IPv6 Module 1 – ISIS and iBGP

Objective: Create a basic physical lab interconnection using IPv6 with one ISIS Area and one BGP AS number (optionally running on top of an existing IPv4 infrastructure).

Prerequisites: Knowledge of Cisco router CLI, previous hands on experience.

The following will be the common topology used for this supplement.

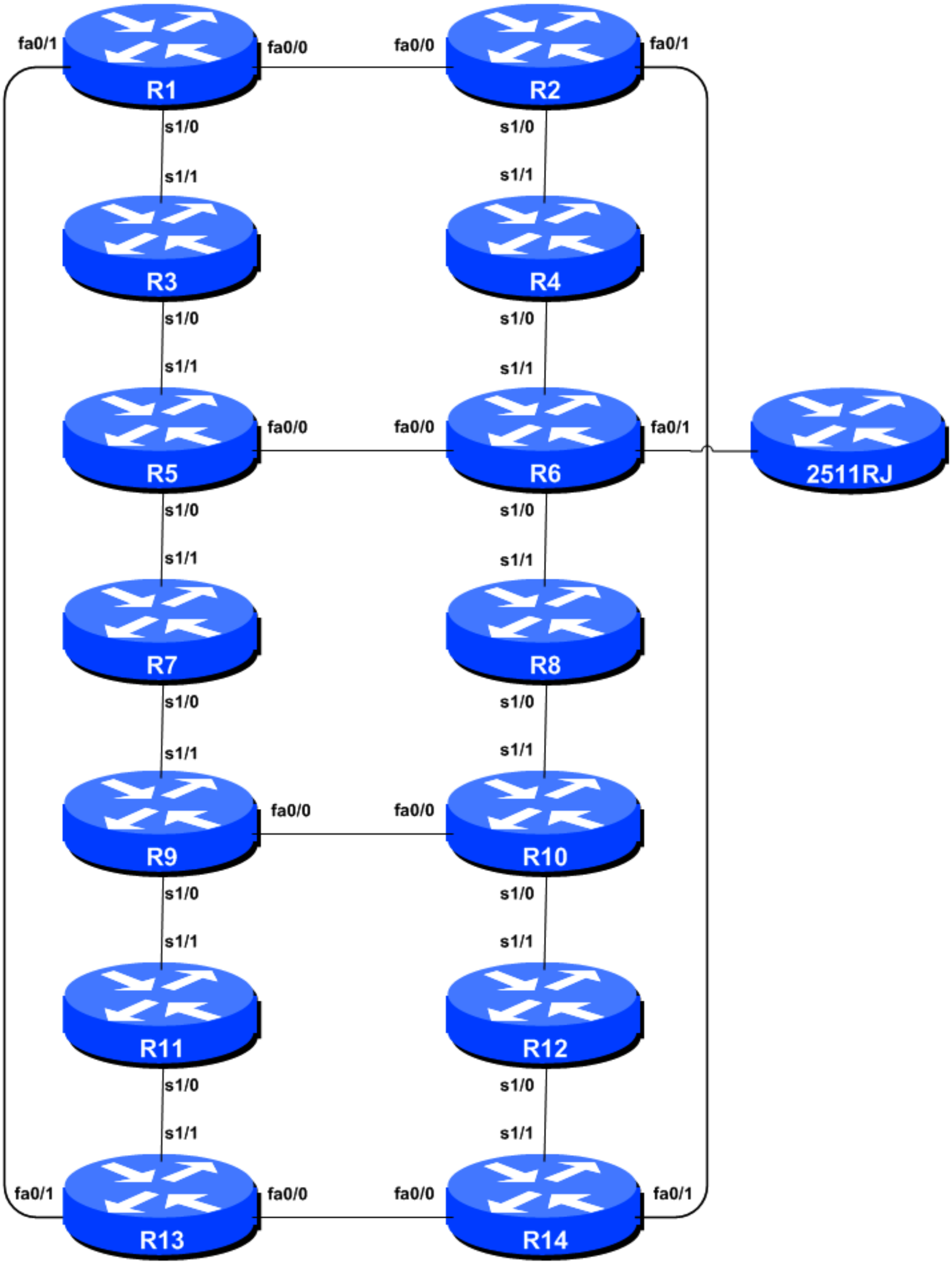


Figure 1 – ISP Lab Basic Configuration

Lab Notes

This Module is intended to supplement Module 1 of the IPv4 Workshop series once that has been completed. The topology and IPv4 configuration should be left exactly as it was at the end of Module 1.

If this lab is attempted without completing the IPv4 version, then the BGP router-id needs to be set before IPv6 BGP will work (ask the instructors how to do this). However this is not a recommended scenario as almost any service provider will plan to deploy IPv6 in parallel with an existing IPv4 infrastructure (this is called dual stack).

The routers used for this portion of the workshop must support IPv6. This is basically any IP Plus image from 12.2T onwards (IP Plus was renamed to Advanced IP Services for most platforms as from 12.3 mainline). As always, it is best to check the Cisco Feature Navigator www.cisco.com/go/fn to be absolutely sure which images set and platform supports IPv6. Unfortunately IPv6 is not part of the basic IP only or Service Provider IOS images used by most ISPs.

Note: these labs assume that the routers used are using a minimum of IOS 12.4 mainline. Syntax predating IOS 12.4 is discussed in the optional sections throughout the workshop.

Lab Exercise

1. **Enable IPv6.** Cisco routers with an IOS supporting IPv6 currently do not ship with IPv6 enabled by default. This needs to be done before any of the following exercises can be completed. To do this, use the following command:

   ```
   Router(config)# ipv6 unicast-routing
   ```

   The router is now configured to support IPv6 Unicast (as well as IPv4 Unicast which is the default). Save the configuration.

2. **Enable IPv6 CEF.** Unlike IPv4, CEFv6 is not enabled by default. So we now need to enable IPv6 CEF also, using the following command:

   ```
   Router(config)# ipv6 cef
   ```

   Nothing will break if IPv6 CEF is not enabled, but more advanced features such as NetFlow will not function without IPv6 CEF being enabled.

3. **Disable IPv6 Source Routing.** Unless you really believe there is a need for it, source routing should be disabled. This option, enabled by default, allows the router to process packets with source routing header options. This feature is a well-known security risk as it allows remote sites to send packets with different source address through the network (this was useful for troubleshooting networks from different locations on the Internet, but in recent years has been widely abused for miscreant activities on the Internet).

   ```
   Router1 (config)# no ipv6 source-route
   ```

4. **IPv6 Addressing Plans.** Addressing plans in IPv6 are somewhat different from what has been considered the norm for IPv4. The IPv4 system is based around the RIRs allocating address space to an LIR (an ISP who is a member of the RIR) based on the needs of the ISP; that allocation is intended to be sufficient for a year of operation without returning to the RIR. The ISP is expected to implement a similar process towards their customers – so assigning address space according to the needs of the customer.

   The system changes a little for IPv6. While the RIRs still allocate address space to their membership according to their membership needs, the justification to receive an IPv6 allocation is somewhat lighter than it is for IPv4. If the ISP can demonstrate a plan to connect at least 200 customers to the Internet using IPv6, they will receive an allocation. However, a bigger advantage starts with the customer assignments made by the ISP – the ISP simply has to assign a /48 to each of their customers. This is the minimum assignment for any site/customer – within this /48 there are a possible 64k subnets, deemed sufficient for all but the largest networks around these days. Within this /48, the smallest unit which can be assigned is a /64 – so every LAN and point to point link receives a /64.

   With this revised system, the address plan for IPv6 is hugely simplified. ISPs assign a single /48 for their network infrastructure, and the remainder of their /32 block is used for customer assignments. This workshop assumes this principle, as will be seen in the following steps.

5. **IPv6 Addresses.** As with the IPv4 portion of this Module we are going to introduce basic concepts of putting together a sensible IPv6 addressing plan for an ISP backbone. The RIRs are typically handing out IPv6 address space in /32 chunks – we assume for the purposes of this lab that our ISP has received a /32. Rather than using public address space, we are going to use 2001:db8::/32, the documentation address for IPv6. In the real world Internet, we would use public address space for our network infrastructure.

   The typical way that ISPs split up their allocated address space is to carve it into three pieces. One piece is used for assignments to customers, the second piece is used for infrastructure point-to-point links, and the final piece is used for loopback interface addresses for all their backbone routers. The schematic in
   Figure 2 shows what is typically done.

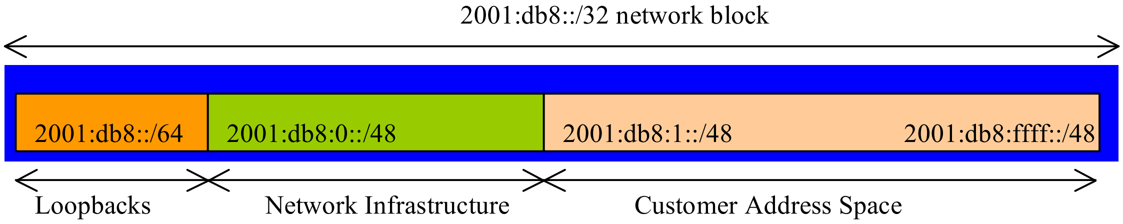
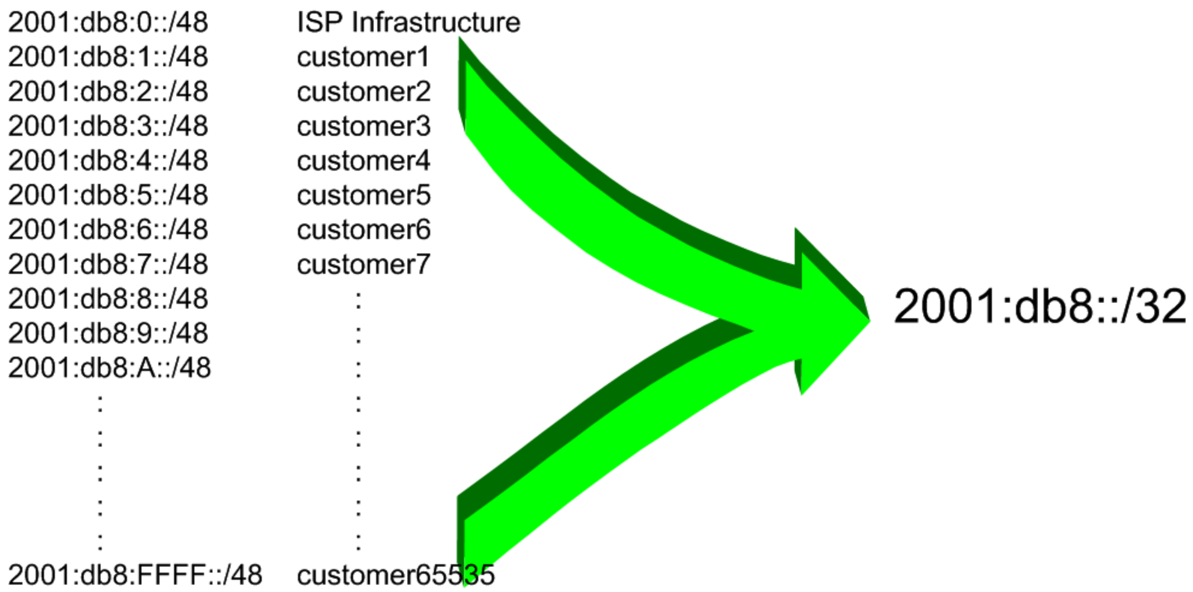


Figure 2 – Dividing allocated block of /32 into Customer, Infrastructure and Loopbacks

   Study the address plan which was handed out as an addendum to this workshop module. Notice how the infrastructure addressing uses the first /48 out of the /32 address block. Notice how we have set a side on /64 out of the infrastructure block for the router loopbacks. ISPs tend to document their addressing plans in flat text files or in spreadsheets – Figure 3 below shows an extract from a typical example (using our addressing scheme here).



2001:db8:0::/48	ISP Infrastructure
2001:db8:1::/48	customer1
2001:db8:2::/48	customer2
2001:db8:3::/48	customer3
2001:db8:4::/48	customer4
2001:db8:5::/48	customer5
2001:db8:6::/48	customer6
2001:db8:7::/48	customer7
2001:db8:8::/48	:
2001:db8:9::/48	:
2001:db8:A::/48	:
:	:
:	:
:	:
:	:
:	:
2001:db8:FFFF::/48	customer65535

2001:db8::/32

Figure 3 - Extract from an ISP Addressing Plan

6. **Back-to-Back Serial Connections.** Each team now needs to assign IPv6 addresses to the serial connections between the routers. See the addressing plan in the Appendices for the recommended addressing plan.

   Note: this lab will **not** use EUI-64 interface addressing, but instead will assign absolute addressing to each interface. The latter is much easier to manage, easier to handle for managing point-to-point peers and neighbour relationships.

   A sample configuration might look like:

   ```
   Router2(config)# interface serial 0/0
   Router2(config-if)# ipv6 address 2001:db8:0:6::1/64
   ```

   Q: What network mask should be used on all IPv6 enabled interfaces?
   A: The network mask should be /64. This is the subnet size used for all LANs, point-to-point links, and so on. While some providers would wish to subdivide the /64 even further, this is counter to RFC4291 which specifies the IPv6 addressing architecture. So all point-to-point links use a /64, all LANs use a /64, etc.

   Note: As discussed in the IPv6 presentation, ISPs are also using /126 and /120 as the subnet mask for point-to-point link addresses. We could have done this in the workshop as well, meaning that only a single /64 would have been required for our infrastructure. However, using /126 or /120 could mean potential problems in the future where developments in the IPv6 standard call on special uses for some of the bits between /65 and /128.

6. **Ethernet Connections.** As for the previous step, assign IPv6 addresses to the Ethernet point-to-point connections. Note that for IPv4 we were prudent in our assignments. For IPv6 the defined LAN subnet size is a /64. No larger, no smaller.

7. **Ping Test #1.** Ping all physically connected subnets of the neighbouring routers. If the physically connected subnets are unreachable, consult with your neighbouring teams as to what might be

wrong. Don’t ignore the problem – it may not go away. Use the following commands to troubleshoot the connection:

```
show ipv6 neighbors                           : Shows the ipv6 neighbour cache
show ipv6 interface <interface> <number>      : Interface status and configuration
show ipv6 interface                           : Summary of IP interface status and configuration
```

8. **Assign IPv6 Addresses to Loopback Interfaces.** While there is no need for a Loopback interface in this lab yet, it is still useful to configure an IPv6 address for it at this time. The loopback will be used for the iBGP peering later on in this lab. Note that OSPF and BGP router IDs are 32 bit integers and in IOS these are derived from the IPv4 address assigned to the Loopback interface (this has potential issues on network devices with no IPv4 address configured).

   Q. Why do you think the lack of any IPv4 address on the router would problem? Ask the lab instructors to discuss.

   As the minimum subnet size possible for IPv6 is a /64, we will assign the first /64 out of our /48 infrastructure block to be used for loopbacks – so we will use 2001:db8:0:0/64 for all the loopbacks. We have 14 routers in our lab – the assigned loopback addresses are:

R1	**2001:db8::1/128**	**R8**	**2001:db8::8/128**
R2	**2001:db8::2/128**	**R9**	**2001:db8::9/128**
R3	**2001:db8::3/128**	**R10**	**2001:db8::a/128**
R4	**2001:db8::4/128**	**R11**	**2001:db8::b/128**
R5	**2001:db8::5/128**	**R12**	**2001:db8::c/128**
R6	**2001:db8::6/128**	**R13**	**2001:db8::d/128**
R7	**2001:db8::7/128**	**R14**	**2001:db8::e/128**

   For example, Router Team 1 would assign the following address and mask to the loopback on Router 1:

```
Router1(config)# interface loopback 0
Router1(config-if)# ipv6 address 2001:db8::1/128
```

   Q: Why do we use /128 masks for the loopback interface address?

   A: There is no physical network attached to the loopback so there can only be one device there. So we only need to assign a /128 mask – it is a waste of address space to use anything else.

9. **ISIS within the same AS.** Each router Team should enable ISIS for IPv6 on their router. As with the IPv4 lab, the ISIS ID should be *workshop* (see example). Again we use level-2 in one area (*49.0001*). The NET should be *49.0001.x.x.x.x.00*, where *x.x.x.x* represents the router loopback IPv4 address. For example, the loopback for Router1 should be *49.0001.0100.0001.5241.00.* Note that we do not set up CLNS adjacencies on the loopback interface so we mark this as a passive interface. You might have to activate ISIS on one interface first before you can mark any interface as passive.

```
Router1(config)# router isis workshop
Router1(config-router)# net 49.0001.0100.0001.5241.00
Router1(config-router)# is-type level-2-only
```

Hint: A nice trick for converting the loopback interface address into the NSAP address is to take the loopback address and put the missing leading zeroes in. For example, Router 5 loopback address is 10.0.15.245; this is rewritten to 010.000.015.245 putting in the missing zeroes. Then rather than having the dot after every third character, move it to be after every fourth character. So 010.000.015.245 becomes 0100.0001.5245.

10. **Setting Wide Metrics.** We also set the metric-style to wide. ISIS supports two types of metric, narrow (historic now) and wide. IOS still defaults to narrow metrics, so we need to enter explicit configuration to change this to wide.

```
Router1(config)# router isis workshop
Router1(config-router)# metric-style wide
```

11. **Activating ISIS on each interface.** Now that the ISIS process is configured, all connected point to point and shared ethernet interfaces need to be configured with ISIS. Otherwise, you may not be able to see network advertisements via ISIS from routers two or more hops away. The example for the Router Team 1 is:

```
Router1(config)# interface fastethernet 0/0
Router1(config-if)# ipv6 router isis workshop
!
Router1(config)# interface fastethernet 0/1
Router1(config-if)# ipv6 router isis workshop
!
Router1(config)# interface serial 0/1
Router1(config-if)# ipv6 router isis workshop
```

Note: the ISIS ID on the interfaces must be matched with the router's ISIS ID.

12. **ISIS Metrics.** Now each team needs to set the ISIS metric on each physical interface. The default ISIS metric for all interface types is 10. Unlike OSPF in IOS, ISIS has no automatic scheme to convert the interface bandwidth into a metric value. ISPs deploying ISIS have to come up with their own scheme (as in fact many ISPs using OSPF now also do)

In the lab we use metric 2 for the Ethernet interfaces and metric 20 for the Serial interfaces. For example:

```
Router1(config)# interface fastethernet 0/0
Router1(config-if)# isis ipv6 metric 2 level-2
!
Router1(config)# interface fastethernet 0/1
Router1(config-if)# isis ipv6 metric 2 level-2
!
Router1(config)# interface serial 0/1
Router1(config-if)# isis ipv6 metric 20 level-2
```

13. **Announcing the Loopback /128.** We do not need to set up ISIS adjacencies on the loopback interface as there are no neighbours there, so we mark it as passive:

```
Router1(config)# router isis workshop
Router1(config-router)# passive-interface Loopback0
```

Note that this will tell ISIS to install the loopback interface address in the ISIS RIB. We do NOT need to add an `ipv6 router isis` statement onto the loopback interface itself. This is different from the required OSPF configuration, and often catches many engineers out, especially those who are learning ISIS after gaining experience with OSPF.

14. **ISIS Adjacencies.** Enable logging of ISIS adjacency changes. This is so that a notification is generated every time the state of a CLNS neighbor changes, and is useful for debugging purposes:

```
Router1(config)# router isis workshop
Router1(config-router)# log-adjacency-changes
```

15. **Ping Test #2.** Ping all loopback interfaces in the classroom. This will ensure the ISIS IGP is connected End-to-End. If there are problems, use the following commands to help determine the problem:

`show ipv6 route`	: see if there is a route for the intended destination
`show clns neighbor`	: see a list of CLNS-IS neighbours that the router sees
`show clns interface`	: see if ISIS is configured and see the IS type
`show isis database`	: see ISIS link state database that the router has learned

Checkpoint #1: *call lab assistant to verify the connectivity. Save the configuration as it is on the router – use a separate worksheet, or the workspace at the end of this Module. You will require this configuration several times throughout the workshop.*

16. **Configuring iBGP Neighbours (Part 1).** All Routers are still in Autonomous System (AS) 10 for this first lab. The extension for BGP which allows it to support multiple protocols is known as an address family. IPv4 unicast is one of the many address families supported – it's the one we are most familiar with. IPv6 is another address family supported by the multiprotocol BGP, and we must configure the new IPv6 peerings to belong to the IPv6 address family. Before we actually configure each IPv6 iBGP peer, we need to disable the router's assumption that all BGP peers are IPv4 unicast. To do this, we use the command as in the example below:

```
Router4(config)# router bgp 10
Router4(config-router)# no bgp default ipv4-unicast
```

17. **Configuring iBGP Neighbours (Part 2).** Before we set up iBGP with our neighbours in our AS, we need to do some basic preparation on the router. The IOS defaults are not optimised for Service Provider networks, so before we bring up BGP sessions, we should set the defaults that we require.

    The default distance for eBGP is 20, the default distance for iBGP is 200, and the default distance for ISIS is 115. This means that there is a potential for a prefix learned by eBGP to override the identical prefix carried by OSPF. Recall from the Routing presentation that there is a distinct separation between BGP and ISIS processes – prefixes present in ISIS will never be found in BGP, and vice-versa. To protect against accidents[1], the eBGP distance is set to 200 also. The command to do this is the bgp distance subcommand, syntax is:

```
distance bgp <external-routes> <internal-routes> <local-routes>
```

[1] There have been several incidents in the past where denial of service attacks on ISP networks have been successful because ISPs have omitted basic routing protocol security. Setting the BGP distances to be greater than any IGP is one of the mitigation methods available.

Note: This should be included in all future BGP configurations in this workshop. For example, for Router4, the configuration might be:

```
Router4(config)# router bgp 10
Router4(config-router)# address-family ipv6
Router4(config-router-af)# distance bgp 200 200 200
```

18. **Configuring iBGP neighbours (Part 3).** Now we can configure the IPv6 iBGP neighbours, as in the example for Router 4 below. The BGP peering will be established using the loopback interface's IP address.

```
Router4(config)#router bgp 10
Router4(config-router)#address-family ipv6
Router4(config-router-af)#neighbor 2001:db8::1 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::1 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::1 description iBGP with Router1
Router4(config-router-af)#neighbor 2001:db8::1 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::2 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::2 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::2 description iBGP with Router2
Router4(config-router-af)#neighbor 2001:db8::2 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::3 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::3 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::3 description iBGP with Router3
Router4(config-router-af)#neighbor 2001:db8::3 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::5 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::5 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::5 description iBGP with Router5
Router4(config-router-af)#neighbor 2001:db8::5 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::6 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::6 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::6 description iBGP with Router6
Router4(config-router-af)#neighbor 2001:db8::6 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::7 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::7 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::7 description iBGP with Router7
Router4(config-router-af)#neighbor 2001:db8::7 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::8 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::8 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::8 description iBGP with Router8
Router4(config-router-af)#neighbor 2001:db8::8 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::9 remote-as 10
Router4(config-router-af)#neighbor 2001:db8::9 update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::9 description iBGP with Router9
Router4(config-router-af)#neighbor 2001:db8::9 activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::a remote-as 10
Router4(config-router-af)#neighbor 2001:db8::a update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::a description iBGP with Router10
Router4(config-router-af)#neighbor 2001:db8::a activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::b remote-as 10
```

```
Router4(config-router-af)#neighbor 2001:db8::b update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::b description iBGP with Router11
Router4(config-router-af)#neighbor 2001:db8::b activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::c remote-as 10
Router4(config-router-af)#neighbor 2001:db8::c update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::c description iBGP with Router12
Router4(config-router-af)#neighbor 2001:db8::c activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::d remote-as 10
Router4(config-router-af)#neighbor 2001:db8::d update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::d description iBGP with Router13
Router4(config-router-af)#neighbor 2001:db8::d activate
Router4(config-router-af)#
Router4(config-router-af)#neighbor 2001:db8::e remote-as 10
Router4(config-router-af)#neighbor 2001:db8::e update-source loopback 0
Router4(config-router-af)#neighbor 2001:db8::e description iBGP with Router14
Router4(config-router-af)#neighbor 2001:db8::e activate
```

Q. Why is *update-source loopback 0* necessary on iBGP?

Use *show bgp ipv6 unicast summary* to check the status of the IPv6 iBGP neighbour connections. If the iBGP session is not up and/or no updates are being sent, work with the Router Team for that neighbour connection to troubleshoot the problem.

---Do step 19 ONLY if there is NO IPv4 configured on the router---

19. **BGP Router ID.** If this module is being used with out any IPv4 configuration being present on the router, then each router team will need to set the router-id for the BGP process. In IOS, the router-id is taken from the IPv4 address of the loopback interface (if configured) otherwise the highest IPv4 address on the router. If there is no IPv4 address configured, the router will not be able to set the BGP router-id, so the BGP process will not function.

    If there is no IPv4 configured on the router, now set the BGP router-id. For this exercise, the value chosen should be 192.168.0.<router-number>.

```
Router4(config)#router bgp 10
Router4(config-rtr)#bgp router-id 192.168.0.4
...etc...
```

20. **Checking the Configuration.** Do a *show run | begin bgp* to check and see what the BGP configuration looks like. Notice how the router has separated the generic part of the BGP configuration from the specific IPv6 address family content. An example of the output follows (IPv4 configuration from the IPv4 module has been omitted):

```
Router4#sh run | b bgp
router bgp 10
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 2001:db8::1 remote-as 10
 neighbor 2001:db8::1 description iBGP with Router2
 neighbor 2001:db8::1 update-source Loopback0
…
 address-family ipv6
  neighbor 2001:db8::1 activate
```

```
   neighbor 2001:db8::2 activate
   neighbor 2001:db8::3 activate
…
 exit-address-family
!
```

21. **Sanity Check.** Remember to use the following commands to ensure you are getting the information you are suppose to be getting:

`show clns neighbor`	: see a list of CLNS-IS neighbours that the router sees
`show isis database`	: see ISIS link state database that the router has learned
`show bgp ipv6 unicast summary`	: see a list of BGP IPv6 **peers** that the router sees
`show bgp ipv6 unicast`	: see a list of BGP IPv6 **paths** that the router sees
`show ipv6 route`	: see all the IPv6 routes that the router has installed
`show isis ipv6 rib`	: see all the ISIS IPv6 routes

Q. Are there routes seen via ***show bgp ipv6 unicast***? If not, why not? Are there any routes tagged "B" when you do a ***show ipv6 route***?

22. **Add Networks via BGP.** Each Router Team will use BGP to advertise the address block assigned to them earlier in the Module. For example, Router Team 1 would add:

```
Router1 (config)# router bgp 10
Router1 (config-router)# address-family ipv6
Router1 (config-router-af)# network 2001:db8::/32
```

Use *show bgp ipv6 unicast* on neighbour's router to see if you are advertising your network via BGP.

Q. Does the network show up via BGP? If not, why?

Enter a static route for the CIDR block. For example, Router 1 would use:

```
Router1 (config)#ipv6 route 2001:db8::/32 Null0
```

Q. Does the network show up via a neighbour's BGP? Use the command *show bgp ipv6 unicast neighbor <neighbour's IP address> advertised-routes* to see what you are exporting to the other router. Physically go to one of your neighbour's routers and check their BGP Table. Explain what you see.

Q. Does the network appear in the router's forwarding table? Use the command *show ipv6 route* to check the local forwarding table. If not, why not?

23. **Add the following commands to BGP**:

```
Router1 (config)#router bgp 10
Router1 (config-router)# address-family ipv6
Router1 (config-router-af)# no synchronization
```

Q. Does the network appear in the router's forwarding table? Now use the command *show ipv6 route* to check the local forwarding table. What does the *no synchronisation* command do in BGP? How does it effect the router's forwarding table?

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

24. **Adding a "customer" route into BGP (Background & Example).** We are now going to add a "customer" route into BGP on each router. Now in the lab we don't have any "customers" as such connected to our routers, so we are going to simulate the connectivity by simply using a Null0 interface. **As an example**, in real life, the configuration to connect a customer would look something like this.

```
ipv6 route 2001:db8:10::/48 Serial 0/5/2 permanent
!
router bgp 64509
 address-family ipv6
  network 2001:db8:10::/48
!
```

This would add a static route pointing 2001:db8:10::/48 towards Serial 0/5/2 – the latter interface would be a fixed link connecting to the customer site. 2001:db8:10::/48 would be the address space that the ISP had assigned to the customer. The BGP network statement would then add the customer address block into the ISP's iBGP.

Note: the **`permanent`** keyword ensures that the static route is always in the routing table, even if the interface physically goes down. Many ISPs use this to ensure they don't have iBGP churn when their customer links go down.

25. **Adding a "customer" route into BGP.** The "customer" address space that each router team will introduce into the iBGP is listed below – we will each use a /48 as that is the minimum address space for end-sites.

R1	**2001:db8:1::/48**	**R8**	**2001:db8:8::/48**
R2	**2001:db8:2::/48**	**R9**	**2001:db8:9::/48**
R3	**2001:db8:3::/48**	**R10**	**2001:db8:a::/48**
R4	**2001:db8:4::/48**	**R11**	**2001:db8:b::/48**
R5	**2001:db8:5::/48**	**R12**	**2001:db8:c::/48**
R6	**2001:db8:6::/48**	**R13**	**2001:db8:d::/48**
R7	**2001:db8:7::/48**	**R14**	**2001:db8:e::/48**

Each team should now set up a static route pointing to the **NULL0** interface for the /48 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 Router11:

```
Router11 (config)# ipv6 route 2001:db8:b::/48 Null0
Router11 (config)# router bgp 10
Router11 (config-router)# address-family ipv6
Router11 (config-router-af)# network 2001:db8:b::/48
```

26. **Check the BGP table.** Are there routes seen via ***show bgp ipv6***? If not, why not? Once every team in the class has done their configuration, each team should see the aggregate as well as the fourteen /48s introduced in the previous step. If this is not happening, work with your neighbours to fix the problem.

Checkpoint #3 : *call the lab assistant to demonstrate the current BGP table.*

27. Traceroute to all routers. Once you can ping all the routers, try tracing routes to all the routers using *trace x:x* command. For example, Router Team 1 would type:

```
Router1# trace 2001:db8::c
```

to trace a route to Router R12. If the trace times out each hop due to unreachable destinations, it is possible to interrupt the *traceroute* using the Cisco break sequence CTRL-^.

Q. Why do some trace paths show multiple IP addresses per hop?

A. If there are more than one equal cost paths, ISIS will “load share” traffic between those paths.

```
Router1>trace 2001:db8::c

Type escape sequence to abort.
Tracing the route to 2001:db8::c

  1 2001:db8:0:3::2    4 msec
    2001:db8:0:2::2    0 msec
    2001:db8:0:3::2    0 msec
  2 2001:db8:0:f::2    4 msec
    2001:db8:0:8::2    4 msec
    2001:db8:0:f::2    0 msec
  3 2001:db8:0:13::1   4 msec *  4 msec
Router1>
```

28. Other Features in ISIS and BGP. Review the documentation or use command line help by typing *?* to see other *show* commands and other ISIS and BGP configuration features.

Review Questions

1. What IOS show command(s) will display the router’s IPv6 forwarding table?

2. What IOS show command(s) will display the router’s IPv6 ISIS database?

3. What IOS show command(s) will display the router’s IPv6 BGP route table?