Lab 4: IP & TCP

ECSE 308 Introduction to Communication Systems and Networks

Chenyi Xu 260948311

Yongru Pan 261001758

Abstract: The lab is split into three parts. The first part of the lab focused on using the tracert utility to trace the routing path of Internet Protocol (IP) packets sent from your computer to the destination. The second section concentrate on investigating 802.11 frames specifically beacon frames and frames used for association and disassociation. Last but not least, the third section focuses on using Wireshark to collect TCP traces and investigates the TCP protocol functions including TCP connection establishment phase, TCP flow control, TCP termination phase and TCP congestion control.

Introduction

IP also known as the Internet Protocol. It is responsible for routing packets between devices across networks using unique IP addresses.

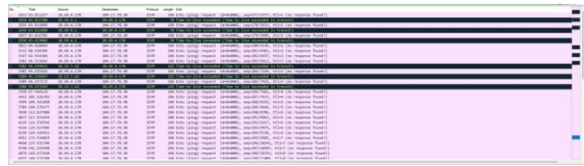
Tracer tool maps the route taken by packets from a source to a destination, providing insights into the intermediate network nodes and latency.

802.11 frames manage tasks such as beaconing, association, and data transfer between devices and access points, ensuring efficient wireless connectivity.

Transmission Control Protocol (TCP) is a transport layer protocol that provides reliable, connection-oriented communication over the internet. TCP's mechanisms, including the three way handshake, flow control, congestion control, and connection termination, are critical for ensuring accurate and orderly data delivery.

Analysis

Part I: Internet Protocol (IP)



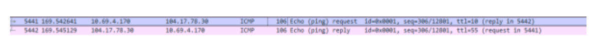
Q1: How many ICMP packets are in the list plane?

39

Q2: How many probe packets are sent from the source to the destination for each TTL?

3

Q3. The last few echo-request ICMP packets are followed by the echo-reply ICMP packets. Compare one of them with the corresponding reply. Determine which fields are similar and which fields are different? Explain the reason.



5441	169.542641	10.69.4.170	104.17.78.30	ICMP	106	Echo (ping) request id=0x0001, seq=306/12801, ttl=10 (reply in 5442)
5442	169.545129	104.17.78.30	10.69.4.170	ICMP	106	Echo (ping) reply id=0x0001, seq=306/12801, ttl=55 (request in 5441)

They have different TTL values since they follow different routes. Their source and destination are swapped. They have the same length, ID, and sequence. They have different header checksum. Request one is 0x0000, Reply one is 0x0f98.

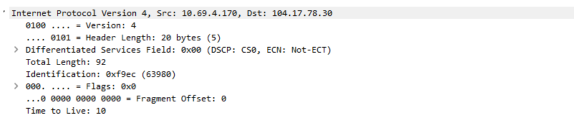
Q4. What are the TTL values for these last few packets? Determine the number of routers between the source and destination based on these TTL values?

They have a TTL request of 10 and a reply TTL value of 55. Between the source and destination, there are 10 routers.

Q5. Examine the IP packet header of the last echo-request ICMP packet, what is the value *in the Protocol field? What does this filed indicate?*

The protocol used is ICMP(1). This field indicated that the destination is reachable.

Q6. How many bytes are in the IP header? How many bytes are in the payload of this IP packet? Explain how you determined the number of payload bytes.



```
Internet Protocol Version 4, Src: 10.69.4.170, Dst: 104.17.78.30
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 92
    Identification: 0xf9ec (63980)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 10
```

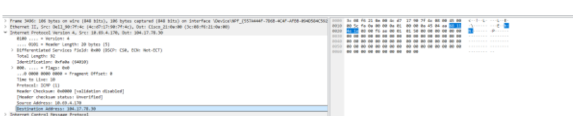
The header contains 20 bytes. The total length is 92 bytes. The payload can be calculated by subtracting the header from total which is equal to 72 bytes for the payload.

Q7. Has this IP packet been fragmented? Explain how you determined whether or not the packet has been fragmented.

```
   0... .... = Reserved bit: Not set
   .0.. .... = Don't fragment: Not set
   ..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
```

No, because the bits destinated for the fragmentation (fragmentation offset) is set to be 0. If it were to be fragmented, the bit value would be set to 1.

Q8. How the IP address of www.acm.org can be found? Determine the packet and the field in the packet that contains this information.



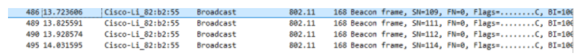
It can be found from the destination address field of echo request packet. It is 68 11 4e 1e

Part II: 802.11 frames

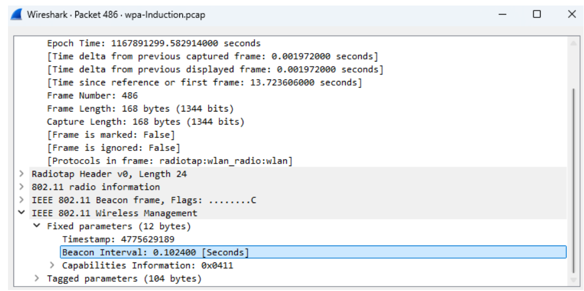
Q1: What is the SSID of the access point that is issuing the beacon frame?

The SSID is "Coherer".

Q2: What are the time intervals between transmissions of beacon frames? Does the beacon frame contain this information?



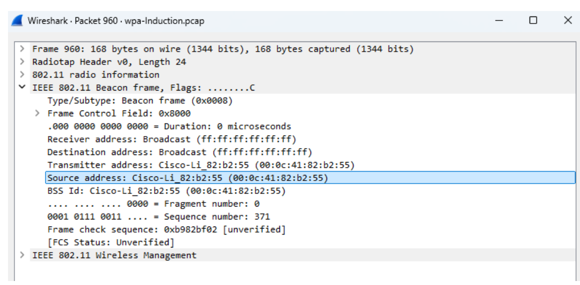
```
486 13.723606   Cisco-Li_82:b2:55   Broadcast   802.11   168 Beacon frame, SN=109, FN=0, Flags=........C, BI=10
489 13.825591   Cisco-Li_82:b2:55   Broadcast   802.11   168 Beacon frame, SN=111, FN=0, Flags=........C, BI=10
490 13.928574   Cisco-Li_82:b2:55   Broadcast   802.11   168 Beacon frame, SN=112, FN=0, Flags=........C, BI=10
495 14.031595   Cisco-Li_82:b2:55   Broadcast   802.11   168 Beacon frame, SN=114, FN=0, Flags=........C, BI=10
```



```
Wireshark · Packet 486 · wpa-Induction.pcap
    Epoch Time: 1167891299.582914000 seconds
    [Time delta from previous captured frame: 0.001972000 seconds]
    [Time delta from previous displayed frame: 0.001972000 seconds]
    [Time since reference or first frame: 13.723606000 seconds]
    Frame Number: 486
    Frame Length: 168 bytes (1344 bits)
    Capture Length: 168 bytes (1344 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: radiotap:wlan_radio:wlan]
> Radiotap Header v0, Length 24
> 802.11 radio information
> IEEE 802.11 Beacon frame, Flags: ........C
v IEEE 802.11 Wireless Management
  v Fixed parameters (12 bytes)
        Timestamp: 4775629189
        Beacon Interval: 0.102400 [Seconds]
      > Capabilities Information: 0x0411
  > Tagged parameters (104 bytes)
```

The Beacon frame includes this information in the Fixed Parameters->Beacon Interval as 0.1024 seconds

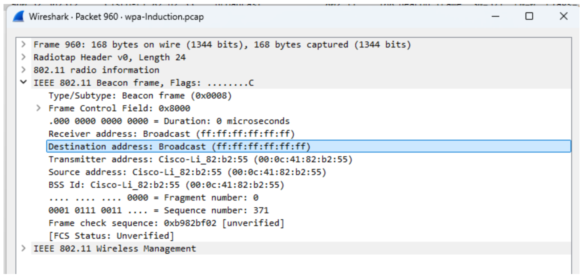
Q3: What is the source MAC address in the beacon frame?



```
Wireshark · Packet 960 · wpa-Induction.pcap
> Frame 960: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits)
> Radiotap Header v0, Length 24
> 802.11 radio information
v IEEE 802.11 Beacon frame, Flags: ........C
    Type/Subtype: Beacon frame (0x0008)
  > Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: Cisco-Li_82:b2:55 (00:0c:41:82:b2:55)
    Source address: Cisco-Li_82:b2:55 (00:0c:41:82:b2:55)
    BSS Id: Cisco-Li_82:b2:55 (00:0c:41:82:b2:55)
    .... .... .... 0000 = Fragment number: 0
    0001 0111 0011 .... = Sequence number: 371
    Frame check sequence: 0xb982bf02 [unverified]
    [FCS Status: Unverified]
> IEEE 802.11 Wireless Management
```

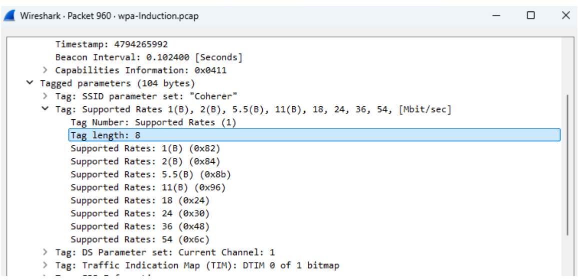
The source MAC address is 00:0c:41:82:b2:55 as shown in the above screenshot.

Q4: What is the destination MAC address in the beacon frame? What does this address mean?



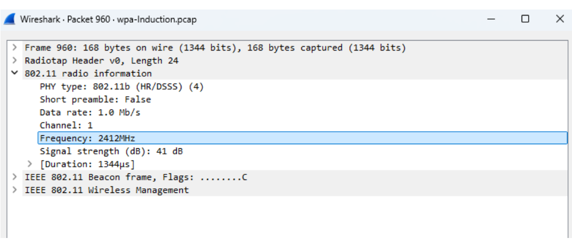
The destination address is (ff:ff:ff:ff:ff:ff) as shown in the above screenshot. It is a broadcast address that send to all device within the range.

Q5: How any data rates can the access point support?



8 data rates can access point support.

Q6: Examine the beacon frame. What frequency does the advertised network use?



It uses the frequency of 2412MHz.

Q7: By looking at the list plane, indicate what type of packets have the smallest size? What type has the largest size?

The packets of “Acknowledgement” has the smallest size. The packets of “Data” has the largest size

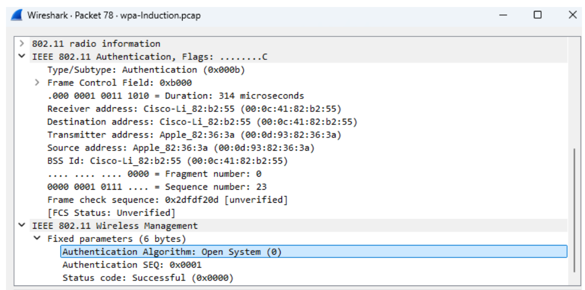
Q8. Before sending data to Apple_82:36:3a, what frames are exchanged between this device and the access point?

Probe Request -> Probe Response

Authentication Request -> Authentication Response

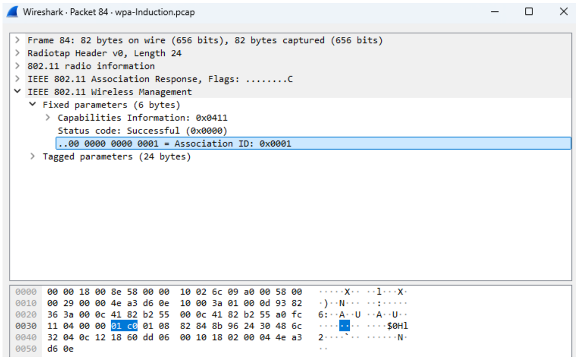
Association Request -> Association Response

Q9. Examine the Authentication frame sent by Apple_82:36:3a. Does the host want the authentication to require a key or be open?



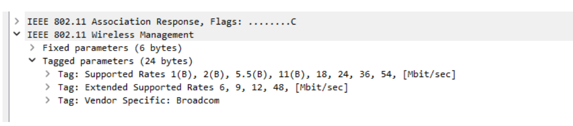
It says that it is an open system, so no key is required.

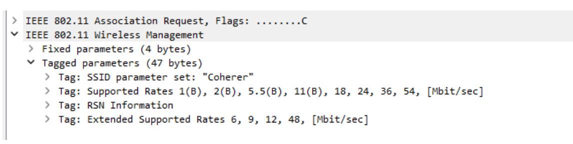
Q10. Examine the response Authentication frame sent by the AP to Apple_82:36:3a. What is the Association ID for this host? What is the usage of this ID?



The association ID for this host is 0x0001. The ID is used to identify the client's device within the network.

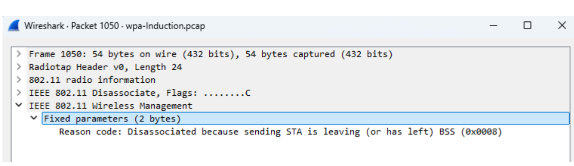
Q11. What transmission rates is the host willing to use? The AP? To answer this question, you will need to look into the parameter fields of the 802.11 wireless LAN management frame.





They are both willing to use the same rate. Which means that in this case, they will be able to connect.

Q12. Examine the Disassociation frame sent by Apple_82:36:3a to the AP. What is the reason that this user sent the Disassociation frame?



The reason is that the sending STA is leaving.

Part III: TCP

Q1. How many TCP datagrams are exchanged between your computer and the server to establish the TCP connection? Why each of these segments is needed to setup the TCP connection.

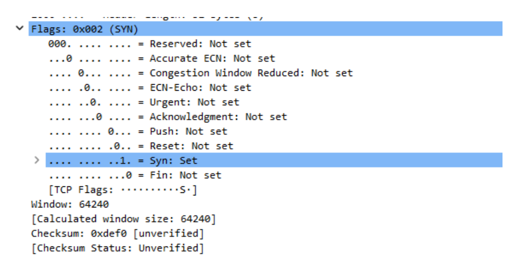
A total of three TCP datagrams are exchanged during the handshake process. It is referred as TCP three way handshake. First segment is SYN, the client imitates a connection by sending the SYN packet with the sequence number. The second segment is SYN-ACK, the server acknowledges the SYN packet by sending a SYN-ACK packet. The acknowledgement number is the client's sequence number plus 1. The SYN-ACL flag are sets indicating the server is ready to establish a connection, and it acknowledges the client's initial SYN. The last segment is ACK, The client, upon receiving the SYN-ACK packet, acknowledges the acknowledgement by sending a final ACK packet.

Q2: Which end point started the TCP Connection Establishment phase?

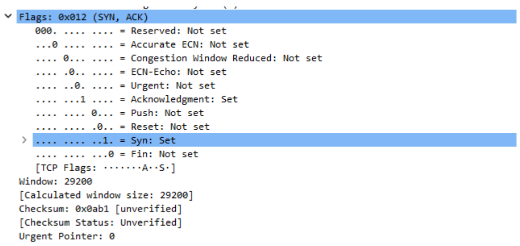
The client end which is me for this lab.

Q3: Which flags are set in each of these TCP datagrams?

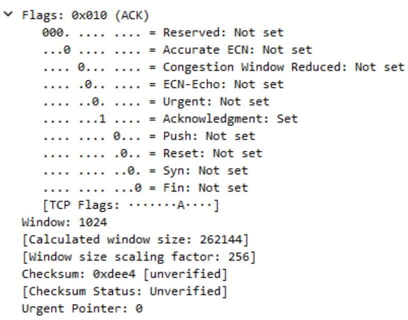
For SYN, SYN flag is set.



For SYN ACK, Acknowledgement and Syn flags are set.



For ACK, Acknowledgement is set.



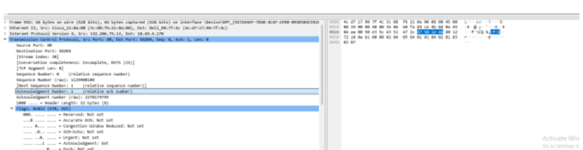
Q4: What is the initial value of the sequence number on the client's side?

0

Q5: What is the intimal value of the sequence number on the server's side?

0

Q6: What is the value of the acknowledgement field in the SYN ACK datagram? How did the server determine the value?

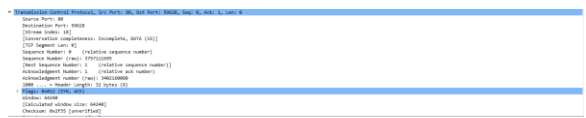


Q7. For the TCP SYN datagram, determine the following

a. Source port number: 59628
b. Destination port number : 80
c. Size of the window: 64240
d. The header length: 32 Bytes

Q8. For the TCP SYN ACK datagram, determine the following

a. The source port number: 80
b. The destination port number: 59628
c. The size of the window: 64240
d. The header length: 32 Bytes



Q9. What is the usage of the window field in the TCP segments?

The window field in TCP segments is used for flow control. It specifies the maximum number of bytes the sender is allowed to transmit before receiving an acknowledgment (ACK) from the receiver. This ensures efficient data transmission without overwhelming the receiver's buffer capacity.

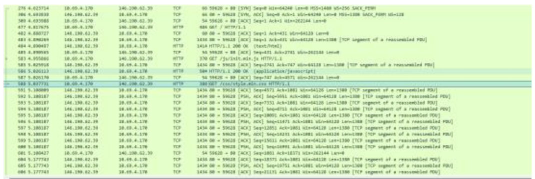
Q10. Consider the TCP segment containing the HTTP GET as the first segment in the TCP connection. For the first three TCP segments, answer the following questions:

a. When was each segment sent?

477 4.817675

583 4.955866

588 5.037731



b. At which time was the ACK for each segment received?

482 4.888727

585 5.025918

591 5.108009

c. Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the three segments?

Difference 1 = 4.888727 - 4.817675 = 0.069595

Difference 2 = 5.025918 - 4.955866 = 0.070052

Difference 3 = 5.037731 - 5.037731 = 0

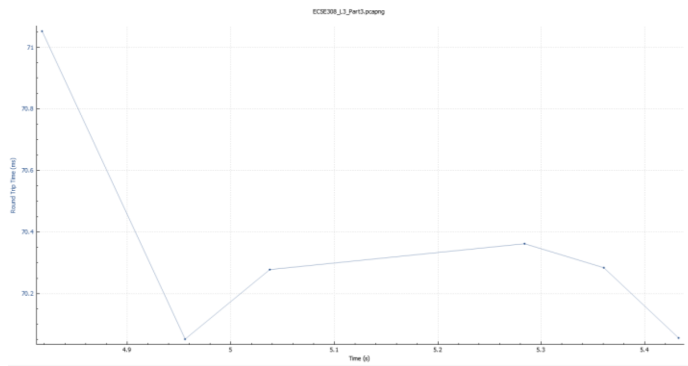
RTT 1 = 71.05 ms
RTT 2 = 70.05 ms
RTT 3 = 70.25 ms

d. What is the estimated RTT value after the receipt of each ACK?

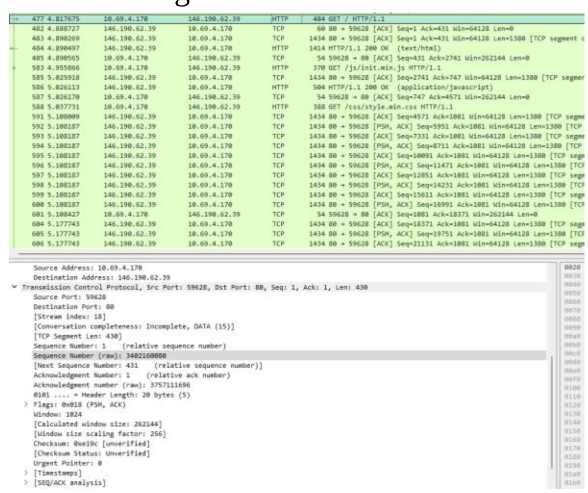
Estimated RTT 1 = (1 – 0.125) * 110 ms + 0.125 * (71.05 ms) = 105.13 ms

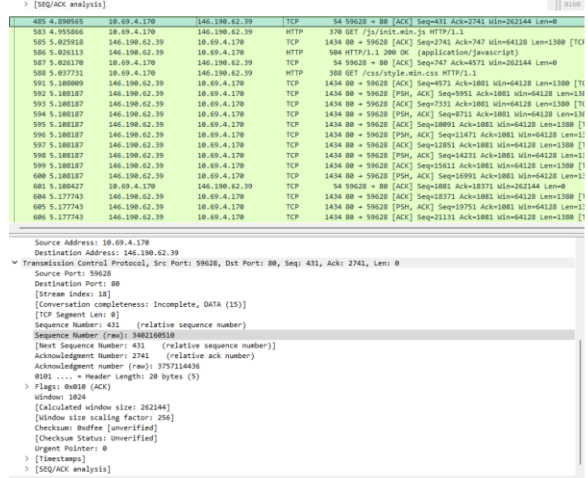
Estimated RTT 2 = (1 – 0.125) * 110 ms + 0.125 * (70.05 ms) = 105.01 ms

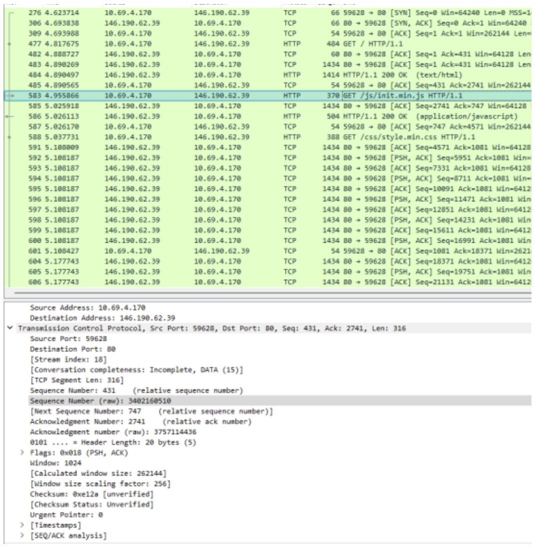
Estimated RTT 3 = (1 – 0.125) * 110 ms + 0.125 (70.25 ms) = 105.03 ms

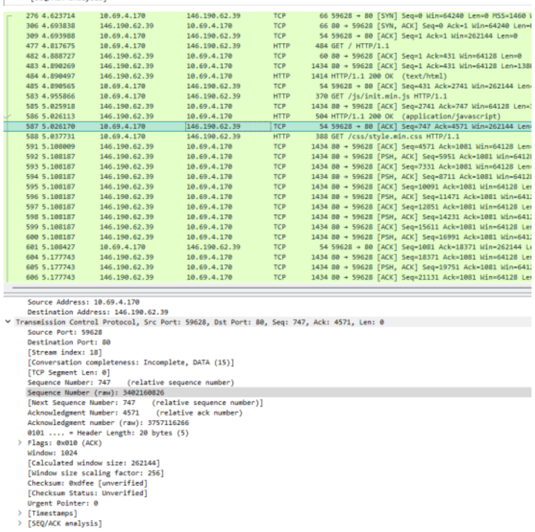


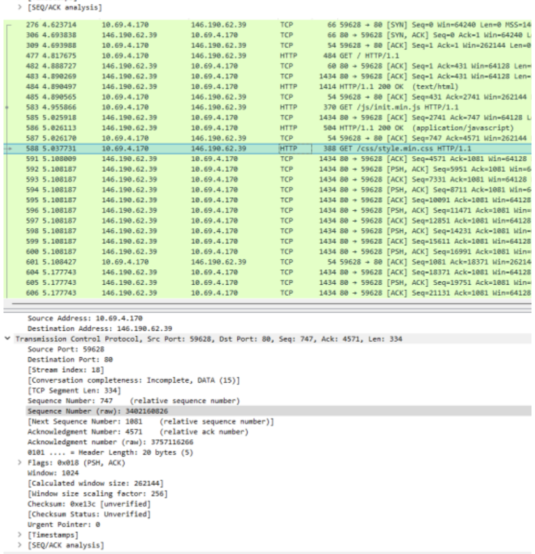
e. What is the length of each of the first three TCP segments?

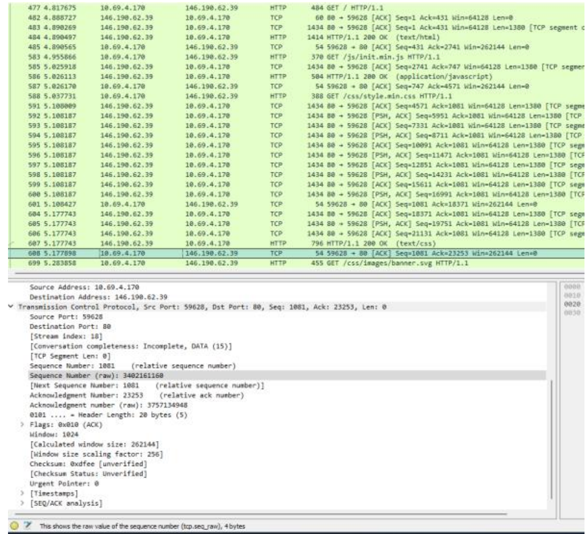












Length = ACK # - initial ACK #

Length 1 = 2741 – 1 = 2740

Length 2 = 4571 – 2741 = 1830

Length 3 = 23253 – 4571 = 18682

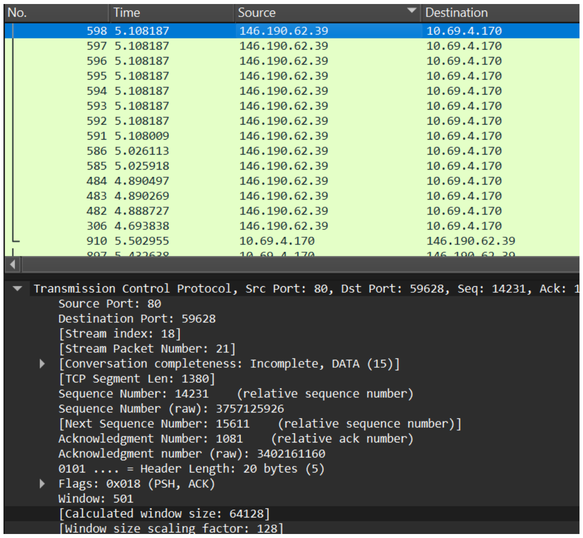
Q11: Are the client's port number and the server's port number the same in the entire trace? What is the usage of the port number?

Yes. They are the same in the entire trace. The client's port number is 59628 and the server's port number is 80, so the two of them are not the same.

The client's port number is used to identify the application or process initializing the connection on the clients' device.

The server's port number is used to communicate with the HTTP service running on the server.

Q12: What is the minimum amount of available buffer space advertised at the *received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?*



The minimum amount of available buffer space advertised at the received for the entire trace is 64128. Because this is the minimum calculated window size found from all of the frames that are sent from the server to this computer.

No, it has not throttled the sender, because the minimum amount of windows size has never reached zero.

Q13: Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

There are no retransmitted segments in this trace file. The trace info are checked to reassure,

Q14: How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment.

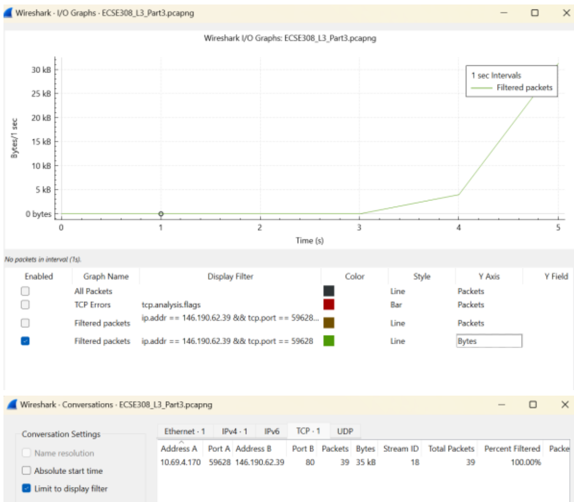
From the screenshot, and if we set the client as the receiver, we observed the following information:

No. Packet	Source	ACK number	Seq Number
309	10.69.4.170	1	1
485	10.69.4.170	2741	431
587	10.69.4.170	4571	747
601	10.69.4.170	18371	1081
608	10.69.4.170	23253	1081
910	10.69.4.170	30922	2272

The amount of data acknowledged varies between 1830 bytes (minimum) and 13800 bytes (maximum). The jumps between ACK numbers represent the number of bytes received and successfully processed before acknowledgment.

There are cases where the receiver appears to acknowledge every other segment. For example, from packet 485 to 587 and from packet 601 to 608, the data acknowledgment skips intermediate values, suggesting delayed or selective ACK behavior.

Q15: Calculate the throughput (bytes transferred per unit time) for the TCP connection? Explain how you obtained this value.



From the result obtained in the I/O graph as well as the conversation window, the throughput can be calculated as follows.

In the I/O graph's x-axis, we have obtained that the total time is 5 seconds and from the conversation tab's byte column the total data transferred during the TCP connection can also be obtained as 35kB. The throughput is equal to Total Bytes Transferred/Total Time = 35kB/7s = 7000 bytes/second

Q16: How many TCP datagrams are exchanged for the termination phase?

If the termination is normal and is not interrupted by an RST, then the number of datagrams involved is 4. According to the lecture "client, server each close their side of connection: send TCP segment with FIN bit = 1. They then each respond to received FIN with ACK: on receiving FIN, ACK can be combined with own FIN"

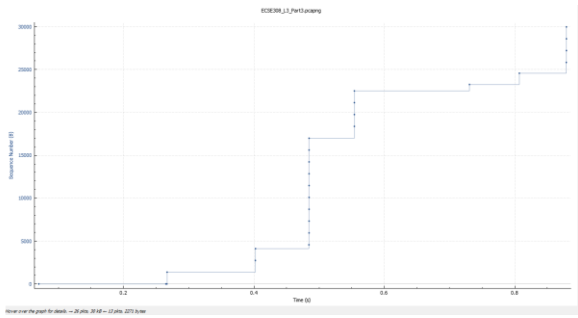
Q17: Which end point started the Connection Termination phase?

From this screenshot, it is the 10.69.4.170 end point that started the termination phase.

Q 18. What flags are set in each of the segments used for connection termination?

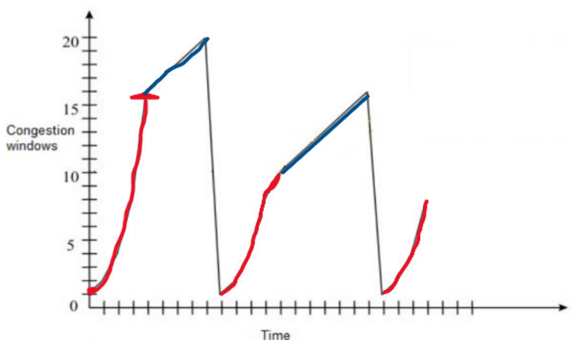
The first [FIN, ACK] is sent by the client to initiate connection termination. The first [ACK] is sent by the server to acknowledge the client's FIN. The second [FIN, ACK] is sent by the server to terminate the connection to the client. The second [ACK] is sent by the client to acknowledge the connection.

Q19: Use the Time-Sequence-Graph (Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the server. Can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over? Explain your answer.



There is no congestion avoidance. Because there is no sign of a linear growth in the sequence number which it is usually characterized by.

20. Locate the different phases of the congestion control mechanism on the below graph. Also describe the congestion control algorithm.



In the above graph, the red part is the slow start phase, and the blue part is the congestion avoidance phase.

The congestion algorithm has two phases. The first one is when cwnd < ssthresh, this is the slow start phase. This is when the congestion window start small and grows exponentially. This phase starts with when cwnd equals to 1 maximum segment size, and each time isand ACK is received, the cwnd doubles. The second phase is when cwnd >= ssthresh, the TCP is in congestion avoidance phase, where the TCP only has linear growth to prevent congestion. The cwnd in this phase grows more slowly compared to when in the slow start phase. These two phases will alternate according to the real time network feedback.

Conclusion

This lab provided hands-on experience with IP, 802.11 wireless frames, and TCP protocols. We explored routing paths, packet fragmentation, and IP header structures, examined 802.11 management frames, and analyzed TCP operations such as connection establishment, flow control, and termination using Wireshark. These exercises enhanced our understanding of network protocols and their role in ensuring reliable and efficient communication.