Module 1 – Basic Topology and Router Setup

Objective: Create a basic physical lab with IP addressing and essential router configuration. 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.

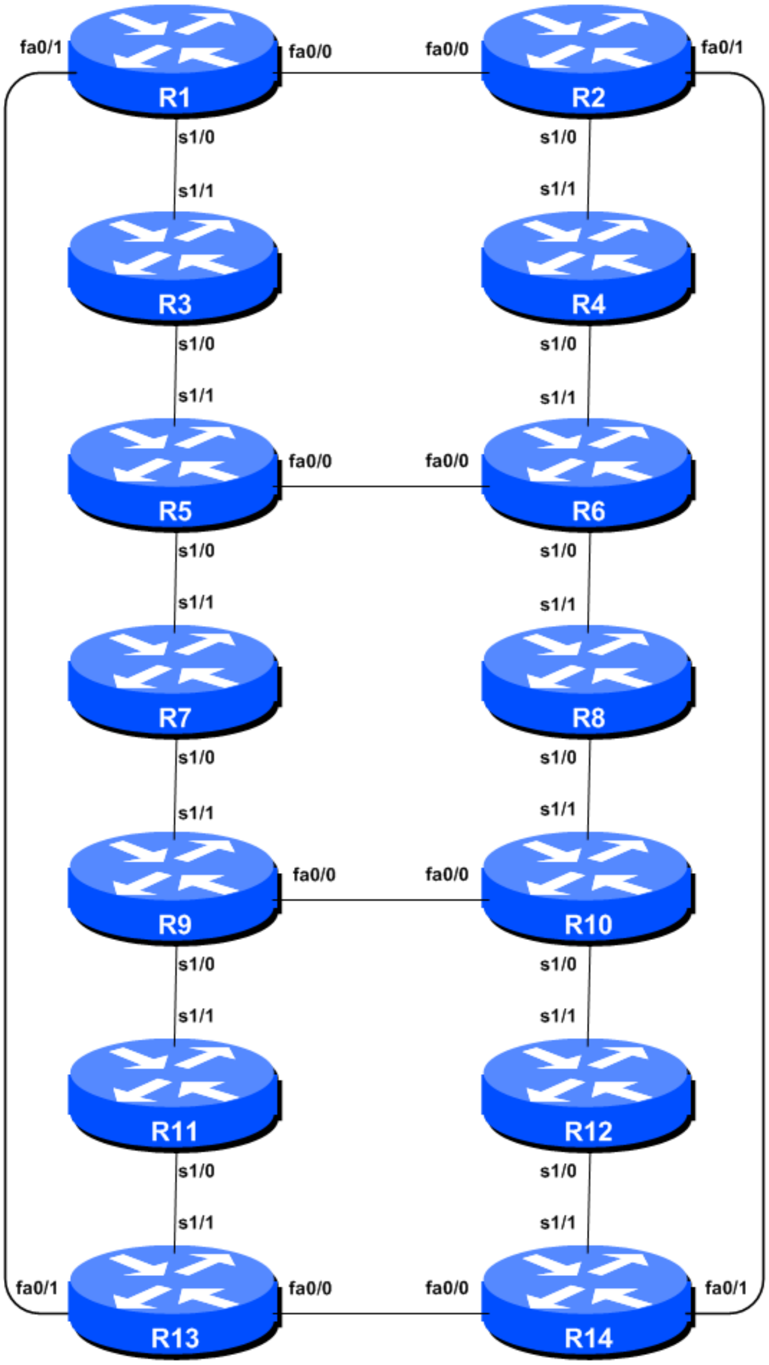


Figure 1 – ISP Lab Basic Configuration

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**. There is no purpose to building BGP while the chosen IGP is not functioning properly. BGP relies on the IGP to find its neighbours and next hops, and an improperly or non-functioning IGP 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.

- 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 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. **Introducing the lab.** This workshop uses Cisco IOS routers running IOS, but on the Dynamips systems – Dynamips translates the Cisco 7200 router MIPS processor instructions in IOS to those of the host system, allowing Cisco IOS images, and therefore network configurations, to be run on a host PC system (usually Linux or MacOS based).

   The lab will have been preconfigured by the instructors, allowing participants to enter the following exercises directly. Please read the following steps carefully.

3. **Accessing the lab.** The instructors will assign routers to each class group, and will indicate the method of access to the Dynamips server. This will usually be by wireless – if this is the case, make a note of the SSID and any password required. Also make a note of the IP address (IPv4, as Dynamips only supports IPv4 access) of the Dynamips server.

   Access to Dynamips will be by telnet, to a high port, which the instructor will specify. Each participant should ensure that their device has a suitable telnet client. Linux and MacOS system have access to a shell command prompt (or Terminal) programme, which allows telnet at the command line. Windows users can use the Windows "Command Prompt" with the telnet client there, but it's notoriously unreliable. Better to install software such as Putty, TeraTerm, HyperTerm or similar third party telnet client.

   Using the client, connect to the router you have been assigned; for example, to connect to the console port of Router 1:

   ```
   telnet 10.10.0.241  2001
   ```

   or to Router 12:

   ```
   telnet 10.10.0.241  2012
   ```

   Once connected, you will see the Dynamips response, followed by the login or command prompt of the router:

   ```
   bash-3.2$ telnet 10.10.0.241 2001
   Trying 10.10.0.241...
   Connected to dynamips.
   Escape character is '^]'.
   Connected to Dynamips VM "r1" (ID 0, type c7200) - Console port


   User Access Verification

   Username:
   ```

If the “Connected to Dynamips VM” won’t appear, even after hitting the Return key several times, please request help from the workshop instructors.

4. **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)#
   ```

5. **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
   ```

6. **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
   ```

7. **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
   ```

8. **Usernames and Passwords.** All router usernames should be ***isplab*** and the password should be ***lab-PW***. We will make the enable secret ***lab-EN***. Please do **not** 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-EN
   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 more secure md5 based encryption.

9. **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.

10. **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 configure the router to send the log messages to a SYSLOG server.

```
Router1 (config)# no logging console
Router1 (config)# logging buffered 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.

11. **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".

```
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

[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.

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.

12. **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.

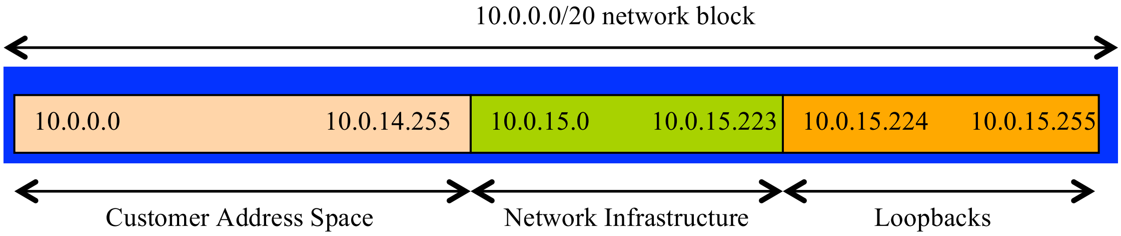
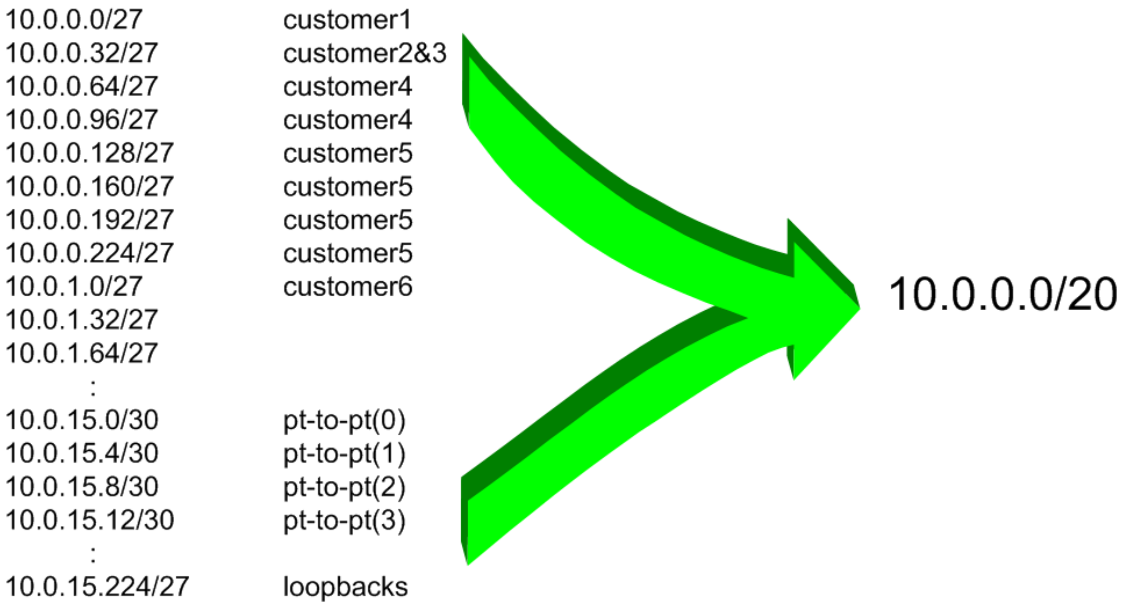


Figure 2 – Dividing allocated block of /20 into Customer, Infrastructure and Loopbacks

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).



10.0.0.0/27	customer1
10.0.0.32/27	customer2&3
10.0.0.64/27	customer4
10.0.0.96/27	customer4
10.0.0.128/27	customer5
10.0.0.160/27	customer5
10.0.0.192/27	customer5
10.0.0.224/27	customer5
10.0.1.0/27	customer6
10.0.1.32/27	
10.0.1.64/27	
:	
10.0.15.0/30	pt-to-pt(0)
10.0.15.4/30	pt-to-pt(1)
10.0.15.8/30	pt-to-pt(2)
10.0.15.12/30	pt-to-pt(3)
:	
10.0.15.224/27	loopbacks

Figure 3 – Extract from an ISP addressing plan

13. **Back-to-Back Serial Connections.** Configure the serial connections as shown in Figure 1. We are using a mix of FastEthernet (100Mbps) and Serial (2Mbps) interfaces for this lab – later on we will configure IGP metrics and these two interface speeds require significantly different metrics given the relative bandwidths available. 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 10.0.15.17 255.255.255.252
Router2(config-if)# description 2 Mbps Link to Router4 via DTE/DCE Serial
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
```

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).

14. **Ethernet Connections.** The Ethernet links between the routers should also be configured now. 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. Here is an example – note again the best practice configuration on the interface:

```
Router1(config)# interface fastethernet 0/0
Router1(config-if)# ip address 10.0.15.1 255.255.255.252
Router1(config-if)# description Ethernet Link to Router2
Router1(config-if)# no ip redirects
Router1(config-if)# no ip directed-broadcast
Router1(config-if)# no ip proxy-arp
Router1(config-if)# no shutdown
```

15. **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

16. **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 12, 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.

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

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?