Lab Report 2: Network Simulation

Student Number	Student name:	Contribution Percentage:
S4099343	Yew Cher Ooi	50%
S3849194	William Cataldo	50%
Group:	5	
Session:	Monday 4.30PM-6.30PM	
Lab Demonstrator Name:	Tharindu Uduptiya	

1. ABSTRACT

This lab will introduce participants to Cisco Packet Tracer through a set of short logical workshops involving the creation and simulation of simple ideal networks. It will also strengthen participants knowledge of OSI Layers 2 and 3 through the study of protocol fields using the Wireshark application in conjunction with command prompt functions built into participants devices.

2. INTRODUCTION AND OBJECTIVE

Part 1

Understand the principles of simulations and explore the logical workspace and operations of Cisco Packet Tracer.

This will involve opening and exploring Cisco Packet Tracer, placing and connecting network components in the workspace, and sending messages and signals between them through tools in the app.

Part 2

Understand the main principles of OSI Layer 2 protocols and devices through exploration of fields in ethernet frames, exploration of ARP protocol, and the seeing the relationship between MAC IP addressing.

This will involve the capturing of traffic of specific protocols and scrutinising them in the Wireshark application, specifically IPv4, IPv6, and ARP protocols to find the contents and purpose of their headers and the fields contained within them.

Part 3

Understand the main principles of OSI Layer 3 protocols and devices through the exploration of fields in IP packets/datagrams, exploration of the ICMP protocol, and seeing packet fragmentation in action.

This will involve the capturing of traffic of specific protocols and scrutinising them in the Wireshark application, specifically IPv4, IPv6, and ICMP protocols to find the contents and purpose of their headers and the fields contained within them.

3. RESULTS AND DISCUSSION

Part 1

Exercises

4.1 Creating a simple topology:

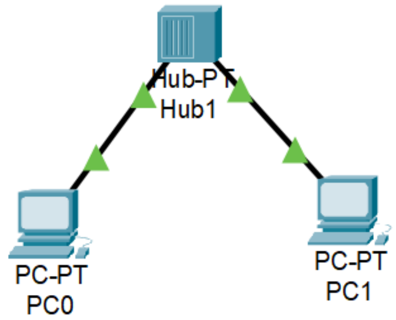

Fig 1. Network topology comprising of one network hub and two PCs.

For connecting the hub to each PC, a copper straight-through cable is used.

4.2 Creating a simple topology with switch:

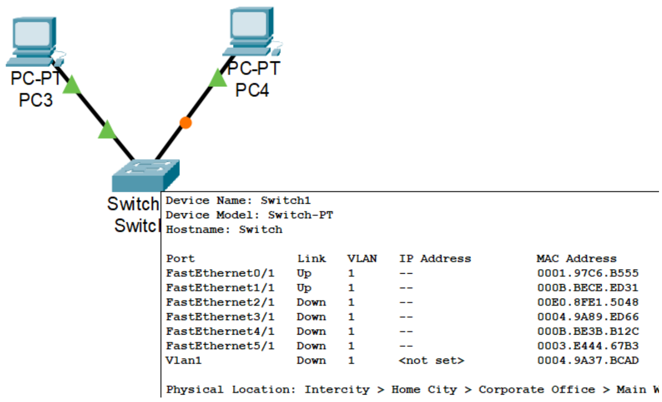

Fig 2. Network topology comprising of one network switch and two PCs.

Q: How does the physical interface of a switch look like?

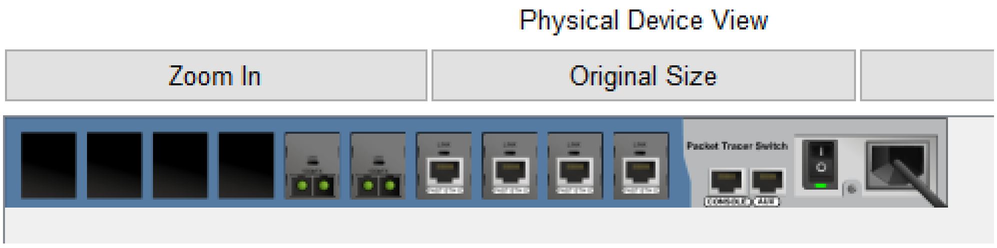

Fig 3. Physical device view of a switch

It has many ethernet ports for both copper cabling and fibre optics, a power switch and an IEC power socket.

4.3 Creating multilevel topologies

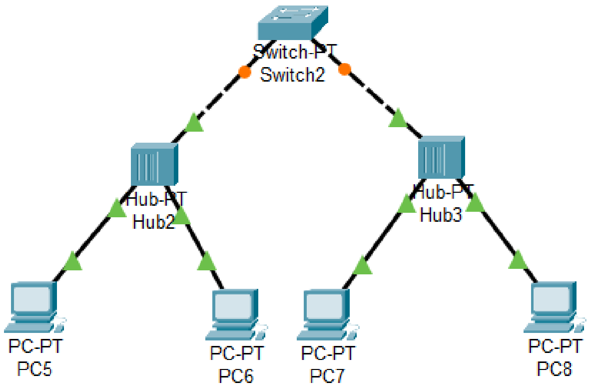

Fig 4. Multilevel topology

For connecting the switch to the hubs, a copper cross-over cable is used since they are both devices that function in Layer 2.

Questions

1. Learn how to customize the Cisco Packet Tracer options, for doing that click on **help>tutorials**. It will open a browser and a tutorial will be displayed. Click on getting started and click the three tutorials about options.

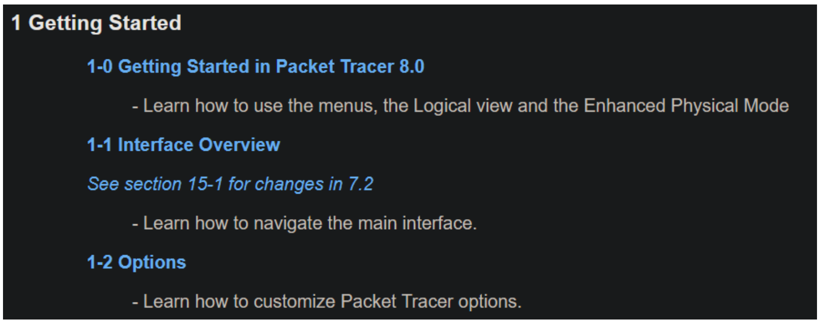

Fig 5. Learning about the basics through the built-in tutorials.

2. Learn how to create a topology, for doing that click on **help>tutorials**. It will open a browser and a tutorial will be displayed. Click on getting started and click the second tutorial about options.

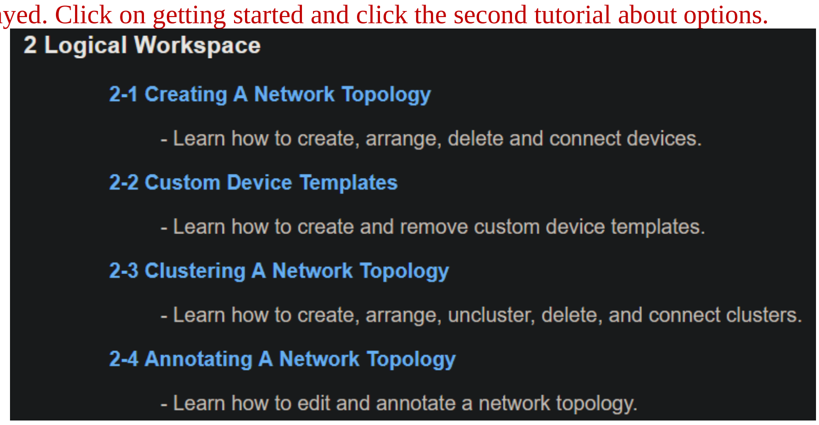

Fig 6. Learning about creating a topology through the built-in tutorials.

3. Provide and explain two options of **Common Tools Bar**.

The Common Tools Bar is the top bar shown in with the white background shown in Fig 1. The first one shown with the Hotkey ‘N’ lets us place a custom note.

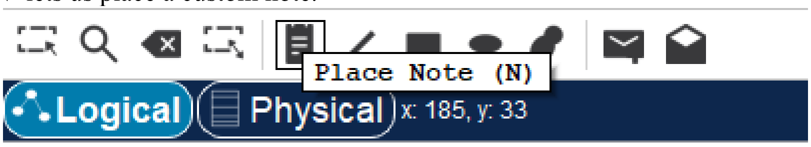

Fig 7. “Place Note” option on the Common Tools Bar

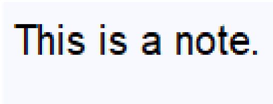

Fig 8. Written note with the "Place Note" tool

The second tool shown is the "Select Tool". It lets us select multiple elements and perform operations on all of them.

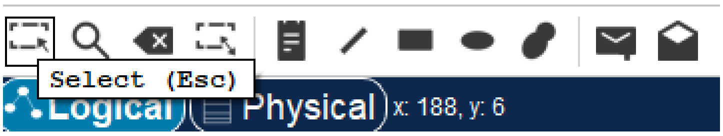

Fig 9. "Select' option on the Common Tools Bar

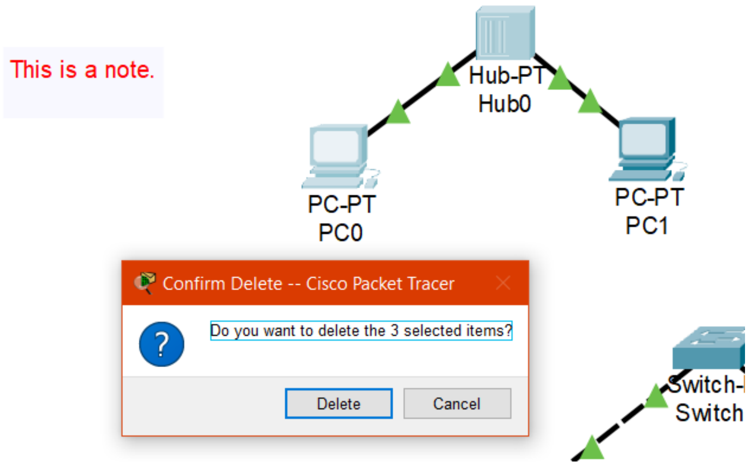

Fig 10. Deleting selected elements.

4. Provide and explain four **Device-Type** that can be used on the simulator.

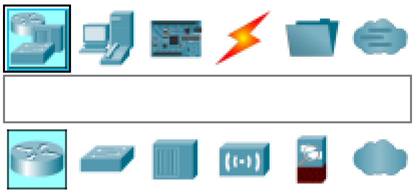

Fig 11. Device List at the bottom-left of the GUI

1. Network Switches: A network device that filters and forwards packets in a LAN network. It determines the destination of the packet by examining the MAC address in the header. Integrity checks are also done depending on the switch type.
2. Network Hubs: A multi-port repeater that repeats data sent in one input node to all other nodes.
3. End Devices: Simulates source and destination devices that send data over networks (e.g. PC, Servers, Printers).
4. Boards: Simulates a microcontroller board with General Purpose Input Output (GPIO) pins and programmable chips that be used to perform operations on signals.

5. Provide your personal feedback about the simulator, is it friendly? Is it useful?

Yes, it is quite user-friendly. One feature we particularly enjoyed was the interactable physical interface (Physical Device View) of each device. As a simulator, it does a good job of providing on-hands experience with managing networks.

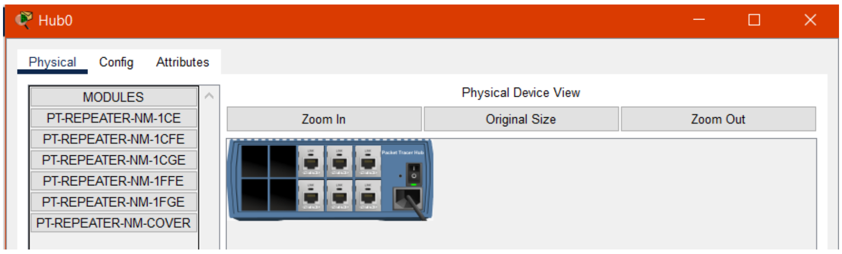

Fig 12. Physical Device View

Part 2

Exercises

4.1 Capturing and analysing ethernet frames.

Step 7:

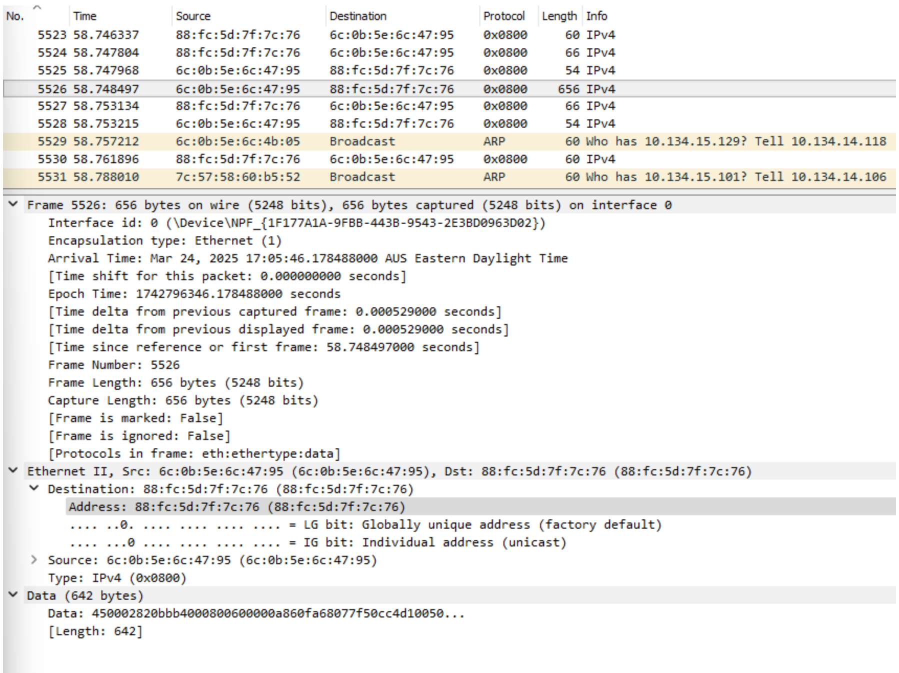

Fig 13. Frame 5526 on Wireshark after filtering out the IPv4 Protocol

The expanded frame in Fig 13. contains the HTTP GET message.

1. What is the 48-bit ethernet address?

```
Address: 6c:0b:5e:6c:47:95 (6c:0b:5e:6c:47:95)
```

Fig 14. Source Address of Frame 5526

2. How does Wireshark know the brand of your PC?

This information is not visualized in the destination address shown in Fig 14. But in technicality, the first three bytes of the MAC address determines the manufacturer.

3. What is the 48-bit destination of the ethernet frame?

```
Address: 88:fc:5d:7f:7c:76 (88:fc:5d:7f:7c:76)
```

Fig 15. Destination Address of Frame 5526

4. How does Wireshark know the brand of the remote device?

This information is not visualized in the destination address shown in Fig 13. But in technicality, the first three bytes of the MAC address determines the manufacturer.

5. Give the hexadecimal value for the two-byte frame field. What upper layer protocol does this correspond to?

```
0000  88 fc 5d 7f 7c 76 6c 0b  5e 6c 47 95 08 00 45 00   ..].|vl. ^lG...E.
0010  02 82 0b bb 40 00 80 06  00 00 0a 86 0f a6 80 77   ....@... .......w
0020  f5 0c c4 d1 00 50 d8 db  58 83 45 bc 46 1e 50 18   .....P.. X.E.F.P.
0x0800
```

Fig 16. The two-byte frame field

The hex values 0x0800 corresponds to the IPv4 protocol.

Step 8:

```
5550 59.001207  6c:0b:5e:6c:47:95      88:fc:5d:7f:7c:76  0x0800   54 IPv4
5551 59.004733  24:6a:0e:c9:44:4b      IPv4mcast_fb       0x0800   82 IPv4
5552 59.004733  fe80::fe43:d45f:b80…   ff02::fb           MDNS    102 Standard query 0x0000 PTR _googlecast._tcp.local,
5553 59.008472  88:fc:5d:7f:7c:76      6c:0b:5e:6c:47:95  0x0800   60 IPv4
5554 59.009191  88:fc:5d:7f:7c:76      6c:0b:5e:6c:47:95  0x0800  295 IPv4
5555 59.009526  38:ca:84:ad:01:3a      Broadcast          ARP      60 Who has 10.134.15.126? Tell 10.134.15.222
5556 59.017450  6c:0b:5e:6c:47:65      Broadcast          ARP      60 Who has 10.134.15.10? Tell 10.134.15.127
5557 59.030673  38:ca:84:ad:01:49      Broadcast          ARP      60 Who has 10.134.15.247? Tell 10.134.14.85
5558 59.051726  6c:0b:5e:6c:47:95      88:fc:5d:7f:7c:76  0x0800   54 IPv4

Frame 5554: 295 bytes on wire (2360 bits), 295 bytes captured (2360 bits) on interface 0
    Interface id: 0 (\Device\NPF_{1F177A1A-9FBB-443B-9543-2E3BD0963D02})
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 24, 2025 17:05:46.439182000 AUS Eastern Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1742796346.439182000 seconds
    [Time delta from previous captured frame: 0.000719000 seconds]
    [Time delta from previous displayed frame: 0.000719000 seconds]
    [Time since reference or first frame: 59.009191000 seconds]
    Frame Number: 5554
    Frame Length: 295 bytes (2360 bits)
    Capture Length: 295 bytes (2360 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:data]
Ethernet II, Src: 88:fc:5d:7f:7c:76 (88:fc:5d:7f:7c:76), Dst: 6c:0b:5e:6c:47:95 (6c:0b:5e:6c:47:95)
    Destination: 6c:0b:5e:6c:47:95 (6c:0b:5e:6c:47:95)
        Address: 6c:0b:5e:6c:47:95 (6c:0b:5e:6c:47:95)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 88:fc:5d:7f:7c:76 (88:fc:5d:7f:7c:76)
        Address: 88:fc:5d:7f:7c:76 (88:fc:5d:7f:7c:76)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
Data (281 bytes)
    Data: 45000119d81540002106f1198077f50c0a860fa60050c4d1...
    [Length: 281]
```

Fig 17. Frame 5554 on Wireshark after filtering out the IPv4 Protocol

The expanded frame in Fig 17. contains the HTTP OK message.

Q1: What is the value of the ethernet source address?

```
Address: 6c:0b:5e:6c:47:95 (6c:0b:5e:6c:47:95)
```

Fig 18. Source Address of Frame 5554

6c:0b:5e:6c:47:95

Q2: What is the destination address of the ethernet frame? Is this the ethernet address of your computer?

```
Address: 88:fc:5d:7f:7c:76 (88:fc:5d:7f:7c:76)
```

Fig 19. Destination Address of Frame 5554

88:fc:5d:7f:7c:76

Yes, the source and destinations are reversed from the HTTP GET message.

Q3: Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

```
0000  88 fc 5d 7f 7c 76 6c 0b  5e 6c 47 95 08 00 45 00   ..].|vl. ^lG...E.
0010  02 82 0b bb 40 00 80 06  00 00 0a 86 0f a6 80 77   ....@... .......w
0020  f5 0c c4 d1 00 50 d8 db  58 83 45 bc 46 1e 50 18   .....P.. X.E.F.P.
0x0800
```

Fig 20. The two-byte frame field

The hex values 0x0800 corresponds to the IPv4 protocol.

4.2 Capturing and analysing Address Resolution Protocol (ARP)

Step 1:

```
C:\Users\Quickemu>arp -a

Interface: 192.168.1.155 --- 0x10
  Internet Address      Physical Address      Type
  192.168.1.1           b4-fb-e4-82-31-1c     dynamic
  192.168.1.101         dc-e5-5b-61-5c-ff     dynamic
  192.168.1.103         bc-0f-f3-bc-94-ed     dynamic
  192.168.1.160         00-90-a9-d3-39-a9     dynamic
  192.168.1.217         9c-6b-00-3e-3d-7d     dynamic
  192.168.1.246         c0-79-82-c1-40-82     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Fig 21. CLI output of $ arp -a

The command line is `arp -a`. It shows the ARP table of the computer. The table is a translation table of the IP addresses and their respective MAC addresses. The type can either be `dynamic` which means it will timeout after a certain period if it does not get refreshed or `static` which means it is fixed and will not change automatically.

Step 2:

```
C:\Users\Quickemu>arp -d *

C:\Users\Quickemu>arp -a

Interface: 192.168.1.155 --- 0x10
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16     static
```

Fig 22. CLI output of $ arp -a ; after clearing ARP cache

The command line to clear all ARP cache is `arp -d *`. The wildcard states to delete all entries. Now we can see that the ARP table is mostly empty.

Step 4:

```
1868 14.002879   6c:0b:5e:6c:47:95   Broadcast           ARP   42 Who has 10.134.15.254? Tell 10.134.15.166
1869 14.003405   88:fc:5d:7f:7c:76   6c:0b:5e:6c:47:95   ARP   60 10.134.15.254 is at 88:fc:5d:7f:7c:76

Command Prompt

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : rmit.edu.au
   Link-local IPv6 Address . . . . . : fe80::a3eb:5d95:350:4a00%4
   IPv4 Address. . . . . . . . . . . : 10.134.15.166
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.134.15.254
```

Fig 23. Frames 1868-1869 and ipconfig output

Q1: What are the hexadecimal values for the source and destination address in the Ethernet frame containing the ARP request message?

The corresponding frame is frame 1868, the source address is 6c:0b:5e:6c:47:95, and the destination address is Broadcast, in hexadecimal values this is ff:ff:ff:ff:ff:ff.

Q2. Give the hexadecimal value for the two-byte Ethernet Frame type field. What upper layer protocol does this correspond to?

```
Type: ARP (0x0806)
[Stream index: 1]
```

Fig 24. Two-byte Ethernet Frame type field

The hexadecimal values 0x0806 correspond to the ARP protocol.

(a) What is the question of the ARP?

```
Who has 10.134.15.254? Tell 10.134.15.166
```

Fig 25. ARP question

The question is "Who has [target]? Tell [sender].

(b) Does the ARP message contain the IP and MAC address of the sender?

```
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HP_6c:47:95 (6c:0b:5e:6c:47:95)
    Sender IP address: 10.134.15.166
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.134.15.254
```

Fig 26. ARP frame field of Frame 1868

Yes, it does, as seen in Fig 26.

(c) Does the ARP message contain the IP and MAC address of the target?

No, it does not, it only includes the IP address of the target, as seen in Fig 24. The target MAC address in the request is just the broadcast address.

(d) What is the target IP address?

The target IP address is 10.134.15.254.

(e) What is the value of the opcode field within the ARP-payload part of the Ethernet frame?

```
00 01
```

Fig 27. Opcode hexadecimal value of Frame 1868

The hexadecimal value for the opcode field is 0x0001 and referring to the opcode field in Fig 26 the value means that it is a request.

Q3. Now find the ARP reply that was sent in response to the ARP request.

Referring to Fig 23, the ARP reply is the proceeding frame, 1869.

(a) What is the answer of the ARP?

```
10.134.15.254 is at 88:fc:5d:7f:7c:76
```

Fig 28. ARP Answer

The answer is "10.134.15.254 is at 88:fc:5d:7f:7c:76".

(b) Does the ARP message contain the IP and MAC address of the sender?

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Cisco_7f:7c:76 (88:fc:5d:7f:7c:76)
    Sender IP address: 10.134.15.254
    Target MAC address: HP_6c:47:95 (6c:0b:5e:6c:47:95)
    Target IP address: 10.134.15.166
```

Fig 29. ARP frame fields of Frame 1869

Referring to Fig 29, yes it does.

(c) Does the ARP message contain the IP and MAC address of the target?

Referring to Fig 29, yes it does.

(d) What is the target IP address?

The target IP address is 10.134.15.166.

(e) What is the value of the opcode field within the ARP-payload part of the Ethernet frame?

Fig 30. Opcode hexadecimal value of Frame 1869

The hexadecimal value for the opcode field is 0x0002, and referring to the opcode field in Fig 29 the value means that it is a reply.

Q4. Check again the ARP table and provide the output.

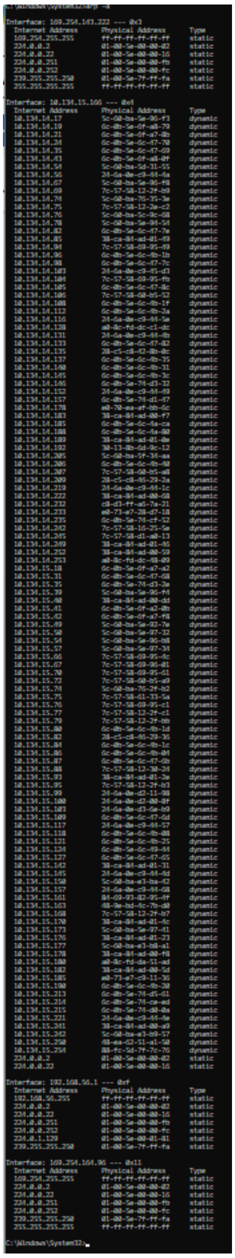

Fig 31. New ARP table

4.3 Basic configuration of a network switch

Step 2:

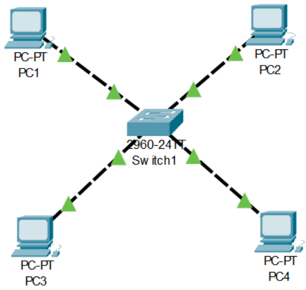

Fig 32. Topology created according to manual

Step 4:

Q1: Explore the show command 'show ?'

```
Switch>show ?
  arp                  Arp table
  cdp                  CDP information
  clock                Display the system clock
  crypto               Encryption module
  dtp                  DTP information
  etherchannel         EtherChannel information
  flash:               display information about flash: file system
  history              Display the session command history
  interfaces           Interface status and configuration
  ip                   IP information
  lldp                 LLDP information
  mac                  MAC configuration
  mac-address-table    MAC forwarding table
  mls                  Show MultiLayer Switching information
  privilege            Show current privilege level
  sessions             Information about Telnet connections
  ssh                  Status of SSH server connections
  tcp                  Status of TCP connections
  terminal             Display terminal configuration parameters
  users                Display information about terminal lines
  version              System hardware and software status
  vlan                 VTP VLAN status
  vtp                  VTP information
```

Fig 33. Output of 'show ?'

It brings up a help menu.

Q2: Explore the interfaces of the switch; what is the command line? What is displayed?

The command line is a terminal where we can type commands and receive telemetry on what is happening in the switch.

IOS Command Line Interface

```
C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(25r)FX, RELEASE SOFTWARE (fc4)
Cisco WS-C2960-24TT (RC32300) processor (revision C0) with 21039K bytes of memory.
2960-24TT starting...
Base ethernet MAC Address: 0003.E4B0.DA77
Xmodem file system is available.
Initializing Flash...
flashfs[0]: 1 files, 0 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 64016384
flashfs[0]: Bytes used: 4670455
flashfs[0]: Bytes available: 59345929
flashfs[0]: flashfs fsck took 1 seconds.
...done Initializing Flash.

Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4
```

Fig 34.. Part of the command line shown at startup

When it starts up, we can see the BIOS starting and the router completing Power-on self-test (POST). We can also see the hardware properties of the switch and respective licenses.

Q3: Check the ARP table, what is inside?

```
Switch>show arp

Switch>
```

Fig 35. Checking ARP table

There is no output, the table is empty.

Q4: Check the MAC address table, what is inside?

```
Switch>show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
```

Fig 36. Checking MAC address table

The table is also empty.

Step 5:

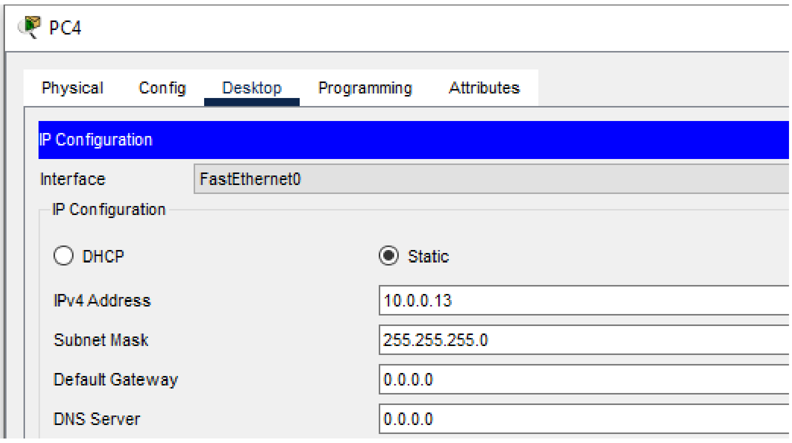

Fig 37. Configuration of PC4

Step 6:

Q1: Check the ARP table (arp -a), what is inside?

```
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
C:\>
```

Fig 38. ARP table of PC4

PC4 was chosen for this step. The table is empty.

Q2: Ping one of the other PCs

```
C:\>ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes=32 time<1ms TTL=128
Reply from 10.0.0.10: bytes=32 time<1ms TTL=128
Reply from 10.0.0.10: bytes=32 time<1ms TTL=128
Reply from 10.0.0.10: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Fig 39. Pinging PC1 from PC4

The IP address of PC1 is 10.0.0.10.

Q3: Check the ARP table (arp -a), what is inside?

```
C:\>arp -a
  Internet Address      Physical Address      Type
  10.0.0.10             0090.2b78.216a        dynamic
```

Fig 40. ARP table of PC4

The ARP table is now populated with the IP and MAC addresses of PC1.

Step 7:

Q1: Check the ARP table (of the switch), what is inside?

```
Switch>show arp
```

Fig 41. ARP table of Switch

The ARP table still appears empty.

Q2: Check the MAC address table (of the switch), what is inside?

```
Switch>show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.c9ca.b472    DYNAMIC     Fa0/1
   1    000c.cf17.a888    DYNAMIC     Fa0/3
```

Fig 42. MAC Address table of switch

The MAC address table is now updated with both PCs MAC Addresses and the ports they are connected to.

Step 8: Ping all the PCs

Pinging from the network switch does not successfully yield anything, so all the PCs are pinged from PC4 in the same way as Fig 39.

```
Switch>ping 10.0.0.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.10, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Switch>ping 10.0.0.11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.11, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Switch>ping 10.0.0.12

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.12, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Switch>ping 10.0.0.13

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.13, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Fig 43. Trying to ping from switch

Q1: Check the MAC address table (of the switch), what is inside?

```
Switch>show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.9621.7969    DYNAMIC     Fa0/2
   1    0030.f244.8803    DYNAMIC     Fa0/3
   1    0090.2b78.216a    DYNAMIC     Fa0/1
   1    00d0.ff76.eeb4    DYNAMIC     Fa0/4
```

Fig 44. MAC Address table of the switch

The MAC address table is now updated with every PCs MAC Addresses and the ports they are connected to.

Q2: What is the role of a Switch? What is the role of MAC Address Table

The role of the switch is to connect devices in a network and forward data packets between them. A switch will only send data to the desired recipient on the network, and it knows the recipient's location from its mac-address-table and from data contained in the IPv4 packet header created by the transmitter device. [7]

The role of the MAC address table is to keep track of the addresses of all devices in the switches network to give it an accurate path for digital traffic so that packets reach their intended destination only.

Questions

1. What is the default amount of time and entry remains in your ARP cache before being removed?

   According to sources online [1], the ARP lifetime for windows devices is generally between 10-20 minutes. Though connections are refreshed each time they are accessed.

2. Why is an ARP table required? At which layer of communication is the MAC address requested? Is it possible to access the internet without MAC address?

   The ARP table is required to map IP addresses to each MAC address in LAN, which allows devices to communicate with each other and with the internet.
   MAC addresses are requested in the Layer 2 protocol in both the OSI model and the TCP/IP model.
   Generally, a MAC address is not required itself to access the internet as that process needs an IP address (Internet Protocol) not a MAC address. The issues you may face by not having a MAC address are that communication between devices on your own network may be suboptimal.

3. Repeat section 3 (see appendix) of the exercises using a HUB to determine whether HUBS and Switches are equal or not.

   In the current configuration, just replacing the switch with a hub will not work. As cross-over cables can only be used between devices that operate at the same layer. The switch here is operating at Layer 2, the Data Link Layer, which does handle MAC addressing. Hubs only operate at Layer 1 since they only repeat the message from one node in all other nodes.

   By replacing the cross-over cables with straight-through cables, we can ping the other PCs.

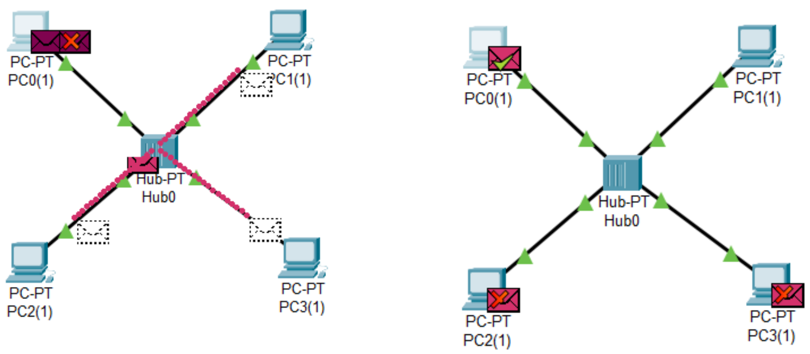

Fig 45. Data moving in a hub configuration

Hubs and switches are not equal. While switches keep and maintain a NAT table to directly send packets to their intended recipients, hubs do not, instead forwarding data to every connected device regardless of the intended recipient. Simply put, switches are smarter than hubs.

Part 3

Exercises

4.1 Capturing and analysing IP packets and datagrams

Step 2:

```
H:\>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
            [-4] [-6] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet (IPv4-only).
    -i TTL         Time To Live.
    -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                   and has no effect on the type of service field in the IP
                   Header).
    -r count       Record route for count hops (IPv4-only).
    -s count       Timestamp for count hops (IPv4-only).
    -j host-list   Loose source route along host-list (IPv4-only).
    -k host-list   Strict source route along host-list (IPv4-only).
    -w timeout     Timeout in milliseconds to wait for each reply.
    -R             Use routing header to test reverse route also (IPv6-only).
                   Per RFC 5095 the use of this routing header has been
                   deprecated. Some systems may drop echo requests if
                   this header is used.
    -S srcaddr     Source address to use.
    -c compartment Routing compartment identifier.
    -p             Ping a Hyper-V Network Virtualization provider address.
    -4             Force using IPv4.
    -6             Force using IPv6.
```

Fig 46. "ping" output

Simply brings up a help manual

Step 3:

The command "ping www.google.com -n 20", referring to the manual means that 20 echo requests will be sent to the target address.

Step 5:

451	5.503677	6c:0b:5e:6f:a7:a2	Broadcast	ARP	60	Who has 10.134.14.212? Tell 10.134.15.18
452	5.504053	HewlettP_a6:7a:21	Broadcast	ARP	60	Who has 10.134.15.122? Tell 10.134.14.232
453	5.527991	6c:0b:5e:6c:47:7c	Broadcast	ARP	60	Who has 10.134.15.34? Tell 10.134.14.98
454	5.541741	6c:0b:5e:74:cf:bf	Broadcast	ARP	60	Who has 10.134.15.172? Tell 10.134.15.189
455	5.545175	6c:0b:5e:6c:4b:00	Broadcast	ARP	60	Who has 10.134.15.101? Tell 10.134.14.184
456	5.577422	10.134.15.166	172.217.167.100	ICMP	74	Echo (ping) request id=0x0001, seq=443/47873, tt…
457	5.583976	7c:57:58:69:95:75	Broadcast	ARP	60	Who has 10.134.14.187? Tell 10.134.15.149
458	5.592015	172.217.167.100	10.134.15.166	ICMP	74	Echo (ping) reply id=0x0001, seq=443/47873, tt…
459	5.592015	24:6a:0e:d2:11:60	Broadcast	ARP	60	Who has 10.134.15.100? Tell 10.134.14.218
460	5.640390	24:6a:0e:c9:44:4b	Broadcast	ARP	60	Who has 10.134.15.242? Tell 10.134.14.131
461	5.644773	6c:0b:5e:6c:47:8c	Broadcast	ARP	60	Who has 10.134.15.122? Tell 10.134.14.105
462	5.660945	24:6a:0e:d3:6e:ae	Broadcast	ARP	60	Who has 10.134.14.79? Tell 10.134.15.102
463	5.668938	6c:0b:5e:6f:a7:de	Broadcast	ARP	60	Who has 10.134.14.187? Tell 10.134.15.17
464	5.685384	38:ca:84:ad:00:77	Broadcast	ARP	60	Who has 10.134.15.55? Tell 10.134.15.246
465	5.694414	5c:60:ba:76:35:99	Broadcast	ARP	60	Who has 10.134.14.202? Tell 10.134.15.58

Fig 47. Wireshark capture contents

4.2 A look at the captured trace

Step 1:

Referring to Fig 47, the request message chosen here is Frame 456.

```
> Frame 456: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{1F177A1A-
> Ethernet II, Src: HP_6c:47:95 (6c:0b:5e:6c:47:95), Dst: Cisco_7f:7c:76 (88:fc:5d:7f:7c:76)
v Internet Protocol Version 4, Src: 10.134.15.166, Dst: 172.217.167.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xb101 (45313)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.134.15.166
    Destination Address: 172.217.167.100
    [Stream index: 20]
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4ba0 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 443 (0x01bb)
    Sequence Number (LE): 47873 (0xbb01)
    [Response frame: 458]
  > Data (32 bytes)
```

Fig 48. Expanded view of frame 456

Q1: What is the IP address of your computer? What is the source and destination address?

```
Internet Protocol Version 4, Src: 10.134.15.166, Dst: 172.217.167.100
```

Fig 49. Source and destination addresses of frame 456

Source address: 10.134.15.166; Destination address: 172.217.167.100

My IP address is 10.134.15.166 , as confirmed in cmd.

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : rmit.edu.au
   Link-local IPv6 Address . . . . . : fe80::a3eb:5d95:350:4a00%4
   IPv4 Address. . . . . . . . . . . : 10.134.15.166
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.134.15.254
```

Fig 50. ipconfig output on host pc

Q2: Within the IP packet header, what is the value in the upper layer protocol field?

```
Protocol: ICMP (1)
```

Fig 51. Upper layer protocol field of frame 456

The value in the Upper Layer Protocol field is 1, which refers to ICMP.

Q3: How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

```
Internet Protocol Version 4, Src: 10.134.15.166, Dst: 172.217.167.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
```

Fig 52. IP Header length of frame 456

```
Data (32 bytes)
    Data: 6162636465666768696a6
    [Length: 32]

Bytes 42-73: Data (data.data)
```

Fig 53. Payload length of frame 456

Referring to Fig 52 and 53, the IP header length is 20 bytes, and the number of payload bytes is 32 bytes. We can determine that by hovering over the last set of hex strings and Wireshark will tell us that bytes 42-73 (32 bytes) are reserved for the payload (as seen in Fig 53).

Q4: Has this IP datagram been fragmented? Explain how you determined whether the datagram was fragmented or not.

```
000. .... = Flags: 0x0
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
```

Fig 54. Flags of frame 456

Referring to Fig 54, in the IPv4 header it is shown that the fragment offset is 0. This would mean that the packet has not been fragmented, which seems reasonable as the total frame length is 592 bits. TCP's maximum segment size (MSS) is roughly 536 bits for IPv4 [6], and MSS does not include the IP and ICMP headers which are 20 and 40 bytes respectively, leaving us well below.

Q5: What is the value in the Identification field and the TTL field?

```
Identification: 0xb101 (45313)
```

Fig 55. Identification field of frame 456

```
Time to live: 128
```

Fig 56. TTL field of frame 456

The value in the identification field is '0xb101' or 45313 in decimal.

The value in the TTL field is 128, which refers to 128 seconds.

Step 2:

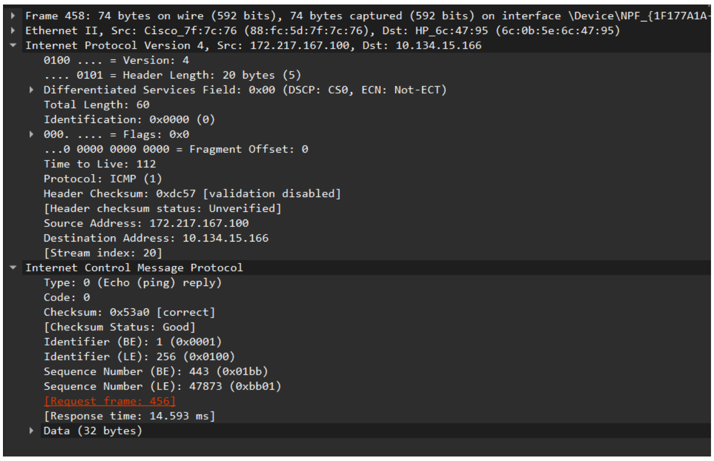

Fig 57. Expanded view of frame 458 (REPLY message)

Q1: What is the IP address of your computer? What is the source and destination address?

Fig 58. Source and destination addresses of frame 458

Source address: 172.217.167.100; Destination address: 10.134.15.166
My IP address is 10.134.15.166, referring to Fig 50.

Q2: Within the IP packet header, what is the value in the upper layer protocol field?

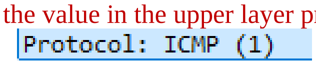

Fig 59. Upper layer protocol field of frame 458

The value in the Upper Layer Protocol field is 1, which refers to ICMP.

Q3: How many bytes are in the IP header? How many bytes are in the payload of the IP datagram?

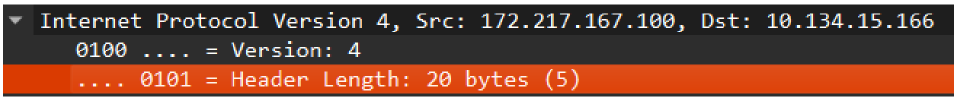

Fig 60. IP Header length of frame 458

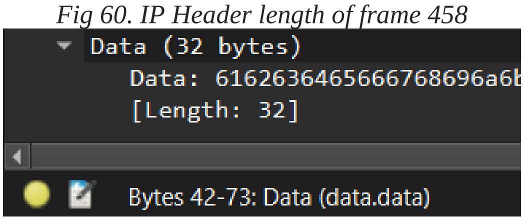

Fig 61. Payload length of frame 458

Referring to Fig 60 and 61, the IP header length is 20 bytes, and the number of payload bytes is 32 bytes.

Q4: Has this IP datagram been fragmented?

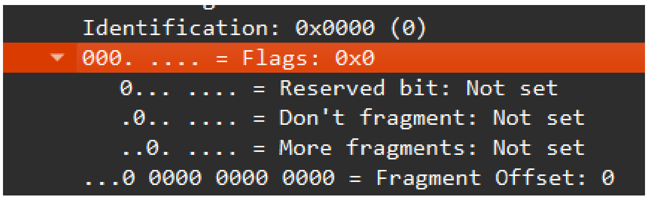

Fig 62. Flags of frame 458

Referring to Fig 62, in the IPv4 header it is shown that the fragment offset is 0. It also has no true values on its fragmentation flags. This This would mean that the packet has not been fragmented, which seems reasonable as the total frame length is 592 bits. TCP's maximum segment size (MSS) is roughly 536 bits for IPv4 [6], and MSS does not include the IP and ICMP headers which are 20 and 40 bytes respectively, leaving us well below.

Q5: What is the value in the Identification field and the TTL field?

```
Identification: 0x0000 (0)
```

Fig 63. Identification field of frame 458

```
Time to live: 112
```

Fig 64. TTL field of frame 458

The value in the identification field is 0.
The value in the TTL field is 112, which refers to 112 seconds.

Q6: Do these values remain unchanged for the ICMP Echo Reply sent to your computer by the nearest (first hop) router? Why?

The values that should remain unchanged for all ICMP Echo replies are:

- Type and Code: ICMP echoes always have a type value of 0 and a code value of 0
- ICMP Identifier and Sequence number: The identifier field and sequence number field will both always match the echo request's corresponding fields. (referring to Fig 48 & 57)
- Data: The echo echoes the data sent in the request.

4.3 Fragmentation

Q1: Analyse the first fragment of the fragmented IP datagram (92):

```
> Frame 92: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
> Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
> Data (1480 bytes)
```

Fig 65. Frame 92

(a) What information in the IP header indicates that the datagram has been fragmented?

The flags field indicates "more fragments"

```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x32f9 (13049)
  v Flags: 0x01 (More Fragments)
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..1. .... = More fragments: Set
    Fragment offset: 0
```

(b)

Fig 66. "More fragment" flag in frame 92

(c) What information in the IP header indicates whether this is the first fragment versus a latter fragment?

But the fragment offset is still 0 (referring to Fig X) so this must be the first fragment. The third line under the 'flags' header denoting more fragments to come also says "set". The datagram is set to be reassembled in IPv4 frame: 93. Frame 93 should contain the only other segment.

```
Reassembled IPv4 in frame: 93
```

Fig 67. Reassembled field of frame 92

(d) How long is this IP datagram?

The data itself is 1480 bytes but including the headers, the total length of the datagram is 1514 bytes (referring to the top of Fig X)

Q2: Analyse the second fragment of the fragmented IP datagram (93):

```
> Frame 93: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits)
> Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
> Internet Control Message Protocol
```

Fig 68. Frame 93

(a) What information in the IP header indicates that this is not the first datagram fragment?

The fragment offset indicates that this is not the first datagram fragment.

```
Fragment offset: 1480
```

Fig 69. Fragment offset of frame 93

(b) Are there more fragments? How can you tell?

We can tell there will be no more fragments because the “More fragments” flag is not set.

```
Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
Fragment offset: 1480
```

Fig 70. “More fragments” flag of frame 93

(c) What fields change in the IP header between the first and the second fragment?

Referring to Fig X and Fig X

- Total Length: changed from 1514 to 562
- More fragments flag went from ‘set’ to ‘not set’
- Fragment offset: since the second fragment was offset by the total length of the first minus the IPv4 header length. (1480)
- The header checksum has changed, because it’s a different header.
- The last line has changed since it is acknowledging all the data it has received in the second packet.

4.4 Basic analysis of routers (static route)

Q1: What is the IP address of PC0 and PC1?

```
FastEthernet0 Connection:

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::250:FFF:FEA9:25B5
   IPv6 Address....................: ::
   IPv4 Address....................: 1.0.0.2
   Subnet Mask.....................: 255.0.0.0
   Default Gateway.................: ::
                                     1.0.0.1
```

Fig 71. “ipconfig” output of PC0

IP address of PC0: 1.0.0.2

```
FastEthernet0 Connection:

   Connection-specific DNS Suffix..:
   Link-local IPv6 Address.........: FE80::209:7CFF:FE06:BD29
   IPv6 Address....................: ::
   IPv4 Address....................: 4.0.0.2
   Subnet Mask.....................: 255.0.0.0
   Default Gateway.................: ::
                                     4.0.0.1
```

Fig 72. “ipconfig” output of PC1

IP address of PC1: 4.0.0.2

Q2: From PC0 ping PC1, is ping working?

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 4.0.0.2

Pinging 4.0.0.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Reply from 4.0.0.2: bytes=32 time<1ms TTL=125

Ping statistics for 4.0.0.2:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 4.0.0.2

Pinging 4.0.0.2 with 32 bytes of data:

Reply from 4.0.0.2: bytes=32 time<1ms TTL=125
Reply from 4.0.0.2: bytes=32 time<1ms TTL=125
Reply from 4.0.0.2: bytes=32 time<1ms TTL=125
Reply from 4.0.0.2: bytes=32 time<1ms TTL=125

Ping statistics for 4.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Fig 73. Pinging PC1 from PC0

The first three requests time out but the ones starting from the fourth work normally.

Q3: In PC0 using the command line 'tracert [PC1 IP address]', check what is the path for reaching PC1.

```
C:\>tracert 4.0.0.2

Tracing route to 4.0.0.2 over a maximum of 30 hops:

  1   0 ms      0 ms      0 ms      1.0.0.1
  2   0 ms      0 ms      0 ms      2.0.0.2
  3   0 ms      0 ms      0 ms      3.0.0.2
  4   0 ms      0 ms      0 ms      4.0.0.2

Trace complete.
```

Fig 74. Tracing PC1 from PC0

The path is 4 hops long and includes:
1.0.0.1 (Router 1) → 2.0.0.2 (Router 2) → 3.0.0.2 (Router 3) → 4.0.0.2 (PC1)
*Router IP addresses confirmed from their "FastEthernet 0/0" configurations.

Q4: (a) What does the ip route display? Is it the routing table?

```
Router>enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    1.0.0.0/8 is directly connected, FastEthernet0/0
C    2.0.0.0/8 is directly connected, FastEthernet0/1
S    3.0.0.0/8 [1/0] via 2.0.0.2
S    4.0.0.0/8 [1/0] via 2.0.0.2
```

Fig 75. Output of $ show ip route; in Router1

The output shows 4 entries that make up the routing table. The first two have the code C that refers to the destinations of PC0 and Router 2 which are directly connected (through dynamic routing), and the last two entries have the code S that are static routings manually set in the configuration. It states the destinations and mentions neighbouring devices as paths to connect to Router 3 and PC1.

(b) What does the arp display?

```
Router# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  1.0.0.1                 -   00D0.FF60.1A01  ARPA   FastEthernet0/0
Internet  1.0.0.2                 8   0050.0FA9.25B5  ARPA   FastEthernet0/0
Internet  2.0.0.1                 -   00D0.FF60.1A02  ARPA   FastEthernet0/1
Internet  2.0.0.2                 8   0090.2189.AD01  ARPA   FastEthernet0/1
Router# show running config
```

Fig 76. Output of $ show arp; in Router1

The ARP table shows the IP addresses of the two directly connected devices, PC0 and Router 2, accompanied by their MAC addresses. It also shows the protocol, how long it has been connected in minutes and the interface of the router it is connected to.

(c): What does the running-config display?

```
Router#show running-config
Building configuration...

Current configuration : 635 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
!
!
!
!
ip cef
no ipv6 cef
!
!
!
!
!
!
!
!
!
!
!
!
spanning-tree mode pvst
!
!
!
!
!
!
interface FastEthernet0/0
 ip address 1.0.0.1 255.0.0.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 2.0.0.1 255.0.0.0
 duplex auto
 speed auto
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
ip route 3.0.0.0 255.0.0.0 2.0.0.2
ip route 4.0.0.0 255.0.0.0 2.0.0.2
!
ip flow-export version 9
!
!
!
!
!
!
!
!
line con 0
!
line aux 0
!
line vty 0 4
 login
!
!
!
end
```

Fig 77. Output of $ show running-config; in Router1

The command outputs the entire configuration that the router is built with. It includes the size of the configuration, the firmware version, its hostname, the IP versions that are to be enabled, the dynamic routes connected to its interfaces, the static routes and miscellaneous options that are not used.

Q5: Repeat this with all the routers.

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

S    1.0.0.0/8 [1/0] via 2.0.0.1
C    2.0.0.0/8 is directly connected, FastEthernet0/0
C    3.0.0.0/8 is directly connected, FastEthernet0/1
S    4.0.0.0/8 [1/0] via 3.0.0.2
```

Fig 78. Output of $ show ip route; in Router2

```
Router#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  2.0.0.1                 14   00D0.FF60.1A02  ARPA   FastEthernet0/0
Internet  2.0.0.2                  -   0090.2189.AD01  ARPA   FastEthernet0/0
Internet  3.0.0.1                  -   0090.2189.AD02  ARPA   FastEthernet0/1
Internet  3.0.0.2                 14   000D.BD21.4601  ARPA   FastEthernet0/1
```

Fig 79. Output of $ show arp; in Router2

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

S    1.0.0.0/8 [1/0] via 3.0.0.1
S    2.0.0.0/8 [1/0] via 3.0.0.1
C    3.0.0.0/8 is directly connected, FastEthernet0/0
C    4.0.0.0/8 is directly connected, FastEthernet0/1
```

Fig 80. Output of $ show ip route; in Router3

```
Router#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  3.0.0.1                 16   0090.2189.AD02  ARPA   FastEthernet0/0
Internet  3.0.0.2                  -   000D.BD21.4601  ARPA   FastEthernet0/0
Internet  4.0.0.1                  -   000D.BD21.4602  ARPA   FastEthernet0/1
Internet  4.0.0.2                 16   0009.7C06.BD29  ARPA   FastEthernet0/1
```

Fig 81. Output of $ show arp; in Router3

In the routing tables for each router, the connected flags are updated to whichever device is directly connected and static flags are updated to the devices manually set in configuration, whilst the route (noted after via) states the neighbouring path to take.
The ARP tables of each router are updated with whatever device is directly connected to itself.

Since the running-config for each router is very long, their individual screenshots won't be posted here. But to note the differences between each router's configuration is just their respective direct connections (through their interfaces) and static routing configurations.

Questions

1. What is the role of the TTL field and why is it important?

   The role of the Time-To-Live (TTL) field in an IP header is to reduce lost network traffic. Sometimes packets can get stuck in low-speed paths or in looping paths and they do not reach their destination within the intended window of time. While the TTL field is not important for either of the end devices which can both retransmit on the failure to acknowledge a packet has been received, it is helpful for the whole network as it removes this lost data from the network, freeing up bandwidth in the network for other communications.

2. What are the three main characteristics of Layer 3 that you learn in this lab?

   The three characteristics of the Layer 3 OSI model are Addressing, Routing, and Fragmentation (or Flow control).
   Examples of addressing in this lesson included learning about IP addresses and MAC addresses.
   Examples of routing in this lesson included seeing the pathing through the simple network's devices in Cisco Packet Tracer.
   Examples of flow control in this lesson included looking at the fragmentation of packets in Wireshark and seeing the different fragmentation flags in the IPv4 header.

3. What is the main role of routers in a network?

   The router discovers the topology of the network so that flow control and can be established through the discovery of optimal paths between devices and through adaptation to high traffic areas, bottlenecks, and failing connections. The role of the router is to control the connections between the devices on the network and assigns routes for them to transmit their information on based on its learning.

4. CONCLUSION

Wireshark and Cisco Packet Tracer are great tools for learning about simple network tasks such as routing and addressing. From our discoveries during these lab exercises we have been able to so visual representations of addressing, routing, and fragmentation and other flow control measures such as the TTL and fragmentation flag fields in the IP headers. Another feature of the lab was learning about ARP table which are a method of storing MAC and IP pairs that have been discovered from devices connected to a network. The ARP table is useful for saving bandwidth by reducing the need for broadcast messages on the network as the intended recipient can be sent information directly if its IP and MAC addresses have been stored correctly. In these lab sessions we saw visual representations of IP and MAC addresses many times. We learned the IP addresses come in both forms IPv4 and IPv6, the former of the two is a 32-bit address that uses the decimal format, and the latter is a 128-bit address using hexadecimal format instead. And we learned that the MAC address is a 48-bit ethernet address in the hexadecimal format specific to each physical device. This means that the devices distributor and name can often be found by cross-referencing its MAC address with listed documentation.

5. REFERENCES

[1] AnandK@TWC, "How to clear ARP Cache in Windows 11/10," *The Windows Club*, Feb. 24, 2025. https://www.thewindowsclub.com/how-to-clear-arp-cache-in-windows (accessed Apr. 13, 2025).
[2] "How to Find Your IP Address on Windows, Mac, iPhone, & Android," *How to Find Your IP Address on Windows, Mac, iPhone, & Android*. https://www.avg.com/en/signal/find-ip-address
[3] "Lab Exercise -ICMP." Available: https://kevincurran.org/com320/labs/wireshark/lab-icmp.pdf
[4] Little, P., and Cardenas, M., "Use of Studio Methods in the Introductory Engineering Design Curriculum," Journal of Engineering Education, Vol. 90, No. 3, 2001, pp. 309-318.
[5] Nunally, J., Psychometric Theory, 2nd ed., New York, N.Y.: McGraw-Hill, 1978.
[6] "What is MSS (maximum segment size)?," *Cloudflare.com*, 2025. https://www.cloudflare.com/en-gb/learning/network-layer/what-is-mss/ (accessed Apr. 13, 2025).
[7] "What is a network switch? | Switch vs. router," *Cloudflare.com*, 2024. https://www.cloudflare.com/en-au/learning/network-layer/what-is-a-network-switch/
[8] "Wireshark Lab 6: Internet Protocol," *Maxwell Sullivan: Computer Science*, Mar. 26, 2013. https://maxwellsullivan.wordpress.com/2013/03/26/wireshark-lab-6-internet-protocol/