**Department of**
**Computer Science**

Loughborough University

# Networks Modules
# Lab Reference Guide

Iain Phillips
Asma adnane

Academic year 2018-2019

# Contents


**1 Introduction** **5**

1.1 Overview . . . . . 5

1.2 Lab Safety and Security . . . . . 5

1.3 How to use this guide . . . . . 6

1.4 Equipment Summary . . . . . 6

**2 Laptop Configuration** **7**

2.1 Introdoction . . . . . 7

2.2 Ubuntu Basic Installation . . . . . 7

2.3 Additional Packages . . . . . 10

2.4 Laptop IP Address Configuration . . . . . 10

**3 Network Equipment Fundamentals** **12**

3.1 Introduction . . . . . 12

3.2 Router Hardware Fundamentals . . . . . 13

3.3 Cisco 2801 Interfaces . . . . . 13

3.4 Switch Hardware Fundamentals . . . . . 14

3.5 Check your Understanding . . . . . 14

**4 Cabling and Connections in the Lab** **15**

4.1 Introduction . . . . . 15

4.2 Management Cabling . . . . . 15

4.3 Network Cabling . . . . . 16

**5 Introduction to IOS** **18**

5.1 Introduction . . . . . 18

5.2 Console Connection . . . . . 18

5.3 Basic Cisco prompts . . . . . 19

5.3.1 User Exec Mode . . . . . 19

5.3.2 Privileged Exec Mode . . . 19
5.4 Basic Configuration Commands . . . 20
5.5 Introduction to Interface Configuration . . . 21
5.6 Saving Configuration - RAM, NVRAM and TFTP . . . 21
5.7 Check your Understanding . . . 23

**6 IP Addressing, Routing and Interface Configuration 24**
6.1 Introduction to the Internet Protocol (IP) . . . 24
6.2 IP Routing Background . . . 24
6.2.1 How does it work? . . . 25
6.2.2 Connecting routers together . . . 25
6.3 IPv4 Addressing . . . 26
6.3.1 Example: Loughborough University . . . 26
6.3.2 Address Classes . . . 26
6.3.3 Special IP Addresses . . . 27
6.3.4 Why use different classes? . . . 27
6.3.5 Subnetting . . . 28
6.3.6 Subnet masks . . . 28
6.3.7 Calculating a netmask from slash notation. . . . 29
6.4 Assigning an IPv4 Address to a Cisco interface . . . 30
6.4.1 Why do I see “FastEthernet 0/1 is down, line protocol is down” . . . 30
6.4.2 What’s “no shutdown” . . . 30
6.4.3 IP addressing for serial links . . . 31
6.4.4 IP addressing for HWICs . . . 31
6.4.5 1-port HWICs . . . 32
6.5 Understanding a network diagram . . . 32
6.5.1 Exercise - network planning, network diagram and “show run” . . . 33
6.6 Routing Basics . . . 34
6.6.1 Distance Vector Routing Protocols . . . 34
6.6.2 Convergence . . . 34
6.6.3 Routing Information Protocol (RIP) . . . 35
6.7 Switching on routing in a Cisco Router . . . 36
6.7.1 RIP . . . 36
6.7.2 IS-IS . . . 36
6.8 Domain lookup . . . 37
6.9 Security & Remote access . . . 37

6.10 BGP . . . . . 38
6.11 Passive Interfaces . . . . . 39
6.12 Saving configurations . . . . . 39
6.13 Testing and Troubleshooting . . . . . 40

# Chapter 1

# Introduction

## 1.1 Overview

This document is a general reference guide to the use of the networks laboratory equipment in the Computer Science Department at Loughborough University. This equipment is used to support several taught modules, in both undergraduate and postgraduate programmes within the department. The specific learning objectives of those modules will differ in both depth and scope and so and the lab exercises will not necessarily be the same. Any such exercises are therefore detailed as part of the specific module material, either in lecture notes or lab exercise books rather than in this generic guide, which can be used as a reference source for the lab work in all modules.

This has been developed by Iain Phillips and Asma Adnane at Loughborough University and has incorporated some material previously developed by Ian Napier, Nick Falkner and Olaf Maennel at the University of Adelaide.

## 1.2 Lab Safety and Security

The laboratory contains valuable networking equipment. There is also a large amount of electrical cabling provided to the room. Most computer equipment uses the same kind of power cable - a moulded plastic three-hole plug connected to a standard local power plug. Networking equipment is no different. Because networking equipment, and routers in general, are relatively delicate pieces of equipment, care should always be taken when inserting or removing the power cable for a router or switch. Also, power cables must be up off the floor and correctly secured to a solid point to prevent accidental tripping from pulling a piece of networking hardware onto the floor. Power cables should also be of the correct current rating and, preferably, should be the power cables that are supplied with the equipment. This can be a warranty issue with certain suppliers. If you can't use the cable that came with the unit, always use the best cable you can find.

Floor cables present a tripping hazard and care must be taken when walking around the room and also when deciding where to place your cables. Some experiments require the use of very long patch cables between network components in different parts of the lab as temporary connections. Particular care should be taken to route these away from busy walkways.

Students may not eat, drink or bring food or liquids into the laboratory. Only authorised students may be in the room. Your friends may be fascinated but, unfortunately, they can't come in

unless a member of staff is present and happy for this to happen. Please note that there is a CCTV camera for security in the room.

If there is a fire alarm, you should leave immediately leaving all equipment as it is. All students and staff should then evacuate as normal. Please familiarise yourself with the evacuation procedure now and if you have any questions, please ask one of the supervisors.

The lab equipment is used to set up self-contained networks and must not under any circumstances be connected to the University network or the public Internet. You must also not connect any standard lab computers (the iMacs) to the networking equipment. However, if you wish to connect personally owned computers, you may do this at your own risk and under your own support.

## 1.3 How to use this guide

This guide gives a general description of the equipment but also goes into some depth in describing how to configure particular aspects of the network routers and switches which are included. In some cases, general principles of the operation of various networking systems and protocols are described. This is to ensure that the theory taught in lectures has a link to the practical application of that theory on real network devices. There may then be some overlap between the two, depending on the specific module material. Wherever possible, the sections on configuration stand alone as reference material and do not necessarily need to be read in sequence. Students may pick the appropriate sections to help them with the exercises and problems set in their individual modules.

## 1.4 Equipment Summary

The lab equipment consists of a number of self-contained sets, each with:

- 3 Laptops.
- 3 Cisco routers (probably 2801s, but some are 2900-series).[1, 2]
- 1 Cisco Catalyst switch.
- various lengths and types of network cables.
- power supplies for laptops and power cables.
- a single console cable with USB dongle for connecting the laptops to the network devices.

The Cisco routers are mounted in semi-portable racking units which also loosely house the Catalyst switches, which are not the rack mounted type. The intention is that each set can be used individually by a small group of students for a variety of routing and switching experiments but that the sets can also be networked with each other to simulate the Internet.

The lab is a stand-alone experimental test-bed, **none of the devices should be connected to any university network.** Students are allowed and encouraged to perform your own online research to find out what they need to know. However, please use the regular IT infrastructure provided in the lab for such activities. In addition to these sets of equipment, there is also a single central server set up, with a central network switch. This part is under the control of the lab supervisor and individual groups may use this resource to do network installations of the Ubuntu operating system and as a central inter-group connection point where appropriate.

# Chapter 2

# Laptop Configuration

## 2.1 Introdoction

Your module exercises will probably require you to install some or all of the three laptop computers with the Ubuntu distribution of GNU/Linux and configure them for networking. One of the laptops at least will need to be installed as it is required later to run a terminal session to configure the routers and switches. The laptops come in an “undefined” state. This means you may find an operating system on them or not, you may or may not be able to login. No DVDs are provided; the lab is set up for network installation of these machines (Well, this is a network course). To access the central server you will have to connect your laptop to the central network switch.

One more word about the laptops: some of them sometimes hang during the bootup. This is a problem with the laptop and typically resolves after a second restart. Please don’t be concerned about this problem - unfortunately, there is nothing we can do about this at the moment. It’s a controller problem. We are (almost) in the real world now!

## 2.2 Ubuntu Basic Installation

To get a physical connection, the laptop ethernet adapter should be connected to the central switch using a long ethernet cable. **only ports 16 to 21 on the central switch are configured for this.**

Once you are connected, boot the laptop and hold down **F12**. From the boot menu select the option **’LAN’** as shown in Figure 2.1 to install over the network.

The computer will then attempt to find a server, which provides all the data for the installation. In a first step the computer needs to find an IP address. This is done via DHCP and can take some time. See Figure 2.2.

Via the DHCP (Dynamic Host Configuration Protocol) the laptop will automatically receive an IP address. (This will be covered later in your module lectures). You don’t have to bother too much about this for now. At some point during the installation, the process may ask you if you want to continue without a “default route”, this is okay. Just proceed without a default route here as for installation we will only need to communicate with the install server.

Follow through the installation steps as they appear, making appropriate choices for the options given. However please ensure that you select:

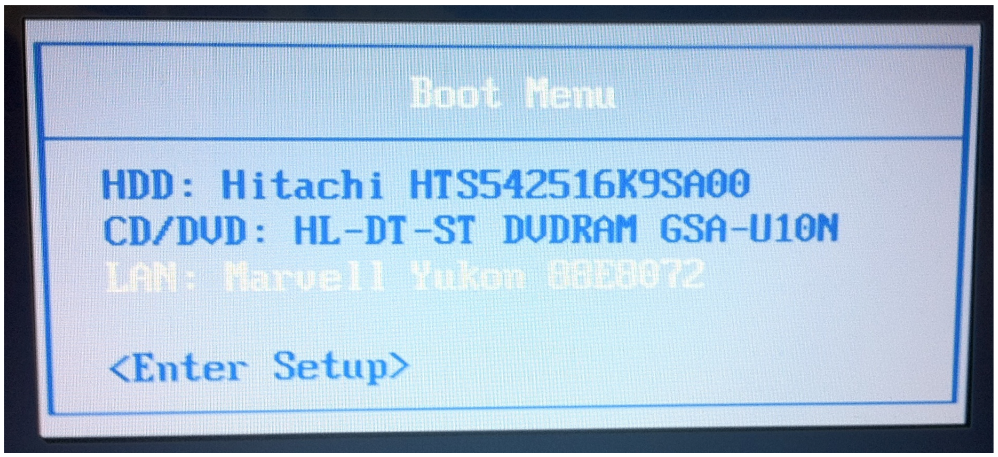



Figure 2.1: Selecting the Network Boot option

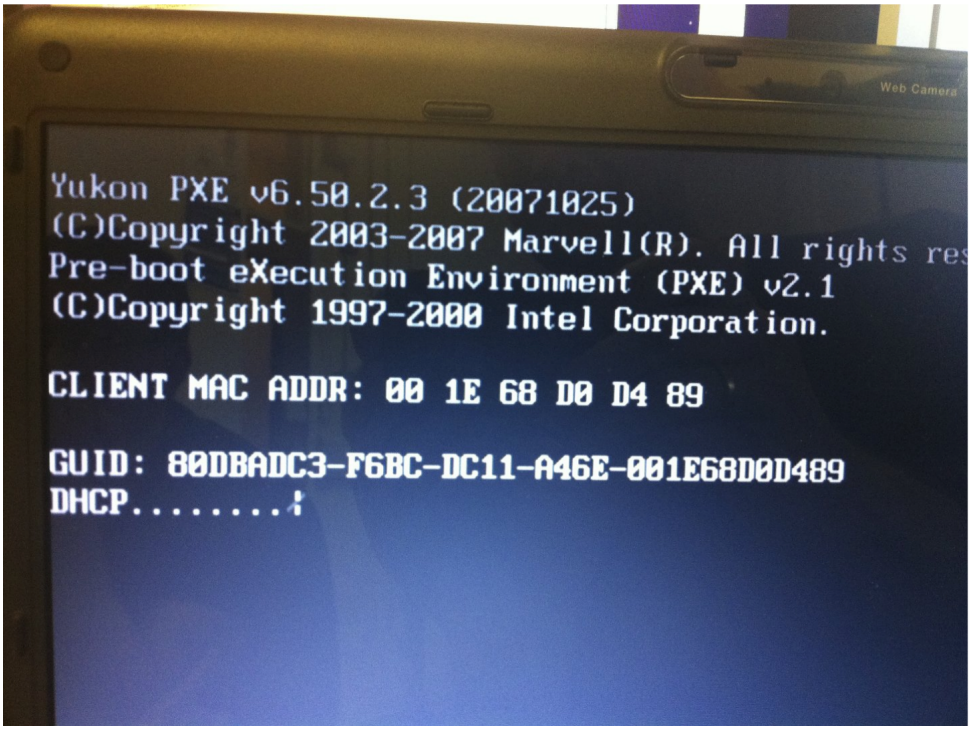



Figure 2.2: Obtaining a an IP Address using DHCP

- Command-line only install of Ubuntu.
- UK keyboard (if you run auto keyboard detect, then you might end up with a US-keyboard layout, which will cause problems later).

At some point during the installation process you will be asked to choose a mirror of the Ubuntu archive. Those are preconfigured places in the Internet, which provides over-the-network installations. However, our test-lab is **not** connected to the Internet. Instead we have setup our own mirror. You therefore will only be able to continue the installation, if you select *enter information manually"* on the top of the selection menu as shown in Figure mirror.

When requested for the server hostname, you will have to enter the IP address of the Ubuntu server, as shown in Figure 2.4. The server that we have setup is **10.2.2.1**. Enter /**ubuntu**/ as the path.

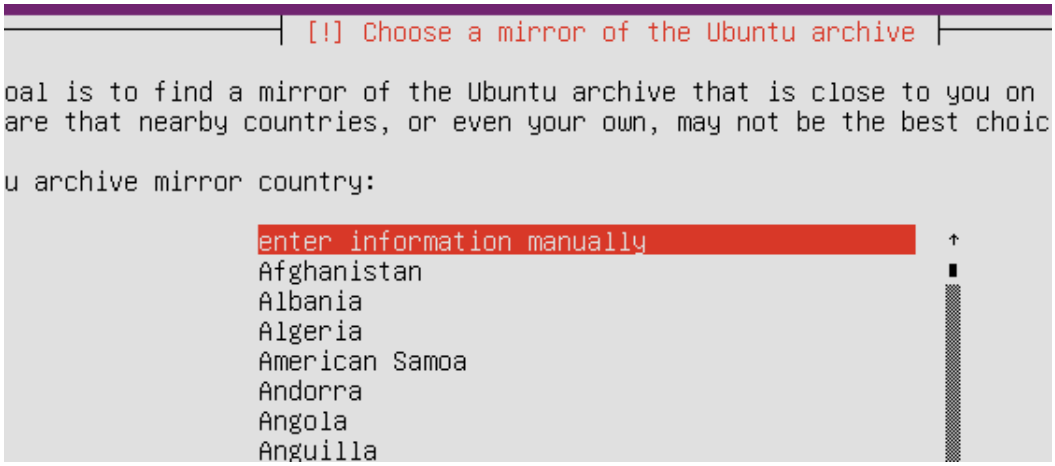



Figure 2.3: Setting the local Mirror

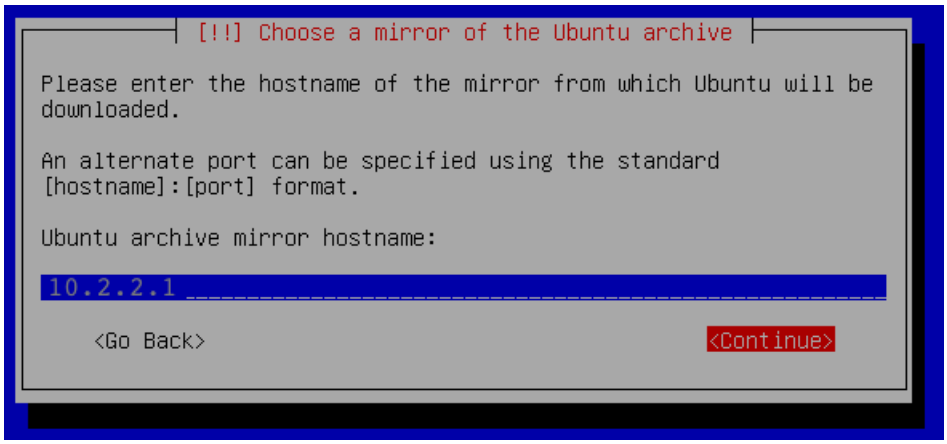



Figure 2.4: entering the Server Hostname by IP Address

Feel free to erase any previous data and/or operating system on the laptop. It is recommended that you simply select the option to use the entire disk for the new Ubuntu OS, as illustrated in Figure 2.5

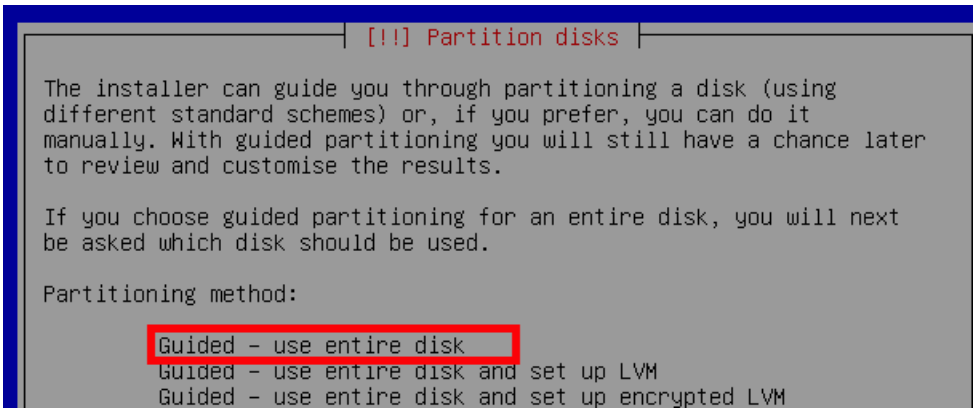



Figure 2.5: Selecting “Entire Disk"

Follow the Instructions on screen to completion and the basic Ubuntu installation is done.

## 2.3 Additional Packages

The module exercises may require you to set up one of the laptops for a specific purpose, such as to act as terminals for router configuration or as servers of various types. You will therefore need to install packages. *Recall:* At no point should you connect your computer to the real Internet. As you have no network connectivity to the outside world, you can only add packages from the central server in the lab. To do this you need to use the package manager from the command-line. Some examples of packages which may be useful are:

- apache2
- openbsd-inetd
- tftpd (e.g., tftp-hpa and tftpd-hpa)
- minicom
- bind9 (and dnsutils)
- traceroute

Once again, it is appropriate for you to do your own research into the necessary commands for the lab exercises set in your module, using internet searches or the `man` linux command. Hint: try `man apt-get` to find out how to use the synaptic package manager. You should also be clear about exactly what the command `sudo` is doing.

**Again, if you have difficulties finding the answers to your questions, please do not hesitate to ask your supervisors**.

## 2.4 Laptop IP Address Configuration

It is normal practice to work from the command-line for all configuration and set-up tasks in the lab. Developing and demonstrating the skills to do this is an important learning objective of the Networking Modules. Our aim is to teach some fundamentals and consider it therefore essential that everyone is able to setup a network with commands such as `ifconfig`.

Depending on the requirements of the specific module, you will probably have to choose an appropriate IP addressing scheme for the various exercises. This will involve using subnet masking to make efficient use of any address space allocated to your group. Having mapped out your address scheme, the laptops can be set up with the correct IP Addresses, net masks and if needed default gateways.

In Ubuntu, the file `/etc/network/interfaces` holds information on IP addresses for your computer. You might also want to brush-up your knowledge about the Unix command `ifconfig`, which will prove very useful. Using an Internet search to find out answers to questions for yourself is a key part of practical networking and you are expected to make use of it in these lab sessions. However, you may find the Unix command `man` very helpful, which gives you information on what a particular command is doing.

Note that after you've installed the machines the `/etc/network/interfaces` file is set up to configure the interfaces with dhcp. This creates a local process to keep the dhcp lease updated, which you need to kill. Often the easiest way to do this is to change your configuration in `/etc/network/interfaces` to `static` (filling in the other fields as necessary, see `man interfaces`) and reboot.

**If you have difficulties finding the answers to questions you may have, please do not hesitate to ask your supervisors. However, many of their answers will be to direct you to the right information, rather than simply telling you the answer.**

# Chapter 3

# Network Equipment Fundamentals

## 3.1 Introduction

Switches and routers are the workhorses of modern networking. In the OSI reference model (and the TCP/IP or DoD reference model), switches operate at *layer 2* (Data Link) and routers operate at *layer 3* (Network).

Switches are used in Local Area Networks (LANs) to connect groups of machines that are usually centrally organised or under a common administrative body. In an ethernet LAN, what you send out onto the LAN is an ethernet *frame* which is then handed to the Physical layer to carry out the physical transmission.

Routers are used to connect these LANs together, usually over Wide Area Networks (WANs), so that computers in one place can communicate with computers in another. Routers use the Internet Protocol (IP) to forward *IP packets* from one router to another.

Figure 3.1 shows the relationships between LANs, WANs and the equipment that connects them. Note the symbols for routers and switches - these aren't just restricted to Cisco equipment. In addition to the laptops whose set-up is described in Chapter 2, the portable lab

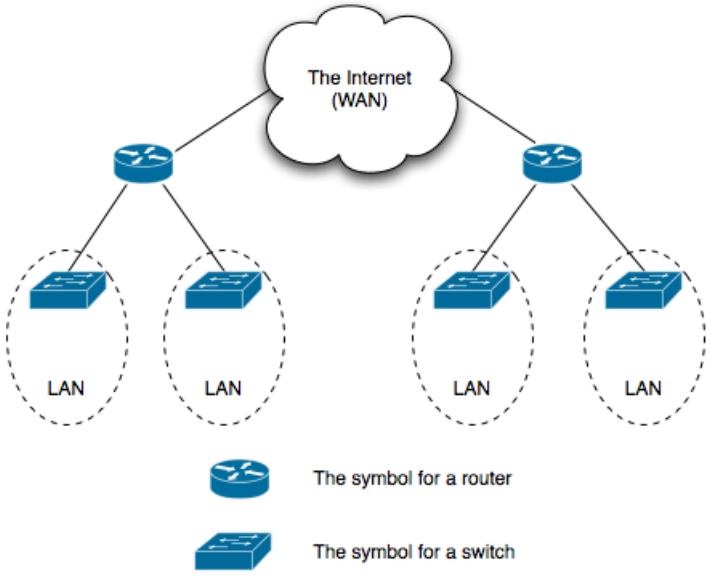



Figure 3.1: LANs, WANs, Routers and Switches"

contains specific networking equipment:

- Routers - Cisco 2801 Integrated Services Routers.
- Switches - Cisco Catalyst 2960 8 port 10/100 MB Switches.
- Console cables.
- Category 5e network cables.

These are housed in a small racking enclosure for each set of kit; the routers are rack mounted and the switches are loosely installed. This kind of rack mounting arrangement is a miniature version of that which would be found in the comms cabinet of most small to medium sized enterprises. Although Cisco equipment has been chosen for the lab, the general principles are the same for other manufacturer's products.

## 3.2 Router Hardware Fundamentals

The fundamental role of a router to receive packets from one of its interfaces and forward them to the correct interface for their onward journey to the ultimate destination. They move packets from one network to another but they also determine the best path to do this. An interface is either a physical device, which is part of or plugs into the router, or it is a logical device that works in conjunction with a physical interface. A router will have the following constituent parts:

- **Chassis and power cord**. Some commercial routers (but not the Cisco 2801 models in out lab) also have an external power supply unit. The purpose of the power-cord and chassis are pretty obvious - the power cord provides power to the unit and the chassis provides a frame for holding the internal electronics and interfaces.
- **Console and Auxilliary (Aux) ports**. These are management ports. Without these, you couldn't talk to the router unless a network interface was active. The *console* connection allows you to connect to the router directly when physical access is available without using the working network. This means that a router can come from the factory, without any configuration and be configured in a secure way by the administrator communicating directly to the console port before being connected to the network. This is not just convenient but also an important security feature.The *Aux port*, short for auxiliary, can also be used to hook up a modem so that an administrator can talk directly to the router, over the phone line, without having to use the network. This can be an important feature to maintain remote connectivity even if there are network problems. This may prevent the need for expensive and inconvenient site visits by a network engineers.
- **Interfaces (Fixed or slot-based).** Interfaces talk to networks. If your router has no network interfaces, then it won't be doing much routing. Your router should really have at least two interfaces. Interfaces come in many different types but can be broken down into two basic groups: WAN interfaces and LAN interfaces.
- **Status lights**. These give a quick, usually very basic indication of the status of the router.

## 3.3 Cisco 2801 Interfaces

Most of the routers in the lab kits are Cisco 2801 Integrated Services Routers (ISRs). In Cisco-speak, this means that they are capable of performing a number of different routing tasks. To

help with this, the 2801 has a number of slots on the front, which can take different interface cards. These are called Wide-Area Interface Cards (WICs), High-Speed WICs (HWICs), Voice Interface Cards (VICs) or Voice and Wide-Area Interface Cards (VWICs). To simplify the control circuitry in a 2801, you can't put certain cards into certain slots but, to help your memory, this is clearly written in small letters on the chassis underneath each slot. We will be using Ethernet technology, which is provided by our routers using an HWIC (even though Ethernet is not traditionally considered a Wide-Area technology).

## 3.4 Switch Hardware Fundamentals

Like a router, a switch has a chassis, a power cable and some network ports. It provides a way to get ethernet frames from one switch port to another. It is in some ways analogous to a router, but rather than forwarding according to layer 3 IP addresses in an IP packet header, it forwards according to layer 2 MAC addresses in the ethernet frame header. Switches are of course blissfully unaware of IP or any layer 3 information. The switch ports are designed to take RJ-45 terminated cables.

Basic setting up a switch to create a sinle LAN is pretty simple. Turn it on, plug things in, data flows through your local-area network. It's actually not this simple in operation and it can be quite complex to set-up. A lot of work has gone into standards and compliance at both layer 1 and layer 2 to ensure that you can plug most things into a switch and frames will flow. An example of a more complex switch set up occurs when Virtual LANS (VLANS) are implemented. A single physical switch can be configured to act logically as a number of completely separate switches, thus splitting the LAN into smaller, logically unconnected networks.

## 3.5 Check your Understanding

This chapter gives a basic description of the physical components in the lab kits and their role in simple networks. The short quiz below is designed to help you check your understanding. Try to answer the questions first without reference to any notes and then if necessary read up to get the answers. Where there are choices, delete whichever doesn't apply. If there's a blank, fill it in!

1. Routers interconnect networks over a _____
2. Switches are used to connect machines in a ____
3. For a router to actually route, it should have at least ____ interfaces.
4. Console connections are a type of _____ port.
5. Which slot holds the HWIC on the Cisco 2801 ISR?

# Chapter 4

# Cabling and Connections in the Lab

## 4.1 Introduction

One of the most important jobs in networking is getting the cables into the right place. Whether it's a cable connecting your company to a WAN, the link from your printer to a switch or the cable that connects your management port to your computer - it all has to work. Within commercial buildings, network cabling is installed according to structured cabling specifications, using cables in trunking, wall ports, network patch panels and cabinets. In the lab however, all our connections are made using temporary patch cables. The connections used in the lab can be divided into two distinct types, *Management Cabling* and *Network Cabling*.

## 4.2 Management Cabling

Cisco equipment, like most routers, have a console port that allows their management and configuration without the requirement to go over a network. This is a convenient method to set up an initial secure configuration *before* connecting the device with potentially insecure factory defaults in place to a secure network. As an example of the risks of installing factory default configured devices, consider wireless networks. Most Internet security breaches occur through wireless networks that have not been properly secured as they are still using the (well known to hackers) factory defaults. Commercial routers avoid this potential risk by shipping with a configuration that allows you to log into the console port but all of the interfaces are shut down. That way, even if you connect one of these routers into a network, no packets traverse the links.

The downside of a console port is that it is not, generally, a traditional network port - it's a type of serial port supporting the RS-232 signalling standard. You'll see serial ports like this on the back of PCs in either 9 or 25 pin configurations. They're often called things like COM 1 or COM 2. Before we had USB, which is another serial standard, mice and modems used to get plugged into these. Serial connections have to be configured correctly so that both sides of the connection agree on the way that messages are going to be sent. To set up a serial connection correctly, you have to establish the following settings:

- The communications rate (how much data will be sent in a second).
- How many data bits will be used.
- The parity.

- The stop bits.
- The flow control used.

The only information needed in the lab is the settings required to communicate via the console port on the Cisco routers and switches:

- **Device: /dev/ttyUSB0**
- **9600 baud**
- **8 data bits**
- **No parity**
- **1 stop bit**
- **No flow control.**

The terminal emulation software used will be determined to some extent by the OS chosen for the laptops in the lab. Whichever terminal emulation software you're using (HyperTerminal on Windows or minicom on Unix/MacOS), you will need to make your settings the same. *Most likely you have to change only the communications rate to 9600 baud*. To get signals from your computer to the console port, you need more than a piece of software. You will need a physical connection to the console port. Cisco uses a special cable called a *rollover* cable. This looks like a flat network cable but it only works as a connector cable between a computer and a Cisco router console port. We also have an additional issue. Modern laptops do not have anymore serial ports. We therefore have use an USB-serial-adapter cable. On Linux this works without any extra drivers. You may notice that on those Toshiba-Laptops in the lab, the USB-roll-over converter may only work on the **right side** USB-port and does not work with the USB-port on the left side.

## 4.3 Network Cabling

Local-area networks have, after years of competing standards, pretty much settled into a single standard for interconnection cabling. This standard is Category 5, 5/e or 6 Unshielded Twisted Pair (UTP) cabling using RJ-45 ends. The category information deals with the technical specification of the cable and describes how many pairs of wire are used, how they are shielded and even how they are twisted. The coloured conductor insulation is usually visible in the clear blocks at each end of the cables. These colours indicate individual wires. There are 4 pairs in a UTP cable, although not all of them may be used.

For Ethernet 802.2 frames, the most common LAN protocol, UTP is good for about 100 metres. If you want to send a signal further than 100 metres, you'll either have to pick a different technology or use a repeater to pick up the signal and send it out again. There is no right or wrong end to a UTP cable. You should be able to plug the cable in either way and get the same result. UTP patch cables are generally produced in two different ways: straight-through and cross-over. To connect a router to a switch, or a switch to a computer, you need a straight-through cable. Historically, to connect two devices of the same type together (for example a switch port to another switch port), it would have been necessary to use a cross-over cable. These are now less common, as most switch ports have an auto-sensing facility which automatically sets up the input and output pair connections correctly.

Note that the HWIC ports and the laptop ports are autosensing. The main gigabit ports on the left of the router are not. Think about what sorts of cables you would need to connect two routers together by any pair of ports. Similarly think about connecting a laptop to a router. Cross-over cables are a scarce resource, only use them when necessary.

# Chapter 5

# Introduction to IOS

## 5.1 Introduction

IOS is the Internetwork Operating System for Cisco routers and switches. It's not the only router OS in use but it is the one you need to know to configure the Cisco kit in the lab. IOS allows you to change what the router does. IOS also runs on Cisco switches and this allows us to change what they do as well. What we're going to do in this section is give you an introduction to IOS so that we can get you configuring a router.

## 5.2 Console Connection

To access the iOS prompts and enter the commands needed to initially configure the router, it is necessary to establish a console connection using the router console port and a terminal. In the lab, the terminal function is realised using a terminal emulation program installed on one of the laptops. If the laptop is running Linux, you will be using **minicom**. If windows was installed on the laptop, **hyperterminal** is the usual emulator. In this guide we will assume that **minicom** is to be used.

Before you start the connection program, you need to establish a physical connection between the laptop and the router. Take the blue console cable and plug the RJ-45 end into the router and the the 9-pin end into the USB-to-serial converter-dongle. The USB can then be connected directly to the computer.

The basic Ubuntu installation will need the **minicom** package installed from the central server (Section 2.3). It can then be started by typing : **sudo minicom** at the command prompt. To connect via minicom, the parameters for Cisco comms (given in Section 4.2 need to be entered. The minicom serial port setup menu is accessed by typing `CTRL-A, Z, O` and pressing `ENTER`.

After a brief pause (which is always slightly longer than you expect), you should see text start to scroll up the screen as the router boots. When the router has finished booting it will present you with a prompt. Have a look at what scrolls up the screen. The router is going through a set of self-test and booting operations as it brings itself up to a usable state. Finally, after everything else has been done, it will allow you to log in. You will be presented with a line that looks like this:

**Router con0 is now available**

Press RETURN to get started.

The first time you log-into the router you

You will then be presented with a prompt that looks like this, e.g.:

```
1 router>
```

(We will present IOS commands and responses in the format above. Note the line and the number to its left are not part of the command, but there so we can refer to specific lines.)

## 5.3 Basic Cisco prompts

Cisco iOS has a number of different modes, each of which has its own options and privileges. These modes each present different prompts to the user. It is important to be familiar with these different prompts and know how to navigate around them.

### 5.3.1 User Exec Mode

The > prompt tells you that you are in User Exec Mode. This is mostly used to view statistics. Type ? to see which commands are available to you.

The biggest problem with user exec mode is that you can't change anything or view the configuration. Obviously, this is less than useful to you if you want to change anything. Your router is shipped from a factory with a blank or default configuration - without a higher level of login, you'll never be able to change anything.

### 5.3.2 Privileged Exec Mode

The best thing about user exec mode is that it's a natural stepping-stone to *privileged exec mode*. To get to this mode, type **enable** at the ">" prompt and hit return. You'll be prompted for a password. This looks like this:

```
1 router>enable
2 router#
```

Note that the prompt has changed from a > to a #. This lets you know that you have switched modes - it's informative and it's also a warning.

If you want to get out of privileged exec mode, you type **disable** and your prompt will return to >

Let's try some commands in this mode. You can view the configuration of the router, look at the state of the interfaces and see which version of the IOS the router is running - among other things.

Here are some commands to try. When you try them, answer the question associated with the command.

- `show version`. At the bottom of the text, you will see the words Configuration Register. What is your configuration register set to?
- `show interface summary`. How many interfaces are installed in the router? What are their names?

- `show interface fastethernet 0/0`. The very first line will say something like:

```
!FastEthernet0/0 is ...., line protocol is .....
```

  What are the missing words?

- **Interface Numbering**. Before we go any further, you have to know how Cisco routers number their interfaces. As we can have several of the same kind of interface in the same router, Cisco equipment has a numbering scheme based on either fixed configuration or the position of an interface card in a slot. Fixed interfaces, like the Ethernet ports on the front of the 2801, are numbered 0/0 and 0/1.

- **Configuration Mode**. Even though you can look at a lot of things when you're in privileged exec mode, you still can't change very much. Before you leave privileged mode, type **show run** to show the configuration that the router is currently using. You'll be able to page through this by hitting the space bar. Some of the commands that you see here are the ones that you'll be changing later on.

  To get to configuration mode, we use the **config** command. If we were to type that by itself, the router would then ask if we wanted to configure from the terminal, memory or network. We want to configure from the terminal so, to save time, we'll use the **config t** command.

```
router#config t

Enter configuration commands, one per line. End with CNTL/Z}

router(config)#
```

  That last line tells you that you have entered configuration mode. Be very careful in this mode - this is where you can really break things.

  (Don't worry too much, most of the time we can just power cycle the router and fix things. Even if you save your changes we can restore a working configuration.)

  You can leave configuration mode by holding down `CTRL-Z`. You should get back to the plain # prompt.

  The prompt will change slightly depending on which part of the router you're configuring. Sometimes this is more helpful than others - just keep an eye on the prompt as it can help remind you what you're doing.

## 5.4 Basic Configuration Commands

Let's look at two basic configuration commands. As you try each one, answer the question that accompanies it.

- **Hostname**. If you type **hostname testing** at the configuration prompt, you'll change the router's name. What has happened to the configuration prompt?

```
testing(config)#
```

- **Router**. This is a router, so why don't we try turning routing on? (This isn't going to work properly but it will illustrate the point.) Type **router rip** at the configuration prompt. What has happened to the configuration prompt?

```
testing(config-router)#
```

Type exit, then change the hostname back to what it was when you started. Then type exit or CTRL-Z to exit configuration mode.

What does the prompt look like now?

```
<original hostname>#
```

Re-enter configuration mode and type `interface fastethernet 0/0`. You are now in interface configuration mode. Configuring the interfaces is one of the most important things we do as nothing is going to work unless we set these up correctly.

## 5.5 Introduction to Interface Configuration

Routers have to know where to put network traffic so that it goes to the right place. When they send out a packet to go to another network, they will only send it out an interface that should have a path to that network. We'll talk about routing in detail in the next section.

To get routers working correctly, you have to set up the router so that the interfaces of routers connected across the WAN are correctly defined and, if required, have any special characteristics set. Establishing a WAN connection can be quite elaborate, even with most of the detail hidden, but we'll be doing simple configuration.

What all IP WAN interfaces must have are:

- An Internet Protocol (IP) address
- A subnet mask
- A command to switch themselves on

Once this is set up, we can tell the router how to establish connections between interfaces. More importantly, the traffic coming in one interface can be sent out another interface and get to its destination.

This important subject is described more fully in Chapter 6.

For now, exit configuration mode and leave yourself at the privileged exec mode prompt (#).

## 5.6 Saving Configuration - RAM, NVRAM and TFTP

When you've taken the time to set up a router, it would be a real pain to have to re-enter all of that data whenever the router reboots.

- Enter configuration mode and change the router's hostname to Sandwich.
- Exit configuration mode and switch the router off at the power switch. Wait 5 seconds and then turn the router on again. After the router has booted, login and enter privileged exec mode.
- What is the hostname now?

- Whatever it was before the change.

The reason that the hostname has changed back is that you didn't save your changes. While some pieces of networking equipment will automatically save what you do, a complex system like a router will change in response to your commands but will not save your changes - just in case you did something you didn't really want to do.

A Cisco router has Random Access Memory, which does a number of things, but it also keeps the running version of the IOS configuration. When you make changes in configuration mode from the terminal, you are making a change in the RAM version. This changes what the router is doing right now but a reboot or power failure will bring back the old version.

The version that you see when you start up is loaded from Non-Volatile RAM (NVRAM). This memory will survive a reload or a power failure but, in order to get a changed configuration into this memory, you have to issue a command.

You don't issue the backup command from configuration mode; you issue it from privileged exec mode.

Go into config mode and change the hostname to Section2. Exit configuration mode and type **copy running-config startup-config**. This command will copy the running version of the IOS configuration from RAM to NVRAM.

Instead of power-cycling the router, you can use the **reload** command from privileged exec mode to restart the router.

Reload the router, wait for the boot sequence to complete and login. You should now see the hostname has survived the reboot.

You can even save a copy of your configuration on another machine if you want to. You do this using the Trivial File Transfer Protocol (TFTP). This is very similar to the copy to NVRAM but requires some additional information.

You need to know where the TFTP server is (its IP address), that you can connect to. We will learn how to do that later.

```
router\#copy running tftp

Address or name of remote host []? xxx.xxx.xxx.xxx}

Destination filename [router-confg]? Section2-confg
```

The good thing is that you can also restore the configuration from a tftp server. You may have guessed what the command is: **copy tftp run**

First of all, change the hostname to Sect2, then get back to privileged exec mode.

```
router\#copy tftp running

Address or name of remote host []?xxx.xxx.xxx.xxx}

Source filename []? Section2-confg
```

Before you finish, copy the lab from the TFTP server into NVRAM using **copy tftp startup**. The IP address of the TFTP server is the same as for the other examples and the filename is `<Routername>-cfg` If you were using LabA, then the filename is `laba-cfg`.

Once you've successfully copied the file, type **reload** and then hit return when prompted to

confirm your decision. This will reboot the router and it should come up in the default configuration.

When you're finished, get out of minicom by typing `CTRL-A`, followed by q. Select 'Yes' by hitting return and then log out of the Unix server.

Guess now how to store a running configuration on a USB key. The lab supervisor may provide you with some USB keys.

## 5.7 Check your Understanding

This chapter gives a solid introduction to Cisco iOS:

- How to access it on the routers in the lab.
- The main modes and their prompts.
- Navigating around and checking status.
- Some basic configuration options.
- Saving and restoring router configurations.

It should be noted that many of these basics apply equally to Cisco iOS implemented on Switches.

The short quiz below is designed to help you check your understanding. Try to answer the questions first without reference to any notes and then if necessary read up to get the answers. Where there are choices, delete whichever doesn't apply. If there's a blank, fill it in!

1. You saw three different prompts while working with the router. What are the prompts and which mode do they represent?
2. What is the command to enter privileged exec mode?
3. What is the command to enter configuration mode?
4. What is the key sequence to exit configuration mode?
5. What must all IP WAN interfaces have in order to work?

Feel free to have a look around in privileged and config mode. Just make sure that you put the default config back into the right place before you leave.

(As a final note, we often use the term *enable mode* to mean privileged exec mode.)

# Chapter 6

# IP Addressing, Routing and Interface Configuration

## 6.1 Introduction to the Internet Protocol (IP)

Internet protocol (IP) is essentially to the day-to-day running of the Internet, unsurprisingly. IP sits at layer 3 of the OSI model although it's actually a protocol from the US Department of Defence network model.

To explain how IP works, we have to go back to our discussion of LANs and WANs. Switches work by sending frames out ports but the frames are addressed using a MAC address (often called Ethernet address). This is a 6-byte number, often written in Hexadecimal notation, which is unique to a given network card or computer. An example is 00:04:F2:20:1B:04. It's non-hierarchical and is often based on a range of numbers assigned to a given vendor. An analogy to the way LANs work is someone standing in a corridor and shouting out a telephone number until someone recognises their number and yells back.

As you can guess, this scales really badly. Modern switches learn where devices are and, once they've found them, they only shout down the appropriate ports. Even with this, there is no guaranteed way of predicting where a device is going to be until you've found it. This can be inefficient in large LANs - it would be crippling in WANs.

Internet Protocol assigns a logical address to a networked device. As we'll discuss, IP addressing is hierarchical, which means that we can go looking for an address based on where it should be - rather than shouting across the world looking for someone's computer. Most importantly, the IP address associated with a router will allows us to find it and, once we've found that, the router can then search its LAN to find the machine we're looking for. (The router will use the Address Resolution Protocol, ARP, to find the MAC address of the client with the target IP address. It then uses standard LAN communication to send the required information to the client.)

## 6.2 IP Routing Background

A route is just an instruction to the router that says "If you get a packet that looks like this, send it out through this interface." To get a packet from once place to another, where it can eventually be delivered, a router must know at least the following information:

- Destination address
- Which of its neighbour routers have knowledge about remote networks.
- Possible routes to remote networks.
- The best route to a remote network.
- How to keep track of the routing information.

There are two ways that the router can find out about remote networks:

1. Neighbour routers
2. Network administrators

Obviously, you can't know about every network, which is why the router has to be capable of querying other routers in order to establish where to send things. If you set a route by hand, this is a *static route*. *Dynamic routing* occurs when the router communicates with another router and finds out about networks from the other router. Usually, we use a combination of these two approaches.

The router uses a *routing table* to keep track of which routes are associated with which interface and any other information that is relevant for that route. Both static and dynamic routing makes changes to the routing table - if it's not in the table, it's not going to be used.

### 6.2.1 How does it work?

A client sends out data for another client on a remote network. This goes through the LAN until it reaches the router. The router receives it on one interface. It inspects the destination IP address to see where it should go. If there is no entry in its routing table that matches the address, and it has no *default route*, it will send back an ICMP `"destination unreachable"` message. If there is an entry in the routing table, the router will packet-switch the packet to the correct interface and send it on its way.

If the client had been sending data to a local machine, one sitting on the same LAN, it would have sent out the message on the LAN and waited for the right machine to pick it up. Because it was a remote machine, the client would send out the message on the LAN but it would send it to the router. Every other machine on the LAN will ignore this packet and the router would send it out as described above.

### 6.2.2 Connecting routers together

Because routers must be able to talk to each other, indirectly through the WAN cloud, every interface, which participates in IP routing, must have a valid IP address.

A more subtle point is that router interface IP addresses must be in a certain relationship to each other to allow them to communicate. After all, if you use routers to talk to different networks, how can you link two routers together unless they belong to the same network?

If that seems confusing, hang in there. We're about to discuss IP addressing.

## 6.3 IPv4 Addressing

We'll be talking about Internet Protocol version 4 (IPv4) in this section. IPv4 has a simple addressing structure. IP addresses are made up of 4 bytes, often called octets. They're normally written with dots between them, this is called dotted notation. A byte can have values from 0 to 255 so the range of possible IP addresses is 0.0.0.0 to 255.255.255.255. Not all of these addresses are valid - some have special meaning, as we'll see later.

With 4 bytes, we have a total of 32 bits that we can use to make up different numbers. A quick bit of arithmetic will tell you that this gives you over 4 billion possible IP addresses. However, we split this into two portions so that we can group machines together in a logical manner. Why? If we used a flat addressing scheme, every router in the world would have to keep an entry for the router that knew where to find one of 4 billion possible addresses. If we use a hierarchical scheme, we can go looking for a router that knows where to find a group of IP addresses. This greatly reduces the amount of routing information required.

### 6.3.1 Example: Loughborough University

Every IP address inside the University looks like 158.125.xxx.xxx. This is the network address for every machine of the University network. The University web server has the IP address 158.125.1.208. The 1.208 is the node or host address of the web server.

When someone accesses the University's web server, all their router has to do is to fire out a packet which is looking for the network address 158.125.0.0/16 Once the packet gets here, the University's routing backbone will work out where to send it to find the correct LAN. Even inside the University, routers will talk to routers because we have a lot of networks on the one campus.

### 6.3.2 Address Classes

Originally, IP addresses were split into classes to make it easier to sort out which part of the IP address designated the network and which part designated the host. The original classes were A, B, C, D and E. With all of the valid address ranges listed, there are some special cases which are still not valid.

Table 6.1: default

| IP Class | Start IP | End IP |
|---|---|---|
| Class A | 0.0.0.0 | 127.0.0.0 |
| Class B | 128.0.0.0 | 191.255.0.0 |
| Class C | 192.0.0.0 | 223.225.255.0 |

- Class A addresses allocated the first byte for network addresses and the last three bytes for host addresses. Valid class A addresses start with 0 and go up to 127.
- Class B addresses allocated the first two bytes for network addresses and the last two bytes for host addresses. Valid class B addresses start with 128 and go up to 191
- Class C addresses allocated the first three bytes for network addresses and the last byte for host addresses. Valid class C addresses start with 192 and go up 223.

- Class D is used for multicast networks and Class E is for research purposes.

**Question:** What class of network has been assigned to Loughborough University?

### 6.3.3 Special IP Addresses

A network address that is made up of all binary zero is taken to mean this network or segment. A network address of all binary 1s means "all networks". Similarly, a node address of all 0s means "any host on this network" while a node address of all 1s means "all hosts on this network".

If everything is set to zero (0.0.0.0), this could mean "any network" but Cisco routers take this to designate the default route. We'll talk about the default route later. If everything is set to 1s (255.255.255.255), this is a *broadcast* to every node on the current network.

We'll talk more about broadcasts and network addresses shortly.

### 6.3.4 Why use different classes?

These classes divide up networks by their size. Networks with millions of machines are found in class A - but there are only 128 class A networks available. Class B networks can take 65,534 node addresses in each of the 65,534 possible class B networks. Class C networks are the smallest class, with only 254 nodes in each of the 2 billion possible class C networks.

We don't just use classes to separate networks (this is referred to as *classful* networking). We can also use variable-length subnet masks (VLSM) to provide *classless* networking. We'll talk about this briefly but we don't need to go into too much detail here. What you need to know is that we can group hosts into subnets, however we do it.

The most important question of all is: Why do we do this?

Firstly, we can logically group our computing resources in an efficient way and make good use of the IP address space.

However, the most important reason is that we can use subnetting to reduce traffic on a network. There are very good reasons why broadcast messages can't be sent out to every host on the network. If a network is very large, then the number of broadcasts can also get very large - causing a heavy load on the routing backbone and packet switching engines.

Ideally, you should create lots of small subnets where most of the traffic is from machines in the subnets to each other. In this case, most of your traffic is LAN-based, rather than through the router or out the WAN link to another router in a different place.

A large network can also have a greater potential for problems because it is so large and potentially complex. It's much easier to spot problems in smaller networks and they cause less widespread problems.

Finally, a really big network may span countries or states. At this point you will be using your WAN links extensively and this costs money. If you group your machines logically and set your network up so that most of the traffic is local, you won't spend all of your cash on your WAN links.

So, in summary, here are four good reasons to subnet below the class level:

1. Reduced Network Traffic

2. Better Network Performance
3. Simplified Management
4. Big World-Spanning Networks Cost A Lot

### 6.3.5 Subnetting

Sometimes it's handier to be able to group nodes together in a way that is more *fine-grained* than the classes would allow. For example, the School of Computer Science does not have 65,000 machines in it so it doesn't need it's own Class B network range. You could give it a class C address and try to squeeze machines in but, if you wanted to have more than 254 machines, you'd have to add another class C.

The University has been assigned the Class B network address 158.125.xxx.xxx. Rather than just allocate the numbers out, we divide this large range into several smaller ranges - we subnet our network space to form subnetworks.

For a long time, the University was only able to subnet the class B into what looked like, but weren't, class C subnets. So, there was a 158.125.1.xxx network, 158.125.2.xxx network and many others where the first three bytes specified the network and the last byte was used for the node.

This is one of the big advantages of subnetting. We can continue to use the hierarchical nature of the University's class B while getting the apparent use of some class Cs. If we had just gone out and bought some class C addresses then we would have to set up more complex routing instructions to make sure that every one knew that this group of class C networks actually all belonged to us.

Now, it's very important that you realise that we do not actually use Class Cs here. We use a subnetted class B. Class Cs must have a network number that starts with 192 and goes up to 223. So what's going on?

### 6.3.6 Subnet masks

A subnet mask is used to tell the router, and any machines using IP, which of the bits in the IP address will be used for the network address and which bits are used for the node address. This way, any machine will know when something is local and when something is remote.

The default subnet masks for the Classful networks are:

| Class | Format | Default Subnet Mask |
|---|---|---|
| A | network.node.node.node | 255.0.0.0 |
| B | network.network.node.node | 255.255.0.0 |
| C | network.network.network.node | 255.255.255.0 |

A subnet mask masks out the network component of the address to tell the configured equipment where the node bits are. As you would expect, Class A addresses define node addresses by ignoring the first byte.

I can't change the class B default mask to be 255.0.0.0 - this would break the whole class set-up because I could then start using the second byte for node addresses. Since everyone else in the world expects me NOT to do that, I'll be breaking standards if I try that.

However, I can use a class B address with a mask of 255.255.255.0. In this case, I'm not going to cause a problem outside of my own network providing that I use this consistently. When I use a class B address with a class C netmask, I can make it look like I'm using a class C with only 254 usable node addresses.

This is how the University subnets worked for a very long time. Schools and departments received a portion of the class B address (a three-byte network number) and used this in conjunction with a netmask of 255.255.255.0.

Thus, when the decision was made as to whether something was local or needed to go to the router, it was made based on the three leading bytes rather than the default two. This localised traffic into schools and reduced load on the router backbone.

A later development was the introduction of Variable Length Subnet Masks (VLSM) and Classless Inter-Domain Routing (CIDR). Rather than only being to mask based on whole bytes, you can use CIDR and VLSM to mask based on individual bits.

A common way of representing a netmask is to show it after the IP address as the number of bits which make up the mask. This is separated from the IP address by a "/" character.

Thus, IP address 158.125.1.208 with a netmask of 255.255.255.0 can be represented as 158.125.1.208/24.

With CIDR, we can use a number of bits rather than whole bytes. However, we still can't use fewer bits for a given class of address than are used in the default netmask. This is because every router in the world still expects you to respect the class boundaries, even if you start subnetting them later on.

Since you know that we have 4 bytes to play with, and therefore 32 bits, does that mean that it would be legal to specify a netmask as xxx.xxx.xxx.xxx/32?

No! This is a single host, but not a network! How about /31? We nominally have two possible node addresses (0 and 1). From the section on special IP addresses, these are already in use and can't be used for node addresses.

The largest netmask possible is /30. This leaves 2 bits on the end, giving us the node numbers 00, 01, 10 and 11. 00 and 11 are reserved - the /30 netmask only allows us to have two IP addresses in the subnet.

How is that useful? We'll talk about this later in network diagrams.

### 6.3.7 Calculating a netmask from slash notation.

If I've got an IP address with its netmask in slash notation, such as 158.125.1.1/30, how do I calculate how it looks in the dotted netmask (255.xxx.xxx.xxx) format?

The easiest way is to work out how many full bytes are in use and then add up the remaining bits for the last octet. (A byte is also called an octet when you're talking about IP addresses.)

For /30, we have three whole bytes (because that's /24) plus 6 bits. So the first three bytes will look this 255.255.255 (because 8 binary 1s in a row gives you 255). The last byte only has 6 bits set and, because we always fill in from the left, this gives us 128 + 64 + 32 + 16 + 8 + 4, which is 252. So the final netmask in dotted notation is 255.255.255.252.

You should really know the summed powers of two because it will make your life a lot easier when calculating netmasks. Why do you need to know both? Because a lot of ISPs provide your information in slash notation but Cisco routers need you to enter it in dotted notation.

## 6.4 Assigning an IPv4 Address to a Cisco interface

We're not going to be able to route IP packets anywhere unless we get some interfaces up and running with IP. Fortunately, this is very easy - we just have to assign a valid IP address and subnet mask and then switch the interface on. When an IP address is assigned to an operational interface, the router will enable this interface for IP routing.

Something that we've alluded to previously, but not stated explicitly, is that each interface should have an IP address in a different network. You can sometimes combine interfaces together to increase your bandwidth or to load balance, but in this case they look like one big interface. The router has to choose between different interfaces so that it can send the packet to the right network. You cannot configure the router so that two different interfaces have IP addresses in the same subnet - the router wouldn't be able to choose between them!

Let's say that we're going to use a Class C address with a /24 subnet mask. This IP address looks like this: 192.168.100.1/24. The network address is 192.168.100 and the node address is 1. The netmask for this, in dotted notation, is 255.255.255.0.

Log into the Unix machine, connect up the console, switch on minicom and boot the router. Once it has finished booting, log in and go to configuration mode.

We are going to configure the second FastEthernet interface on the router. This should be FastEthernet 0/1. We will assign the IP address and then bring it into an operational state.

```
1 LabA(config)# int fa0/1
2 LabA(config-if)#ip address 192.168.100.1 255.255.255.0
3 LabA(config-if)#no shutdown
```

You will probably get some messages on your screen telling you that the interface has been configured and has started up. You will also, unless you've plugged something into that port, get a message almost immediately afterwards saying that it's gone down again. We'll cover why this happens in a second.

Exit configuration mode and type **show interface fa0/1**. This will now show you the details of the configuration for this interface. It should also say that it is down. Why is it down? Interfaces shut themselves down when they don't detect anything connected. This makes sense because it stops the router from trying to send packets to a destination that can't be reached.

You can use **show interface** on any interface to see what's happening.

### 6.4.1 Why do I see "FastEthernet 0/1 is down, line protocol is down"

You see two different down statements because one refers to the physical connection and one refers to the data link layer protocols. The first 'down' tells you that there is a physical problem (Layer 1). The second 'down' tells you that there is no Layer 2 connectivity. You'll often see 'FastEthernet 0/1 is up, line protocol is down' when you have issued a **no shutdown** but haven't plugged in a cable.

### 6.4.2 What's "no shutdown"

This is a "cisco-ism". To shutdown an interface, you enter configuration mode, go to interface configuration mode and then type **shutdown**. Rather than have two commands, one positive and one negative, for each situation, Cisco provides the **no** command that provides the neg-

ative command. If we wanted to remove the IP address from the interface, we'd type **no ip address 10.5.1.5 255.255.255.0** after selecting the appropriate interface.

### 6.4.3 IP addressing for serial links

You set an IP address on any IP interface in the same way. However, for some WAN links, this isn't enough as there may be special requirements to connect to the network provider cloud. We'll come back to this later. What you need to remember at the moment is how to set an IP address.

### 6.4.4 IP addressing for HWICs

This gets a little more complicated. HWICs are the small 4-port (on 1-port) plug in modules on the router. These should be considered logically as a separate switch and can only be used if we configure a VLAN on the router side of that switch-router interface.

To create a VLAN issue the vlan command (in configure mode):

```
interface vlan <num>
```

Choose a number bigger than 1 and put this on your network diagram.

To add a port from an 4-port HWIC to a VLAN:

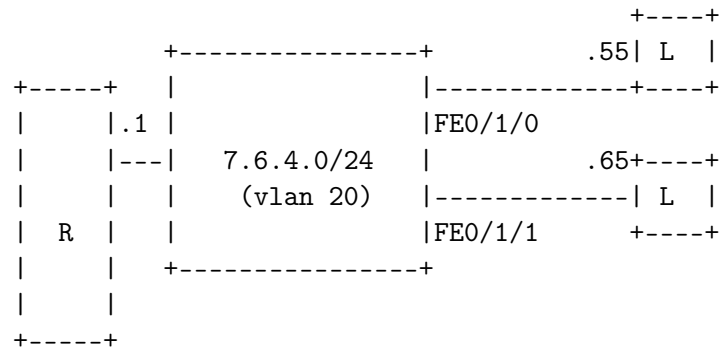
```
interface fastethernet 0/1/<port>
switchport mode access
switchport access vlan <num>
```

(The 2nd line above switches the port to `access` mode as opposed to `trunk` mode. More on this later. The 3rd line is the line that actually assigns the port to the specific vlan.)

To add an IP address to the router's side of the VLAN configure the VLAN as if it was an interface:

```
interface vlan <num>
ip address 7.3.4.5 255.255.255.0
```

You may wish to draw this something like this:



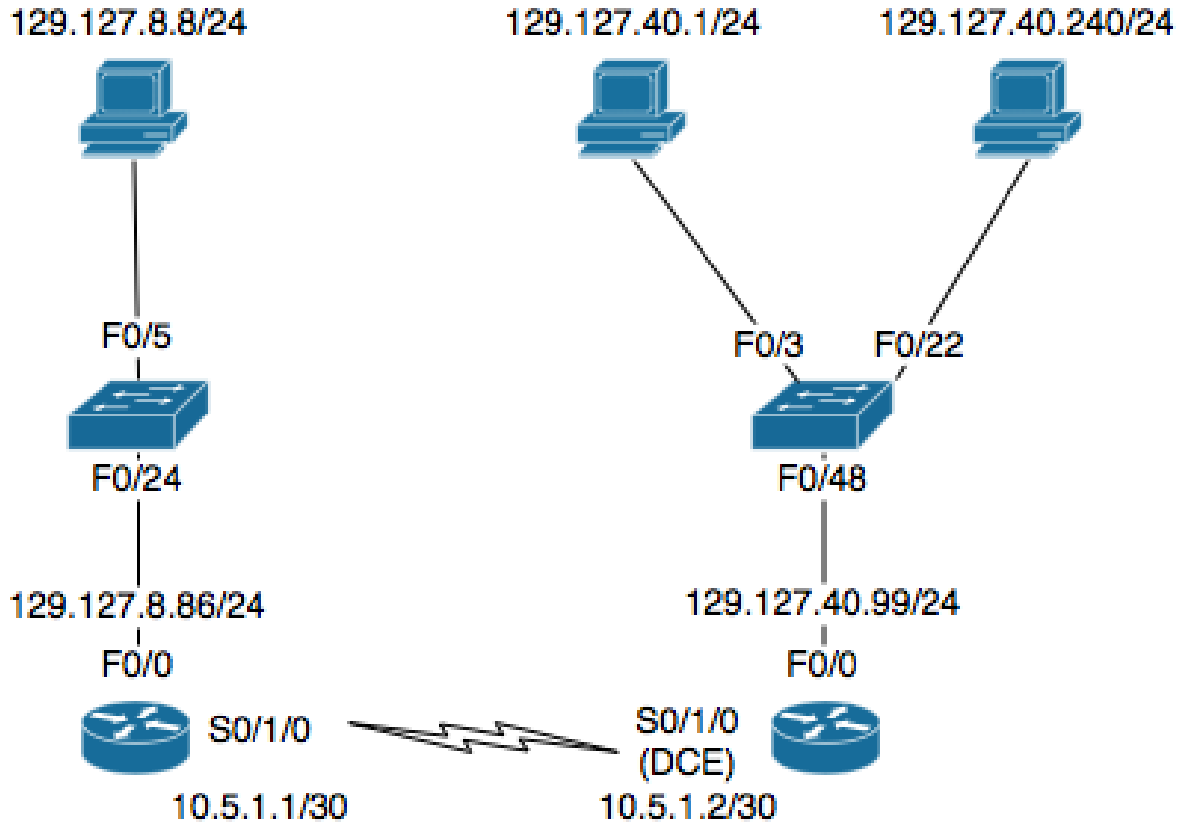

(20) is the VLAN id

Note that the config mode command `interface vlan 42` lets you change layer-3 related information about the VLAN, such as the IP address. Whereas `vlan 42` lets you change the layer-2 information, including `no shutdown`.

### 6.4.5 1-port HWICs

These act as normal router interfaces and can be configured as `interface fa 0/1/0`.

## 6.5 Understanding a network diagram




Network diagrams are designed to communicate all of the details you need to set up IP addressing in a network. You should be provided with an IP address and netmask for every device that needs to work at the Layer 3 level. Note that switches don't necessarily need an IP address. Most switches can be assigned an IP address so that you can configure them over the network but, since they work at Layer 2, all they need to be able to do is to read frames and pass them out the appropriate ports.

Look at the diagram above. You've already seen the symbols for routers and switches before. The additional symbol is for a "generic" piece of client computing equipment such as a PC, laptop, Mac or Unix server. There are specific Cisco symbols for different types of equipment but we won't be using them here.

You'll also see that interfaces are referred to with every link that goes into or out of a piece of network hardware. This is to show you where the cables are supposed to go. For example, the PC on the left is plugged into port 5 of the switch that, in turn, is connected to FastEthernet 0/0

on the router via the switch's port 24. You should always plug the cables into the listed ports because it is quite possible to get different behaviour depending upon which port you're using.

Have a closer look at the router symbol on the left. Note that we're using a shortened version of the interface names to clarify which interfaces have which IP addresses. Reading the diagram, which IP address is associated with the first port in the Serial WIC in slot 1?

This will vary from router to router. `show run` or `show int fa0/1/0` will show you.

You may remember that we briefly discussed the /30 network, which contains 2 valid host addresses, as being very useful. If you look at the IP addresses of the serial links which connect one router to another, you'll see that it's a /30 subnet.

Why is this good? Because we have to put every interface on the router into a different subnet, we want to make the best use of the subnets that we have available. If I had used the subnet 10.5.1/24 then I would have 254 possible node addresses. Unfortunately, the only other thing on this connection is another router. This is a point-to-point serial link so there will only ever be two IP addresses in use. In this case, we can use a /30 subnet mask and use a small fraction of the 10.5.1/24 space. We only need two IP addresses so we only ask for two. This means that I could use parts of this subnet elsewhere if I wanted to. If you're interested, look up VLSM in a textbook.

### 6.5.1 Exercise - network planning, network diagram and "show run"

To see how the interfaces are configured on your router, go to privileged exec mode and type **show run** (This command can be abbreviated **sh run**). Produce a network diagram for all routers in your rack based on the current configuration and draw it.

Make sure that your plan also includes your PCs/Laptops. They should be connected to different interfaces on different routers.

Plan for appropriate IP addresses on all interfaces.

Realise your plan by cabling according to your sketch. Configure the routers and PCs accordingly. You will notice that you will have to setup static routing to achieve connectivity.

```
ip route <dest_ip> <mask> <gateway_ip> <distance>
```

For example:

```
hostname(config)\# ip route 10.10.10.0 255.255.255.0 192.168.1.1
```

This enables you to add a static route.

The `dest_ip` and `mask` is the IP address for the destination network and the `gateway_ip` is the address of the next-hop router. The `distance` is the administrative distance for the route, this level of detail is not of importance to us here. Just accept the default, which is 1. For your information, the administrative distance is a parameter used to compare routes among different routing protocols. The default administrative distance for static routes is 1, giving it precedence over routes discovered by dynamic routing protocols but not directly connect routes. We will learn later that default administrative distance for routes discovered by ISIS is 110. If a static route has the same administrative distance as a dynamic route, the static routes take precedence. Connected routes always take precedence over static or dynamically discovered routes.

Use the master network diagram to assess this. Demonstrate that your network works, by downloading a webpage from a server that is at least two router-hops away.

## 6.6 Routing Basics

Even though you have marked interfaces as 'ready for routing' by giving them IP addresses, no routing will actually take place until you start using a routing protocol.

Remember that we have three different kinds of routing: static, dynamic and default. Because we are working towards a dynamic routing protocol, we're going to concentrate on dynamic routing.

Briefly, *default routing* is used when a router only has one interface connecting it to the WAN or other routers. This is referred to as a stub router. This is quite useful on occasion but we won't be dealing with it here.

As before, static routing allows us to tell the router which interfaces go to where. However, this requires us to be right when we set it up and to stay right. If our router uses two other routers to get to a network, and we set up a static route to tell our router where that network is, this only works while every other router in the chain can also get packets to the network. If a router somewhere in the cloud loses the connection, we will continue to send packets out the selected interface but we will get all of them back with an ICMP 'destination unreachable' message. Because this is a static route, we may never go looking for an alternative.

Networks are fairly dynamic but a lot of their fault tolerance and resilience comes from the use of dynamic routing protocols. Dynamic routing protocols use messages to work out who can get to which network and, importantly, what the cost of taking a particular route will be.

Routers can often choose between multiple routes. To do this they have to assess the *trustworthiness* of the route and the *metric* of the route. The *administrative distance* of a route indicates its trustworthiness. Networks that are directly connected to the router, via its own interfaces, have an administrative distance (AD) of 0. This will be used over any other possible route. The least trustworthy routes have an AD of 255 and will never be used.

The *metric* of a route will be different depending upon which routing protocol is in use. For the distance vector routing protocol we'll be looking at, a route is judged in terms of how many hops are needed to get to the target network. Each time a packet goes through a router, that's a hop. If two routes have the same AD, the one with the lowest metric value will be chosen.

Now you can start to see the advantages of dynamic routing. If a route goes bad and becomes unreachable, we have criteria for choosing a new route that will work better.

### 6.6.1 Distance Vector Routing Protocols

DV protocols pass complete routing tables out to all of its neighbouring routers. These then combine any received routing table entries with their own routing tables. Because the routers are essentially gossiping with each other to work out the routes, we often call this *routing by rumour*.

When two routes can be used to get to the same network, the AD is checked first and, if this is the same, the route with the lowest metric will be used.

### 6.6.2 Convergence

Routing table convergence occurs when all of the routers have shared information about which networks they know about. Once everyone has shared information and each router knows the best path to each network, the tables are converged. Routers start by only knowing about the

directly connected interfaces and, as more information comes in, add and modify routes to finally arrive at a converged table.

For example, Router C is advertising a route to 10.5.1/24 that it has received from Router B. The AD and metric for this route is [120/5]. Router D, neighbouring C, has a route to 10.5.1/24 which has a metric of [120/3]. Once C gets this advertisement, this becomes [120/4] to reflect the hop count-based metric has been incremented by going through another router. C will now learn this pathway to 10.5.1/24 and will send any packets for this network down the interface that connects it to D.

Eventually, all routers should have worked out the best routes to every other router and this learning process stops - until the network changes.

The protocols we're talking about today are Interior Gateway Protocols (IGPs) and are designed to be used with routers that are all under the same administrative control, or *autonomous system (AS)*. Obviously, it would be almost impossible to get global convergence. Instead, we get convergence inside our little group of routers and then use Exterior Gateway Protocols (EGPs) to connect ASs together. EGPs are beyond the scope of this project.

It's possible that, while routers are learning routes from each other, we can produce routing loops that block convergence. A complete discussion of this is beyond the scope of the course but we'll briefly discuss two of the possible ways to prevent loops.

Maximum Hop Count - If we only allow the metric to get to a certain point before it is considered infinite, we can't run off to infinity. As we'll see, RIPv1 permits a hop count up to 15. Anything beyond that is unreachable.

Split Horizon - This sounds far more complex than it is. Since we are routing by rumour, we listen to the interfaces to hear about routes advertised by a neighbouring router on that interface. Split horizon gets rid of loops by enforcing the rule that we cannot send routing information back down the interface that we learnt it from. If I get an advertisement for a network on my fa0/1 interface, I can send it out on any valid interfaces EXCEPT for fa0/1. I can only send my own networks out on every interface.

### 6.6.3 Routing Information Protocol (RIP)

RIP is a distance-vector routing protocol that sends out its complete routing table every 30 seconds. RIP uses hop count as its metric, with a maximum legal hop count of 15. It's reasonable to use RIP in small networks but it scales badly and is inefficient in both CPU resources and bandwidth in larger networks.

If you have a router which is advertising a route with a metric of 15, anyone who picks up this route will not be able to use it because the final metric will be 16 (15 + 1 hop for a new router = 16) and 16 is unreachable.

RIP only uses *classful routing*, which means that every device in the network must use the same subnet mask. This is bad in practice because it prevents us from using CIDR and VLSM but it's easier to configure.

## 6.7 Switching on routing in a Cisco Router

### 6.7.1 RIP

RIP, due to it's limitations, is not widely used anymore. However, it is pretty easy to setup.

```
Router(config)#router rip
```

Look at your running-config and find the IP addresses for the interfaces fa0/0. Based on the class of the address, work out the network number and pad out any missing octets with 0s. Then enter these network numbers using the **network** command as shown below. Replace xxx with the correct network number.

```
Router(config-router)#network xxx.0.0.0
Router(config-router)#network xxx.xxx.xxx.0
```

Use `CTRL-Z` to exit configuration mode and verify that you have added routes by using the `show ip route` command.

Before you verify the routes, you'll need to wait until at least one other adjacent router has been configured with either RIP or ISIS (you will learn more about ISIS in the lectures). **To switch routing off**, go into configuration mode and type:

```
Router(config)#no router rip
```

Exit configuration mode. All routing is now shut down on that router.

### 6.7.2 IS-IS

We will use ISIS routing for our IGP in the experiments. Do not start experimenting with dynamic routing until you have completely understood the static examples.

First of all you need to give each router its own loopback address. Note that most computers, e.g. laptops, desktops all use 127.0.0.1 as their loopback, routers do not have the same loopback address. To set a loopback address then you need to add an address to `interface loopback 0`

```
Router(config)# interface loopback 0
Router(config-if)# ip address X.X.X.X Y.Y.Y.Y
Router(config-if)# ipv6 address Z:Z:Z:Z:Z:Z:Z:Z/N
```

Usually loopback addresses from /32s or /128s for v6.

Now to continue to setup ISIS you need to create the NSAP (Network Service Access Point address). This is formed according to the following standard:

- Starts with AFI (49)
- Then area ID (0001)
- System ID (set it to loopback munged as follows):
  - If loopback is `ABC.DEF.GHI.JKL` (with leading zeros in place)
  - System id is `ABCD.EFGH.IJKL`

- End with 00.

For example for loopback address `7.7.7.1` the NSAP would be `49.0001.0070.0700.7001.00`. Note that if all loopback IP addresses are unique then all NSAP will be unique too. This is critical to the operation of ISIS.

To start ISIS and set the NSAP. In our experiments we will stick to a single level ISIS.

```
Router(config)# router isis
Router(config)# net XX.XXXX.XXXX.XXXX.XX
Router(config)# is-type level-1
```

Finally we indicate which interfaces are to take part in ISIS routing and which versions of IP.

```
Router(config-if)# ip router isis
Router(config-if)# ipv6 router isis
```

When ISIS is running the following commands are all useful to see what is happening:

```
  sh isis neighbors
  sh isis top
  sh isis database
  sh ip route
  sh ipv6 route
```

## 6.8 Domain lookup

When you type a word into a router that iOS doesn't recognise, it assumes this is a computer name (from DNS) and tries to resolve it to an address so that it can remote login. This can be annoying if you don't have a DNS setup, so can be turned off with:

```
Router(config)#no ip domain lookup
```

## 6.9 Security & Remote access

Accessing routers via a serial interface can become tedious, not least because it often takes a few attempts to get the hardware to work. It's also slow and if the routers are distributed around a campus requires walking to remote locations. Given we have ethernet interfaces at 100Mbps or greater it makes sense to use these. To do this we need to enable remote access. This requires setting up some usernames and passwords.

1) Setting Router's hostname:

```
router(config)#hostname R1
```

2) Setting domain name (This has to be done, even though you can't, at this stage, use it)

```
R1(config)#ip domain name dt.lboro
```

3) Username & password for *line* (1st) authentication:

```
R1(config)#username USER password PASS
```

4) *Enable password* (2nd) authentication:

```
R1(config)#enable password ENPASS
```

5) Entering line vty configuration mode: A line is a logical interface through which we telnet/ssh to the router, means router will accept 5 simultaneous sessions (numbered 0 to 4).

```
R1(config)#line vty 0 4
```

6) Allowing ssh and telnet on vty:

```
R1(config-line)#transport input ssh telnet
```

7) Telling the router to use locally defined username/password (set in step3) to authenticate telnet/ssh sessions (the alternative is a centralised lookup service):

```
R1(config-line)#login local
```

8) Extra configurations required only for ssh. Setting the SSH version to 2:

```
R1(config)#ip ssh version 2
```

9) To generate rsa (ssh) pair of public/private keys:

```
R1(config)#crypto key generate rsa general-keys
.
.
.
How many bits in the modulus [512]: 1024
```

(It must be 1024 otherwise version 2 doesn't work)

10) Finally increase the minimum key size otherwise modern ssh clients don't work.

```
R1(config)#ip ssh dh min size 4096
```

11) to use it:

```
    ssh -l username remote-ip-addr
```

## 6.10 BGP

To enable BGP you need an AS number. Enter this with

```
R1(config)# router bgp <asn>
R1(config-router)#
```

When in `config-router` mode you can add neighbours with the following command:

```
R1(config)# router bgp <asn>
R1(config-router)# neighbor 110.110.10.1 remote-as 101
```

Note the spelling of neighbor and also that if the remote AS specified is your AS, then this will be an IBGP neighbor.

For IBGP routes it makes sense to make the source (and destination) of the IBGP messages loop back addresses. To change the source address use the following:

```
R1(config-router)# neighbor 110.110.10.1 update-source loopback0
```

If you do this for EBGP, then make sure that the loopback address of your external gateway is in BGP.

Any prefixes you wish to be exchanged over BGP need to be added with the `network` command:

```
R1(config)# router bgp <asn>
R1(config-router)# network 10.10.10.0 mask 255.255.255.0
```

On cisco networks are only added to BGP if they exist on a interface on your router. It is simplest to add all your routed prefixes at this stage.

BGP can take some time to converge, so monitor your routing tables and BGP information with:

```
R1# sh ip route
R1# sh bgp
```

## 6.11 Passive Interfaces

When an interface is part of a routing protocol this means two things:

1. Their subnets are distributed by the protocol.
2. Routing protocol messages are sent from this interface.

Sometimes (think when) you do not want the latter of these to happen. To do this then you need to set the interfaces to passive:

```
R1(config)# router isis
R1(config-isis)# passive-interface <interface id>
```

## 6.12 Saving configurations

When things are working it's often useful to save configurations off the routers. This is especially important as sometimes routers break and need replacing. There are various places to save configs, including: plug in USB flash drives and over the network. For network config saving, you need to create a tftp server on one of your laptops. To do this install the !tftpd-hpa! package with `apt-get`. By default, and the default is good, this receives files and saves them to the `/var/lib/tftpboot` folder on that laptop. To save a file you need to already have a file in that folder with the name you want to use (you can create this with `touch <filename>` and it needs to be writable by all users (`chmod`). Then:

```
R1# copy running-config tftp://<laptop-ip>/<laptop-filename>
```

will copy the file over. This is readable with `less` on the laptop.

## 6.13 Testing and Troubleshooting

`show ip route` is great if everything is working. What if all you have are C lines in your routing table? We'll go through this methodically.

1. First make sure that all of your interfaces are up and working correctly. If you type **sh protocol** in enable mode, you will see if routing has been turned on and whether your interfaces have both layer 1 and layer 2 connectivity.

   (Here's a tip, if the interface is completely down - check the cable is plugged in to the right thing. If there are status lights, see if they are green.)

   `sh ip protocol` should also show you what your IP addresses are on each interface. It's much easier to read than wading through `sh running`.

2. `show ip protocols` will show you what RIP is doing. Check that it is handing out information for the correct networks.

If all of this looks right, then you need to check your connectivity to the other routers. The first two steps should have eliminated any wrong-doing on your part. Now we have to check your peers.

Go to enable mode and try to contact one of the other routers. Use the **ping** command and type:

```
1 Router\#ping 192.168.10.3
```

but use the correct IP address. Ping should return 5 exclamation marks to tell you that it can get through (!!!!!). Sometimes the first one is not there as the router is awaiting an ARP timeout, but anything else means that something somewhere has gone wrong.

Things to check:

- Your IP addresses are set correctly - remember that you must be in the same subnet!
- The cable is working.
- The other router has also been configured with RIP and is working.

# Bibliography

[1] https://www.cisco.com/c/en/us/support/routers/2800-series-integrated-services-routers-isr/tsd-products-support-series-home.html

[2] https://www.cisco.com/c/en/us/support/routers/2900-series-integrated-services-routers-isr/tsd-products-support-series-home.html