Module 9 – BGP Configuration Essentials Lab

Objective: Using the network configured in Module 6, to introduce more BGP Configuration Essentials for application in ISP networks

Prerequisite: Modules 6, 7 and 8.

Topology :

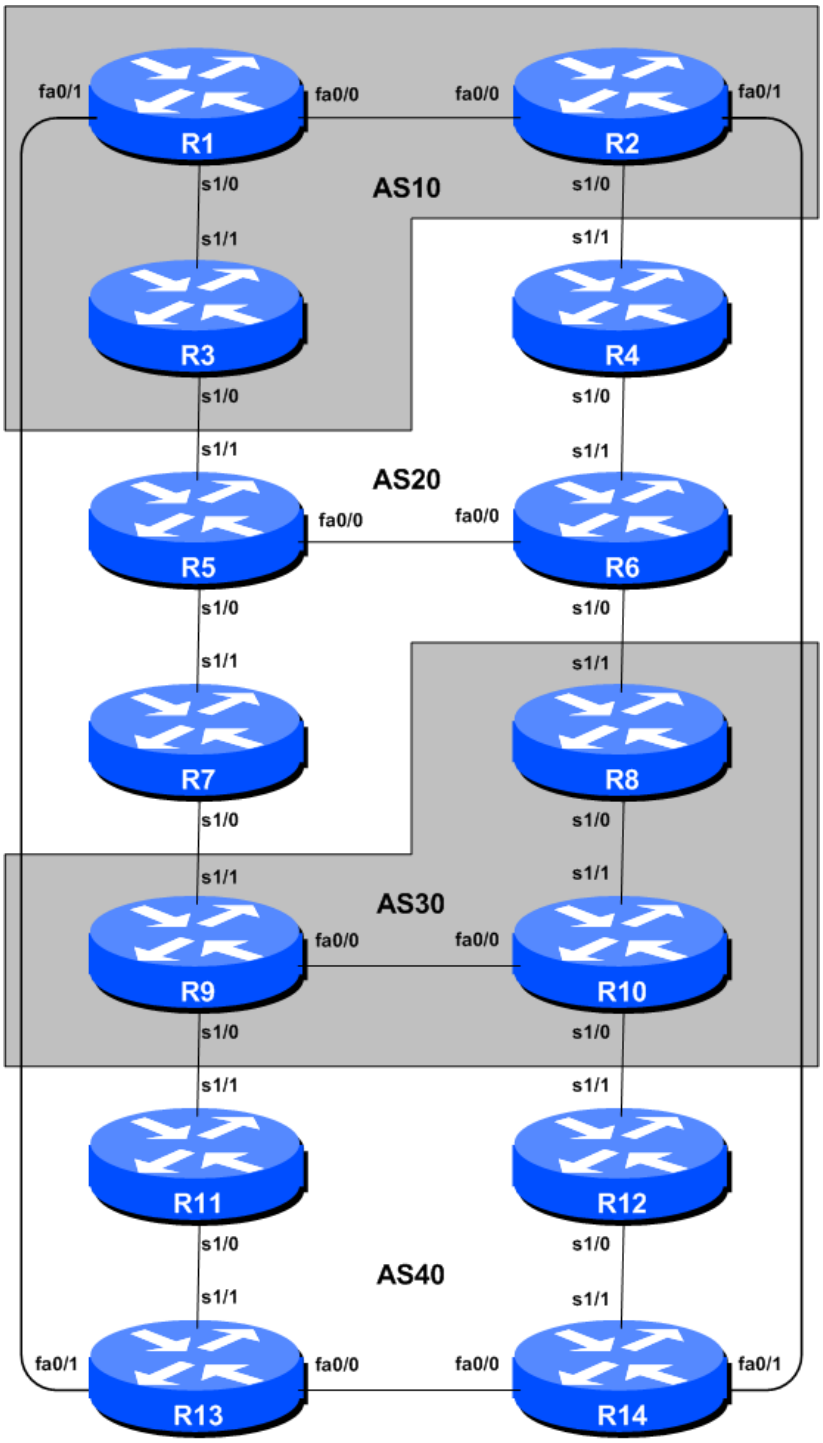

Figure 1 – BGP AS Numbers

Lab Notes

This module demonstrates some of the more recent features added to BGP in Cisco IOS. These are now considered good practice by many of the ISPs in the Internet industry, and are introduced here to give workshop participants hands on experience of these features prior to deployment on their own networks.

Before starting this module, retain the topology and router configurations as used at the start of Module 8. This requires the removal of **all** the filtering, route-maps, and community configurations examined in Modules 7 and 8.

Recommendation: Remember, if any configuration on a router is not in use, **it should be removed**. Surplus configuration usually gives rise to delayed error detection and debugging of configurations in cases of routing problems or other network failures.

The links shown in Figure 1 represents connectivity between AS's. It is assumed that all the routers within an AS connect to each other.

Lab Exercises

1. **Restore Configuration.** Restore the lab configuration to what it was at the start of Module 8/end of Module 7. The lab should be made up of 4 ASes, each AS is running either OSPF or ISIS internally, iBGP overlaid on that, and eBGP between the AS neighbours. Good configuration practices learned in Module 7 should be retained (for example, use peer-groups for iBGP, soft-reconfiguration if you think it will help you, etc).

2. **Path MTU discovery.** The default MTU for all communications originating from the router is 512 bytes – while this may be sufficient for most light use purposes, ISP networks tend to place larger stresses on routers. Enabling path MTU discovery on the router will ensure that the router will use the optimum (i.e. largest) MTU possible for a communication. For example, a router with several BGP neighbours and exchanging the full Internet routing table with each neighbour will be able to transfer this routing table almost 3 times faster over Ethernet or serial connections with path MTU discovery enabled (allowing 1500 byte packets) than with using the default MTU of 576 bytes.

   Enable path MTU discovery on your router:

   ```
   Router(config)#ip tcp path-mtu-discovery
   Router(config)#
   ```

   While there may not be much visible difference in router performance in the workshop lab, participants are encouraged to add this command to their default router configuration.

   Once this configuration has been entered, each BGP peering should be reset, and the new MTU should now be displayed in the results of `show ip bgp neighbor`.

3. **IP unreachables.** When implementing BGP in an ISP network, the classic and recommended way of inserting a prefix into the BGP table is by configuring a network statement in BGP and a matching static route to the Null0 interface (the so-called pull-up route). We saw this used in Modules 1 and 6.

The benefit delivered to the ISP network by using this method is that any traffic destined for any IP address covered by that address block will have a final destination, regardless as to whether the IP address is routed on the network or not. For example, if a customer is using a /25 address range out of the ISPs /20 address block, and that customer disconnects from the Internet to allow maintenance on their connection, traffic trying to reach the /25 address block will be "caught" by the aggregate's null route. This means the traffic doesn't traverse the ISP's backbone before dying on the aggregation router, but is caught "early" on as it enters the backbone. This is operationally tidier for many ISPs, and can be less confusing for Internet users as well.

(The static route to Null0 has many uses, and is one of the tools used frequently in helping with defeating denial of service attacks on service provider and end user networks.)

However, the side effect of doing this is that the router has to send a response that the packet has reached a destination – this response is that the destination is "unreachable". Each packet generates one response – an ICMP unreachable message. For a stream of packets, this can introduce some burden on the router CPU, so many ISPs configure the Null0 interface to not send ICMP unreachables – the packets end up at the Null0 interface and are silently discarded. This is much lighter on the router CPU.

First ask your neighbour to ping any IP address in your address block – the router should respond with a series of "UUUUU", indicating a host unreachable.

Now disable the sending of ICMP unreachables on your router's Null0 interface:

```
interface Null 0
 no ip unreachables
```

If you now ask your neighbour repeat the ping – there should be no response at all, the router simply displaying a series of "…..", indicating no response what so ever.

Try this with your neighbour's address block too, before and after they have disabled the ICMP unreachable processing on their router.

4. **OSPF & ISIS timers.** Many ISPs no longer retain the standard IGP timers that have been the defaults in IOS for many years. Technology has moved forwards rapidly, and what were considered acceptable defaults in the early nineties given the processor technology then are now rather out of date.

OSPF Lab: The IOS default for OSPF is for hellos to be sent every 10 seconds, with the deadtimer set for 40 seconds (in other words, the router will have had to miss 4 hellos/keepalives before it considers the neighbour has disappeared). Each router team should now reduce the OSPF timers from the defaults to new values of 3 seconds for hellos, 15 seconds for the deadtimer. This configuration has to be applied to each interface where an OSPF adjacency is expected. For example:

```
interface serial 0/0
 ip ospf hello-interval 3
 ip ospf dead-interval 15
```

ISIS Lab: The IOS default for ISIS is for hellos to be sent every 10 seconds, with the deadtimer set for 30 seconds (in other words, the router will have had to miss 3 hellos/keepalives before it considers the neighbour has disappeared). Each router team should now reduce the ISIS timers from the defaults to new values of 3 seconds for hellos, 15 seconds for the deadtimer (so making the hello-multiplier the value 5). This configuration has to be applied to each interface where an ISIS adjacency is expected. For example:

```
interface serial 0/0
 isis hello-interval 3
 isis hello-multiplier 5
```

5. **BGP timers.** For the reasons mentioned in the previous step, many ISPs also no longer retain the standard IOS BGP timers for their iBGP and eBGP. IOS defaults are 60 second keepalives, and 180 second dead time (in other words, if the router misses 3 keepalives it tears the session down). For this lab, each time will reduce the BGP timers to 15 second keepalives and 45 second deadtime.

   ```
   router bgp 1
    timers bgp 15 45
   ```

   To make the peering use these new timer values, restart the BGP sessions on the router. The two peers will negotiate timer values (usually the lowest values will win) to be used for the BGP session. Check with a `show ip bgp neighbor` command.

   What values an ISP should use in their own network really come down to experience as well as a bit of experimentation. While the IOS defaults will work perfectly well, significant gains in performance and routing failover can be achieved by reducing the timers. Review the BGP Best Practices presentation for more advice.

6. **Deterministic MED.** Another essential BGP configuration considered industry best practice is the use of deterministic MED. Normally if a BGP speaker hears an external prefix from several different connections to the same ASN, path selection will be carried out according to the order that the prefixes are heard in (oldest through to the most recent). For an ISP with multiple connections to another ASN, this usually proves to be very annoying, as the best path will be different each time a link is reset. To fix this, we enable deterministic MED, and that ensures that the prefixes are sorted in a predefined order (according to AS-PATH) and will give us the same best-path selection result each time. To activate deterministic MED, each router team should now configure the following global BGP command:

   ```
   router bgp 1
    bgp deterministic-med
   ```

7. **Shutting down BGP peerings.** A useful feature is the ability to shut down eBGP peerings – most BGP implementations have this functionality now, and IOS has had it for a few years (even though many operators don't realise so). If a peering needs to be temporarily disabled, either to help with troubleshooting, or to suspend a peering, etc, then the shutdown option is substantially better to use than the old method of simply removing the configuration altogether. The syntax is very straight forward – `neighbor x.x.x.x shutdown`. To reactivate the neighbour again, the reverse is used – `no neighbor x.x.x.x shutdown`. Be careful with the reactivation command – forgetting the shutdown option will result in the entire configuration for this neighbour peering being removed.

For this step, all router teams are going to shutdown the eBGP peerings they have configured on their routers. Routers 2 and 9 have two eBGP peerings, the rest have just one. An example configuration might be:

```
router bgp 20
 neighbor 10.30.15.1 remote-as 30
 neighbor 10.30.15.1 shutdown
!
```

which shuts down the BGP peering between the local router and its 10.30.15.1 neighbour.

***Checkpoint #1**: Call the lab instructors and let them know that you have completed the module up to this point. Once the lab instructors have explained the eBGP multihop section coming next, you will be asked to carry on with this Module.*

STOP AND WAIT HERE

8. **Using eBGP multihop.** Some ISPs use a concept called eBGP-multihop for their eBGP peering sessions. While this practise is strongly discouraged unless circumstances dictate otherwise, it is worth looking at the configuration so that the concepts can be understood.

   eBGP multihop basically means that the eBGP router disables the check that the eBGP neighbour is accessible on a directly connected interface of the local router. This allows the service provider to establish an eBGP session between the local and a distant router in the neighbouring AS. Some ISPs do this as they have a policy of only running BGP in their core network devices, and don't or won't support BGP on their aggregation/edge devices.

   This step will reconfigure the lab eBGP sessions to use eBGP multihop instead of direct neighbour peering. The IOS configuration command is `neighbor x.x.x.x ebgp-multihop N` where N represents the number of TCP hops the neighbour x.x.x.x is away from the local router. Some ISPs make N equal to 255 – which means the neighbour can be up to 255 hops away. Other ISPs actually specify how many hops away the peer is. The former is easy to manage – you don't need to worry about internal topology changes of your neighbouring ISP – but it can be highly risky too, as it is possible that the peering path between your router and your ebgp-multihop neighbour might change to one that you do not expect[1] (or want).

   A more complete configuration example would be as follows. The static route for the 1.2.3.4/32 destination is required – BGP cannot use a next-hop which has been learned by BGP (and as ISPs don't run IGPs between their networks, a static route is what is required).

   ```
   router bgp 55
    neighbor 1.2.3.4 remote-as 56
    neighbor 1.2.3.4 ebgp-multihop 5
   !
   ip route 1.2.3.4 255.255.255.255 serial 0
   !
   ```

[1] It is for this reason that many ISPs simply won't permit or support ebgp-multihop configuration on their networks.

Finally, ebgp-multihop is usually run between loopback interfaces on peering routers. It doesn't have to be, but it by and large is, and for the same reasons that iBGP is run between loopback interfaces. Loopbacks rarely change, whereas physical interface addresses can quite often do so.

Now convert all the eBGP sessions so that they use ebgp-multihop between the loopback interfaces of the peering routers. Use the correct number for the TCP hop count. And remember the static route to point to the remote router address. An example configuration might be:

```
router bgp 20
 neighbor 10.30.15.224 remote-as 30
 neighbor 10.30.15.224 ebgp-multihop 2
!
ip route 10.30.15.224 255.255.255.255 serial 0/0
!
```

Check that the eBGP session comes up – if it doesn't check with your neighbour that they have also completed this step. The prefixes advertised by your neighbouring AS will have next hop of the peer address (as before).

Note that the configuration for the neighbour 10.30.15.1 should be simply shut down (as it was in the previous step!) because we will be removing the ebgp-multihop configuration and reverting to the direct eBGP configuration at the end of this exercise.

Checkpoint #2*: Call the lab instructors and let them know that you have completed the module up to this point. Once the lab instructors have demonstrated the eBGP multihop configurations to the rest of the class, you will be asked to carry on with this Module.*

STOP AND WAIT HERE

9. **Remove ebgp-multihop configuration.** The ebgp-multihop configuration added in the previous steps should now be removed. And the prior directly connected point-to-point configuration should be re-enabled (in other words, do a `no neighbor x.x.x.x shutdown` on the affected eBGP peering). So something like the command sequence following:

```
Router1#conf t
Router1(config)#router bgp 10
!
! First reactivate directly connected eBGP with Router 13
!
Router1(config-router)#no neighbor 10.10.15.14 shutdown
!
! Now remove Router 13 eBGP peering which used ebgp-multihop
!
Router1(config-router)#no neighbor 10.40.15.226
!
Router1(config-router)#end
Router1#
```

Once this is complete, there should be no more ebgp-multihop configuration in the network. The previous steps were there simply to show how it is configured. As mentioned earlier in the module, and in the BGP presentations, use of ebgp-multihop is strongly discouraged.

10. **Prefix Threshold Warning.** In the lab network we are dealing with only a limited number of prefixes – each router is originating one prefix into the BGP routing system. In the Internet, there are well over one hundred thousand prefixes, and only routers with large amounts of memory can handle such large numbers. It's for this reason that many ISPs implement a so-called cut-off system on their routers – if they receive more prefixes than they expect from an eBGP peer, then the router can first warn them and then tear down the BGP session if the warnings are ignored and the router is in danger of running out of memory. (A downed BGP session is more likely to be noticed more quickly than an oscillating BGP session caused by a router which has run out of memory. Similarly, a downed BGP session will have less severe impact than a BGP session which is receiving thousands of spurious prefixes and causing havoc in the ISP's routing system.)

    The IOS BGP subcommand to set this maximum prefix is this:

    ```
    neighbor x.x.x.x maximum-prefix n [threshold] [warning-only] [restart m]
    ```

    where `threshold` is the percentage of `maximum-prefix` when the router will start sending warning messages, `warning-only` stops the router from tearing the peering down when `max-prefix` is reached, and `restart` *m* is the time in minutes before the router will attempt to restart the eBGP session.

    For this step, the Router Teams will set a maximum prefix limit on all their eBGP sessions. There are 4 prefixes visible in the routing system, so it's suggested that a useful limit to use might be 10. The configuration to do so is similar to this:

    ```
    router bgp 20
     neighbor 10.30.15.1 remote-as 30
     neighbor 10.30.15.1 max-prefix 10
    !
    ```

    This means that the router will tear down the BGP session when the number of prefixes learned from Router8 exceeds 20. With existing IOS defaults, the router will warn by log messages that the limit is getting close when it has received 15 prefixes (or 75% of the threshold).

Checkpoint #3: *Call the lab instructors and let them know that you have completed the module up to this point. Once everyone has reached this stage, the lab instructors will introduce more prefixes to the BGP routing system, as described in the next step.*

11. **Testing max-prefix.** The lab instructors will now introduce more prefixes to the routing system in the workshop lab network. All Router Teams should watch for the warning log messages appearing in their router log files, for example like:

    ```
    %BGP-4-MAXPFX: No. of unicast prefix received from 10.10.15.14 reaches 8, max 10
    ```

    and when the actual max-prefix limit has been reached, messages such as:

    ```
    %BGP-3-MAXPFXEXCEED: No. of unicast prefix received from 10.10.15.14: 11 exceed limit 10
    ```

    and the resultant tear down of the BGP session. Router Teams should experiment with using the warning-only keyword, as well as the restart timer. The instructors will allow sufficient time for this, and will demonstrate each option before moving on with this Module.

Checkpoint #4*: Wait at this point until the Lab Instructors have completed demonstrating the maximum-prefix function available for the BGP session.*

STOP AND WAIT HERE

12. **Maximum-prefix summary.** Before moving onwards tidy up from the previous steps, but leave a usable and working maximum-prefix configuration in place. It is recommended that maximum-prefix should become part of the ISP's standard BGP configuration. All operators should know exactly how many prefixes they should expect from each eBGP neighbour and should set an appropriate limit.

13. **Limiting AS-path length.** As with limiting the number of prefixes learned from a peer, the router can also be configured to limit the length of AS-paths learned from peers. Configuring this limit means that the router will discard all prefixes with an AS-path length longer than the limit.

    The syntax for this command is as follows:

    ```
    bgp maxas-limit n
    ```

    where *n* is the maximum AS-path length permissible for any prefix. All Router Teams should configure this in their BGP configurations. It is suggested that *n* should be 5 for this workshop – this is two more than the longest AS-path length ever likely to be seen (there are 4 ASNs!).

14. **Effect of limiting AS-path length.** On completion of the previous step, the router teams should now artificially increase the length of AS-paths of prefixes announced in eBGP sessions to their neighbours. To do this, they should use a route-map with the `as-path prepend` subcommand on the eBGP peerings.

    Recall that the final exercise in Module 8 demonstrated rudimentary traffic engineering using the as-path prepend construct in a route-map. Consult your notes from Module 8 to construct a route-map to increase the as-path length of the prefix you originate into the routing system when you announce it to your neighbours.

    The following configuration snippet is an example of what is expected:

    ```
    router bgp 20
     neighbor 10.30.15.1 remote-as 30
     neighbor 10.30.15.1 route-map increase-as-out out
    !
    route-map increase-as-out permit 10
     match ip address prefix-list myaddress
     set as-path prepend 20 20 20 20 20
    route-map increase-as-out permit 20
    !
    ip prefix-list myaddress permit 10.20.0.0/20
    !
    ```

    The route-map only applies the as-path prepend of five times AS 20 to the prefix originated by the router above. Other prefixes learned by the router through its other BGP sessions are announced unaltered.

Q. What happens to the announced prefix? You may need to ask to see the console of your neighbour's router.

A. The prefix is discarded. And you should see log messages similar to:

```
%BGP-6-ASPATH: Long AS path 6435 145 6175 109 16713 3549 790 6667 received from
192.168.1.1: More than configured MAXAS-LIMIT
```

in the router logs, and that the prefix with this long path is not present in the BGP table. Try changing the maxas-path limit value and see what happens.

Checkpoint #5*: Wait at this point until the Lab Instructors have completed demonstrating the maxas-path function available for the BGP session.*

STOP AND WAIT HERE

15. **Maximum AS-path length summary.** Before moving onwards tidy up from the previous steps, removing the route-map which was added to artificially increase the as-path length of the originated prefixes, but leave a usable and working maximum-prefix configuration in place. It is recommended that maximum as-path length limit should become part of the ISP's standard BGP configuration. The Internet is only around 5 ASNs deep on average, and a maximum length seen over the last decade, including prepends (and excluding accidents!), has been 27 ASNs. Many ISPs set a limit around 25 or 30, and this avoids problems should the inevitable accidents happen.

16. **AS masquerading.** Another useful tool for ISPs, especially those on the acquisition trail, or indeed any service provider merging ASes into one network, is the `local-as` BGP subcommand. This allows a router in one AS to masquerade being in another AS. For example, if one service provider has bought another, it's very easy to migrate the network during maintenance periods from one ASN to another ASN. However, migrating customers and peers can only be done at times that those peers and customers agree upon – and not many will want to migrate their eBGP sessions when the service provider wants to. (From personal experience, the author doesn't know of many customers who would be happy to get up at 4am to reconfigure their eBGP session with their ISP!)

    The local-as BGP subcommand allows the service provider to migrate the ASN the router is using, but still present the original AS to his customers and peers routers. The syntax is as follows:

    ```
    neighbor x.x.x.x local-as N
    ```

    where *N* is the ASN that the service provider wants to masquerade as – i.e. the ASN that the router x.x.x.x is expecting to peer with.

    This section is going to introduce this option into the ebgp-sessions in the network. Rather than doing a wholesale reconfiguration of BGP on all the lab routers, we will introduce "ghost" ASNs between the existing ASNs, according to the following table:

First AS	Inserted AS	Second AS
10	100	20
20	200	30
30	300	40

40	400	10

This means that Router 2 and Router 3 should add in `local-as 100` in their eBGP with Router 4 and Router 5 respectively. Router 4 and Router 5 should replace their peering with AS 10 with a peering with AS 100. (And similar for other AS peerings – the first AS adds in the local-as option, the second AS changes the eBGP session from that of the First AS to that of the Inserted AS.) For example:

On Router 2:

```
router bgp 10
 neighbor 10.10.15.10 remote-as 20
 neighbor 10.10.15.10 local-as 100
!
```

On Router 4:

```
router bgp 20
 neighbor 10.10.15.9 remote-as 100
!
```

Once this work has been completed over the whole workshop network, the eBGP sessions will all be restored, and AS-paths will show the inserted AS.

Checkpoint #6: *Wait at this point until the Lab Instructors have completed demonstrating the local-as function available for the BGP session.*

STOP AND WAIT HERE

17. **local-as summary.** Before moving onwards tidy up from the previous steps, removing the local-as configuration which was added, and restore the original eBGP configuration between the 4 ASes. This feature only needs to be used in circumstances described earlier – but it does exist, and makes many service provide ASN transition scenarios much easier to plan and implement.

18. **Private ASes.** Private AS numbers are often used by service providers for multihoming their customers on to their backbones (for example, as in RFC2270), or for separating out parts of their internal network (e.g. for a lab or development network), or if they have deployed Confederations to scale their iBGP mesh. Private ASNs range from 64512 to 65534 – 1023 are available, more than enough for service provider networks today. As they are intended for private use, their assignment internal to an ISP network is proprietary to that ISP – there does not need to be any coordinated assignment as with the public ASNs.

    For this step, we are going to convert two of the ASes from public numbers to private numbers. AS 10 should change their BGP configuration so that they are in AS 64512 and AS 40 should change their BGP configuration so that they are in AS 65500. This will involve changing the entire BGP configuration, including the iBGP. The easiest way of doing this to make an electronic copy of the BGP configuration as it is now, replace the instances of AS 10 with AS 64512 and AS 40 with AS 65500 respectively, remove the existing BGP configuration from the routers in AS 10 and AS 40, and apply the new configuration. Note that the routers in AS 20 and AS 30 will have to change all instances of AS 10 and AS 40 to the new ASNs.

The eBGP peerings between Router 1 and Router 13, and Router 2 and Router 14 should be **shut down** for this step – the only transit path between AS 10/64512 and AS 40/65500 will be through AS 20 and AS 30.

For example, the main parts of the Router 3 BGP configuration changes from:

```
router bgp 10
 neighbor 10.10.15.224 remote-as 10
 neighbor 10.10.15.225 remote-as 10
 neighbor 10.20.15.1 remote-as 20
!
```

to:

```
router bgp 64512
 neighbor 10.10.15.224 remote-as 64512
 neighbor 10.10.15.225 remote-as 64512
 neighbor 10.20.15.1 remote-as 20
!
```

The routers in AS 20 and AS 30 which peer with the new private ASes have their configuration changed similar to the following example for Router 5:

```
router bgp 20
 neighbor 10.20.15.2 remote-as 64512
!
```

Once this step has been completed by all participants in the workshop, two ASNs in use will be private, and two will be public. The BGP table anywhere in the lab will show both private and public ASNs in the AS-paths.

19. Private ASNs and the public Internet. As mentioned in the previous step, private ASNs should not be used or announced on the Internet. To ensure that private ASNs are not leaked out to the Internet, configuration which strips private ASNs out of the AS-paths should be added to all eBGP sessions.

All eBGP sessions in the lab should now add configuration to remove these private ASNs. For example:

```
router bgp 20
 neighbor 10.30.15.1 remote-as 30
 neighbor 10.30.15.1 remove-private-AS
!
```

The `neighbor x.x.x.x remove-private-AS` line above will strip out any private ASNs in the announcements it makes to the eBGP neighbour. Once everyone in the workshop has completed this task, observe what has happened to the BGP table – note the absence of private ASNs in the BGP table now.

***<u>Checkpoint #7</u>**: Call the lab instructors and let them know that you have completed the module up to this point. Once the lab instructors have demonstrated the remove-private-AS configurations to the rest of the class, you will be asked to carry on with this Module.*

STOP AND WAIT HERE

20. **Private ASNs and the public Internet (part 2)**. The `remove-private-AS` neighbour subcommand will only remove the private ASNs in certain circumstances. Most especially, if there is a public ASN in the AS path, the private ASNs will not be removed, as this situation is considered as a configuration error. This feature sometimes confuses some operators, so it's worth demonstrating it in the confines of the workshop network.

    The Router Teams operating Routers 1, 2, 13 and 14 should now reactivate the eBGP peering between their routers. Note that the teams will need to change their peering AS numbers (unless this was already done as part of the step before last).

    Once the two BGP sessions are re-established, look at the AS-paths visible throughout the network. Note that in some cases, the private ASNs have reappeared – this is because some routers see the public ASN in amongst the private ASNs, so the `remove-private-AS` subcommand has no effect.

21. **Summary.** This module has introduced some of the more advanced BGP configuration options available for ISPs. While this may be an introductory Workshop module, these configuration options are vital fundamental building blocks of any ISP operation requiring to use BGP for their Internet connectivity.

Review Questions

1. How many private ASNs are there?

2. How do you ensure that private ASNs are always removed from the Internet Routing Table?