iBGP, and Basic eBGP Configuration

Objective: Simulate four different interconnected ISP backbones using a combination of OSPF, internal BGP, and external BGP.

Prerequisites: Module 1

Topology :

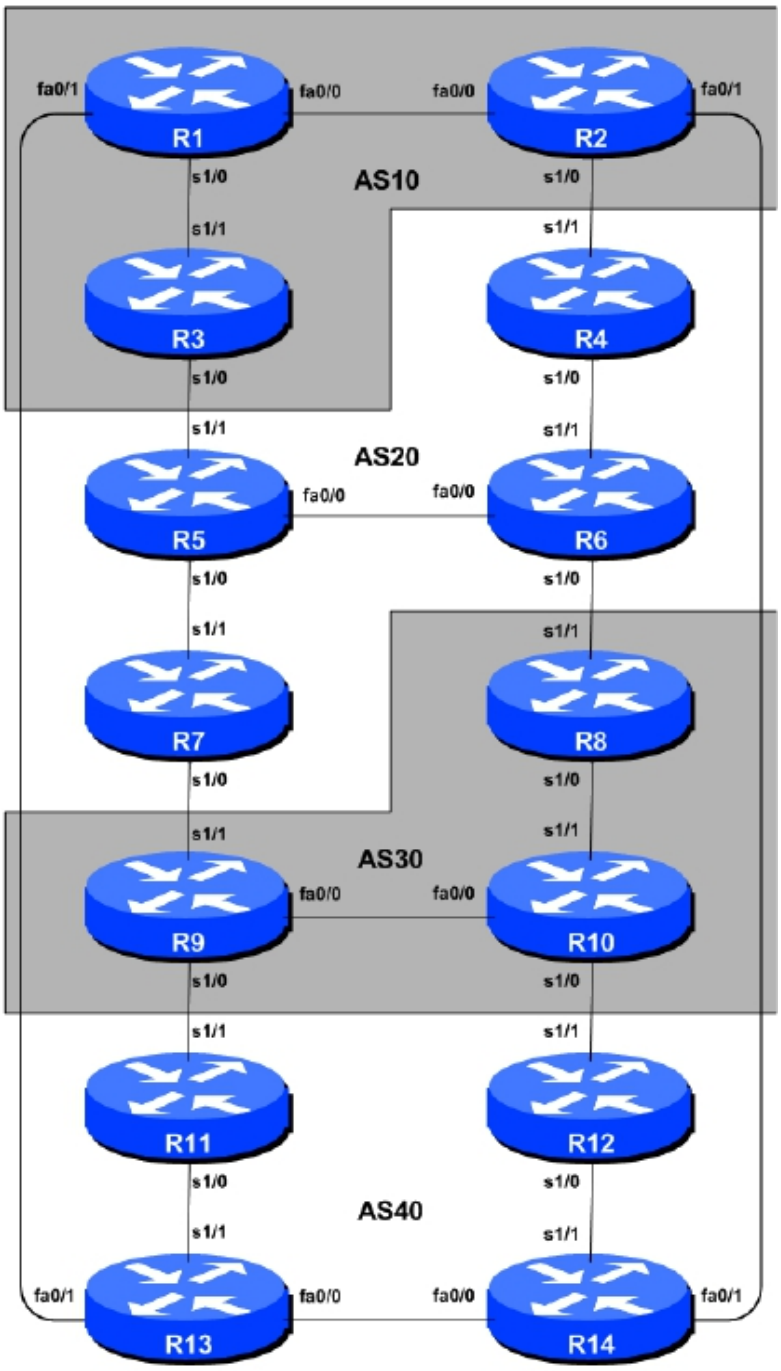

Figure 1 – BGP AS Numbers

Lab Notes

The purpose of this module is to introduce the student to external BGP (eBGP). This is the relationship between different autonomous systems in an "Internet". The classroom is split into four distinct networks, and the teams belonging to each network work together as a typical ISP. Each AS has two links to its neighbouring ASes, and this feature will used throughout a significant portion of this workshop.

The connectivity shown in the diagrams represents links between AS's. It is assumed that all the routers within an AS are physically connected to each other as per Figure 1.

Lab Exercises

1. Connect routers as shown in Figure 1. All routers within an AS must be physically connected and reachable. The relationship between the ASes is as drawn in Figure 2 and gives a view which can be related to the "real world".

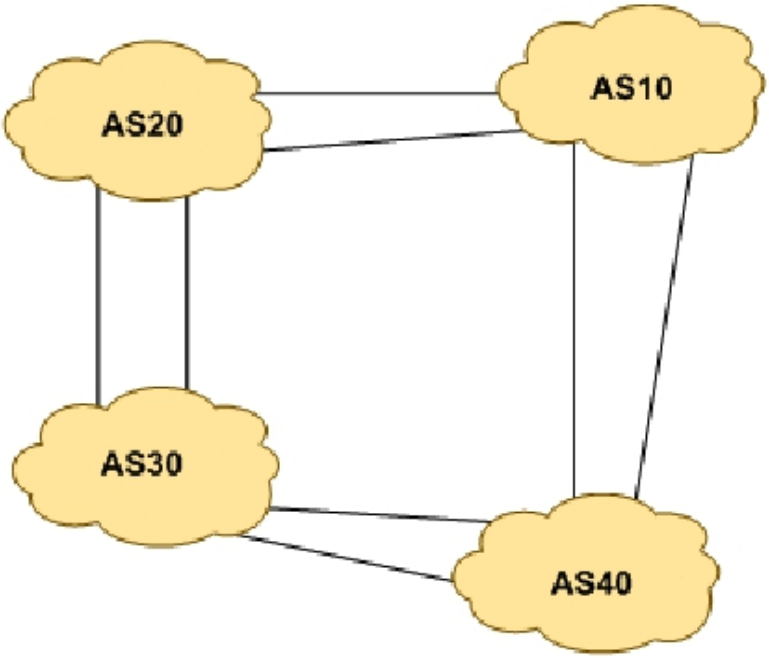

Figure 2 – AS relationship

2. **Re-configure BGP and OSPF.** On each router, remove the BGP and OSPF processes from the previous modules by using the following two commands:

```
Router1(config)#  no  router  bgp  10
Router1(config)#  no  router  ospf  41
```

This will clear the BGP and OSPF configuration ready for the current module.

3. **Remove the IP addressing.** Before we proceed to even think about configuring the routing protocols to match the above AS layout, we need to tidy up the addressing from previous modules. In this step we remove the IP addresses from all physical interfaces, and the loopback

interface. This effectively takes the lab back to the configuration it had prior to step 10 in Module 1. Don't forget to remove **all** the IP addressing.

4. **IP Addressing.** As we did in step 10 of Module 1, we need to come up with a sensible and scalable addressing plan for each AS in this network. Each AS gets their own address block, again a /20 (typical minimum allocation for a starter ISP). This address block should be assigned to links and loopbacks on the routers making up each ASN. The allocations are as follows:

AS10	**10.10.0.0/20**	**AS30**	**10.30.0.0/20**
AS20	**10.20.0.0/20**	**AS40**	**10.40.0.0/20**

Again we need to divide up each address block so that we have customer address space, network infrastructure address space, and some space for loopbacks. Figure 3 below reminds how this could be done:

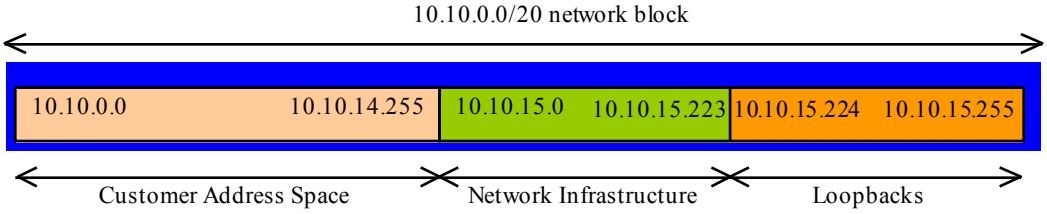

Figure 3 – Dividing allocated block of /20 into Customer, Infrastructure and Loopbacks

Please refer to the accompanying hand out for the address plan which should be used for this module onwards – it is entitled "Addressing Plan – Modules 6 to 9". As with Module 1, configure the addresses on each interface which will be used for this module, and check basic IP connectivity with your immediately adjacent neighbours.

5. **Router Loopback Interface Addressing.** We have set aside a /27 for loopbacks even though each AS has either 3 or 4 routers in it – this leaves more than sufficient room for future expansion. The loopback address assignments which will be used for this module are below:

Router1	**10.10.15.224**	**Router8**	**10.30.15.224**
Router2	**10.10.15.225**	**Router9**	**10.30.15.225**
Router3	**10.10.15.226**	**Router10**	**10.30.15.226**
Router4	**10.20.15.224**	**Router11**	**10.40.15.224**
Router5	**10.20.15.225**	**Router12**	**10.40.15.225**
Router6	**10.20.15.226**	**Router13**	**10.40.15.226**
Router7	**10.20.15.227**	**Router14**	**10.40.15.227**

6. **Configure OSPF on the routers within each AS.** In each AS configure OSPF routing. This means setting up the OSPF process, marking **internal** interfaces as non-passive, and then configuring each **internal** interface and the loopback with the OSPF process. All the routers in an AS will be in the same OSPF *area 0* and use the same OSPF process ID.

OSPF should be configured on internal interfaces **only**. You do not want to set up adjacencies with devices outside your AS. Make sure that there are no *ip ospf* commands on external interfaces. A side effect of this is that external link addresses will not appear in the IGP (see the next section discussion iBGP deployment).

For example, Router 1, with two interfaces connecting to other routers in their AS, would have the following:

```
Router1 (config)# router ospf 10
Router1 (config-router)# passive-interface default
Router1 (config-router)# no passive-interface fastethernet 0/0
Router1 (config-router)# no passive-interface serial 1/0
Router1 (config-router)# log-adjacency-changes
!
Router1 (config)# interface fastethernet 0/0
Router1 (config-interface)# ip ospf 10 area 0
!
Router1 (config-interface)# interface serial 1/0
Router1 (config-interface)# ip ospf 10 area 0
!
Router1 (config-interface)# interface loopback 0
Router1 (config-interface)# ip ospf 10 area 0
!
```

Notes:

- *Passive-interface default* makes sure that OSPF does not attempt to set up adjacencies on any interfaces apart from those specified by the *no passive-interface* commands.
- The number following "*router ospf*" is a process ID and is used inside the router only (so it can be any number). But for this lab we recommend the OSPF process ID be the same as the AS number (which is the convention used by a number of ISPs).

7. **OSPF on Point-to-Point Ethernet Links.** One feature mentioned in the OSPF for ISPs presentations was the option to modify OSPF's behaviour on point-to-point broadcast media links, such as Ethernet, when there are only two devices on that media. If we declare such a situation to be point to point, then OSPF does not try and determine a designated or backup designated router; furthermore, there will be an improvement/simplification in SPF calculations and memory requirements on the router.

   Those router teams which have OSPF configured over an Ethernet interface should now convert OSPF to point to point mode, for example:

```
Router1 (config)# interface fastethernet 0/0
Router1 (config-interface)# ip ospf network point-to-point
```

   The result will be that the DR or BDR entry in the status column of *"show ip ospf neighbor"* will disappear, to be replaced with a FULL. The link is now treated like a point-to-point serial connection.

8. **Ping Test.** Check the routes via OSPF. Make sure you can see all the networks within your AS. Ping all loopback interfaces within your AS. Use the "*show ip ospf neighbor"* and *"show ip route"* commands. If you cannot see the other routers in your AS, you will not be able to bring up BGP in the next steps.

9. **Save the configuration.** Don't forget to save the configuration to NVRAM!

Checkpoint #1 : *call the lab assistant to verify the connectivity.*

10. **Intra Area Authentication – Part 1.** OSPF supports neighbour authentication. This is quite important inside ISP networks to prevent the introduction of improperly configured or unintended equipment.

    Each router team will turn on authentication. This first step will enable the area to support authentication using the *area N authentication message-digest* command. As soon as this is done, all adjacencies will be expected to use neighbour authentication.

    An example configuration for Router6 might be:

```
Router6(config)#router  ospf  20
Router6(config-router)#  area  0 authentication  message-digest
```

    Note that this does not affect the actual adjacencies on the routers – it only tells the router that the area mentioned will use authentication, if it is configured.

11. **Intra Area Authentication – Part 2**. Now that support for authentication in each area has been configured, the second step is to actually set the authentication password to be used, and the interface it has to be used on. The password that should be used for all areas in this example is *cisco*. MD5 encryption should be used rather than the standard simple encryption – to do this, use the *message-digest-key* sub-interface command.

    An example configuration for Router6 might be:

```
Router6(config)#  interface  fastethernet  0/0
Router6(config-if)#  ip  ospf  message-digest-key  1 md5  cisco
!
Router6(config-if)#  interface  serial  1/1
Router6(config-if)#   ip  ospf  message-digest-key  1  md5  cisco
```

    Notice now that the OSPF adjacencies do not come up unless the neighbouring router has also entered the same configuration and key. Notice also how the OSPF adjacencies were reset as the configuration was entered – security is being introduced, so the adjacencies are reset.

    Note: the *message-digest-key* allows up to 255 keys to be set per interface. It is generally not recommended to set more than one per interface, as the router will try and communicate with its neighbours using all keys. If a key needs to be upgraded, common practice then is to set a second key, allowing a graceful changeover without compromising the functioning of the network. Once all the routers on the network are using the new key, the old one should be removed.

12. **Final check.** Use the various "*show ip ospf*" commands to see the OSPF status of the lab network now. Check the routing and the routing table. If you are missing any adjacencies, work with your neighbouring routers in your AS to work out why, and what might have gone wrong with the neighbour authentication.

    Note: Wherever an OSPF session is configured from now on in the workshop, all Router Teams MUST use passwords on these OSPF sessions.

Checkpoint #2 : *call the lab assistant to verify the connectivity.*

13. **Configure iBGP peering between routers within an AS.** Use the loopback address for the iBGP peerings. Also, configure the *network* command to add the address block assigned to each AS for advertisement in BGP. Each router team should announce this address block from their routers.

```
Router1 (config)# router bgp 10
Router1 (config-router)# distance bgp 200 200 200
Router1 (config-router)# no synchronization
Router1 (config-router)# network 10.10.0.0 mask 255.255.240.0
Router1 (config-router)# neighbor 10.10.15.225 remote-as 10
Router1 (config-router)# neighbor 10.10.15.225 update-source loopback 0
Router1 (config-router)# neighbor 10.10.15.225 next-hop-self
Router1 (config-router)# neighbor 10.10.15.225 description iBGP Link to R2
Router1 (config-router)# neighbor 10.10.15.226 remote-as 10
Router1 (config-router)# neighbor 10.10.15.226 update-source loopback 0
Router1 (config-router)# neighbor 10.10.15.226 next-hop-self
Router1 (config-router)# neighbor 10.10.15.226 description iBGP Link to R3
Router1 (config-router)# no auto-summary
Router1 (config-router)# exit
Router1 (config)# ip route 10.10.0.0 255.255.240.0 Null0
```

14. **Next-hop-self configuration.** The previous step introduced the next-hop-self configuration command. Referring to the BGP presentation, the next-hop-self configuration makes the iBGP speaking router use the iBGP source address (in this case the loopback) rather than the external next-hop address (as per the BGP specification). This is industry best practice and means that ISPs do not need to carry external next-hops in their IGP.

15. **Test internal BGP connectivity.** Use the BGP Show commands to ensure you are receiving everyone's routes from within your AS.

16. **Configure passwords on the iBGP sessions.** Passwords should now be configured on the iBGP sessions. Review the presentation why this is necessary. Agree amongst all your team members in your AS what the password should be on the iBGP session, and then apply it to all the iBGP peerings on your router. For example, on Router2's peering with Router3, with "cisco" used as the password:

```
Router2 (config)# router bgp 10
Router2 (config-router)# neighbor 10.10.15.226 password cisco
```

IOS currently resets the iBGP session between you and your neighbouring router whenever an MD5 password is added. So when passwords are added to BGP sessions on live operational networks, this work should be done during a maintenance period when customers know to expect disruptions to service. In the workshop lab, it doesn't matter so much. (Future IOS releases will avoid having this rather serious service disruption.)

Watch the router logs – with the BGP session neighbour changes being logged, any mismatch in the password should be easy to spot. A missing password on one side of the BGP session will result in the neighbouring router producing these errors:

```
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179 to 2.2.2.2:11272
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179 to 2.2.2.2:11272
%TCP-6-BADAUTH: No MD5 digest from 3.3.3.3:179 to 2.2.2.2:11272
```

whereas a mismatch in the configured passwords will result in these messages:

```
%TCP-6-BADAUTH:  Invalid  MD5  digest  from  3.3.3.3:11024  to  2.2.2.2:179
%TCP-6-BADAUTH:  Invalid  MD5  digest  from  3.3.3.3:11024  to  2.2.2.2:179
%TCP-6-BADAUTH:  Invalid  MD5  digest  from  3.3.3.3:11024  to  2.2.2.2:179
```

Checkpoint #3: *Call the lab assistant and demonstrate the password as set on the iBGP session. Once confirmed by the lab assistant, move on to the next steps.*

17. **Configure eBGP peering.** Use Figure 1 to determine the links between the AS's. Addressing for eBGP links between 2 AS's will use the point-to-point interface addresses, ***NOT*** the loopback addresses (review the BGP presentation if you don't understand why).

```
Router1  (config)#  router  bgp  10
Router1  (config-router)#  neighbor  10.10.15.14  remote-as  40
Router1  (config-router)#  neighbor  10.10.15.14  description  eBGP  to  Router13
```

Use the BGP Show commands to ensure you are sending and receiving the BGP advertisements from your eBGP neighbours.

Q. Why can't the loopback interfaces be used for the eBGP peerings?

A. The IP address of a router's loopback interface is not known to external BGP peers, so the external peers will have no way of knowing how to contact each other to establish the peering.

Q. Which BGP show command allows you to see the state of the BGP connection to your peer?

A. Try *show ip bgp neighbor x.x.x.x* – this will give detailed information about the state of the peer. There are subcommands of this one, giving more information about the peering.

Q. Which BGP Show command will allow you to see exactly which networks you are advertising and receiving from your eBGP peers?

A. Try *show ip bgp neighbor x.x.x.x route* – this will show which routes you are receiving from your peer. Likewise, replacing *route* with *advertised-routes* will list the networks which are being announced to your peer. (Note that in general ISP operational practice, there are caveats here – if you apply route-maps and some BGP policies, these will not be processed by the *advertised-routes* command. Use the *advertised-routes* subcommand with due caution.)

18. **Configure passwords on the eBGP session.** Passwords should now be configured on the eBGP sessions between your and your neighbouring ASes. Agree between you and your neighbouring AS what the password should be on the eBGP session, and then apply it to the eBGP peering. For example, on Router2's peering with Router4, with "cisco" used as the password:

```
Router2  (config)#  router  bgp  10
Router2  (config-router)#  neighbor  10.10.15.10  password  cisco
```

As previously for the iBGP session, watch the logs for password mismatches, or missing passwords. As with the iBGP sessions previously, you will find that the router will reset the eBGP session as soon as the password is applied.

Note: Wherever a BGP (either iBGP or eBGP) session is configured from now on in the workshop, all Router Teams MUST use passwords on these BGP sessions.

Checkpoint #4: *Call the lab assistant and demonstrate the password as set on the eBGP session. Once confirmed by the lab assistant, move on to the next steps.*

19. **Adding a "customer" route into BGP.** As we did in Module 1, we are now going to add a "customer" route into BGP on each router. We don't have any "customers" as such connected to our routers in the lab, so we are going to simulate the connectivity by simply using a Null0 interface. The "customer" address space that each router team will introduce into the iBGP is listed below – again we will each use a /26, for simplicity's sake.

R1	10.10.0.0/26	R8	10.30.0.0/26
R2	10.10.0.64/26	R9	10.30.0.64/26
R3	10.10.0.128/26	R10	10.30.0.128/26
R4	10.20.0.0/26	R11	10.40.0.0/26
R5	10.20.0.64/26	R12	10.40.0.64/26
R6	10.20.0.128/26	R13	10.40.0.128/26
R7	10.20.0.192/26	R14	10.40.0.192/26

Each team should now set up a static route pointing to the **NULL0** interface for the /26 that they are to originate. Once the static is set up, the team should then add an entry into the BGP table. Here is an example for Router8:

```
Router8 (config)# ip route 10.30.0.0 255.255.255.192 Null0
Router8 (config)# router bgp 30
Router8 (config-router)# network 10.30.0.0 mask 255.255.255.192
```

20. **Check the BGP table.** Are there routes seen via ***show ip bgp***? If not, why not? Once every team in the class has done their configuration, each team should see the aggregate from each AS as well as the fourteen /26s introduced in the previous step. If this is not happening, work with your neighbours to fix the problem.

Checkpoint #5: *Call the lab assistant to verify the connectivity. Use commands such as "show ip route sum", "show ip bgp sum", "show ip bgp", "show ip route", and "show ip bgp neigh x.x.x.x route | advertise". There should be 4 aggregate prefixes (one for each ISP) and the 14 customer /26's in the BGP table.*

21. **The Importance of Aggregation.** Each AS was allocated a /20 address block. It is expected by all Internet operators that any address space an ISP is using is aggregated as much as possible before it is announced to the rest of the Internet. Subdividing the address space inside an AS is perfectly acceptable and obviously very common (as we have done here) – but most operators consider leaking this subdivided address space out to the Internet at large as antisocial and unfriendly behaviour.

Q. How do you automatically aggregate via BGP smaller address blocks from within your network to a larger address block outside your network? ***Hint:*** *Review the BGP documentation.*

A. The "aggregate-address" command is quite often used to achieve this.

We are not doing any filtering or limitation of the announcements of the "customer" address blocks we have introduced into each ASN. This will be one of the goals of the next modules in the workshop.

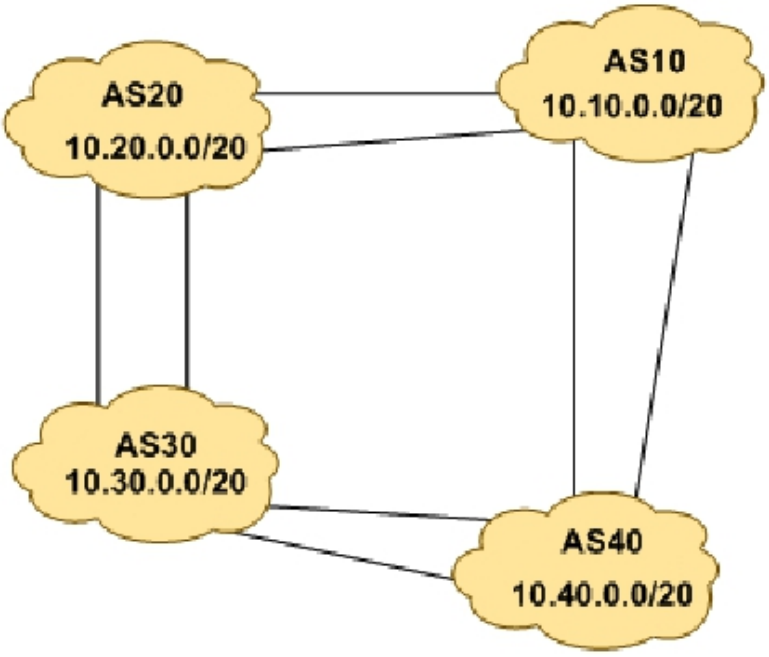

Figure 4 – Aggregates for each ASN

22. **BGP Update Activity (Optional).** Use *debug ip bgp update* to see BGP update activity after clearing a BGP session. To stop the debug running, do *undebug ip bgp update.*

    Warning: it might not be such a good idea to run this debug command on a router receiving the full Internet routing table; using this command in a lab network such as this might show you why!

Review Questions

1. How many *origin types* exist in BGP?

2. List the origin types. **Hint:** Review the BGP presentations.

3. How are they used?

4. Why are passwords necessary on both iBGP and eBGP sessions? What do they protect against?

5. Why is aggregation important for the Internet?