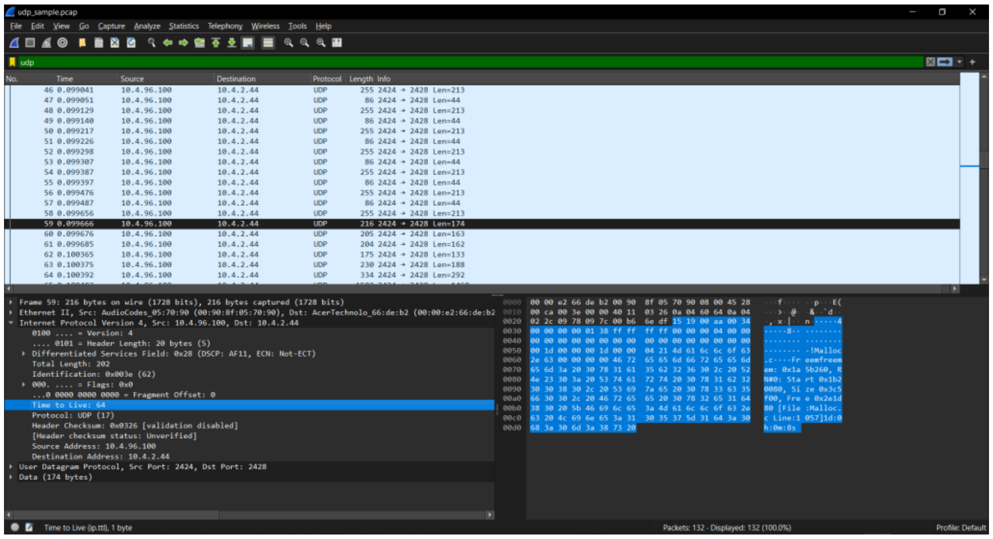
Alp Kural - 71959
Computer Engineering

Comp 416 - Transport Layer Protocols’ Experiments with Wireshark Project

Part 1.1

Question 01



```
udp_sample.pcap
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Wireless  Tools  Help
udp

No.   Time      Source        Destination  Protocol Length Info
   46 0.099041  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   47 0.099051  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   48 0.099129  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   49 0.099140  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   50 0.099217  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   51 0.099226  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   52 0.099298  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   53 0.099307  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   54 0.099387  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   55 0.099397  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   56 0.099476  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   57 0.099487  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   58 0.099656  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   59 0.099666  10.4.96.100   10.4.2.44    UDP       216 2424 → 2428 Len=174
   60 0.099676  10.4.96.100   10.4.2.44    UDP       205 2424 → 2428 Len=163
   61 0.099685  10.4.96.100   10.4.2.44    UDP       204 2424 → 2428 Len=162
   62 0.100365  10.4.96.100   10.4.2.44    UDP       175 2424 → 2428 Len=133
   63 0.100375  10.4.96.100   10.4.2.44    UDP       230 2424 → 2428 Len=188
   64 0.100392  10.4.96.100   10.4.2.44    UDP       334 2424 → 2428 Len=292

› Frame 59: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits)
› Ethernet II, Src: AudioCodes_05:70:90 (00:90:8f:05:70:90), Dst: AcerTechnolo_66:de:b2 (00:00:e2:66:de:b2
⌄ Internet Protocol Version 4, Src: 10.4.96.100, Dst: 10.4.2.44
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x28 (DSCP: AF11, ECN: Not-ECT)
    Total Length: 202
    Identification: 0x003e (62)
  › 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x0326 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.4.96.100
    Destination Address: 10.4.2.44
› User Datagram Protocol, Src Port: 2424, Dst Port: 2428
› Data (174 bytes)

0000  00 00 e2 66 de b2 00 90  8f 05 70 90 08 00 45 28   ···f···· ··p···E(
0010  00 ca 00 3e 00 00 40 11  03 26 0a 04 60 64 0a 04   ···>··@· ·&··`d··
0020  02 2c 09 78 09 7c 00 b6  6e df 15 19 00 aa 00 34   ·,·x·|·· n······4
0030  00 00 00 00 01 38 ff ff  ff ff 00 00 00 04 00 00   ·····8·· ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 1d 00 00 00 1d 00 00  04 21 4d 61 6c 6c 6f 63   ········ ·!Malloc
0060  2e 63 00 00 00 00 46 72  65 65 6d 66 72 65 65 6d   .c····Fr eemfreem
0070  65 6d 3a 20 30 78 31 61  35 62 32 36 30 2c 20 52   em: 0x1a 5b260, R
0080  4e 23 30 3a 20 53 74 61  72 74 20 30 78 31 62 32   N#0: Sta rt 0x1b2
0090  30 30 38 30 2c 20 53 69  7a 65 20 30 78 33 63 35   0080, Si ze 0x3c5
00a0  66 30 30 2c 20 46 72 65  65 20 30 78 32 65 31 64   f00, Fre e 0x2e1d
00b0  38 30 20 5b 46 69 6c 65  3a 4d 61 6c 6c 6f 63 2e   80 [File :Malloc.
00c0  63 20 4c 69 6e 65 3a 31  30 35 37 5d 31 64 3a 30   c Line:1 057]1d:0
00d0  68 3a 30 6d 3a 38 73 20                            h:0m:8s

Time to Live (ip.ttl), 1 byte          Packets: 132 - Displayed: 132 (100.0%)          Profile: Default
```

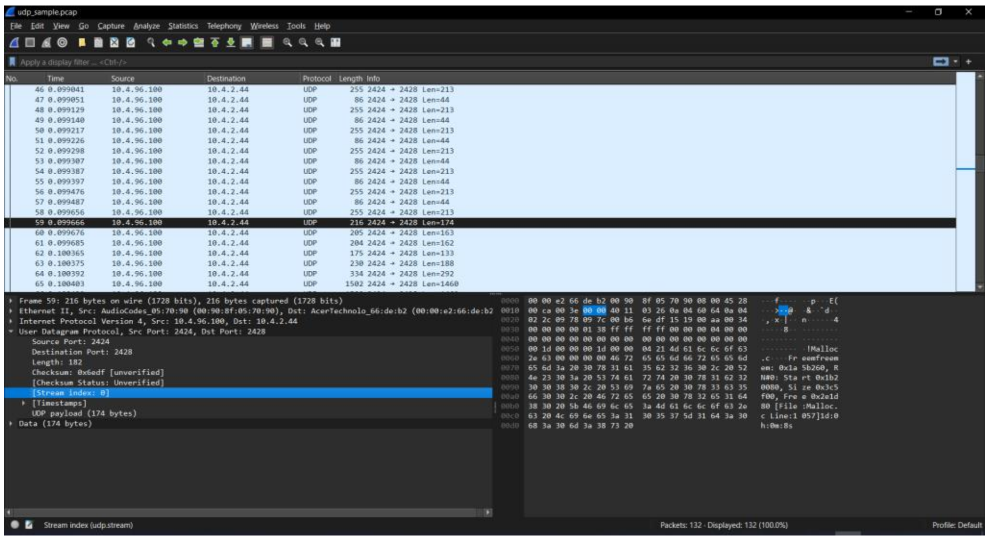
Fig. 1.

Segment’s time to live value is 64.

The Time to Live (TTL) value is a field in the header of Internet Protocol (IP) packets. TTL represents the maximum number of hops that a packet can take in the network before it is discarded.

If the TTL value reaches zero before the TCP or UDP segment arrives at its destination, the router that decrements the TTL to zero will discard the packet.

Question 02



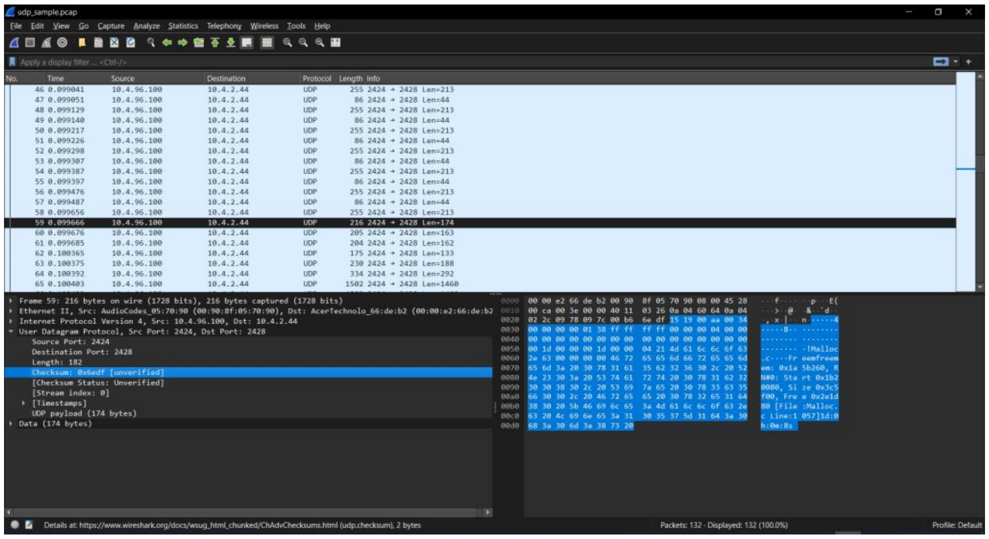
```
udp_sample.pcap
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Wireless  Tools  Help
Apply a display filter ... <Ctrl-/>

No.   Time      Source        Destination  Protocol Length Info
   46 0.099041  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   47 0.099051  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   48 0.099129  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   49 0.099140  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   50 0.099217  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   51 0.099226  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   52 0.099298  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   53 0.099307  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   54 0.099387  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   55 0.099397  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   56 0.099476  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   57 0.099487  10.4.96.100   10.4.2.44    UDP        86 2424 → 2428 Len=44
   58 0.099656  10.4.96.100   10.4.2.44    UDP       255 2424 → 2428 Len=213
   59 0.099666  10.4.96.100   10.4.2.44    UDP       216 2424 → 2428 Len=174
   60 0.099676  10.4.96.100   10.4.2.44    UDP       205 2424 → 2428 Len=163
   61 0.099685  10.4.96.100   10.4.2.44    UDP       204 2424 → 2428 Len=162
   62 0.100365  10.4.96.100   10.4.2.44    UDP       175 2424 → 2428 Len=133
   63 0.100375  10.4.96.100   10.4.2.44    UDP       230 2424 → 2428 Len=188
   64 0.100392  10.4.96.100   10.4.2.44    UDP       334 2424 → 2428 Len=292
   65 0.100403  10.4.96.100   10.4.2.44    UDP      1502 2424 → 2428 Len=1460

› Frame 59: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits)
› Ethernet II, Src: AudioCodes_05:70:90 (00:90:8f:05:70:90), Dst: AcerTechnolo_66:de:b2 (00:00:e2:66:de:b2
› Internet Protocol Version 4, Src: 10.4.96.100, Dst: 10.4.2.44
⌄ User Datagram Protocol, Src Port: 2424, Dst Port: 2428
    Source Port: 2424
    Destination Port: 2428
    Length: 182
    Checksum: 0x6edf [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  › [Timestamps]
    UDP payload (174 bytes)
› Data (174 bytes)

0000  00 00 e2 66 de b2 00 90  8f 05 70 90 08 00 45 28   ···f···· ··p···E(
0010  00 ca 00 3e 00 00 40 11  03 26 0a 04 60 64 0a 04   ···>··@· ·&··`d··
0020  02 2c 09 78 09 7c 00 b6  6e df 15 19 00 aa 00 34   ·,·x·|·· n······4
0030  00 00 00 00 01 38 ff ff  ff ff 00 00 00 04 00 00   ·····8·· ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 1d 00 00 00 1d 00 00  04 21 4d 61 6c 6c 6f 63   ········ ·!Malloc
0060  2e 63 00 00 00 00 46 72  65 65 6d 66 72 65 65 6d   .c····Fr eemfreem
0070  65 6d 3a 20 30 78 31 61  35 62 32 36 30 2c 20 52   em: 0x1a 5b260, R
0080  4e 23 30 3a 20 53 74 61  72 74 20 30 78 31 62 32   N#0: Sta rt 0x1b2
0090  30 30 38 30 2c 20 53 69  7a 65 20 30 78 33 63 35   0080, Si ze 0x3c5
00a0  66 30 30 2c 20 46 72 65  65 20 30 78 32 65 31 64   f00, Fre e 0x2e1d
00b0  38 30 20 5b 46 69 6c 65  3a 4d 61 6c 6c 6f 63 2e   80 [File :Malloc.
00c0  63 20 4c 69 6e 65 3a 31  30 35 37 5d 31 64 3a 30   c Line:1 057]1d:0
00d0  68 3a 30 6d 3a 38 73 20                            h:0m:8s

Stream index (udp.stream)          Packets: 132 - Displayed: 132 (100.0%)          Profile: Default
```

Fig. 2.

The stream index of the UDP segment signifies the numbering of conversations of associated TCP and UDP socket pairs with TCP and UDP being calculated separately. The initial conversation is assigned with the stream index 0, followed by index 1 for the second conversation, and so forth.

The stream index of my segment is 0.

Question 03



udp_sample.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time      Source       Destination  Protocol Length Info
  46 0.099041  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  47 0.099051  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  48 0.099129  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  49 0.099140  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  50 0.099217  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  51 0.099226  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  52 0.099298  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  53 0.099307  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  54 0.099387  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  55 0.099397  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  56 0.099476  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  57 0.099487  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  58 0.099656  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  59 0.099666  10.4.96.100  10.4.2.44    UDP       216 2424 → 2428 Len=174
  60 0.099676  10.4.96.100  10.4.2.44    UDP       205 2424 → 2428 Len=163
  61 0.099685  10.4.96.100  10.4.2.44    UDP       204 2424 → 2428 Len=162
  62 0.100365  10.4.96.100  10.4.2.44    UDP       175 2424 → 2428 Len=133
  63 0.100375  10.4.96.100  10.4.2.44    UDP       230 2424 → 2428 Len=188
  64 0.100392  10.4.96.100  10.4.2.44    UDP       334 2424 → 2428 Len=292
  65 0.100403  10.4.96.100  10.4.2.44    UDP      1502 2424 → 2428 Len=1460
```

```
› Frame 59: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits)
› Ethernet II, Src: AudioCodes_05:70:90 (00:90:8f:05:70:90), Dst: AcerTechnolo_66:de:b2 (00:00:e2:66:de:b2
› Internet Protocol Version 4, Src: 10.4.96.100, Dst: 10.4.2.44
˅ User Datagram Protocol, Src Port: 2424, Dst Port: 2428
    Source Port: 2424
    Destination Port: 2428
    Length: 182
    Checksum: 0x6edf [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  › [Timestamps]
    UDP payload (174 bytes)
› Data (174 bytes)
```

```
0000  00 00 e2 66 de b2 00 90  8f 05 70 90 08 00 45 28   ···f···· ··p···E(
0010  00 ca 00 3e 00 00 40 11  03 26 0a 04 60 64 0a 04   ···>··@· ·&··`d··
0020  02 2c 09 78 09 7c 00 b6  6e df 15 19 00 aa 00 34   ·,·x·|·· n······4
0030  00 00 00 00 01 38 ff ff  ff ff 00 00 00 04 00 00   ·····8·· ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 1d 00 00 00 1d 00 00  04 21 4d 61 6c 6c 6f 63   ········ ·!Malloc
0060  2e 63 00 00 00 00 46 72  65 65 6d 66 72 65 65 6d   .c····Fr eemfreem
0070  65 6d 3a 20 30 78 31 61  35 62 32 36 30 2c 20 52   em: 0x1a 5b260, R
0080  4e 23 30 3a 20 53 74 61  72 74 20 30 78 31 62 32   N#0: Sta rt 0x1b2
0090  30 30 38 30 2c 20 53 69  7a 65 20 30 78 33 63 35   0080, Si ze 0x3c5
00a0  66 30 30 2c 20 46 72 65  65 20 30 78 32 65 31 64   f00, Fre e 0x2e1d
00b0  38 30 20 5b 46 69 6c 65  3a 4d 61 6c 6c 6f 63 2e   80 [File :Malloc.
00c0  63 20 4c 69 6e 65 3a 31  30 35 37 5d 31 64 3a 30   c Line:1 057]1d:0
00d0  68 3a 30 6d 3a 38 73 20                            h:0m:8s
```

Details at: https://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html (udp.checksum), 2 bytes Packets: 132 · Displayed: 132 (100.0%) Profile: Default

Fig. 3.

Checksum value of my segment is 0x6edf (28383).

If the checksum value observed at the receiver's end is different from the transmitted checksum, it suggests that the data may have been corrupted during transmission.

In TCP (Transmission Control Protocol), a similar checksum mechanism is used for error detection. However, there are some differences between UDP and TCP in terms of error handling.

TCP is a reliable, connection-oriented protocol, whereas UDP is connectionless and does not guarantee reliable delivery. If errors are detected using mechanism like checksum, TCP will attempt to recover by retransmitting the data and receiver end discards the corrupted packets, whereas, UDP does not follows the same manner. UDP itself does not specify that the receiver must discard packets with a faulty checksum. It's up to the application using UDP to decide how to handle such situations.

Question 04

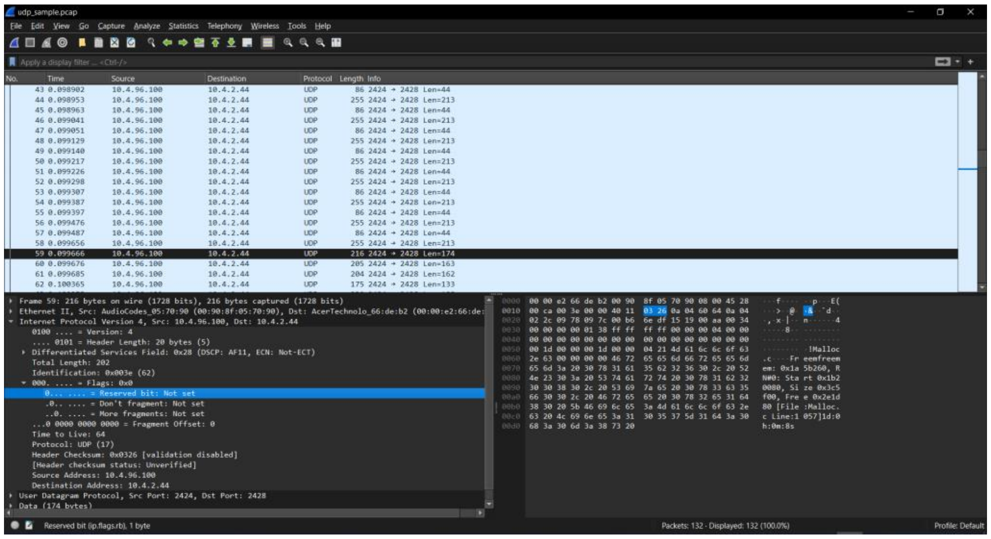

```
udp_sample.pcap
No.  Time       Source       Destination  Protocol Length Info
  43 0.098902   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  44 0.098953   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  45 0.098963   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  46 0.099041   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  47 0.099051   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  48 0.099129   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  49 0.099140   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  50 0.099217   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  51 0.099226   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  52 0.099298   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  53 0.099307   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  54 0.099387   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  55 0.099397   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  56 0.099476   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  57 0.099487   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  58 0.099656   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  59 0.099666   10.4.96.100  10.4.2.44    UDP      216 2424 → 2428 Len=174
  60 0.099676   10.4.96.100  10.4.2.44    UDP      205 2424 → 2428 Len=163
  61 0.099685   10.4.96.100  10.4.2.44    UDP      204 2424 → 2428 Len=162
  62 0.100365   10.4.96.100  10.4.2.44    UDP      175 2424 → 2428 Len=133

Frame 59: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits)
Ethernet II, Src: AudioCodes_05:70:90 (00:90:8f:05:70:90), Dst: AcerTechnolo_66:de:b2 (00:00:e2:66:de:
Internet Protocol Version 4, Src: 10.4.96.100, Dst: 10.4.2.44
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x28 (DSCP: AF11, ECN: Not-ECT)
    Total Length: 202
    Identification: 0x003e (62)
    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: 64
    Protocol: UDP (17)
    Header Checksum: 0x0326 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.4.96.100
    Destination Address: 10.4.2.44
User Datagram Protocol, Src Port: 2424, Dst Port: 2428
Data (174 bytes)

0000  00 00 e2 66 de b2 00 90  8f 05 70 90 08 00 45 28   ···f····  ··p···E(
0010  00 ca 00 3e 00 00 40 11  03 26 0a 04 60 64 0a 04   ···>··@·  ·&··`d··
0020  02 2c 09 78 09 7c 00 b6  6e df 15 19 00 aa 00 34   ·,·x·|··  n······4
0030  00 00 00 00 01 38 ff ff  ff ff 00 00 00 04 00 00   ·····8··  ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0050  00 1d 00 00 00 1d 00 00  04 21 4d 61 6c 6c 6f 63   ········  ·!Malloc
0060  2e 63 00 00 00 00 46 72  65 65 6d 66 72 65 65 6d   .c····Fr  eemfreem
0070  65 6d 3a 20 30 78 31 61  35 62 32 36 30 2c 20 52   em: 0x1a  5b260, R
0080  4e 23 30 3a 20 53 74 61  72 74 20 30 78 31 62 32   N#0: Sta  rt 0x1b2
0090  30 30 38 30 2c 20 53 69  7a 65 20 30 78 33 63 35   0080, Si  ze 0x3c5
00a0  66 30 30 2c 20 46 72 65  65 20 30 78 32 65 31 64   f00, Fre  e 0x2e1d
00b0  38 30 20 5b 46 69 6c 65  3a 4d 61 6c 6c 6f 63 2e   80 [File  :Malloc.
00c0  63 20 4c 69 6e 65 3a 31  30 35 37 5d 31 64 3a 30   c Line:1  057]1d:0
00d0  68 3a 30 6d 3a 38 73 20                            h:0m:8s

Reserved bit (ip.flags.rb), 1 byte        Packets: 132 · Displayed: 132 (100.0%)        Profile: Default
```

Fig. 4.

There exist 3 flags under the IP v4 header listed as reserved bit, don't fragment bit, more fragments bit.

Reserved bit (bit 0) is the least significant bit of the Flags field.

The Reserved bit doesn't have a specific function in the current IPv4 specification and is reserved for potential future use. It should always be set to 0 and ignored by devices that process IPv4 packets.

Question 05

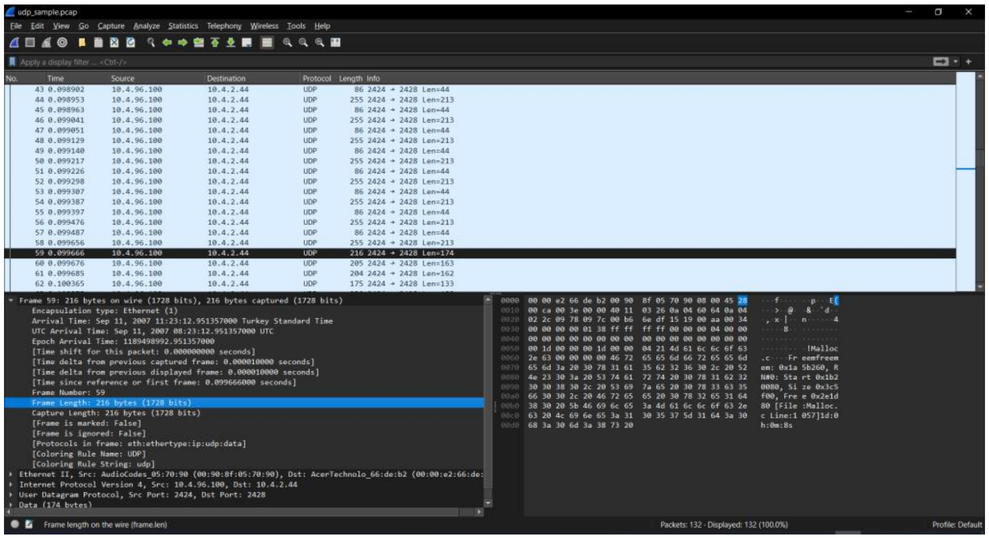
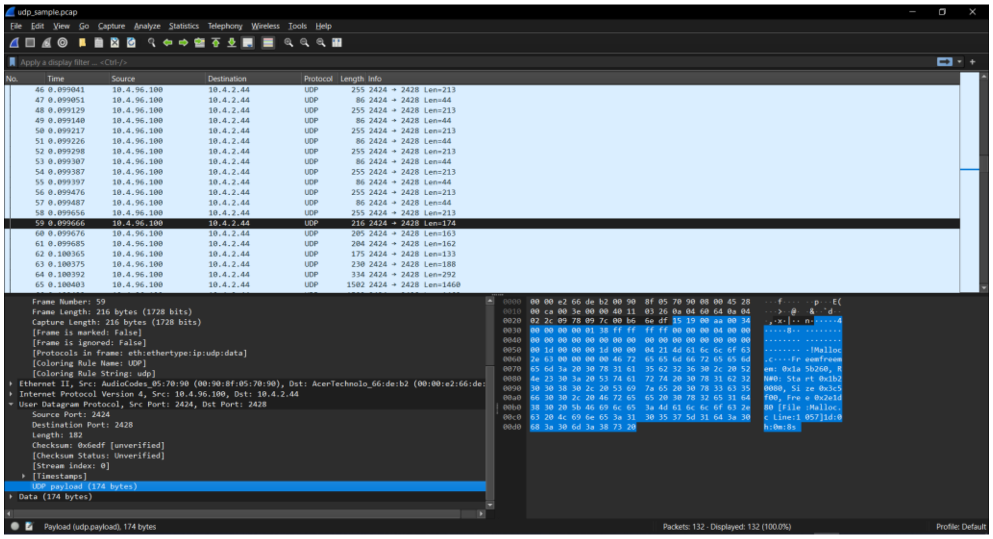

```
udp_sample.pcap
No.  Time       Source       Destination  Protocol Length Info
  43 0.098902   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  44 0.098953   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  45 0.098963   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  46 0.099041   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  47 0.099051   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  48 0.099129   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  49 0.099140   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  50 0.099217   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  51 0.099226   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  52 0.099298   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  53 0.099307   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  54 0.099387   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  55 0.099397   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  56 0.099476   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  57 0.099487   10.4.96.100  10.4.2.44    UDP       86 2424 → 2428 Len=44
  58 0.099656   10.4.96.100  10.4.2.44    UDP      255 2424 → 2428 Len=213
  59 0.099666   10.4.96.100  10.4.2.44    UDP      216 2424 → 2428 Len=174
  60 0.099676   10.4.96.100  10.4.2.44    UDP      205 2424 → 2428 Len=163
  61 0.099685   10.4.96.100  10.4.2.44    UDP      204 2424 → 2428 Len=162
  62 0.100365   10.4.96.100  10.4.2.44    UDP      175 2424 → 2428 Len=133

Frame 59: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 11, 2007 11:23:12.951357000 Turkey Standard Time
    UTC Arrival Time: Sep 11, 2007 08:23:12.951357000 UTC
    Epoch Arrival Time: 1189498992.951357000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000010000 seconds]
    [Time delta from previous displayed frame: 0.000010000 seconds]
    [Time since reference or first frame: 0.099666000 seconds]
    Frame Number: 59
    Frame Length: 216 bytes (1728 bits)
    Capture Length: 216 bytes (1728 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:data]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
Ethernet II, Src: AudioCodes_05:70:90 (00:90:8f:05:70:90), Dst: AcerTechnolo_66:de:b2 (00:00:e2:66:de:
Internet Protocol Version 4, Src: 10.4.96.100, Dst: 10.4.2.44
User Datagram Protocol, Src Port: 2424, Dst Port: 2428
Data (174 bytes)

0000  00 00 e2 66 de b2 00 90  8f 05 70 90 08 00 45 28   ···f····  ··p···E(
0010  00 ca 00 3e 00 00 40 11  03 26 0a 04 60 64 0a 04   ···>··@·  ·&··`d··
0020  02 2c 09 78 09 7c 00 b6  6e df 15 19 00 aa 00 34   ·,·x·|··  n······4
0030  00 00 00 00 01 38 ff ff  ff ff 00 00 00 04 00 00   ·····8··  ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0050  00 1d 00 00 00 1d 00 00  04 21 4d 61 6c 6c 6f 63   ········  ·!Malloc
0060  2e 63 00 00 00 00 46 72  65 65 6d 66 72 65 65 6d   .c····Fr  eemfreem
0070  65 6d 3a 20 30 78 31 61  35 62 32 36 30 2c 20 52   em: 0x1a  5b260, R
0080  4e 23 30 3a 20 53 74 61  72 74 20 30 78 31 62 32   N#0: Sta  rt 0x1b2
0090  30 30 38 30 2c 20 53 69  7a 65 20 30 78 33 63 35   0080, Si  ze 0x3c5
00a0  66 30 30 2c 20 46 72 65  65 20 30 78 32 65 31 64   f00, Fre  e 0x2e1d
00b0  38 30 20 5b 46 69 6c 65  3a 4d 61 6c 6c 6f 63 2e   80 [File  :Malloc.
00c0  63 20 4c 69 6e 65 3a 31  30 35 37 5d 31 64 3a 30   c Line:1  057]1d:0
00d0  68 3a 30 6d 3a 38 73 20                            h:0m:8s

Frame length on the wire (frame.len)        Packets: 132 · Displayed: 132 (100.0%)        Profile: Default
```

Fig. 5.



```
udp_sample.pcap
Apply a display filter ... <Ctrl-/>
No.  Time      Source       Destination  Protocol Length Info
  46 0.099041  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  47 0.099051  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  48 0.099129  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  49 0.099140  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  50 0.099217  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  51 0.099226  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  52 0.099298  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  53 0.099307  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  54 0.099387  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  55 0.099397  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  56 0.099476  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  57 0.099487  10.4.96.100  10.4.2.44    UDP        86 2424 → 2428 Len=44
  58 0.099656  10.4.96.100  10.4.2.44    UDP       255 2424 → 2428 Len=213
  59 0.099666  10.4.96.100  10.4.2.44    UDP       216 2424 → 2428 Len=174
  60 0.099676  10.4.96.100  10.4.2.44    UDP       205 2424 → 2428 Len=163
  61 0.099685  10.4.96.100  10.4.2.44    UDP       204 2424 → 2428 Len=162
  62 0.100365  10.4.96.100  10.4.2.44    UDP       175 2424 → 2428 Len=133
  63 0.100375  10.4.96.100  10.4.2.44    UDP       230 2424 → 2428 Len=188
  64 0.100392  10.4.96.100  10.4.2.44    UDP       334 2424 → 2428 Len=292
  65 0.100403  10.4.96.100  10.4.2.44    UDP      1502 2424 → 2428 Len=1460

    Frame Number: 59
    Frame Length: 216 bytes (1728 bits)
    Capture Length: 216 bytes (1728 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:data]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
  Ethernet II, Src: AudioCodes_05:70:90 (00:90:8f:05:70:90), Dst: AcerTechnolo_66:de:b2 (00:00:e2:66:de:
  Internet Protocol Version 4, Src: 10.4.96.100, Dst: 10.4.2.44
  User Datagram Protocol, Src Port: 2424, Dst Port: 2428
    Source Port: 2424
    Destination Port: 2428
    Length: 182
    Checksum: 0x6edf [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Timestamps]
    UDP payload (174 bytes)
  Data (174 bytes)

0000  00 00 e2 66 de b2 00 90  8f 05 70 90 08 00 45 28   ···f····  ··p···E(
0010  00 ca 00 3e 00 00 40 11  03 26 0a 04 60 64 0a 04   ···>··@·  ·&··`d··
0020  02 2c 09 78 09 7c 00 b6  6e df 15 19 00 aa 00 34   ·,·x·|··  n······4
0030  00 00 00 00 01 38 ff ff  ff ff 00 00 00 04 00 00   ·····8··  ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0050  00 1d 00 00 00 1d 00 00  04 21 4d 61 6c 6c 6f 63   ········  ·!Malloc
0060  2e 63 00 00 00 00 46 72  65 65 6d 66 72 65 65 6d   .c····Fr  eemfreem
0070  65 6d 3a 20 30 78 31 61  35 62 32 36 30 2c 20 52   em: 0x1a  5b260, R
0080  4e 23 30 3a 20 53 74 61  72 74 20 30 78 31 62 32   N#0: Sta  rt 0x1b2
0090  30 30 38 30 2c 20 53 69  7a 65 20 30 78 33 63 35   0080, Si  ze 0x3c5
00a0  66 30 30 2c 20 46 72 65  65 20 30 78 32 65 31 64   f00, Fre  e 0x2e1d
00b0  38 30 20 5b 46 69 6c 65  3a 4d 61 6c 6c 6f 63 2e   80 [File  :Malloc.
00c0  63 20 4c 69 6e 65 3a 31  30 35 37 5d 31 64 3a 30   c Line:1  057]1d:0
00d0  68 3a 30 6d 3a 38 73 20                            h:0m:8s

Payload (udp.payload), 174 bytes          Packets: 132 · Displayed: 132 (100.0%)          Profile: Default
```

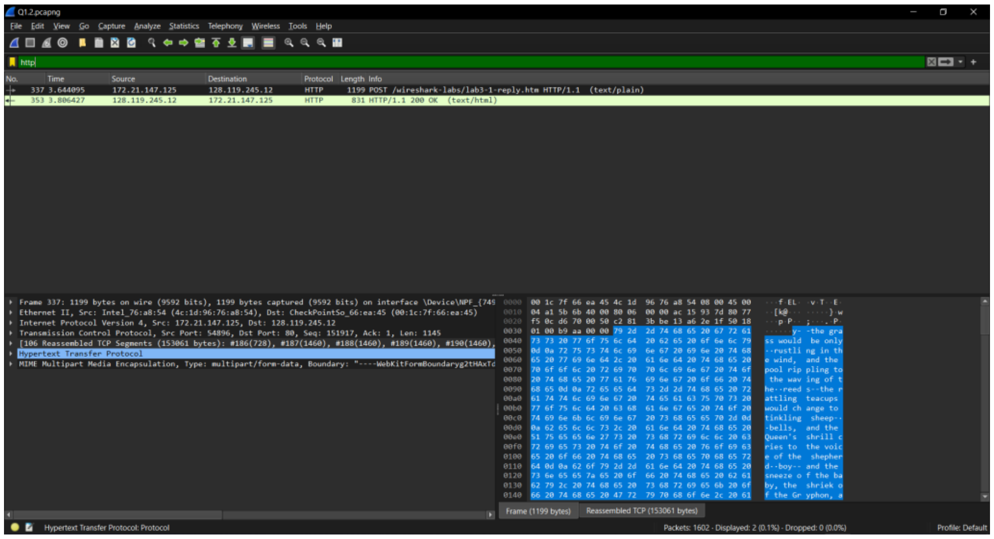
Fig. 6.

The length field refers to the total length of the UDP packet, including the header and data.

Length of the packet is 216 bytes and 174 bytes of it being the data payload.

Part 1.2

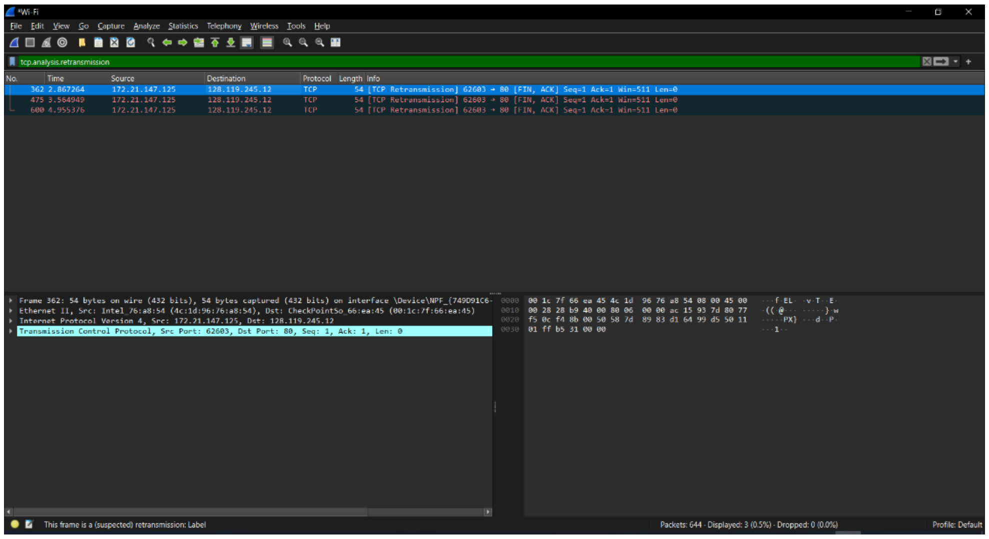
Question 01



```
Q1.2.pcapng
http
No.  Time      Source          Destination     Protocol Length Info
 337 3.644095  172.21.147.125  128.119.245.12  HTTP      1199 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)
 353 3.806427  128.119.245.12  172.21.147.125  HTTP       831 HTTP/1.1 200 OK  (text/html)

  Frame 337: 1199 bytes on wire (9592 bits), 1199 bytes captured (9592 bits) on interface \Device\NPF_{749
  Ethernet II, Src: Intel_76:a8:54 (4c:1d:96:76:a8:54), Dst: CheckPointSo_66:ea:45 (00:1c:7f:66:ea:45)
  Internet Protocol Version 4, Src: 172.21.147.125, Dst: 128.119.245.12
  Transmission Control Protocol, Src Port: 54896, Dst Port: 80, Seq: 151917, Ack: 1, Len: 1145
  [106 Reassembled TCP Segments (153061 bytes): #186(728), #187(1460), #188(1460), #189(1460), #190(1460),
  Hypertext Transfer Protocol
  MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundaryg2tHAxTd

0000  00 1c 7f 66 ea 45 4c 1d  96 76 a8 54 08 00 45 00   ···f·EL·  ·v·T··E·
0010  04 a1 5b 6b 40 00 80 06  00 00 ac 15 93 7d 80 77   ··[k@···  ·····}·w
0020  f5 0c d6 70 00 50 c2 81  3b be 13 a6 2e 1f 50 18   ···p·P··  ;·····P·
0030  01 00 b9 aa 00 00 79 2d  2d 74 68 65 20 67 72 61   ······y-  -the gra
0040  73 73 20 77 6f 75 6c 64  20 62 65 20 6f 6e 6c 79   ss would   be only
0050  0d 0a 72 75 73 74 6c 69  6e 67 20 69 6e 20 74 68   ··rustli  ng in th
0060  65 20 77 69 6e 64 2c 20  61 6e 64 20 74 68 65 20   e wind,   and the
0070  70 6f 6f 6c 20 72 69 70  70 6c 69 6e 67 20 74 6f   pool rip  pling to
0080  20 74 68 65 20 77 61 76  69 6e 67 20 6f 66 20 74    the wav  ing of t
0090  68 65 0d 0a 72 65 65 64  73 2d 2d 74 68 65 20 72   he··reed  s--the r
00a0  61 74 74 6c 69 6e 67 20  74 65 61 63 75 70 73 20   attling   teacups
00b0  77 6f 75 6c 64 20 63 68  61 6e 67 65 20 74 6f 20   would ch  ange to
00c0  74 69 6e 6b 6c 69 6e 67  20 73 68 65 65 70 2d 0d   tinkling   sheep-·
00d0  0a 62 65 6c 6c 73 2c 20  61 6e 64 20 74 68 65 20   ·bells,   and the
00e0  51 75 65 65 6e 27 73 20  73 68 72 69 6c 6c 20 63   Queen's   shrill c
00f0  72 69 65 73 20 74 6f 20  74 68 65 20 76 6f 69 63   ries to   the voic
0100  65 20 6f 66 20 74 68 65  20 73 68 65 70 68 65 72   e of the   shepher
0110  64 0d 0a 62 6f 79 2d 2d  61 6e 64 20 74 68 65 20   d··boy--  and the
0120  73 6e 65 65 7a 65 20 6f  66 20 74 68 65 20 62 61   sneeze o  f the ba
0130  62 79 2c 20 74 68 65 20  73 68 72 69 65 6b 20 6f   by, the   shriek o
0140  66 20 74 68 65 20 47 72  79 70 68 6f 6e 2c 20 61   f the Gr  yphon, a

Frame (1199 bytes)   Reassembled TCP (153061 bytes)
Hypertext Transfer Protocol: Protocol     Packets: 1602 · Displayed: 2 (0.1%) · Dropped: 0 (0.0%)     Profile: Default
```

Fig. 7.

While capturing the packets with WireShark upon uploading the Alice in Wonderland file through a HTTP POST request, filtering the packet captures with the HTTP selection enables the observation of both the source IP address and destination IP address.



```
tcp.analysis.retransmission

No.  Time      Source          Destination     Protocol Length Info
362  2.867264  172.21.147.125  128.119.245.12  TCP      54 [TCP Retransmission] 62603 → 80 [FIN, ACK] Seq=1 Ack=1 Win=511 Len=0
475  3.564949  172.21.147.125  128.119.245.12  TCP      54 [TCP Retransmission] 62603 → 80 [FIN, ACK] Seq=1 Ack=1 Win=511 Len=0
600  4.955376  172.21.147.125  128.119.245.12  TCP      54 [TCP Retransmission] 62603 → 80 [FIN, ACK] Seq=1 Ack=1 Win=511 Len=0

Frame 362: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{749D91C6-
Ethernet II, Src: Intel_76:a8:54 (4c:1d:96:76:a8:54), Dst: CheckPointSo_66:ea:45 (00:1c:7f:66:ea:45)
Internet Protocol Version 4, Src: 172.21.147.125, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 62603, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

0000  00 1c 7f 66 ea 45 4c 1d  96 76 a8 54 08 00 45 00
0010  00 28 28 b9 40 00 80 06  00 00 ac 15 93 7d 80 77
0020  f5 0c f4 8b 00 50 58 7d  89 83 d1 64 99 d5 50 11
0030  01 ff b5 31 00 00

This frame is a (suspected) retransmission: Label
Packets: 644 · Displayed: 3 (0.5%) · Dropped: 0 (0.0%)    Profile: Default
```

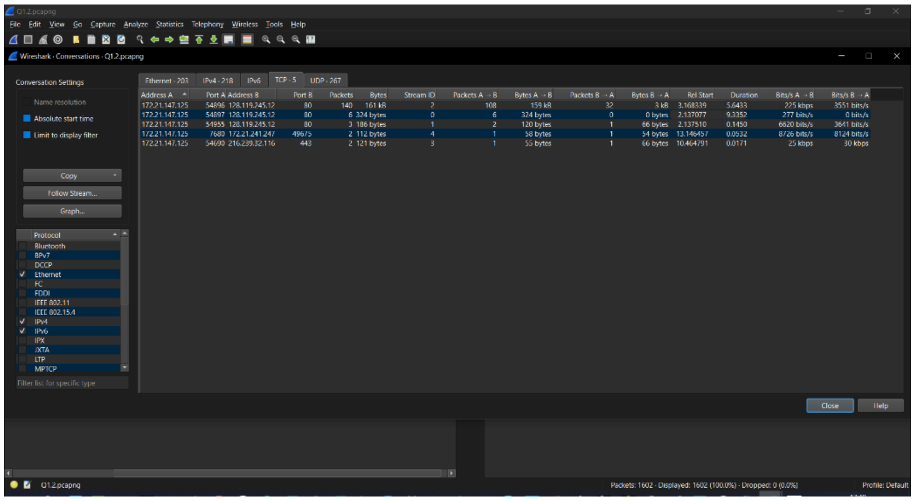
Fig. 8.

After filtering the packet captures to isolate TCP retransmissions, examination of the source and destination addresses allows us to confirm that the TCP transmissions are belong to the HTTP POST request.

Retransmission occurs in communication networks when data packets are lost, corrupted, or not received within a specified time frame.

Question 02

Step 01



Wireshark · Conversations · Q1.2.pcapng

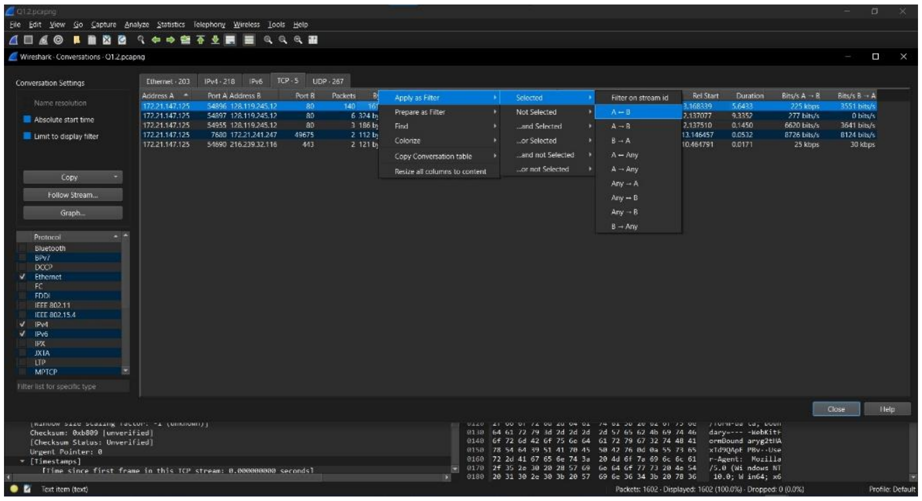
Ethernet · 203 | IPv4 · 218 | IPv6 | TCP · 5 | UDP · 267

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
172.21.147.125	54896	128.119.245.12	80	140	161 kB	2	108	159 kB	32	3 kB	3.168339	5.6433	225 kbps	3551 bits/s
172.21.147.125	54897	128.119.245.12	80	6	324 bytes	0	6	324 bytes	0	0 bytes	2.137077	9.3352	277 bits/s	0 bits/s
172.21.147.125	54955	128.119.245.12	80	3	186 bytes	1	2	120 bytes	1	66 bytes	2.137510	0.1450	6620 bits/s	3641 bits/s
172.21.147.125	7680	172.21.241.247	49675	2	112 bytes	4	1	58 bytes	1	54 bytes	13.146457	0.0532	8726 bits/s	8124 bits/s
172.21.147.125	54690	216.239.32.116	443	2	121 bytes	3	1	55 bytes	1	66 bytes	10.464791	0.0171	25 kbps	30 kbps

Packets: 1602 · Displayed: 1602 (100.0%) · Dropped: 0 (0.0%)   Profile: Default

Fig. 9.

Step 02



Address A	Port A	Address B	Port B	Packets	Rel Start	Duration	Bits/s A → B	Bits/s B → A
172.21.147.125	54896	128.119.245.12	80	140	3.168339	5.6433	225 kbps	3551 bits/s
172.21.147.125	54897	128.119.245.12	80	6	2.137077	9.3352	277 bits/s	0 bits/s
172.21.147.125	54955	128.119.245.12	80	3	2.137510	0.1450	6620 bits/s	3641 bits/s
172.21.147.125	7680	172.21.241.247	49675	2	13.146457	0.0532	8726 bits/s	8124 bits/s
172.21.147.125	54690	216.239.32.116	443	2	10.464791	0.0171	25 kbps	30 kbps

Fig. 10.

Step 03

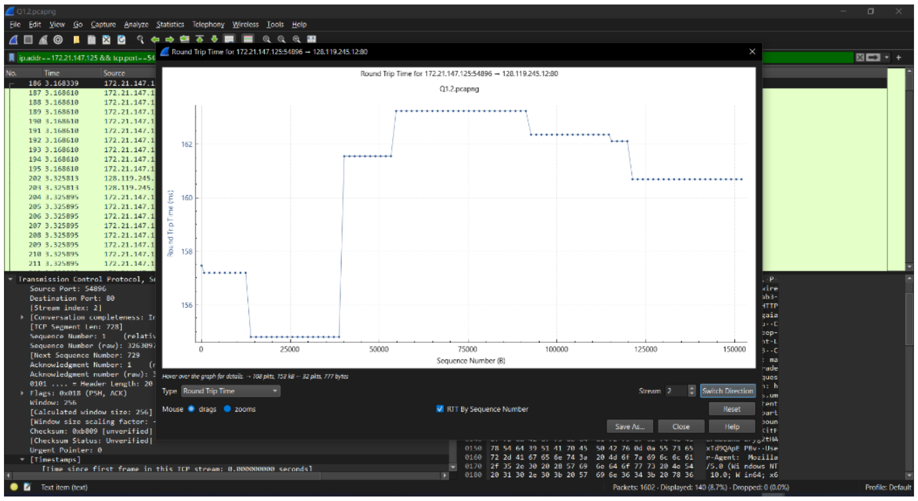


Fig. 11.

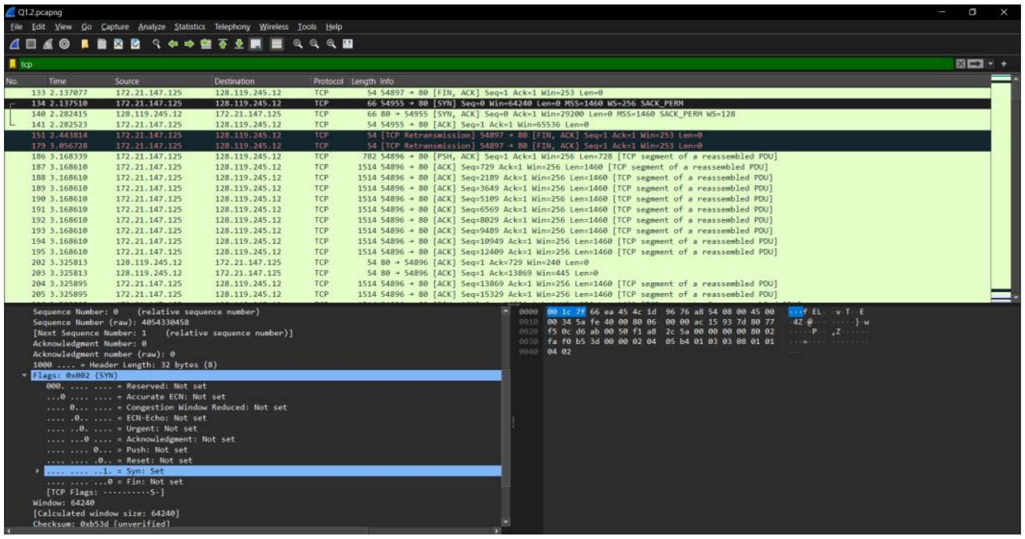
Values of RTT Time through the course of TCP communication of a specific server and a client as can be observed from the graph is between 154-164 ms.

Round Trip Time (RTT) refers to the total time it takes for a data packet to travel from a source to a destination and back again.

There exist variations of RTT through the course of TCP connection between a specific server and a client which might be due to several reasons including routing changes, queueing delay, etc.

Question 03

The TCP header contains essential information such as sequence numbers, flags, etc. These components play crucial roles in facilitating reliable and ordered communication between devices differentiating from UDP.


Q1.2.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp

```
No.  Time      Source          Destination     Protocol Length Info
 133 2.137077  172.21.147.125  128.119.245.12  TCP        54 54897 → 80 [FIN, ACK] Seq=1 Ack=1 Win=253 Len=0
 134 2.137510  172.21.147.125  128.119.245.12  TCP        66 54955 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 140 2.282415  128.119.245.12  172.21.147.125  TCP        66 80 → 54955 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
 141 2.282523  172.21.147.125  128.119.245.12  TCP        54 54955 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
 151 2.443814  172.21.147.125  128.119.245.12  TCP        54 [TCP Retransmission] 54897 → 80 [FIN, ACK] Seq=1 Ack=1 Win=253 Len=0
 179 3.056728  172.21.147.125  128.119.245.12  TCP        54 [TCP Retransmission] 54897 → 80 [FIN, ACK] Seq=1 Ack=1 Win=253 Len=0
 186 3.168339  172.21.147.125  128.119.245.12  TCP       782 54896 → 80 [PSH, ACK] Seq=1 Ack=1 Win=256 Len=728 [TCP segment of a reassembled PDU]
 187 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=729 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 188 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=2189 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 189 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=3649 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 190 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=5109 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 191 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=6569 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 192 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=8029 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 193 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=9489 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 194 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=10949 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 195 3.168610  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=12409 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 202 3.325813  128.119.245.12  172.21.147.125  TCP        54 80 → 54896 [ACK] Seq=1 Ack=729 Win=240 Len=0
 203 3.325813  128.119.245.12  172.21.147.125  TCP        54 80 → 54896 [ACK] Seq=1 Ack=13869 Win=445 Len=0
 204 3.325895  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=13869 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
 205 3.325895  172.21.147.125  128.119.245.12  TCP      1514 54896 → 80 [ACK] Seq=15329 Ack=1 Win=256 Len=1460 [TCP segment of a reassembled PDU]
```

```
Sequence Number: 0    (relative sequence number)
Sequence Number (raw): 4054330458
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1000 .... = Header Length: 32 bytes (8)
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
  [TCP Flags: ··········S·]
Window: 64240
[Calculated window size: 64240]
Checksum: 0xb53d [unverified]
```

```
0000  00 1c 7f 66 ea 45 4c 1d  96 76 a8 54 08 00 45 00   ···f·EL··v·T··E·
0010  00 34 5a fe 40 00 80 06  00 00 ac 15 93 7d 80 77   ·4Z·@········}·w
0020  f5 0c d6 ab 00 50 f1 a8  2c 5a 00 00 00 00 80 02   ·····P··,Z······
0030  fa f0 b5 3d 00 00 02 04  05 b4 01 03 03 08 01 01   ···=············
0040  04 02                                              ··
```

Fig. 12.

ACK (Acknowledge):

The ACK flag is used to acknowledge the successful receipt of data.

When set, it indicates that the acknowledgment number field is valid, confirming the receipt of the data up to the specified sequence number.

Use in Three-Way Handshake: It is crucial during the three-way handshake process, where the ACK flag is set to acknowledge the receipt of the SYN packet, establishing a connection.

SYN (Synchronize):

The SYN flag is used to initiate a connection between two devices.

When a device wants to establish a TCP connection, it sends a TCP packet with the SYN flag set. This initiates the three-way handshake process to synchronize sequence numbers and establish a reliable connection.

PSH (Push):

The PSH flag is used to request immediate data push to the application layer.

When this flag is set, it signals the receiving end to deliver the data to the receiving application as soon as possible without waiting to accumulate more data. It is often used for real-time or interactive applications.

FIN (Finish):

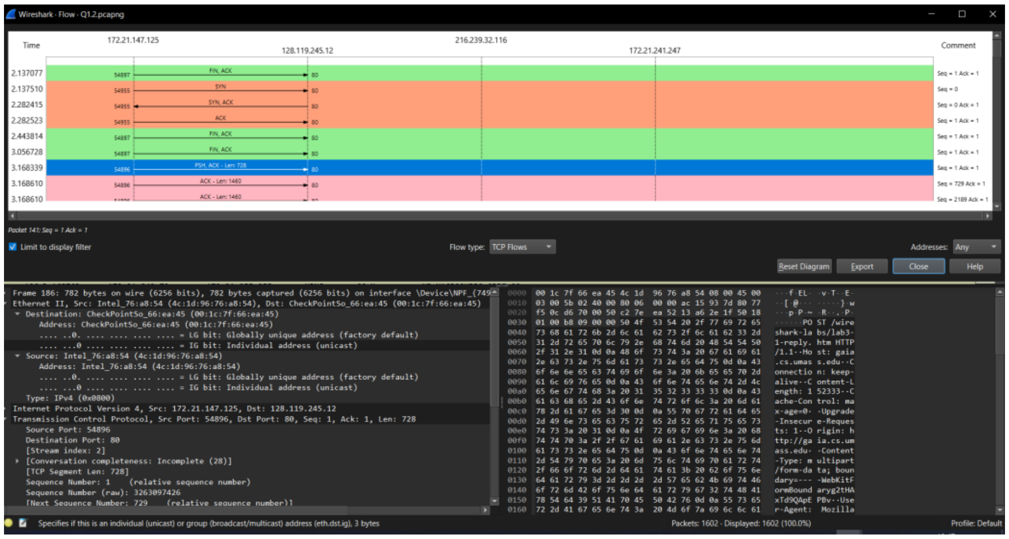
The FIN flag is used to signal the termination of a TCP connection.

When a device wants to end a connection, it sends a TCP packet with the FIN flag set. This initiates the connection termination process, allowing both sides to release resources and gracefully close the connection.

Question 04

TCP: Connection-oriented protocol. Before data transfer begins, a three-way handshake is performed to establish a reliable connection between the sender and receiver. This ensures that both parties are ready to exchange data and that they agree on parameters for the communication session.

UDP: Connectionless protocol. No connection setup is required before sending data. Each packet is sent independently of the others, and there is no guarantee of delivery or order of arrival.



```
2.137077  54897  FIN, ACK            --> 80   Seq = 1 Ack = 1
2.137510  54955  SYN                 --> 80   Seq = 0
2.282415  54955  SYN, ACK            <-- 80   Seq = 0 Ack = 1
2.282523  54955  ACK                 --> 80   Seq = 1 Ack = 1
2.443814  54897  FIN, ACK            --> 80   Seq = 1 Ack = 1
3.056728  54897  FIN, ACK            --> 80   Seq = 1 Ack = 1
3.168339  54896  PSH, ACK - Len: 728 --> 80   Seq = 1 Ack = 1
3.168610  54896  ACK - Len: 1460     --> 80   Seq = 729 Ack = 1
3.168610         ACK - Len: 1460              Seq = 2189 Ack = 1
```

Packet 141: Seq = 1 Ack = 1

```
Frame 186: 782 bytes on wire (6256 bits), 782 bytes captured (6256 bits) on interface \Device\NPF_{749
Ethernet II, Src: Intel_76:a8:54 (4c:1d:96:76:a8:54), Dst: CheckPointSo_66:ea:45 (00:1c:7f:66:ea:45)
  Destination: CheckPointSo_66:ea:45 (00:1c:7f:66:ea:45)
    Address: CheckPointSo_66:ea:45 (00:1c:7f:66:ea:45)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: Intel_76:a8:54 (4c:1d:96:76:a8:54)
    Address: Intel_76:a8:54 (4c:1d:96:76:a8:54)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 172.21.147.125, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 54896, Dst Port: 80, Seq: 1, Ack: 1, Len: 728
  Source Port: 54896
  Destination Port: 80
  [Stream index: 2]
  [Conversation completeness: Incomplete (28)]
  [TCP Segment Len: 728]
  Sequence Number: 1    (relative sequence number)
  Sequence Number (raw): 3263097426
  [Next Sequence Number: 729    (relative sequence number)]
```

```
0000  00 1c 7f 66 ea 45 4c 1d  96 76 a8 54 08 00 45 00   ···f·EL· ·v·T··E·
0010  03 00 5b 02 40 00 80 06  00 00 ac 15 93 7d 80 77   ··[·@··· ·····}·w
0020  f5 0c d6 70 00 50 c2 7e  ea 52 13 a6 2e 1f 50 18   ···p·P·~ ·R···.P·
0030  01 00 b8 09 00 00 50 4f  53 54 20 2f 77 69 72 65   ······PO ST /wire
0040  73 68 61 72 6b 2d 6c 61  62 73 2f 6c 61 62 33 2d   shark-la bs/lab3-
0050  31 2d 72 65 70 6c 79 2e  68 74 6d 20 48 54 54 50   1-reply. htm HTTP
0060  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 67 61 69 61   /1.1··Ho st: gaia
0070  2e 63 73 2e 75 6d 61 73  73 2e 65 64 75 0d 0a 43   .cs.umas s.edu··C
0080  6f 6e 6e 65 63 74 69 6f  6e 3a 20 6b 65 65 70 2d   onnectio n: keep-
0090  61 6c 69 76 65 0d 0a 43  6f 6e 74 65 6e 74 2d 4c   alive··C ontent-L
00a0  65 6e 67 74 68 3a 20 31  35 32 33 33 33 0d 0a 43   ength: 1 52333··C
00b0  61 63 68 65 2d 43 6f 6e  74 72 6f 6c 3a 20 6d 61   ache-Con trol: ma
00c0  78 2d 61 67 65 3d 30 0d  0a 55 70 67 72 61 64 65   x-age=0· ·Upgrade
00d0  2d 49 6e 73 65 63 75 72  65 2d 52 65 71 75 65 73   -Insecur e-Reques
00e0  74 73 3a 20 31 0d 0a 4f  72 69 67 69 6e 3a 20 68   ts: 1··O rigin: h
00f0  74 74 70 3a 2f 2f 67 61  69 61 2e 63 73 2e 75 6d   ttp://ga ia.cs.um
0100  61 73 73 2e 65 64 75 0d  0a 43 6f 6e 74 65 6e 74   ass.edu· ·Content
0110  2d 54 79 70 65 3a 20 6d  75 6c 74 69 70 61 72 74   -Type: m ultipart
0120  2f 66 6f 72 6d 2d 64 61  74 61 3b 20 62 6f 75 6e   /form-da ta; boun
0130  64 61 72 79 3d 2d 2d 2d  2d 57 65 62 4b 69 74 46   dary=--- -WebKitF
0140  6f 72 6d 42 6f 75 6e 64  61 72 79 67 32 74 48 41   ormBound aryg2tHA
0150  78 54 64 39 51 41 70 45  50 42 76 0d 0a 55 73 65   xTd9QApE PBv··Use
0160  72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c 6c 61   r-Agent:  Mozilla
```

Specifies if this is an individual (unicast) or group (broadcast/multicast) address (eth.dst.ig), 3 bytes

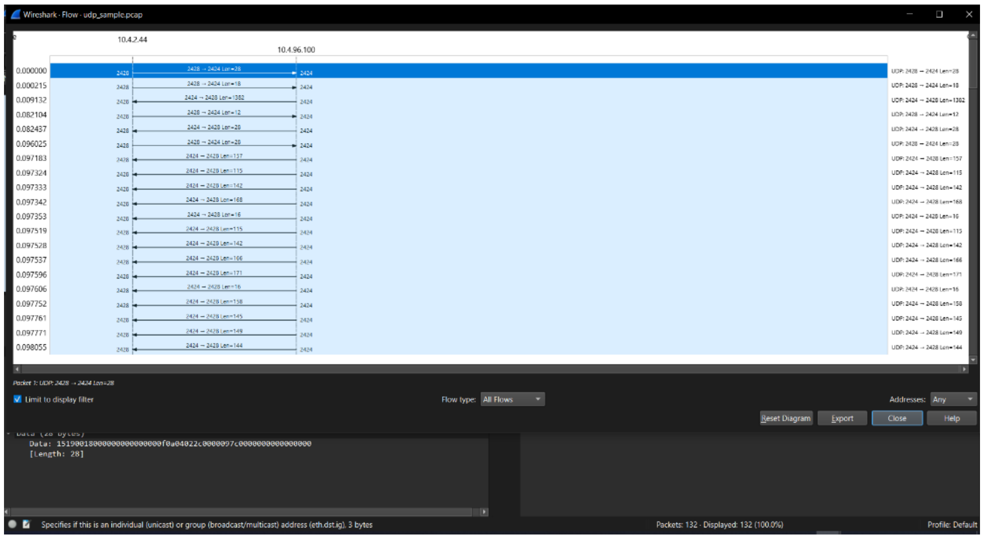
Packets: 1602 · Displayed: 1602 (100.0%)

Profile: Default

Fig. 13.

The TCP PSH (Push) flag functions as a control signal indicating that the recipient device should promptly deliver the data to the receiving application without buffering. Monitoring packets with the push flag enables the tracking of data intended for transfer to the application layer of the receiver.

Above demonstrated the flow diagram of the transferred data with the TCP Protocol. Selected TCP packet carries the data of the HTTP post request. Three-way handshake can be observed in the flow diagram of the TCP protocol.



```
0.000000  UDP: 2428 → 2424 Len=28
0.000215  UDP: 2428 → 2424 Len=18
0.009132  UDP: 2424 → 2428 Len=1382
0.082104  UDP: 2428 → 2424 Len=12
0.082437  UDP: 2424 → 2428 Len=28
0.096025  UDP: 2428 → 2424 Len=28
0.097183  UDP: 2424 → 2428 Len=157
0.097324  UDP: 2424 → 2428 Len=115
0.097333  UDP: 2424 → 2428 Len=142
0.097342  UDP: 2424 → 2428 Len=168
0.097353  UDP: 2424 → 2428 Len=16
0.097519  UDP: 2424 → 2428 Len=115
0.097528  UDP: 2424 → 2428 Len=142
0.097537  UDP: 2424 → 2428 Len=166
0.097596  UDP: 2424 → 2428 Len=171
0.097606  UDP: 2424 → 2428 Len=16
0.097752  UDP: 2424 → 2428 Len=158
0.097761  UDP: 2424 → 2428 Len=145
0.097771  UDP: 2424 → 2428 Len=149
0.098055  UDP: 2424 → 2428 Len=144

Packet 1: UDP: 2428 → 2424 Len=28

    Data: 151900180000000000000000f0a04022c0000097c0000000000000000
    [Length: 28]

Specifies if this is an individual (unicast) or group (broadcast/multicast) address (eth.dst.ig), 3 bytes
Packets: 132 · Displayed: 132 (100.0%)
Profile: Default
```

Fig. 14.

Above demonstrated the flow diagram of the transferred data with the UDP Protocol.

Part 2.1

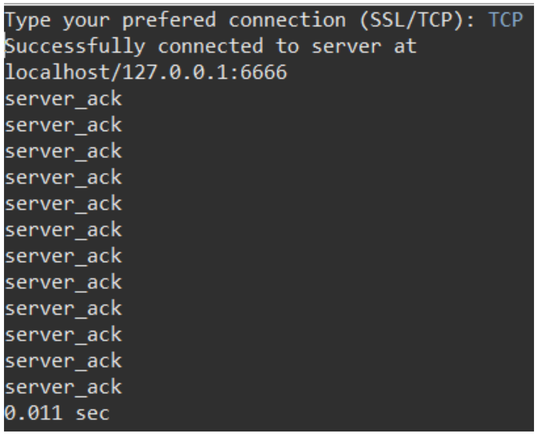
Question 01

Client 01 uses TCP protocol and Client 02 uses both TCP and TLS protocols to communicate with the server.

Question 02

Execution of the Client01, Client02 and Server program snapshots:

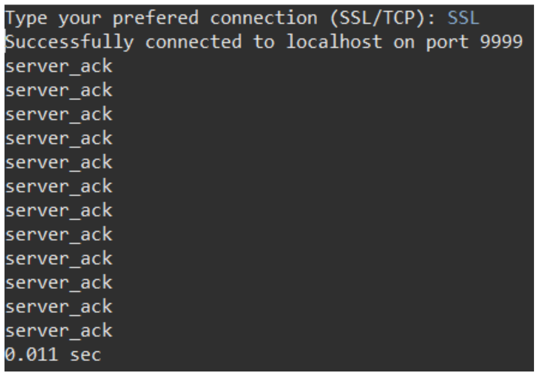
Client01:



```
Type your prefered connection (SSL/TCP): TCP
Successfully connected to server at
localhost/127.0.0.1:6666
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
0.011 sec
```

Fig. 15.

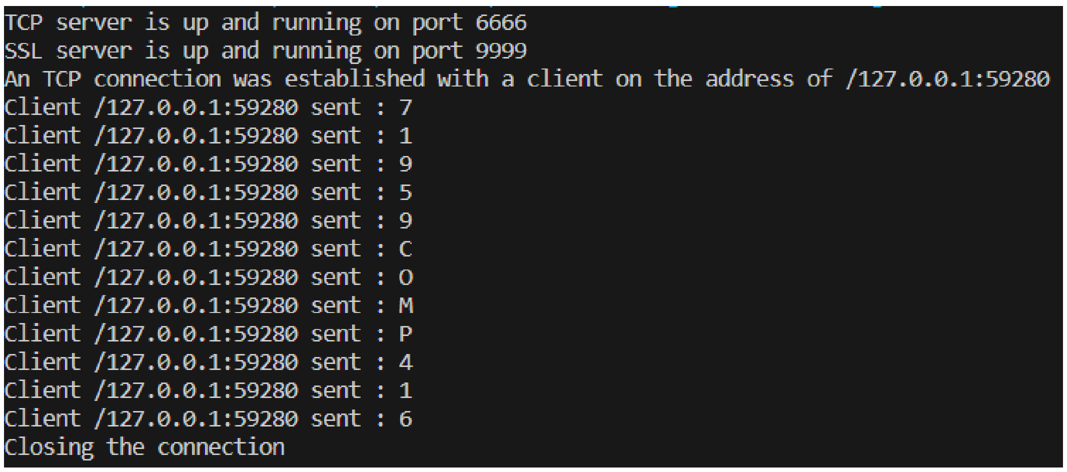
Client02:



```
Type your prefered connection (SSL/TCP): SSL
Successfully connected to localhost on port 9999
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
server_ack
0.011 sec
```

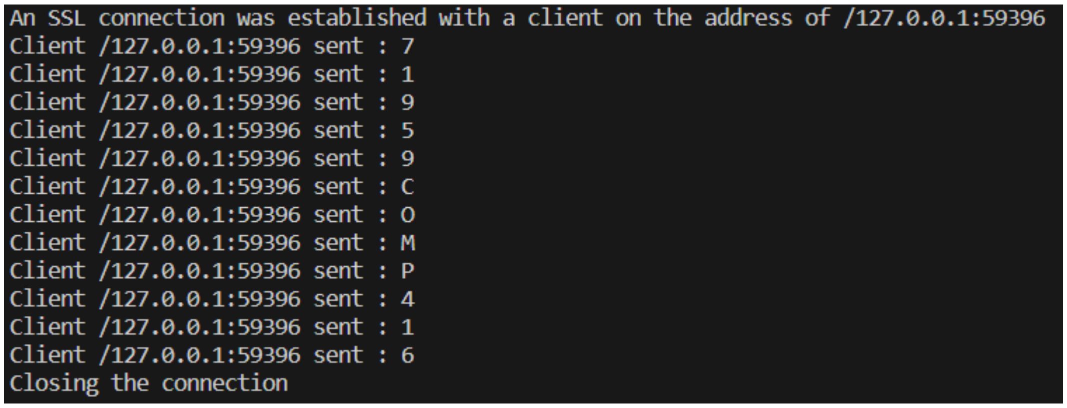
Fig. 16.

Server:



```
TCP server is up and running on port 6666
SSL server is up and running on port 9999
An TCP connection was established with a client on the address of /127.0.0.1:59280
Client /127.0.0.1:59280 sent : 7
Client /127.0.0.1:59280 sent : 1
Client /127.0.0.1:59280 sent : 9
Client /127.0.0.1:59280 sent : 5
Client /127.0.0.1:59280 sent : 9
Client /127.0.0.1:59280 sent : C
Client /127.0.0.1:59280 sent : O
Client /127.0.0.1:59280 sent : M
Client /127.0.0.1:59280 sent : P
Client /127.0.0.1:59280 sent : 4
Client /127.0.0.1:59280 sent : 1
Client /127.0.0.1:59280 sent : 6
Closing the connection
```

Fig. 17.

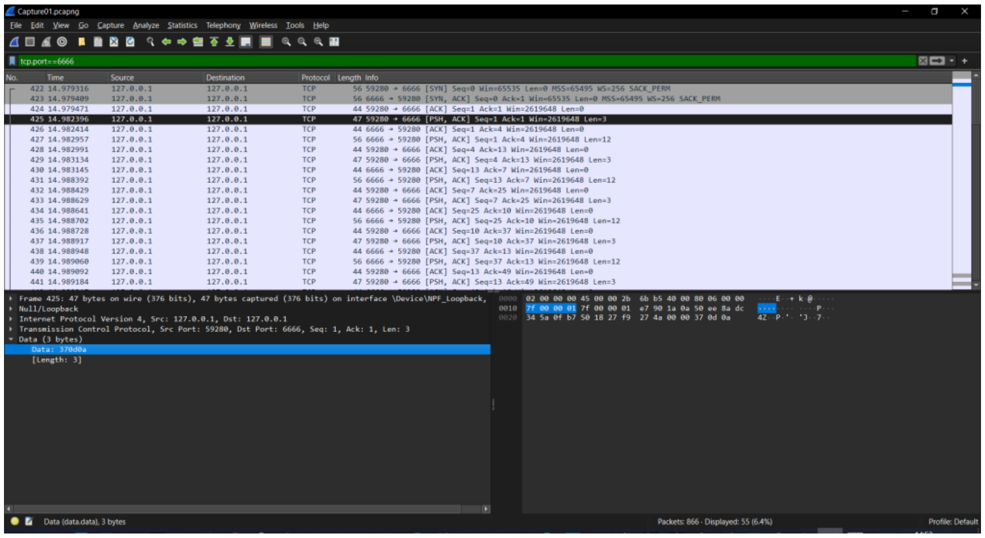


```
An SSL connection was established with a client on the address of /127.0.0.1:59396
Client /127.0.0.1:59396 sent : 7
Client /127.0.0.1:59396 sent : 1
Client /127.0.0.1:59396 sent : 9
Client /127.0.0.1:59396 sent : 5
Client /127.0.0.1:59396 sent : 9
Client /127.0.0.1:59396 sent : C
Client /127.0.0.1:59396 sent : O
Client /127.0.0.1:59396 sent : M
Client /127.0.0.1:59396 sent : P
Client /127.0.0.1:59396 sent : 4
Client /127.0.0.1:59396 sent : 1
Client /127.0.0.1:59396 sent : 6
Closing the connection
```

Fig. 18.

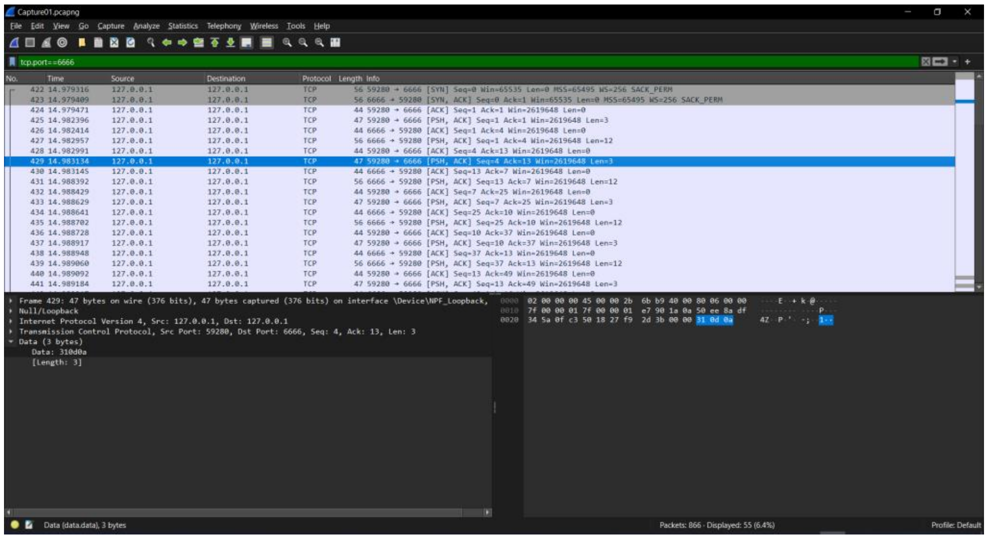
It is possible to find the sent data within the sent packets when mere TCP protocol is used. Whereas, when TLS protocol is used on the top of the TCP protocol, since the sent data are encrypted, it is not possible to observe via Wireshark.

Capture of the Client 01 Communication (Unsecure Communication)



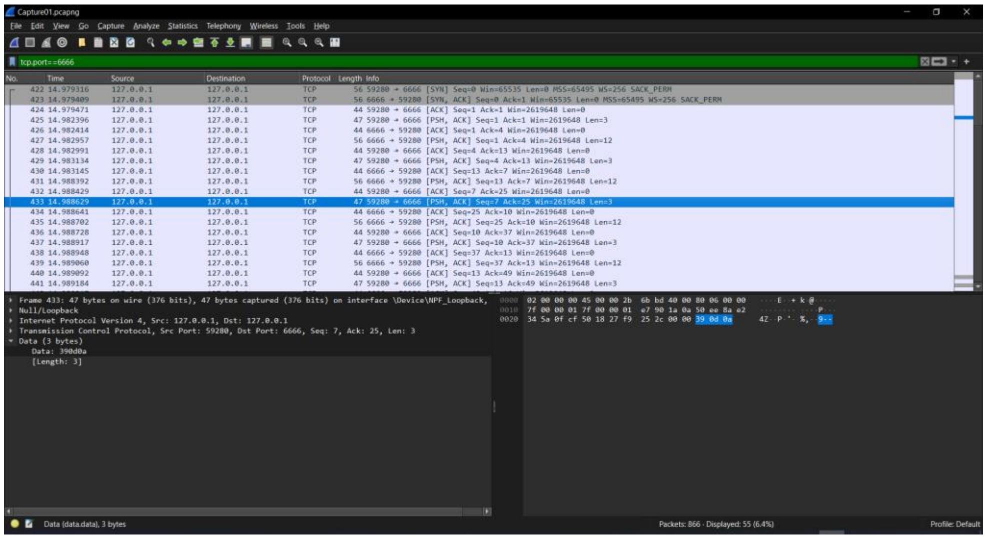
```
Capture01.pcapng
tcp.port==6666
No.  Time       Source     Destination  Protocol Length Info
422 14.979316  127.0.0.1  127.0.0.1    TCP      56 59280 → 6666 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
423 14.979409  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
424 14.979471  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
425 14.982396  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=3
426 14.982414  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=1 Ack=4 Win=2619648 Len=0
427 14.982957  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=1 Ack=4 Win=2619648 Len=12
428 14.982991  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=4 Ack=13 Win=2619648 Len=0
429 14.983134  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=4 Ack=13 Win=2619648 Len=3
430 14.983145  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=13 Ack=7 Win=2619648 Len=0
431 14.988392  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=13 Ack=7 Win=2619648 Len=12
432 14.988429  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=7 Ack=25 Win=2619648 Len=0
433 14.988629  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=7 Ack=25 Win=2619648 Len=3
434 14.988641  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=25 Ack=10 Win=2619648 Len=0
435 14.988702  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=25 Ack=10 Win=2619648 Len=12
436 14.988728  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=10 Ack=37 Win=2619648 Len=0
437 14.988917  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=10 Ack=37 Win=2619648 Len=3
438 14.988948  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=37 Ack=13 Win=2619648 Len=0
439 14.989060  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=37 Ack=13 Win=2619648 Len=12
440 14.989092  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=13 Ack=49 Win=2619648 Len=0
441 14.989184  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=13 Ack=49 Win=2619648 Len=3

> Frame 425: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface \Device\NPF_Loopback,
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 59280, Dst Port: 6666, Seq: 1, Ack: 1, Len: 3
v Data (3 bytes)
    Data: 370d0a
    [Length: 3]

0000  02 00 00 00 45 00 00 2b  6b b5 40 00 80 06 00 00   ····E··+ k·@·····
0010  7f 00 00 01 7f 00 00 01  e7 90 1a 0a 50 ee 8a dc   ········ ····P···
0020  34 5a 0f b7 50 18 27 f9  27 4a 00 00 37 0d 0a      4Z··P·'· 'J··7··

Data (data.data), 3 bytes          Packets: 866 · Displayed: 55 (6.4%)          Profile: Default
```

Fig. 19.



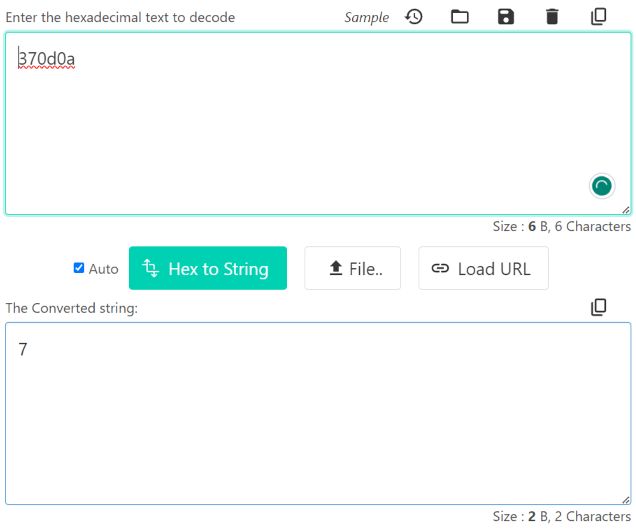
```
Capture01.pcapng
tcp.port==6666
No.  Time       Source     Destination  Protocol Length Info
422 14.979316  127.0.0.1  127.0.0.1    TCP      56 59280 → 6666 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
423 14.979409  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
424 14.979471  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
425 14.982396  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=3
426 14.982414  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=1 Ack=4 Win=2619648 Len=0
427 14.982957  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=1 Ack=4 Win=2619648 Len=12
428 14.982991  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=4 Ack=13 Win=2619648 Len=0
429 14.983134  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=4 Ack=13 Win=2619648 Len=3
430 14.983145  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=13 Ack=7 Win=2619648 Len=0
431 14.988392  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=13 Ack=7 Win=2619648 Len=12
432 14.988429  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=7 Ack=25 Win=2619648 Len=0
433 14.988629  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=7 Ack=25 Win=2619648 Len=3
434 14.988641  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=25 Ack=10 Win=2619648 Len=0
435 14.988702  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=25 Ack=10 Win=2619648 Len=12
436 14.988728  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=10 Ack=37 Win=2619648 Len=0
437 14.988917  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=10 Ack=37 Win=2619648 Len=3
438 14.988948  127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=37 Ack=13 Win=2619648 Len=0
439 14.989060  127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=37 Ack=13 Win=2619648 Len=12
440 14.989092  127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=13 Ack=49 Win=2619648 Len=0
441 14.989184  127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=13 Ack=49 Win=2619648 Len=3

> Frame 429: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface \Device\NPF_Loopback,
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 59280, Dst Port: 6666, Seq: 4, Ack: 13, Len: 3
v Data (3 bytes)
    Data: 310d0a
    [Length: 3]

0000  02 00 00 00 45 00 00 2b  6b b9 40 00 80 06 00 00   ····E··+ k·@·····
0010  7f 00 00 01 7f 00 00 01  e7 90 1a 0a 50 ee 8a df   ········ ····P···
0020  34 5a 0f c3 50 18 27 f9  2d 3b 00 00 31 0d 0a      4Z··P·'· -;··1··

Data (data.data), 3 bytes          Packets: 866 · Displayed: 55 (6.4%)          Profile: Default
```

Fig. 20.



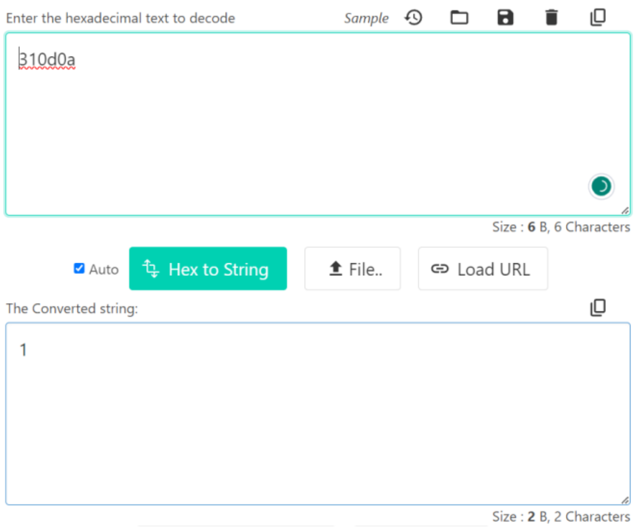
```
tcp.port==6666

No.  Time       Source     Destination  Protocol Length Info
422 14.979316   127.0.0.1  127.0.0.1    TCP      56 59280 → 6666 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
423 14.979409   127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
424 14.979471   127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
425 14.982396   127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=3
426 14.982414   127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=1 Ack=4 Win=2619648 Len=0
427 14.982957   127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=1 Ack=4 Win=2619648 Len=12
428 14.982991   127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=4 Ack=13 Win=2619648 Len=0
429 14.983134   127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=4 Ack=13 Win=2619648 Len=3
430 14.983145   127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=13 Ack=7 Win=2619648 Len=0
431 14.988392   127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=13 Ack=7 Win=2619648 Len=12
432 14.988429   127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=7 Ack=25 Win=2619648 Len=0
433 14.988629   127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=7 Ack=25 Win=2619648 Len=3
434 14.988641   127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=25 Ack=10 Win=2619648 Len=0
435 14.988702   127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=25 Ack=10 Win=2619648 Len=12
436 14.988728   127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=10 Ack=37 Win=2619648 Len=0
437 14.988917   127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=10 Ack=37 Win=2619648 Len=3
438 14.988948   127.0.0.1  127.0.0.1    TCP      44 6666 → 59280 [ACK] Seq=37 Ack=13 Win=2619648 Len=0
439 14.989060   127.0.0.1  127.0.0.1    TCP      56 6666 → 59280 [PSH, ACK] Seq=37 Ack=13 Win=2619648 Len=12
440 14.989092   127.0.0.1  127.0.0.1    TCP      44 59280 → 6666 [ACK] Seq=13 Ack=49 Win=2619648 Len=0
441 14.989184   127.0.0.1  127.0.0.1    TCP      47 59280 → 6666 [PSH, ACK] Seq=13 Ack=49 Win=2619648 Len=3

Frame 433: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface \Device\NPF_Loopback,
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 59280, Dst Port: 6666, Seq: 7, Ack: 25, Len: 3
Data (3 bytes)
    Data: 390d0a
    [Length: 3]

0000  02 00 00 00 45 00 00 2b  6b bd 40 00 80 06 00 00   ····E··+ k·@·····
0010  7f 00 00 01 7f 00 00 01  e7 90 1a 0a 50 ee 8a e2   ········ ····P···
0020  34 5a 0f cf 50 18 27 f9  25 2c 00 00 39 0d 0a      4Z··P·'· %,··9··

Data (data.data), 3 bytes          Packets: 866 · Displayed: 55 (6.4%)          Profile: Default
```

Fig. 21.



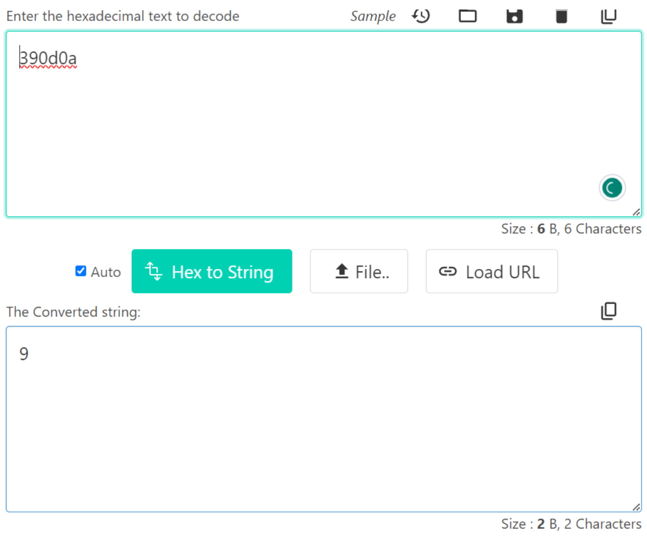
```
Enter the hexadecimal text to decode
370d0a
Size : 6 B, 6 Characters

The Converted string:
7
Size : 2 B, 2 Characters
```

Fig. 22.



```
Enter the hexadecimal text to decode
310d0a
Size : 6 B, 6 Characters

The Converted string:
1
Size : 2 B, 2 Characters
```

Fig. 23.



```
390d0a
```

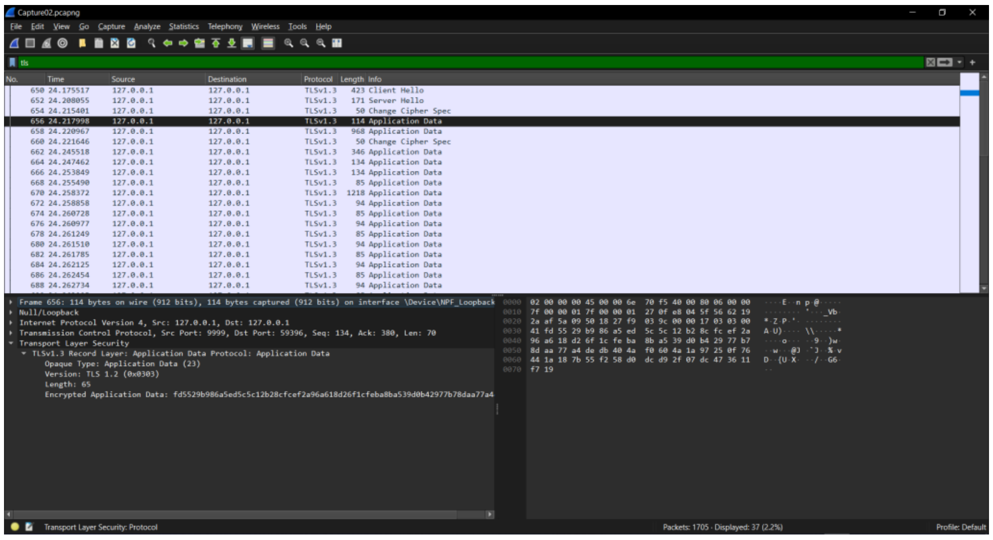
Size : 6 B, 6 Characters

```
9
```

Size : 2 B, 2 Characters

Fig. 24.

Capture of the Client 02 Communication (Secure Connection)


Capture02.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tls

```
No.   Time       Source     Destination  Protocol Length Info
  650 24.175517  127.0.0.1  127.0.0.1    TLSv1.3   423 Client Hello
  652 24.208055  127.0.0.1  127.0.0.1    TLSv1.3   171 Server Hello
  654 24.215401  127.0.0.1  127.0.0.1    TLSv1.3    50 Change Cipher Spec
  656 24.217998  127.0.0.1  127.0.0.1    TLSv1.3   114 Application Data
  658 24.220967  127.0.0.1  127.0.0.1    TLSv1.3   968 Application Data
  660 24.221646  127.0.0.1  127.0.0.1    TLSv1.3    50 Change Cipher Spec
  662 24.245518  127.0.0.1  127.0.0.1    TLSv1.3   346 Application Data
  664 24.247462  127.0.0.1  127.0.0.1    TLSv1.3   134 Application Data
  666 24.253849  127.0.0.1  127.0.0.1    TLSv1.3   134 Application Data
  668 24.255490  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  670 24.258372  127.0.0.1  127.0.0.1    TLSv1.3  1218 Application Data
  672 24.258858  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  674 24.260728  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  676 24.260977  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  678 24.261249  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  680 24.261510  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  682 24.261785  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  684 24.262125  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  686 24.262454  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  688 24.262734  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
```

```
> Frame 656: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface \Device\NPF_Loopback
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 9999, Dst Port: 59396, Seq: 134, Ack: 380, Len: 70
v Transport Layer Security
  v TLSv1.3 Record Layer: Application Data Protocol: Application Data
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 65
      Encrypted Application Data: fd5529b986a5ed5c5c12b28cfcef2a96a618d26f1cfeba8ba539d0b42977b78daa77a4
```

```
0000  02 00 00 00 45 00 00 6e  70 f5 40 00 80 06 00 00   ····E··n p·@····
0010  7f 00 00 01 7f 00 00 01  27 0f e8 04 5f 56 62 19   ········ '···_Vb·
0020  2a af 5a 09 50 18 27 f9  03 9c 00 00 17 03 03 00   *·Z·P·'· ········
0030  41 fd 55 29 b9 86 a5 ed  5c 5c 12 b2 8c fc ef 2a   A·U)···· \\·····*
0040  96 a6 18 d2 6f 1c fe ba  8b a5 39 d0 b4 29 77 b7   ····o··· ··9··)w·
0050  8d aa 77 a4 de db 40 4a  f0 60 4a 1a 97 25 0f 76   ··w···@J ·`J··%·v
0060  44 1a 18 7b 55 f2 58 d0  dc d9 2f 07 dc 47 36 11   D··{U·X· ··/··G6·
0070  f7 19                                              ··
```

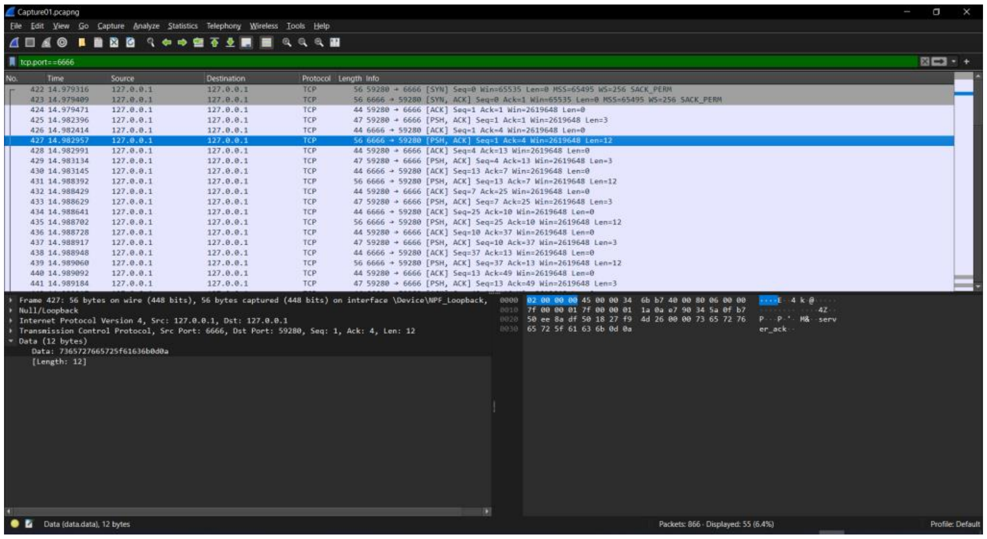
Transport Layer Security: Protocol   Packets: 1705 · Displayed: 37 (2.2%)   Profile: Default

Fig. 25.

Question 03

The absence of encryption in TCP protocol packets allows for the observation of application data in the hexadecimal format within captured packets of the Client 01 – Server communication. In contrast, the TLS protocol encrypts data transfer, preventing the observation of content in captured packets of the Client 02 – Server communication.

Furthermore, the server-client program design does not include sent client data in the server response. Consequently, it is not possible to observe client data within the server response. Only, the server response the string of "server_ack", remains observable in the Client 01 – Server communication.



tcp.port==6666

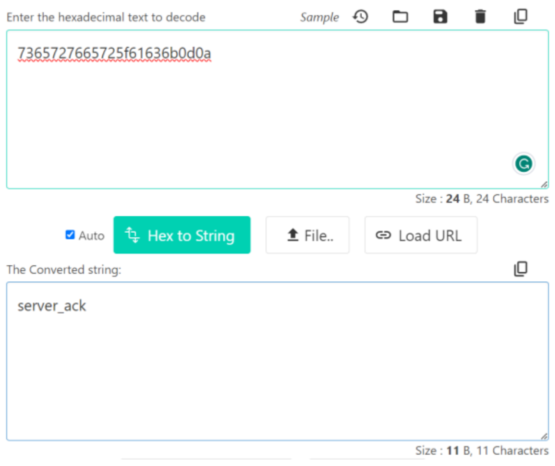
```
No.   Time       Source     Destination Protocol Length Info
  422 14.979316  127.0.0.1  127.0.0.1   TCP      56 59280 → 6666 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
  423 14.979409  127.0.0.1  127.0.0.1   TCP      56 6666 → 59280 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
  424 14.979471  127.0.0.1  127.0.0.1   TCP      44 59280 → 6666 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
  425 14.982396  127.0.0.1  127.0.0.1   TCP      47 59280 → 6666 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=3
  426 14.982414  127.0.0.1  127.0.0.1   TCP      44 6666 → 59280 [ACK] Seq=1 Ack=4 Win=2619648 Len=0
  427 14.982957  127.0.0.1  127.0.0.1   TCP      56 6666 → 59280 [PSH, ACK] Seq=1 Ack=4 Win=2619648 Len=12
  428 14.982991  127.0.0.1  127.0.0.1   TCP      44 59280 → 6666 [ACK] Seq=4 Ack=13 Win=2619648 Len=0
  429 14.983134  127.0.0.1  127.0.0.1   TCP      47 59280 → 6666 [PSH, ACK] Seq=4 Ack=13 Win=2619648 Len=3
  430 14.983145  127.0.0.1  127.0.0.1   TCP      44 6666 → 59280 [ACK] Seq=13 Ack=7 Win=2619648 Len=0
  431 14.988392  127.0.0.1  127.0.0.1   TCP      56 6666 → 59280 [PSH, ACK] Seq=13 Ack=7 Win=2619648 Len=12
  432 14.988429  127.0.0.1  127.0.0.1   TCP      44 59280 → 6666 [ACK] Seq=7 Ack=25 Win=2619648 Len=0
  433 14.988629  127.0.0.1  127.0.0.1   TCP      47 59280 → 6666 [PSH, ACK] Seq=7 Ack=25 Win=2619648 Len=3
  434 14.988641  127.0.0.1  127.0.0.1   TCP      44 6666 → 59280 [ACK] Seq=25 Ack=10 Win=2619648 Len=0
  435 14.988702  127.0.0.1  127.0.0.1   TCP      56 6666 → 59280 [PSH, ACK] Seq=25 Ack=10 Win=2619648 Len=12
  436 14.988728  127.0.0.1  127.0.0.1   TCP      44 59280 → 6666 [ACK] Seq=10 Ack=37 Win=2619648 Len=0
  437 14.988917  127.0.0.1  127.0.0.1   TCP      47 59280 → 6666 [PSH, ACK] Seq=10 Ack=37 Win=2619648 Len=3
  438 14.988948  127.0.0.1  127.0.0.1   TCP      44 6666 → 59280 [ACK] Seq=37 Ack=13 Win=2619648 Len=0
  439 14.989060  127.0.0.1  127.0.0.1   TCP      56 6666 → 59280 [PSH, ACK] Seq=37 Ack=13 Win=2619648 Len=12
  440 14.989092  127.0.0.1  127.0.0.1   TCP      44 59280 → 6666 [ACK] Seq=13 Ack=49 Win=2619648 Len=0
  441 14.989184  127.0.0.1  127.0.0.1   TCP      47 59280 → 6666 [PSH, ACK] Seq=13 Ack=49 Win=2619648 Len=3
```

```
> Frame 427: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_Loopback,
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 6666, Dst Port: 59280, Seq: 1, Ack: 4, Len: 12
v Data (12 bytes)
    Data: 7365727665725f61636b0d0a
    [Length: 12]
```

```
0000  02 00 00 00 45 00 00 34  6b b7 40 00 80 06 00 00   ····E··4 k·@·····
0010  7f 00 00 01 7f 00 00 01  1a 0a e7 90 34 5a 0f b7   ············4Z··
0020  50 ee 8a df 50 18 27 f9  4d 26 00 00 73 65 72 76   P···P·'· M&··serv
0030  65 72 5f 61 63 6b 0d 0a                            er_ack··
```

Data (data.data), 12 bytes   Packets: 866 · Displayed: 55 (6.4%)   Profile: Default

Fig. 26.



Enter the hexadecimal text to decode

7365727665725f61636b0d0a

Size : **24** B, 24 Characters

The Converted string:

server_ack

Size : **11** B, 11 Characters

Fig. 27.

Question 04

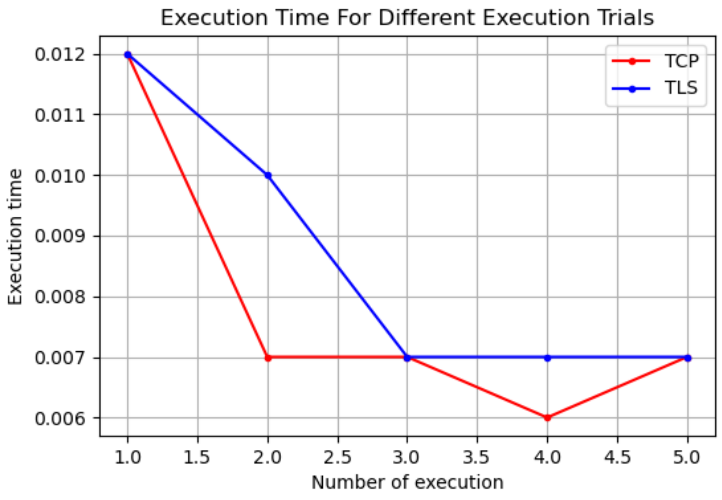
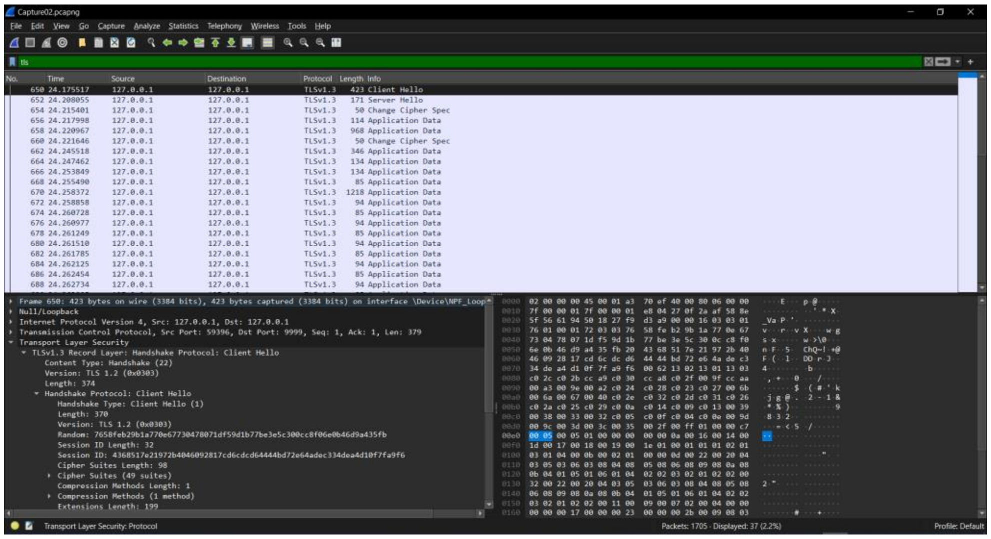


Fig. 28.

The runtime impact of TLS is generally higher compared to TCP alone because of the cryptographic operations involved in securing the communication. The graph constructed from the runtime time cost results of the TCP and TLS protocols of the server-client communication software supports this claim and estimations in terms of time complexity.

Question 05

In the TLS protocol it is possible to observe "Client Hello" and "Server Hello" packets.


Capture02.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tls

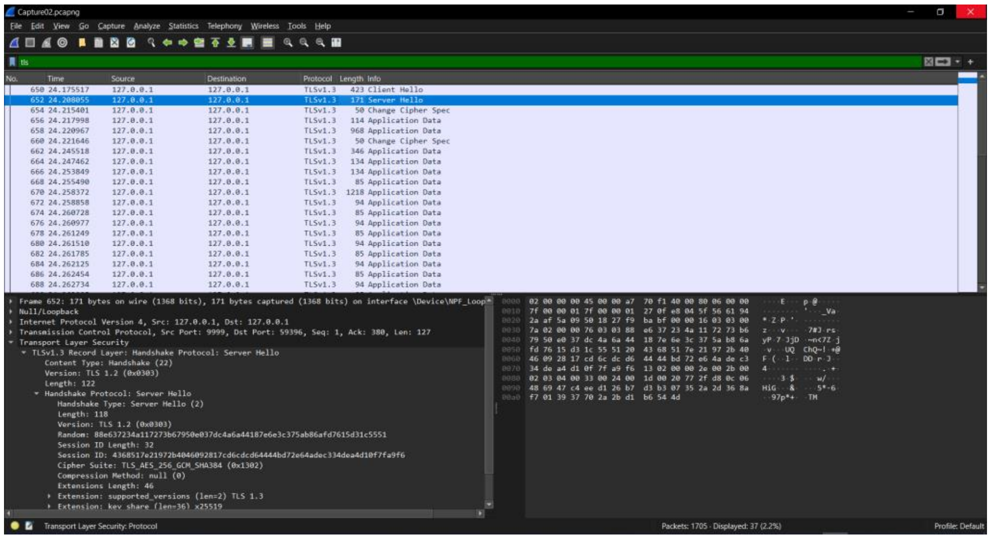
```
No.  Time       Source     Destination  Protocol  Length Info
650 24.175517   127.0.0.1  127.0.0.1    TLSv1.3   423 Client Hello
652 24.208055   127.0.0.1  127.0.0.1    TLSv1.3   171 Server Hello
654 24.215401   127.0.0.1  127.0.0.1    TLSv1.3    50 Change Cipher Spec
656 24.217998   127.0.0.1  127.0.0.1    TLSv1.3   114 Application Data
658 24.220967   127.0.0.1  127.0.0.1    TLSv1.3   968 Application Data
660 24.221646   127.0.0.1  127.0.0.1    TLSv1.3    50 Change Cipher Spec
662 24.245518   127.0.0.1  127.0.0.1    TLSv1.3   346 Application Data
664 24.247462   127.0.0.1  127.0.0.1    TLSv1.3   134 Application Data
666 24.253849   127.0.0.1  127.0.0.1    TLSv1.3   134 Application Data
668 24.255490   127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
670 24.258372   127.0.0.1  127.0.0.1    TLSv1.3  1218 Application Data
672 24.258858   127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
674 24.260728   127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
676 24.260977   127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
678 24.261249   127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
680 24.261510   127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
682 24.261785   127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
684 24.262125   127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
686 24.262454   127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
688 24.262734   127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
```

```
> Frame 650: 423 bytes on wire (3384 bits), 423 bytes captured (3384 bits) on interface \Device\NPF_Loop
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 59396, Dst Port: 9999, Seq: 1, Ack: 1, Len: 379
v Transport Layer Security
  v TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 374
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 370
        Version: TLS 1.2 (0x0303)
        Random: 7658feb29b1a770e67730478071df59d1b77be3e5c300cc8f06e0b46d9a435fb
        Session ID Length: 32
        Session ID: 4368517e21972b4046092817cd6cdcd64444bd72e64adec334dea4d10f7fa9f6
        Cipher Suites Length: 98
      > Cipher Suites (49 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 199
```

```
0000  02 00 00 00 45 00 01 a3  70 ef 40 00 80 06 00 00
0010  7f 00 00 01 7f 00 00 01  e8 04 27 0f 2a af 58 8e
0020  5f 56 61 94 50 18 27 f9  d3 a9 00 00 16 03 03 01
0030  76 01 00 01 72 03 03 76  58 fe b2 9b 1a 77 0e 67
0040  73 04 78 07 1d f5 9d 1b  77 be 3e 5c 30 0c c8 f0
0050  6e 0b 46 d9 a4 35 fb 20  43 68 51 7e 21 97 2b 40
0060  46 09 28 17 cd 6c dc d6  44 44 bd 72 e6 4a de c3
0070  34 de a4 d1 0f 7f a9 f6  00 62 13 02 13 01 13 03
0080  c0 2c c0 2b cc a9 c0 30  cc a8 c0 2f 00 9f cc aa
0090  00 a3 00 9e 00 a2 c0 24  c0 28 c0 23 c0 27 00 6b
00a0  00 6a 00 67 00 40 c0 2e  c0 32 c0 2d c0 31 c0 26
00b0  c0 2a c0 25 c0 29 c0 0a  c0 14 c0 09 c0 13 00 39
00c0  00 38 00 33 00 32 c0 05  c0 0f c0 04 c0 0e 00 9d
00d0  00 9c 00 3d 00 3c 00 35  00 2f 00 ff 01 00 00 c7
00e0  00 05 00 05 01 00 00 00  00 00 0a 00 16 00 14 00
00f0  1d 00 17 00 18 00 19 00  1e 01 00 01 01 01 02 01
0100  03 01 04 00 0b 00 02 01  00 00 0d 00 22 00 20 04
0110  03 05 03 06 03 08 04 08  05 08 06 08 09 08 0a 08
0120  0b 04 01 05 01 06 01 04  02 02 03 02 01 02 02 00
0130  32 00 22 00 20 04 03 05  03 06 03 08 04 08 05 08
0140  06 08 09 08 0a 08 0b 04  01 05 01 06 01 04 02 02
0150  03 02 01 02 02 00 11 00  09 00 07 02 00 04 00 00
0160  00 00 00 17 00 00 00 23  00 00 00 2b 00 09 08 03
```

Transport Layer Security: Protocol   Packets: 1705 · Displayed: 37 (2.2%)   Profile: Default

Fig. 29.



```
Capture02.pcapng
tls
No.   Time       Source     Destination  Protocol Length Info
  650 24.175517  127.0.0.1  127.0.0.1    TLSv1.3   423 Client Hello
  652 24.208055  127.0.0.1  127.0.0.1    TLSv1.3   171 Server Hello
  654 24.215401  127.0.0.1  127.0.0.1    TLSv1.3    50 Change Cipher Spec
  656 24.217998  127.0.0.1  127.0.0.1    TLSv1.3   114 Application Data
  658 24.220967  127.0.0.1  127.0.0.1    TLSv1.3   968 Application Data
  660 24.221646  127.0.0.1  127.0.0.1    TLSv1.3    50 Change Cipher Spec
  662 24.245518  127.0.0.1  127.0.0.1    TLSv1.3   346 Application Data
  664 24.247462  127.0.0.1  127.0.0.1    TLSv1.3   134 Application Data
  666 24.253849  127.0.0.1  127.0.0.1    TLSv1.3   134 Application Data
  668 24.255490  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  670 24.258372  127.0.0.1  127.0.0.1    TLSv1.3  1218 Application Data
  672 24.258858  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  674 24.260728  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  676 24.260977  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  678 24.261249  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  680 24.261510  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  682 24.261785  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  684 24.262125  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data
  686 24.262454  127.0.0.1  127.0.0.1    TLSv1.3    85 Application Data
  688 24.262734  127.0.0.1  127.0.0.1    TLSv1.3    94 Application Data

> Frame 652: 171 bytes on wire (1368 bits), 171 bytes captured (1368 bits) on interface \Device\NPF_Loop
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 9999, Dst Port: 59396, Seq: 1, Ack: 380, Len: 127
v Transport Layer Security
  v TLSv1.3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 122
    v Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 118
        Version: TLS 1.2 (0x0303)
        Random: 88e637234a117273b67950e037dc4a6a44187e6e3c375ab86afd7615d31c5551
        Session ID Length: 32
        Session ID: 4368517e21972b4046092817cd6cdcd64444bd72e64adec334dea4d10f7fa9f6
        Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
        Compression Method: null (0)
        Extensions Length: 46
      > Extension: supported_versions (len=2) TLS 1.3
      > Extension: key_share (len=36) x25519

0000  02 00 00 00 45 00 00 a7  70 f1 40 00 80 06 00 00   ····E···p·@·····
0010  7f 00 00 01 7f 00 00 01  27 0f e8 04 5f 56 61 94   ········'···_Va·
0020  2a af 5a 09 50 18 27 f9  ba bf 00 00 16 03 03 00   *·Z·P·'·········
0030  7a 02 00 00 76 03 03 88  e6 37 23 4a 11 72 73 b6   z···v····7#J·rs·
0040  79 50 e0 37 dc 4a 6a 44  18 7e 6e 3c 37 5a b8 6a   yP·7·JjD·~n<7Z·j
0050  fd 76 15 d3 1c 55 51 20  43 68 51 7e 21 97 2b 40   ·v···UQ ChQ~!·+@
0060  46 09 28 17 cd 6c dc d6  44 44 bd 72 e6 4a de c3   F·(··l··DD·r·J··
0070  34 de a4 d1 0f 7f a9 f6  13 02 00 00 2e 00 2b 00   4···········.·+·
0080  02 03 04 00 33 00 24 00  1d 00 20 77 2f d8 0c 06   ····3·$··· w/···
0090  48 69 47 c4 ee d1 26 b7  d3 b3 07 35 2a 2d 36 8a   HiG···&····5*-6·
00a0  f7 01 39 37 70 2a 2b d1  b6 54 4d                  ··97p*+··TM

Transport Layer Security: Protocol        Packets: 1705 · Displayed: 37 (2.2%)        Profile: Default
```

Fig. 30.

The Client Hello packet is sent by the client to the server to initiate the TLS handshake process. It serves as the client's way of expressing its intention to establish a secure connection with the server.

The contents of the Client Hello packet:

Client Version: The highest TLS version supported by the client.

Random Data: A random value generated by the client, which is used to ensure freshness in the key exchange process.

Session ID: An optional field that allows the client to resume a previous session.

Cipher Suites: A list of cryptographic algorithms supported by the client, in order of preference.

Compression Methods: A list of compression algorithms supported by the client (though compression is rarely used in modern TLS implementations).

The Server Hello packet is sent by the server in response to the Client Hello. It contains the server's acknowledgment of the client's request to establish a secure connection and includes key parameters needed for the encryption process.

The contents of the Server Hello packet:

Server Version: The highest TLS version supported by both the client and server (selected from the client's and server's lists).

Random Data: A random value generated by the server, similar to the one in the Client Hello.

Session ID: An optional field that allows the server to resume a previous session or create a new one.

Cipher Suite: The cryptographic algorithm selected by the server from the client's list, indicating the agreed-upon parameters for securing the communication.

Compression Method: The compression algorithm selected by the server from the client's list (again, compression is rarely used in modern TLS).

Once the Client Hello and Server Hello are exchanged, the TLS handshake continues with the exchange of additional messages, including the server's public key, key exchange parameters, and other necessary information to establish a secure connection. The handshake concludes with both parties having agreed upon a shared secret key, which is then used to encrypt and decrypt the data exchanged during the session.

Question 06

The solution involves a feature called session resumption mechanisms. The idea is to allow a client and server to reuse a previously established session's cryptographic parameters, including the server's certificate, for subsequent connections.

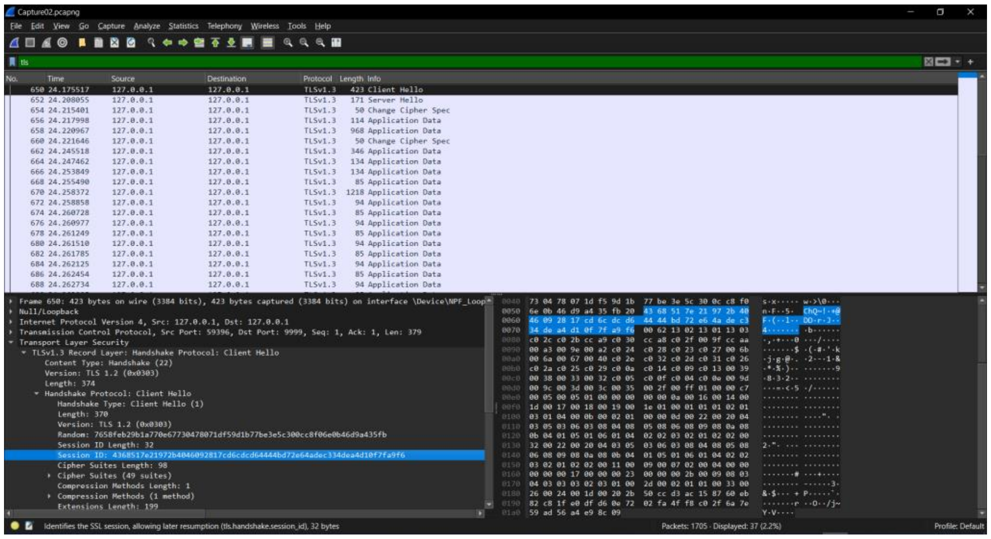
The session resumption process typically involves the use of a session identifier, which is negotiated during the initial TLS handshake.

After the initial handshake and successful establishment of a secure connection, the server generates a unique session identifier.

The server sends this session identifier to the client, and the client can use it in subsequent connections to request the reuse of the previously established session.

The client includes the session identifier in its Client Hello message during the handshake, indicating its desire to resume the session.

If the server recognizes the session identifier and agrees to resume the session, the cryptographic parameters (including the server's certificate) from the previous session are reused, saving the overhead of sending the certificate again.



```
Capture02.pcapng
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Wireless  Tools  Help
tls
No.     Time          Source        Destination    Protocol  Length Info
    650 24.175517     127.0.0.1     127.0.0.1      TLSv1.3    423 Client Hello
    652 24.208055     127.0.0.1     127.0.0.1      TLSv1.3    171 Server Hello
    654 24.215401     127.0.0.1     127.0.0.1      TLSv1.3     50 Change Cipher Spec
    656 24.217998     127.0.0.1     127.0.0.1      TLSv1.3    114 Application Data
    658 24.220967     127.0.0.1     127.0.0.1      TLSv1.3    968 Application Data
    660 24.221646     127.0.0.1     127.0.0.1      TLSv1.3     50 Change Cipher Spec
    662 24.245518     127.0.0.1     127.0.0.1      TLSv1.3    346 Application Data
    664 24.247462     127.0.0.1     127.0.0.1      TLSv1.3    134 Application Data
    666 24.253849     127.0.0.1     127.0.0.1      TLSv1.3    134 Application Data
    668 24.255490     127.0.0.1     127.0.0.1      TLSv1.3     85 Application Data
    670 24.258372     127.0.0.1     127.0.0.1      TLSv1.3   1218 Application Data
    672 24.258858     127.0.0.1     127.0.0.1      TLSv1.3     94 Application Data
    674 24.260728     127.0.0.1     127.0.0.1      TLSv1.3     85 Application Data
    676 24.260977     127.0.0.1     127.0.0.1      TLSv1.3     94 Application Data
    678 24.261249     127.0.0.1     127.0.0.1      TLSv1.3     85 Application Data
    680 24.261510     127.0.0.1     127.0.0.1      TLSv1.3     94 Application Data
    682 24.261785     127.0.0.1     127.0.0.1      TLSv1.3     85 Application Data
    684 24.262125     127.0.0.1     127.0.0.1      TLSv1.3     94 Application Data
    686 24.262454     127.0.0.1     127.0.0.1      TLSv1.3     85 Application Data
    688 24.262734     127.0.0.1     127.0.0.1      TLSv1.3     94 Application Data

> Frame 650: 423 bytes on wire (3384 bits), 423 bytes captured (3384 bits) on interface \Device\NPF_Loop
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 59396, Dst Port: 9999, Seq: 1, Ack: 1, Len: 379
v Transport Layer Security
  v TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 374
    v Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 370
        Version: TLS 1.2 (0x0303)
        Random: 7658feb29b1a770e67730478071df59d1b77be3e5c300cc8f06e0b46d9a435fb
        Session ID Length: 32
        Session ID: 4368517e21972b4046092817cd6cdcd64444bd72e64adec334dea4d10f7fa9f6
        Cipher Suites Length: 98
      > Cipher Suites (49 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 199

0040  73 04 78 07 1d f5 9d 1b  77 be 3e 5c 30 0c c8 f0
0050  6e 0b 46 d9 a4 35 fb 20  43 68 51 7e 21 97 2b 40
0060  46 09 28 17 cd 6c dc d6  44 44 bd 72 e6 4a de c3
0070  34 de a4 d1 0f 7f a9 f6  00 62 13 02 13 01 13 03
0080  c0 2c c0 2b cc a9 c0 30  cc a8 c0 2f 00 9f cc aa
0090  00 a3 00 9e 00 a2 c0 24  c0 28 c0 23 c0 27 00 6b
00a0  00 6a 00 67 00 40 c0 2e  c0 32 c0 2d c0 31 c0 26
00b0  c0 2a c0 25 c0 29 c0 0a  c0 14 c0 09 c0 13 00 39
00c0  00 38 00 33 00 32 c0 05  c0 0f c0 04 c0 0e 00 9d
00d0  00 9c 00 3d 00 3c 00 35  00 2f 00 ff 01 00 00 c7
00e0  00 05 00 05 01 00 00 00  00 00 0a 00 16 00 14 00
00f0  1d 00 17 00 18 00 19 00  1e 01 00 01 01 01 02 01
0100  03 01 04 00 0b 00 02 01  00 00 0d 00 22 00 20 04
0110  03 05 03 06 03 08 04 08  05 08 06 08 09 08 0a 08
0120  0b 04 01 05 01 06 01 04  02 02 03 02 01 02 02 00
0130  32 00 22 00 20 04 03 05  03 06 03 08 04 08 05 08
0140  06 08 09 08 0a 08 0b 04  01 05 01 06 01 04 02 02
0150  03 02 01 02 02 00 11 00  09 00 07 02 00 04 00 00
0160  00 00 00 17 00 00 00 23  00 00 00 2b 00 09 08 03
0170  04 03 03 03 02 03 01 00  2d 00 02 01 01 00 33 00
0180  26 00 24 00 1d 00 20 2b  50 cc d3 ac 15 87 60 eb
0190  82 c8 1f e0 df d6 0e 72  02 fa 4f f8 c0 2f 6a 7e
01a0  59 ad 56 a4 e9 8c 09

Identifies the SSL session, allowing later resumption (tls.handshake.session_id), 32 bytes
Packets: 1705 · Displayed: 37 (2.2%)          Profile: Default
```

Fig. 31.

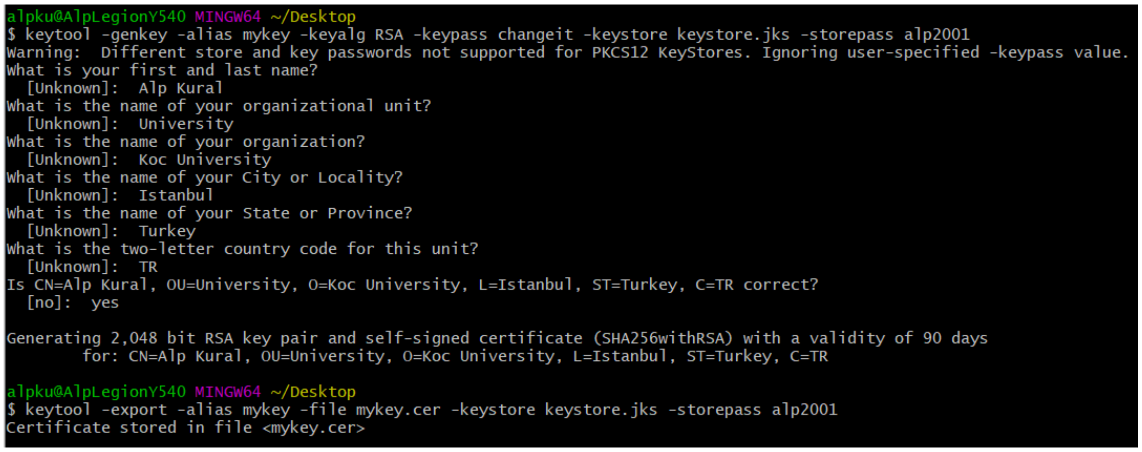
Part 2.2

mykey.cer -> certificate

keystore.jks -> keystore file

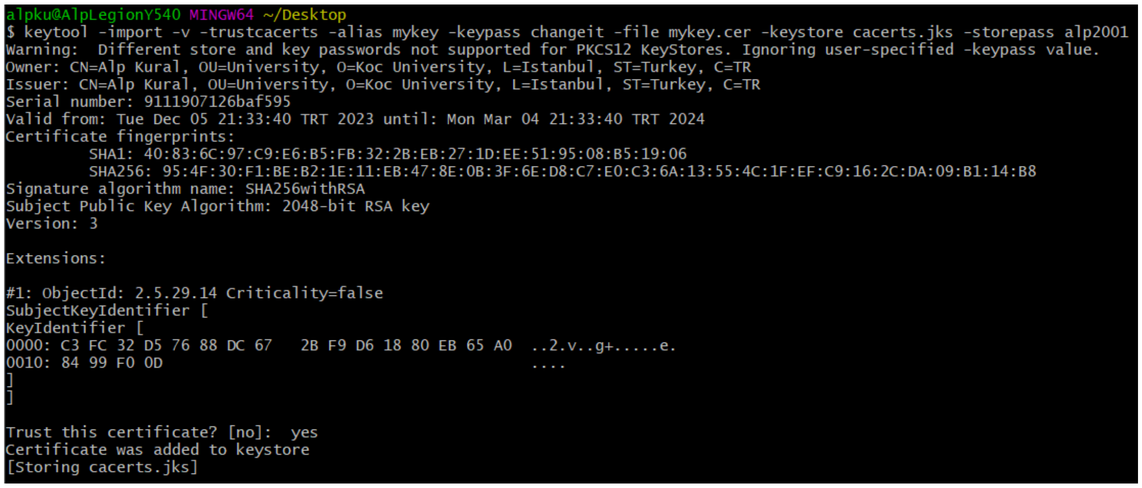
cacerts -> trust store file

The password used to generate keystore and trust store is “alp2001”.



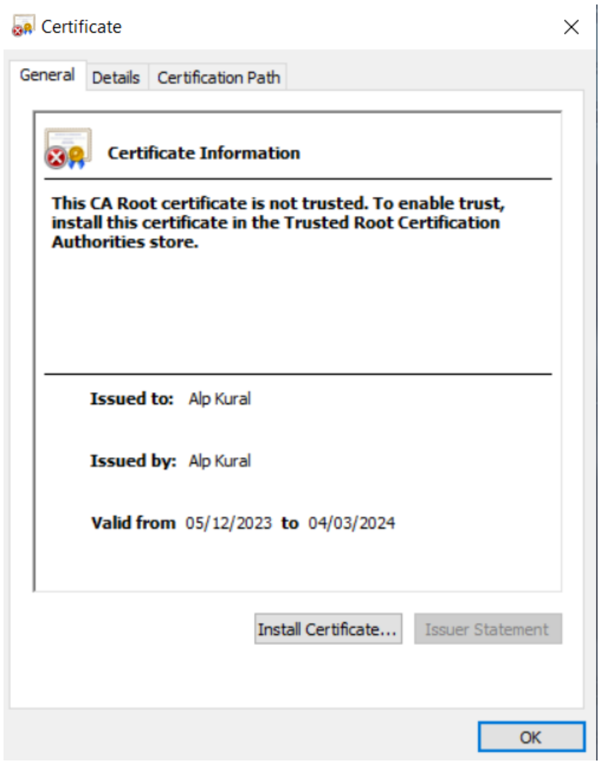
```
alpku@AlpLegionY540 MINGW64 ~/Desktop
$ keytool -genkey -alias mykey -keyalg RSA -keypass changeit -keystore keystore.jks -storepass alp2001
Warning:  Different store and key passwords not supported for PKCS12 KeyStores. Ignoring user-specified -keypass value.
What is your first and last name?
  [Unknown]:  Alp Kural
What is the name of your organizational unit?
  [Unknown]:  University
What is the name of your organization?
  [Unknown]:  Koc University
What is the name of your City or Locality?
  [Unknown]:  Istanbul
What is the name of your State or Province?
  [Unknown]:  Turkey
What is the two-letter country code for this unit?
  [Unknown]:  TR
Is CN=Alp Kural, OU=University, O=Koc University, L=Istanbul, ST=Turkey, C=TR correct?
  [no]:  yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 90 days
        for: CN=Alp Kural, OU=University, O=Koc University, L=Istanbul, ST=Turkey, C=TR

alpku@AlpLegionY540 MINGW64 ~/Desktop
$ keytool -export -alias mykey -file mykey.cer -keystore keystore.jks -storepass alp2001
Certificate stored in file <mykey.cer>
```

Fig. 32.



```
alpku@AlpLegionY540 MINGW64 ~/Desktop
$ keytool -import -v -trustcacerts -alias mykey -keypass changeit -file mykey.cer -keystore cacerts.jks -storepass alp2001
Warning:  Different store and key passwords not supported for PKCS12 KeyStores. Ignoring user-specified -keypass value.
Owner: CN=Alp Kural, OU=University, O=Koc University, L=Istanbul, ST=Turkey, C=TR
Issuer: CN=Alp Kural, OU=University, O=Koc University, L=Istanbul, ST=Turkey, C=TR
Serial number: 9111907126baf595
Valid from: Tue Dec 05 21:33:40 TRT 2023 until: Mon Mar 04 21:33:40 TRT 2024
Certificate fingerprints:
         SHA1: 40:83:6C:97:C9:E6:B5:FB:32:2B:EB:27:1D:EE:51:95:08:B5:19:06
         SHA256: 95:4F:30:F1:BE:B2:1E:11:EB:47:8E:0B:3F:6E:D8:C7:E0:C3:6A:13:55:4C:1F:EF:C9:16:2C:DA:09:B1:14:B8
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: C3 FC 32 D5 76 88 DC 67   2B F9 D6 18 80 EB 65 A0  ..2.v..g+.....e.
0010: 84 99 F0 0D                                        ....
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
[Storing cacerts.jks]
```

Fig. 33.



Certificate Information

This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

Issued to: Alp Kural

Issued by: Alp Kural

Valid from 05/12/2023 **to** 04/03/2024

Fig. 34.