Lab 2 – BGP route filtering and advanced features

Objective: Using the network concepts of Lab 1, use various configuration methods on BGP peerings to demonstrate neighbour filtering and more advanced IOS features.

Topology:

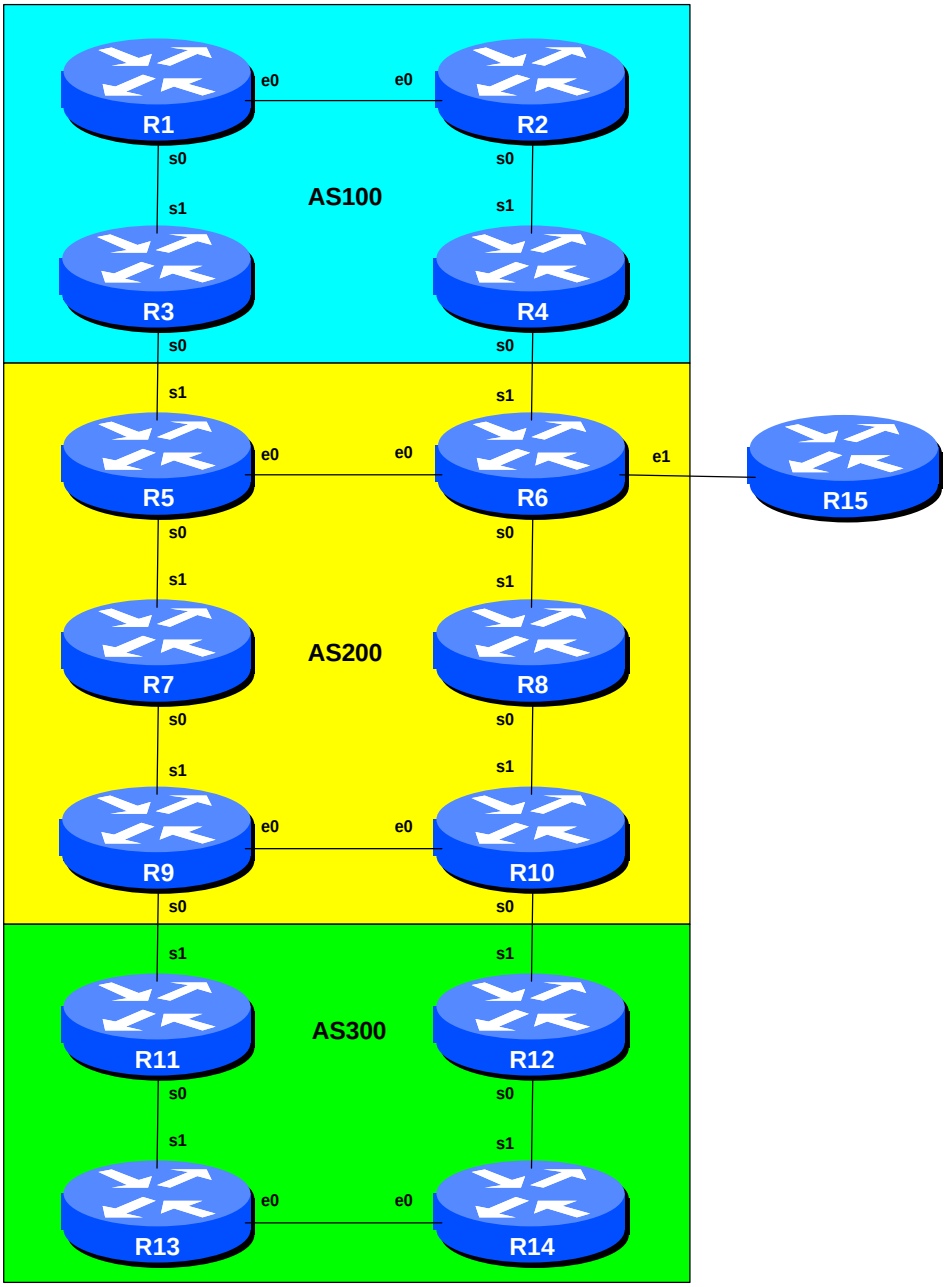

Figure 1 – BGP AS Numbers

Lab Notes

The previous Module provided an introduction to setting up external BGP, but provided no way of controlling which networks are announced to which AS. The purpose of this module is to introduce the student to some of the types of routing policy which are available using BGP.

This lab will use the same IP address ranges, configuration, and topology as the previous lab. The configurations should have all been retained, as requested at the end of the last module.

Lab Exercise

1. **Configure filter based on network address:** This step configures route prefix filtering based on network address. This is done using prefix-lists, and is one method of controlling networks which are exchanged in BGP peerings. The aim here is to configure eBGP peerings so that only networks from **neighbouring** ASes are exchanged. Therefore, prefixes in originated by AS300 will not be seen by AS100.

   Example: Router R3 (peering with R5)

```
!
ip prefix-list out-peer permit 220.10.0.0/19
ip prefix-list out-peer deny 0.0.0.0/0 le 32
!
ip prefix-list in-peer permit 221.19.0.0/19
ip prefix-list in-peer deny 0.0.0.0/0 le 32
!
router bgp 100
 no synchronization
 network 220.10.0.0 mask 255.255.224.0
 neighbor <router5> remote-as 200
 neighbor <router5> description eBGP peering with Router5
 neighbor <router5> soft-reconfiguration in
 neighbor <router5> prefix-list out-peer out
 neighbor <router5> prefix-list in-peer in
 no auto-summary
```

   Example: Router R5 (peering with R3)

```
!
ip prefix-list out-peer permit 221.19.0.0/19
ip prefix-list out-peer deny 0.0.0.0/0 le 32
!
ip prefix-list in-peer permit 220.10.0.0/19
ip prefix-list in-peer deny 0.0.0.0/0 le 32
!
router bgp 200
 no synchronization
 network 221.19.0.0 mask 255.255.224.0
 neighbor <router7> remote-as 222
 neighbor <router7> description Peering with Router3
```


Cisco Systems Inc
170 West Tasman Drive.
San Jose, CA 95134-1706
Phone: +1 408 526-4000
Fax: +1 408 536-4100

```
neighbor <router7> soft-reconfiguration in
neighbor <router7> prefix-list out-peer out
neighbor <router7> prefix-list in-peer in
no auto-summary
```

Note: an IOS prefix-list always has an implicit *deny any* as the last statement even though it is not listed in the configuration. Some ISPs add the implicit *deny any* as they consider it good practice and a security precaution.

Note: these prefix-lists are only applied to peerings with other ASes. These are called **external** peerings (using eBGP). There is usually no need to apply such filters for iBGP peerings.

To implement the above filter on the peering, do *clear ip bgp <neighbour address> soft*. Don't forget, use incremental BGP changes, not hard resets!

Q: why do you need the command *clear ip bgp <neighbour address> soft**?***

A: The BGP process only announces changes in the network routing table to a neighbour. If a new route is added to the network, it will be announced by BGP. However, adding an access-list to the BGP peering requires that the routing table currently being announced to the peer be run through the access-list. This is not a dynamic process as the configuration of the peering has been changed, so *clear ip bgp <neighbour address>* is required to restart the peering session with the neighbour.

2. **Inject other prefixes.** Each AS should now try and inject other prefixes into the BGP table. Recall how to do this. Simply create a BGP network statement with a prefix from RFC1918 address space (for example), plus a static route to the null0 interface, and the prefix will appear in the BGP process. Also, take a subprefix of the address block and inject that into BGP. Each Router team should inject some prefixes as suggested. For example, on Router 1:

```
!
router bgp 100
 no synchronization
 network 172.16.0.0 mask 255.255.255.0
 network 220.10.4.0 mask 255.255.254.0
 no auto-summary
!
ip route 172.16.0.0 255.255.255.0 null0 250
ip route 220.10.4.0 255.255.254.0 null0 250
!
```

will inject the 172.16.0/24 prefix and the 220.10.4.0/23 prefix into the BGP table in AS100. Are these two prefixes visible in AS200? If so, why? If not, why not?

This shows the significance and the necessity of prefix filters on eBGP peerings. Only aggregates should be announced to the Internet routing table unless there are special circumstances, which will be demonstrated in later Labs.

Checkpoint #2: *call the lab assistant to verify the connectivity. Each router team should check peerings to see the effect of this step. Use the "show ip bgp neigh x.x.x.x advertise|route" commands. Once complete and the lab instructors give you the go-ahead, remove the prefix-list configuration and the extra injected prefixes, and carry on with the next step.*

3. **Configure filter based on AS path attribute:** This step configures route prefix filtering based on AS path. This is done using AS path access-lists, and is another method of controlling networks which are exchanged in BGP peerings. Note again that AS100 will not be able to see the prefixes originated by AS300 – and vice-versa.

   Outgoing direction example – Router R9

```
ip as-path access-list 2 permit ^$
ip as-path access-list 3 permit ^300$
!
router bgp 200
 neighbor <router11> remote-as 300
 neighbor <router11> description Peering with Router11
 neighbor <router11> filter-list 2 out
 neighbor <router11> filter-list 3 in
```

   Incoming direction example – Router R11

```
ip as-path access-list 2 permit ^$
ip as-path access-list 3 permit ^200$
!
router bgp 300
 neighbor <router9> remote-as 200
 neighbor <router9> description Peering with Router9
 neighbor <router9> filter-list 2 out
 neighbor <router9> filter-list 3 in
```

   To verify that the regular expression works as intended, use the EXEC command "***show ip bgp regexp*** *<regular-expression>*" to display all the paths that match the specified regular expression. Don't forget that the command *clear ip bgp <neighbour address>* is required to implement this filter.

4. **Inject other prefixes.** Each AS should now try and inject other prefixes into the BGP table. Recall how to do this from earlier in the lab. Simply create a BGP network statement with a prefix from RFC1918 address space (for example), plus a static route to the null0 interface, and the prefix will appear in the BGP process. Also, take a subprefix of the address block and inject that into BGP. Each Router team should inject some prefixes as suggested. For example, on Router 1:

```
!
router bgp 100
 no synchronization
```

```
 network 172.16.0.0 mask 255.255.255.0
 network 220.10.4.0 mask 255.255.254.0
 no auto-summary
!
ip route 172.16.0.0 255.255.255.0 null0 250
ip route 220.10.4.0 255.255.254.0 null0 250
!
```

will inject the 172.16.0/24 prefix and the 220.10.4.0/23 prefix into the BGP table in AS100. Are these two prefixes visible in AS200? If so, why? If not, why not?

This shows a key issue with AS path filtering in eBGP peerings. Only the AS path is filtered – the filter assumes that the neighbouring AS will only announce prefixes it is entitled to announce. This is why most ISPs implement both AS path AND prefix filters on all eBGP peerings.

Checkpoint #3: *call the lab assistant to verify the connectivity. Each router team should again check their peerings to see what the effect is this time. Once complete, retain the filter-list configuration and move on to the next step.*

5. **Combining Prefix Filter and AS-PATH Filter.** Retaining the configuration used previously, now implement prefix filters as in the earlier steps so that only the aggregate blocks are passed between ASes. Use the configuration examples in Step 4 as a hint.

   Example: Router R4 (peering with R6)

```
!
ip prefix-list out-peer permit 220.10.0.0/19
ip prefix-list out-peer deny 0.0.0.0/0 le 32
!
ip prefix-list in-peer permit 221.19.0.0/19
ip prefix-list in-peer deny 0.0.0.0/0 le 32
!
ip as-path access-list 2 permit ^$
ip as-path access-list 3 permit ^200$
!
router bgp 100
 no synchronization
 network 220.10.0.0 mask 255.255.224.0
 neighbor <router6> remote-as 200
 neighbor <router6> description eBGP peering with Router6
 neighbor <router6> soft-reconfiguration in
 neighbor <router6> prefix-list out-peer out
 neighbor <router6> prefix-list in-peer in
 neighbor <router6> filter-list 2 out
 neighbor <router6> filter-list 3 in
 no auto-summary
!
```

   Are the subprefixes prefixes injected by AS100 still visible in AS200? If so, why? If not, why not?

Checkpoint #4: *call the lab assistant to verify the connectivity. Each router team should again check their peerings to see what the effect is this time. Once complete, remove the filter-list and prefix-list configurations and move on to the next step.*

6. **Setting BGP Communities.** Each router team should assign a community to the /19 network block they have been allocated in Lab 1. Review the BGP documentation to find out how to do this. Each router should set a community of format *[AS number]:4000*. (There is no significance attached to the 4000 number other than this is the number we will use to tag all prefixes which are to be announced to eBGP peers.)

   Example for Router R1:

```
ip bgp-community new-format
!
route-map community-tag permit 10
 set community 100:4000
!
router bgp 100
 no synchronization
 network 220.19.0.0 mask 255.255.224.0 route-map community-tag
 neighbor ibgp-peers peer-group
 neighbor ibgp-peers description Peer-Group used for all iBGP peers in AS100
 neighbor ibgp-peers remote-as 100
 neighbor ibgp-peers update-source loopback 0
 neighbor ibgp-peers send-community
 neighbor <router2> peer-group ibgp-peers
 neighbor <router3> peer-group ibgp-peers
 neighbor <router4> peer-group ibgp-peers
 no auto-summary
!
ip route 220.19.0.0 255.255.224.0 null0 250
```

   Check that the network appears with its community in the BGP routing table. Don't forget the "send-community" directive on all iBGP and eBGP peerings! IOS does not pass on communities to BGP peers by default.

   Check that all the prefixes originated in the workshop lab are present in the network. There currently should be no filters present on the eBGP peerings between the 3 ASes.

Checkpoint #5: *call the lab assistant and demonstrate how the community has been set for your network using the "show ip bgp" commands. Also, demonstrate that you can see the communities set by your internal and external BGP peers.*

7. **Configure incoming prefix filter based on route-map on community attribute.** The aim here is to only accept networks which are received from the neighbouring external BGP peer. (This is

Cisco Systems Inc
170 West Tasman Drive.
San Jose, CA 95134-1706
Phone: +1 408 526-4000
Fax: +1 408 536-4100

similar to what was being attempted in the previous steps with prefix and AS path filtering.) For example, R6 should only accept the network aggregate originated by AS100, and should use the knowledge of the community attached to the network to achieve this.

Example on Router R6:

```
route-map infilter permit 10
 match community <community-list-no>
!
ip community-list <community-list-no> permit 100:4000
!
router bgp 200
 neighbor <router4> remote-as 100
 neighbor <router4> description eBGP with Router4
 neighbor <router4> send-community
 neighbor <router4> route-map infilter in
!
```

Example on Router R12:

```
route-map infilter permit 10
 match community <community-list-no>
!
ip community-list <community-list-no> permit 100:4000
ip community-list <community-list-no> permit 200:4000
!
router bgp 300
 neighbor <router10> remote-as 100
 neighbor <router10> description eBGP with Router10
 neighbor <router10> send-community
 neighbor <router10> route-map infilter in
!
```

The <community-list-no> choice is up to each router team – it is not announced in any BGP peering or used in any other way apart from identifying the community-list (compare with the access-list number).

Note that Routers 3, 4, 11 and 12 will be filtering two communities. The aim at the end of this step is to have full connectivity across the workshop lab, but all prefixes apart from the aggregates originated by each AS. (If Router15 is connected to the lab, Router6 should also attach the community 200:4000 to 192.168.1.0/24 network.)

What is seen in the BGP table now? Why are the subprefixes and the RFC1918 prefixes no longer seen by the neighbouring AS?

8. Configure the BGP route-dampening feature.

Example:

```
router bgp 200
 bgp dampening
```

Talk to other ASes and try triggering a network flap by removing and then reinstalling some static routes or shutting down and reactivating interfaces which have eBGP peerings across them. Then check *show ip bgp dampening* to see the penalty assigned to prefixes. "Flapping" the network a few times will result in the network being dampened by the local router.

Try changing the dampening parameters and check the result.

Example:

```
router bgp 200
 bgp dampening 10 200 400 30
```

Cisco's default values are **bgp dampening 15 750 2000 30**. Each flap attracts a fixed penalty value of 1000.

The options for the *bgp dampening* command are:

10 – half-life (in minutes)
200 – reuse limit (penalty value at which the route will be reused)
400 – suppress at (penalty value at which the route will be suppressed)
30 – suppress limit (max time in minutes the route will be suppressed)

Penalty decays at a granularity of 5 seconds. So every 5 seconds, the penalty will be reduced according to the half-life time (exponential decay). One flap attracts a penalty of 1000 units. So in this example, the route will be suppressed the first time it flaps. Once 800 units have been subtracted from the flap, the route will be reannounced. Once the penalty drops below half the reuse limit, the flap information is deleted from the flap table – the next flap will start counting from zero again.

9. Controlled BGP Dampening:

Try using a route-map for more precise control of dampening. This is a more common occurrence on the Internet, as many ISPs want to apply dampening to the routes which cause the biggest stability problems, not simply apply a blanket value everywhere.

This example dampens the routes listed in prefix-list damp-prefix.

```
ip prefix-list damp-prefix permit x.x.x.x m.m.m.m
!
route-map damp-some permit 10
 match ip address prefix-list damp-prefix
!
```


Cisco Systems Inc
170 West Tasman Drive.
San Jose, CA 95134-1706
Phone: +1 408 526-4000
Fax: +1 408 536-4100

```
router bgp 200
 bgp dampening route-map damp-some
```

Checkpoint #7: *call the lab assistant to verify the operation of dampening and its configuration.*

10. **Summary:** This module has introduced some of the basic features available to configure BGP peerings in Cisco IOS. The reader is encouraged to try further permutations of the configuration examples given here. Community usage is gaining in popularity as the feature is now recognised to give considerable advantages in controlling routing policy between different ASes. BGP route flap dampening, soft reconfiguration, and peer-groups are also widely used in ISP backbones as they considerably ease administration and configuration of an operational network. For recommended operational parameters for BGP route flap dampening configuration for Internet connected sites, the reader should consult http://www.ripe.net/docs/ripe-210.html and the Cisco IOS Essentials document found at http://www.cisco.com/public/cons/isp. For more information about the algorithms behind Cisco's implementation of route-flap dampening, the reader should consult RFC2439.

Review Questions:

CONFIGURATION NOTES

Documentation is critical! You should record the configuration at each ***Checkpoint***, as well as the configuration at the end of the module.

Cisco Systems Inc
170 West Tasman Drive.
San Jose, CA 95134-1706
Phone: +1 408 526-4000
Fax: +1 408 536-4100