IPv6 Module 1a – OSPF

Objective: Create a basic physical lab interconnection using IPv6 with one OSPF Area running on top of an existing IPv4 infrastructure.

Prerequisites: IPv4 Lab Module 1, knowledge of Cisco router CLI, and previous hands on experience.

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

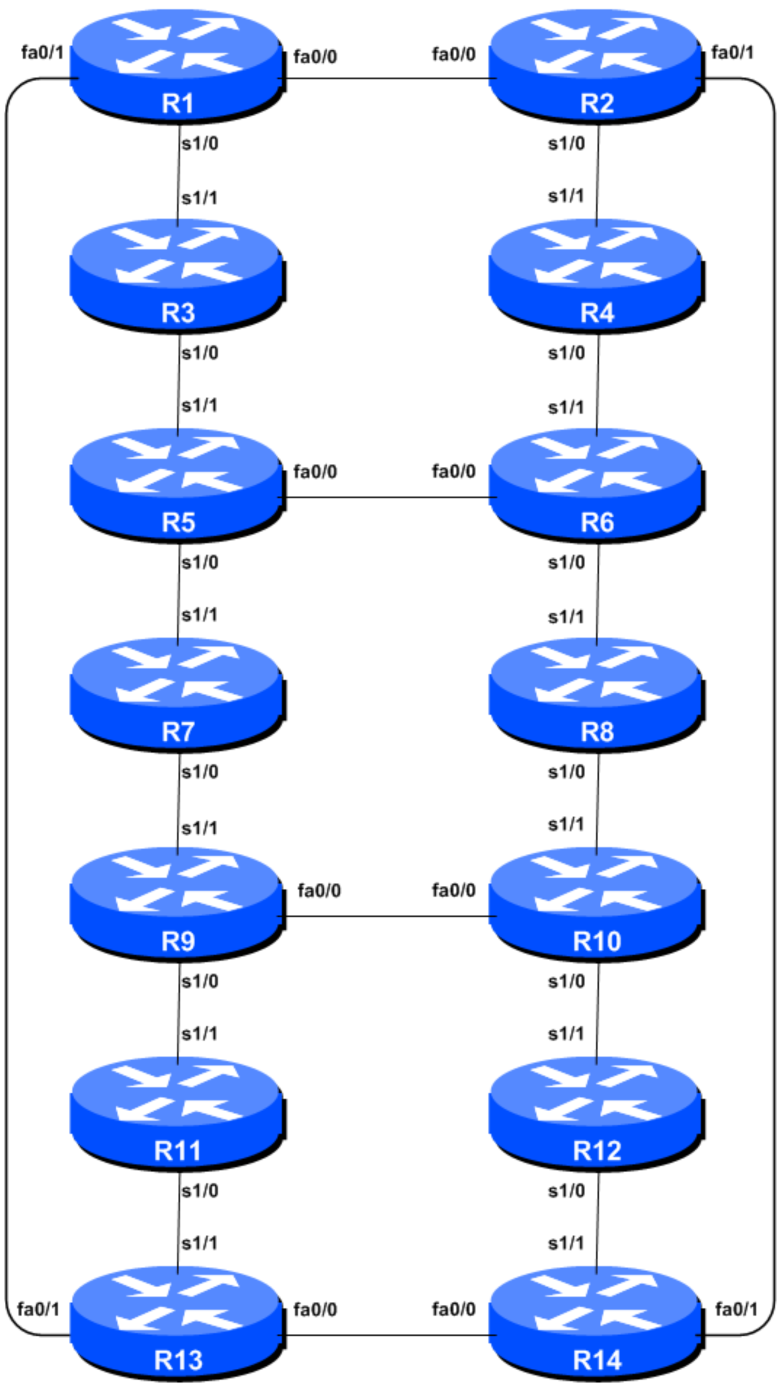

Figure 1 – ISP Lab Basic Configuration

Lab Notes

This Module is intended to supplement Module 1 of the IPv4 version of this 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.

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. 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 64k possible 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. **Note: This workshop will adopt the recommendations of RFC6164 and use a /127 mask for each point-to-point link – even though the link still has a /64 reserved for it.**

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.

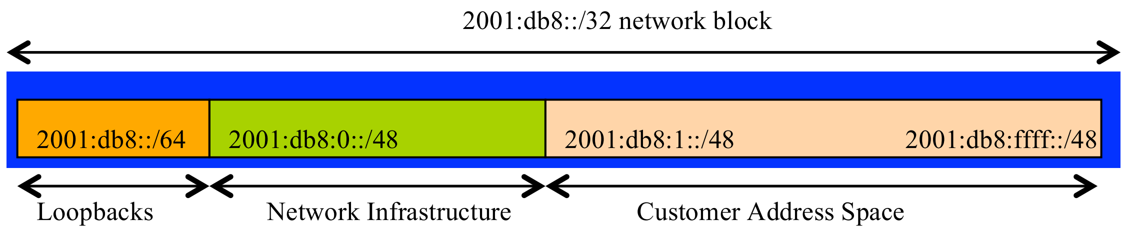

Figure 2 – Dividing allocated block of /32 into Customer, Infrastructure and Loopbacks

   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).

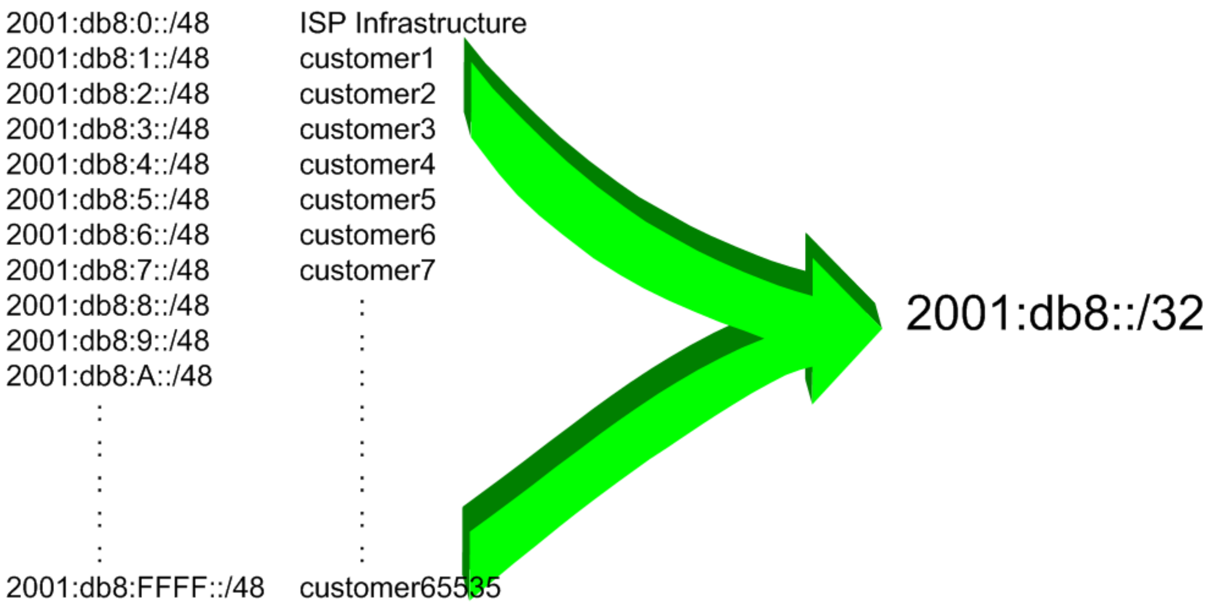
2001:db8:0::/48 ISP Infrastructure
2001:db8:1::/48 customer1
2001:db8:2::/48 customer2
2001:db8:3::/48 customer3
2001:db8:4::/48 customer4
2001:db8:5::/48 customer5
2001:db8:6::/48 customer6
2001:db8:7::/48 customer7
2001:db8:8::/48
2001:db8:9::/48
2001:db8:A::/48
2001:db8:FFFF::/48 customer65535
2001:db8::/32

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 1/0
   Router2(config-if)# ipv6 address 2001:db8:0:6::/127
   ```

   Q: What network mask should be used on all IPv6 enabled interfaces?
   A: The network mask should be /127. This is the subnet size used for all point-to-point links as recommended in RFC6164. We still reserve the entire /64 for this point-to-point link though, allowing simpler operational scalability should future changes be required.

   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, but chose to follow RFC6164 recommendations. We could also have numbered all our point-to-point links out of a single /64. However, this 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.

7. **Ethernet Connections.** As for the previous step, assign IPv6 addresses to the Ethernet point to point connections.

8. **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
```

9. **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.

Checkpoint #1: *call lab assistant to verify the connectivity. Demonstrate that you can ping and telnet to the adjacent routers.*

10. **OSPF within the same AS.** Each router Team should enable OSPF for IPv6 on their router. As with the IPv4 lab, the OSPF process identifier should be *41* (see example). (The OSPF process identifier is just a number to uniquely identify this OSPF process on this router. It is not passed between routers.) IPv6 OSPF is implemented slightly differently from the IPv4 counterpart in IOS – the latter made use of network statements to both find adjacencies and inject prefixes into the OSPF Link State Database. For IPv6, setting up the basic OSPF process is an independent configuration activity, such as in the example below. Note that all interfaces should be marked as passive by default:

```
Router1(config)#ipv6 router ospf 41
Router1(config-rtr)#passive-interface default
```

Once the OSPF process is running, the interfaces whose network link addresses are required in the OSPF Link State Database should be configured. This is done by going to the actual interface and attaching it to the OSPF process, as this example shows:

```
Router1(config)#interface loopback 0
Router1(config-if)#ipv6 ospf 41 area 0
!
Router1(config)#interface serial 1/0
Router1(config-if)#ipv6 ospf 41 area 0
!
...etc...
```

Finally, those interfaces over which we expect to form OSPF adjacencies should be marked as active interfaces. For this, we return to the main OSPF router process and mark those interfaces with the *no passive-interface* subcommand:

```
Router1(config)#ipv6 router ospf 41
Router1(config-rtr)#no passive-interface serial 1/0
Router1(config-rtr)#no passive-interface fastethernet 0/0
...etc...
```

Note that the loopback interface has no "network" attached to it, so there is no way it can be connected to another device in such a way as to form an OSPF adjacency.

11. **OSPF Adjacencies.** Enable logging of OSPF adjacency changes. This is so that a notification is generated every time the state of an OSPF neighbour changes, and is useful for debugging purposes:

```
Router2(config)#ipv6 router ospf 41
Router2(config-rtr)#log-adjacency-changes detail
```

12. **Avoiding Traffic Blackhole on Reboot[1].** When a router restarts after being taken out of service, OSPF will start distribute prefixes as soon as adjacencies are established with its neighbours. In the next part of the workshop lab, we will be introducing iBGP. So if a router restarts, OSPF will start up well before the iBGP mesh is re-established. This will result in the router landing in the transit path for traffic, with out the routing table being completed by BGP. There will not be complete routing information on the router, so any transit traffic (from customer to peer or upstream, or vice-versa) will be either dropped, or resulting in packets bouncing back and forth between adjacent routers. To avoid this problem, we require the router to not announce it is availability until the iBGP mesh is up and running. To do this, we have to provide the following command:

```
Router1(config)#ipv6 router ospf 41
Router1(config-router)#max-metric router-lsa on-startup wait-for-bgp
```

This sets up OSPF such that all IPv6 routes via this router will be marked as unreachable (very high metric) until iBGP is up and running. Once iBGP is running, the prefixes distributed by OSPF will revert to standard metric values, and the router will pass transit traffic as normal.

[1] This feature may not be available for OSPFv3 on all IOS releases.

13. **Ping Test #2.** Ping all loopback interfaces in the classroom. This will ensure the OSPF 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 ipv6 ospf`	: see general OSPF information
`show ipv6 ospf interface`	: Check if OSPF is enabled on all intended interfaces
`show ipv6 ospf neighbor`	: see a list of OSPF neighbours that the router sees

<u>Checkpoint #2:</u> *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.*

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

15. **Other Features in OSPF.** Review the documentation or use command line help by typing *?* to see other *show* commands and other OSPF 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 OSPF database?