

# 21CY681 - INTERNET PROTOCOL LAB

Name: KARTHIKA P

Date: 27<sup>th</sup> October 2022

Assignment Topic: TCP AND UDP

Register Number: CB. EN. P2CYS22001

## AIM:

To analyse TCP and UDP using Wireshark.

## APPARATUS REQUIRED:

Wireshark

## QUESTIONS:

1. Open the pcap file "tcp" in Wireshark to answer the following questions.

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

2 0.023172	128.119.245.12	80 192.168.1.102	1161 TCP	62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 S
3 0.023265	192.168.1.102	1161 128.119.245.12	80 TCP	54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0

Source ip address :192.168.1.102

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?

Ip address of gaia.cs.umass.edu:128.119.245.12

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?

[SYN] Seq=0 Win=16384 Len=0 MSS=1460 S

Sequence number for TCP SYN segment is 0.

```

Flags: 0x002 (SYN)
Window: 16384
[Calculated window size: 16384]
Checksum: 0xf6e9 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (8 bytes), Maximum segme

```

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?

```

[SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK

```

Sequence number of SYNACK segment is 0.

The value of acknowledgement field is 1.

Since SYN , ACK is notified in that sequence.

```

Flags: 0x012 (SYN, ACK)
Window: 5840
[Calculated window size: 5840]
Checksum: 0x774d [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (8 bytes), Maximum segme

```

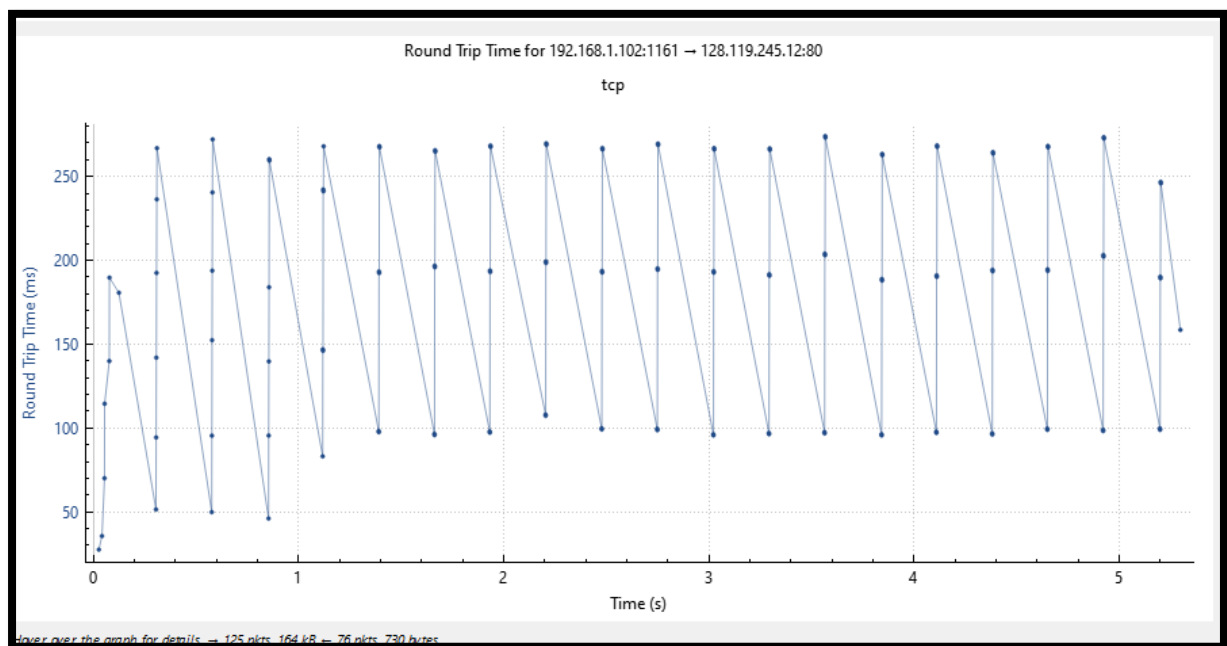
e) What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

```

4 0.026477 192.168.1.102 1161 128.119.245.12 80 TCP 619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP se

```

f) Plot the RTT graph using Wireshark.



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

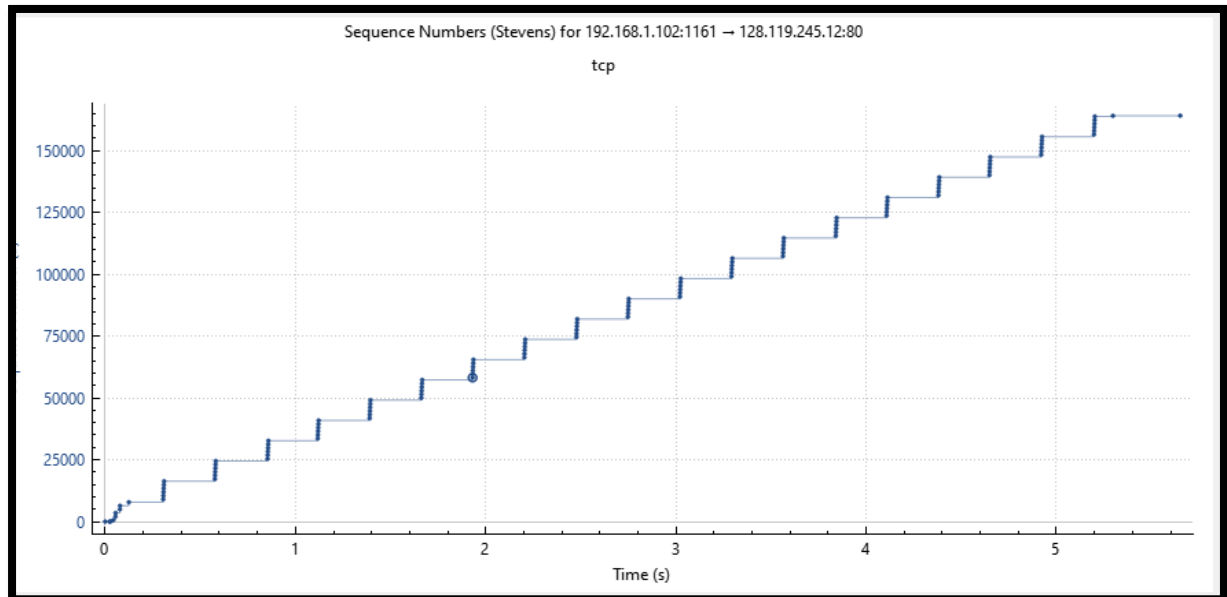
Frame: 4, payload: 0-564 (565 bytes)]  
Frame: 5, payload: 565-2024 (1460 bytes)]  
Frame: 7, payload: 2025-3484 (1460 bytes)]  
Frame: 8, payload: 3485-4944 (1460 bytes)]  
Frame: 10, payload: 4945-6404 (1460 bytes)]  
Frame: 11, payload: 6405-7864 (1460 bytes)]

619 1161 -> 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP segment of a reassembled PD...

The length of each segments is separated as 565 bytes.

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

Since there is no dropping in the graph, so there is no retransmitted segments.



i) What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Throughput - total amount data / total amount of time

First seq number Pack no-4 and last ack, pack no- 202

Data:  $(202 - \text{packet no})164091 - 1(4 - \text{packet no}) = 164090$  Time:  $5.455830000 - 0.026477 = 5.429353$

And divide with data 0.300 converted to kilo bites = 30.2 kB/sec

2. Open the pcap file "udp" in Wireshark to answer the following questions

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

```
Source Port: 4334
Destination Port: 161
<Source or Destination Port: 4334>
<Source or Destination Port: 161>
Length: 58
Checksum: 0x65f8 [unverified]
[Checksum Status: Unverified]
```

There are four fields ,they are source port, destination port, length,checksum.

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.

The length of source port is 2 bytes.

Source Port (udp.srcport), 2 bytes

the length of the data length is 2 bytes.

Length in octets including this header and the data (udp.length), 2 bytes

The length of destination port is 2 bytes.

Destination Port (udp.dstport), 2 bytes

The length of checksum is 2 bytes.

Details at: [https://www.wireshark.org/docs/wsug\\_html\\_chunked/ChAdvChecksums.html](https://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html) (udp.checksum), 2 bytes

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
<Source or Destination Port: 137>
<Source or Destination Port: 137>
Length: 70
Checksum: 0x3eea [unverified]
[Checksum Status: Unverified]
[Stream index: 11]
[Timestamps]
UDP payload (62 bytes)

```

Here , the payload is 62 bytes and 8 bytes for the four header.

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

The protocol number of udp is 17.

```

Time to Live: 128
Protocol: UDP (17)

```

```

ce 00 08 74 4f 36 23 08 0
00 80 11 00 00 c0 a8 01 6
89 00 46 3e ea 97 f2 85 0
00 20 45 4f 45 50 45 49 4

```

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.

Time	Source	Source port	Destination	Destination port	Protocol	Length	Info
1 0.000000	192.168.1.102	4334	192.168.1.104	161	SNMP	92	get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.1.0
2 0.016960	192.168.1.104	161	192.168.1.102	4334	SNMP	93	get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.1.0

Here we can identify that from source ip has been transferred to the destination and then from destination ip it got response to the source.

## RESULT:

Thus, we have analysed TCP and UDP using wireshark.

