

IPv6 Module 1c – iBGP

Objective: Create a basic physical lab interconnection using IPv6 with chosen IGP and one BGP AS number running on top of an existing IPv4 infrastructure.

Prerequisites: IPv6 Module 1a (OSPF) or IPv6 Module 1b (ISIS).

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

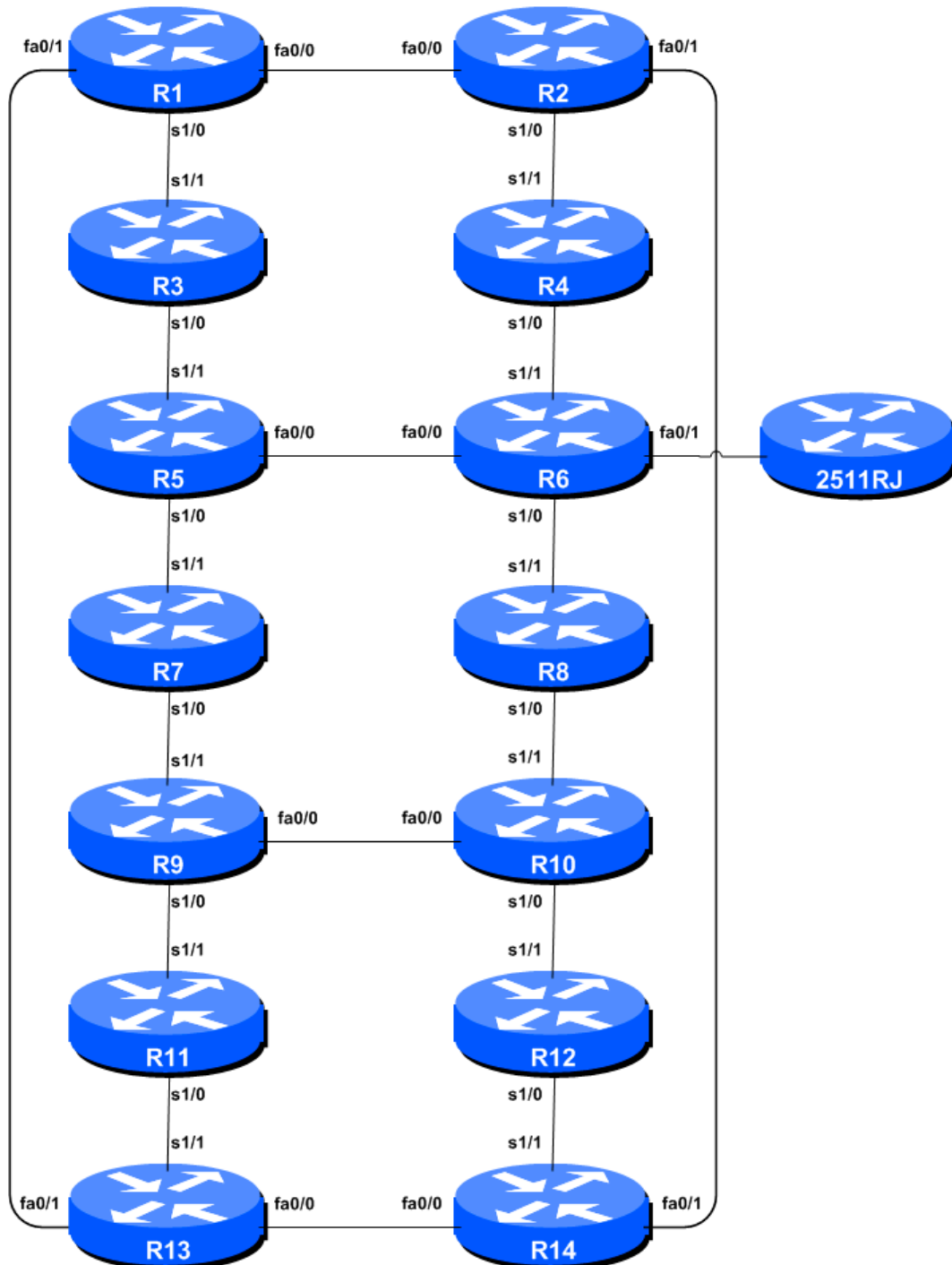


Figure 1 – ISP Lab Basic Configuration

Lab Notes

This Module is intended to provide the iBGP portion of Module 1 of the IPv6 version of this Workshop series. It should be completed once the IPv6 IGP has been deployed.

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. **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
```

2. **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 OSPF is 110. 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 OSPF processes – prefixes present in OSPF will never be found in BGP, and vice-versa. To protect against accidents¹, 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>
```

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
```

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.

3. 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.

- 4. 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
!
```

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

<code>show ipv6 ospf</code>	: see general IPv6 OSPF information
<code>show ipv6 ospf interface</code>	: see a list of IPv6 OSPF interfaces that the router sees
<code>show ipv6 ospf neighbor</code>	: see a list of IPv6 OSPF neighbours that the router sees
<code>show ipv6 ospf database</code>	: see IPv6 OSPF link state database the router has learned
<code>show bgp ipv6 unicast summary</code>	: see a list of BGP IPv6 peers that the router sees
<code>show bgp ipv6 unicast</code>	: see a list of BGP IPv6 paths that the router sees
<code>show ipv6 route</code>	: see all the IPv6 routes that the router has installed

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*?

- 6. 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?

7. 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 #1: call the lab assistant to verify the connectivity.

8. 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.

- 9. 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
```

- 10. 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 #2: call the lab assistant to demonstrate the current BGP table.

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

Review Questions

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