Lab 3: wireshark

Operative Systems

Dilan R. Ramirez

March 13, 2020

**PART 1**

**The ARP traffic**

1. Which frame numbers contain the request and response?

* Request

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Time | Source | Destination | Protocol | Length | Info |
| 1 | 0.000000 | VMware\_97: d3:0d | Broadcast | ARP | 42 | Who has 192.168.94.2? Tell 192.168.94.152 |

* Response

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Time | Source | Destination | Protocol | Length | Info |
| 2 | 0.000178 | VMware\_f2:55: b3 | VMware\_97: d3: 0d | ARP | 60 | 192.168.94.2 is at 00:50:56:f2:55:b3 |

1. What is the IP address being requested?

R = 192.168.94.2

1. What protocol layers are involved (and why)?

R = Layer 2 – Datalink because in this layer detect possible errors that may occur in the physical layer. Therefore, once an IP address is requested, so it makes an entry with the requested Mac address into an ARP table. It can partially involve layer 3 – network because it involves packets encapsulated with a source and destination in the ethernet frame. the packet includes both the sender hardware MAC address and IP.

1. Why was only one ARP request required when there are multiple hosts?

R = ARP Translate internet layer address to link-layer address. Therefore, it only needs one request using only one port to get the MAC to address in the LAN and the rest of the ports in the LAN will know the link-layer address.

**The DNS traffic**

1. Explain what information is contained in each frame (just summarize at a very high level - a few words are fine).

R = The frames 3 and 4 requests and asking for a query to the hostname looked up, using as source the IP address 192.168.94.15 and destination 192.168.94.2 which is the server. The frames 5 and 6 are the response from the server with the query already answered. Now the source is the server IP address 192.168.94.2 which contains the query already satisfied and destination is the interface of the device asking for that website. Frame 5 has a response time of 0.06856 seconds, and frame 6 has a response time of 0.07072 seconds. Both responses succeeded without errors.

1. What are the hostname being looked up and its IP address?

R = robust.cs.utep.edu - 192.168.94.152

1. What protocol layers are involved (and why)?

R = link layer, ethernet layer, network layer (IPv4), Transport layer (UDP), and Application Layer (DNS). It uses the ethernet layer to get a packet into a frame to be transmitted to the next layer. IPv4 is used to identify the devices on the network. Transport layer (UDP) establish the connection between the application on the internet, in this case, the looked-up hostname. Finally, the application layer (DNS), which maps the internet names to IP address to then use a query to require and locate the website desired.

**The HTTP traffic**

1. What URL is being requested?
   * /~freudent/test.html
2. What protocol layers are involved (and why)?
   * ethernet layer, network layer (IPv4), Transport layer (TCP), application layer (HTTP) it uses the same protocol layer used before except for TCP and HTTP. TCP is used to connect network devices with the internet. it also controls and the data transmission over the internet. HTTP is used for connecting devices with other devices across the internet. I ask the servers to send the contents of their website and submit the data using online forms.
3. Which frames contain messages related to establishing and closing the connection used for the HTTP traffic?
   * frame 8 asks messages related to establishing. frame 15 responds closing the connection.
4. What are the server's IP address and port?
   * IP address:129.108.18.226 – port: 80
5. What is the server's initial sequence number (ISN)?
   * 508
6. What are the client's IP address and port?
   * IP address: 192.168.94.152 – port: 51562
7. What is the client's ISN?
   * 160
8. Which frames contain

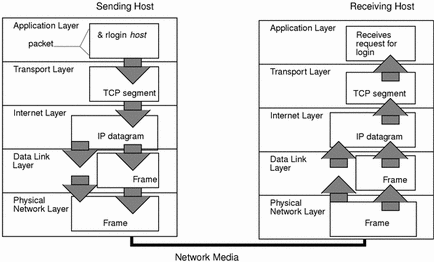
* The HTTP request
  + - frame 10
* HTTP ACK
  + - frame 9, 11, 13, 15, 16.
* HTTP headers
  + - frame 11
* HTTP response
  + - frame 12

**Part 2**

Use Wireshark to capture the interaction between your browser and the result of clicking on the “course schedule” tab of http://www.cs.utep.edu/cs/ web page. Examine this trace. Explain how HTTP is encapsulated within TCP and IP.

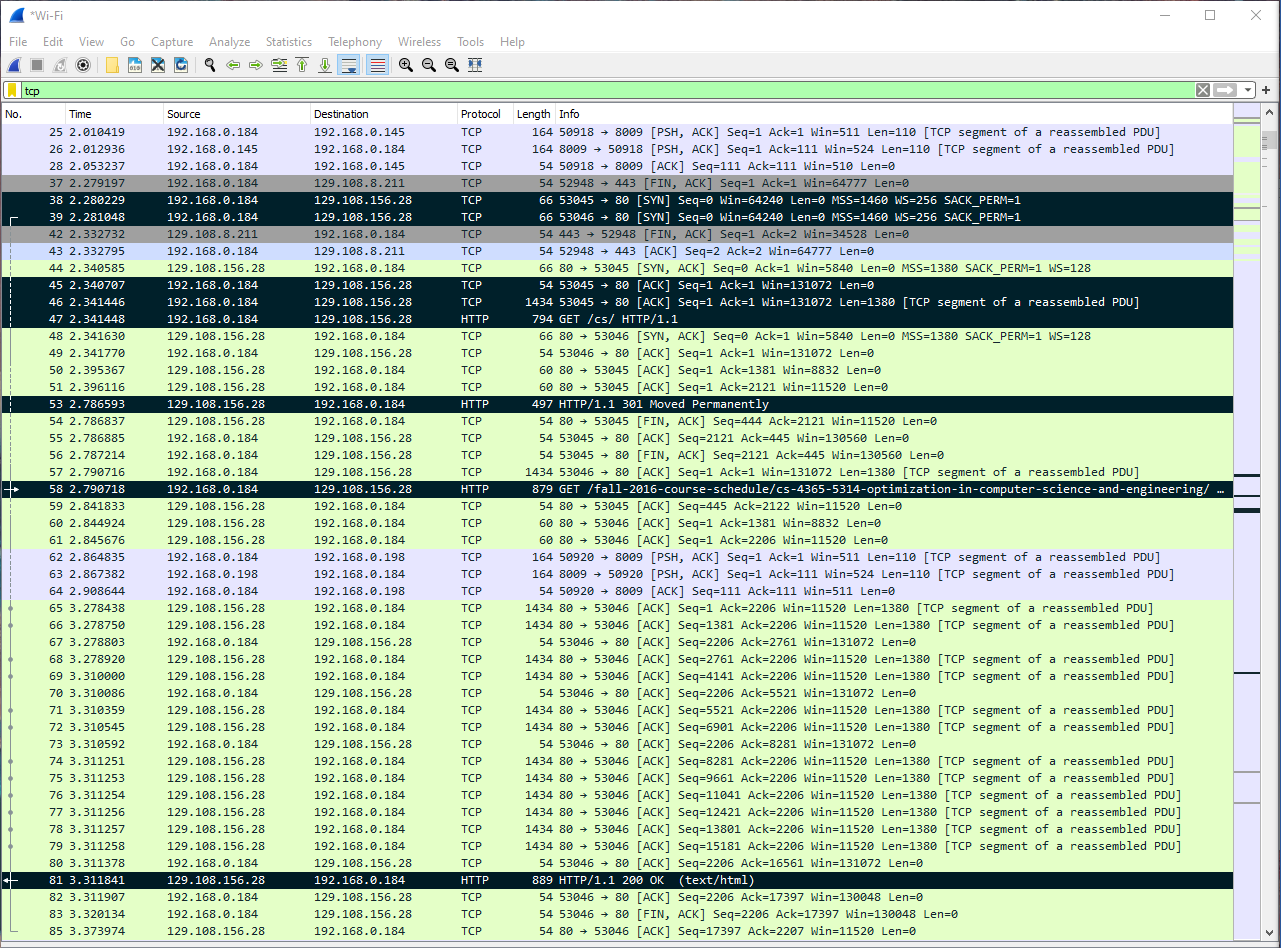
* Information from headers starts to encapsulate in frame number 38 where TCP sends a segment called SYN to establish the connection. In frame 45 TCP returns a segment called ACK to know that successfully the segment was received. In frame 46 TCP sends another ACK segment to then proceeds to send the data and establish the three-way handshake. In frame 47 the HTTP protocol proceeds to get the desired URL in this case /cs/ In frame 53, HTTP responds with the /cs/ query found. In frame 58 there is another HTTP request which is to get the “course schedule” redirection. TCP and IP keep encapsulating information in the headers to return the desired direction. The request is responded in frame 81 with “OK” to the request and returns the desired page. “https://www.utep.edu/cs/course-schedules/Spring%202020%20Course%20Schedule.html”

* Encapsulation refers to when a protocol adds data to the packet header during sending data. All the process begins in the application layer of the sending host. This packet contains the command to execute. In the transportation layer (TCP) begins the data encapsulation. TCP handles the data coming from the app layer where encapsulation starts. TCP header contains sender and recipient ports, checksum, etc. after that, the three-way handshake exchange occurs when establishing the TCP connection. Packets from TCP and UDP pass to the internet layer (IPv4). The IP protocol handles these packets to convert them in IP datagrams to being delivered to the receiving host. In the IP datagrams, the IP attaches an IP header to the packet header among the data added by TCP in this case. Now, this is encapsulation between TCP and IP. The physical layer takes the IP addresses and convert them into hardware addresses and sends the frame out to the network. The receiving host takes the frame and converts all the addresses to IP addresses and then sends them to the transport layer to finally send them to the application layer where the DNS protocol is in charge of looking for the desired URL by doing queries and getting the desired information.

**(references)**

Was the trace that you captured more difficult to analyze than the provided one? Why?

* It was more difficult because I had more packets and not just the ones that I needed. The encapsulation process was different from part 1 capture analyzed. However, after understanding the encapsulation process and looking carefully for the desired packets in Wireshark, I could succeed.



**References**

“How a Packet Travels Through the TCP/IP Stack.” Data Encapsulation and the TCP/IP Protocol Stack, Oracle Corporation, 2010, https://docs.oracle.com/cd/E19455-01/806-0916/ipov-32/index.html.