.1 is directly connected, Loopback0<br>2.0.0.0/32 is subnetted, 1 subnets<br>C 2.2.2.2 is directly connected, Serial1/2 | 1.0.0.0/32 is subnetted, 1 subnets<br>C 1.1.1.1 is directly connected, Serial1/1<br>2.0.0.0/32 is subnetted, 1 subnets<br>C 2.2.2.2 is directly connected, Loopback0 |

The following console message tells us that BGP is up:

%BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up

What about setting the TTL to 2 using the "ebgp-multihop 2" command; this command does the same thing as the "disable-connected-check" command the only difference here is that this command sets the TTL to 2, whereas, the "disable-connected-check" command sets the TTL to 1.

So if the routers are directly connected and they are using their loopback interfaces for peering a hop count of one is enough. So based on this philosophy, what if we have the following topology, what should we set the TTL?


Well, from R1's perspective 3.3.3.3 is two hops away, remember hops are routers so the TTL should be set to 2. Let's test and verify:

| On R1: | On R2: |
|---|---|
| R1(config)#int lo0<br>R1(config-if)#ip addr 1.1.1.1 255.255.255.255<br><br>R1(config)#int s1/2<br>R1(config-if)#clock rate 64000<br>R1(config-if)#ip addr 12.1.1.1 255.255.255.0<br>R1(config-if)#No shut<br><br>R1(config)#router ospf 1<br>R1(config-router)#network 0.0.0.0 0.0.0.0 area 0<br><br>R1(config)#router bgp 100<br>R1(config-router)#neighbor 3.3.3.3 remote-as 300<br>R1(config-router)#neighbor 3.3.3.3 update-source lo0 | R2(config)#int s1/3<br>R2(config-if)#clock rate 64000<br>R2(config-if)#ip addr 23.1.1.2 255.255.255.0<br>R2(config-if)#No shut<br><br>R2(config)#int s1/1<br>R2(config-if)#ip addr 12.1.1.2 255.255.255.0<br>R2(config-if)#No shut<br><br>R2(config)#router ospf 1<br>R2(config-router)#netw 0.0.0.0 0.0.0.0 area 0 |

On R3:

```
R3(config)#int lo0
R3(config-if)#ip addr 3.3.3.3 255.255.255.255

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

R3(config)#router ospf 1
R3(config-router)#network 0.0.0.0 0.0.0.0 area 0

R3(config)#router bgp 300
R3(config-router)#neighbor 1.1.1.1 remote-as 100
R3(config-router)#neighbor 1.1.1.1 update-source lo0
```

Let's set the TTL to two and verify our philosophy:

On R1:

```
R1(config)#router bgp 100
R1(config-router)#neighbor 3.3.3.3 ebgp-multihop 2
```

On R3:

```
R3(config)#router bgp 300
R3(config-router)#Neighbor 1.1.1.1 ebgp-multihop 2
```

The following console message reveals that the routers formed an adjacency:

%BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up

Now, let's raise this to CCIE level. What if we are asked to configure these routers (R1 and R3) such that they establish an EBGP session using "disable-connected-check", is this possible without configuring GRE, IPnIP, Xconnects, AToM,or bridging?

Well let's try:

| On R3: | On R1: |
|---|---|
| R3(config)#router bgp 300<br>R3(config-router)#No neighbor 1.1.1.1 ebgp-multihop 2<br>R3(config-router)#Neig 1.1.1.1 disable-connected-check<br><br>R3#Show run \| s router bgp 300 | R1(config)#router bgp 100<br>R1(config-router)#No neighbor 3.3.3.3 ebgp-multihop 2<br>R1(config-router)#neig 3.3.3.3 disable-connected-check<br><br>R1#Show run \| s router bgp |

| | |
|---|---|
| router bgp 300<br>bgp log-neighbor-changes<br>neighbor 1.1.1.1 remote-as 100<br>neighbor 1.1.1.1 disable-connected-check<br>neighbor 1.1.1.1 update-source Loopback0 | router bgp 100<br>bgp log-neighbor-changes<br>neighbor 3.3.3.3 remote-as 300<br>neighbor 3.3.3.3 disable-connected-check<br>neighbor 3.3.3.3 update-source Loopback0 |

We can see that BGP is not going to come up, so let's configure these routers to form an adjacency without changing any configuration in BGP:

On R1

Within the MPLS domain, the MPLS TTL is decremented at each MPLS hop. When an MPLS encapsulated IP packet exits the MPLS domain, the MPLS TTL is propagated to the IP header. The following command we are instructing the router not to do that:

R1(config)#No mpls ip propagate-ttl

The following command instructs the second hop router not to pop the label.

R1(config)#MPLS ldp explicit-null

R1(config)#int s1/2
R1(config-if)#mpls ip

| | |
|---|---|
| On R2:<br><br>R2(config)#no mpls ip propagate-ttl<br>R2(config)#mpls ldp explicit-null<br><br>R2(config)#int S1/1<br>R2(config-if)#mpls ip<br><br>R2(config-if)#int S1/3<br>R2(config-if)#mpls ip | On R3:<br><br>R3(config)#No mpls ip propagate-ttl<br><br>R3(config)#mpls ldp explicit-null<br><br>R3(config)#int s1/2<br>R3(config-if)#mpls ip |

%BGP-5-ADJCHANGE : neighbor 1.1.1.1 Up

We can see that the session came up.