ASSIGNMENT-4

21CY681– INTERNET PROTOCOL LAB

NAME : RAHUL RAJ

REGISTER NUMBER: CYS22011

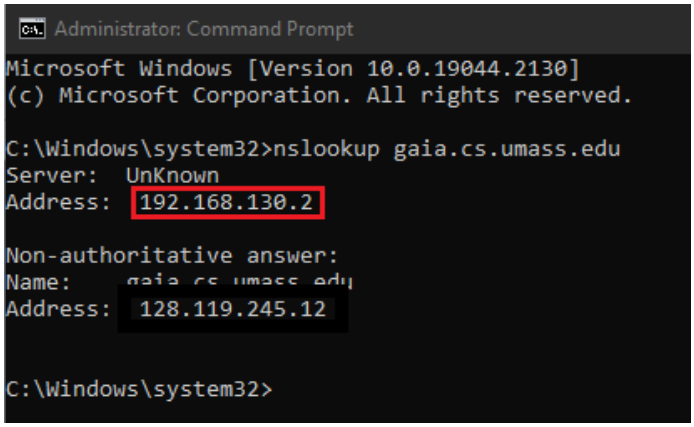
TITLE : ANALYZING TRANSPORT LAYER PROTOCOLS USING WIRESHARK

DATE OF ASSIGNMENT PROVIDED: 27/10/2022

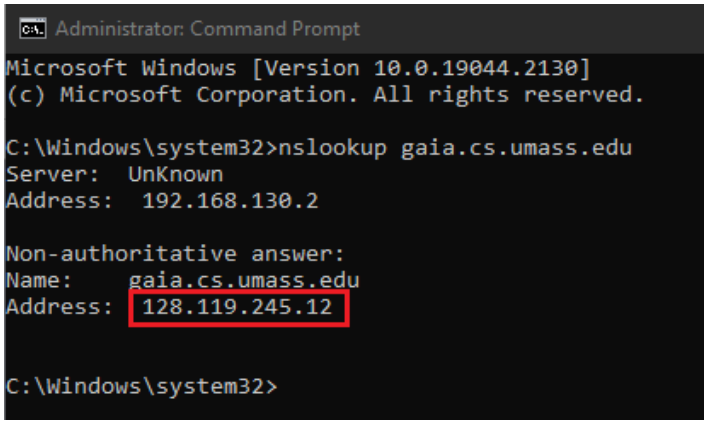
AIM: To analyze transport layer protocols using Wireshark.

TCP: -

1, a) What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.u.edu?



b) What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?



c) What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

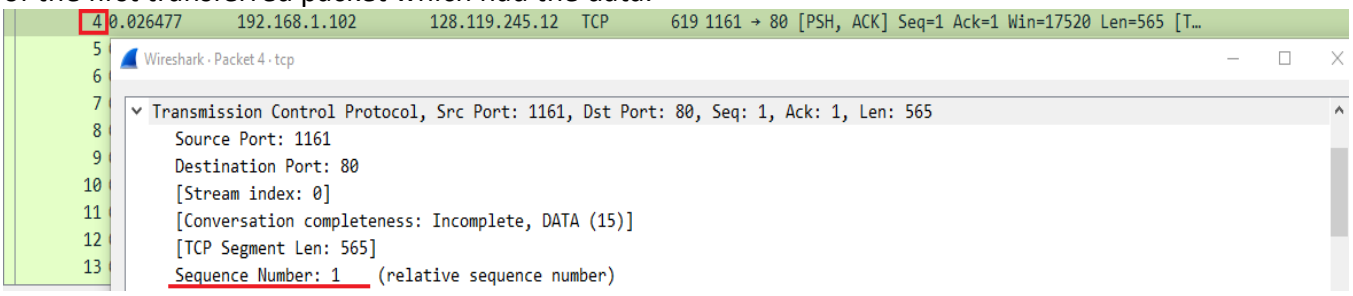
```
TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_P…
Wireshark · Packet 1 · tcp
  0111 .... = Header Length: 28 bytes (7)
  Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Accurate ECN: Not set
    .... 0... .... = Congestion Window Reduced: Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
```

d) What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

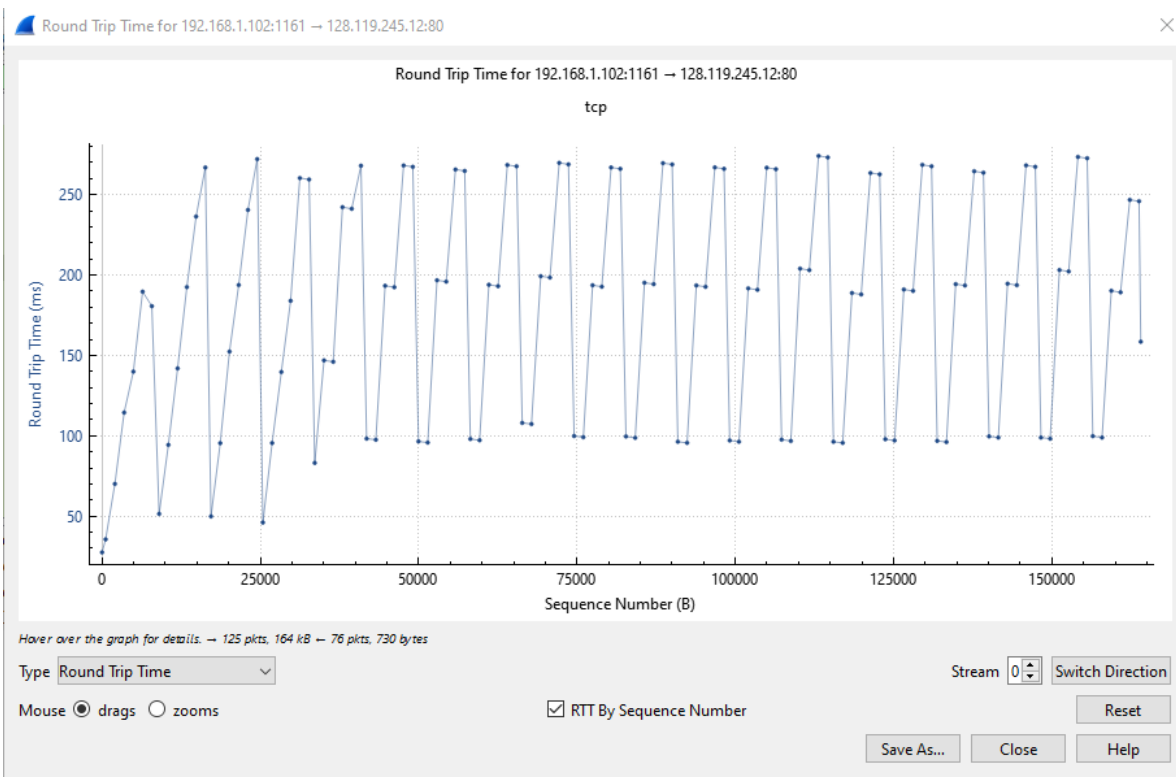
```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 883061786
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 3486    (relative ack number)
Acknowledgment number (raw): 232132498
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
  000. .... .... = Reserved: Not set
  ...0 .... .... = Accurate ECN: Not set
  .... 0... .... = Congestion Window Reduced: Not set
  .... .0.. .... = ECN-Echo: Not set
  .... ..0. .... = Urgent: Not set
  .... ...1 .... = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... .... .0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  .... .... ...0 = Fin: Not set
  [TCP Flags: ·******A····]
```

The value of ACK in SYNACK segment is 1 and the sequence number of the SYNACK segment sent is 0.

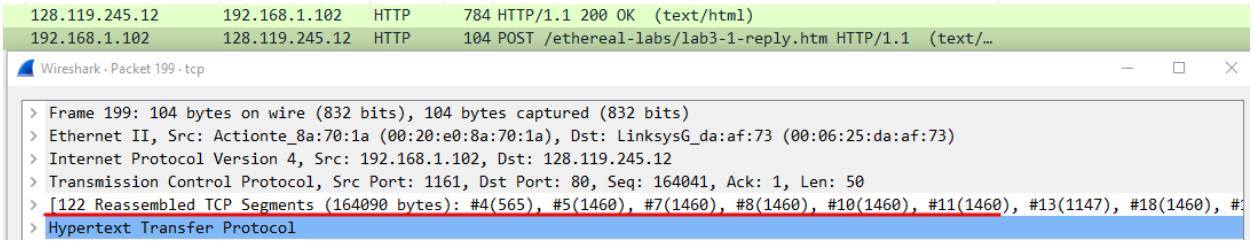
e) What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field. In order to find the SEQ number of the POST command, we need to see the SEQ number of the first transferred packet which had the data.



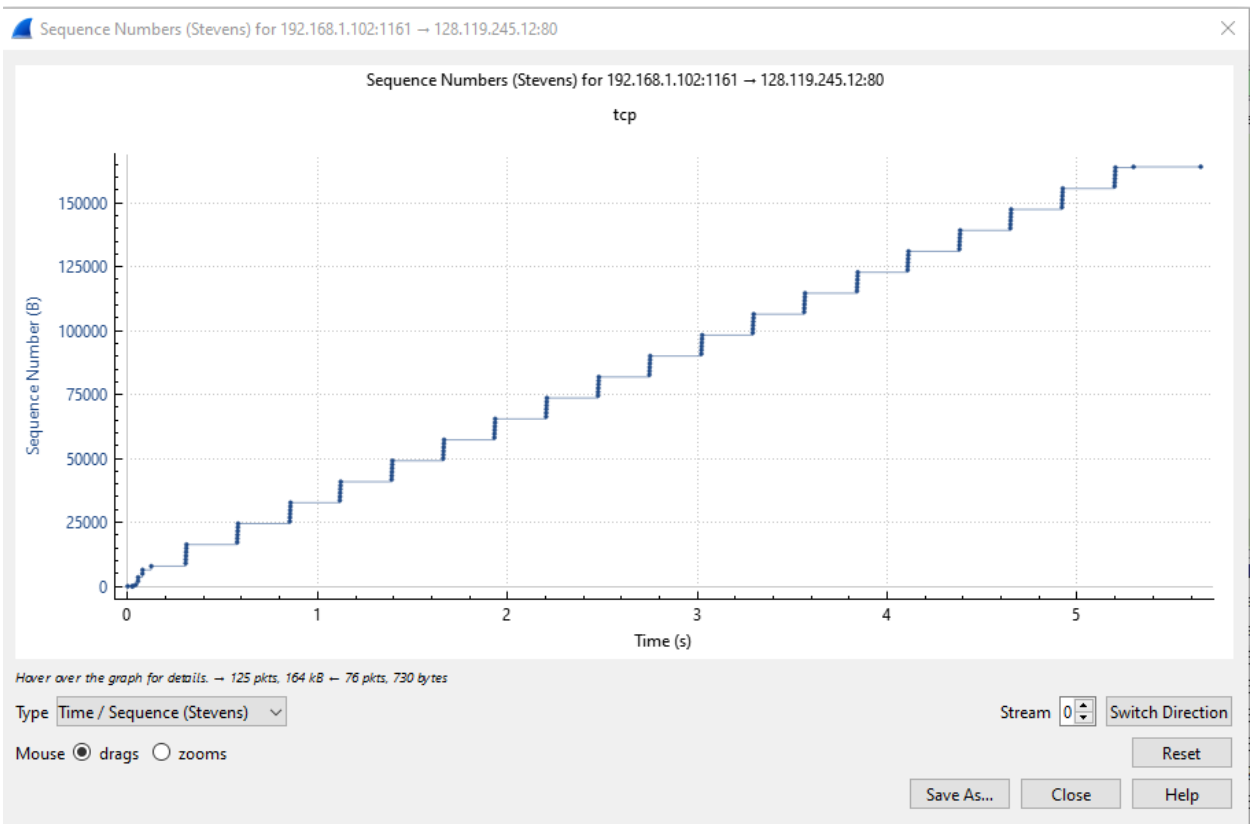
f) Plot the RTT graph using Wireshark.



g) What is the length of each of the first six TCP segments (HTTP POST)?



h) 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 retransmitting segments in the file.

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

Throughput = Total amount of data transferred / Total amount of time the final value of ACK packet is 164091, so the total amount of data transferred is 164091.

```
60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
```

Time of first packet since reference is 0.026477000

```
[Time since first frame in this TCP stream: 0.026477000 seconds]
```

Time of last packet since reference is 5.455830000

```
[Time since first frame in this TCP stream: 5.455830000 seconds]
[Time since previous frame in this TCP stream: 0.007943000 seconds]
```

Throughput = 164090/(5.455830000 – 0.026477000) = 302222 bytes => **<u>30 kilobytes per second</u>**

<u>UDP: -</u>

j) Select one UDP packet from your trace. From this packet, determine how many fields the are in the UDP header. Name these fields.

There are 4 fields in UDP header

No.	Time	Source	Destination	Protocol	Length
1	0.000000	192.168.1.102	192.168.1.104	SNMP	92
2	0.016960	192.168.1.104	192.168.1.102	SNMP	93
11	3.016971	192.168.1.102	192.168.1.104	SNMP	92
12	3.034127	192.168.1.104	192.168.1.102	SNMP	93
13	6.033719	192.168.1.102	192.168.1.104	SNMP	92
14	6.050808	192.168.1.104	192.168.1.102	SNMP	93

```
> Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
> Ethernet II, Src: Dell_4f:36:23 (00:08:74:4f:36:23), Dst: HewlettP_61:eb:e
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 192.168.1.104
v User Datagram Protocol, Src Port: 4334, Dst Port: 161
    Source Port: 4334
    Destination Port: 161
    Length: 58
    Checksum: 0x65f8 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  v [Timestamps]
      [Time since first frame: -0.016960000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (50 bytes)
```

k) By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.

Source Port (udp.srcport), 2 bytes

<u>Total 8 bytes</u> (2x4 fields=8).

l) The value in the Length field is the length of what? Verify your claim with your captured UDP packet.

```
    Source Port: 137
    Destination Port: 137
    Length: 70
    Checksum: 0x3eea [unverified]
    [Checksum Status: Unverified]
    [Stream index: 11]
  ⌄ [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (62 bytes)
```

m) What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation.

```
  ⌄ 000. .... = Flags: 0x0
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.102
    Destination Address: 192.168.1.104
```

The protocol number for UDP is 17. In hexadecimal it is 0x11.

n) Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. (Hint: for a second packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets.

When 4334 is the source address the destination is 161 in request. In response it is the exact opposite.

Destination	Protocol	Length	Info	Source code	
192.168.1.104	SNMP	92	get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0		4334
192.168.1.102	SNMP	93	get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0		161