

BGP Policy Lab - Communities

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 BGP Communities and related BGP operational commands.

Pre-requisites

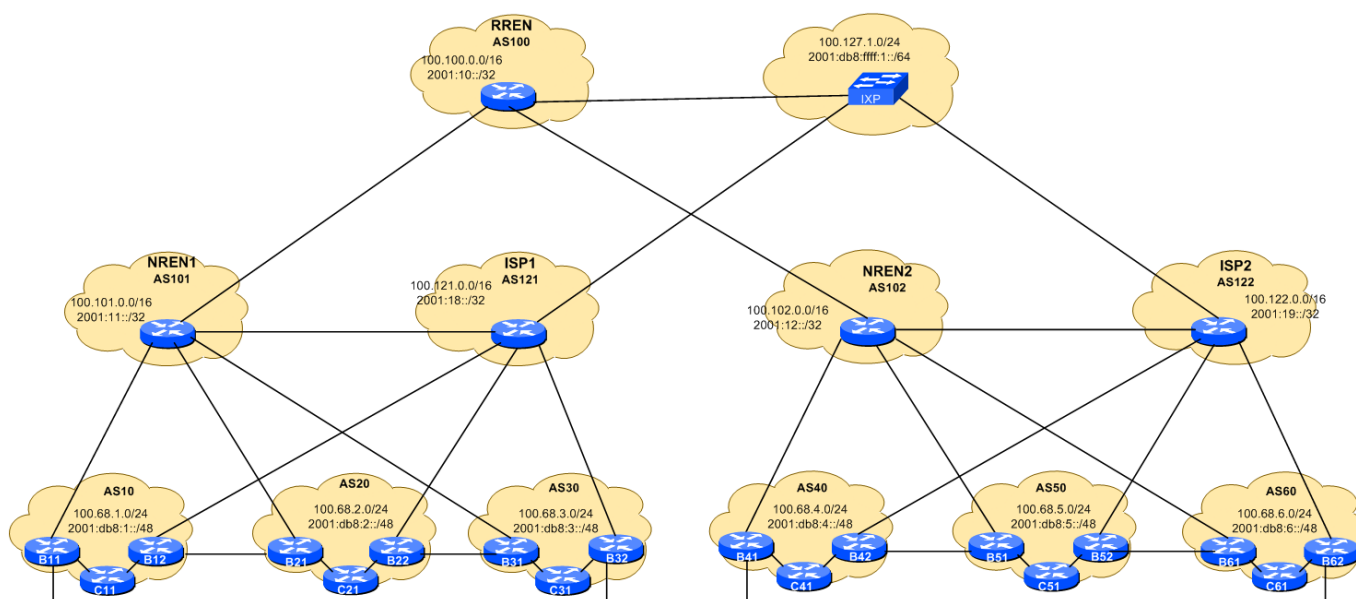
This exercise builds upon the configurations implemented in the basic BGP routing lab. You must:

- 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.

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.

Tidy up earlier Local Preference configuration

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). Here is one example:

B11:

```
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. So both BX1 and BX2 should have the route-map setting local preference removed, as per the above example.

Configuring Community Support on NRENs and RREN

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.

Setting up the ISPs to announce commodity routes

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 100.101.2.1 route-map set-prepend-commodity out  
    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  
  !  
  address-family ipv6  
    network 2001:db8::/32  
    network 2001:db9::/32  
    network 2001:dba::/32  
    network 2001:dbb::/32  
    neighbor 2001:11:0:20:: route-map set-prepend-v6commodity out  
    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
```

```
!
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 100.102.2.1 route-map set-prepend-commodity out
    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
  address-family ipv6
    network 2001:dbc::/32
    network 2001:dbd::/32
    network 2001:dbe::/32
    network 2001:dbf::/32
    neighbor 2001:12:0:20:: route-map set-prepend-v6commodity out
    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
  !
  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
```

Local Preference Configuration using BGP Communities

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 B11:

```
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 neighbours:

```
clear ip bgp external in
clear bgp ipv6 unicast external in
```

Verify that communities are being set and transmitted by the NREN - for example on B11:

```
B11# show ip bgp 100.68.2.0
B11# show ip bgp 100.68.4.0
```

Also, look at all the prefixes which have community set in them. This is a good way of checking that everything is working properly.

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

Do you see all the R&E routes now?

Question: Why do some have the best path and others do not?

Now we set all the prefixes transited by the ISP to us to local preference 80, as per the earlier table. Here is an example configuration:

On B12:

```
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 neighbours:

```
clear ip bgp external in  
clear bgp ipv6 unicast external in
```

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.

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

Appendix - BGP Table

Included for completeness and to aid discussion, here is the BGP table as seen on C21 at the end of this lab exercise.

```
C21#sh ip bgp  
BGP table version is 29, local router ID is 100.68.2.3
```

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	100.68.1.0/24	100.68.2.1	0	200	0	10 i
* i	100.68.2.0/24	100.68.2.2	0	100	0	i
* i		100.68.2.1	0	100	0	i
*>		0.0.0.0	0		32768	i
*>i	100.68.3.0/24	100.68.2.2	0	200	0	30 i
*>i	100.68.4.0/24	100.68.2.1	0	150	0	101 100 102
40 i						
*>i	100.68.5.0/24	100.68.2.1	0	150	0	101 100 102
50 i						
*>i	100.68.6.0/24	100.68.2.1	0	150	0	101 100 102
60 i						
*>i	100.100.0.0/16	100.68.2.1	0	150	0	101 100 i
*>i	100.101.0.0/16	100.68.2.1	0	150	0	101 i
*>i	100.102.0.0/16	100.68.2.1	0	150	0	101 100 102 i
*>i	100.121.0.0/16	100.68.2.2	0	100	0	121 i
*>i	100.122.0.0/16	100.68.2.2	0	80	0	121 122 i
*>i	172.16.0.0	100.68.2.2	0	80	0	121 65001 i
*>i	172.17.0.0	100.68.2.2	0	80	0	121 65002 i
*>i	172.18.0.0	100.68.2.2	0	80	0	121 65003 i
*>i	172.19.0.0	100.68.2.2	0	80	0	121 65004 i
*>i	172.20.0.0	100.68.2.2	0	80	0	121 122 65005
i						
*>i	172.21.0.0	100.68.2.2	0	80	0	121 122 65006
i						
*>i	172.22.0.0	100.68.2.2	0	80	0	121 122 65007
i						
*>i	172.23.0.0	100.68.2.2	0	80	0	121 122 65008
i						

From:

<https://workshops.nsrc.org/dokuwiki/> - **Workshops**

Permanent link:

<https://workshops.nsrc.org/dokuwiki/2016/nsrc-asren-bgp/7-lab-bgp-policy-community>

Last update: **2016/09/04 11:33**

