BGP Policy Lab

Introduction

The purpose of this exercise is to:

- Apply the concepts of BGP policy learned in class to achieve the desired traffic patterns, particularly in an academic environment.
- Learn how to use Local Preference, BGP Communities, AS Path Prepending and related BGP operational commands.

Pre-requisites

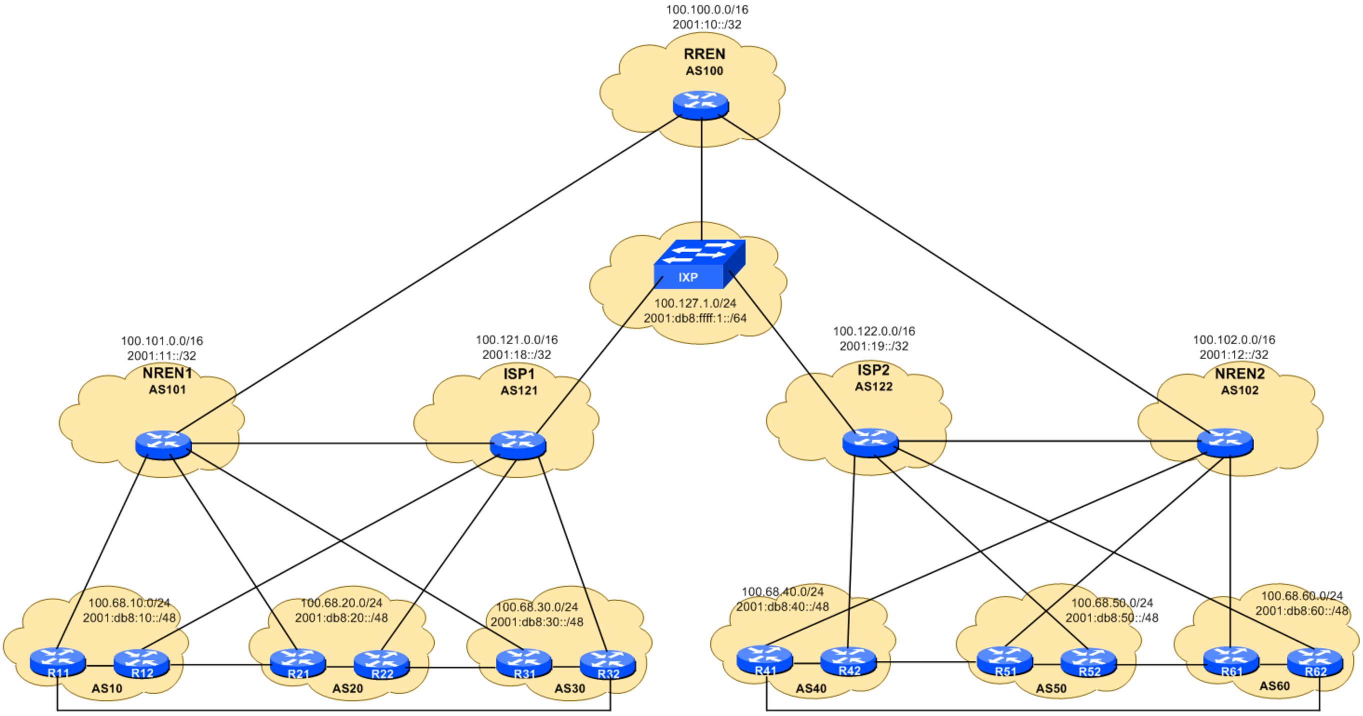
This exercise builds upon the configurations implemented in the basic BGP routing lab. You must:

- Verify that all your BGP sessions are up
- Be able to see every lab prefix in your routing table
- Be able to ping and traceroute successfully to any other router in the lab.

Remember, all the above applies to both IPv4 and IPv6.

Lab Network Topology and Address Assignments

The following diagram should serve as a visual reminder of the topology of the lab network and the address blocks assigned to each group, ISP, NREN, etc.



Routing Policy in academic networks

Research and Education Networks (RENs) are designed for high throughput and low latency. In many cases their links are also subsidized by governments and other organizations. Therefore, it is common in academic environments to want to apply routing policies that prefer these paths over the “commodity” (commercial) ones.

Local Preference

Our first goal is to configure our routers to prefer the paths via the NREN for outgoing traffic to ALL destinations.

Preferring NREN routes

Use the Local Preference attribute to prefer all routes learned via the NREN:

R11 (to NREN):

```
route-map set-lpref-nren permit 10
 description Set High Local Pref on everything
 set local-preference 150
!
router bgp 10
 address-family ipv4
  neighbor 100.101.1.1 route-map set-lpref-nren in
 address-family ipv6
```

```
  neighbor 2001:11:0:10:: route-map set-lpref-nren in
```

R12 (to ISP):

```
route-map set-lpref-isp permit 10
 description Set Low Local Pref on everything
 set local-preference 50
!
router bgp 10
 address-family ipv4
  neighbor 100.121.1.1 route-map set-lpref-isp in
 address-family ipv6
  neighbor 2001:18:0:10:: route-map set-lpref-isp in
```

What is the default local preference in Cisco IOS?

Notice that we are setting a higher preference on the NREN side, and a lower preference on the ISP side. **Explain why this would be useful.**

Check your BGP routes. The next hop in R11 should be the address of your NREN's router (except for your own prefix). In R12, the next hop should be R11's loopback address. Remember the 'next-hop-self' parameter?

```
show ip bgp
show bgp ipv6 unicast
```

All good now, right?

Wait!… What about the prefixes of ASs with whom you are peering directly? Remember the path selection algorithm? What comes first, highest local preference or shortest AS path?

Increasing local preference for direct peers

We will now modify the route map to apply a higher local preference to prefixes originated by your direct peers. So we want high preference on prefixes originated by the NREN and by the ISP. We also want high preference on prefixes transited to us by the NREN (as they will be from other NREN customers or the RREN).

Notice the AS Path access list below. How does it work?

What we have done, rather than listing all the ASNs above, we have said that anything originated by any neighbouring AS should be matched, and in the route-map we will set a local-preference even higher than that for the regular routes heard from the neighbours.

On R11:

```
ip as-path access-list 1 permit ^[0-9]+$
!
no route-map set-lpref-nren
!
```

```
route-map set-lpref-nren permit 10
 description Set High Local Pref for adjacent ASNs
 match as-path 1
 set local-preference 200
route-map set-lpref-nren permit 20
 description Set Local Pref for rest of NREN routes
 set local-preference 150
!
router bgp 10
 address-family ipv4
  neighbor 100.101.1.1 route-map set-lpref-nren in
 address-family ipv6
  neighbor 2001:11:0:10:: route-map set-lpref-nren in
!
```

On R12:

```
ip as-path access-list 1 permit ^[0-9]+$
!
no route-map set-lpref-isp
!
route-map set-lpref-isp permit 10
 description Set High Local Pref for adjacent ASNs
 match as-path 1
 set local-preference 200
route-map set-lpref-isp permit 20
 description Set Low Local Pref for rest of ISP routes
 set local-preference 50
!
router bgp 10
 address-family ipv4
  neighbor 100.121.1.1 route-map set-lpref-isp in
 address-family ipv6
  neighbor 2001:18:0:10:: route-map set-lpref-isp in
!
```

Use BGP refresh to make sure that the policies are applied:

```
clear ip bgp * in
clear ip bgp * out
clear bgp ipv6 unicast * in
clear bgp ipv6 unicast * out
```

Check your BGP routes again. What is the next hop towards your direct peers' prefixes? Can you explain what is happening now?

Preferring Local Peers over upstreams

In the previous step we saw that traffic from our immediate neighbouring groups went via our NREN provider rather than via the direct peering. For example AS10 traffic to AS20 and AS30 is going via the NREN, not over the direct peering links. We need to fix this now.

We need to create a new route-map for the bi-lateral peerings we have with the other groups.

R11 peering with R32

```
ip as-path access-list 1 permit ^[0-9]+$
!
route-map set-lpref-peer permit 10
 description Set High Local Pref for adjacent ASNs
 match as-path 1
 set local-preference 200
!
router bgp 10
 address-family ipv4
  neighbor 100.68.30.21 route-map set-lpref-peer in
 address-family ipv6
  neighbor 2001:db8:30:11::0 route-map set-lpref-peer in
!
```

The other router groups should come up with a similar configuration for R12 through R62. The route-map will be the same, just allowing prefixes from the neighbouring AS into the network, and setting the local preference high. Once this is applied, all prefixes from immediately adjacent ASes will be local preference 200.

Note that the route-map *set-lpref-peer* does not have a *line 20*. This ensures that any prefix a peer sends to the local AS must only be originated by the adjacent AS, and not be transited from anywhere else.

Explain why this might be a good idea.

Use BGP route refresh to make sure that the policies are applied:

```
clear ip bgp * in
clear ip bgp * out
clear bgp ipv6 unicast * in
clear bgp ipv6 unicast * out
```

Check your BGP routes again. What is the next hop towards your direct peers' prefixes? (**Hint: the path should be direct now!**)

STOP - Checkpoint One

All groups must finish this part before continuing. **Please do NOT continue until the instructor says so**.

As part of the review, before we move on to the next BGP policy section, the instructors will display the BGP table from one of the groups routers on the overhead screen. Take a look at your BGP table now be ready to explain what the best paths are from your router, your AS, to the ISPs, to the NRENs and to the RREN. Do the paths seem reasonable to you?

What about in your day to day situation? If you are an NREN operator, would you want the paths you see now? Or if you are a campus network administrator, what would your view be? Discuss.

AS-Path Prepending

At this point we have influenced outbound traffic only. Now we want to influence the traffic COMING IN to our AS. We want traffic to come to us via the R&E networks as much as possible.

In the case of this lab, every other group is already preferring the NREN link for their outgoing traffic. For groups connected to your same NREN, the traffic towards you will NOT go via the commodity (commercial) Internet. However, this is not the case for groups connected to other NRENs.

To see this, check your paths towards groups NOT connected to your NREN. For example, from AS10:

```
R11# show ip bgp 100.68.40.0
R11# traceroute 100.68.40.1
R11# show bgp ipv6 unicast 2001:db8:40::/48
R11# traceroute 2001:db8:40::1
```

Notice that the traffic leaves via the R&E networks, but then enters AS40 through their commercial ISP.

The same happens with traffic coming back to you from other NRENs. How can you influence their path selection so that traffic towards you enters via your NREN?

We will now use a technique called AS path prepending, which consists of adding extra “fake” hops to a path using our ASN multiple times.

AS Path Prepending to ISP

Prepend your AS number twice in the path announced to your ISP:

On R12:

```
ip prefix-list AS10-prefix permit 100.68.10.0/24
!
route-map set-prepend permit 10
 description 2x prepend on our prefix block
 match ip address prefix-list AS10-prefix
 set as-path prepend 10 10
route-map set-prepend permit 20
!
ipv6 prefix-list AS10-v6-prefix permit 2001:db8:10::/48
!
route-map set-v6-prepend permit 10
```

```
 description 2x prepend on our prefix block
 match ipv6 address prefix-list AS10-v6-prefix
 set as-path prepend 10 10
route-map set-v6-prepend permit 20
!
router bgp 10
 address-family ipv4
  neighbor 100.121.1.1 route-map set-prepend out
 address-family ipv6
  neighbor 2001:18:0:10:: route-map set-v6-prepend out
!
```

Use BGP refresh to re-announce your prefix to the ISP:

```
R12# clear ip bgp 100.121.1.1 out
R12# clear bgp ipv6 unicast 2001:18:0:10:: out
```

Ask remote groups (connected to the other NRENs), to verify that their paths towards you do not traverse the commercial ISPs.

STOP - Checkpoint Two

All groups must finish this part before continuing. **Please do NOT continue until the instructor says so**.

BGP Communities

Now let's reflect on our initial outbound policy. Since our NREN carries commodity Internet prefixes in addition to R&E prefixes, we decided to use the Local Preference attribute to send **everything** via the NREN.

In reality this may not be optimal, because the NREN may not always have the best paths towards the rest of the Internet and also because we're not taking advantage of our dual connections to load-balance our outbound traffic.

What we really need is a way to tell **which prefixes are originated from the R&E community**, so that we prefer the NREN link when sending to **those** prefixes only, and let the rest be decided by the regular BGP selection process. This is where BGP Communities are useful.

Remove the configurations from the Local Preference section. Notice the correct order in which this should be done (hint: do not remove something if it's still referenced by something else):

R11:

```
router bgp 10
 address-family ipv4
  no neighbor 100.101.1.1 route-map set-lpref-nren in
 address-family ipv6
  no neighbor 2001:11:0:10:: route-map set-lpref-nren in
!
no route-map set-lpref-nren
```

Remember to do the equivalent on the other router in your group.

RENs use BGP communities (basically tags) to mark groups of routes together as a unit, which makes it easier for their neighbours to apply policies to those groups of routes.

In this particular case, the NRENs carry research and education (R&E) routes, as well as commercial Internet routes. The R&E routes are marked with a special community (99) as they are received from each customer. Also, the NREN passes those communities on to other customers and to the RREN.

Notice that the NRENs and the RREN also use the communities to set a higher local preference value, in order to prefer the R&E paths. This is because they also can learn those prefixes via the ISPs with whom they peer.

NREN1:

```
ip bgp-community new-format
!
route-map set-RE-comm permit 10
 description Tag what we get from RE customer
 set community 101:99
!
ip community-list 1 permit 100:99
!
route-map set-RE-lpref permit 10
 description Look for RE routes from RREN
 match community 1
 set local-preference 150
route-map set-RE-lpref permit 20
 description Other routes are untouched
!
router bgp 101
 address-family ipv4
  neighbor 100.101.1.2 send-community
  neighbor 100.101.1.2 route-map set-RE-comm in
  neighbor 100.101.1.6 send-community
  neighbor 100.101.1.6 route-map set-RE-comm in
  neighbor 100.101.1.10 send-community
  neighbor 100.101.1.10 route-map set-RE-comm in
  neighbor 100.100.1.1 send-community
  neighbor 100.100.1.1 route-map set-RE-lpref in
 address-family ipv6
  neighbor 2001:11:0:10::1 send-community
  neighbor 2001:11:0:10::1 route-map set-RE-comm in
  neighbor 2001:11:0:11::1 send-community
  neighbor 2001:11:0:11::1 route-map set-RE-comm in
  neighbor 2001:11:0:12::1 send-community
  neighbor 2001:11:0:12::1 route-map set-RE-comm in
  neighbor 2001:10:0:10:: send-community
  neighbor 2001:10:0:10:: route-map set-RE-lpref in
!
```

NREN2:

```
ip bgp-community new-format
!
route-map set-RE-comm permit 10
 description Tag what we get from RE customer
 set community 102:99
!
ip community-list 1 permit 100:99
!
route-map set-RE-lpref permit 10
 description Look for RE routes from RREN
 match community 1
 set local-preference 150
route-map set-RE-lpref permit 20
 description Other routes are untouched
!
router bgp 102
 address-family ipv4
  neighbor 100.102.1.2 send-community
  neighbor 100.102.1.2 route-map set-RE-comm in
  neighbor 100.102.1.6 send-community
  neighbor 100.102.1.6 route-map set-RE-comm in
  neighbor 100.102.1.10 send-community
  neighbor 100.102.1.10 route-map set-RE-comm in
  neighbor 100.100.1.5 send-community
  neighbor 100.100.1.5 route-map set-RE-lpref in
 address-family ipv6
  neighbor 2001:12:0:10::1 send-community
  neighbor 2001:12:0:10::1 route-map set-RE-comm in
  neighbor 2001:12:0:11::1 send-community
  neighbor 2001:12:0:11::1 route-map set-RE-comm in
  neighbor 2001:12:0:12::1 send-community
  neighbor 2001:12:0:12::1 route-map set-RE-comm in
  neighbor 2001:10:0:11:: send-community
  neighbor 2001:10:0:11:: route-map set-RE-lpref in
!
```

The regional REN (RREN) connects multiple NRENs, so they replace communities in the R&E routes learned from NRENs with their own community:

RREN:

```
ip bgp-community new-format
!
ip community-list 1 permit 101:99
ip community-list 1 permit 102:99
!
route-map set-RE-comm permit 10
 match community 1
 set community 100:99
 set local-preference 150
route-map set-RE-comm permit 20
!
router bgp 100
 address-family ipv4
  neighbor 100.100.1.2 send-community
  neighbor 100.100.1.2 route-map set-RE-comm in
  neighbor 100.100.1.6 send-community
  neighbor 100.100.1.6 route-map set-RE-comm in
 address-family ipv6
  neighbor 2001:10:0:10::1 send-community
  neighbor 2001:10:0:10::1 route-map set-RE-comm in
  neighbor 2001:10:0:11::1 send-community
  neighbor 2001:10:0:11::1 route-map set-RE-comm in
```

Explain the purpose of replacing the NREN communities at the RREN, before they are passed on to other NRENs.

The only thing we still need to do is tag the routes originated by the NRENs and RREN and being R&E routes too, otherwise there is every likelihood we will see best paths via ISPs.

NREN1 example:

```
route-map RE-comm-tag permit 10
 set community 101:99
!
router bgp 101
 address-family ipv4
  network 100.101.0.0 mask 255.255.0.0 route-map RE-comm-tag
 address-family ipv6
  network 2001:11::/32 route-map RE-comm-tag
!
```

Similar configuration needs to be set up for NREN2 and the RREN. Use the same route-map, just set the community to be appropriate for that REN.

ISPs will originate additional prefixes to represent the rest of the commodity Internet. The configurations below will make the BGP table look as though ISP1 has AS65001, AS65002, AS65003 and AS65004 as customers, and ISP2 has AS65005, AS65006, AS65007 and AS65008 as customers.

ISP1:

```
ip prefix-list v4-commodity-1 permit 172.16.0.0/16
ip prefix-list v4-commodity-2 permit 172.17.0.0/16
ip prefix-list v4-commodity-3 permit 172.18.0.0/16
ip prefix-list v4-commodity-4 permit 172.19.0.0/16
```

```
!
ipv6 prefix-list v6-commodity-1 permit 2001:db8::/32
ipv6 prefix-list v6-commodity-2 permit 2001:db9::/32
ipv6 prefix-list v6-commodity-3 permit 2001:dba::/32
ipv6 prefix-list v6-commodity-4 permit 2001:dbb::/32
!
route-map set-prepend-commodity permit 10
 match ip address prefix-list v4-commodity-1
 set as-path prepend 65001
route-map set-prepend-commodity permit 20
 match ip address prefix-list v4-commodity-2
 set as-path prepend 65002
route-map set-prepend-commodity permit 30
 match ip address prefix-list v4-commodity-3
 set as-path prepend 65003
route-map set-prepend-commodity permit 40
 match ip address prefix-list v4-commodity-4
 set as-path prepend 65004
route-map set-prepend-commodity permit 50
!
route-map set-prepend-v6commodity permit 10
 match ipv6 address prefix-list v6-commodity-1
 set as-path prepend 65001
route-map set-prepend-v6commodity permit 20
 match ipv6 address prefix-list v6-commodity-2
 set as-path prepend 65002
route-map set-prepend-v6commodity permit 30
 match ipv6 address prefix-list v6-commodity-3
 set as-path prepend 65003
route-map set-prepend-v6commodity permit 40
 match ipv6 address prefix-list v6-commodity-4
 set as-path prepend 65004
route-map set-prepend-v6commodity permit 50
!
router bgp 121
 address-family ipv4
  network 172.16.0.0 mask 255.255.0.0
  network 172.17.0.0 mask 255.255.0.0
  network 172.18.0.0 mask 255.255.0.0
  network 172.19.0.0 mask 255.255.0.0
  neighbor 100.121.1.2 route-map set-prepend-commodity out
  neighbor 100.121.1.6 route-map set-prepend-commodity out
  neighbor 100.121.1.10 route-map set-prepend-commodity out
  neighbor 100.127.1.2 route-map set-prepend-commodity out
  neighbor 100.127.1.3 route-map set-prepend-commodity out
  neighbor 100.101.2.1 route-map set-prepend-commodity out
 !
 address-family ipv6
  network 2001:db8::/32
  network 2001:db9::/32
  network 2001:dba::/32
  network 2001:dbb::/32
  neighbor 2001:18:0:10::1 route-map set-prepend-v6commodity out
  neighbor 2001:18:0:11::1 route-map set-prepend-v6commodity out
  neighbor 2001:18:0:12::1 route-map set-prepend-v6commodity out
  neighbor 2001:db8:ffff:1::2 route-map set-prepend-v6commodity out
  neighbor 2001:db8:ffff:1::3 route-map set-prepend-v6commodity out
  neighbor 2001:11:0:20:: route-map set-prepend-v6commodity out
!
ip route 172.16.0.0 255.255.0.0 null0
ip route 172.17.0.0 255.255.0.0 null0
ip route 172.18.0.0 255.255.0.0 null0
ip route 172.19.0.0 255.255.0.0 null0
!
ipv6 route 2001:db8::/32 null0
ipv6 route 2001:db9::/32 null0
ipv6 route 2001:dba::/32 null0
ipv6 route 2001:dbb::/32 null0
```

ISP2:

```
ip prefix-list v4-commodity-1 permit 172.20.0.0/16
ip prefix-list v4-commodity-2 permit 172.21.0.0/16
ip prefix-list v4-commodity-3 permit 172.22.0.0/16
ip prefix-list v4-commodity-4 permit 172.23.0.0/16
!
ipv6 prefix-list v6-commodity-1 permit 2001:dbc::/32
ipv6 prefix-list v6-commodity-2 permit 2001:dbd::/32
ipv6 prefix-list v6-commodity-3 permit 2001:dbe::/32
ipv6 prefix-list v6-commodity-4 permit 2001:dbf::/32
!
route-map set-prepend-commodity permit 10
 match ip address prefix-list v4-commodity-1
 set as-path prepend 65005
route-map set-prepend-commodity permit 20
 match ip address prefix-list v4-commodity-2
 set as-path prepend 65006
route-map set-prepend-commodity permit 30
 match ip address prefix-list v4-commodity-3
 set as-path prepend 65007
route-map set-prepend-commodity permit 40
 match ip address prefix-list v4-commodity-4
 set as-path prepend 65008
route-map set-prepend-commodity permit 50
!
route-map set-prepend-v6commodity permit 10
 match ipv6 address prefix-list v6-commodity-1
 set as-path prepend 65005
route-map set-prepend-v6commodity permit 20
 match ipv6 address prefix-list v6-commodity-2
 set as-path prepend 65006
```

```
route-map set-prepend-v6commodity permit 30
 match ipv6 address prefix-list v6-commodity-3
 set as-path prepend 65007
route-map set-prepend-v6commodity permit 40
 match ipv6 address prefix-list v6-commodity-4
 set as-path prepend 65008
route-map set-prepend-v6commodity permit 50
!
router bgp 122
 address-family ipv4
  network 172.20.0.0 mask 255.255.0.0
  network 172.21.0.0 mask 255.255.0.0
  network 172.22.0.0 mask 255.255.0.0
  network 172.23.0.0 mask 255.255.0.0
  neighbor 100.122.1.2 route-map set-prepend-commodity out
  neighbor 100.122.1.6 route-map set-prepend-commodity out
  neighbor 100.122.1.10 route-map set-prepend-commodity out
  neighbor 100.127.1.1 route-map set-prepend-commodity out
  neighbor 100.127.1.3 route-map set-prepend-commodity out
  neighbor 100.102.2.1 route-map set-prepend-commodity out
 address-family ipv6
  network 2001:dbc::/32
  network 2001:dbd::/32
  network 2001:dbe::/32
  network 2001:dbf::/32
  neighbor 2001:19:0:10::1 route-map set-prepend-v6commodity out
  neighbor 2001:19:0:11::1 route-map set-prepend-v6commodity out
  neighbor 2001:19:0:12::1 route-map set-prepend-v6commodity out
  neighbor 2001:db8:ffff:1::1 route-map set-prepend-v6commodity out
  neighbor 2001:db8:ffff:1::3 route-map set-prepend-v6commodity out
  neighbor 2001:12:0:20:: route-map set-prepend-v6commodity out
!
ip route 172.20.0.0 255.255.0.0 null0
ip route 172.21.0.0 255.255.0.0 null0
ip route 172.22.0.0 255.255.0.0 null0
ip route 172.23.0.0 255.255.0.0 null0
!
ipv6 route 2001:dbc::/32 null0
ipv6 route 2001:dbd::/32 null0
ipv6 route 2001:dbe::/32 null0
ipv6 route 2001:dbf::/32 null0
```

We are now going to set local preference on the prefixes we learn from our various transits and peers, according to the following table:

Neighbour	Route Type	Local Preference
Bi-lateral	All	200
NREN	R&E	150
	Commodity	70
Commercial	All	80

Set local preference ONLY on the R&E routes (marked with the R&E community) learned from the NREN. Notice that your NREN is also passing you the communities set by the regional REN, so you need to match either one.

Also notice that we (should) still have the route-map which sets the local preference to 200 on the prefixes originated by our bi-lateral peers.

On R11:

```
ip bgp-community new-format
!
ip community-list 1 permit 100:99
ip community-list 1 permit 101:99
!
route-map set-lpref-nren permit 10
 description Look for RE routes
 match community 1
 set local-preference 150
route-map set-lpref-nren permit 20
 description The rest are Commodity routes
 set local-preference 70
!
router bgp 10
 address-family ipv4
  neighbor 100.101.1.1 route-map set-lpref-nren in
 address-family ipv6
  neighbor 2001:11:0:10:: route-map set-lpref-nren in
!
```

Refresh to/from your neighbors:

```
R11# clear ip bgp * in
R11# clear ip bgp * out
```

```
R11# clear bgp ipv6 unicast * in
R11# clear bgp ipv6 unicast * out
```

Verify that communities are being set and transmitted:

```
R11# show ip bgp 100.68.20.0
R11# show ip bgp 100.68.40.0
```

On R12:

```
ip as-path access-list 1 permit ^[0-9]+$
!
route-map set-lpref-isp permit 10
 description Look for ISP originated routes
 match as-path 1
!
```

```
route-map set-lpref-isp permit 20
 description All ISP routes
 set local-preference 80
!
router bgp 10
 address-family ipv4
  neighbor 100.121.1.1 route-map set-lpref-isp in
 address-family ipv6
  neighbor 2001:18:0:10:: route-map set-lpref-isp in
!
```

Refresh to/from your neighbors:

```
R12# clear ip bgp * in
R12# clear ip bgp * out
```

```
R12# clear bgp ipv6 unicast * in
R12# clear bgp ipv6 unicast * out
```

Check your BGP routes again.

```
show ip bgp
show ip route
show bgp ipv6 unicast
show ipv6 route
```

The result should be that you now prefer the NREN path for any prefix originated by an R&E member. For all other prefixes, including the ones from the commercial Internet, your routers will choose based on BGP defaults.

STOP - Checkpoint Three

All groups must finish this part before continuing. Please do **NOT** continue until the instructor says so.

Look closely at the BGP table now. Have we achieved what we set out to do?

Multihoming with Partial Routes and Defaults

Another way to load-balance outbound traffic in our multihoming setup is to play with partial routing tables and default routes. The idea is that our routers will prefer the more specific R&E routes coming from the NREN, and the rest of the outgoing traffic will use the ISP. Only if the ISP fails, our non-R&E traffic will leave through the NREN. Similarly, if the NREN link fails, the ISP will route all our outbound traffic.

This has the advantage of reducing our routing table size, and therefore convergence time. The disadvantage is that we may not always follow the best paths, but it might be a good compromise.

We are going to ask the NREN to only send us R&E routes, plus the default route:

NREN1:

```
ip community-list 1 permit 100:99
ip community-list 1 permit 101:99
!
route-map send-RE-only permit 10
 match community 1
!
router bgp 101
 address-family ipv4
  neighbor 100.101.1.2 route-map send-RE-only out
  neighbor 100.101.1.2 default-originate
  neighbor 100.101.1.6 route-map send-RE-only out
  neighbor 100.101.1.6 default-originate
  neighbor 100.101.1.10 route-map send-RE-only out
  neighbor 100.101.1.10 default-originate
 address-family ipv6
  neighbor 2001:11:0:10::1 route-map send-RE-only out
  neighbor 2001:11:0:10::1 default-originate
  neighbor 2001:11:0:11::1 route-map send-RE-only out
  neighbor 2001:11:0:11::1 default-originate
  neighbor 2001:11:0:12::1 route-map send-RE-only out
  neighbor 2001:11:0:12::1 default-originate
!
```

NREN2:

```
ip community-list 1 permit 100:99
ip community-list 1 permit 102:99
!
route-map send-RE-only permit 10
 match community 1
!
router bgp 102
 address-family ipv4
  neighbor 100.102.1.2 route-map send-RE-only out
  neighbor 100.102.1.2 default-originate
  neighbor 100.102.1.6 route-map send-RE-only out
  neighbor 100.102.1.6 default-originate
  neighbor 100.102.1.10 route-map send-RE-only out
  neighbor 100.102.1.10 default-originate
 address-family ipv6
  neighbor 2001:12:0:10::1 route-map send-RE-only out
  neighbor 2001:12:0:10::1 default-originate
  neighbor 2001:12:0:11::1 route-map send-RE-only out
  neighbor 2001:12:0:11::1 default-originate
  neighbor 2001:12:0:12::1 route-map send-RE-only out
  neighbor 2001:12:0:12::1 default-originate
```

```
!
```

Similarly, we will ask the ISP to only send us a default route:

ISP1:

```
ip prefix-list default permit 0.0.0.0/0
ipv6 prefix-list ipv6-default permit ::/0
!
router bgp 121
 address-family ipv4
  neighbor 100.121.1.2 default-originate
  neighbor 100.121.1.2 prefix-list default out
  neighbor 100.121.1.6 default-originate
  neighbor 100.121.1.6 prefix-list default out
  neighbor 100.121.1.10 default-originate
  neighbor 100.121.1.10 prefix-list default out
 address-family ipv6
  neighbor 2001:18:0:10::1 default-originate
  neighbor 2001:18:0:10::1 prefix-list ipv6-default out
  neighbor 2001:18:0:11::1 default-originate
  neighbor 2001:18:0:11::1 prefix-list ipv6-default out
  neighbor 2001:18:0:12::1 default-originate
  neighbor 2001:18:0:12::1 prefix-list ipv6-default out
!
```

ISP2:

```
ip prefix-list default permit 0.0.0.0/0
ipv6 prefix-list ipv6-default permit ::/0
!
router bgp 122
 address-family ipv4
  neighbor 100.122.1.2 default-originate
  neighbor 100.122.1.2 prefix-list default out
  neighbor 100.122.1.6 default-originate
  neighbor 100.122.1.6 prefix-list default out
  neighbor 100.122.1.10 default-originate
  neighbor 100.122.1.10 prefix-list default out
 address-family ipv6
  neighbor 2001:19:0:10::1 default-originate
  neighbor 2001:19:0:10::1 prefix-list ipv6-default out
  neighbor 2001:19:0:11::1 default-originate
  neighbor 2001:19:0:11::1 prefix-list ipv6-default out
  neighbor 2001:19:0:12::1 default-originate
  neighbor 2001:19:0:12::1 prefix-list ipv6-default out
!
```

Check what you are now receiving from your NREN and your ISP:

```
R11# show ip bgp neighbors 100.101.1.1 routes
```

```
R11# show bgp ipv6 uni neighbors 2001:11:0:10:: routes
R11# show ip route 0.0.0.0 0.0.0.0
R11# show ipv6 route ::/0
```

```
R12# show ip bgp neighbors 100.121.1.1 routes
R12# show bgp ipv6 uni neighbors 2001:18:0:10:: routes
R12# show ip route 0.0.0.0 0.0.0.0
R12# show ipv6 route ::/0
```

At this point you should see that each of your routers has a default route pointing to its upstream peer.

Check your default route on both routers:

```
show ip bgp 0.0.0.0 0.0.0.0
show ip route 0.0.0.0 0.0.0.0
```

```
show bgp ipv6 uni ::/0
show ipv6 route ::/0
```

Also, check your BGP routing table. Has it shrunk?

```
show ip bgp
show bgp ipv6 unicast
```

Confirm that you now see a default route from your ISP, with local-preference 100. And you should also see a default-route from your NREN, with local-preference 70 (based on the communities set in the previous exercise).

Summary

What have we achieved here?

We have connected our end-site to a local peer, an NREN and an ISP. The best path for for our local (bi-lateral) peer is over our peering link. The best path for all REN routes is via the NREN. The best path for all other routes is via the ISP.

- Should the link to the ISP fail, we will get backup via the NREN to access the commodity networks.
- Should the link to the NREN fail, we will get backup via the ISP to access R&E networks.

How did we achieve this?

- We tagged all routes from our bi-lateral peer with *local-preference of 200*.
- We looked for REN routes from our NREN tagged with the REN community and set *local-preference of 150*.
- We heard the *default route* from our NREN, and tagged it with low *local-preference of 70*.
- We heard the *default route* from our ISP, and left it with default *local-preference of 100*.

Discuss with the lab instructors about testing the failure modes of your group's network connectivity.

Appendix A - BGP Table: First Policy Scenario

Included for completeness and to aid discussion, here is the BGP table as seen on R62 at the first checkpoint in this lab exercise.

```
R62#sh ip bgp
BGP table version is 24, local router ID is 100.68.60.2
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
 *>i 100.68.10.0/24   100.68.60.1              0    150      0 102 100 101
10 i
 *                    100.122.1.9                    50      0 122 121 10 i
 *>i 100.68.20.0/24   100.68.60.1              0    150      0 102 100 101
20 i
 *                    100.122.1.9                    50      0 122 121 20 i
 *>i 100.68.30.0/24   100.68.60.1              0    150      0 102 100 101
30 i
 *                    100.122.1.9                    50      0 122 121 30 i
 *   100.68.40.0/24   100.122.1.9                    50      0 122 40 i
 *>                   100.68.60.22             0    200      0 40 i
 *   100.68.50.0/24   100.122.1.9                    50      0 122 50 i
 *>i                  100.68.60.1              0    200      0 50 i
 * i 100.68.60.0/24   100.68.60.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 100.100.0.0/16   100.68.60.1              0    150      0 102 100 i
 *                    100.122.1.9                    50      0 122 100 i
 *>i 100.101.0.0/16   100.68.60.1              0    150      0 102 100 101 i
 *                    100.122.1.9                    50      0 122 100 101 i
 *   100.102.0.0/16   100.122.1.9                    50      0 122 102 i
 *>i                  100.68.60.1              0    200      0 102 i
 *>i 100.121.0.0/16   100.68.60.1              0    150      0 102 100 121 i
 *                    100.122.1.9                    50      0 122 121 i
 *>  100.122.0.0/16   100.122.1.9              0    200      0 122 i
```

Appendix B - BGP Table @ ISP1: Second Policy Scenario

Included for completeness and to aid discussion, here is the BGP table as seen on ISP1 at the second checkpoint in this lab exercise.

What do you think about the path from ISP1 to AS40, AS50 and AS60? Why is the best path via ISP2, rather than via the peering with the RREN?

```
ISP1#sh ip bgp
BGP table version is 18, local router ID is 100.121.0.1
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
 *   100.68.10.0/24   100.127.1.3                            0 100 101 10 i
 *>                   100.101.2.1                            0 101 10 i
 *                    100.121.1.2              0             0 10 10 10 i
 *   100.68.20.0/24   100.127.1.3                            0 100 101 20 i
 *>                   100.101.2.1                            0 101 20 i
 *                    100.121.1.6              0             0 20 20 20 i
 *   100.68.30.0/24   100.127.1.3                            0 100 101 30 i
 *>                   100.101.2.1                            0 101 30 i
 *                    100.121.1.10             0             0 30 30 30 i
 *   100.68.40.0/24   100.101.2.1                            0 101 100 102
40 i
 *>                   100.127.1.2                            0 122 102 40 i
 *                    100.127.1.3                            0 100 102 40 i
 *   100.68.50.0/24   100.101.2.1                            0 101 100 102
50 i
 *                    100.127.1.3                            0 100 102 50 i
 *>                   100.127.1.2                            0 122 102 50 i
 *   100.68.60.0/24   100.127.1.3                            0 100 122 60 i
 *>                   100.127.1.2                            0 122 60 i
 *   100.100.0.0/16   100.101.2.1                            0 101 100 i
 *                    100.127.1.2                            0 122 100 i
 *>                   100.127.1.3              0             0 100 i
 *   100.101.0.0/16   100.127.1.2                            0 122 100 101 i
 *                    100.127.1.3                            0 100 101 i
 *>                   100.101.2.1              0             0 101 i
 *   100.102.0.0/16   100.101.2.1                            0 101 100 102 i
 *                    100.127.1.2                            0 122 102 i
 *>                   100.127.1.3                            0 100 102 i
 *>  100.121.0.0/16   0.0.0.0                  0         32768 i
 *   100.122.0.0/16   100.127.1.3                            0 100 122 i
 *>                   100.127.1.2              0             0 122 i
```

Appendix C - BGP Table: Third Policy Scenario

Included for completeness and to aid discussion, here is the BGP table as seen on R32 at the third checkpoint in this lab exercise.

```
R32#sh ip bgp
BGP table version is 33, local router ID is 100.68.30.2
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
 *   100.68.10.0/24   100.121.1.9                     80      0 121 101 10 i
 *>                   100.68.30.22             0    200      0 10 i
 *   100.68.20.0/24   100.121.1.9                     80      0 121 101 20 i
 *>i                  100.68.30.1              0    200      0 20 i
 * i 100.68.30.0/24   100.68.30.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *   100.68.40.0/24   100.121.1.9                     80      0 121 100 102
40 i
 *>i                  100.68.30.1              0    150      0 101 100 102
40 i
 *   100.68.50.0/24   100.121.1.9                     80      0 121 100 102
50 i
 *>i                  100.68.30.1              0    150      0 101 100 102
50 i
 *   100.68.60.0/24   100.121.1.9                     80      0 121 100 102
60 i
 *>i                  100.68.30.1              0    150      0 101 100 102
60 i
 *   100.100.0.0/16   100.121.1.9                     80      0 121 100 i
 *>i                  100.68.30.1              0    150      0 101 100 i
 *   100.101.0.0/16   100.121.1.9                     80      0 121 101 i
 *>i                  100.68.30.1              0    150      0 101 i
 *   100.102.0.0/16   100.121.1.9                     80      0 121 100 102 i
 *>i                  100.68.30.1              0    150      0 101 100 102 i
 *>  100.121.0.0/16   100.121.1.9              0             0 121 i
 *>  100.122.0.0/16   100.121.1.9                     80      0 121 122 i
 *>  172.16.0.0       100.121.1.9              0     80      0 121 65001 i
 *>  172.17.0.0       100.121.1.9              0     80      0 121 65002 i
 *>  172.18.0.0       100.121.1.9              0     80      0 121 65003 i
 *>  172.19.0.0       100.121.1.9              0     80      0 121 65004 i
 *>  172.20.0.0       100.121.1.9                     80      0 121 122 65005
i
 *>  172.21.0.0       100.121.1.9                     80      0 121 122 65006
i
 *>  172.22.0.0       100.121.1.9                     80      0 121 122 65007
i
 *>  172.23.0.0       100.121.1.9                     80      0 121 122 65008
i
```

Appendix D - BGP Table: Fourth Policy Scenario

Included for completeness and to aid discussion, here is the BGP table as seen on R21 at the end of this lab exercise.

```
R21#sh ip bgp
BGP table version is 96, local router ID is 100.68.20.1
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
 r>i 0.0.0.0          100.68.20.2              0    100      0 121 i
 r                    100.101.1.5                    70      0 101 i
 *   100.68.10.0/24   100.101.1.5                   150      0 101 10 i
 *>                   100.68.10.21             0    200      0 10 i
 * i 100.68.20.0/24   100.68.20.2              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 100.68.30.0/24   100.68.20.2              0    200      0 30 i
 *                    100.101.1.5                   150      0 101 30 i
 *>  100.68.40.0/24   100.101.1.5                   150      0 101 100 102
40 i
 *>  100.68.50.0/24   100.101.1.5                   150      0 101 100 102
50 i
 *>  100.68.60.0/24   100.101.1.5                   150      0 101 100 102
60 i
 *>  100.100.0.0/16   100.101.1.5                   150      0 101 100 i
 *>  100.101.0.0/16   100.101.1.5              0    150      0 101 i
 *>  100.102.0.0/16   100.101.1.5                   150      0 101 100 102 i
```