Lab Book: Purpose of Hardware and Protocols Associated with Networking Computer Systems

Contents

Purpose of Hardware and Protocols Associated with Computer Networks

Lab Objectives

1. Use CASBIT to explain the purpose of network devices utilised in a typical computer network.
2. Use CASBIT to explain the purpose of protocols used to support Internet services.

Lab Resources

- Packet Tracer (PT) 6.01 or higher.
- Computer with Windows OS, XP or higher.

Task 1 – Explain the Purpose of Network Protocols and Protocol Models

1. At the human level, some communication rules are formal and others are simply understood based on custom and practice. For network devices to successfully communicate, a network protocol suite must describe precise requirements and interactions. Networking protocols define a common format and set of rules for exchanging messages between devices.

2. List some common rules that you follow when having a conversation with your friends. Do the rules change if you had the same conversation with your teacher?

3. A protocol suite is a set of protocols that work together to provide comprehensive network communication services. A protocol suite may be specified by a standards organization or developed by a vendor.

4. Use the Internet to identify standards organisations involved with developing network protocols for use in LAN and WAN environments:

5. The protocols IP, HTTP, and DHCP are all part of the Internet protocol suite known as Transmission Control Protocol/IP (TCP/IP). The TCP/IP protocol suite is an open standard, meaning these protocols are freely available to the public, and any vendor is able to implement these protocols on their hardware or in their software. TCP/IP protocols are installed on a PC when a networking interface card (NIC) is fitted.

6. A layered **reference** model, such as the TCP/IP model, is often used to help visualize the interaction between various protocols. A layered model depicts the operation of the protocols occurring within each layer, as well as the interaction of protocols with the layers above and below each layer:

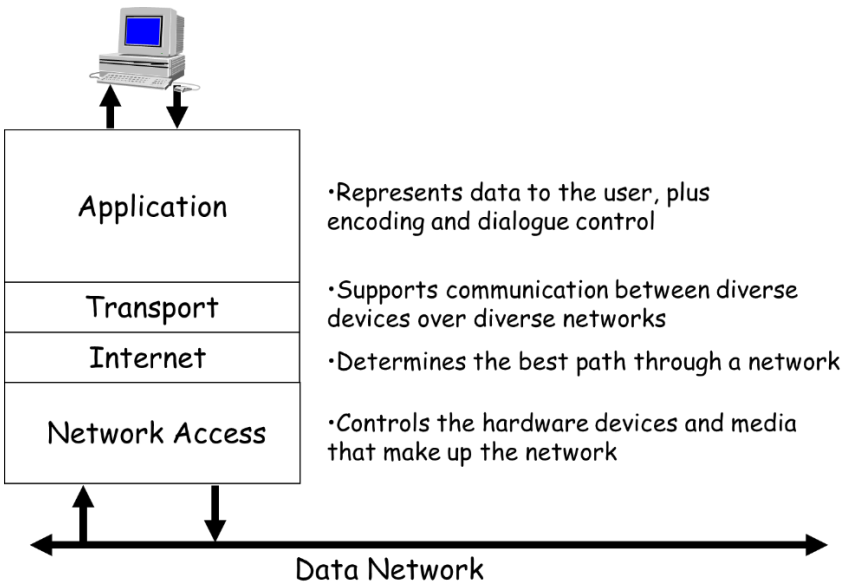

Figure 1 – TCP/IP Reference Model

7. A reference model is not intended to be an implementation specification or to provide a sufficient level of detail to define precisely the services of the network architecture. The primary purpose of a reference model is to aid in clearer **understanding** of the functions and processes involved.

8. As data from a computer operating system is passed down through the protocols defined in the TCP/IP model on its way to be transmitted across the network media, various protocols add information to it at each level. This is commonly known as the **encapsulation** process. There are many different protocol defined for use within the TCP/IP reference model, as shown in Fig2:

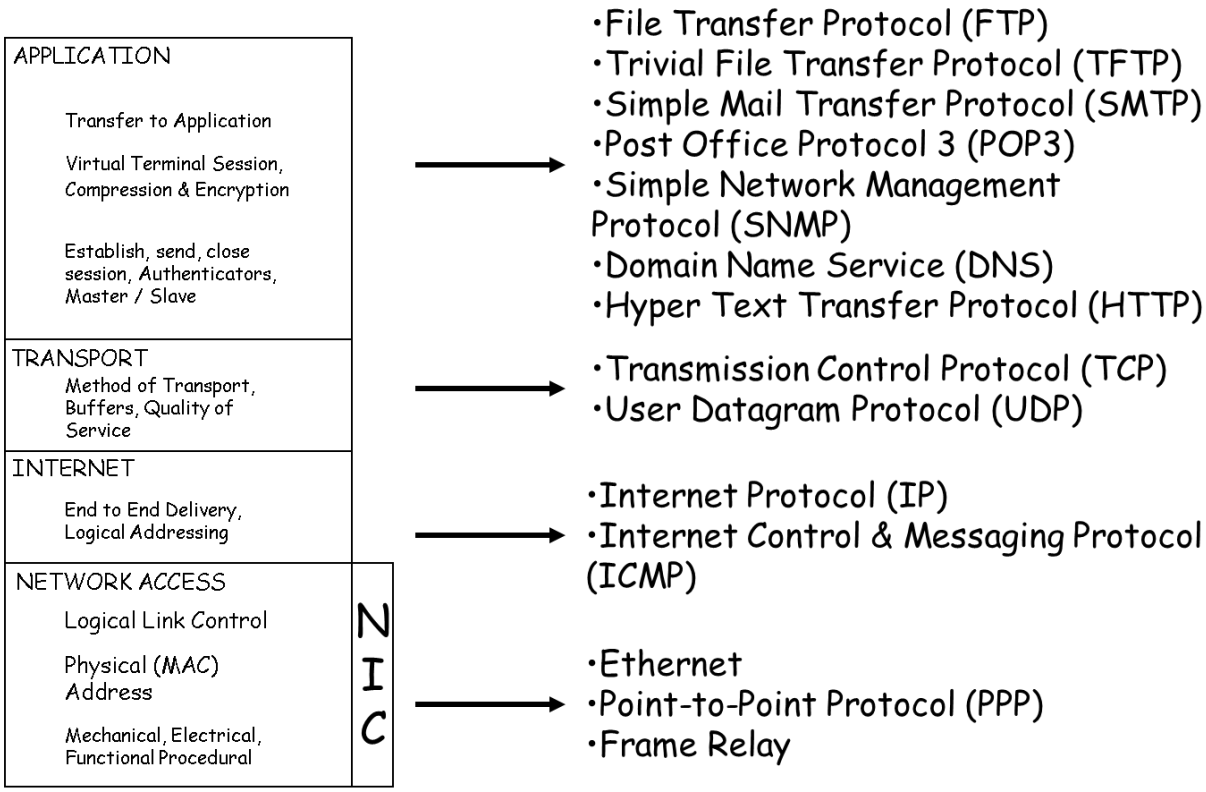

Figure 2 – TCP/IP Reference Model – Example Protocols

9. The form that a block of data takes at any layer of the TCP/IP model is called a Protocol Data Unit (PDU). During encapsulation, each succeeding layer encapsulates the PDU that it receives from the layer above in accordance with the protocol being used.

9. At each stage of the process, a PDU has a different name to reflect its new appearance. Although there is no universal naming convention for PDUs, they are typically named according to the protocols of the TCP/IP suite.

10. To explore the encapsulation process, open the CASBIT.pkz Packet Tracer file and select *Simulation* Mode, then the *Event List* on the left of the simulation tab, and ensure that Hyper Text Transfer Protocol (HTTP) packets are selected for capture. HTTP is the protocol used to transfer web pages from the web server on which they are stored, to the web browser on Bob's PC.

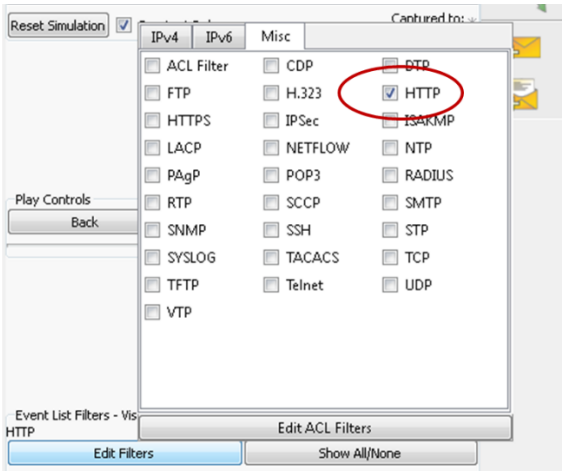

Figure 3 – HTTP Filter Configuration

11. Go to Bob's PC and select the *Desktop* tab, and then the *Web Browser*. Type the address of the Birmingham City University Web Site, *www.bcu.ac.uk* into the address bar and select the *Go* button.

12. Select *Auto Capture/Play* on the Simulation tool bar, and watch the HTTP packets traverse the network. From the *Simulation Panel*, select the 4th packet and the then select the *Outgoing PDU Details* tab:

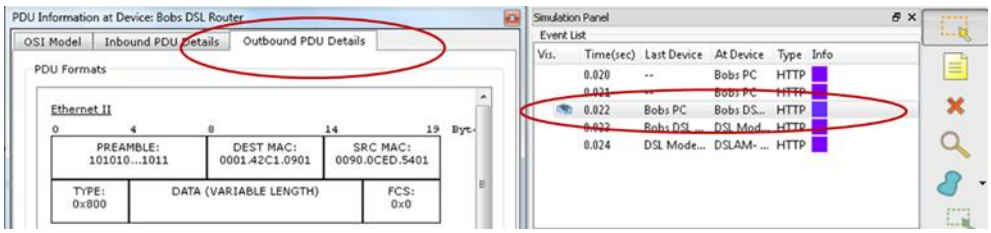

Figure 4 – Protocol Inspection

13. This provides a pictorial representation of all the various network protocols involved in the transmission of this particular packet, displayed as they pass through the layers of the TCP/IP reference model:

14. The **Application** layer is shown at the bottom of the PDU Details tab. Starting at the bottom at '1', you can see that HTTP has issued a request for a web page (GET/HTTP/1.1)., confirming that HTTP is an application layer protocol, that services web requests from web browser and web server software installed on the computer.

15. The **Transport** layer is shown at '2', with the HTTP data being encapsulated by TCP, within the TCP *Data (Variable)* field, creating a PDU called a **segment**. TCP is responsible for chopping up data into chunks suitable for transmission in IP packets, but in this case, the HTTP data is quite small, so this is not necessary, and only one segment is created. The TCP encapsulation process adds additional information to allow the protocol to perform transport layer functions.

16. The **Internet** layer is shown at '3', with the TCP segment being encapsulated by IP, within the IP *Data (Variable Length)* field, creating a PDU called a **packet**. This adds additional information such as a destination and source IP addresses, which allow the packet to be routed through IP networks.

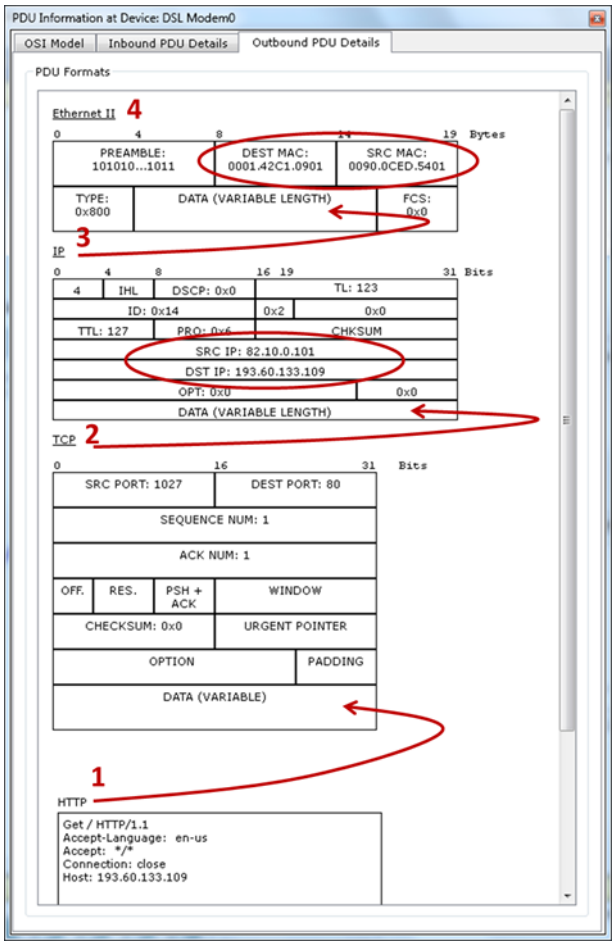

Figure 5 – Detailed Protocol Inspection

17. The **Network Access** layer is shown at '4', with the IP packet being encapsulated into the *Data (Variable Length)* field of an Ethernet (it's actually FastEthernet) **Frame**, which adds information such as destination and source MAC addresses for use by Ethernet switches. **Do not** close this tab down, as you will refer back to it again.

18. The protocols used within the TCP/IP reference model all add protocol-specific information within the PDU that they create, to allow them to perform specific network **tasks** required to send data between computers on an IP network:

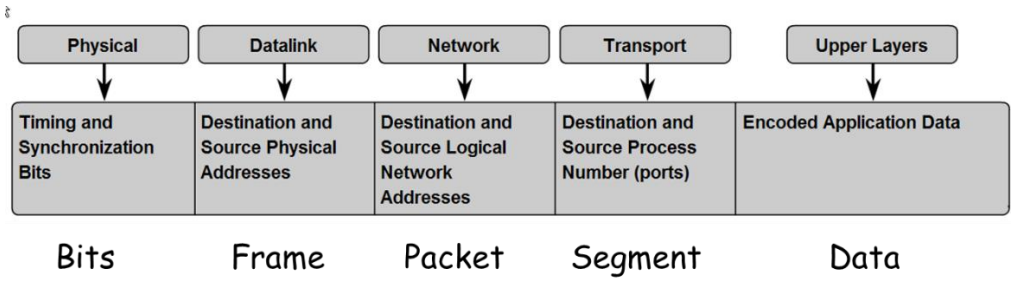

Fig 6 – Protocol PDUs and Network Functions

Task 2 – Transport Layer Investigation

19. The two most common Transport layer protocols of TCP/IP protocol suite are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Both protocols manage the communication of *multiple* operating system programs (applications) that are all trying to send their data across a network using the same NIC on the computer. The differences between the two protocols are the *specific* functions that each protocol implements.

20. Sending some types of data (for example, a streaming video) across a network, as one complete communication stream, could use all of the available bandwidth and prevent other communications from occurring at the same time. It also makes error **recovery** and **retransmission** of damaged data difficult.

21. **Segmenting** the data into smaller chunks enables many different communications, from many different users, to be interleaved (multiplexed) on the same network. Segmentation of the data by transport layer protocols also provides the means to both send and receive data when running multiple applications at the same time on a computer.

22. Without segmentation, only one application would be able to receive data. For example, a streaming video, the media would be completely consumed by the one communication stream instead of shared. You could not be able to receive emails, chat on instant messenger, or view web pages while also viewing the video.

23. To identify each segment of data, the transport layer adds to the **segment** a header containing binary data. This header contains fields of bits. It is the values in these fields that enable different transport layer protocols to perform different functions in managing data communication.

24. The transport layer is also responsible for managing the reliability requirements of data communication, so that if segments are lost or damaged during their trip across the network, another copy of the damaged segment can be sent. Different applications have different transport reliability requirements. For example, when accessing web pages using HTTP and emails using POP3/SMTP, reliability is very important (you don't want web pages or emails full of errors), so TCP is used as the Transport layer protocol to encapsulate theses protocols.

25. Other types of data, such as voice or video are time-sensitive, and require a simple transport layer protocol like UDP, that allows multiple data paths through the computer NIC, but without the delays caused by providing a reliable service.

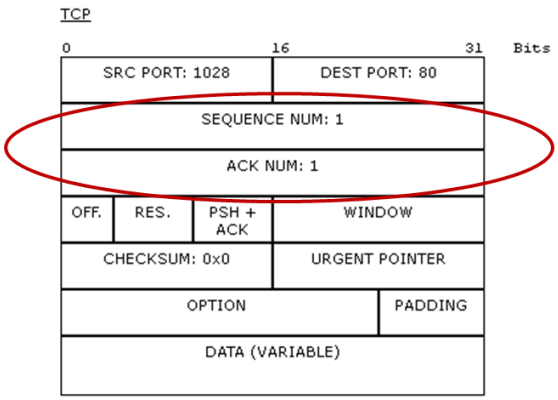

Fig 7 – TCP Segment – Sequence Numbering

26. Returning to Packet Tracer, re-examine the Outbound PDU Detail tab opened in step 12. Note the *Sequence Number* and *Ack* (for acknowledgement) *Number* fields. These are used to identify segments that are part of the same data conversation, and allow segments that are damaged or missing to be re-sent. Damaged segments are identified by information contained in the *checksum* field.

27. When talking to someone, you are using a type of sequencing when you ask someone to repeat something they said because you didn't hear them, or didn't understand what was said.

28. If a computer using TCP sends segments too rapidly, it may overwhelm the receiving computer, or the network connecting them, resulting in their loss or damage. Although TCP can use its Sequence and Acknowledgement numbers to re-transmit these segments, it is not a particularly efficient way of transmitting data. Instead, computers using TCP use the **Window** field to agree the amount of data contained in segments that they are capable of handling, a process called *Windowing*.

29. When talking to someone, you are using a type of windowing when you ask them to slow down when they are speaking too quickly!

30. UDP is designed to transfer data segments as quickly as possible, so it does not employ sequencing or windowing, as these mechanisms can actually slow down the transmission of data. It does mean however, that UDP does not provide a reliable service, and any damaged or lost segments cannot be retransmitted.

31. Both TCP & UDP use **port** numbers to keep data from different application layer protocols separate as they pass through the transport layer. When you sent the HTTP request from Bob's web browser to the BCU web server, TCP on Bob's PC identified itself using a unique port number, and requested web services from the server using a well-known port number used only by web servers.

32. Run the HTTP Packet Tracer simulation test again, as explained in steps 10 to 12. Allow all the packets to travel to the server and back to Bob's PC. Select a PDU going to the server, and a PDU returning to Bob's PC, and examine the TCP port numbers of each:

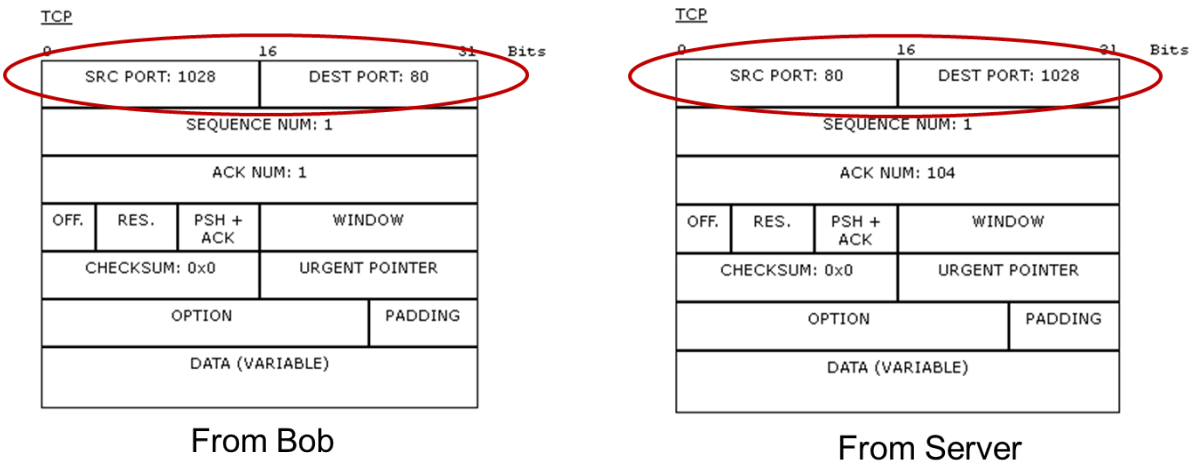

Fig 8 – TCP Segment – Port Numbering

33. Examining the TCP segment sent by Bob's PC, it has identified this data communication as originating on port 1028 (your values may be different), which means that HTTP is active on port 1028. The TCP segment from Bob's PC has also identified a destination port on the web server of 80, which is the well-known port that HTTP running on web servers is active on. If you examine the a TCP segment returning from the server, you'll see that the port numbers have swapped – the source is port 80, and the destination is 1028 (your value may be different, but it will be the same as the source port used by Bob's PC).

34. Use the Internet to identify the well-known server ports for the following application layer protocols:

    POP3: _____

    SMTP: _____

    HTTPS: _____

    DNS: _____

35. You can confirm the port numbers used by POP3 and SMTP by sending an email between Ann and Bob using the Packet Tracer. Open the *CASBIT.pkz* file and select *Simulation* Mode, then the *Event List* on the left of the simulation tab, and ensure that POP3 and SMTP packets are selected for capture:

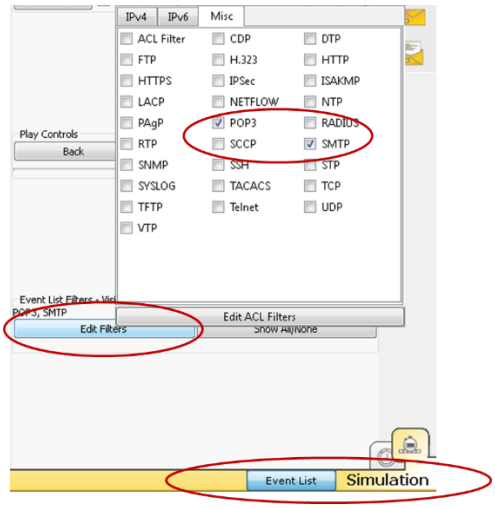

Figure 9 – POP3/SMTP Filter Configuration

36. Write an email to Bob (bob@isp.co.uk) from the email client on Ann's PC, and select *send*, then use the *Auto Capture/ Play* button to run the simulation – the email will be sent to the ISP mail server using SMTP:

37. Examine any packet that Ann's PC sent to the email server, and examine the *Outbound PDU Detail*s tab. Does the destination port display the well-known port used by SMTP servers?

    _____

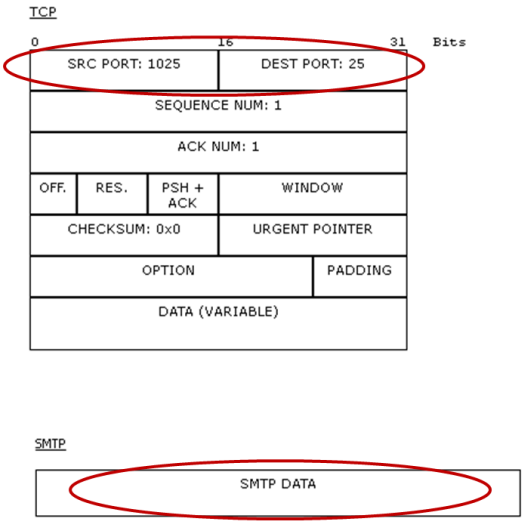

Figure 10 – TCP Segment Carrying SMTP

38. From Bob's PC, open up the email client and retrieve the email from Ann using simulation mode – remember to use the *Auto Capture/ Play* button to run the simulation. Examine any packet that is sent from the email server to Bob's PC, and examine the *Outbound PDU Details* tab. Is the application layer using POP3, and is it using the correct well-known port for POP3?

39. Because application layer protocols use a different source port to identify current data communication sessions, the Transport layer protocols TCP and UDP can direct arriving segments to the correct instance of a particular protocol. For example, you may have opened many different web browser tabs, but each will be using a different **source** port, so arriving HTTP data segments will be directed to the correct tab based on their **destination** port.

Task 3 – Transport Layer Challenge Activity

40. Explain how you would run a simulation test in Packet Tracer to confirm the well-known port used by HTTPS:

Task 4 – Internet Layer Investigation

41. The **Internet** layer provides services to allow computers to exchange data across a network. To accomplish this, the Internet network layer uses four basic processes:

    - **Addressing Computers:** In the same way that a phone has a unique telephone number, user devices such as computers must be configured with a unique address for identification on the network. An end device with a configured Internet layer address is referred to as a **host**, regardless of the type of device.

    - **Encapsulation:** The Internet layer receives *segments* from the transport layer. In a process called encapsulation, the Internet layer adds header information, such as the address of the source (sending) and destination (receiving) hosts. Adding the Internet layer information creates a Protocol Data Unit (PDU) called a *packet*.

    - **Routing:** The Internet layer provides services to direct packets to a *destination* host on another network. To travel to other networks, the packet must be processed by a *router*. The role of the router is to select paths for and direct packets toward the *destination* host in a process known as *routing*. A packet may cross many routers before reaching the destination host.

    - **De-encapsulation:** When the packet arrives at the Internet layer of the destination host, the host checks the address in the packet header. If the destination address within the header matches its own address, the header is removed from the packet. This process of removing headers from lower layers is known as de-encapsulation. After the packet is de-encapsulated by the Internet layer, the Transport layer PDU (segment) is revealed, and can be processed by the appropriate Transport layer protocol (TCP or UDP).

42. Unlike the Transport layer, which *manages* the transmission of data belonging to Application layer protocols running on each host (e.g. HTTP, POP3, SMTP), Internet layer protocols simply transport packets between hosts, with no regard to the protocols they are carrying or the transmission media they are running across.

43. The Internet Protocol (IP) is the Internet layer protocol used within the TCP/IP protocol suite. IP was designed as a protocol with a low overhead, providing only the functions that are necessary to deliver a packet from a source to a destination over a network or a series of connected networks. The protocol was not designed to track and manage the flow of packets, as this is the job of Transport layer protocols such as TCP.

44. There are currently two different versions of IP used within computer networks, IPv4 and IPv6. Use the Internet to provide an example of an address used by each protocol:

    _____

    _____

45. What is the advantage of using IPv6?

46. The IPv4 address consists of a 32-bit **binary** number, divided into four 8-bit segments called octets. For ease of use, the IPv4 address is usually represented in a dotted **decimal** notation, with each octet given as a decimal number, for example 192.168.0.1. This address allows you to identify the **network** on which a device is located, and its unique **identity** within that network.

47. In order to allow you and your network devices to be able to figure out the network and unique identity information, all IP addresses require a **subnet mask**, which is used to mark the binary bits of the address that identify the network address part of the IPv4 address. For example, a computer with an IP address of 192.168.0.1 with a subnet mask of 255.255.255.0 is on network *192.168.0*, and has a unique identity on that network of *1*.

48. Work out the network and unique identities for hosts using the following IPv4 addresses and subnet masks:

    10.10.10.2 255.0.0.0

    172.16.15.12 255.255.0.0

49. Each host device that wishes to communicate using the IP protocol must have a unique IP address, so that packets can be routed to the correct destination. Imagine if there were two houses in your street that had the same number, the postman would not be able to decide which house to deliver letters addressed to that particular number.

50. For IPv6, the range of available addresses is so large, there is no problem assigning unique address to all host devices that wish to communicate using IPv6. However, the older IPv4 protocol has a much smaller range of addresses, and most of these have already been assigned, so techniques have been developed to make best use of the addresses remaining.

51. One IP address conservation technique use **private** IP addresses, which are assigned in three ranges – 10.x.x.x, 192.168.x.x and 172.16.x.x – 172.31.x.x. Bob and Ann's wireless routers are using a well-known private address range (anything starting with 192.168.x.x), which cannot be used on the Internet. When they send data packets from their networks, the source address used is converted to the single **public** address assigned to the home router Internet interface, which has been assigned using DHCP by the ISP. This process is called **Network Address translation** (NAT). A public address is unique, so no other device in the world can use it.

52. Check each of Ann and Bob's devices to ensure that the have received an IPv4 address via DHCP from the wireless router LAN pool by selecting each device in turn and opening up the *desktop* tab to review the *IP configuration*:

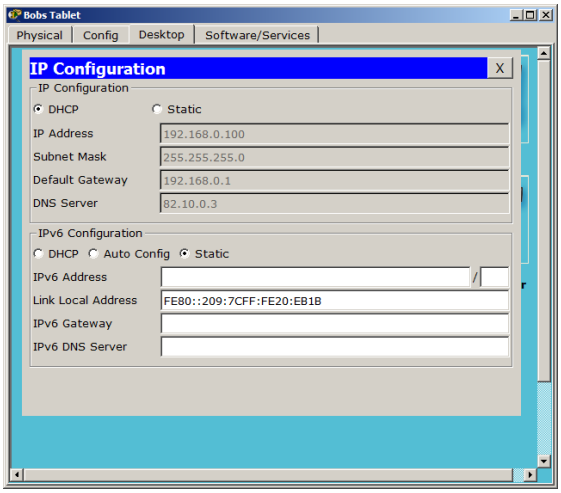

Figure 11 – Host DHCP Configuration

53. Return to Bob and Ann's home router's GUI and select the *Status* link. What IP address has been assigned to each routers Internet interface and what ISP default gateway is being used?

54. Select *Simulation* Mode, then the *Event List* on the left of the simulation tab, and ensure that Hyper Text Transfer Protocol (HTTP) packets are selected for capture. HTTP is the protocol used to transfer web pages from the web server on which they are stored, to the web browser on Bob's PC.

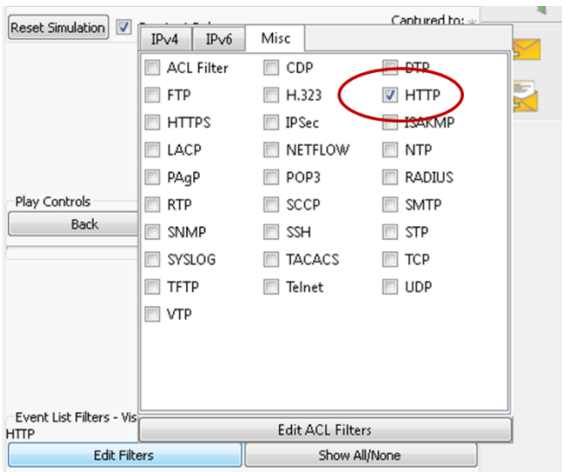

Figure 12 – HTTP Filter Configuration

55. Go to Bob's PC and select the *Desktop* tab, and then the *Web Browser*. Type the address of the Birmingham City University Web Site, *www.bcu.ac.uk* into the address bar. Select the *Go* button.–

56. Select Auto Capture/Play on the Simulation tool bar, and watch the HTTP packets traverse the network. Examine the first packet in the *Simulation Panel* Event List – what is the source IP address? (Clue – the packet came from Bob's PC):

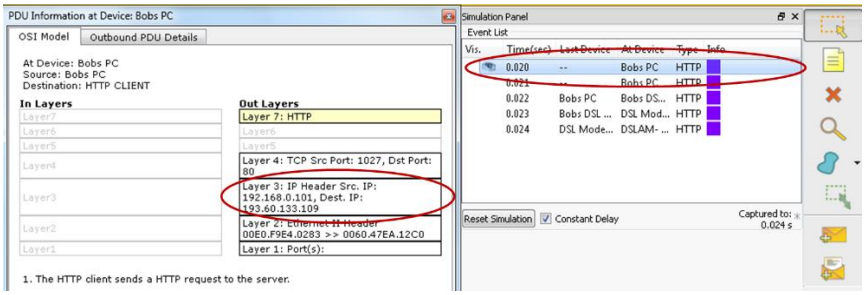

Figure 13 – Source Address Inspection

57. Examine the 4th packet shown in the Event List (any packet that has passed through Bob's home router will suffice), which should have performed a NAT translation – what is the source address of the packet?

---

58. You should see a new source address related to Bob's home router, which means that NAT is operational and effectively 'hiding' the private IP addresses assigned to Bob's device behind a single public IP address, assigned to the home router by the ISP using DHCP. This can then be routed through the Internet to the destination address. The destination address should be that of the bcu.ac.uk web server.

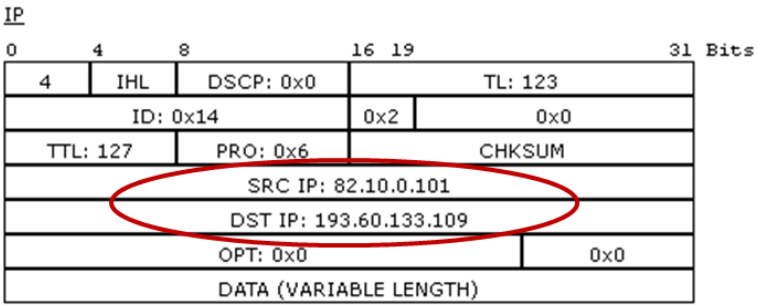

Figure 14 – IPv4 from Bob's PC to bcu.ac.uk

59. Now examine one of the packets returning from bcu.ac.uk to Bob's PC. The Destination and source IP addresses should now have swapped over, to allow the packet to be routed to Bob's Network:

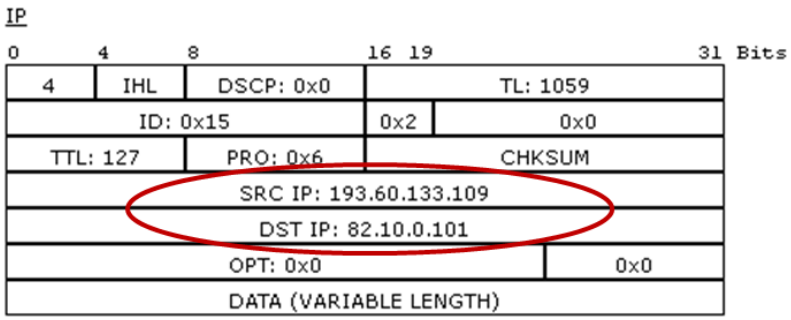

Figure 15 – IPv4 from bcu.ac.uk to Bob's PC

60. In order for routers to be able to forward packets to a *destination* network, they must build a routing table, which identifies the direction in which to switch the packets. You can view the routing table of the *Internet Router* using the Packet Tracer **Inspect** tool:

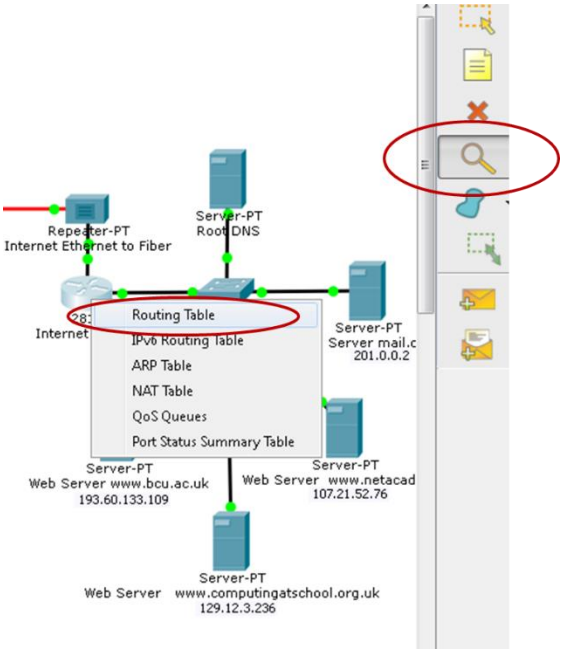

Figure 15 – IPv4 from bcu.ac.uk to Bob's PC

61. The routing table shows that the router has learnt about all the various networks in the Packet Tracer topology, and has decided upon the best interface to forward IPv4 packets based on their destination address:

Routing Table for Internet Router

Type	Network	Port	Next Hop IP	Metric
C	82.10.0.0/16	FastEthernet0/0	---	0/0
C	107.21.0.0/16	FastEthernet0/1.10	---	0/0
C	129.12.0.0/16	FastEthernet0/1.20	---	0/0
C	193.60.133.0/24	FastEthernet0/1.30	---	0/0
C	200.0.0.0/24	FastEthernet0/1.40	---	0/0
C	201.0.0.0/24	FastEthernet0/1.50	---	0/0

Figure 16 – Routing Table

62. The routing table contains an entry for the network 82.10.0.0 255.255.0.0 network, which, if you look at Fig 14 is the network address that Bob's home router is converting all Bob's devices source IP addresses to using NAT. Thus the Internet Router thinks that the best path back to Bob's network is using its local interface FastEthernet0/0 (Fa0/0). If you allow your cursor to hover over the links attached to the Internet Router, it will display their identity. Does Fa0/0 lead back toward Bob and Ann's networks?

63. The routers decide which is the best path to a destination network based on a numerical value called a **metric**. The lower the metric value associated with an interface, the more likely a router is to forward packets out of that interface. Most metrics are calculated using an algorithm that considers the speed of the paths between routers, with faster paths being preferred, thus they receive lower metric values.

64. In the example above, all the paths have a metric value of 0/0, which is the lowest value possible, as the router is directly connected to the networks in the routing table, and doesn't have to go through another router to reach them.

Task 5 – Internet Layer Challenge Activity

65. Open the *CASWANT.pkz* file and use the *inspect* tool to view the IPv4 routing table on R6. What is the metric for the 82.10.0.0 255.255.0.0 destination network, and which interface will R6 use to forward packet to it? Why do you think R6 chose this path?

66. Select R4, and go to its physical tab and switch it off. Wait a minute, and then examine the R6 routing table again – what has happened to the metric for network 82.10.0.0 0.0.255.255, and which interface is R6 using to get there?

Task 6 – Network Access Layer Investigation

67. The Network Access layer prepares a *packet* for transport across the local media by encapsulating it with a header and a trailer to create a **frame**. To accomplish this, the Network Access layer uses four basic processes:

    - **Addressing Computers:** Typically provides a **physical** address that is used to identify the computer on the local network. This address is used for the delivery of frames between devices in the local network, and cannot be used for end-to-end deliver of data across many networks – that is the job of packets using IP.

    - **Encapsulation:** The Network Access layer receives *packets* from the Internet layer. In a process called encapsulation, the Network Access layer adds header information, such as the physical address of the source (sending) and destination (receiving) hosts. Adding the Network Access layer information creates a Protocol Data Unit (PDU) called a *frame*.

    - **Switching:** The Network Access layer provides services to direct frames to a *destination* host on the local network. In an Ethernet LAN, frames can be switched between local computers using an Ethernet *switch*. The role of the switch is to forward frames to local host computers based on the *destination* physical address contained in the frame. Multiple switches can be connected together to form large networks, that can forward frames at very high speeds (often faster than routers can forward packets). To send data to another network, frames are switched to the local router, which will process the IP destination address from the packet encapsulated in the frame to allow it to make a routing decision.

    - **De-encapsulation:** When the frame arrives at an interface, the computer checks the physical address in the frame header. If the destination address within the header matches its own address, the header is removed from the frame. This process of removing headers from lower layers is known as de-encapsulation. After the packet is de-encapsulated by the Network Access layer, the Internet layer PDU (packet) is revealed, and can be processed by the appropriate Internet layer protocol (IPv4 or IPv6).

68. In a TCP/IP network, all Network Access layer protocols work with IP at the Internet layer. However, the actual Network Access protocols used depends on the **logical** topology of the network and the physical media used. The wide range of physical media available within computer networking means that there is a wide range of Network Access layer protocols. Within the *CASBIT.pkz* topology, only the (Fast) Ethernet Network Access layer protocol is used within the various computer networks.

69. Open the *CASBIT.pkz* Packet Tracer topology, and select Ann's PC desktop. In order to discover the physical address assigned to the PC, it is necessary to open up a **command line** window:

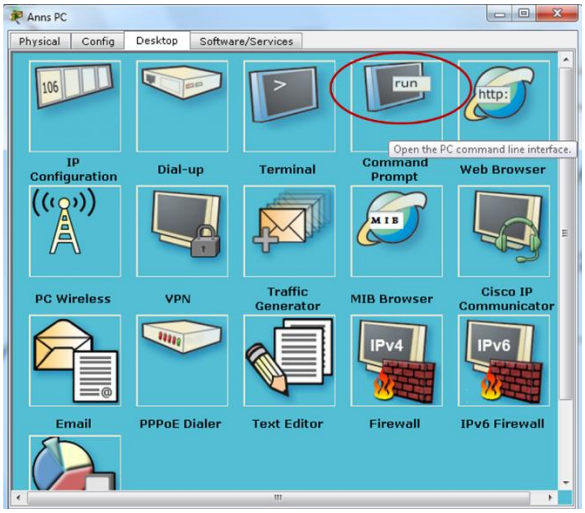

Figure 17 – Command Line Interface

70. Once the command line is open, type in **ipconfig /all** to display all the addressing information that is applied to the PCs NIC:

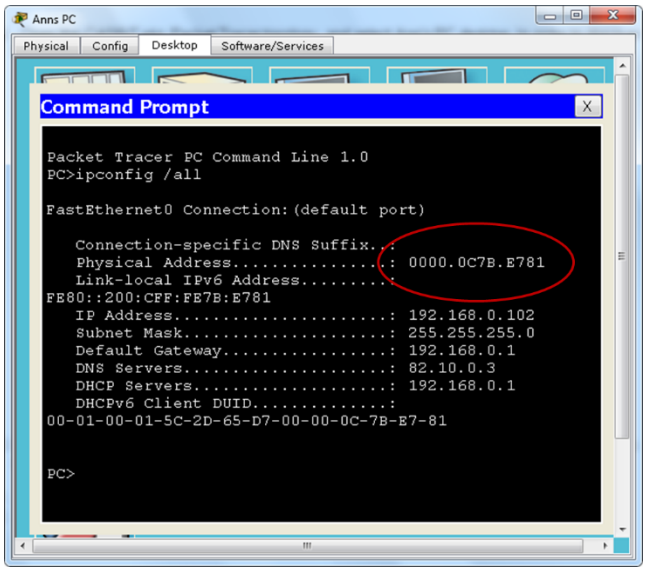

Figure 18 – NIC Address Configuration

71. The physical address shown is assigned to the NIC card by the manufacturer, and is often called a 'burned in address'. Because all the computers in this topology are using FastEthernet NICs, the physical address is in a format appropriate for the protocol, and is referred to as a **Media Access Control** (MAC) Address. This term is only used when referring to Ethernet physical addresses – other types of Network Access protocols use different names for the physical address that they use.

72. You can see the MAC addresses being used to deliver frames between PCs in the local network by running a **ping** test from the command line that you have opened on Ann's PC. First, write down the IP address and MAC address from Ann's PC and laptop:

    PC IP Address: _____

    PC MAC Address: _____

    Laptop IP Address: _____

    Laptop MAC Address: _____

73. From Ann's PCs command line interface, use the ping command to send a test IP packet to the laptop (your address for the laptop might be different to the one shown below in Fig 19, so use the one you have written down):

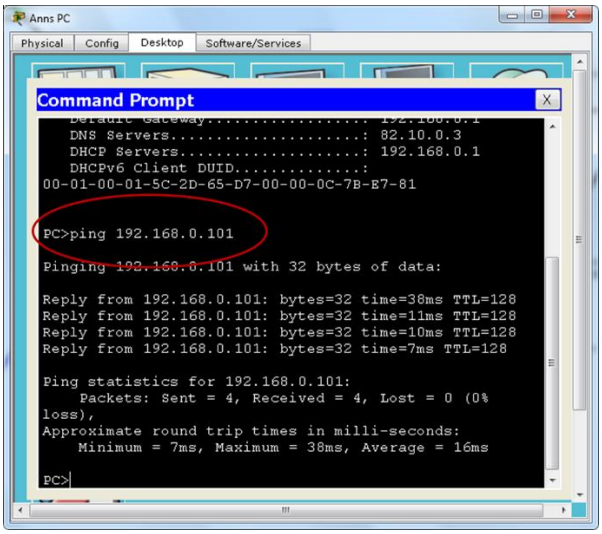

Figure 19 – Ping test to 192.168.0.101

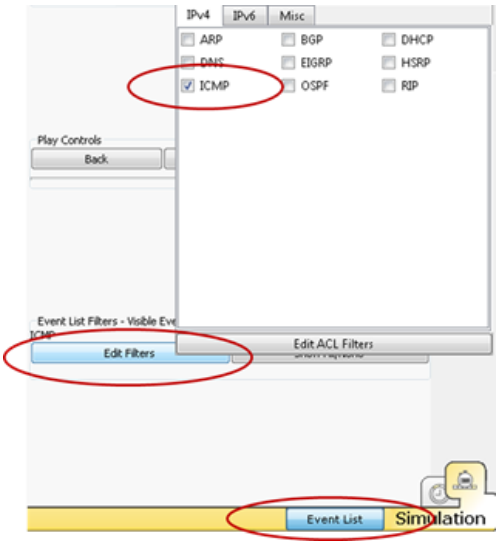

Figure 20 – Edit Filter for ICMP Capture

74. The ping test send 4 test IP packets from the PC, and the laptop replies to each one, which is shown as '**Reply from 192.168.0.101'** in the command line interface. Now that you know that the test works, try it again, but this time in *simulation* mode, to allow you to capture and analyze the packets. You will need to edit the filter list to capture ICMP (Internet Control and Messaging Protocol), which is used to send and receive test IP packets:

75. Enter the same ping command to send test packets to Ann's laptop, and then select the Auto Capture Play button to start them on their way. Once you have seen one '**Reply from 192.168.0.101'** on the PC command line interface, you can stop the simulation capture. You can now look at the captured ICMP traffic in the *Outbound PDU* tab, and compare the MAC addresses used by the Ethernet frames to carry them, and confirm that they are the same as you wrote down above in step 72:

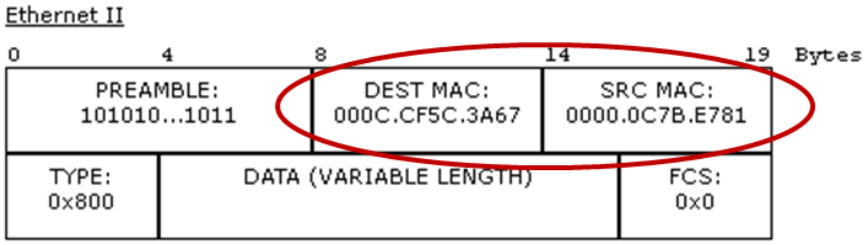

Figure 21 – Frames from Ann's PC

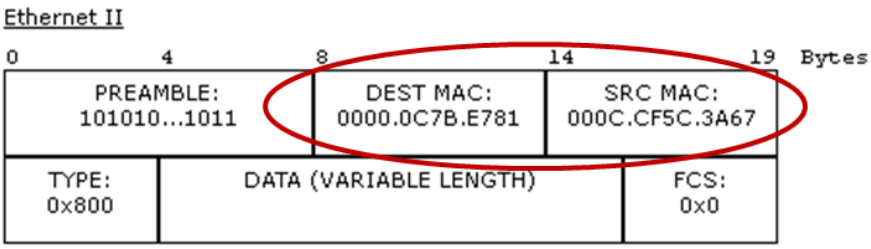

Figure 22 – Frames from Ann's Laptop

76. Examining the frames that you have captured from Ann's **PC**, which MAC addresses are being used as the source and destination?

77. Examining the frames that you have captured from Ann's **Laptop**, which MAC addresses are being used as the source and destination?

Task 7 – Network Access Layer Challenge Activity

78. In simulation mode, send a ping test to the bcu.ac.uk server at IP address 193.60.133.109 from Ann's PC. Look at the second packet that is sent from Ann's PC, and write down the source and destination MAC addresses:

Source MAC Address:

Destination MAC Address:

79. Can you decide which device is using the MAC address that is shown as the destination address? Explain why it is being used instead of the MAC address of the bcu.ac.uk server.