**Assignment 1, Network Technology, IT1662**

**Name:** Olfsen Valones

**Admin** **no**: 220274X

**Module Name & Group:** Network Technology, IT1662-T02

Table of Contents

Introduction Page 1

Network Application Page 1

How it works Page 1 - 3

What is the OSI Reference Model ? Page 4

Encapsulation & Decapsulation Page 4 - 5

* Request Process (Client-Server) Page 6 - 11
* Reply Process (Server-Client) Page 12 - 16

Conclusion Page 17

References Page 17 - 18

**Introduction**

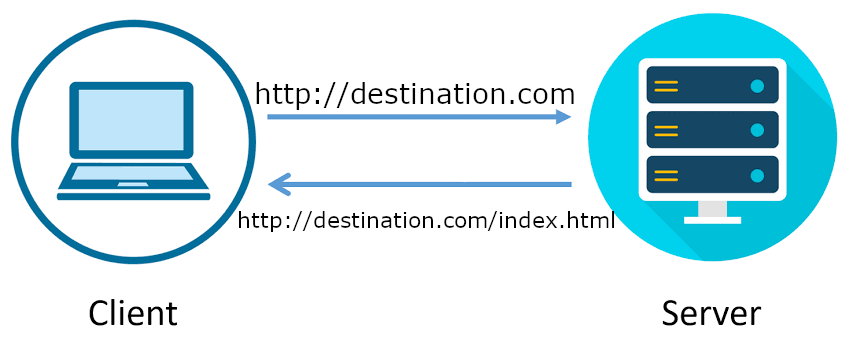
The purpose of this assignment is to explain how network communications occur from a source to a destination. It’ll also cover how encapsulation and decapsulation happens as data flows between end and intermediary devices in the network.

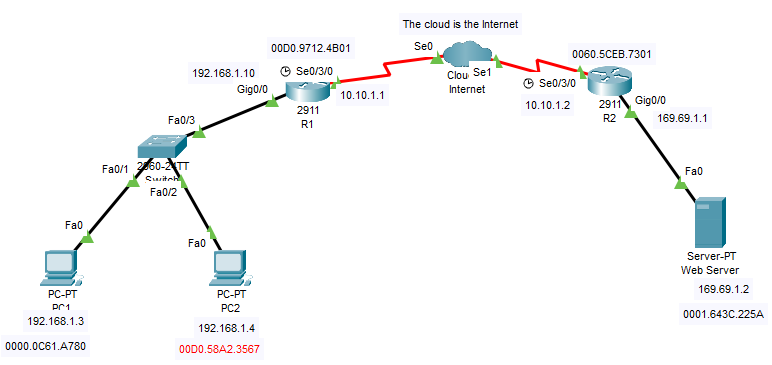
**Network Application: Loading a webpage**

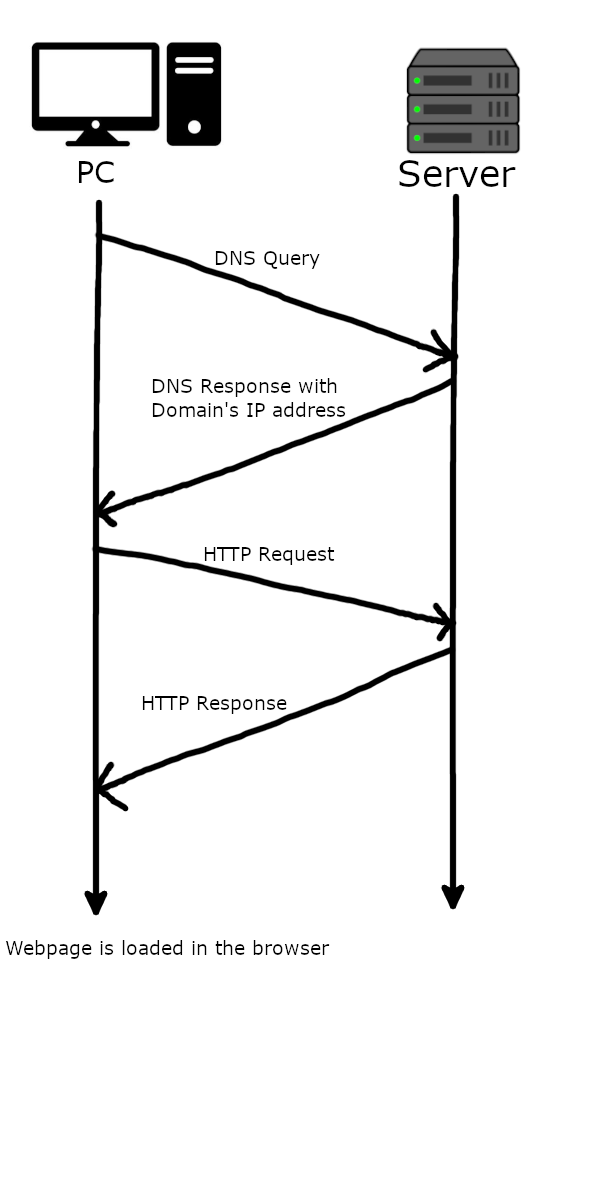
The choice of network communication to be explained is a client PC sending a request to a web server to load a webpage. The protocols used will be HTTP and DNS. Loading a webpage is used when trying to browse the internet, and we do it through entering in the Domain Name of a website into the address bar.

**How it works (Loading a webpage)**

A user would enter their desired URL into the browser, using the DNS protocol. From there, the request would go through the Internet until it reaches the destination, which is a web server. The web server processes the request and sends back the desired HTML file to load on the user’s browser using HTTP.

Overview graphic of PC sending a request to the server to access the webpage (Fig 1):

Network Diagram containing client computer, web server, internet (cloud), and intermediary devices (Fig 2):

Communication Diagram between PC and Server to show the communications done over time (Fig 3):

**What is the OSI Reference Model**

The OSI (Open Systems Interconnection) Reference Model was created by the International Organization for Standardization (ISO) to enable different computer systems to communicate with one another through a standardized protocol.

Timeline

Description automatically generated(Fig 4: The OSI Model and its functions)

Each layer contains a set of rules/protocols, with layers providing services to the one above it, while using services from the layer below it.

**Encapsulation and Decapsulation**

*Encapsulation* is the process of adding information to the application data in the form of headers. At the Application layer, data from the user is obtained and given a header with the protocol used (e.g HTTP). From there is goes down the other layers, getting another header from Layer 6 (Presentation) and Layer 5 (Session).

At Layer 4 (Transport), a new header is added to the data with information about the TCP protocol, along with the source and destination port numbers. As it’s a HTTP transmission, the destination port will be 80. In Layer 4, the data is segmented and is now called a segment PDU.

The segment goes down to Layer 3 (Network), where it’s encapsulated with an IP header, giving it a source and destination address. This PDU is a packet, and will be passed to the next layer, Layer 2.

At Layer 2 (Data Link), the packet is encapsulated with another new header, and this time, trailer too. The Data Link layer is the only layer to have a header and trailer. This header contains information on the source and destination’s MAC (Media Access Control) address. However, unlike Layer 3 with its destination address being the end device in another network, the MAC address when initially encapsulated is the MAC address of the default gateway.

A Layer 2 PDU is called a *frame*. And in the case of a transmission through a copper cable in the network, it’s an Ethernet frame.

The trailer that’s added at Layer 2 contains a frame check sequence (FCS). This is done for error checking purposes to determine if the data received has been tampered with or corrupted during transmission. If the data does not match with the FCS, the data frame is dropped immediately, else transmission of the frame was successful.

The final layer, Layer 1 (Physical), is where the frame is being encoded into binary (0’s and 1’s) before sent through the physical medium. In this layer, there is no encapsulation done, only the encoding of the frame to binary.

Now that the encapsulated data is travelling through the network and has arrived at the destination end device, it’ll have to undo the encapsulation process to obtain the data. This processing of undoing the headers is called decapsulation (or de-encapsulation).

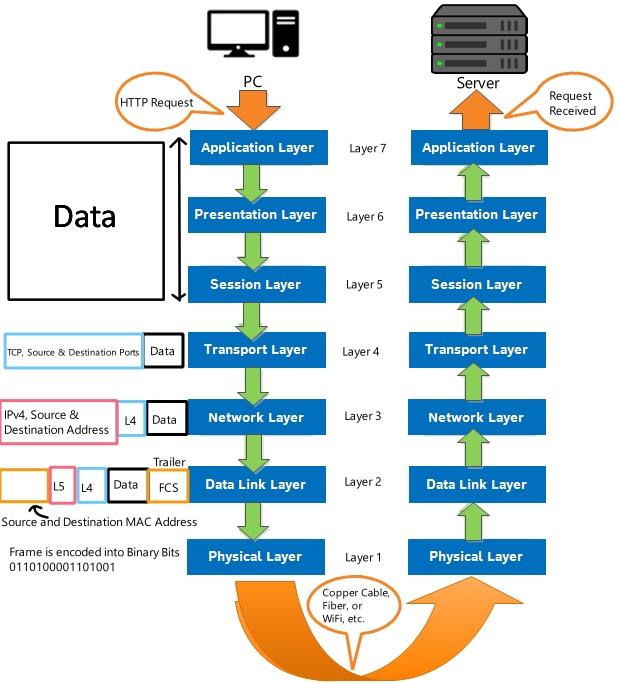
*Decapsulation* begins at the Layer 2. Now the process is being done from the bottom of the OSI Model upwards as opposed to Encapsulation (downwards from Layer 7). The Ethernet frame will check for the destination MAC address, determining if the frame was intended for them. Upon successful confirmation it’d decapsulate the frame and send it up to Layer 3.

Layer 3 is where the IP addresses are checked. After determining the destination IP address belongs to them, the packet is decapsulated and the segment is passed up to Layer 4.

At Layer 4, the start of a TCP Handshake is established as the end device receives the segment, along with information on the source’s IP address and port number. It then would process the payload data and reassemble the segments before passing it up the layers, until it reaches Layer 7, defining it to be a HTTP request from a source. In response it’d create a HTTP reply and the data repeat the encapsulation process on its side before sending it back to the source to load the webpage.

**Request Process with Encapsulation and Decapsulation (HTTP)**

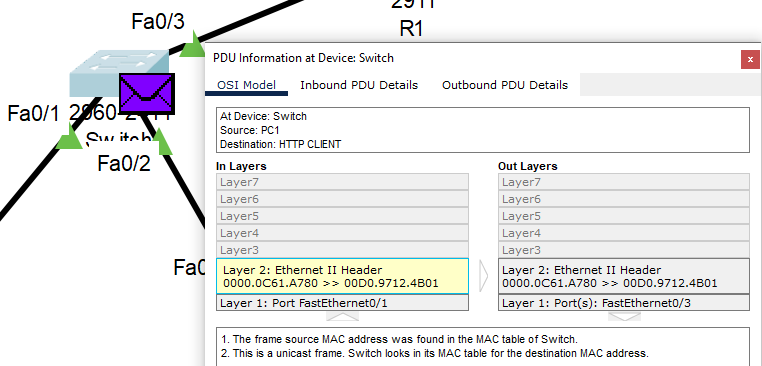
(Fig 5. Network Layer Diagram (Client to Server))

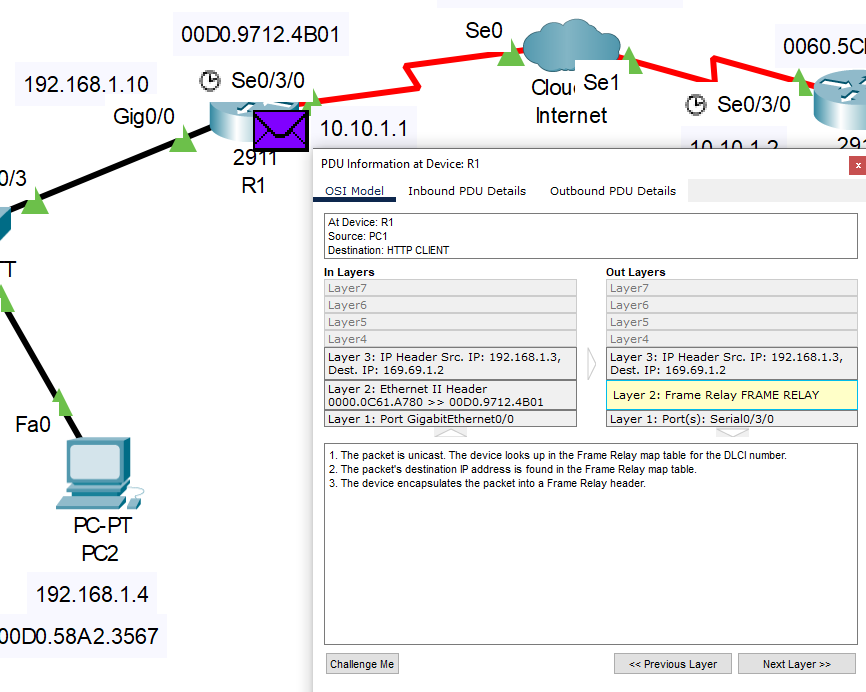


Upon successfully receiving a DNS response, the client will send a HTTP request to access the webpage. The source IP address is the client, and the destination address is the Web Server.

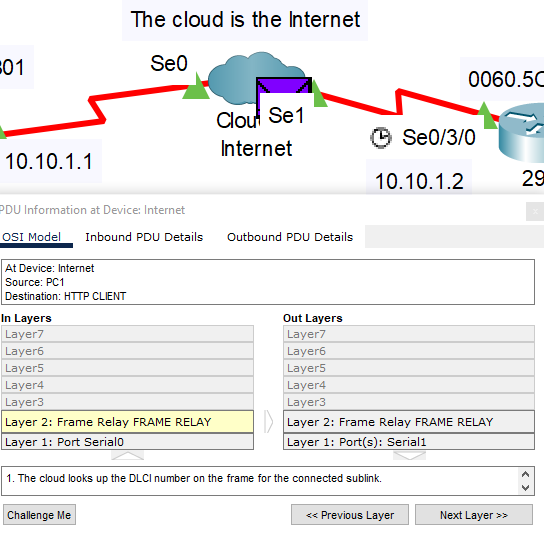
Graphical user interface

Description automatically generatedThe protocol used will be TCP, and information such as source, destination port (80 for HTTP), and the start of the TCP Handshake are encapsulated into the data, forming a segment. That segment is encapsulated into a packet, with source and destination IP addresses. At Layer 2, a frame header containing the destination MAC address of the default gateway (R1: 00D0.9712.4B01) and a trailer for error checking is added to the packet, now becoming a frame before its encoded into 1’s and 0’s to be transmitted to the next point in the network through port FastEthernet0. (Fig 5)

The next point, the switch, receives the frame from FastEthernet0/1, forwarding it to the router after checking the destination MAC address through FastEthernet0/3. (Fig 6)

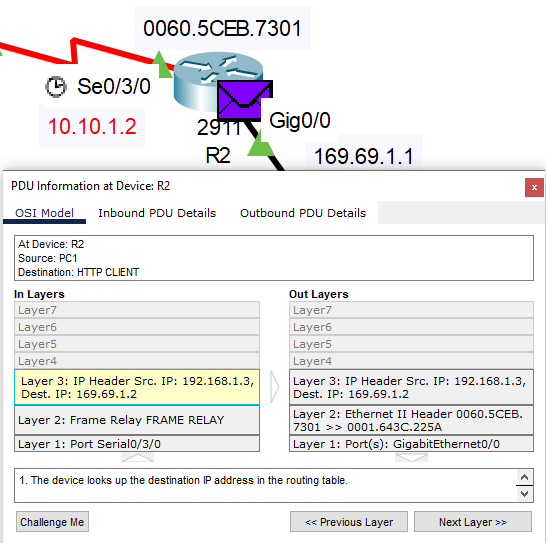
The router receives the frame from GigabitEthernet0/0, matching its destination MAC address to itself before decapsulating it and checking the packet’s destination address in a routing table. Once the route has been identified, the packet’s destination IP address is encapsulated, into a Frame Relay header as it’ll be going through serial connections, and into the Internet through Serial0/3/0. (Fig 7)

At the Cloud (or Internet), a serial connection receives the frame and checks the DLCI (Data Link Connection Identifier) number. This is a private path for the data to travel across to get to its next point, R2.

(Fig 8)

R2 will receive the frame from its Serial0/3/0 port and decapsulate it to send it to Layer 3, where the destination IP address is searched in R2’s routing table. Once found, the packet is encapsulated with a Layer 2 header. Through ARP (Address Resolution Protocol), the destination MAC address is found as its associated with the Web Server’s address of 169.69.1.2.

Now as a frame, its encoded into bits and sent through a copper straight through to the Web Server by port GigabitEthernet0/0. (Fig 9)

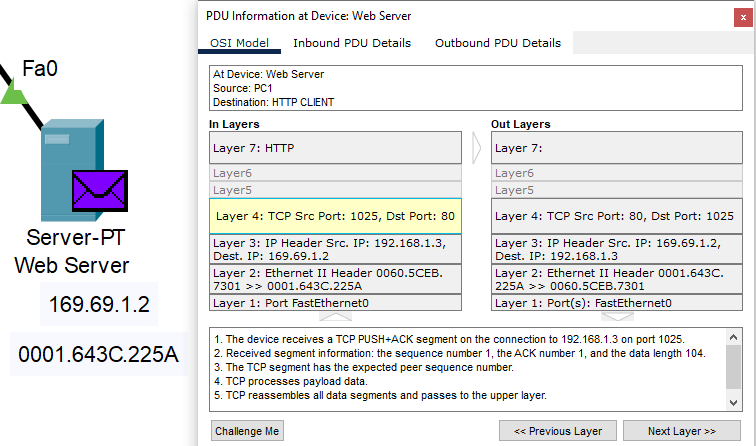


At the destination, the Web Server receives the frame and has its destination MAC address checked. As it matches, the frame is decapsulated and passed to Layer 3. The destination IP address matches the Web Server’s IP address, getting decapsulated again and sent to Layer 4.

In Layer 4, it receives information that a PC is trying to initiate a TCP Handshake with it through a PUSH + ACK segment. It then processes the segments and reassemble it before passing it to Layer 7, the Application Layer.

Now that the Web Server has received a HTTP Request, it will now send a HTTP Reply to the PC.

(Fig 10)

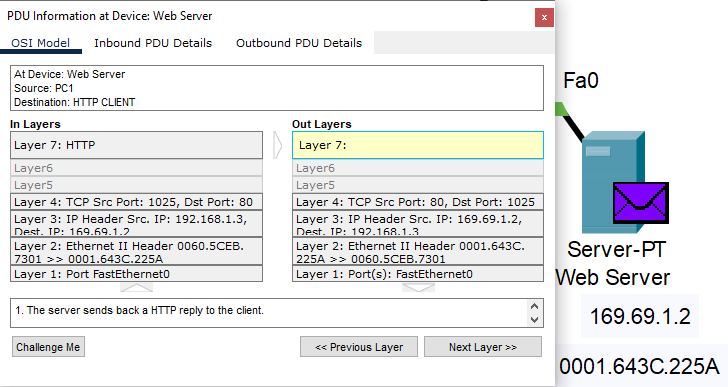


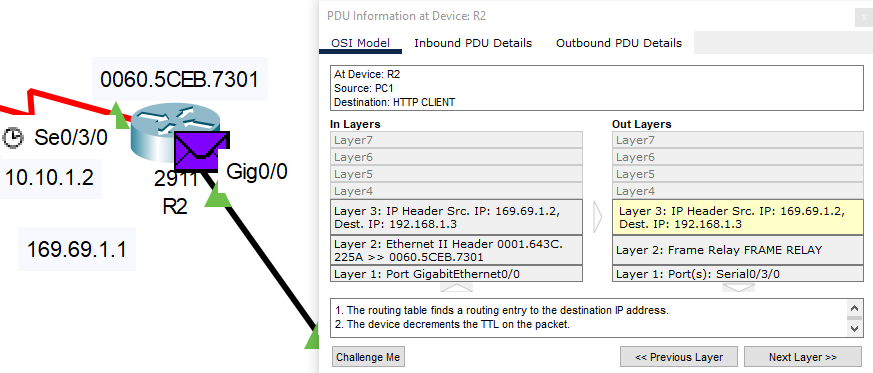
**Reply Process with Encapsulation and Decapsulation (HTTP)**

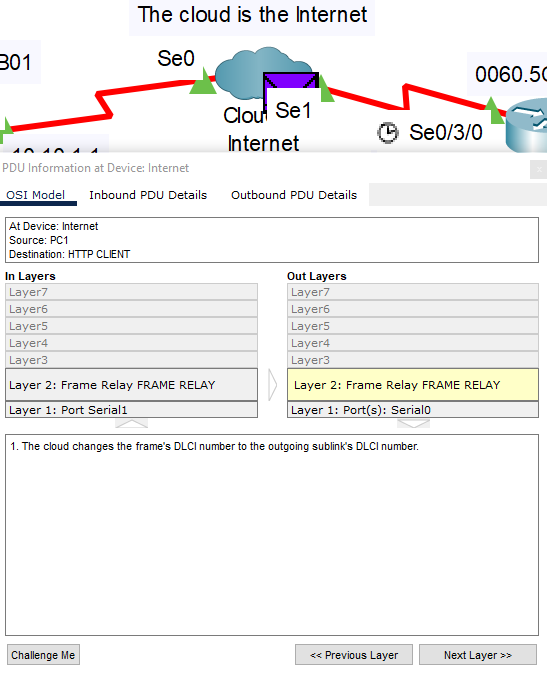
After decapsulating the layers to receive the HTTP request, the Web Server sends a HTTP Reply back to the PC. The data is first encapsulated at Layer 4 with TCP, source port 80 (HTTP) and the destination port of the PC.

At Layer 3, the segment is encapsulated with an IP Header, containing the source IP address (Server) and destination IP address (PC), and default gateway.

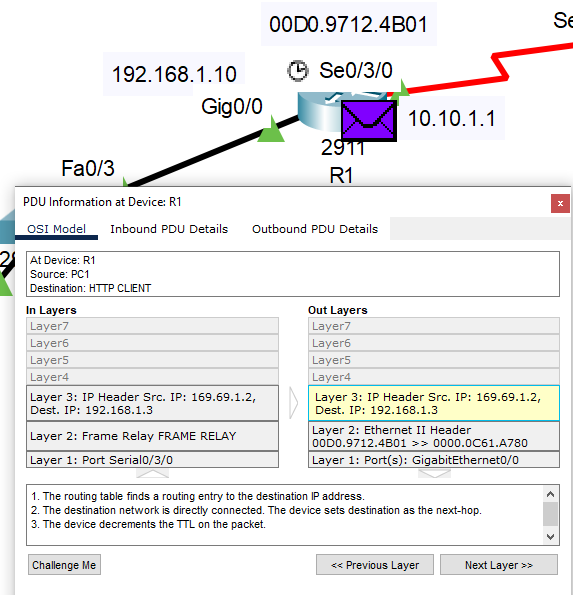
The packet is passed to Layer 2 with a source (0001.643C.225A) and destination MAC address of R2 (0060.5CEB.7301) obtained through ARP. Now the packet is encapsulated into an ethernet frame and sent out to the next point as binary bits through the Server’s FastEthernet0 port. (Fig 12)

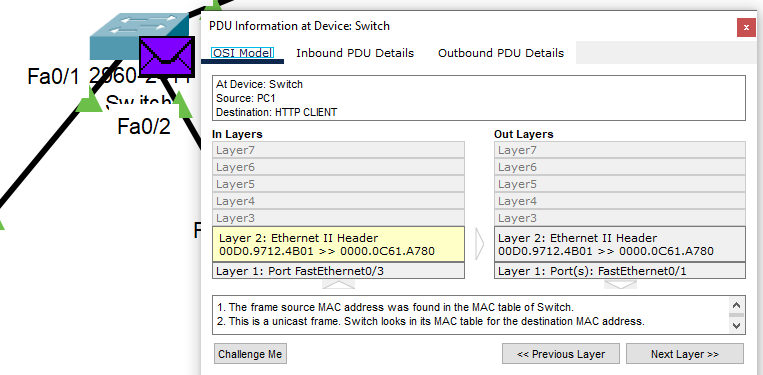


The next point, R2, receives the frame and processes the destination MAC address, decapsulating it to Layer 3. Layer 3 searches the IP address on a routing table before getting an entry to the destination address. It’s then encapsulated into a frame relay header and sent into the Internet through Serial0/3/0 (Fig 13)

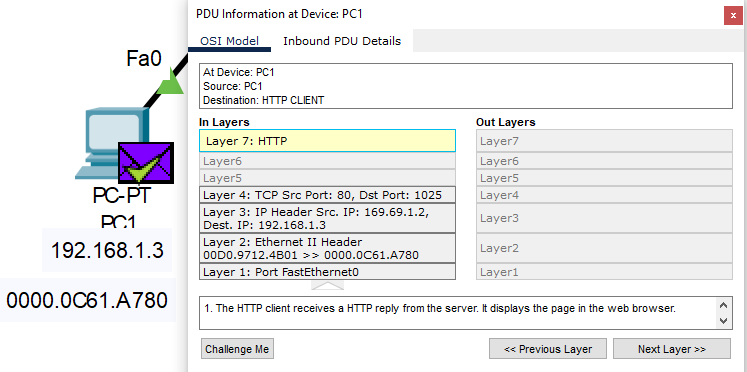
In the Cloud, it receives a frame, looks up the outgoing DLCI number for it to be going through before being sent out to R1. (Fig 14)

R1 receives the frame from the Internet and decapsulates it for Layer 3 to process. The IP address is looked up and finds an entry to the destination address, with the last intermediary device being a switch. The packet is encapsulated with the source MAC address being R1 (00D0.9712.4B01), and the destination address being the PC, the frame is sent through port GigabitEthernet0/0 towards the switch. (Fig 15)



The switch, the last intermediary device gets the frame from R1 and sends the frame towards Port FastEthernet0/1 by using its MAC address. (Fig 16)

Finally, the PC receives the frame and checks the destination MAC address before it decapsulates it to go up to Layer 3. The packet’s destination IP address matches the PC’s and it gets decapsulated again to Layer 4.

In Layer 4, it contains the TCP ACK segment from the Web Server, meaning that a connection is established and the content to load the webpage will arrive soon. After a while it receives the acknowledgement from the Web Server and its contents are decapsulated from Layer 2 to 4 before going up to Layer 7, the Application Layer where it processes the HTTP Reply and the desired page is displayed in the web browser. (Fig 17)

(Fig 18: Successful display of the webpage)

Graphical user interface, text, application, email

Description automatically generated

**Conclusion**

Having to work on this assignment has showed me the depth of computer networking, how the network application I chose works, and how the OSI Model is applied on each step, from start to finish.

From encapsulation to decapsulation, the layers that end and intermediary devices used have been interesting to learn about.

**List of References**

<https://industry40.co.in/wp-content/uploads/2021/05/2021-05-16_13h06_38.png> (Network Layer Diagram, Fig 11 (Edited))

[https://miro.medium.com/max/720/1\*8-fT6K1o6nHiBRxKppcqOg.png](https://miro.medium.com/max/720/1*8-fT6K1o6nHiBRxKppcqOg.png) (HTTP Request, Fig 1 (Edited))

<https://www.imperva.com/learn/wp-content/uploads/sites/13/2020/02/OSI-7-layers.jpg.webp> (OSI Model, Fig4 (Edited))

Fig 2, 5-10, 12-17 (Packet Tracer’s Simulation Mode)

Fig 18 (Packet Tracer’s Web Browser)

*Definition of DLCI*. PCMAG. (n.d.). Retrieved November 20, 2022, from https://www.pcmag.com/encyclopedia/term/dlci

Singla, A. (2022, September 5). *Application layer in OSI model*. GeeksforGeeks. Retrieved November 20, 2022, from https://www.geeksforgeeks.org/application-layer-in-osi-model/

*What is frame check sequence? - definition from Techopedia*. Techopedia.com. (2011, August 18). Retrieved November 20, 2022, from https://www.techopedia.com/definition/24793/frame-check-sequence

*What is the OSI model? | CloudFlare*. What is the OSI Model? (n.d.). Retrieved November 20, 2022, from https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/

YouTube. (2017). *5.Data Encapsulation Osi Tcpip* . *YouTube*. Retrieved November 20, 2022, from https://www.youtube.com/watch?v=xaKvGnnuYmk&t=315s.

YouTube. (2020). *Osi Model Deep Dive* . *YouTube*. Retrieved November 20, 2022, from https://www.youtube.com/watch?v=oVVlMqsLdro.