Advanced Routing Workshop

BGP Policy Lab

Contents

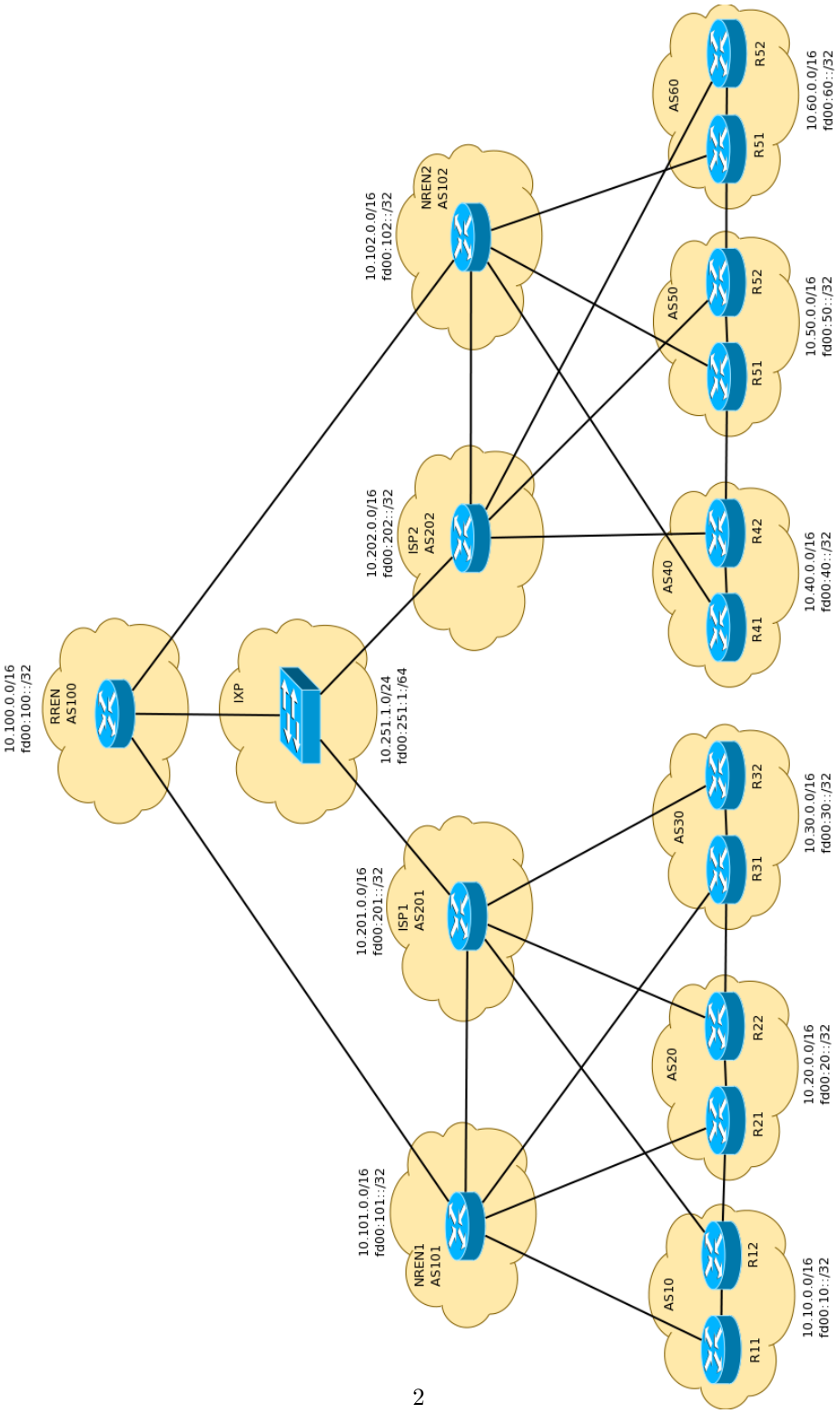

Figure 1: Multi-homed Topology

1 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.

2 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.

3 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.

4 Local Preference

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

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

R11:

```
route-map set-lpref permit 10
 set local-preference 150
route-map set-lpref permit 20
!
router bgp 10
 address-family ipv4
  neighbor 10.101.254.1 route-map set-lpref in
 address-family ipv6
  neighbor fd00:101:fe:: route-map set-lpref in
```

R12:

```
route-map set-lpref permit 10
 set local-preference 50
route-map set-lpref permit 20
!
router bgp 10
 address-family ipv4
  neighbor 10.201.254.1 route-map set-lpref in
 address-family ipv6
  neighbor fd00:201:fe:: route-map set-lpref 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. Can you think of a reason why this could be useful?

Check your BGP routes. The next hop should be the P2P address of your NREN's router (except for your own prefix).

```
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?

2. Modify the route map to apply a higher local preference attribute to prefixes originated by your direct peers.

Notice the AS path access list. How does it work?

R11:

```
ip as-path access-list 1 permit ^[0-9]+$
!
no route-map set-lpref
!
route-map set-lpref permit 10
 match as-path 1
 set local-preference 200
route-map set-lpref permit 20
 set local-preference 150
route-map set-lpref permit 30
```

Notice that we need to also apply the route-map to the bi-lateral peering.

R12:

```
ip as-path access-list 1 permit ^[0-9]+$
!
no route-map set-lpref
!
route-map set-lpref permit 10
 match as-path 1
 set local-preference 200
route-map set-lpref permit 20
 set local-preference 50
route-map set-lpref permit 30
!
router bgp 10
 address-family ipv4
  neighbor 10.10.254.6 route-map set-lpref in
 address-family ipv6
  neighbor fd00:10:fe::3 route-map set-lpref 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? (Hint: the path should be direct!)

3. STOP - Checkpoint

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

5 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 10.40.0.0
R11# traceroute 10.40.255.1
R11# show bgp ipv6 unicast fd00:40::/32
R11# traceroute fd00:40:ff::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.

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

R12:

```
ip prefix-list AS10-prefix permit 10.10.0.0/16
!
route-map set-prepend permit 100
 match ip address prefix-list AS10-prefix
 set as-path prepend 10 10
route-map set-prepend permit 200
!
ipv6 prefix-list ipv6-AS10-prefix permit fd00:10::/32
!
route-map ipv6-set-prepend permit 100
 match ipv6 address prefix-list ipv6-AS10-prefix
 set as-path prepend 10 10
route-map ipv6-set-prepend permit 200
```

```
!
router bgp 10
 address-family ipv4
  neighbor 10.201.254.1 route-map set-prepend out
 address-family ipv6
  neighbor fd00:201:fe:: route-map ipv6-set-prepend out
!
```

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

```
R12# clear ip bgp 10.201.254.1 out
R12# clear bgp ipv6 unicast fd00:201:fe:: out
```

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

2. STOP - Checkpoint

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

6 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.

1. 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 10.101.254.1 route-map set-lpref in
 address-family ipv6
  no neighbor fd00:101:fe:: route-map set-lpref in
!
no route-map set-lpref
```

Remember to do the equivalent thing on the other router.

RENs use BGP communities (basically tags) to mark groups of routes together as a unit, which makes it easier for their members 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
 set community 101:99
route-map set-RE-comm permit 20
!
ip community-list 1 permit 100:99
!
route-map set-RE-lpref permit 10
 match community 1
 set local-preference 150
route-map set-RE-lpref permit 20
!
router bgp 101
 address-family ipv4
  neighbor 10.101.254.2 send-community
  neighbor 10.101.254.2 route-map set-RE-comm in
  neighbor 10.101.254.6 send-community
  neighbor 10.101.254.6 route-map set-RE-comm in
  neighbor 10.101.254.10 send-community
  neighbor 10.101.254.10 route-map set-RE-comm in
  neighbor 10.100.254.1 send-community
```

```
  neighbor 10.100.254.1 route-map set-RE-lpref in
 address-family ipv6
  neighbor fd00:101:fe::1 send-community
  neighbor fd00:101:fe::1 route-map set-RE-comm in
  neighbor fd00:101:fe::3 send-community
  neighbor fd00:101:fe::3 route-map set-RE-comm in
  neighbor fd00:101:fe::5 send-community
  neighbor fd00:101:fe::5 route-map set-RE-comm in
  neighbor fd00:100:fe:: send-community
  neighbor fd00:100:fe:: route-map set-RE-lpref in
!
```

NREN2:

```
ip bgp-community new-format
!
route-map set-RE-comm permit 10
 set community 102:99
route-map set-RE-comm permit 20
!
ip community-list 1 permit 100:99
!
route-map set-RE-lpref permit 10
 match community 1
 set local-preference 150
route-map set-RE-lpref permit 20
!
router bgp 102
 address-family ipv4
  neighbor 10.102.254.2 send-community
  neighbor 10.102.254.2 route-map set-RE-comm in
  neighbor 10.102.254.6 send-community
  neighbor 10.102.254.6 route-map set-RE-comm in
  neighbor 10.102.254.10 send-community
  neighbor 10.102.254.10 route-map set-RE-comm in
  neighbor 10.100.254.5 send-community
  neighbor 10.100.254.5 route-map set-RE-lpref in
 address-family ipv6
  neighbor fd00:102:fe::1 send-community
  neighbor fd00:102:fe::1 route-map set-RE-comm in
  neighbor fd00:102:fe::3 send-community
  neighbor fd00:102:fe::3 route-map set-RE-comm in
  neighbor fd00:102:fe::5 send-community
  neighbor fd00:102:fe::5 route-map set-RE-comm in
  neighbor fd00:100:fe::2 send-community
```

```
  neighbor fd00:100:fe::2 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-in permit 10
 match community 1
 set community 100:99 additive
 set local-preference 150
route-map set-RE-comm-in permit 20
!
router bgp 100
 address-family ipv4
  neighbor 10.100.254.2 send-community
  neighbor 10.100.254.2 route-map set-RE-comm-in in
  neighbor 10.100.254.6 send-community
  neighbor 10.100.254.6 route-map set-RE-comm-in in
 address-family ipv6
  neighbor fd00:100:fe::1 send-community
  neighbor fd00:100:fe::1 route-map set-RE-comm-in in
  neighbor fd00:100:fe::3 send-community
  neighbor fd00:100:fe::3 route-map set-RE-comm-in in
```

ISPs will announce additional prefixes to represent the rest of the commodity Internet. Notice that we are prepending "fake" ASNs so that they do not appear to be originated by the ISP.

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 prefix-list v4-commodity-1
 set prepend 65001
route-map set-prepend-commodity permit 20
 match prefix-list v4-commodity-2
 set prepend 65002
route-map set-prepend-commodity permit 30
 match prefix-list v4-commodity-3
 set prepend 65003
route-map set-prepend-commodity permit 40
 match prefix-list v4-commodity-4
 set prepend 65004
route-map set-prepend-commodity permit 50
 match prefix-list v6-commodity-1
 set prepend 65001
route-map set-prepend-commodity permit 60
 match prefix-list v6-commodity-2
 set prepend 65002
route-map set-prepend-commodity permit 70
 match prefix-list v6-commodity-3
 set prepend 65003
route-map set-prepend-commodity permit 80
 match prefix-list v6-commodity-4
 set prepend 65004
route-map set-prepend-commodity permit 90

router bgp 201
 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 10.201.254.2 route-map set-prepend-commodity out
  neighbor 10.201.254.6 route-map set-prepend-commodity out
  neighbor 10.201.254.10 route-map set-prepend-commodity out
  neighbor 10.101.254.13 route-map set-prepend-commodity out
  neighbor 10.251.1.2 route-map set-prepend-commodity out
  neighbor 10.251.1.3 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 fd00:101:fe::6 route-map set-prepend-commodity out
  neighbor fd00:201:fe::1 route-map set-prepend-commodity out
```

```
  neighbor fd00:201:fe::3 route-map set-prepend-commodity out
  neighbor fd00:201:fe::5 route-map set-prepend-commodity out
  neighbor fd00:251:1::2 route-map set-prepend-commodity out
  neighbor fd00:251:1::3 route-map set-prepend-commodity 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 prefix-list v4-commodity-1
 set prepend 65005
route-map set-prepend-commodity permit 20
 match prefix-list v4-commodity-2
 set prepend 65006
route-map set-prepend-commodity permit 30
 match prefix-list v4-commodity-3
 set prepend 65007
route-map set-prepend-commodity permit 40
 match prefix-list v4-commodity-4
 set prepend 65008
route-map set-prepend-commodity permit 50
 match prefix-list v6-commodity-1
 set prepend 65005
route-map set-prepend-commodity permit 60
 match prefix-list v6-commodity-2
 set prepend 65006
```

```
route-map set-prepend-commodity permit 70
 match prefix-list v6-commodity-3
 set prepend 65007
route-map set-prepend-commodity permit 80
 match prefix-list v6-commodity-4
 set prepend 65008
route-map set-prepend-commodity permit 90

router bgp 202
 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 10.202.254.2 route-map set-prepend-commodity out
  neighbor 10.202.254.6 route-map set-prepend-commodity out
  neighbor 10.202.254.10 route-map set-prepend-commodity out
  neighbor 10.102.254.14 route-map set-prepend-commodity out
  neighbor 10.251.1.1 route-map set-prepend-commodity out
  neighbor 10.251.1.3 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 fd00:102:fe::7 route-map set-prepend-commodity out
  neighbor fd00:202:fe::1 route-map set-prepend-commodity out
  neighbor fd00:202:fe::3 route-map set-prepend-commodity out
  neighbor fd00:202:fe::5 route-map set-prepend-commodity out
  neighbor fd00:251:1::1 route-map set-prepend-commodity out
  neighbor fd00:251:1::3 route-map set-prepend-commodity 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
```

2. 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 still need to set a higher local preference on the prefixes originated by our direct peers.

R11:

```
ip bgp-community new-format
!
ip as-path access-list 1 permit ^[0-9]+$
!
ip community-list 1 permit 100:99
ip community-list 1 permit 101:99
!
no route-map set-lpref
!
route-map set-lpref permit 10
 match as-path 1
 set local-preference 200
route-map set-lpref permit 20
 match community 1
 set local-preference 150
route-map set-lpref permit 30
!
router bgp 10
 address-family ipv4
  neighbor 10.101.254.1 route-map set-lpref in
 address-family ipv6
  neighbor fd00:101:fe:: route-map set-lpref in
!
```

Refresh to/from your neighbors:

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

Verify that communities are being set and transmitted:

```
R11#show ip bgp 10.20.0.0
R11#show ip bgp 10.40.0.0
```

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.

7 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 memory requirements and convergence time. The disadvantage is that we may not always follow the best paths, but it might be a good compromise.

R11: Remove the route-map from the previous step.

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
  no neighbor 10.101.254.2 send-community
  no neighbor 10.101.254.6 send-community
  no neighbor 10.101.254.10 send-community
  neighbor 10.101.254.2 route-map send-RE-only out
  neighbor 10.101.254.2 default-originate
  neighbor 10.101.254.6 route-map send-RE-only out
  neighbor 10.101.254.6 default-originate
  neighbor 10.101.254.10 route-map send-RE-only out
  neighbor 10.101.254.10 default-originate
 address-family ipv6
  no neighbor fd00:101:fe::1 send-community
  no neighbor fd00:101:fe::3 send-community
```

```
  no neighbor fd00:101:fe::5 send-community
  neighbor fd00:101:fe::1 route-map send-RE-only out
  neighbor fd00:101:fe::1 default-originate
  neighbor fd00:101:fe::3 route-map send-RE-only out
  neighbor fd00:101:fe::3 default-originate
  neighbor fd00:101:fe::5 route-map send-RE-only out
  neighbor fd00:101:fe::5 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
  no neighbor 10.102.254.2 send-community
  no neighbor 10.102.254.6 send-community
  no neighbor 10.102.254.10 send-community
  neighbor 10.102.254.2 route-map send-RE-only out
  neighbor 10.102.254.2 default-originate
  neighbor 10.102.254.6 route-map send-RE-only out
  neighbor 10.102.254.6 default-originate
  neighbor 10.102.254.10 route-map send-RE-only out
  neighbor 10.102.254.10 default-originate
 address-family ipv6
  no neighbor fd00:102:fe::1 send-community
  no neighbor fd00:102:fe::3 send-community
  no neighbor fd00:102:fe::5 send-community
  neighbor fd00:102:fe::1 route-map send-RE-only out
  neighbor fd00:102:fe::1 default-originate
  neighbor fd00:102:fe::3 route-map send-RE-only out
  neighbor fd00:102:fe::3 default-originate
  neighbor fd00:102:fe::5 route-map send-RE-only out
  neighbor fd00:102:fe::5 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 201
 address-family ipv4
  neighbor 10.201.254.2 default-originate
  neighbor 10.201.254.2 prefix-list default out
  neighbor 10.201.254.6 default-originate
  neighbor 10.201.254.6 prefix-list default out
  neighbor 10.201.254.10 default-originate
  neighbor 10.201.254.10 prefix-list default out
 address-family ipv6
  neighbor FD00:201:FE::1 default-originate
  neighbor FD00:201:FE::1 prefix-list ipv6-default out
  neighbor FD00:201:FE::3 default-originate
  neighbor FD00:201:FE::3 prefix-list ipv6-default out
  neighbor FD00:201:FE::5 default-originate
  neighbor FD00:201:FE::5 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 202
 address-family ipv4
  neighbor 10.202.254.2 default-originate
  neighbor 10.202.254.2 prefix-list default out
  neighbor 10.202.254.6 default-originate
  neighbor 10.202.254.6 prefix-list default out
  neighbor 10.202.254.10 default-originate
  neighbor 10.202.254.10 prefix-list default out
 address-family ipv6
  neighbor FD00:202:FE::1 default-originate
  neighbor FD00:202:FE::1 prefix-list ipv6-default out
  neighbor FD00:202:FE::3 default-originate
  neighbor FD00:202:FE::3 prefix-list ipv6-default out
  neighbor FD00:202:FE::5 default-originate
  neighbor FD00:202:FE::5 prefix-list ipv6-default out
!
```

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

```
R11#show ip bgp neighbors 10.101.254.1 routes
R11#show bgp ipv6 uni neighbors fd00:101:fe:: routes
```

```
R11#show ip route 0.0.0.0 0.0.0.0
R11#show ipv6 route ::/0

R12#show ip bgp neighbors 10.201.254.1 routes
R12#show bgp ipv6 uni neighbors fd00:201:fe:: 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. This is an OK situation. But let's say that we want the ISP to handle all the non-R&E outbound traffic.

Configure your RX2 router to assign a higher local preference to the default announced by the ISP:

R12:

```
ip prefix-list default permit 0.0.0.0/0
ipv6 prefix-list ipv6-default permit ::/0
!
route-map set-lpref-default permit 10
 match ip address prefix-list default
 set local-preference 150
!
route-map set-lpref-ipv6-default permit 10
 match ip address prefix-list ipv6-default
 set local-preference 150
!
router bgp 10
 address-family ipv4
  neighbor 10.201.254.1 route-map set-lpref-default in
 address-family ipv6
  neighbor fd00:201:fe:: route-map set-lpref-ipv6-default in
!
```

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 shrinked?

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