Department of Computer Science and Engineering

Islamic University of Technology (IUT)

A subsidiary organ of OIC

Laboratory Report

CSE 4412: Data Communication and Networking Lab

Name	**: Mirza Mohammad Azwad**
Student ID	**: 200042121**
Section	**: BSc in SWE**
Semester	**: 4th Semester**
Academic Year	**: 2022-23**
Date	**: 13/01/2023**
Lab No	**: 02**

Title: Understanding the basics of OSI Model

Objective:

1. Examine HTTP Web Traffic
2. Display Elements of the TCP/IP Protocol Suite

Devices/ Software Used:

We used the Cisco Packet Tracer simulated on our laptops for this lab. Cisco Packet Tracer is basically a packet path simulator that can be run on local computers to simulate and understand different network connections.

Working Procedure:

At first, we learned about a basic overview of the OSI(Open Systems Interconnected) Model which consists of 7 layers namely the physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and the application layer. But in practice, only 5 layers are traditionally used that make up the TCP/IP(Transmission Control Protocol/Internet Protocol) protocol suite which consists of the physical layer, data link layer, network layer, transport layer, and application layer with the session layer and presentation layer fused with the application layer. Firstly we realize that the HTTP protocol involves the use of the application layer but the main observable idea was that the packet is created and transmitted to and from the server.

The first step in this project was to observe the internal configuration of the web server and the web client. For that, we double-clicked on the web server and were taken to the config menu which had a form of navigation that allowed us to go to the services menu. The services menu had options for different protocols but our focus was mainly on the HTTP protocol and thus we observed the contents of the HTTP protocol that included the index.html or the entry page for that web server.

We learned that the web server had the DNS feature turned on and we learned about a basic idea of the DNS and that it is able to translate the web address or URL to an IP address that helps locate the page similar to how different pages are located in the world wide web.

Next in the web client, we observed a similar manner by double-clicking and then moving on to the desktop option that had a variety of features. Features that are similar to a typical lightweight operating system. Our focus for this experiment was the browser. We entered the address www.osi.local which was an address whose IP address was present in the server to be resolved using the DNS feature. Upon entering the IP address we reached the index.html contents in the presentable format in the web client.

This part of the experiment was carried out in real-time.

After this part, we reached the main goals of the experiment. Simulating how packets are transferred and understanding the HTTP web traffic. For that we first observe the simulation mode, the previous tests were done in real-time mode. In the simulation mode, we can observe the additional packet transfers as a form of simulation.

The first observation in simulation mode was that the web client sent a request to the server and the server returned packets to the web client. The packet contained details of the web page hosted by the web server in place. The first request results in some packet transfer from client to server and it is shown as a purple envelope that goes to the server from the client in the form of a request. In the second stage, the server returns the data that was requested in our case the web page to the client as a packet also shown as a purple envelope.

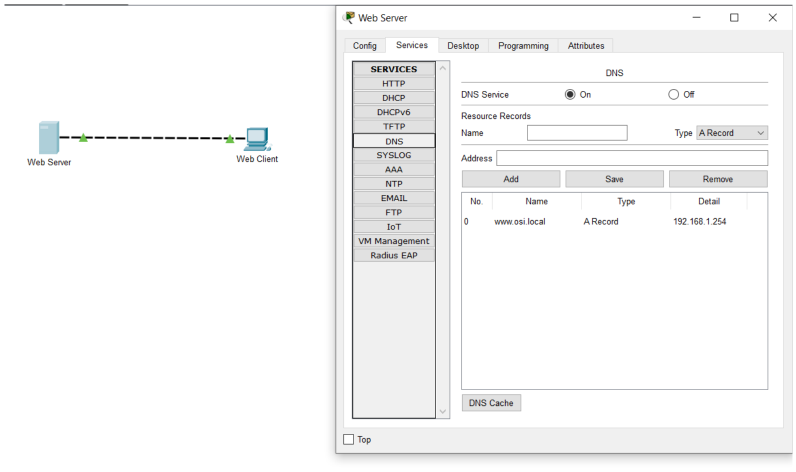
The second observation is when we look at the simulation panel. We see 4 sets of data every time we access the URL www.osi.local. These 4 sets of data are transferred within a span of 0.004 seconds.

Upon clicking on each set of data we can come across some more information regarding each of the 7 layers and how it is working. In the Cisco packet manager, each of the layers can be observed as an in-layer and an out-layer. For the in-layer, we also have an associated Inbound PDU option. PDU stands for Protocol Data Unit which is essentially units of exchange between components on a network that communicate via various networking protocols. For the out-layer, we also have an associated outbound PDU. Which shows the protocols involved in transferring data into and out of a sender or receiver.

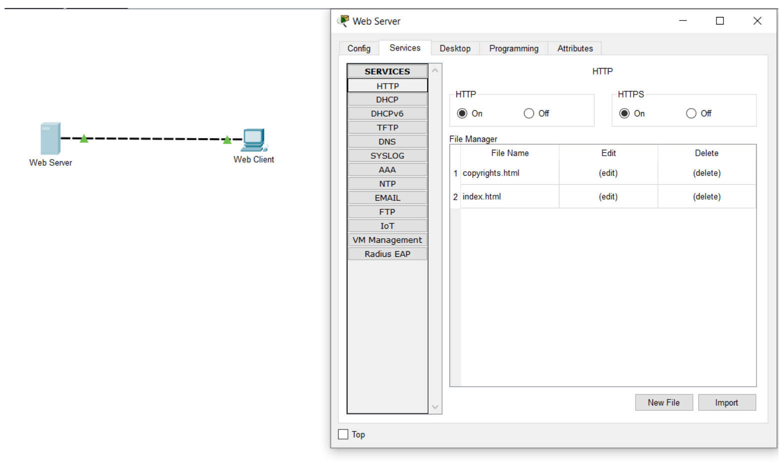
The layers also display the detailed set of activities being performed in those layers and the PDU options show the details of the different packets being transferred along with the number of bits, the different segments of the packets, etc. Now let’s observe the diagrams related to the experiment.

Diagram of the experiment:

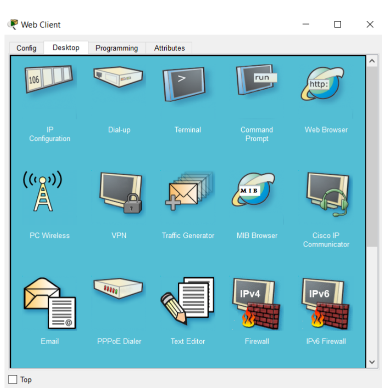
Initially, we observe the DNS option of the web server

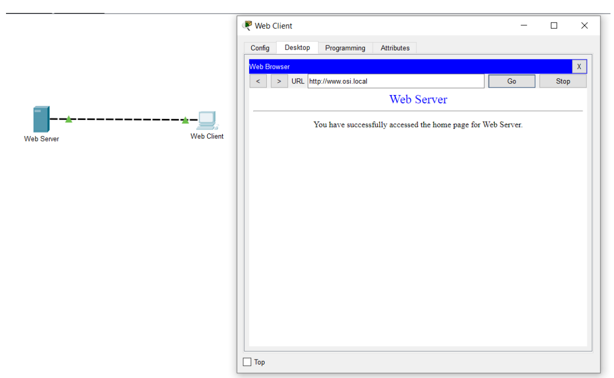


Then we observe the HTTP information about the web server

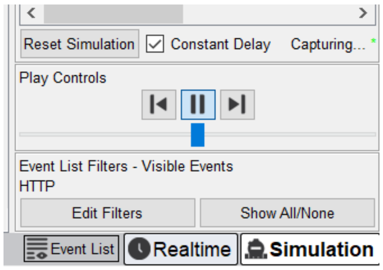


We access the desktop option of the web client followed by the web browser option where we access the URL www.osi.local

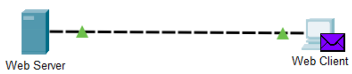


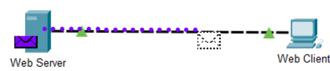


Then we shift from our current realtime mode to simulation mode

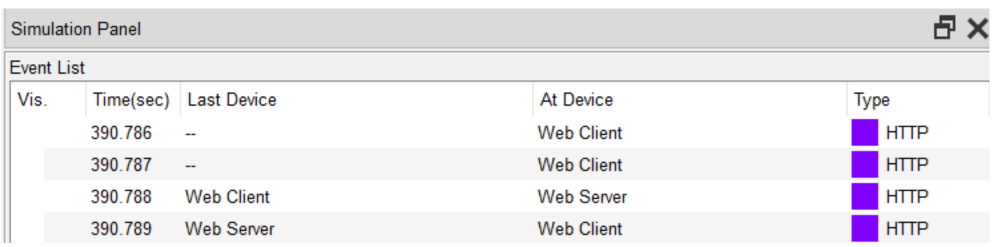


Then in simulation mode, we observe the transfer of packets from the client computer to the server computer and vice versa in the simulation mode





Then we move on to the simulation panel where there are 4 rows representing 4 states of the simulation within a span of 0.004 seconds.

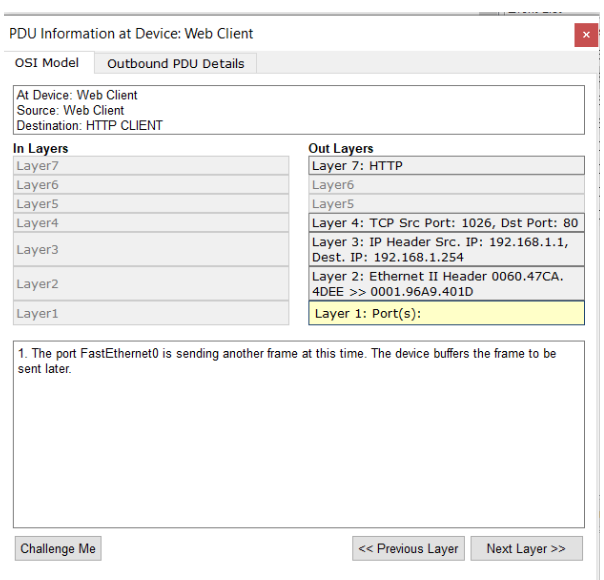


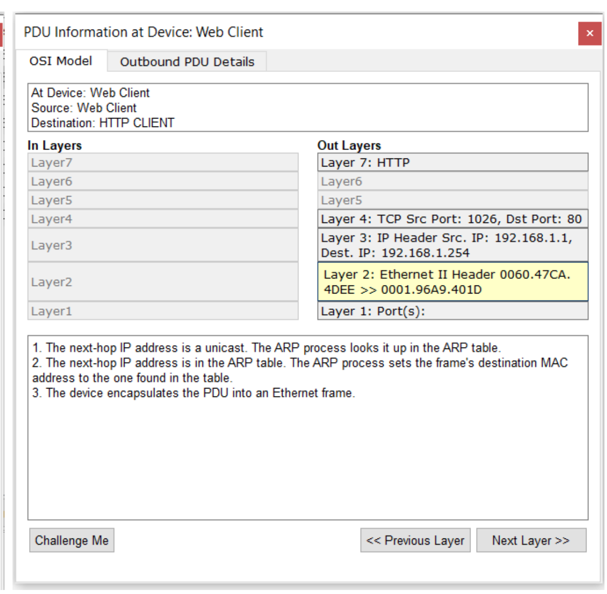
Simulation Panel

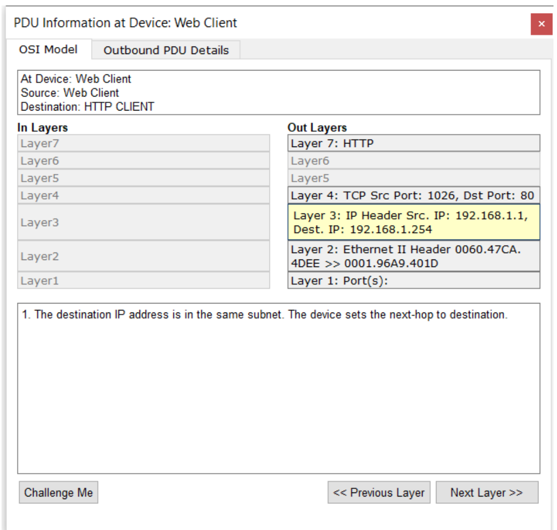
Event List

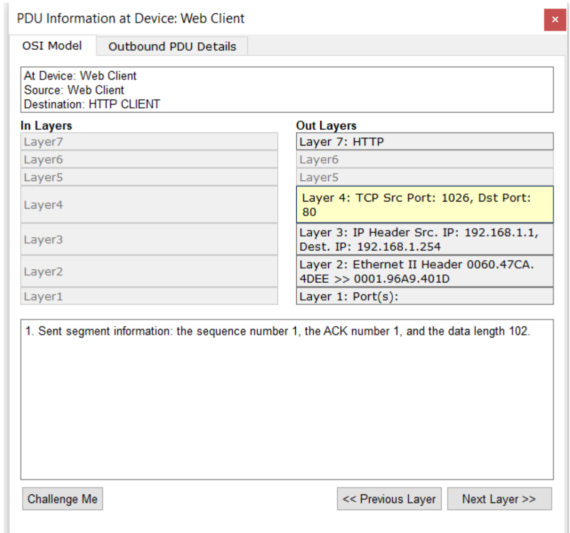
Vis.	Time(sec)	Last Device	At Device	Type
	390.786	--	Web Client	HTTP
	390.787	--	Web Client	HTTP
	390.788	Web Client	Web Server	HTTP
	390.789	Web Server	Web Client	HTTP

We see the information for the first set or state, this shows that 5 of the 7 layers of the OSI model are active. Followed by the information shown for each set. Only the out layer is active in this case.

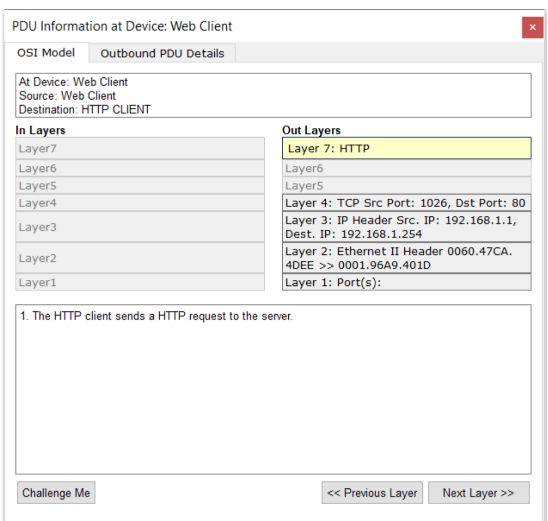


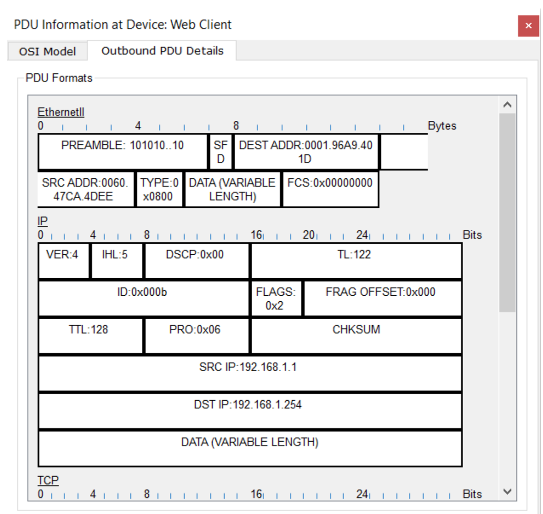


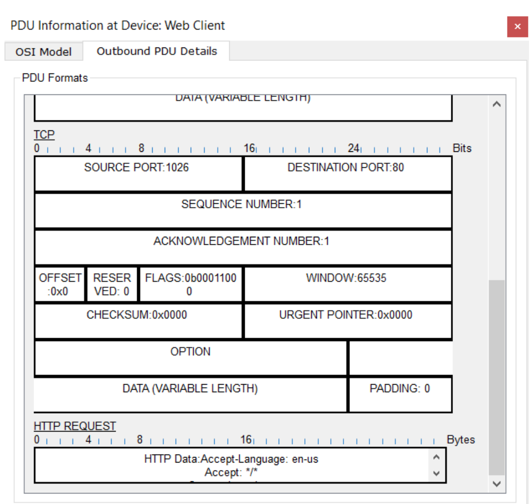




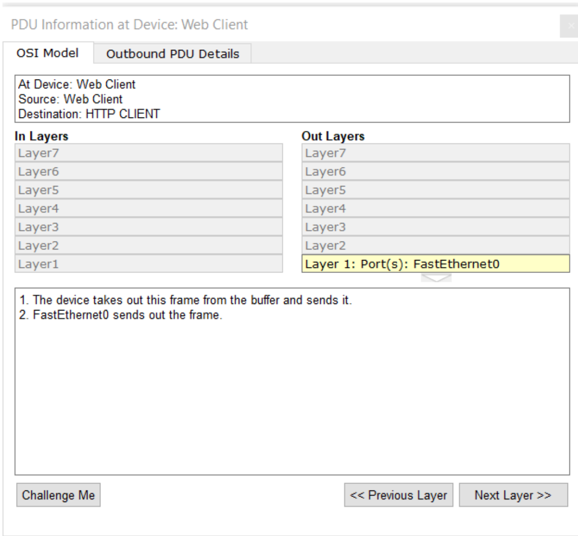
In this phase, we also see the outbound PDU as well as the information on the outbound PDU

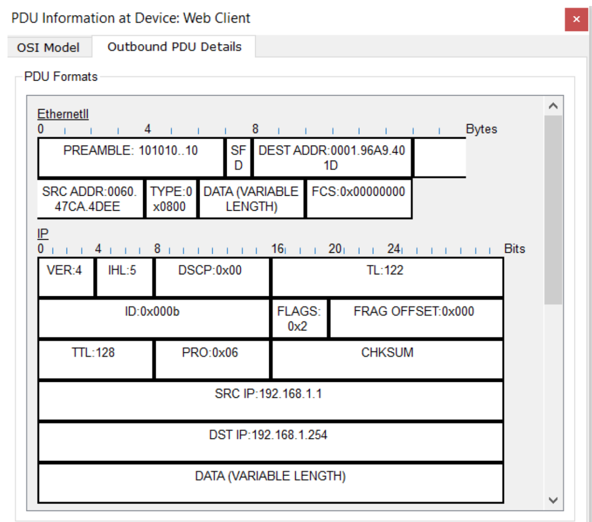


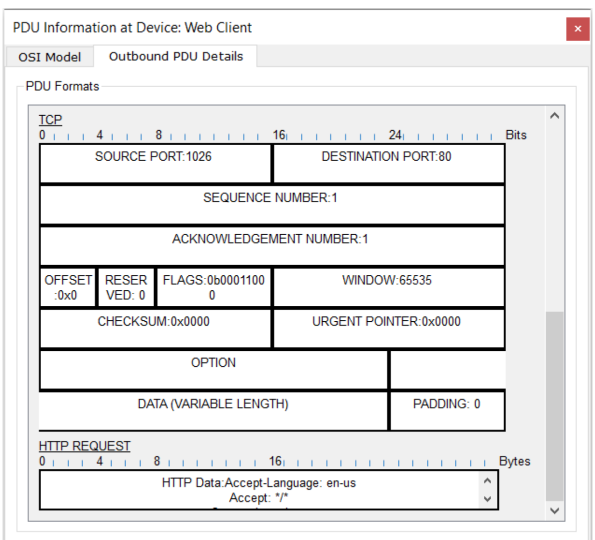




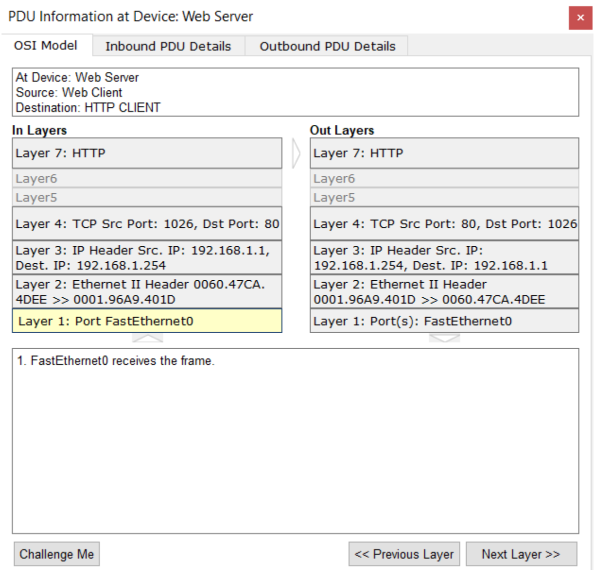
This represents the next stage in the simulation panel. This state only shows one active out-layer layer 1.

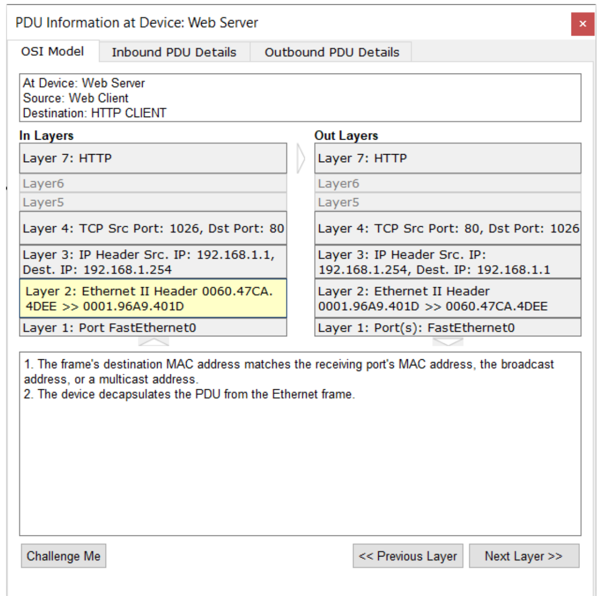


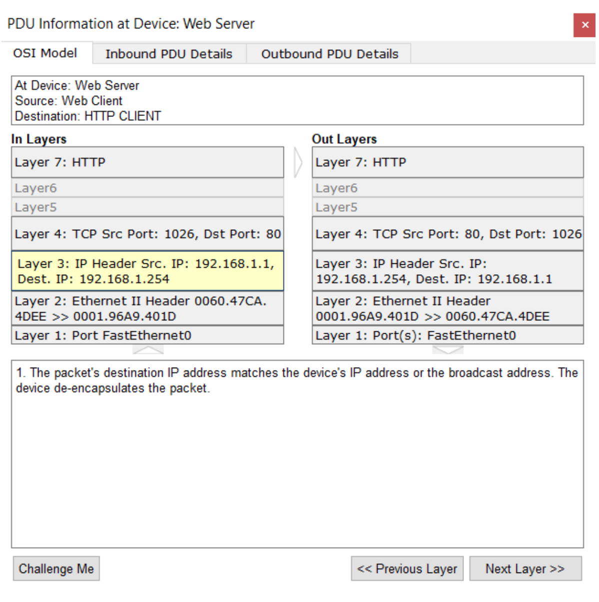


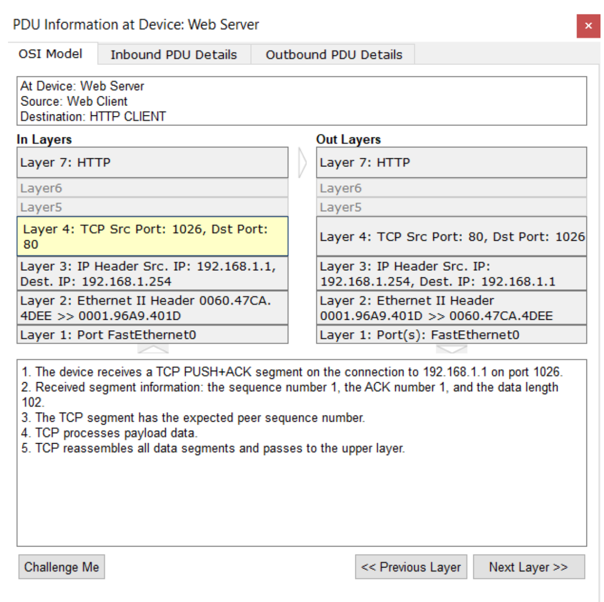


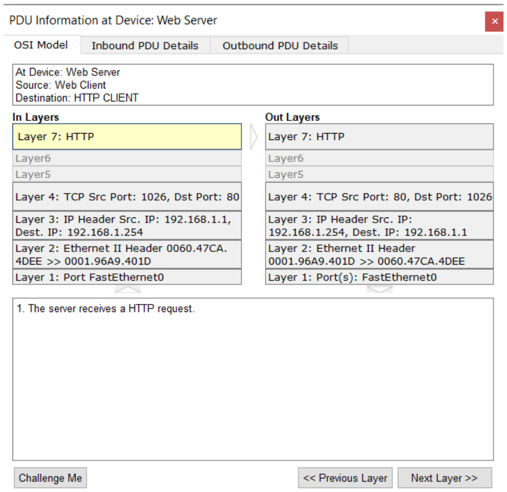
Then we move on to the next stage in the web server. The pictures show the details of the different layers, its activities and the bits of data transferred for the In Layer and the Out Layer.

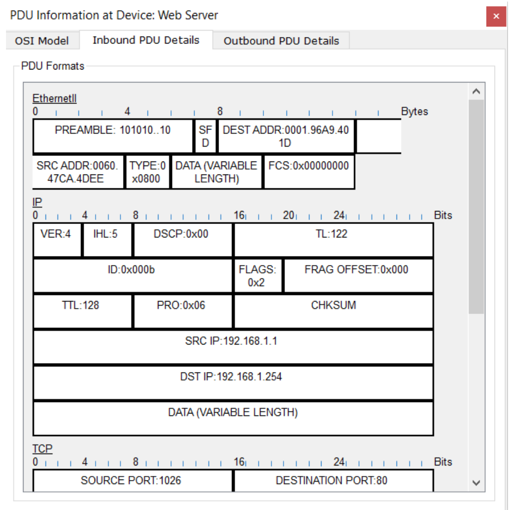


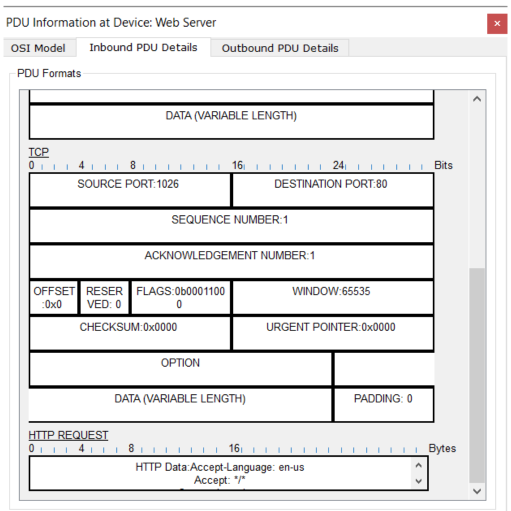


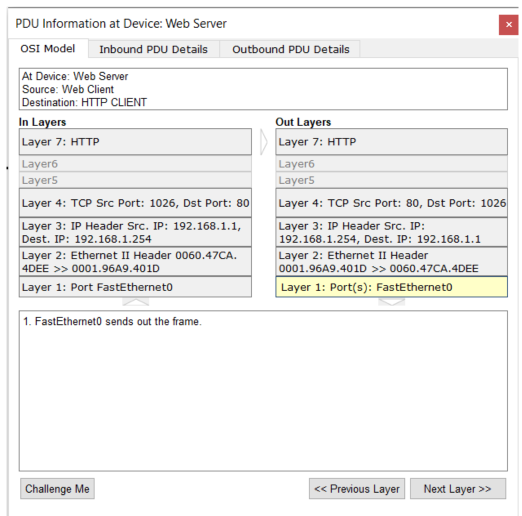


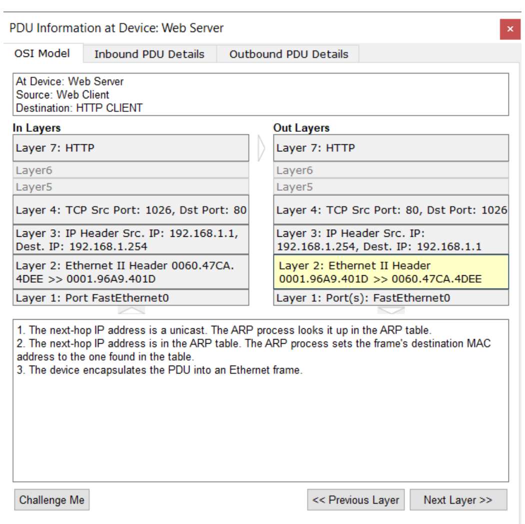


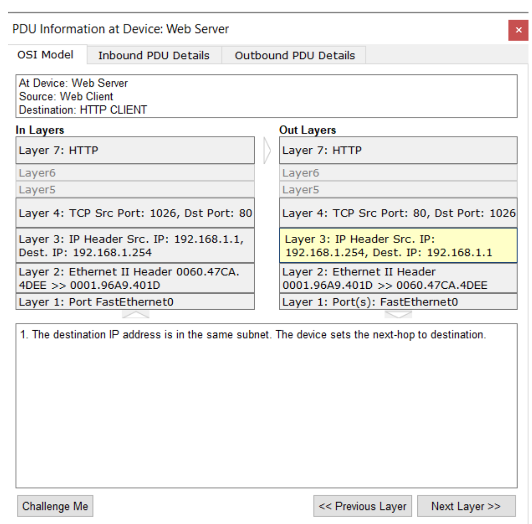

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Web Server
Source: Web Client
Destination: HTTP CLIENT
In Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 1026, Dst Port: 80
Layer 3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.254
Layer 2: Ethernet II Header 0060.47CA.4DEE >> 0001.96A9.401D
Layer 1: Port FastEthernet0
Out Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 80, Dst Port: 1026
Layer 3: IP Header Src. IP: 192.168.1.254, Dest. IP: 192.168.1.1
Layer 2: Ethernet II Header 0001.96A9.401D >> 0060.47CA.4DEE
Layer 1: Port(s): FastEthernet0
1. The server receives a HTTP request.
Challenge Me
<< Previous Layer
Next Layer >>

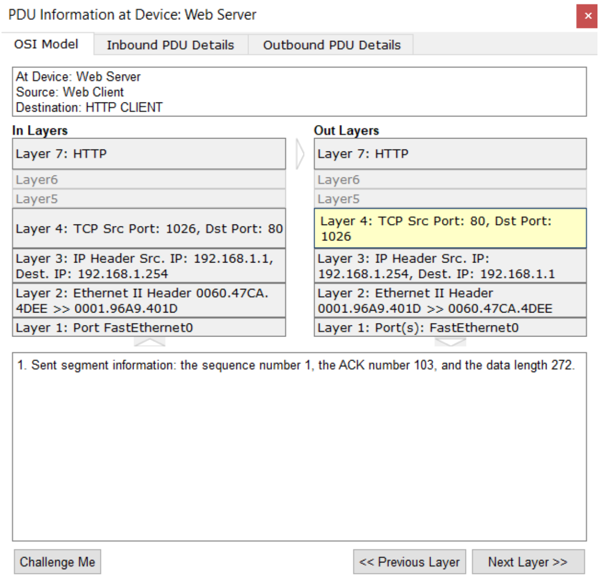

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
EthernetII
0 4 8 Bytes
PREAMBLE: 101010..10
SFD
DEST ADDR:0001.96A9.401D
SRC ADDR:0060.47CA.4DEE
TYPE:0x0800
DATA (VARIABLE LENGTH)
FCS:0x00000000
IP
0 4 8 16 20 24 Bits
VER:4
IHL:5
DSCP:0x00
TL:122
ID:0x000b
FLAGS: 0x2
FRAG OFFSET:0x000
TTL:128
PRO:0x06
CHKSUM
SRC IP:192.168.1.1
DST IP:192.168.1.254
DATA (VARIABLE LENGTH)
TCP
0 4 8 16 24 Bits
SOURCE PORT:1026
DESTINATION PORT:80

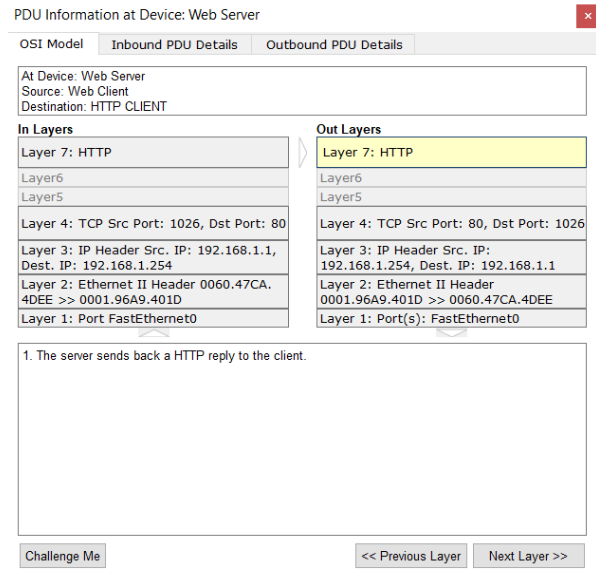

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
DATA (VARIABLE LENGTH)
TCP
0 4 8 16 24 Bits
SOURCE PORT:1026
DESTINATION PORT:80
SEQUENCE NUMBER:1
ACKNOWLEDGEMENT NUMBER:1
OFFSET :0x0
RESERVED: 0
FLAGS:0b0001100 0
WINDOW:65535
CHECKSUM:0x0000
URGENT POINTER:0x0000
OPTION
DATA (VARIABLE LENGTH)
PADDING: 0
HTTP REQUEST
0 4 8 16 Bytes
HTTP Data:Accept-Language: en-us
Accept: */*

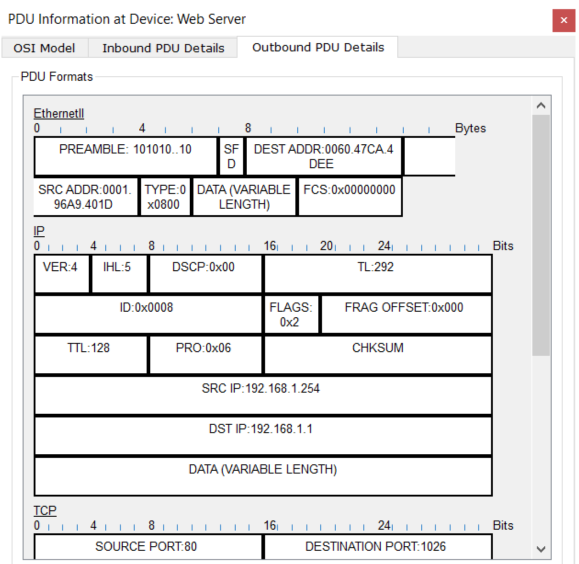

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Web Server
Source: Web Client
Destination: HTTP CLIENT
In Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 1026, Dst Port: 80
Layer 3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.254
Layer 2: Ethernet II Header 0060.47CA.4DEE >> 0001.96A9.401D
Layer 1: Port FastEthernet0
Out Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 80, Dst Port: 1026
Layer 3: IP Header Src. IP: 192.168.1.254, Dest. IP: 192.168.1.1
Layer 2: Ethernet II Header 0001.96A9.401D >> 0060.47CA.4DEE
Layer 1: Port(s): FastEthernet0
1. FastEthernet0 sends out the frame.
Challenge Me
<< Previous Layer
Next Layer >>

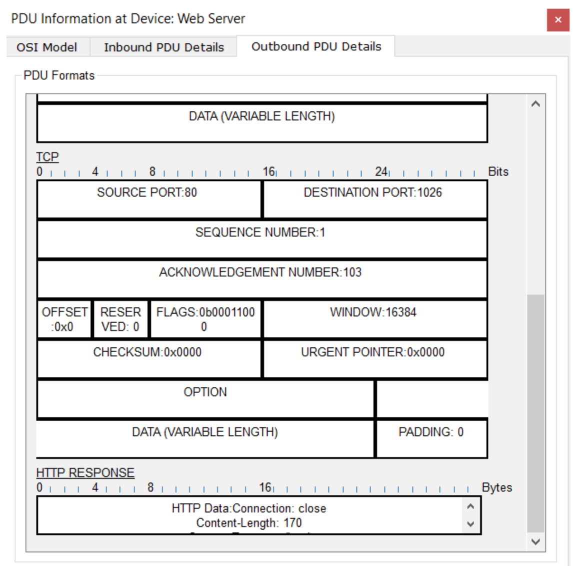

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Web Server
Source: Web Client
Destination: HTTP CLIENT
In Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 1026, Dst Port: 80
Layer 3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.254
Layer 2: Ethernet II Header 0060.47CA.4DEE >> 0001.96A9.401D
Layer 1: Port FastEthernet0
Out Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 80, Dst Port: 1026
Layer 3: IP Header Src. IP: 192.168.1.254, Dest. IP: 192.168.1.1
Layer 2: Ethernet II Header 0001.96A9.401D >> 0060.47CA.4DEE
Layer 1: Port(s): FastEthernet0
1. The next-hop IP address is a unicast. The ARP process looks it up in the ARP table.
2. The next-hop IP address is in the ARP table. The ARP process sets the frame's destination MAC address to the one found in the table.
3. The device encapsulates the PDU into an Ethernet frame.
Challenge Me
<< Previous Layer
Next Layer >>


PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Web Server
Source: Web Client
Destination: HTTP CLIENT
In Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 1026, Dst Port: 80
Layer 3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.254
Layer 2: Ethernet II Header 0060.47CA.4DEE >> 0001.96A9.401D
Layer 1: Port FastEthernet0
Out Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 80, Dst Port: 1026
Layer 3: IP Header Src. IP: 192.168.1.254, Dest. IP: 192.168.1.1
Layer 2: Ethernet II Header 0001.96A9.401D >> 0060.47CA.4DEE
Layer 1: Port(s): FastEthernet0
1. The destination IP address is in the same subnet. The device sets the next-hop to destination.
Challenge Me
<< Previous Layer
Next Layer >>

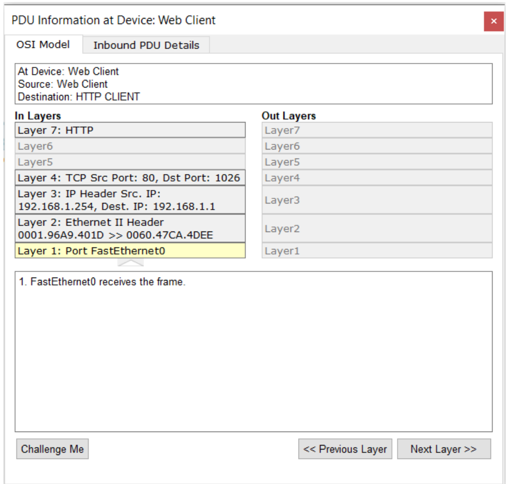

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Web Server
Source: Web Client
Destination: HTTP CLIENT
In Layers
Out Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 1026, Dst Port: 80
Layer 3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.254
Layer 2: Ethernet II Header 0060.47CA.4DEE >> 0001.96A9.401D
Layer 1: Port FastEthernet0
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 80, Dst Port: 1026
Layer 3: IP Header Src. IP: 192.168.1.254, Dest. IP: 192.168.1.1
Layer 2: Ethernet II Header 0001.96A9.401D >> 0060.47CA.4DEE
Layer 1: Port(s): FastEthernet0
1. Sent segment information: the sequence number 1, the ACK number 103, and the data length 272.
Challenge Me
<< Previous Layer
Next Layer >>

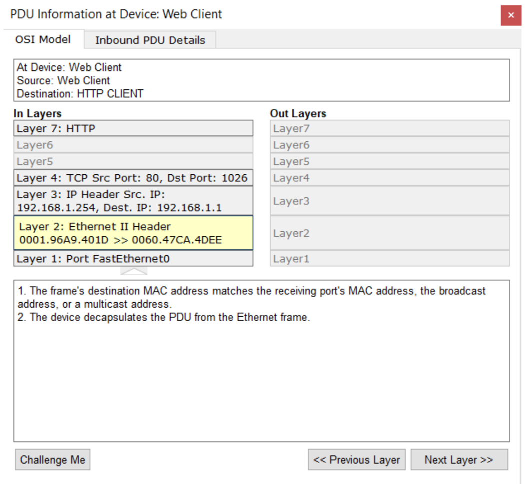

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Web Server
Source: Web Client
Destination: HTTP CLIENT
In Layers
Out Layers
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 1026, Dst Port: 80
Layer 3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.254
Layer 2: Ethernet II Header 0060.47CA.4DEE >> 0001.96A9.401D
Layer 1: Port FastEthernet0
Layer 7: HTTP
Layer6
Layer5
Layer 4: TCP Src Port: 80, Dst Port: 1026
Layer 3: IP Header Src. IP: 192.168.1.254, Dest. IP: 192.168.1.1
Layer 2: Ethernet II Header 0001.96A9.401D >> 0060.47CA.4DEE
Layer 1: Port(s): FastEthernet0
1. The server sends back a HTTP reply to the client.
Challenge Me
<< Previous Layer
Next Layer >>

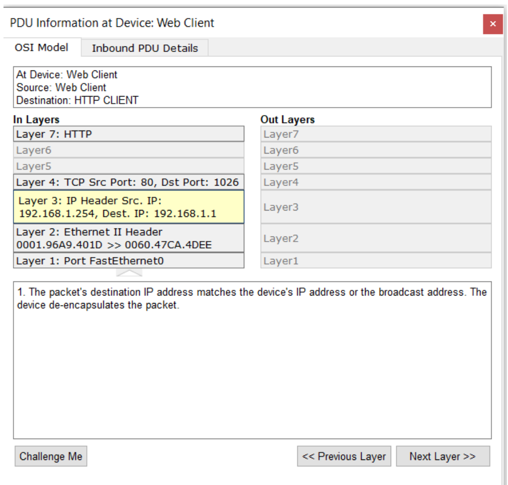

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
EthernetII
Bytes
PREAMBLE: 101010..10
SFD
DEST ADDR:0060.47CA.4DEE
SRC ADDR:0001.96A9.401D
TYPE:0x0800
DATA (VARIABLE LENGTH)
FCS:0x00000000
IP
Bits
VER:4
IHL:5
DSCP:0x00
TL:292
ID:0x0008
FLAGS: 0x2
FRAG OFFSET:0x000
TTL:128
PRO:0x06
CHKSUM
SRC IP:192.168.1.254
DST IP:192.168.1.1
DATA (VARIABLE LENGTH)
TCP
Bits
SOURCE PORT:80
DESTINATION PORT:1026

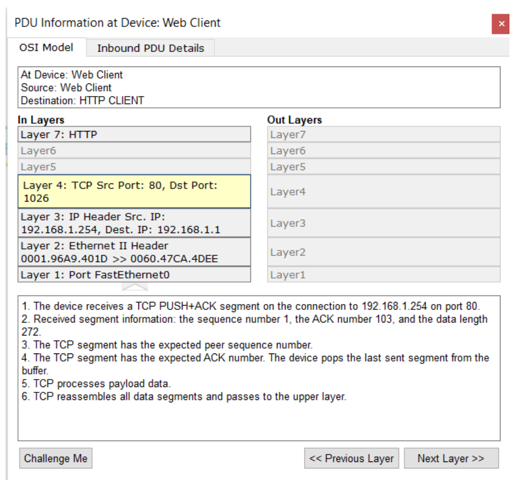

PDU Information at Device: Web Server
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
DATA (VARIABLE LENGTH)
TCP
Bits
SOURCE PORT:80
DESTINATION PORT:1026
SEQUENCE NUMBER:1
ACKNOWLEDGEMENT NUMBER:103
OFFSET :0x0
RESERVED: 0
FLAGS:0b0001100 0
WINDOW:16384
CHECKSUM:0x0000
URGENT POINTER:0x0000
OPTION
DATA (VARIABLE LENGTH)
PADDING: 0
HTTP RESPONSE
Bytes
HTTP Data:Connection: close
Content-Length: 170

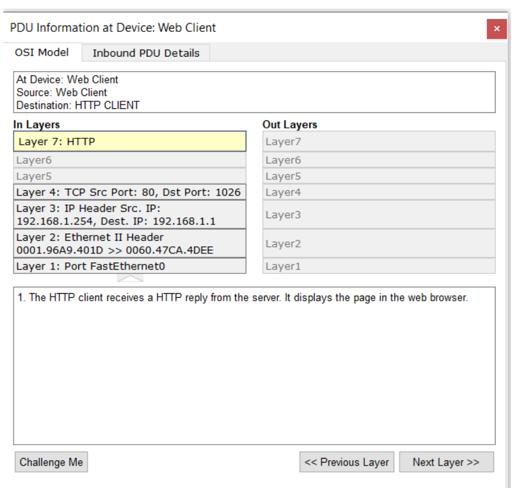
Then we move on to the next and final stage for this transfer of packets that is in the web client. We see that only the in layer is active.

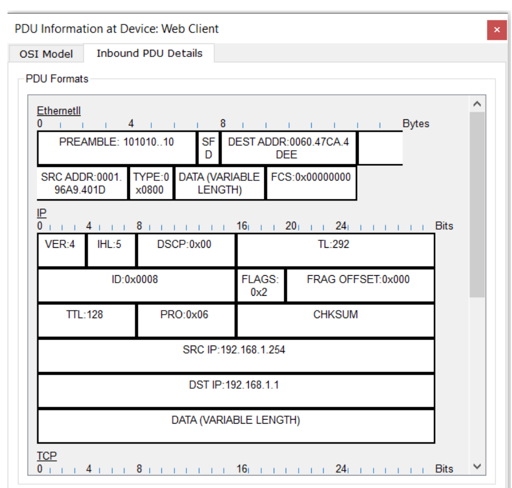


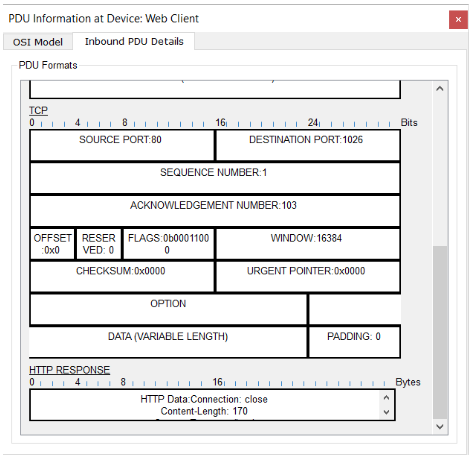












Observation:

The first observation would be during the simulation mode we could see a purple envelope move from the client to the server and back. It is essentially an example of a client-server model where the client requests a packet from the server, after which the server returns an appropriate packet based on the request from the client.

This exchange of packets is seen in the application layer perspective but to scrutinize it even further we see that all the layers except the 5th and 6th layers, that being the session layer and the presentation layers were active. This can be the main reason why in most cases we consider the TCP/IP protocol suite over the more typical OSI model as the TCP/IP protocol suite involves 5 layers excluding the 5th and 6th layers by merging them with the 7th layer of the OSI model.

We observe the first stage of packet transfer. If we observe the in a layer in a top-down manner. The first layer or the application layer sends an HTTP request to the server. The HTTP request abides by the HTTP protocol which accepts data with language en-us and it may contain '/'. Then the Transport layer breaks that information into segments with each segment having a sequence number, ACK number, and data. The transport layer also identifies the source port and the destination port. Then we move on to the network layer, which fixes the source IP address and the destination IP address. Since in this case, the client and the server are in the same subnet so the IP address for the source and the destination can be directly evaluated without involving a NAT(Network Address Translator). The destination IP address is set as the next-hop IP address. This layer also translates the segments to packets with a network layer header, called a datagram with appropriate information about the source and destination IP address in the header. After which we have the data link layer. The packets are translated to data frames that can be passed to the physical layer. First, the next-hop IP address is looked up in the ARP(Address Resolution Protocol) table. For sending a request, the next-hop IP address is unicast and upon looking at the ARP table, the MAC(Media Access Control) address for the receiver device can be obtained. So the destination MAC address can be added to an ethernet frame or the data frame and then it can be passed on to the physical layer. In our case, the physical layer made use of port FastEthernet0. In the first stage, the device buffers the frames from the link layer to be sent layer.

In the second stage, the device takes out this frame from the buffer and sends it via FastEthernet0.

In the third stage, we deal with the web server. In the web server, both the in layer and the out layers are active. The in layer receives the request and the out layer generates data as per the received request. First, the physical layer of the in-layer receives the ethernet frames from the client. The frame's destination port address is used to evaluate the receiver's destination MAC address, unicast address, or broadcast address. Then in this layer, the device de-encapsulates or evaluates the data of the data frame to be passed to the network layer. In the network layer, the IP address is matched with the device's IP address or the broadcast address. Then the de-encapsulated data frame is converted into packets which are again de-encapsulated into segments that can be evaluated by the transport layer. In the transport layer, the device receives a TCP PUSH+ACK on the sender's IP address and port number 1026. The transport layer then evaluates the data before passing it to the application layer as a data segment. The server then receives an HTTP request from the data segment. Then in the out layer, the server sends back an HTTP response. The application layer prepares and passes the HTTP reply to the transport layer. And then in a similar way to the first stage, the HTTP reply is translated into ethernet frames and passed to the physical layer which transmits the HTTP reply to the client.

The last and final stage for a single request involves only the in-layer. In the in-layer, the physical layer receives the ethernet frames, evaluates them, and then sends them to its corresponding layers. It is similar to how the in-layer worked in the third stage but the key difference lies in the transport layer. The difference is that the device pops the last sent request from the buffer queue so that the same request isn't sent again by the out layer because an appropriate response is received. This is essentially a form of automatic repeat request if the package isn't received with integrity. The rest of the application of the transport layer is similar to the third stage but the data segment now contains the response which is the data for the index.html in various segments. The various segments are accumulated and processed and then passed to the application layer. The application layer then receives the processed data and displays it on the web browser of the client. And thus a request is evaluated.

Challenges:

First and foremost, I don't really know how the PUSH+ACK works and it seems a bit perplexing for me to evaluate how the bits are actually segmented, I would have to learn that further. Moreover, I am still having confusion regarding how the IP addresses are routed if there are multiple devices connected to the router. Although it requires a network address translator but in which layer the network address translator operates, and how it deals with subnets is still baffling to me. The other issue was resizing the screenshots for this document and remembering all the contents of the class, my notes weren't satisfactory and I hope to change my noting style from the next class to better accommodate the information I have to write in the report.