# CS 315 : Computer Networks Lab
# Assignment - 5
# Wireshark Lab: TCP

**Ayush Mallick**
**CS22BT008**

---

## Part-0

```
ayushm@ayushm-HP-Pavilion-x360-Convertible-14-cd0xxx:~$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 492  bytes 46695 (46.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 492  bytes 46695 (46.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.200.240.35  netmask 255.255.240.0  broadcast 10.200.255.255
        inet6 fe80::ffd0:a85f:6677:d0b  prefixlen 64  scopeid 0x20<link>
        ether 28:3a:4d:63:21:71  txqueuelen 1000  (Ethernet)
        RX packets 87049  bytes 31317886 (31.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 178071  bytes 329683633 (329.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

## Part-1

### 1.1

Number of GET requests : *0*
Number of POST requests : *1*

### 1.2

*200 OK*

### 1.3

**a.**

Source IP address : *10.200.240.35*
Destination IP address : *128.119.245.12*

**b.**

Source port number : *54308*
Destination port number : *80*

**c.**

Three packets are exchanged during the TCP 3-way handshake: SYN, SYN-ACK, ACK

**d.**

**SYN**

Sequence number (raw) : *25144223*
Sequence number (relative) : *0*
Acknowledgment number (raw) : *0*
Acknowledgement number (relative) : *0*

**SYN-ACK**

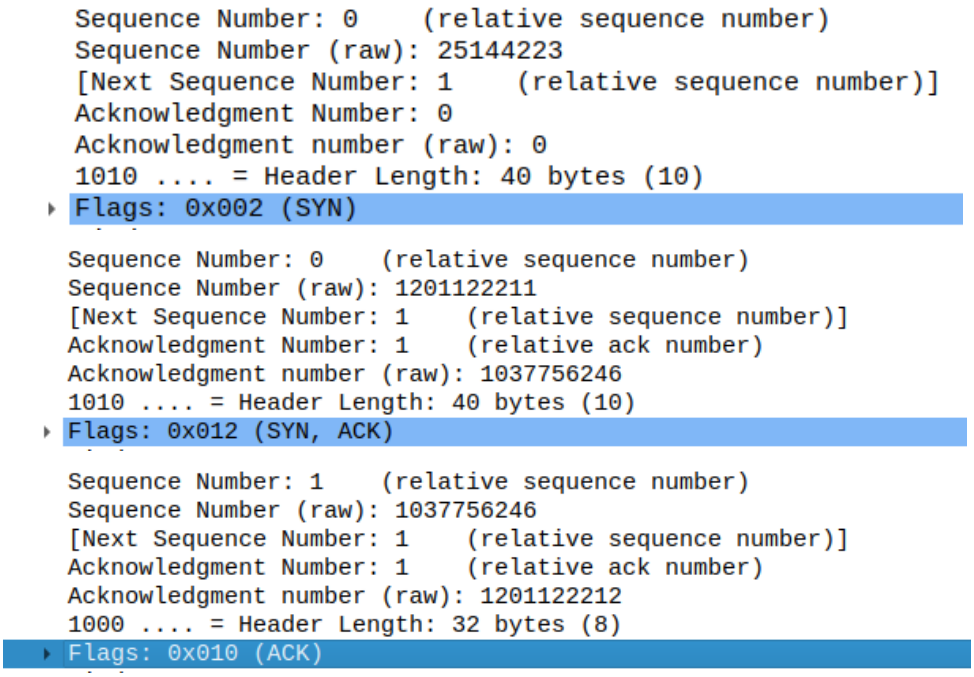
Sequence number (raw) : *1201122211*
Sequence number (relative) : *0*
Acknowledgment number (raw) : *1037756246*
Acknowledgement number (relative) : *1*

**ACK**

Sequence number (raw) : *1037756246*
Sequence number (relative) : *1*
Acknowledgment number (raw) : *1201122212*
Acknowledgement number (relative) : *1*

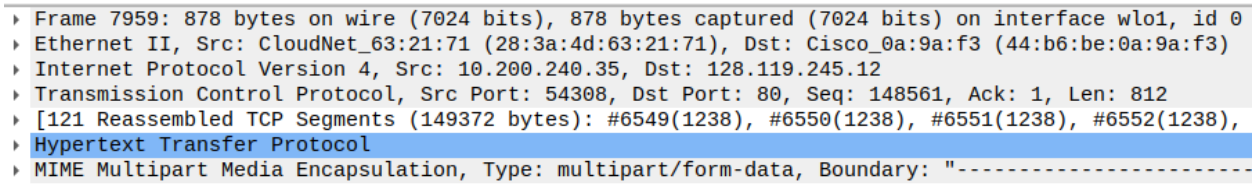
**e.**

The ‘Flags’ field confirms SYN, SYN-ACK, and ACK packets.



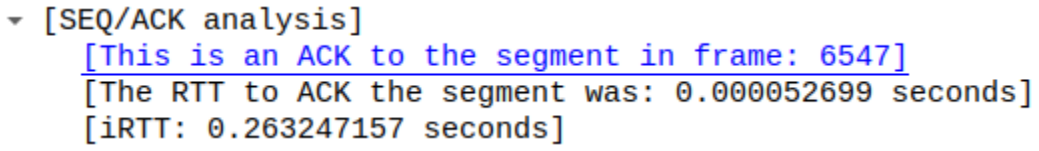
**f.**

*121* reassembled TCP segments.




**1.4**

RTT : *0.263247157* seconds




## Part-2

**2.1**

The Reassembled TCP Segments field in the HTTP POST packet indicates that the file was broken into multiple TCP segments during transmission. This happens because the size of the file exceeds the Maximum Segment Size (MSS) allowed by TCP, and the data is split into smaller chunks for efficient delivery. The field shows the total size of the file and how it was reconstructed at the destination from the multiple TCP segments.

**2.2**

The first segment contains the HTTP POST header, which includes details such as the destination URL and metadata about the file (e.g., Content-Length, Content-Type, and any encoding used). This segment confirms that the upload process was correctly initiated and provides the server with essential information about the file.

The last segment indicates the end of the file and ensures that all the file data has been transmitted successfully without truncation. It contains the final chunk of file data and possibly any trailing metadata (if included). Its arrival helps the receiver verify that the file size matches the Content-Length specified in the header, ensuring the integrity of the file.

**2.3**

The actual file contents begin in the TCP segment immediately after the HTTP POST header. The header of this segment ends with an empty line (CRLF), and the file contents start immediately after this.

### 2.4

Total size of reassembled TCP segments : *149372* bytes
This total size is equal to the file's size (in bytes) plus the size of the HTTP POST header, representing the complete data sent in the POST request, including both the header and the file contents.

### 2.5

The length of each TCP segment is *1238* bytes, except the last segment, which is *812* bytes.

## Part-3

### 3.1

Maximum Segment Size (MSS) value : *1250* bytes

```
▾ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  ▾ TCP Option - Maximum segment size: 1250 bytes
      Kind: Maximum Segment Size (2)
      Length: 4
      MSS Value: 1250
```

### 3.2

**First TCP segment**

Bytes in flight : *1238*
Calculated window size : *64256*

**Last TCP segment (HTTP POST packet)**

Bytes in flight : *62712*
Calculated window size : *64256*

The Bytes in flight value changes as new segments are sent or acknowledged, and the remaining unacknowledged data decreases.

| No. | Time | Source | Destination | Protocol | Length | Bytes in flight | Calculated window size | Info |
|---|---|---|---|---|---|---|---|---|
| 6549 | 2.897954662 | 10.200.240.35 | 128.119.245.12 | TCP | 1304 | 1238 | 64256 | 54308 → 80 [ACK |
| 7959 | 3.710659427 | 10.200.240.35 | 128.119.245.12 | HTTP | 878 | 62712 | 64256 | POST /wireshark |

### 3.3

Window size scaling factor : *128*

```
Window: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x1eef [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
```