Basic Topology and OSPF

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

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

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

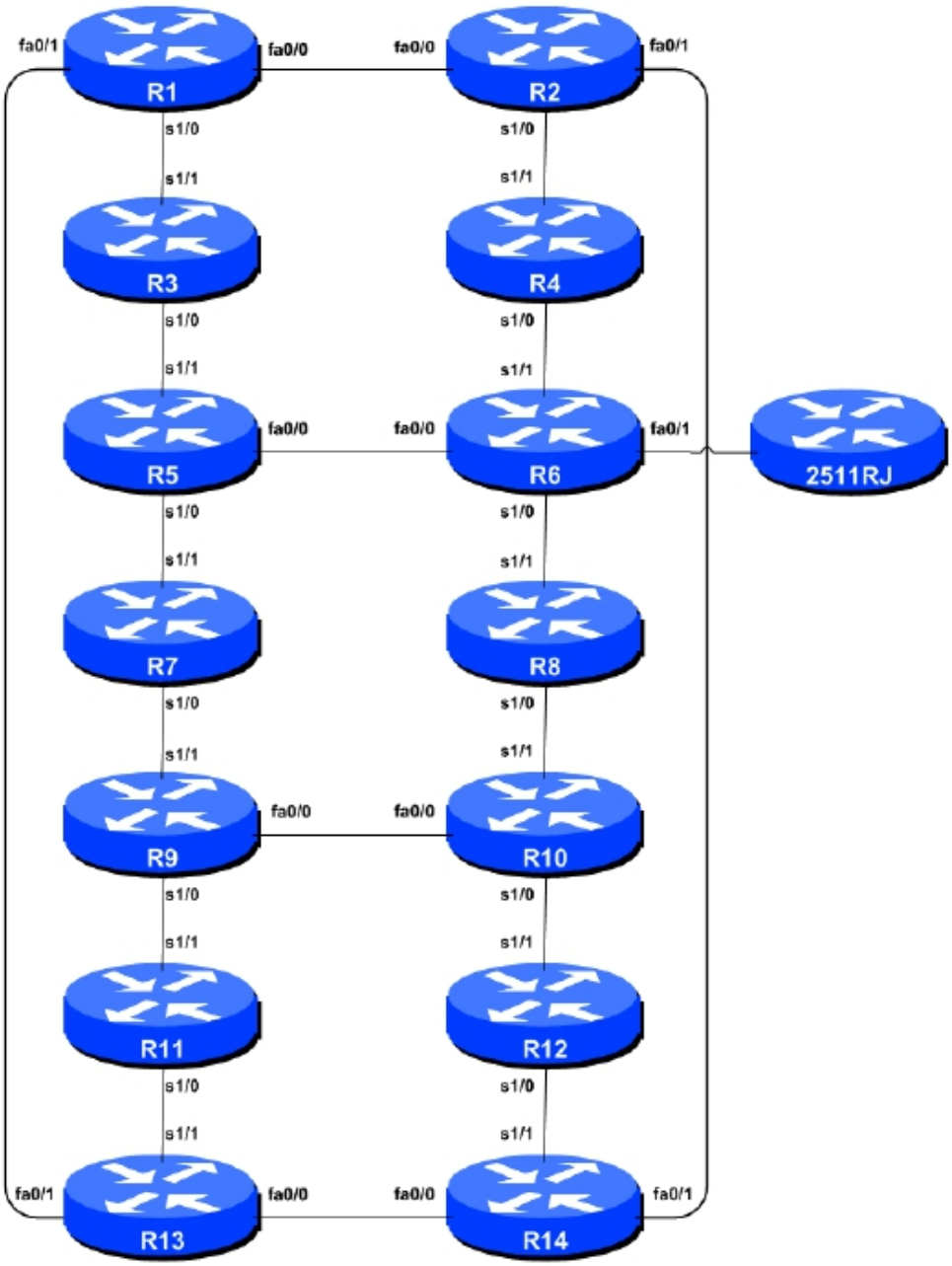

Figure 1 – ISP Lab Basic Configuration

Lab Notes

This workshop is intended to be run on a Dynamips server with the appropriate lab topologies set up. The routers in the Dynamips environment are using service provider IOS. The configurations and configuration principles discussed below will work on all Cisco IOS Release 12.4 onwards. Earlier Cisco IOS releases are not supported but will mostly work using the notes below; they will miss some of the features covered.

The purpose of this module is to construct the workshop lab and introduce everyone to the basic principles of constructing and configuring a network. An important point to remember, and one that will be emphasised time and again through out this workshop, is that there is a distinct sequence to building an operational network:

- After the **physical design** is established, the connections between the hardware should be built and verified.
- Next, the routers should have the **base configuration** installed, and basic but sufficient security should be set up.
- Next the **basic IP connectivity** be tested and proven. This means assigning IP addresses on all links which are to be used, and testing the links to the neighbouring devices.
- Only once one router can see its neighbour does it make sense to start configuring routing protocols. And **start with the IGP** (OSPF is chosen for this workshop). There is no purpose to building BGP while the chosen IGP (in this case OSPF) is not functioning properly. BGP relies on OSPF to find its neighbours and next hops, and an improperly or non-functioning OSPF will result in much time wasted attempting to debug routing problems.
- Once the IGP is functioning properly, the **BGP configuration** can be started, first internal BGP, then external BGP.
- **Remember to RTFM.** What is RTFM? It is critical that ISP Network Engineers fully utilise all information resources. The #1 source is the documentation. ***Read The F#$% Manual (RTFM)*** is the traditional phase used to inform engineers that the answer is in the documentation and go read it.
- Finally, **documentation**. Documentation is often overlooked or forgotten. It is an ongoing process in this workshop. If the instructor asks you to document something, either on the whiteboard in the class, or at the back of this booklet, it is in your best interests to do so. There can never be too much documentation, and documentation at the time of network design and construction can usually saves much frustration at a future date or event.

Lab Exercise

1. **Routers and the Workshops participants.** This workshop is laid out such that a group of two students will operate a single router. 14 routers generally imply at least 28 participants. For workshops with larger numbers of participants, groups of three should configure a single router. The Workshop Instructors will divide the routers amongst the workshop participants. In the following notes, a "router team" refers to the group assigned to one particular router.

2. **Router Hostname.** Each router will be named according to the table location, Router1, Router2, Router3, etc. Documentation and labs will also refer to *Router1* as R1. At the router prompt, first go into enable mode, then enter "config terminal", or simply "config" by itself:

```
Router> enable
Router# config terminal
Enter configuration commands, one per line. End with CNTL/Z.

Router(config)# hostname Router1
Router1(config)#
```

3. **Turn Off Domain Name Lookups.** Cisco routers will always try to look up the DNS for any name or address specified in the command line. You can see this when doing a *trace* on a router with no DNS server or a DNS server with no in-addr.arpa entries for the IP addresses. We will turn this lookup off for the labs for the time being to speed up traceroutes.

```
Router1 (config)# no ip domain-lookup
```

4. **Disable Command-line Name Resolution.** The router by default attempts to use the various transports it supports to resolve the commands entered into the command line during normal and configuration modes. If the commands entered are not part of Cisco IOS, the router will attempt to use its other supported transports to interpret the meaning of the name. For example, if the command entered is an IP address, the router will automatically try to connect to that remote destination. This feature is undesirable on an ISP router as it means that typographical errors can result in connections being attempted to remote systems, or time outs while the router tries to use the DNS to translate the name, and so on.

```
Router1 (config)# line con 0
Router1 (config-line)# transport preferred none
Router1 (config-line)# line vty 0 4
Router1 (config-line)# transport preferred none
```

5. **Disable 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 ip source-route
```

6. **Usernames and Passwords.** All router usernames should be ***isplab*** and all passwords should be ***lab-PW***. Please do **<u>not</u>** change the username or password to anything else, or leave the password

unconfigured (access to vty ports is not possible if no password is set). It is essential for a smooth operating lab that all participants have access to all routers.

```
Router1 (config)# username isplab secret lab-PW
Router1 (config)# enable secret lab-PW
Router1 (config)# service password-encryption
```

The *service password-encryption* directive tells the router to encrypt all passwords stored in the router's configuration (apart from *enable secret* which is already encrypted).

Note A: There is the temptation to simply have a username of *cisco* and password of *cisco* as a lazy solution to the username/password problem. Under no circumstances must any service provider operator ever use easily guessable passwords as these on their live operational network[1].

Note B: for IOS releases prior to 12.3, the username/secret pair is not available, and operators will have to configure username/password instead. The latter format uses type 7 encryption, whereas the former is the slightly more secure md5 based encryption. IOS 15.1 onwards uses SHA256 as a replacement for MD5.

7. **Enabling login access for other teams.** In order to let other teams telnet into your router in future modules of this workshop, you need to configure a password for all virtual terminal lines.

```
Router1 (config)# aaa new-model
Router1 (config)# aaa authentication login default local
Router1 (config)# aaa authentication enable default enable
```

This series of commands tells the router to look locally for standard user login (the username password pair set earlier), and to the locally configured enable secret for the enable login. By default, login will be enabled on all vtys for other teams to gain access.

8. **Configure system logging.** A vital part of any Internet operational system is to record logs. The router by default will display system logs on the router console. However, this is undesirable for Internet operational routers, as the console is a 9600 baud connection, and can place a high processor interrupt load at the time of busy traffic on the network. However, the router logs can also be recorded into a buffer on the router – this takes no interrupt load and it also enables to operator to check the history of what events happened on the router. In a future module, the lab will configuration the router to send the log messages to a SYSLOG server.

```
Router1 (config)# no logging console
Router1 (config)# logging buffer 8192 debug
```

which disables console logs and instead records all logs in a 8192byte buffer set aside on the router. To see the contents of this internal logging buffer at any time, the command "`sh log`" should be used at the command prompt.

9. **Save the Configuration.** With the basic configuration in place, save the configuration. To do this, exit from enable mode by typing "end" or "<ctrl> Z", and at the command prompt enter "write memory".

[1] This sentence cannot be emphasized enough. It is quite common for attackers to gain access to networks simply because operators have used familiar or easily guessed passwords.

```
Router1(config)#^Z
Router1#  write  memory
Building  configuration...
[OK]
Router1#
```

It is highly recommended that the configuration is saved quite frequently to NVRAM, especially in the workshop environment where it is possible for power cables to become dislodged. If the configuration is not saved to NVRAM, any changes made to the running configuration will be lost after a power cycle.

Log off the router by typing exit, and then log back in again. Notice how the login sequence has changed, prompting for a "username" and "password" from the user. Note that at each checkpoint in the workshop, you should save the configuration to memory – remember that powering the router off will result in it reverting to the last saved configuration in NVRAM.

10. IP Addresses. This Module will introduce the basic concepts of putting together a sensible addressing plan for an ISP backbone. We are building one autonomous system out of the 14 routers we have in the lab. The RIRs are typically handing out IPv4 address space in /20 chunks (depends on which RIR region) – we assume for the purposes of this lab that our ISP has received a /20. Rather than using public address space, we are going to use a portion of 10/8 (RFC1918 or private address space) for this lab. 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.

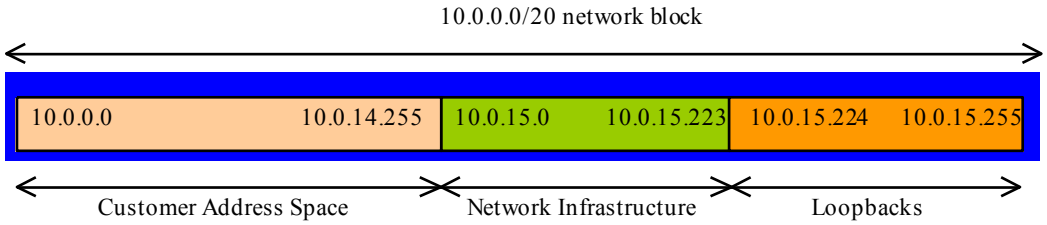

Figure 2 – Dividing allocated block of /20 into Customer, Infrastructure and Loopbacks

Study the address plan which was handed out as an addendum to this workshop module. Notice how the infrastructure addressing starts at 10.0.15.0 and carries on up to 10.0.15.70 – this leave us room to grow the network by more point-to-point links, up to 10.0.15.223 in fact. Notice how we have set a side just a single /27 for the router loopbacks – but we have only used the 14 addresses from 241 up to 254 for our network, leaving some spare for future growth (not that we have future growth planned for the workshop), an entirely realistic proposition for an ISP backbone. Indeed, 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).

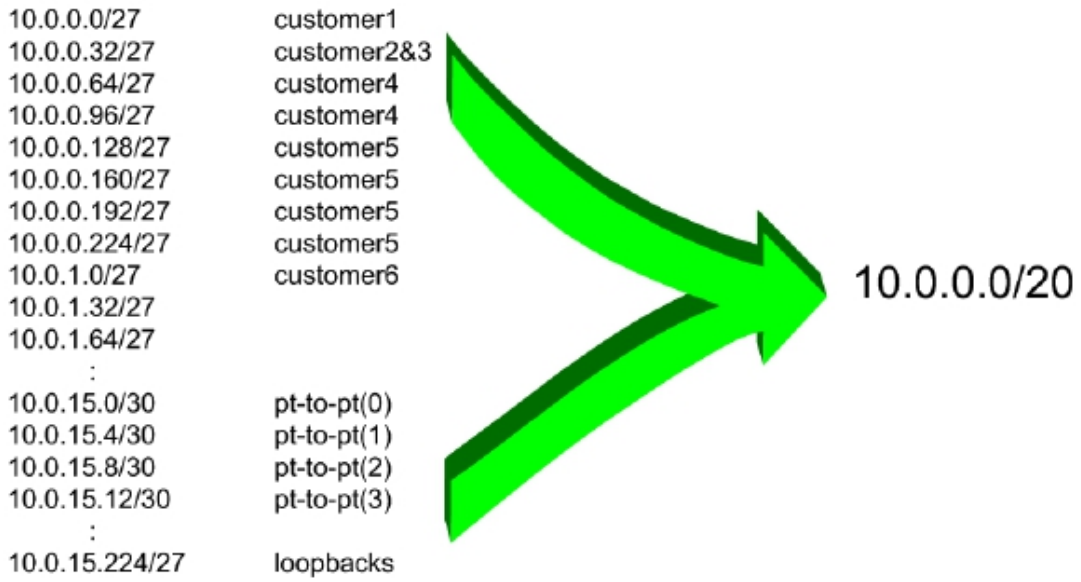

Figure 3 – Extract from an ISP addressing plan

11. **Back-to-Back Serial Connections.** Connect the serial connections as in Figure 1. The DCE side of a back-to-back serial connection is configured with the ***clock rate*** command that drives the serial circuit. (Older versions of IOS used the ***clockrate*** command, now hidden but still functional.) Physically check the cable to see which side is DCE and which is DTE. On some routers, the command ***show controller <interface>*** will show DCE/DTE status. For example, on a Cisco 3620 router, ***show controllers serial 0/0*** will produce a result that will display whether the cable connected to serial 0/0 is a DTE or DCE.

    Once the DTE and DCE cables have been determined and the ***clock rate*** command has been applied, configure the IP address (as per the addressing plan discussed earlier) and other recommended BCP commands that are recommended for each ISP's Interface:

```
Router2(config)#  interface  serial  1/0
Router2(config-if)#  ip  address  100.1.17.1  255.255.255.252
Router2(config-if)#  description  2 Mbps  Link  to  Router4  via  DTE/DCE  Serial
Router2(config-if)#  bandwidth  2000
Router2(config-if)#  clock  rate  2000000
Router2(config-if)#  no  ip  redirects
Router2(config-if)#  no  ip  directed-broadcast
Router2(config-if)#  no  ip  proxy-arp
Router2(config-if)#  no  shutdown
```

    NOTE: The lab instructors will have drawn a large network map on the white-board in the workshop lab. When the IP addresses are assigned, please annotate them and inform the instructor. All the point to point links **MUST** be annotated there so that other Router Teams can document and understand the links and routing in this and future modules.

    Q: What network mask should be used on point-to-point links?

    A: On serial interfaces, the network mask should be /30 (or 255.255.255.252 in dotted quad format). There is no point in using any other size of mask as there are only two hosts on such a

link. A 255.255.255.252 address mask means 4 available host addresses, of which two are usable (the other two representing network and broadcast addresses).

12. **Ethernet Connections.** The Ethernet links between the routers will be made using *cross-over* RJ-45 cables – these will directly connect the Ethernet ports on the two routers without the requirement for an Ethernet switch. IP subnets will again be taken from the Addressing Plan. Don't make the mistake of assigning a /24 mask to the interface address – there are only two hosts on the Ethernet connecting the two routers, so a /30 mask should be entirely sufficient.

13. **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 arp                                  : Shows the Address resolution protocol
show interface <interface> <number>       : Interface status and configuration
show ip interface                         : Brief summary of IP interface status and configuration
```

14. **Create Loopback Interfaces.** Loopback interfaces will be used in this workshop for many things. These include generating routes (to be advertised) and configuring some BGP peerings. As discussed earlier in Step 10, we will use part of the allocated IP address block for loopback interfaces. Most ISPs tend to set aside a contiguous block of addresses for use by their router loopbacks. For example, if an ISP had 20 routers, they would need a /27 (or 32 host addresses) to provide a loopback address for each router. We have 14 routers in our lab – to be prudent and allow for growth, we will set aside a /27 (allows us 32 loopbacks) but only use 14 of them. The assigned loopback addresses are:

R1	**10.0.15.241/32**	**R8**	**10.0.15.248/32**
R2	**10.0.15.242/32**	**R9**	**10.0.15.249/32**
R3	**10.0.15.243/32**	**R10**	**10.0.15.250/32**
R4	**10.0.15.244/32**	**R11**	**10.0.15.251/32**
R5	**10.0.15.245/32**	**R12**	**10.0.15.252/32**
R6	**10.0.15.246/32**	**R13**	**10.0.15.253/32**
R7	**10.0.15.247/32**	**R14**	**10.0.15.254/32**

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)#ip address 10.0.15.241 255.255.255.255
```

Q: Why do we use /32 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 /32 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.*

15. **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[2].

```
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
```

16. **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
```

.

17. **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 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
```

18. **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:

[2] It's 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.

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

19. **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.

20. **(Optional). Enable DNS name and address resolution on the routers.** If the workshop instructors have set up the nameserver in the workshop at this stage, all router teams should now enable DNS lookups on their routers. OSPF is carrying all the prefixes, including the network connecting to Router15, around the classroom, so all routers should be able to see Router15.

```
Router2(config)#ip  domain-lookup
Router2(config)#ip  name-server  192.168.1.4
Router2(config)#ip  domain-name  workshop.net
```

These commands undo what was configured in step 3 at the beginning of the module. Make sure that you can ping the nameserver before you do this. If you can't ping the nameserver, investigate why.

Note that the team operating Router 6 will be added to add configuration which will allow the DNS server address block to be propagated throughout the classroom network. The extra configuration for Router6 is as follows:

```
router  ospf  41
 network  192.168.1.0  0.0.0.255  area  0
!
interface  FastEthernet0/1
 ip  addr  192.168.1.254  255.255.255.0
 no  shutdown
!
```

21. **(Optional). Enable OSPF name lookups on the routers.** Following from the previous step, now enable OSPF name lookups on the router.

```
Router2(config)#ip  ospf  name-lookup
```

This command enables the display of the OSPF router-ids as domain names. So, rather than displaying the following output with name lookups disabled:

```
router2>sh  ip  ospf  neigh


Neighbor  ID      Pri    State            Dead  Time    Address         Interface
10.0.15.241         1    FULL/BDR         00:00:36      10.0.15.1       FastEthernet0/0
10.0.15.244         1    FULL/   -        00:00:32      10.0.15.18      Serial1/0
10.0.15.254         1    FULL/DR          00:00:38      10.0.15.26      FastEthernet0/1
```

the router will display the following:

```
router2#sh  ip  ospf  neigh

Neighbor  ID       Pri  State            Dead  Time    Address         Interface
router1.worksho     1   FULL/BDR         00:00:33      10.0.15.1       FastEthernet0/0
router4.worksho     1   FULL/   -        00:00:39      10.0.15.18      Serial1/0
router14.worksh     1   FULL/DR          00:00:35      10.0.15.26      FastEthernet0/1
```

which is much more informative.

22. 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.*

23. 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>
```

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

25. **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 IP Protocol does Ping and Traceroute use?

2. Ping the IP address of your neighbour's router (for example 10.0.15.2). Look at the time it took for the ping to complete. Now Ping the IP address of your router on the same segment (for example 10.0.15.1). Look at the time it took to complete a ping. What are the results? Why is there a difference?

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

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