Task 4.1C: Above and Beyond Pass

Application layer

Discussing the concept of TLS and its handshake process based on the Wireshark screen grabs.

- At first, I opened my internet browser and cleared the cache. This helped me to avoid any previous data to mingle with the newly captured packet.

- I launched Wireshark and chose my network interface (such as Wi-Fi) to begin a fresh packet capture.

- I then entered an HTTPS URL, such as https://www.discoverourtown.com, into my browser.

- I ended the Wireshark packet capture when the webpage had fully loaded.

- Evaluate the handshake in TLS.

- I limited my Wireshark view to TLS packets by using the filter tls. I searched for the initial packets used for handshakes, which typically contain:

- The handshake begins with a ClientHello message sent by the client, which is my browser. TLS versions and cipher suites that are supported are examples of this.

- In response, the server selects the TLS version and cipher suite from the client's choices and sends back a ServerHello message.

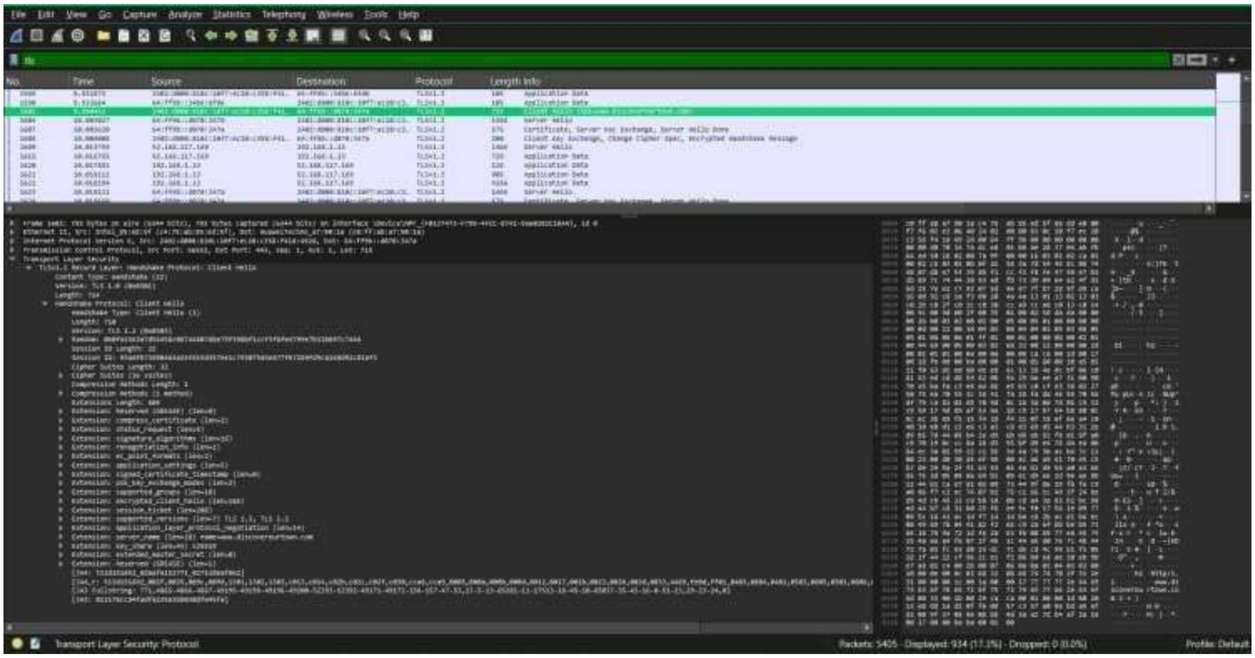
- Certificate: To verify its identity, the client receives a digital certificate from the server.

- The server signals that its portion of the handshake is complete with the message ServerHelloDone.

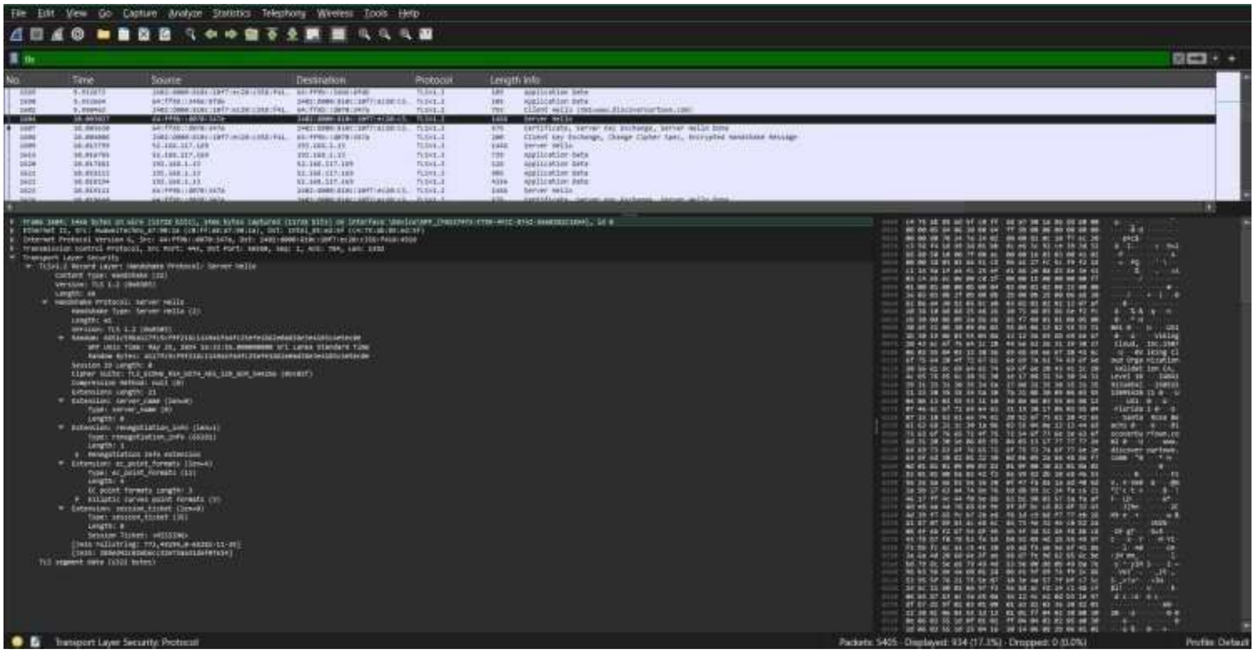
- ClientKeyExchange: Using the public key of the server, the client transmits a pre-master secret that has been encrypted.

- This message is sent by the client and server to transition to encrypted communication (ChangeCipherSpec).

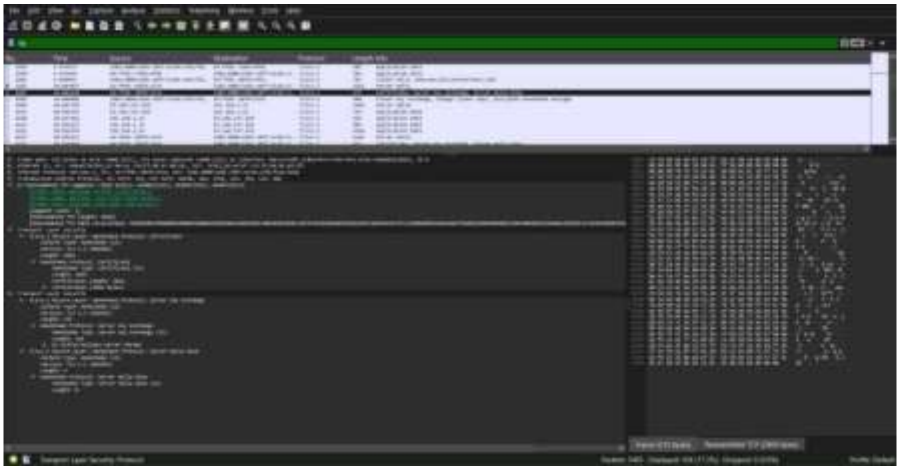
- Completed: To indicate that the handshake is finished, both sides send a completed message.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tls
No. Time Source Destination Protocol Length Info
Transport Layer Security: Protocol
Packets: 5405 · Displayed: 934 (17.3%) · Dropped: 0 (0.0%)
Profile: Default

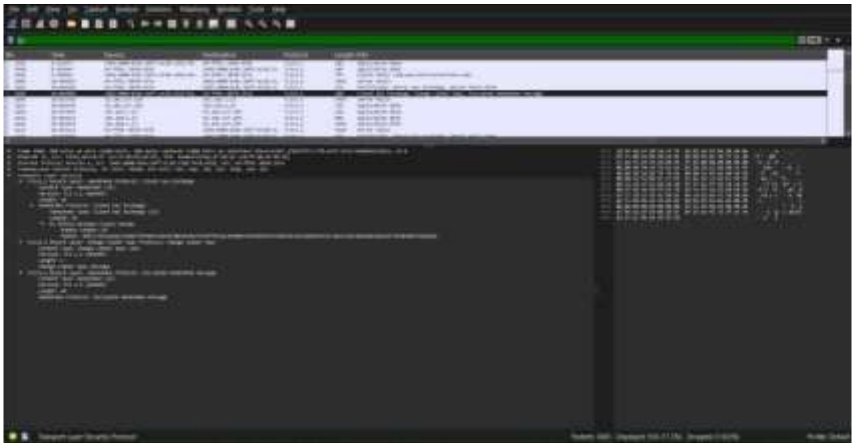
Hello, Client
I located the message ClientHello. It contained information on cipher suites and available TLS versions (such as TLS 1.2 and 1.3).


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tls
No. Time Source Destination Protocol Length Info
Transport Layer Security: Protocol
Packets: 5405 · Displayed: 934 (17.3%) · Dropped: 0 (0.0%)
Profile: Default

Hi, Server
The ServerHello message was recognized by me. The server selected the cipher suite and TLS version in addition to sending a random value.



Accreditation
I found the Certificate message indicating that the client had received the server's public key certificate. This was employed to confirm the identification of the server.



Exchange of Client Keys
The ClientKeyExchange message was located. A pre-master secret was encrypted by the client using the server's public key before being transferred to the server.

1. Can you use Wireshark to analyze HTTPS? Give an explanation for your response. If so, give examples of how we can accomplish it. If not, is there another way we might analyze HTTPS without sacrificing security?

- Because HTTPS traffic is encrypted, it is difficult to analyze directly. However, the TLS handshake and the first few unencrypted packets can be recorded by Wireshark.
  As proof, I displayed the TLS handshake packets that I had recorded using Wireshark.

Active Class 3: It's always DNS, No It's 192.168.1.2 (Module 2)

Describe email

- ✓ Email communication mostly uses the Simple Mail Transfer Protocol (SMTP) application layer protocol.
- ✓ For dependable data transfer, SMTP depends on TCP (Transmission Control Protocol), the underlying transport layer protocol.

- The fundamental actions required to send an email from user A to user B are
    - ✓ A Mail User Agent (MUA) such as Outlook, Thunderbird, or a web-based email client is used by User A to write emails.
    - ✓ Using the SMTP protocol, the MUA sends an email by connecting to the SMTP server, which is usually offered by the email service provider.
    - ✓ Using a number of Mail Transfer Agents (MTAs), the SMTP server receives the email and forwards it to the recipient's SMTP server (if different).
    - ✓ The email is received by the recipient's SMTP server, which then saves it in the recipient's mailbox.
    - ✓ With the use of a MUA and protocols like POP3 (Post Office Protocol 3) and IMAP (Internet Message Access Protocol), User B retrieves the email from their mailbox.

- Some of the instructions for sending an email that is defined by the SMTP protocol includes HELO, defining the connection, MAIL FROM defining the sender, RCPT TO, defining the receiver of the message and DATA for sending the body of the message. The two main protocols that are used in getting the emails from the mail server are the POP3, and IMAP. POP3 downloads mails to the local device while IMAP allows for using the mailbox from different devices at a go. Some other parameter of the email are TLS/SSL for security while transmitting the email and DNS (Domain Name System) for identifying the email server address.

Summary for Module 2 active classes 2 and 3

- First, I examined the details of the TLS, a cryptographic system that need to be used to ensure that communications over the internet is secure Secondly, I outlined the TLS handshake, which is a process of creating a secure communication channel between a client and server by negotiating. When deciphering the status of messages that were sent, I employed Wireshark and analyzed packet captures to understand a series of messages that occured during the TLS handshake. That, in turn, helped me gain additional knowledge of the subsequent negotiation of encryption keys and the onset of data encryption as a result. The protocol gave people a good insight into how data transmitted over any network is protected, who authorize it and how that data stays secret.

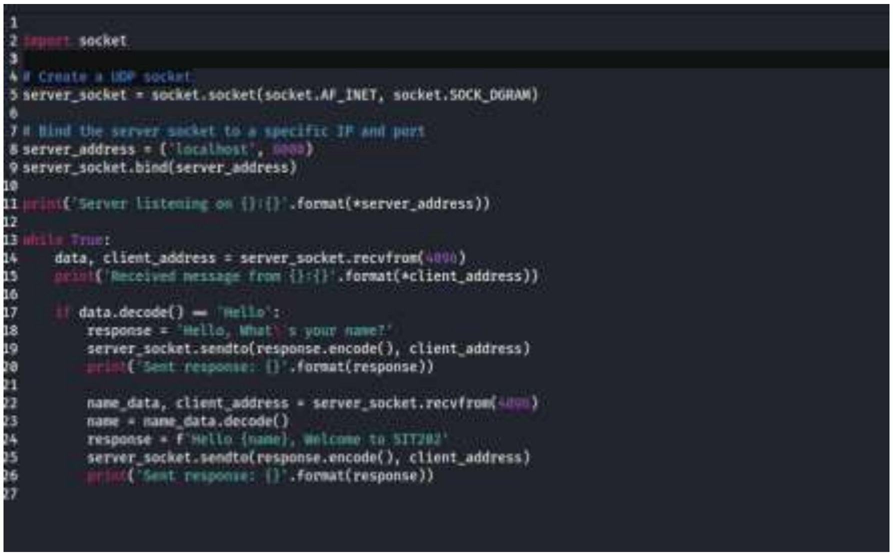
- Besides TLS, I researched how email works since it is one of the most internet essential tools. The basic protocol associated with the sending of emails was the next discussed one which was known as the Simple Mail Transfer Protocol or SMTP for short. We also looked at the design of the Internet and the Transport Control Protocol (TCP) that is used at the transport layer together with the SMTP. It was easy to describe the client-server notion along with the SMTP servers when giving the barest of the steps required to transfer an e-mail message from one user to the other.

Reflection of Module 2 active classes 2 and 3

- It was rather possible to understand the basics of computer communications and Internet security studying the example of email and TLS. Studying TLS handshakes gives an understanding of the machinery behind security connections and shows the importance of the encryption to protect information. From this practical lesson, I enhanced my understanding of the concept of cybersecurity and clearly saw how it is important to use secure encryption measures to prevent cyber-attacks.
- In addition, the email explanation helped untangle the complex process of sending messages through the internet. By being able to consider the process at length I was able to gain insight into how intricate the fundamental infrastructure behind email services are. Thus, this investigation revealed the value of stable protocols such as SMTP and TCP for maintaining user to user free communion regardless of the number of emails services involved.
- Therefore, the analysis of email correspondence and TLS promoted the development of knowledge about protocols at the Internet level and their roles, ensuring the security and productivity of the data exchange process. Through nice of these core concepts, I now have better understanding on how the modern digital communication networks work, allowing me and us, to be more informed as we try to shape the Internet technology realm as it continues to advance.

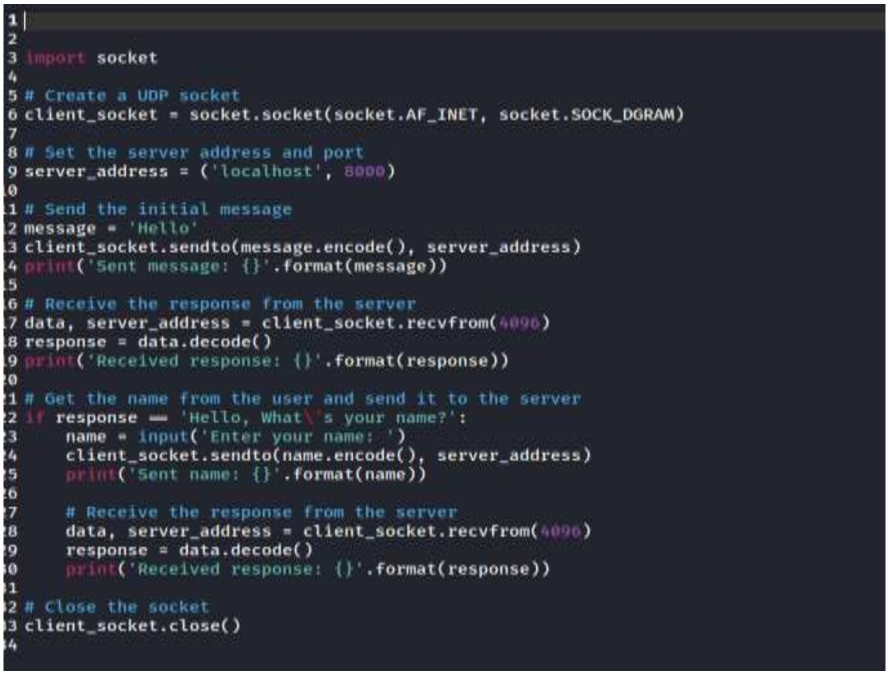
Active class 4: UDP - Unreliable Data Protocol? (Module 3)

Server.py



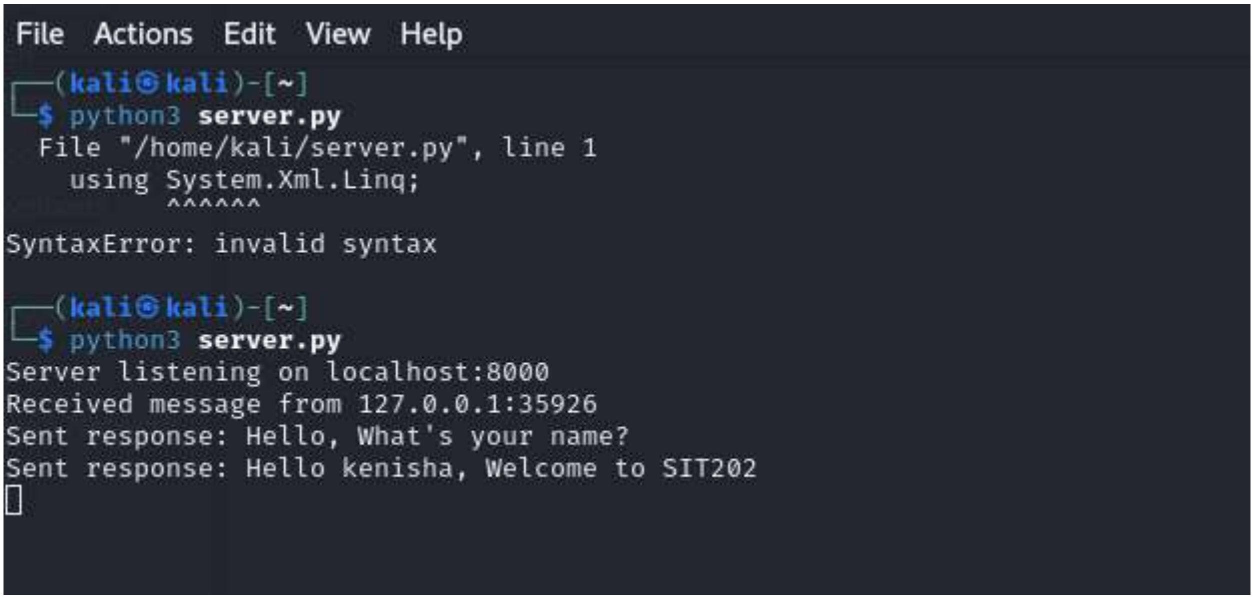
```python

import socket

# Create a UDP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the server socket to a specific IP and port
server_address = ('localhost', 8000)
server_socket.bind(server_address)

print('Server listening on {}:{}'.format(*server_address))

while True:
    data, client_address = server_socket.recvfrom(4096)
    print('Received message from {}:{}'.format(*client_address))

    if data.decode() == 'Hello':
        response = 'Hello, What\'s your name?'
        server_socket.sendto(response.encode(), client_address)
        print('Sent response: {}'.format(response))

        name_data, client_address = server_socket.recvfrom(4096)
        name = name_data.decode()
        response = f'Hello {name}, Welcome to SIT202'
        server_socket.sendto(response.encode(), client_address)
        print('Sent response: {}'.format(response))

```

Client.py



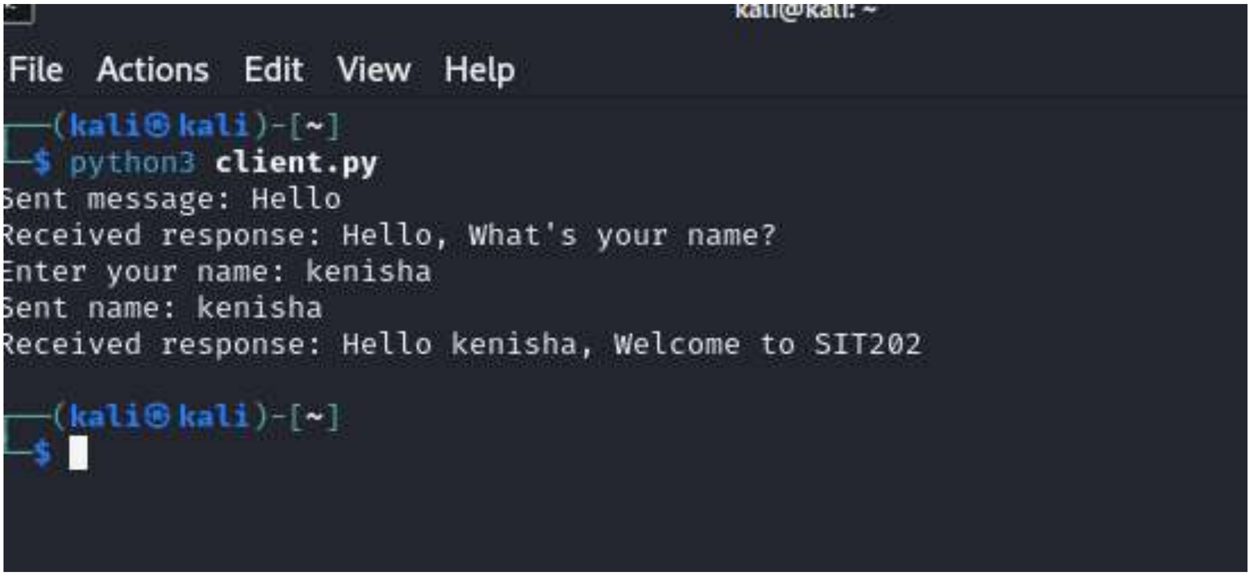
```python


import socket

# Create a UDP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Set the server address and port
server_address = ('localhost', 8000)

# Send the initial message
message = 'Hello'
client_socket.sendto(message.encode(), server_address)
print('Sent message: {}'.format(message))

# Receive the response from the server
data, server_address = client_socket.recvfrom(4096)
response = data.decode()
print('Received response: {}'.format(response))

# Get the name from the user and send it to the server
if response == 'Hello, What\'s your name?':
    name = input('Enter your name: ')
    client_socket.sendto(name.encode(), server_address)
    print('Sent name: {}'.format(name))

    # Receive the response from the server
    data, server_address = client_socket.recvfrom(4096)
    response = data.decode()
    print('Received response: {}'.format(response))

# Close the socket
client_socket.close()

```

Server Side Terminal



```
┌──(kali㉿kali)-[~]
└─$ python3 server.py
  File "/home/kali/server.py", line 1
    using System.Xml.Linq;
          ^^^^^^
SyntaxError: invalid syntax

┌──(kali㉿kali)-[~]
└─$ python3 server.py
Server listening on localhost:8000
Received message from 127.0.0.1:35926
Sent response: Hello, What's your name?
Sent response: Hello kenisha, Welcome to SIT202
```

Client side Terminal



```
┌──(kali㉿kali)-[~]
└─$ python3 client.py
Sent message: Hello
Received response: Hello, What's your name?
Enter your name: kenisha
Sent name: kenisha
Received response: Hello kenisha, Welcome to SIT202

┌──(kali㉿kali)-[~]
└─$ 
```

Active class 5: How can I transport my application data reliably? (Module 3)

1. **Stop-and-Wait**
   - ✓ In stop and wait, the sender transmits a single packet, and waits before transmitting the next packet till the receiver sends an acknowledgement or ACK. In case the acknowledgment for the packet is not received in a given duration, the packet is resent. This method tends to produce high latency because the sender must wait for an acknowledgment after each packet therefore it may not be efficient.
   - ✓ Look at this as sending one package at a time and not sending the next one until you receive response (like a signed confirmation). In terms of speed, it is rather slow, however the messages' meaning is quite comprehensible. I'm sure folks can well imagine having a delivery person stand by for each and every box! This is good for simple, low-bandwidth situations where slowness isn't an issue it is very important to note that unlike some protocols or systems, the use of this is good for simple, low-bandwidth situations where slowness isn't an issue.

2. **Go-Back-N**
   - ✓ Go-Back-N restricts the maximum number of packets sent over a connection to the window size which allows the sender to transmit a number of packets before receipt of an acknowledgement. The error forcing the sender to retransmit the lost packet and any number of subsequent packets after an error occurrence is a factor. This technique is more efficient than Stop and-Wait but its disadvantage is that if failures occur frequently then several packets will be transmitted unnecessarily.
   - ✓ It means that the delivery person can pick up as many items as the person wishes in this case without a possibility of the decision being denied. The person has to come back and bring all subsequent packages in the unlikely event that the first package was misplaced. While this is a lot faster than stop-and-wait, a high number of packets having been dropped wastes bandwidth. This is a good compromise between simplicity and optimal results for average data transmission.

3. **Selective Repeat**
   - ✓ As with Go-Back-N, Selective Repeat enables a large number of packets to be transmitted before an acknowledgment to them is expected. If any error is detected, only the individual packet that was lost or was in error is resent, not all subsequent packets. Selective Repeat is better than Go-Back-N because this protocol minimizes unnecessary retransmission, which makes it efficient once errors are present. It is somewhat like arranging packages by an intelligent mailbox.
   - ✓ Full advantage can be taken of a postman who brings packages as the mailbox will categorize them. This is the most efficient way because the packets which were not received at the first go will be resent. But the implementation of is the most challenging. This is best in environments where and when there is need to have fast and reliable data transfers and where efficiency is of essence.

Summary for Module 3 active classes 4 and 5

- The change to the Python software was to add onto a prior client-server UDP communication setting I had established. The older client.py and server.py scripts had to be updated in order to allow for a more Speaking as well as Listening in a real time conversation. The words “Hello” and” Hello, what is your name? Were included into the protocol. Greeting are used and then the customer provides the server with his or her name and receives a welcome.

- Concerning the protocols, the discussion focused on how the fundamental three; Stop-and-wait, Go-back-N and Selective Repeat were different from each other. All protocols’ specific working performance features, benefits and drawbacks were considered. I have also questioned which of these is the optimal one in terms of usability, speeds at which the protocols operate and the ability to recover from errors.

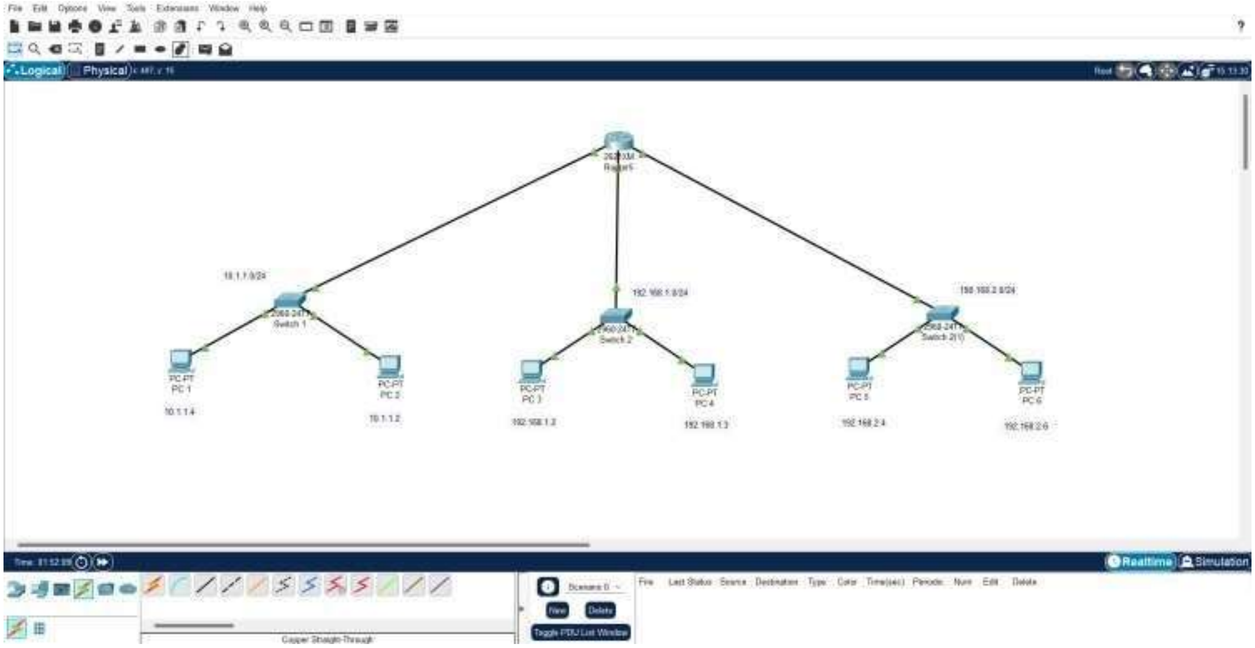
- As will be seen in the subsequent section on TCP congestion control, understanding it was crucial in preserving network stability and reliability. The discussion was centered on role of congestion control technologies in avoiding network congestion and how such technologies are important in issues to do with resource allocation and system failure. While, the congestion control mechanism of TCP includes one of its flexible features since it is capable of changing the transmission rates proportional to the network conditions.

Reflection of Module 3 active classes 4 and 5

- Replacing Python programs offered valuable information of the subject of sockets through stressing the client-server relation and message transmitting techniques. Although this was a shallow practice exercise, comprehension was increased by constructing a basic network communication system through the implementation of UDP sockets. Therefore, by analyzing these protocols for their concrete parameters and uses in practice, the authors’ comprehension of them was improved as well. This way, understanding the strengths and weaknesses of the protocol, and comparing the two and contrasting, this study was able to provide a better appreciation of its effectiveness and inefficiencies primarily on how important it is to choose the right protocol for a particular form of communication needs.

- Apart from filling the knowledge gaps about TCP congestion control, it outlined the difficulties inherent to resource management in distributed settings. Enhanced comprehension of the principles of network communication result from analyzing the congestion control algorithms to explain the role of TCP in managing reliable and highly efficient data flow throughout networks.

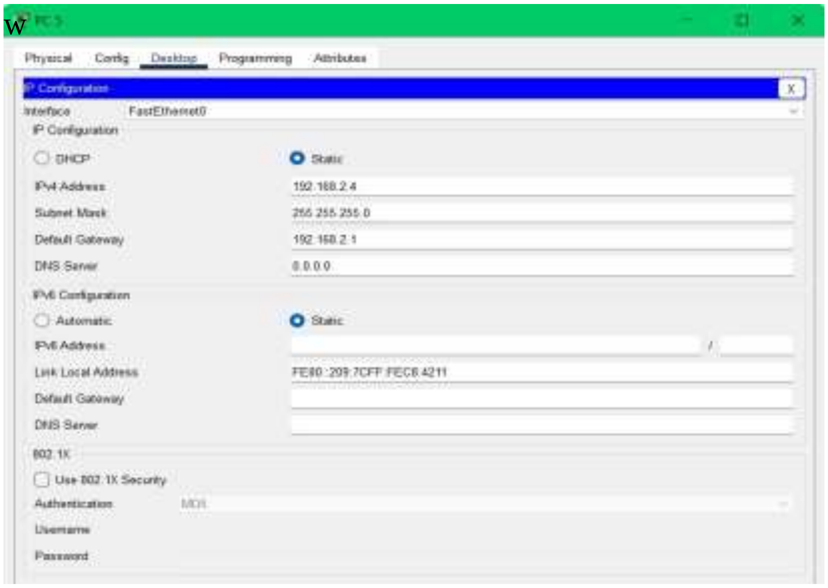
Active class 6: How do I get from My home to Your home - Journey of an IP packet (Module 4)



- PC5 and PC6 were connected to the newly formed LAN by me. Assigned to it is the subnet mask 192.168.2.0/24. The following is the LAN3 configurations that we have: I configured the new LAN with the IP addresses as a static one. According to the assignment, the IP address 192.168.2.4 was assigned to the PC5.

PC5

W



PC5

Physical | Config | Desktop | Programming | Attributes

P Configuration

Interface: FastEthernet0

IP Configuration

○ DHCP ● Static

IPv4 Address: 192.168.2.4

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.2.1

DNS Server: 0.0.0.0

IPv6 Configuration

○ Automatic ● Static

IPv6 Address: /

Link Local Address: FE80::209:7CFF:FEC8:4211

Default Gateway:

DNS Server:

802.1X

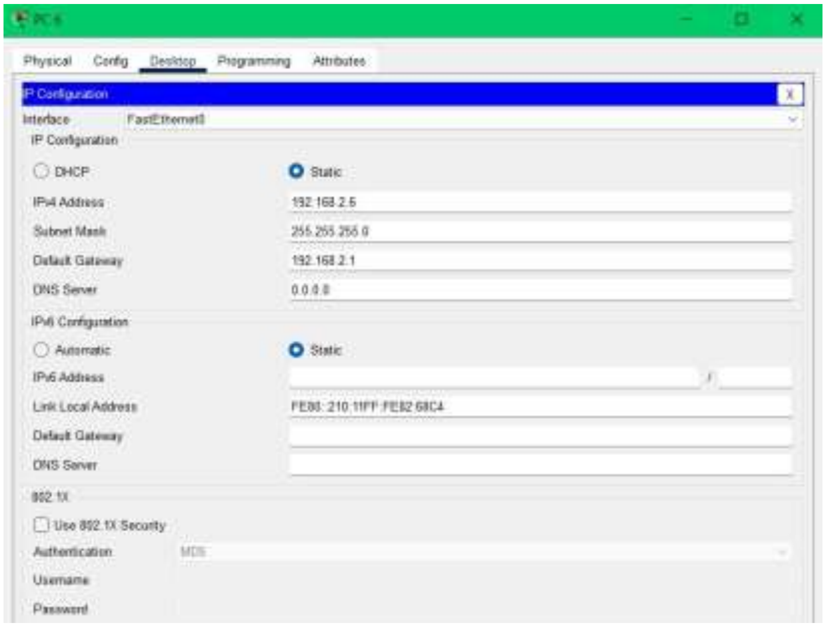
□ Use 802.1X Security

Authentication: MD5

Username

Password

PC6



PC6

Physical | Config | Desktop | Programming | Attributes

P Configuration

Interface: FastEthernet0

IP Configuration

○ DHCP ● Static

IPv4 Address: 192.168.2.5

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.2.1

DNS Server: 0.0.0.0

IPv6 Configuration

○ Automatic ● Static

IPv6 Address: /

Link Local Address: FE80::210:11FF:FE82:68C4

Default Gateway:

DNS Server:

802.1X

□ Use 802.1X Security

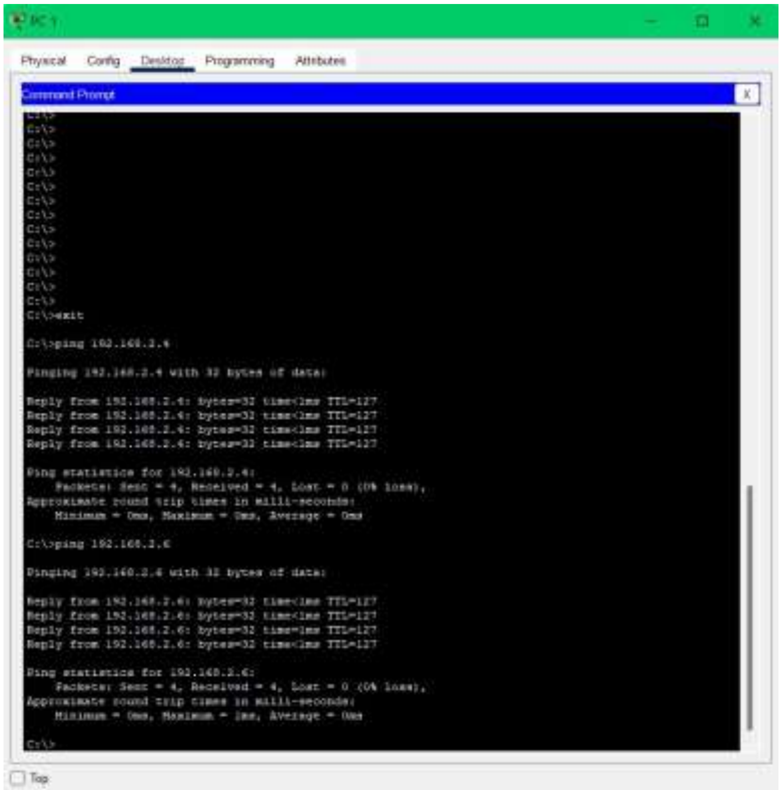
Authentication: MD5

Username

Password

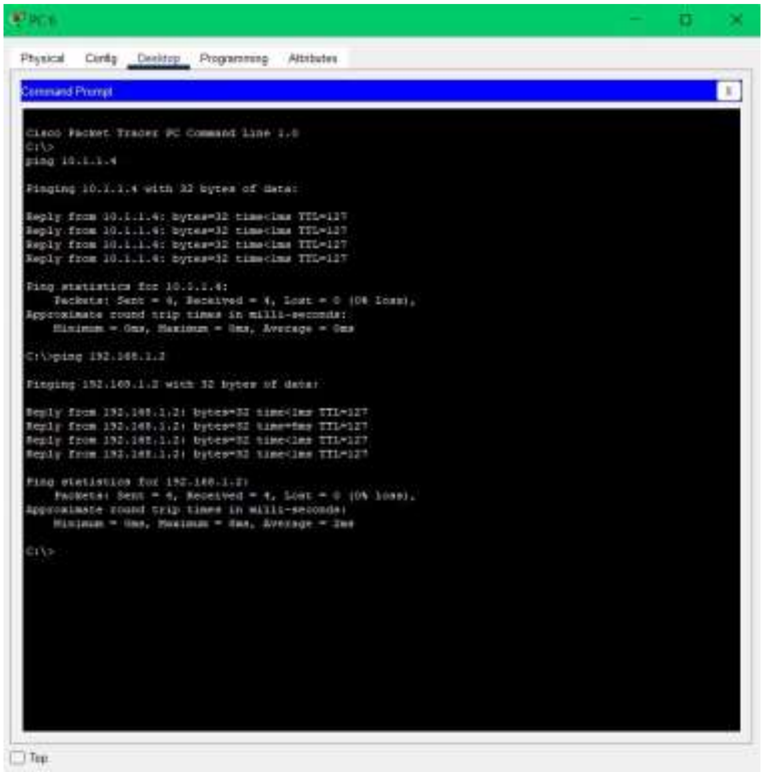
- ✓ To ascertain a connectivity to this LAN3, I had to add more switch and change a router from 1941 to 2621XM. I had to add the NM-1E after switching the router because it has one 10BaseT Ethernet port that can be used to connect a LAN backbone for supporting 24 synchronous/asynchronous ports or six PRI connecting to ISDN lines. Furthermore, I also plugged in the PC6 into the LAN 3 network.

Here’s ping from PC1 to PC5 and then from PC1 to PC6



```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>exit

C:\>ping 192.168.2.4

Pinging 192.168.2.4 with 32 bytes of data:

Reply from 192.168.2.4: bytes=32 time<1ms TTL=127
Reply from 192.168.2.4: bytes=32 time<1ms TTL=127
Reply from 192.168.2.4: bytes=32 time<1ms TTL=127
Reply from 192.168.2.4: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.2.6

Pinging 192.168.2.6 with 32 bytes of data:

Reply from 192.168.2.6: bytes=32 time<1ms TTL=127
Reply from 192.168.2.6: bytes=32 time<1ms TTL=127
Reply from 192.168.2.6: bytes=32 time=1ms TTL=127
Reply from 192.168.2.6: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.2.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

To ensure connectivity from LAN3 to other LANS, I pinged from PC6(LAN3) to PC1(LAN1) and PC3(LAN2)



```
Cisco Packet Tracer PC Command Line 1.0
C:\>
ping 10.1.1.4

Pinging 10.1.1.4 with 32 bytes of data:

Reply from 10.1.1.4: bytes=32 time<1ms TTL=127
Reply from 10.1.1.4: bytes=32 time<1ms TTL=127
Reply from 10.1.1.4: bytes=32 time<1ms TTL=127
Reply from 10.1.1.4: bytes=32 time<1ms TTL=127

Ping statistics for 10.1.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time=6ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 6ms, Average = 1ms

C:\>
```

Summary for Module 4 active class 6

- In order to perform the “Above and Beyond” activity for Active Class 6, I added a new PC into the network created during Activity 3 connecting PC5 with the IP address 198.168.2.4. To achieve this, the current network connectivity has to be reflected on the IP address of PC5. Moreover, I employ the use of the “ping” tool to check whether the PC1 and PC5 were well connected. But for this experiment I was able to have live practical experience in extending the network capabilities As a result of this venture I was able to understand IP addressing and network topology at a deeper level.

Reflection of Module 4 active class 6

- This assignment also offered participants a chance to apply and build on the networking concepts identified in the Active Class 6 activities. Performing an addition of devices and checking the network connectivity also helped to enhance my knowledge on the issue of IP addressing, subnetting as well as the communication protocols in the network. This task required me to put my critical and analytical skills into operation while I systematically tried to understand the structure of network architecture as well as its configuration. Taking everything into consideration, the above and Beyond challenge was a good practice of the actual operation and made me better understand the network layer.

Summary and reflection of Module 2

Summary

- In Module 2, I learned different network application architectures include the Peer-to-Peer (P2P) and Client-Server models. I learned their issues, channels of passing information and information structures. Web services and databases have enormous advantages in Client-Server architecture focusing the centralized control for clients to request services. P2P networks, on the other hand, distribute work among the peers, thus offering scalability at the same time that they pose administrative problems.

- The lesson dedicated significant time to talking about sockets and interprocess communication (IPC), or the way by which processes in a network can convey with one another. Another aspect was the HTTP protocol, which rules client-server communication in the internet space. The important ideas which were highlighted were the effects they have on the performance of connection over the internet, HTTP on the other hand is stated less as being stateless, the differences between persistent and non-permanent connections.

- It also included the Domain Name System (DNS) and its structure, and basic concept of Recursive and iterative approach in DNS. Network safety was also discussed in detail including aspect of encryption and protection against threats including DNS spoofing and man-in-the-middle attacks.

Reflection

- By the time I was editing the notebook, I understood that the lessons on HTTP protocol, Client-Server architectures, and P2P were rather helpful. I benefited from the knowledge I acquired on host communication and data transmission in the module which I have background information from. Enhanced my technical competency of the job connected the gap of theory and practical by doing works of live projects like socket programming and DNS setting. In consideration of all the benefits and drawbacks, the module offered a good foundation for subsequent research in networking, security and application.

Module 3 Summary and Reflection

Summary

- I gained a clear understanding of the TCP/IP idea in module 3, with reference to the Transport Layer role of delivering reliable data delivery between the networks. After comparing between UDP and TCP, I found that TCP guarantees reliable transmission through connection-oriented communication it employs flow management and congestion control and checking for transmission errors. However, UDP protocol offers connectionless services at higher speeds, making it ideal for use in real time clients that do not necessitate regular connection reliability such as on streaming services.

- In as much as the two elements were learned in the module, the distinctions between the Transport and Network Layers were also stressed. Process-to-process communication is designed by the Transport Layer and the delivery of the packets for the hosts is done through the Network Layer. I then considered connections in TCP and more explicitly the three ways handshake to establish a connection and the four ways handshake to close it. Also, I learned features of multiplexing and demultiplexing, which help in managing the communication flow between diverse applications.

- I realized from the practical demonstrations that while applications that require data integrity such as web browsing use TCP, applications that require speed such as online gaming or video streaming use UDP. Another lesson learned was getting to understand how TCP on manages retransmissions and how it comes up with the round-trip time (RTT) so as to be in a position to determine how networks manage to handle packet and delays.

Reflection

- Reflecting on the practice from Module 3, the most important thing to bear in mind was to grasp multiple roles that TCP and UDP have in data transfer thoroughly. I now have a better appreciation of how power flow control, congestion control, and retransmission of TCP is to maintaining data integrity used in programmes like file transfer programs as well as browsing. This is really significant when each packet should reach the target destination on time and in sequence. However, because the UDP puts more importance on conceiving speed with efficacy than on the costs of establishing the connection, it is most valuable in the real-time application such as gaming and streaming whereby a few talk loss is tolerable.

- As for my previous knowledge on the OSI model and this module's theme, which concerned the function of the transport layer in end-to-end communication, those concepts were overlapping significantly. In particular, it facilitated my level of grasping how discovered theoretical concepts as far as network communication is concerned, such as connection-oriented and connectionless, manifest in practical applications. Real life scenarios which were provided here like three way handshake of TCP or using UDP for video streaming gave me a clear understanding of these protocols.

- In my opinion the course team concentrated on this material for the reason that anybody who is attending network engineering or similar courses would require knowledge of, has to have some understanding of the transport layer. Writing networked applications always meant understanding how to receive and send trustworthy data, as well as how to work around performance and network problems. Also, such elements as round-trip time computations and demultiplexing along with multiplexing have enriched the set of the problems I can solve and prepared me for further networking subjects.

- Altogether, this module enabled me to fill the gap of the knowledge difference between book learning and the practice by providing me with the necessary background information and practical skills required to work with the network protocols in real life situations. I feel much more confident now, knowing how to approach the problems of the network and learning how to optimize the communication and ensure that the delivered services are both reliable and efficient. With such an understanding of networking and application development continuing to improve this is the first step in their better comprehension to me.

Summary and Reflection of Module 4

Summary

- Every Internet-connected device contains a network layer used to send and receive segments from a broadcasting host to a receiving host. In this Unit, I examined specific processes which are inherent in these steps in the Module 4. The two main network operations which were discussed in this module were routing and forwarding. Routing involves employing route finding techniques such as BGP and OSPF to determine the best path by which packets should take in order to get from source to the destination. Forwarding is the technique of forwarding the packet from input of the router to the output connection within a locale.

- Two other areas I explored were the data plane and the control plane. On one hand, the control plane is solely responsible for decision and network routing logic whereas on the other hand, it concerns with the actual forwarding of the incoming packets as well as the flow of traffic. It was useful for the module to understand a couple of network service models like Best-Effort Service whose delivery time, delivery order, or bandwidth is not guaranteed.

- Additionally, design of routers was discussed whereby internal components such as input ports and switches, switching fabrics, and the output ports were described in detail regarding how routers use them to manage receipt of packets, switches and transmission of packets. Indeed, the role of subnetting and IP addressing— activities that involves breaking the large network into small sub-networks as noted in the module by assigning and managing the IP addresses with the help of the subnet masks and CIDR notation— was stressed.

- I also learned how NAT works, through which a large group of inside devices can use one public IP address, and DHCP which is used to assign IP addresses dynamically. Also I researched on IPv6 as a possible replacement to the IPv4 address noting a use of 128 bit addressing, and increases in network management and routing.

- Finally, the module explained how this great advance in the entire IP address management system, better problem solving techniques, and efficient transference of the data. Among the topics presented as important for network architecture optimization, connectivity maintenance, and networking future preparedness were subnetting, DHCP, NAT and IPv6.

Reflection

- In this module, the most important thing I learnt was how the data plane functions within the network layer; how routers work, particularly in terms of packet forwarding. Now I am aware of the essential difference in speed between the control plane, which runs in Milliseconds and the data plane which runs in Nano seconds. I was able to enhance the understanding of how I can confidently explain and manage efficient network performance and data transmission.

- This subject added to my previous knowledge of computer networks by introducing me to CIDR and subnetting, which are vital in the tackling of the IP address and mapping of the network. In some ways I feel that I now fully understand how to apply the feature of DHCP for dynamic IP allocation than before and also understand how to implement subnetting in the right manner.

- The course team may have paid particular attention on this content because it provides good background information with regards to networks, which is central to debugging and optimizing their performance. This means that it is important to have adequate knowledge of these components such as DHCP, NAT and IPv6 in order to create safe and efficient networks and especially for home and business use.

- Thus, regarding myself, I found the practical sessions containing the acting out scenes rather useful. While performing the laboratory assignments, I such things as routing tables, analyzing the packet flows and configuring the subnet masks, to help augment the theory learnt. Another problem I faced was with subnetting at the initial stages but I managed to overcome this with some practice. These encounters emphasized an importance of having detailed focus on the situation, given that small changes in configuration of the network can affect its function significantly.

- Thus, this session really has equipped me with the basic technical knowhow, required to overcome any networking head ache in future. The knowledge of IP address management, routers and other aspects within the network layer shall be quite helpful when it entails attaining and implementing network solutions for security and high performance.