Module 1a – OSPFv2

Objective: Create a basic physical lab interconnection with one OSPF Area. Ensure that all routers, interfaces, cables and connections are working properly.

Prerequisites: The setup section of Module 1.

The following will be the common topology used for the first series of labs.

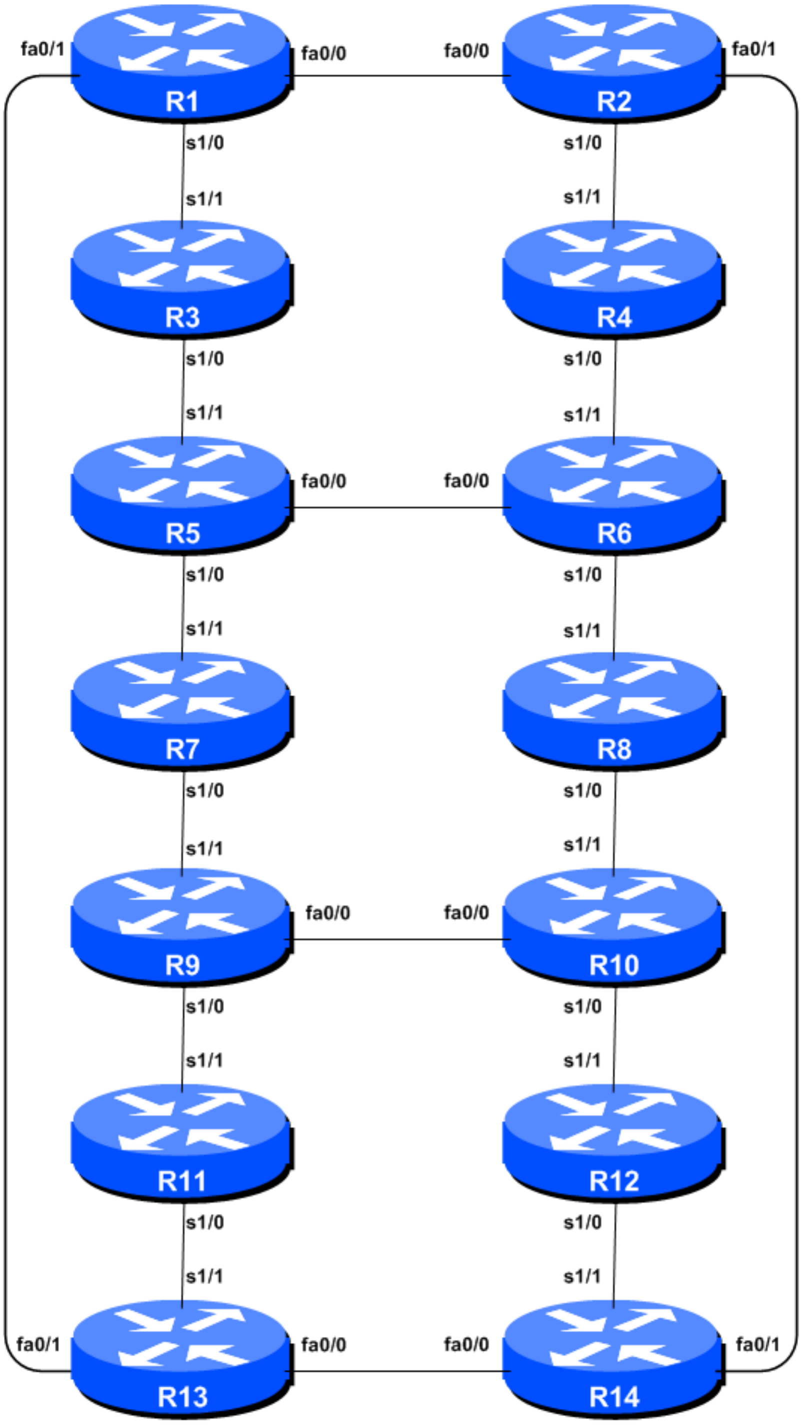

Figure 1 – ISP Lab Basic Configuration

Lab Notes

This lab continues from the previous one by adding IS-IS to the configured, addressed and confirmed working interfaces. Please refer to the Setup Module for further information and reference points about the purpose of Module 1.

Lab Exercise

1. **OSPF with one area in the same AS – activate the OSPF process.** Each router Team should enable OSPF on their router. 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.)

   ```
   Router1(config)#router ospf 41
   ```

   The default IOS configuration should be changed so that all interfaces are marked as passive for OSPF by default. This suppresses routing updates on all router interfaces and stops the router from unintentionally forming OSPF adjacencies over external facing interfaces, and the potential problems this may bring[1].

   ```
   Router1(config-router)#passive-interface default
   ```

   Any interfaces over which OSPF adjacencies should be formed need to be marked with the *no passive-interface* subcommand.

   ```
   Router1(config-router)#no passive-interface fastethernet 0/0
   Router1(config-router)#no passive-interface fastethernet 0/1
   Router1(config-router)#no passive-interface serial 1/0
   ```

2. **Activating OSPF on each interface.** Now that the OSPF process is configured, each team should activate OSPF on the individual router interfaces as required. Unlike previous releases of IOS, IOS 12.4 and later releases also allow OSPF to be run on a link (rather than just on a subnet). Rather than using the older (and confusing) "network" statement, we now activate OSPF on each interface that will form an adjacency:

   ```
   Router1(config)#interface serial 1/0
   Router1(config-if)#ip ospf 41 area 0
   !
   Router1(config-if)#interface fastethernet 0/0
   Router1(config-if)#ip ospf 41 area 0
   !
   Router1(config-if)#interface fastethernet 0/1
   Router1(config-if)#ip ospf 41 area 0
   ```

   .

3. **Announcing the Loopback /32.** The loopback interface also requires OSPF to be activated on it. Even though there is no adjacency to be formed (because there is no physical neighbour and the

[1] It is a common error in many ISP configurations to have the IGP active on all interfaces on the router. There have been many documented accidents where a customer IGP has established a connection with the ISP's IGP, resulting in a cross pollution of routing information, and the resulting traffic chaos. Switching off this ability by marking all interfaces passive by default helps avoid forgetfulness or errors at a later date.

interface is marked as passive by default in the previous step), we need to declare OSPF on the loopback interface so that the IP address used for the loopback is placed into the OSPF RIB.

```
Router1(config)#interface loopback 0
Router1(config-if)#ip ospf 41 area 0
```

4. **OSPF Adjacencies.** Each team should enable logging of OSPF adjacency changes. (**Note:** From IOS 12.4 onwards, *log-neighbor-changes* is activated by default when OSPF is first configured). 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)#router ospf 41
Router2(config-router)#log-adjacency-changes
```

5. **Avoiding Traffic Blackhole on Reboot.** 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)#router ospf 41
Router1(config-router)#max-metric router-lsa on-startup wait-for-bgp
```

This sets up OSPF such that all 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.

6. **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 ip route` : see if there is a route for the intended destination

`show ip ospf` : see general OSPF information

`show ip ospf interface` : Check if OSPF is enabled on all intended interface

`show ip ospf neighbor` : see a list of OSPF neighbours that the router sees

Checkpoint #2: *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.*

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

```
Router1# trace 10.0.15.252
```

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 router12

Type escape sequence to abort.
Tracing the route to router12.workshop.net (10.0.15.224)

  1 fe0-0.router2.workshop.net (10.0.15.2) 4 msec
    fe0-1.router13.workshop.net (10.0.15.6) 0 msec
    fe0-0.router2.workshop.net (10.0.15.2) 0 msec
  2 fe0-0.router14.workshop.net (10.0.15.54) 4 msec
    fe0-1.router14.workshop.net (10.0.15.26) 4 msec
    fe0-0.router14.workshop.net (10.0.15.54) 0 msec
  3 ser0-0.router12.workshop.net (10.0.15.69) 4 msec *  4 msec
Router1>
```

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

9. **Advanced Configuration.** Those router teams who have completed this module should refer to Module 11 of the Advanced BGP Workshop. The set-up steps have been extended to include all the basic requirements of a router being used in an ISP backbone. While waiting for the module to complete, now would be a good time to review the advanced Module and incorporate the additions to the configuration used here.

Review Questions

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

2. What IOS show command(s) will display the router's OSPF database?