

SYS2 : NET Laboratory 1 : The Internet and Networks

These weekly lab scripts have been written as a self-study resource i.e. an activity to be worked on at your own pace, in your own time. The timetabled practical sessions are an opportunity to get advice and guidance. Therefore, you may not always finish each lab during the timetabled session, but do finish each lab's tasks in your own time. Hardware labs are open Mon-Fri, 9:00-17:00.

The aim of this lab is to introduce some of the concepts and terminology used in regards to the Internet and the network equipment that we shall be using this term. Before starting this lab make sure you have watched video **Lecture 1**. This introduction is made slightly more complex as we have not yet discussed any of the technology which the Internet is built upon. Therefore, apologies in advance we will be skimming over a number of different concepts in this lab. However, I want to start to introduce some of these key concepts and terminology used through the use of a few hands-on examples. Trust me the nuts and bolts will follow :). At the end of this practical you will understand how to:

- Determine network latency, round trip time and bandwidth.
- Identify the paths (routes) taken by network packets when connecting to a host.
- Connect to the Raspberry Pi test system e.g. via a terminal or remote desktop.

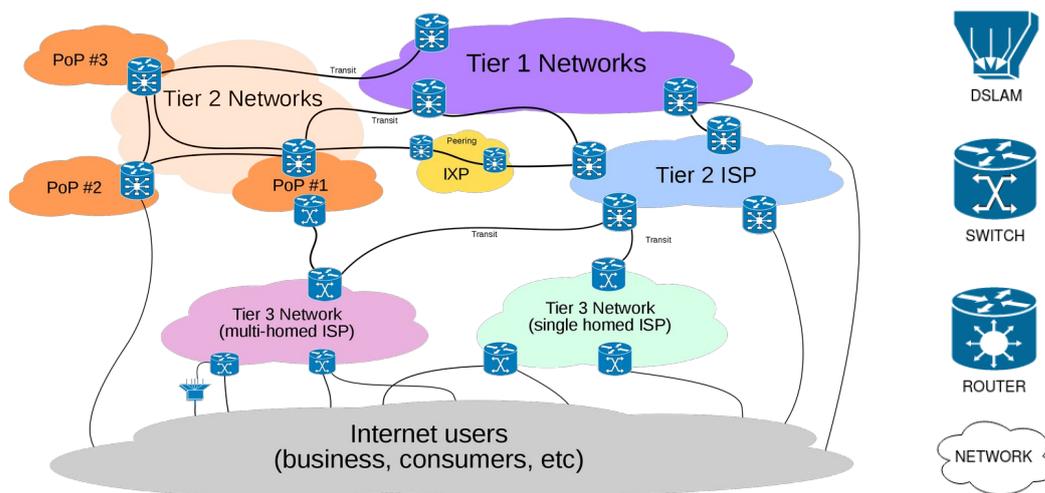


Figure 1 : The Internet, a network of networks

To start our journey into networking lets first consider what the Internet is, so referring to the fount of all knowledge:

“The Internet (or internet) is the global system of interconnected computer networks that uses the Internet protocol suite (TCP/IP) to communicate between networks and devices. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing.”

<https://en.wikipedia.org/wiki/Internet>

Point 1: the first key point to pick out here is that we have a communicating network of computers, using a network of networks (different Tiers) to pass information between themselves, as shown in figure 1.

Note, for me the most complex part of the Internet is how traffic is routed between these different networks (Tiers) i.e. how does data from one computer (host) get to the correct destination. For the moment we will gloss over that problem, but we shall return to this in later lectures / practicals.

This collection of communication paths are typically based on the packet switching model rather than fixed private circuits (wires), as used in a classic telephone network. A key characteristic we need to consider when communicating across this network is latency. That is the time it takes a message to travel from one host to another. This can be expressed by the following equations:

$$\text{Latency (one-way delay)} = \text{Propagation delay} + \text{Transmission delay} + \text{Queueing delay}$$

$$\text{Propagation delay (in a vacuum)} = \text{Distance} \div \text{Speed of light}$$

$$\text{Transmission delay} = \text{Data Size} / \text{Bandwidth}$$

$$\text{Queueing delay} \propto \text{how busy the network is}$$

Figure 2 : network latency and bandwidth calculations

A network's bandwidth is measured in bits per second (bps), the maximum amount of data that can be transmitted over a network in a given time. It represents the capacity of the network i.e. a measure of the volume of data that can be sent, whereas a network's speed relates to how fast individual packets travel, again measured in bps :(.

Point 2: the second point to pick out of our original definition of what the Internet is, is that we will be communicating using the Internet protocol suite (TCP/IP), as shown in figure 3. This uses a layered model, a protocol stack, to allow data to be communicated across a network.

Application	5 : Network processes and user applications
Transport	4 : Host-to-host connectivity and reliability
Network	3 : Path determination and logical addressing
Link	2 : Physical addressing
Physical	1 : Communications media, signals and hardware

Figure 3 : Internet protocol suite

Note, the reason why we have a protocol stack rather than a single communications protocol is to manage complexity. The problems faced at the physical layer e.g. the electrical interface and those

at the application layer e.g. email, are fundamentally different problems. Therefore, just like any OO software design we separate design concerns, with each layer focusing on one thing and optimised for that one task. However, just like any real world OO software design problem, sometimes there is information leaks between layers :).

Finally, the network of networks that form the Internet can be classified by their size, as shown in figure 4. For more information on these refer to:

https://en.wikipedia.org/wiki/Wide_area_network

- Wide Area Network (WAN)
- Radio Access Network (RAN)
- Metropolitan Area Network (MAN)
- Campus Area Network (CAN)
- Local Area Network (LAN)
- Personal Area Network (PAN)
- Body Area Network (BAN)
- Nanoscale

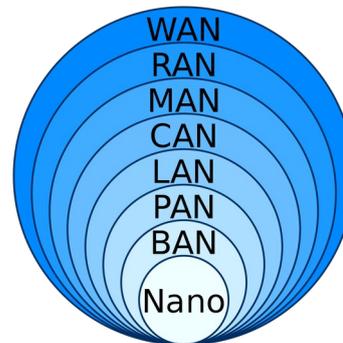


Figure 4 : Network classifications

Task 1

To get things started we are going to consider some of the primary requirements and characteristics of any network. Firstly, the ability to identify hosts connected to that network, the network layer, layer 3 and the Internet protocol (IP).

The internet protocol (RFC 791: <https://tools.ietf.org/html/rfc791>) was created 1970s and is still the fundamental protocol on top of which most other protocols are based. Network interfaces on each host connected to the Internet are assigned a globally unique 32bit IP address, typically represented using the “dot notation” shown in figure 5.

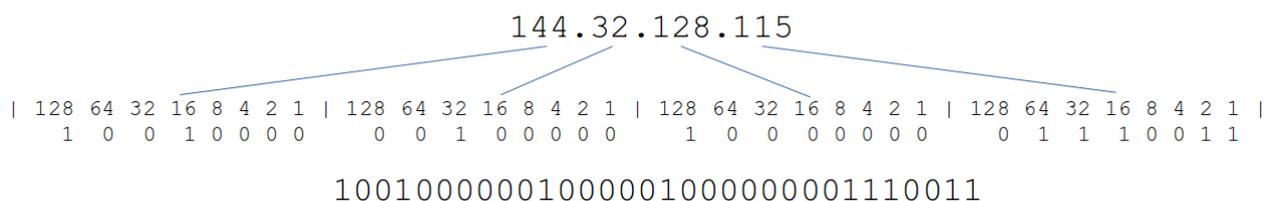


Figure 5 : Internet protocol v4 address

This breaks the 32bit address down into four bytes, then converts each byte into an 8bit decimal value i.e. IP addresses can range from all 0s (0.0.0.0) to all 1s (255.255.255.255), in theory giving 2^{32} (4294967296) available addresses.

IMPORTANT : remember we may use the dot notation but an IP address is just a 32bit number.

Note, with the explosion of devices needing to connect to the Internet we have run out of IPv4 addresses i.e. the 2^{32} available addresses. Therefore, we are starting to move over to Internet Protocol v6 (RFC 2460: <https://tools.ietf.org/html/rfc2460>) which uses an 128bit address,

represented using an eight, four digit hexadecimal notation. IPv4 is a “dead” protocol, but we still live in the land of IPv4 for a little longer :).

To identify the IP address of the Windows PC you are using you can use the `ipconfig` command, click on the start button and select the command prompt.

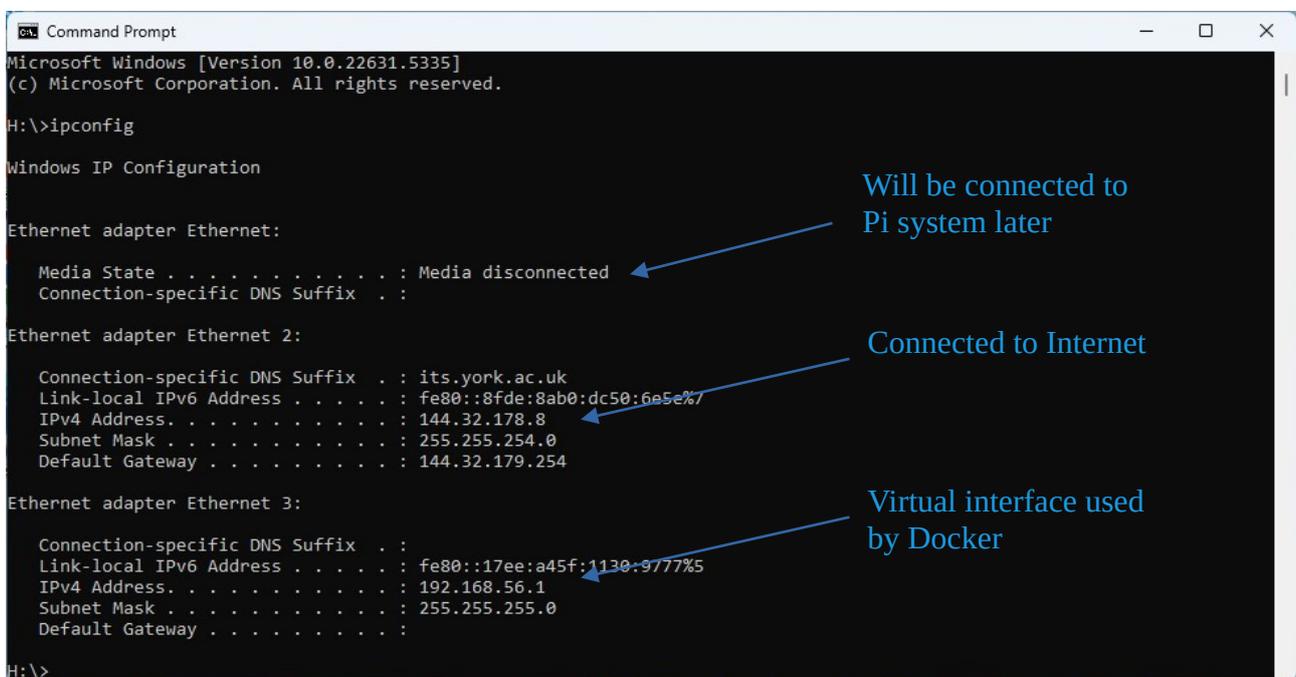


-> Command Prompt

In the command prompt enter the command below and then press the return  key.

```
ipconfig
```

Depending on what PC you are using it may have one or more Network Interface Controllers (NIC). On the lab PCs the network interface that can access the Internet is the one with an IP address starting with 144.32.X.X, as shown in figure 6.



```

Microsoft Windows [Version 10.0.22631.5335]
(c) Microsoft Corporation. All rights reserved.

H:\>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix . : its.york.ac.uk
    Link-local IPv6 Address . . . . . : fe80::8fde:8ab0:dc50:6e5e%1
    IPv4 Address. . . . . : 144.32.178.8
    Subnet Mask . . . . . : 255.255.254.0
    Default Gateway . . . . . : 144.32.179.254

Ethernet adapter Ethernet 3:

    Connection-specific DNS Suffix . :
    Link-local IPv6 Address . . . . . : fe80::17ee:a45f:1130:9777%5
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

H:\>
  
```

Annotations in the image:

- Blue arrow pointing to "Media disconnected" for Ethernet adapter Ethernet: **Will be connected to Pi system later**
- Blue arrow pointing to "IPv4 Address" for Ethernet adapter Ethernet 2: **Connected to Internet**
- Blue arrow pointing to "Link-local IPv6 Address" for Ethernet adapter Ethernet 3: **Virtual interface used by Docker**

Figure 6 : Example Ipconfig command

Note, in Linux you would use the commands : `ifconfig` or `ip addr`.

A useful command-line tool to test if a host is connected to a network is `ping`. The name comes from sonar terminology, where you send out a pulse of sound and listen for the reflected echo from an underwater object. In a network we send out a request for acknowledgement packet from one host to another and measure the round trip time (RTT) i.e. the time it takes for a packets to travel from your machine across the network to the target host, then for it to process this packet and transmit back an acknowledgement. Hosts do not need to respond to a ping request, its depends on the OS, firewalls etc, so just because a host doesn't respond to a ping doesn't mean its not there :).

To use the `ping` command, click on the start button and select the command prompt.



In the command prompt enter the command below and then press the return  key.

```
ping 144.32.128.40
```

This will cause your machine to transmit four “request for acknowledgement” packets, measure their round trip times (RTT), then print a summary, as shown in figure 7.

```
H:\>ping 144.32.128.40

Pinging 144.32.128.40 with 32 bytes of data:
Reply from 144.32.128.40: bytes=32 time<1ms TTL=252

Ping statistics for 144.32.128.40:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Figure 7 : Ping command

Note, as we are pinging a local machine (on campus) the round trip delay is very small i.e. less than 1ms so it is rounded down to 0ms. We can also run the same command in Linux, doing this we get a little more resolution, as shown in Appendix A.

Remember the ping command returns RTT not latency. Latency is the delay in one direction. Also, the outbound delays and the inbound delays may be different i.e. packets can take different paths, as shown in figure 8, when going to and from the destination host. RTT is the sum of these latencies.

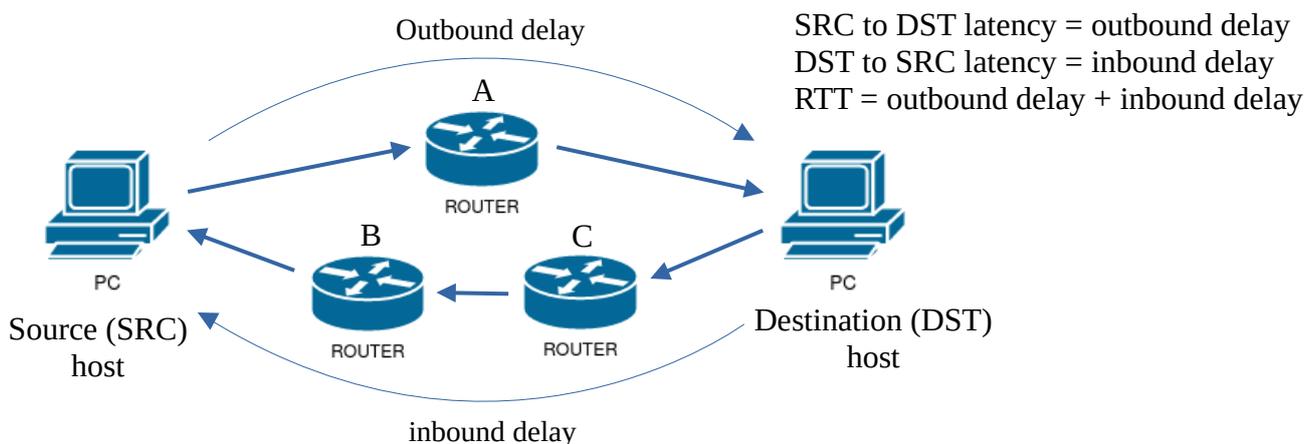


Figure 8 : Latency vs RTT

For more info on the ping command refer to :

<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ping>

Note, we will be looking at the protocols used by the ping command i.e. the ICMP protocol, in a lot more detail in later labs.

Task : ping a few of the IP addresses listed in figure 9 e.g. ping London and Japan, can you see how the RTT varies with distance?



Location	Webpage	IP address	Distance
UK (York)	https://www.cs.york.ac.uk	144.32.128.40	0 km
UK (London)	https://www.imperial.ac.uk	146.179.42.148	337 km
Germany	https://www.mi.fu-berlin.de	160.45.117.199	1,419 km
Italy	https://www.unitn.it/	40.119.148.37	1,723 km
Argentina	https://lafhis.dc.uba.ar/home	157.92.26.93	11,267 km
USA (Alaska)	https://www.cs.uaf.edu	137.229.25.25	6,594 km
USA (Massachusetts)	https://www.eecs.mit.edu	35.231.163.5	5,137 km
Russia	https://cs.msu.ru/en	188.44.50.110	5,649 km
Australia	https://cis.unimelb.edu.au	202.9.95.188	15,124 km
Japan	https://www.s-im.t.kyoto-u.ac.jp	133.3.140.24	9,252 km

Figure 9 : CS department IP addresses

As previously highlighted a host does not have to respond to “request for acknowledgement” packets used by the ping command. The IP addresses in figure 9 were tested at the time this script was written, however, the targeted host IP address may have changed. A table of typical ping times for different countries can be found here:

<https://wondernetwork.com/pings/London>

Task 2

Another useful command line tool is `traceroute`, this command allows you to see each hop a packets makes on its journey across the Internet to the destination host i.e. the different routers / networks the packet travels across.

Note, a router is a machine that connects different networks together e.g. the different Tiers shown in figure 1. When it receives a packet it uses its routing table to determine which network it should pass the packet onto. We will be looking at its internals in later lectures and practical :).

The `traceroute` command tracks a packet across the Internet using the Time To Live (TTL) field, as displayed by the ping command shown in figure 7. This field is used in a number of different protocols to ensure that “lost” / unrouted packets are removed from the network. A transmitting host will set this value to the maximum number of “hops” it expects a packet to take e.g. typical values are 255, 128 and 64. The selected value determines how quickly a “lost” packet is removed, as each time a packet is passed through a router to a different network this value is decremented by 1. When this value reaches 0 the routing device sends back an error message “Time exceeded” to the transmitting host and the packet is deleted i.e. removed from the network.

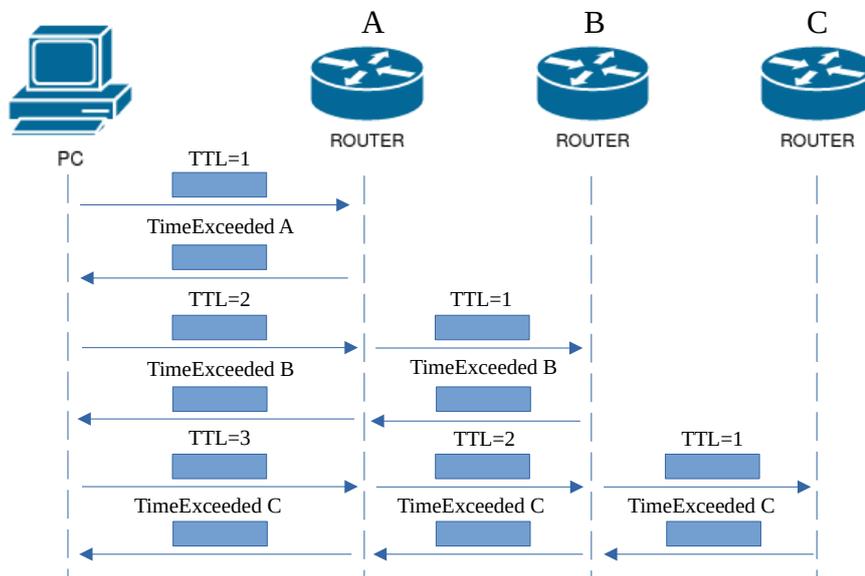


Figure 10 : trace route packet sequence

The `traceroute` command sends out a series of packets, incrementing the TTL field each time i.e. 1,2,3 ..., as shown in figure 10. It then records who sends back a “Time exceeded” error message. This allows it to identify what routers are used to route this packet across the network.

However, just like the `ping` command, routers do not have to send back an error message. In these cases the `tracert` program times-out and displays a “*”, it then increments the TTL field and tries to identify the next router on this path, or the final destination host.

Note, the default max number of hops is set to 30 i.e. if the `tracert` program does not reach the specified destination host with a TTL value of 30 it stops.

In Windows the command is `tracert` and in Linux its `traceroute`. For more information on the `tracert` command refer to :

<https://en.wikipedia.org/wiki/Traceroute>

To use the `tracert` command, click on the start button and select the command prompt.



-> Command Prompt

In the command prompt enter the command below and then press the return  key.

```
tracert 144.32.128.40
```

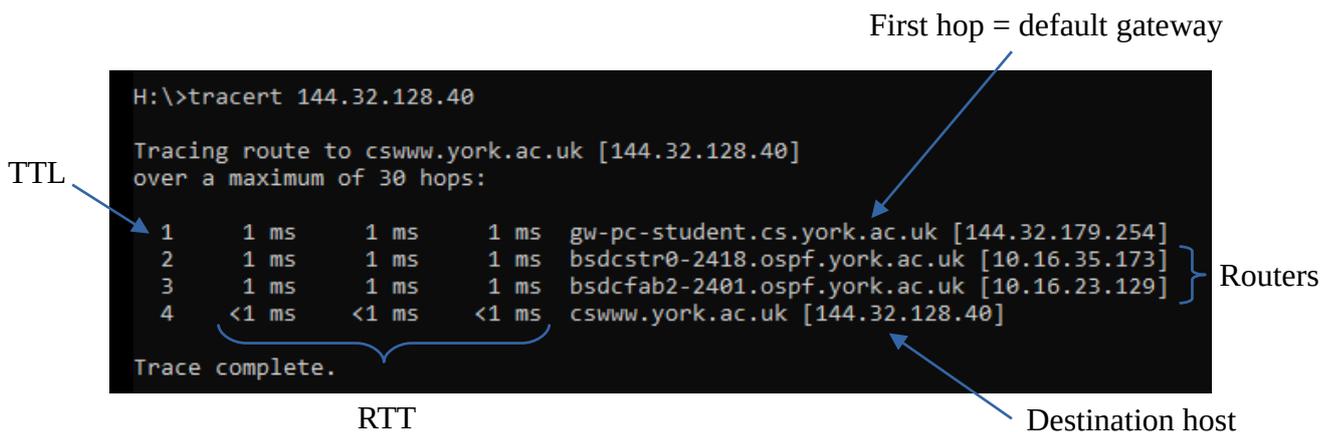


Figure 11 : Traceroute command

The first hop is to the default gateway (GW). This defines the router the PC should send packets to when it does not know where to send them i.e. when the destination host is not on its local network. How the following routers then work out where to send this packet is a discussion for a later lectures, here be dragons :). The default gateway in the lab (144.32.179.254) and can also be identified using the `ipconfig` command shown in figure 6.

Note, `tracert` sends out three packets to test each hop, the round trip time (delay) is displayed in a column with the returning routers IP address (just one if the same, multiple if different routes taken). If the router does not return back a “Time exceeded” error message a time-out occurs and a “*” is displayed.

Task : perform the `tracert` command for the MIT address listed in figure 9. Can you identify the location of each hop from the router’s name e.g. `ae26.manckh-sbr2.ja.net`? Where do you

think this router is located? Hint, `ja.net` = Janet, the Joint Academic Network in the UK.

Note, in addition to inferring a routers location from its name there are also lots of different web based tools to help “locate” a machine via its IP address. Using the website below enter your PC's IP address. How good is it at finding your location?

<https://dnschecker.org/ip-location.php>

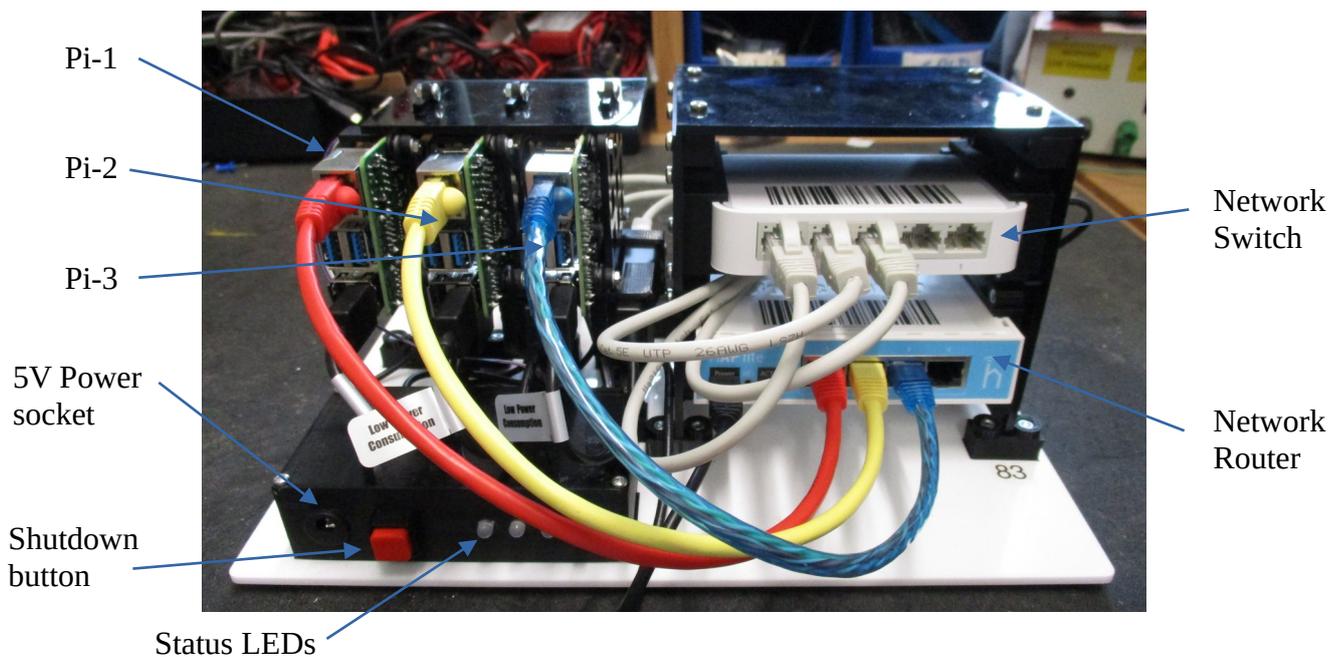


Figure 12 : Raspberry Pi test rig

Task 3

To allow you to develop and test your own networks we shall be using the Raspberry Pi test rig shown in figure 12, thus avoiding any possible “issues” on the department’s network :). This system is connected to its own private network. Each Raspberry Pi has four network interfaces:

- wlan0 : wireless network interface
- eth0 : on-board Ethernet interface
- eth1 : USB adapter Ethernet interface
- loop : virtual network interface i.e. network traffic on this interface is processed internally by the operating systems and not passed onto the physical network.

Typically each Network Interface Controller (NIC) is assigned one IP address, however, we can assign multiple IP addresses to each NIC to produce some more “interesting” networks :).

Note, the network cables connected to the router i.e. **RED**, **YELLOW** and **BLUE** cables, must be connected as shown in figure 12, otherwise the system will not work. The network cables connected to the switch i.e. the **GREY** cables, can be connected in any order or position.

The NICs in this Raspberry Pi test rig are connected together using a network switch and router, the

default configuration can be represented as a block diagram as shown in figure 13. In later labs we will be connecting this cluster of Raspberry Pis to your neighbours to form a larger cluster, or to the lab's private network, allowing you to connect to additional remote servers.

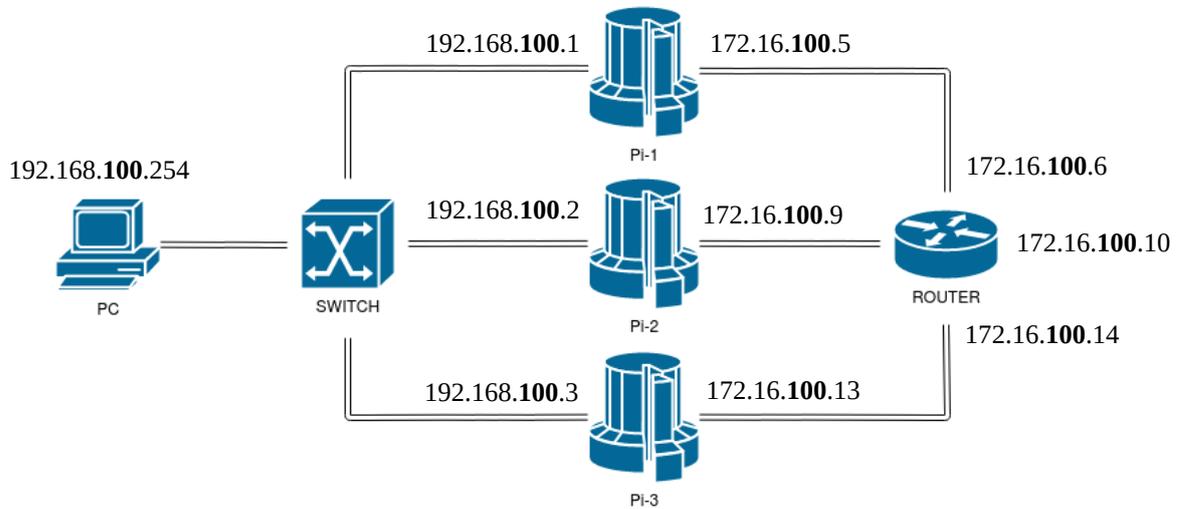


Figure 13 : Example Raspberry Pi network configuration for desk **100**

Note, technically the icons used for the Raspberry Pi in figure 13 are for a super-computer i.e. the Cray-1 shown in Appendix B. However, when viewed from above it resembles a Pie chart, so a slice of Pie, perhaps a slice of Raspberry Pi :).

The Raspberry Pi test rigs are stored in a plastic boxes in the front and rear cabinets in CSE168/9. When needed, do help yourself to one of these systems. Ensure you get the box with the **same** number as the desk you are using. You can identify the desk number by the sticker next to the monitor. The matching box number is on the end of the box and on the bottom plate of the test rig, as shown in figure 14.



Box number



Number location on Test rig



Desk number

Figure 14 : Raspberry Pi test rig box for desk 45

Remove the test system from the box. The system's power supply shown in figure 15 is available from your desks draw, or boxes located on the side benches. Connect the power supply as shown, then the power supply into a mains socket at the back of the desk.

IMPORTANT, do not use any other power supply i.e. it must be a +5V, other power supplies in the

lab will damage the Raspberry Pi and the network equipment. Remember once the magic smoke escapes there is no going back :), soooo check, check and check again.



Power supply



Power sockets



Booting



Pi-1 ready



Pi-1 + Pi-2 ready



Pi-1 + Pi-2 + Pi-3 ready

Figure 15 : Raspberry Pi power supply. ONLY USE THIS POWER SUPPLY

When powered the status LEDs on the front box will be **ORANGE**. These will turn **GREEN** when its associated Pi has finished booting, as shown in figure 15. To connect the Raspberry Pi system to the host PC use the **GREY** Ethernet cable at the rear of the desk and plug this into the network switch, as shown in figure 16.

Note, do not use any other coloured Ethernet cable, it must be the Ethernet cable connected to the PC's secondary Ethernet port. This cable can be plugged into any free network socket on the switch. When inserting this cable you should hear a definite "click" as the top clip latches into place. When removing this cable do make sure you push down this clip before trying to pull out the cable.

To identify the IP address of the NIC on the PC that is connected to the Raspberry Pi test rig we can again use the `ipconfig` command, click on the start button and select the command prompt.



-> Command Prompt

In the command prompt enter the command below and then press the return  key.

ipconfig

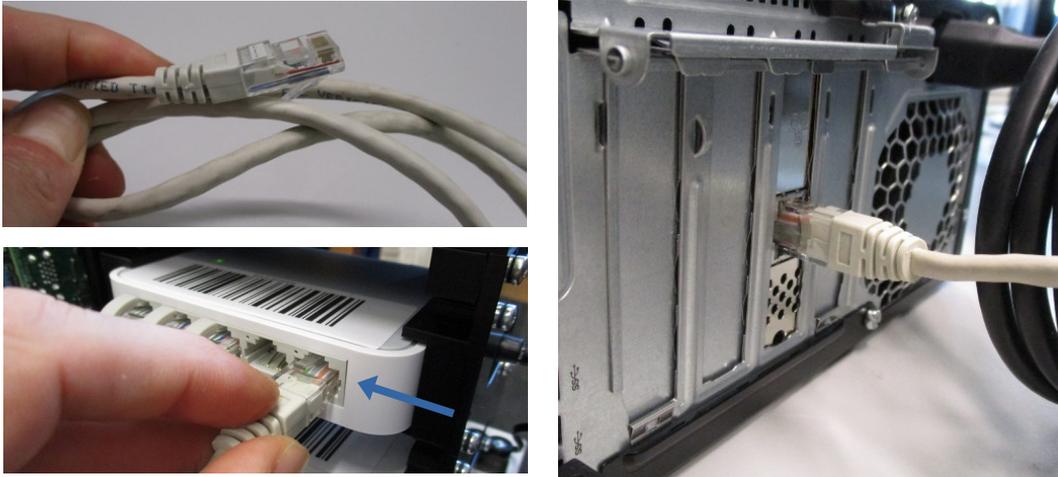
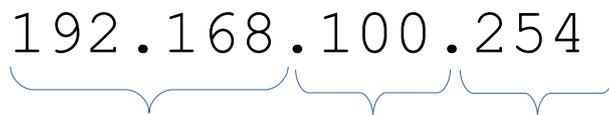


Figure 16 : Ethernet cable

```
Ethernet adapter Ethernet:
Connection-specific DNS Suffix . . . :
Link-local IPv6 Address . . . . . : fe80::802f:38ac:4a:3df1%16
IPv4 Address. . . . . : 192.168.100.254
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
```

Figure 17 : Example ipconfig command output

The networking interface that can access the Raspberry Pi is the one with an IP address starting with 192.168.X.254, as shown in figure 17 i.e. for this PC its IP address is 192.168.100.254.



Private IP address “Network/Box ID” Machine ID

Figure 18 : IP address breakdown

Note, if the 192.168.DESK.254 address is not listed or it has been assigned an IP address starting 169.X.X.X, sorry you will need to reboot the PC :(Also don't confuse this physical interface with the virtual Docker interface i.e. the docker interface also starts with 192.

The PC's IP address can be broken down into specific fields, as shown in figure 18. The first two numbers indicate that this is a private network IP address i.e. 192.168.X.X. This "starting" digits are reserved for private networks and can not be used on the Internet i.e. are not Internet routable. The third number indicates the specific network / location, given by the desk / box number i.e. all hosts on desk 100 will start with the IP address 192.168.100.X. The final number is used to identify the particular machine (host) connected to this network. Therefore, on desk 100 Pi-1 is assigned the address 192.168.100.1, Pi-2 the address 192.168.100.2, Pi-3 the address 192.168.100.3, and finally PC in the address 192.168.100.254.

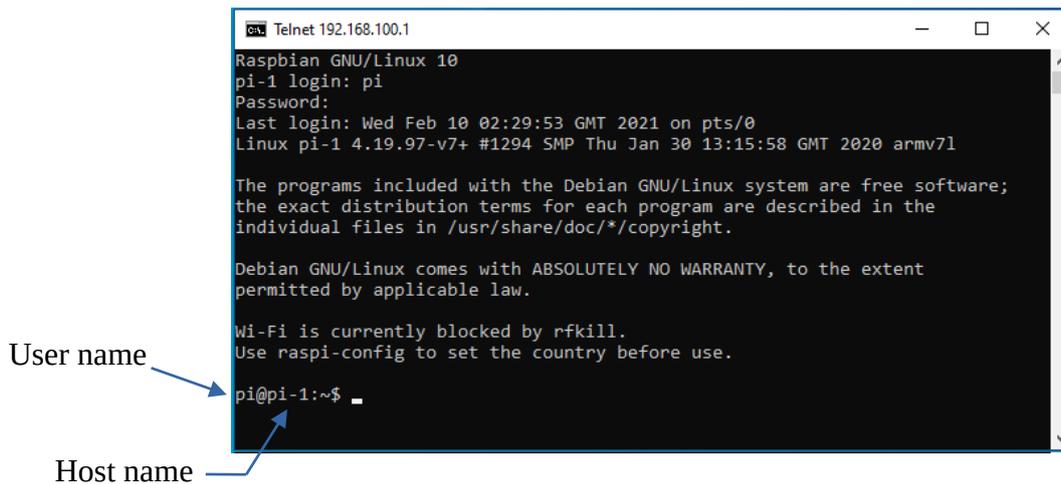


Figure 19 : Telnet

To open a terminal on these remote computers i.e. a command prompt on each Pi, we can use either the `telnet` or `ssh` commands. To start a Telnet session on Pi-1 click on the start button and select the command prompt.



-> Command Prompt

In the command prompt enter the command below and then press the return  key.

```
telnet 192.168.X.1          (IP address will vary, X=Box/Desk)
```

At the login prompt enter the following user-name and password, as shown in figure 19.

```
user-name : pi
password  : 12345          (secret special password :)
```

Note, for security reasons the password characters are not displayed on the screen as you type.

This will open a command terminal or shell on the remote machine i.e. this terminal is running commands on the Pi not the PC. The OS on the Raspberry Pi is a version of Linux. In this OS to list the network interfaces on a machine enter the command below and press the return  key.

```
ifconfig -a
```

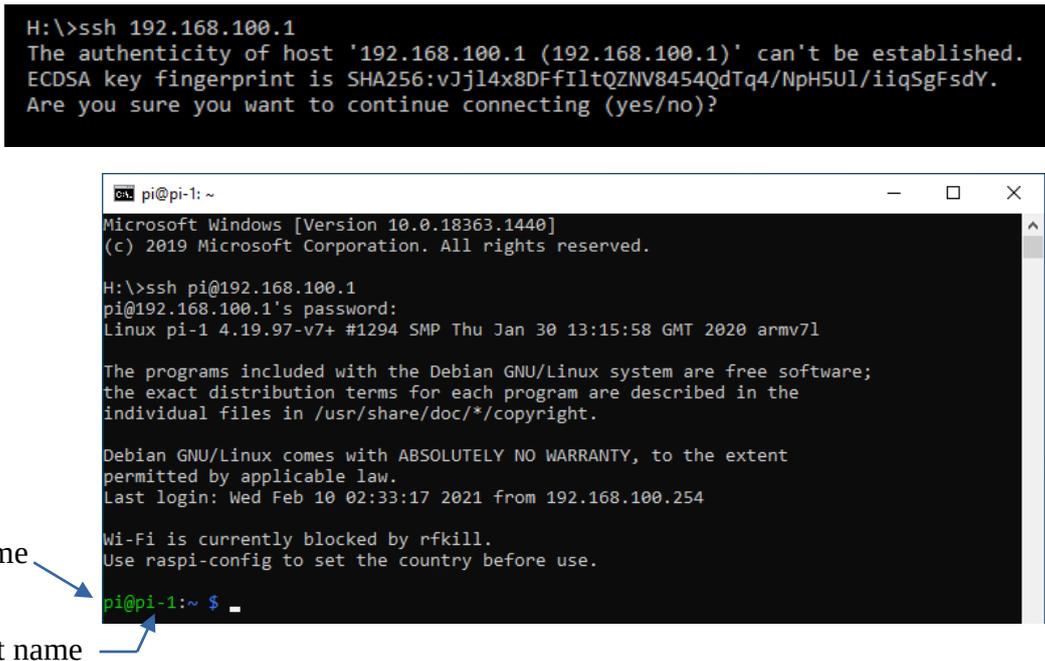
Task : run the `ifconfig` command on Pi-1. This will list the NICs on the remote machine and

their assigned IP addresses. How many network interfaces does Pi-1 have, what are their names and what IP addresses has been assigned to them?

Hints, interfaces listed will be similar to those shown in figure 13. In total you should see six NICs. Some are virtual / internal i.e. only accessible on that Pi. Some are external i.e. connected to physical hardware / cables. Can you spot which is which?

To exit this remote command terminal enter the command below and then press the return  key.

exit



```
H:\>ssh 192.168.100.1
The authenticity of host '192.168.100.1 (192.168.100.1)' can't be established.
ECDSA key fingerprint is SHA256:vJj14x8DFfIltQZNV8454QdTq4/NpH5U1/iiqSgFsdY.
Are you sure you want to continue connecting (yes/no)?

pi@pi-1: ~
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

H:\>ssh pi@192.168.100.1
pi@192.168.100.1's password:
Linux pi-1 4.19.97-v7+ #1294 SMP Thu Jan 30 13:15:58 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 10 02:33:17 2021 from 192.168.100.254

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

pi@pi-1:~$
```

Figure 20 : SSH host confirmation message (top), SSH terminal (bottom)

An alternative to Telnet is Secure Shell (SSH). To connect to Pi-1 using SSH click on the start button and select the command prompt.



-> Command Prompt

In the command prompt enter the command below and then press the return  key.

```
ssh pi@192.168.X.1          (X=Box/Desk)
```

Note, remember the IP address for your Pi-1 will be different. You also need to pass the user-name (Pi) otherwise it will default to your PC user-name. If this is the first time you have SSHed into Pi-1 you will be asked to confirm that you wish to log into this machine, as shown in figure 20, enter :

yes

At the login prompt shown in figure 20 enter the same password (12345). This will again open a

command terminal on the remote machine. However, now all communications between the Pi and the PC are encrypted, were as when you use Telnet, data is sent as plain text i.e. in theory anyone connected to the network can see what you are typing :(.

Note, we will look at this problem in the next lab. As a general rule you would **not** use telnet on the Internet as it is not secure i.e. people could capture your user-name and password. If you are on a private network and you know what machines are connected to it, you could use Telnet. However, we would typically use SSH, just in case, you never know who is listening :).

To state the obvious there is an increased processing load associated with SSH when compared to Telnet, as you have to encrypt and decrypt the network packets. Normally this is not an issue, but for small embedded systems this may cause problems e.g. limited memory / processor time.

You can SSH or Telnet from one Raspberry Pi to another or from the PC by entering the following commands:

```
telnet -l pi 192.168.X.Y
```

OR

```
ssh pi@192.168.X.Y
```

IP address will vary X=Box/Desk (1 to 85) Y=Pi (1 to 3).
--

Note, the username and password for each Raspberry Pi are the same : pi and 12345. To exit a Telnet or SSH terminal type: `exit`.

Task : make sure you can Telnet or SSH to each Pi, you will need this skill in later labs. You can also connect to the Pi using its name rather than its IP address as shown below:

```
telnet -l pi pi-1
```

OR

```
ssh pi@pi-1
```

Task : from the SSH terminal on Pi-1 use the `ping` command to test the connections to the different hosts connected to this private network i.e. ping Pi-1, Pi-2, Pi-3 and the PC. Their IP addresses are shown in figure 13, remember to change 100 to your desk number. Examine the ping times, why is one address faster than the others?

Note, a small difference between Linux and Windows is that in Linux the ping command does not stop after four packets, it will continue until the keys CTRL-C are pressed. However, in Linux we can specify the number of packets to be sent using the `-c` option (packet count) e.g. to ping Pi-2 four time you would use the command :

```
ping -c 4 192.168.X.2          (X=Box/Desk)
```

To connect to another desk e.g. Pi-1 in a different test rig, both systems need to be connected to the same Local Area Network (LAN), the lab's network. This is available using the **GREEN** network cables, accessed using the network socket mounted on the desk, shown in figure 21.

IMPORTANT, when inserting this plug you should hear a definite “click” as the top clip latches into place. When removing this plug do make sure you push down this clip before trying to pull out the cable.

Task : plug the **GREEN** cable into the spare socket on the network switch, as shown in figure 21. This will connect your Pi system to the lab’s internal network. It may take a couple of minutes for the main network switch to detect that your system is connected. To test if this connection is active ping each of the top level servers shown in figure 22 e.g. the NTP time server 192.168.100.1 etc.

Note, listen out for a sound and look for a green light when you ping one of these servers :). For more details have a look at Appendix C.

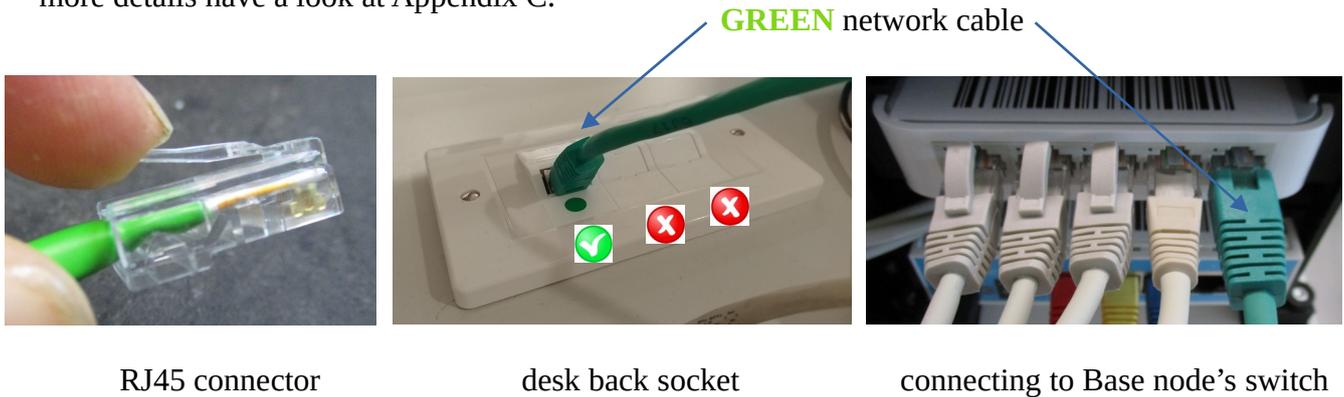


Figure 21 : connecting to the lab’s network

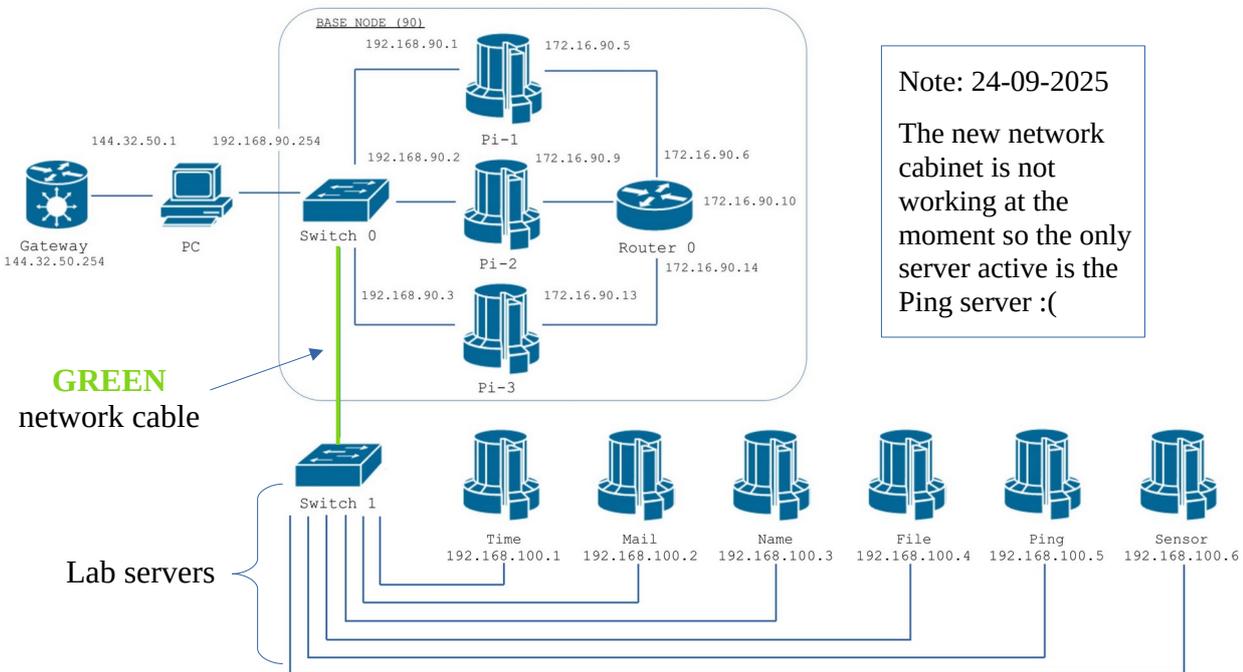


Figure 22 : top level lab network servers

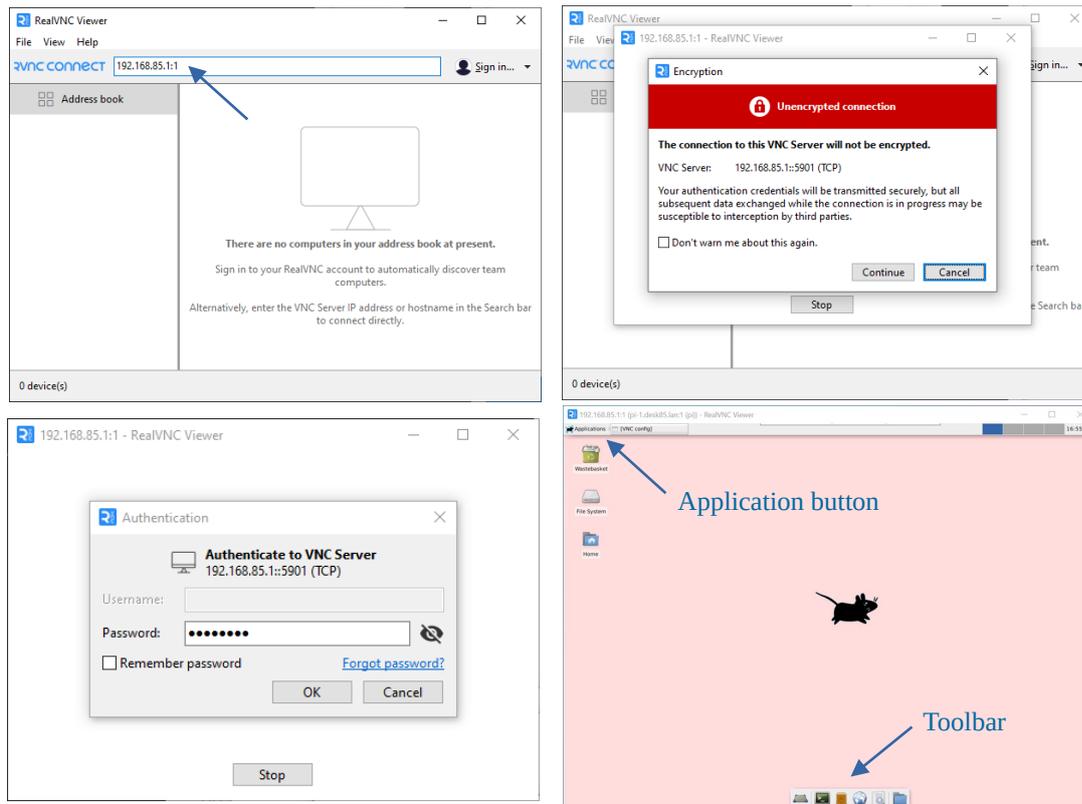


Figure 23 : VNC remote desktop

Task 4

A graphical alternative to opening a command terminal on the Raspberry Pis is to open a remote desktop using VNC (Virtual Networking Computing), as shown in figure 23. For more information refer to:

https://en.wikipedia.org/wiki/Virtual_Network_Computing

To connect to the VNC server running on each Raspberry Pi click on the PC start button and select VNC Viewer.



-> RealVNC Viewer

First time you run this program you will be asked to sign-in, **don't**, rather click on the use without signing in option. Next, enter the IP address of the Raspberry Pi you wish to connect to and its display number. The IP address of the Raspberry Pi e.g. Pi-1 is specific to your desk/box, however, the display number i.e. “: 1” will be the same:

VNC server : 192.168.X.1:1 (X=Box/Desk)

Press enter. A pop up window will appear warning you that this connection is not encrypted, as we are using the lab network this is fine so click Continue. Then enter the password below and click

Ok, you can now access a remote desktop on Pi-1

Password : 12345678

If you do not see a pink (Pi1), yellow (Pi2) or blue (Pi3) background as shown in figure 23, you may need to increase the VNC colour level (depth) i.e. the number of colours used to display the remote desktop. To do this close the remote desktop, then in RealVNC select:

File -> Preferences ...

Click on Expert settings and update the ColorLevel option as shown in figure 24. Then reopen the remote desktop.

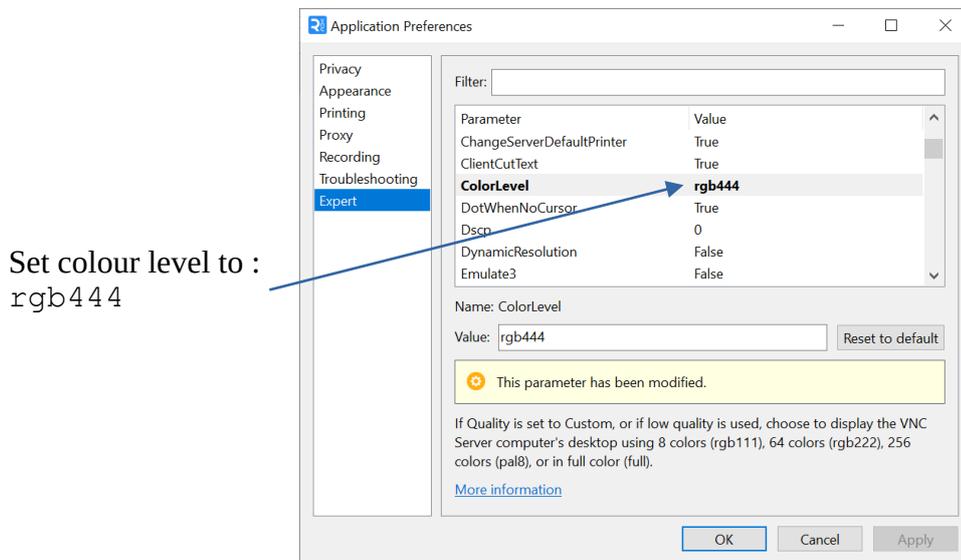


Figure 24 : VNC colour depth

To open a terminal on the remote desktop click on the Terminal icon on the bottom toolbar, as shown in figure 25.



Figure 25 : Remote desktop toolbar

Alternatively, click on the menu button , then select Terminal Emulator. To close this remote desktop left click on top right X icon, or move the mouse over the middle of the top title bar and a drop down menu will appear with additional options.

Task : make sure you can Telnet, SSH and VNC into each Pi as we will be using these skills in the next lab. Really, do STOP and take the time to check that you can do this.

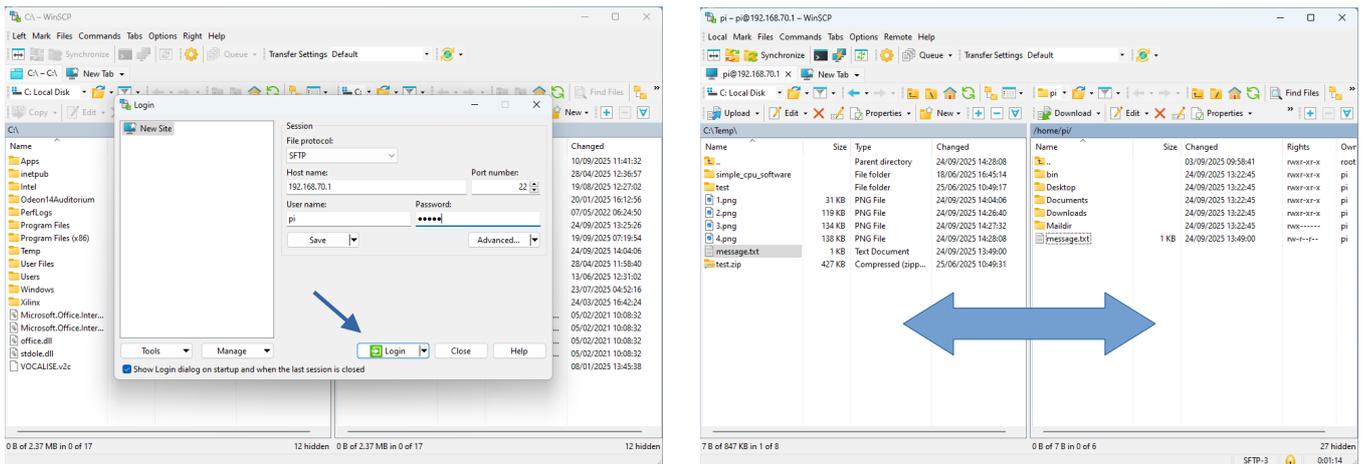


Figure 26 : WinSCP, file transfers between PC and Pi

Task 5

To transfer files between the PC and a Raspberry Pi we can mount the Raspberry Pi filing system across the network. On the PC open WinSCP :



-> WinSCP

In the Login pop-up window enter the following information:

- File Protocol : SFTP
- Host Name : 192.168.X.Y (X=Box/Desk, Y=Pi (1 to 3))
- Port Number : 22
- User name : pi
- Password : 12345

To connect, left click on the Login button Login , as shown in figure 26 .

Note, files are transferred between these computers using the Secure File Transfer Protocol (SFTP), we will be looking at this in more detail in lab 4.

Task : on the PC use Notepad to create a new text file message.txt in C:\temp. Using WinSCP move this file to Pi-1 i.e. drag and drop. Next, open a VNC session to Pi-1. Using the File Browser on the Pi double click on this text file to open it, change and save its contents. Finally move this file back to the PC i.e. drag and drop the file from Pi-1 to C:\temp, overwriting the original file. Double click on this file to open it and confirm these changes were saved.

You can also perform these file transfers using the command line via the command scp i.e. Secure CoPy. On the PC open a new command prompt:



-> Command Prompt

Using the `cd` command change directory to `C:\temp`, then enter the command:

```
scp message.txt pi@192.168.X.1:/home/pi (X=Box/Desk)
```

When prompted enter the Pi password. The file `message.txt` will now be transferred from the PC to Pi-1's home directory `/home/pi`. The commands needed to transfer a file from the PC to the Pi and from the Pi to the PC are shown in figure 27.

Note, transferring files between the PC and Pi is another key skill you will need in later labs. Do make sure you can write files to the Pi and then read them from the PC.

```
C:\Users\mjf5>cd c:\temp

c:\Temp>ls
1.png 2.png 3.png 4.png 5.png message.txt simple_cpu_software test test.zip

c:\Temp>scp message.txt pi@192.168.70.1:/home/pi
pi@192.168.70.1's password:
message.txt                               100%  7    1.4KB/s  00:00

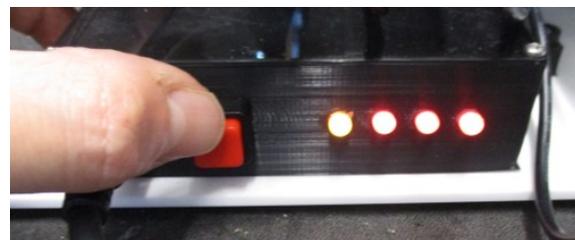
c:\Temp>scp pi@192.168.70.1:/home/pi/message.txt .
pi@192.168.70.1's password:
message.txt                               100%  7    1.0KB/s  00:00
```

Important "." = current directory

Figure 27 : SCP file transfer from Pi-1 to PC



Press button to start shut down



LEDs flash RED during shut down process



LEDs fixed ORANGE, shut down process complete

Figure 28 : shutdown process

When you have finished, close all open VNC, Telnet and SSH sessions. To shutdown the Raspberry Pi system press the big **RED** button on the front box. This triggers the shutdown process on each Pi.

During this shutdown process the status LEDs will flash RED, as shown in figure 28. Do not unplug the power until these LEDs have stopped flashing for at least 5 seconds i.e. change to a fixed **ORANGE**. The Raspberry Pi system can then be unplugged and placed back into its box and the power supply returned its box. Finally, return the box to the front or rear cabinet from where it came from.

Note, on older versions of the Pi there was a reset pin allowing the user to turn on the Pi when shutdown. However, this was removed in the version we are using, sooo, to turn the Pi back on you need to pull out and plug back in the power cable i.e. a power cycle :(. Also, when removing the network cable do make sure you push down on the clip before trying to pull out the cable. The network cables can also be stored in the desk draw.

Appendix A : York ping results in Linux

```
mike@mike-Aspire ~ $ ping 144.32.128.40
PING 144.32.128.40 (144.32.128.40) 56(84) bytes of data.
64 bytes from 144.32.128.40: icmp_seq=1 ttl=251 time=1.03 ms
64 bytes from 144.32.128.40: icmp_seq=2 ttl=251 time=0.956 ms
64 bytes from 144.32.128.40: icmp_seq=3 ttl=251 time=0.912 ms
64 bytes from 144.32.128.40: icmp_seq=4 ttl=251 time=0.873 ms
64 bytes from 144.32.128.40: icmp_seq=5 ttl=251 time=0.931 ms
64 bytes from 144.32.128.40: icmp_seq=6 ttl=251 time=0.943 ms
64 bytes from 144.32.128.40: icmp_seq=7 ttl=251 time=0.997 ms
^C
--- 144.32.128.40 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6025ms
rtt min/avg/max/mdev = 0.873/0.949/1.032/0.051 ms
mike@mike-Aspire ~ $
```

Figure A1 : Linux ping command results

Appendix B : Cray-1 super-computer



Figure B1 : Cray-1 super computer (<https://en.wikipedia.org/wiki/Cray-1>)

Appendix C : Ping server



Figure C1 : Ping server

Remember the RTT gives a good indication of distance, but equally this delay could be down to network congestion, server load, slow network technologies, or me adding a transport delay :).

More information on this server is available here :

http://simplecputdesign.com/networking_ping_node/index.html