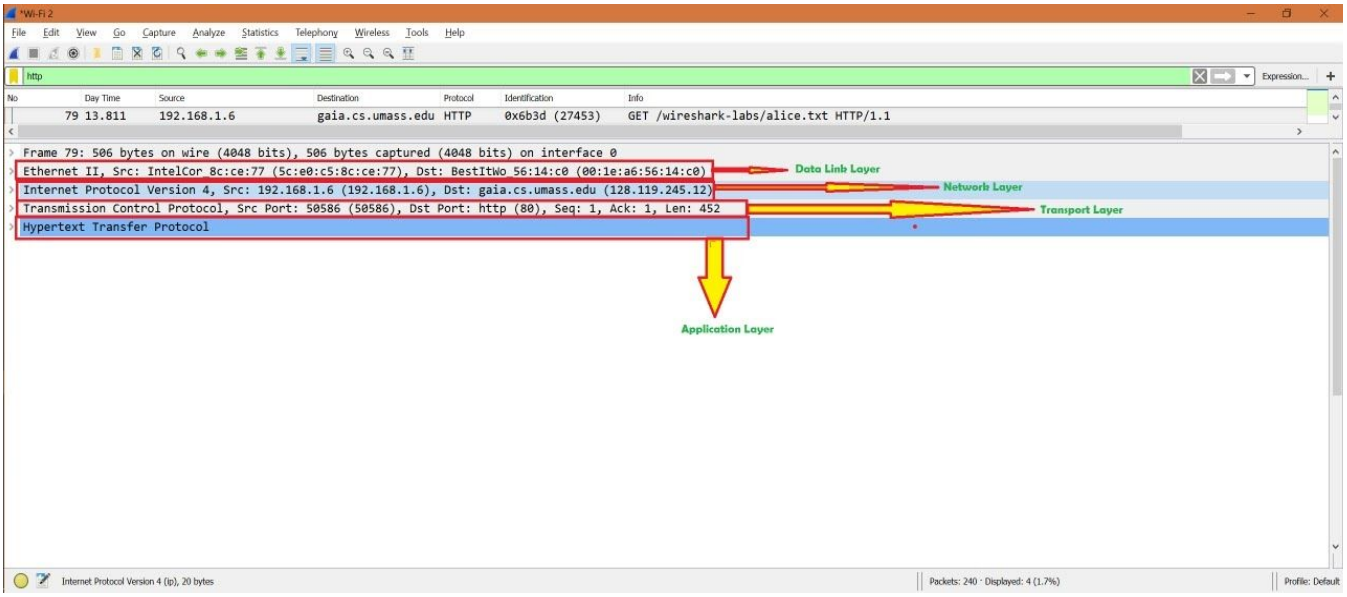
NAME : RITIK MANDLOI
ROLL NO: **180101066**

Drive Link: https://drive.google.com/drive/folders/1eCXhZfCtlLMkU-MP4z1UY00HyZdyQNvo?usp=sharing

CS342 LAB ASSIGNMENT 2

Q1)



http

```
No    Day Time   Source        Destination         Protocol  Identification   Info
   79 13.811     192.168.1.6   gaia.cs.umass.edu   HTTP      0x6b3d (27453)   GET /wireshark-labs/alice.txt HTTP/1.1
```

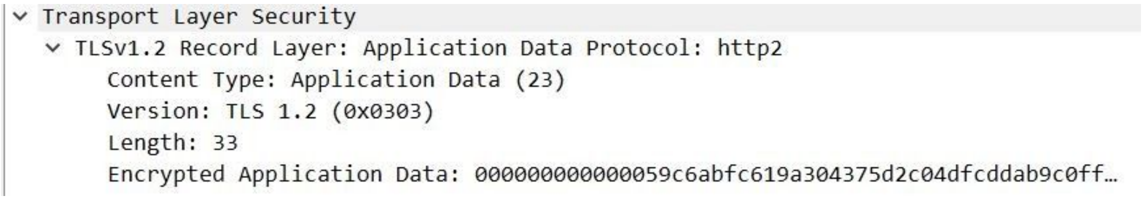
```
> Frame 79: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface 0
> Ethernet II, Src: IntelCor_8c:ce:77 (5c:e0:c5:8c:ce:77), Dst: BestItWo_56:14:c0 (00:1e:a6:56:14:c0)
> Internet Protocol Version 4, Src: 192.168.1.6 (192.168.1.6), Dst: gaia.cs.umass.edu (128.119.245.12)
> Transmission Control Protocol, Src Port: 50586 (50586), Dst Port: http (80), Seq: 1, Ack: 1, Len: 452
> Hypertext Transfer Protocol
```

Internet Protocol Version 4 (ip), 20 bytes | Packets: 240 · Displayed: 4 (1.7%) | Profile: Default

1. Protocols used in different layers are mentioned below with each layer and corresponding data with explanation.

a) **Application layer**
Protocol used : TLS v1.2, HTTP

Transport Layer Security (TLS), and its now-deprecated predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to provide communications security over a computer network. The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications.
i)Content-type ii)Legacy version iii)Length: iv)Protocol messages v)MAC and padding
Record Version is 16- byte value formatted in network order and record Length is a 16-byte value.When Wireshark can't determine how part of a packet should be formatted, it marks that chunk as “Data”.

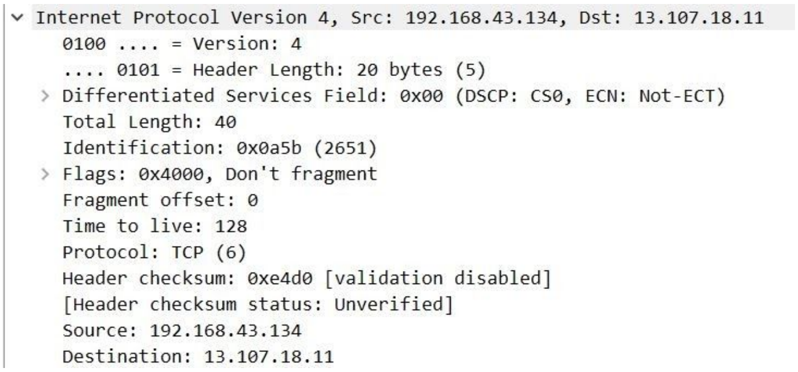


```
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: http2
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 33
      Encrypted Application Data: 000000000000059c6abfc619a304375d2c04dfcddab9c0ff…
```

The Application-Data-Protocol indicates the protocol used at the application layer, which is HTTP (secured with TLS) in this case. Content-Type indicates the type of payload, which is simply the Application Data in this case. In total there are a total of 4 types of Content-type. Version tells the protocol version used for securing the communication, which is TLSv1.2 here. Length is simply the length of the message which is 33 in this case. When the TLS layer is involved in handshaking the following "Client Hello" message is seen.

HyperText Transfer Protocol:
HTTP header fields provide required information about the request or response, or about the object sent in the message body.
There are four types of HTTP message headers: General-header, Client Request-header, Server response-header, Entity-header

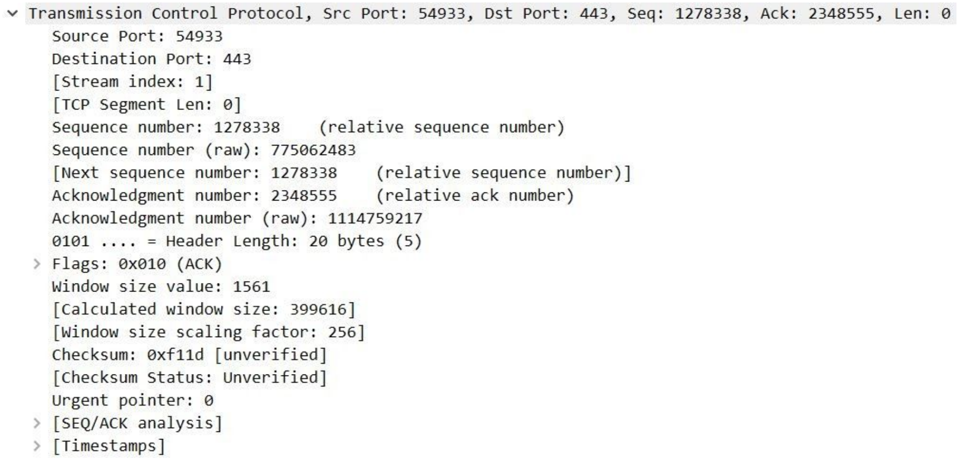
b)**Network Layer:**
Protocol used : IPv4 Packet format: IPv4 is one of the core protocols of standards-based internetworking methods on the Internet.IP is responsible for delivering data packets from the source host to the destination host. IPv4 is a connectionless protocol for use on packet-switched networks.
Description of the fields: The version of IP is 4(IPv4).The header length is 20 bytes and each word is 4 bytes.Total Length of the packet is 40 byte.CS0 implies that the service type is the best effort and Not-ECT implies that the packet is not using ECN Transport.The value of Time to live(TTL) is 128 hops. Header checksum is used to detect any error. Source contains my IP address and destination contains outlook.office.com IP address.



```
v Internet Protocol Version 4, Src: 192.168.43.134, Dst: 13.107.18.11
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x0a5b (2651)
  > Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0xe4d0 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.43.134
    Destination: 13.107.18.11
```

c)Transport Layer
Protocol used :TCP(Transmission Control Protocol)

Some important fields TCP headers have:
Source and Destination Port,Checksum , Sequence number, Flags, Acknowledgement number, Window size, Header Length, Urgent Pointer



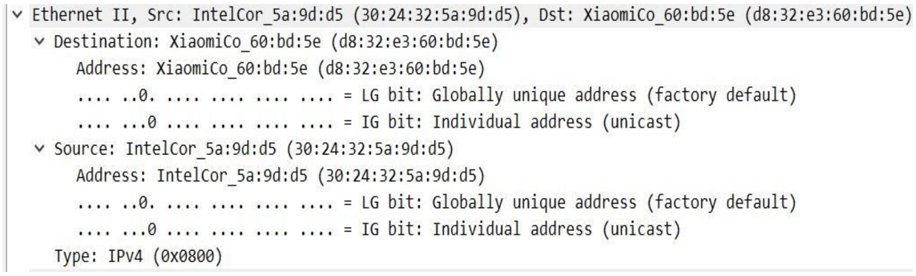
```
v Transmission Control Protocol, Src Port: 54933, Dst Port: 443, Seq: 1278338, Ack: 2348555, Len: 0
    Source Port: 54933
    Destination Port: 443
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 1278338    (relative sequence number)
    Sequence number (raw): 775062483
    [Next sequence number: 1278338    (relative sequence number)]
    Acknowledgment number: 2348555    (relative ack number)
    Acknowledgment number (raw): 1114759217
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 1561
    [Calculated window size: 399616]
    [Window size scaling factor: 256]
    Checksum: 0xf11d [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
```

The source and destination ports describe the particular process the packet is sent for. The header length is 20 bytes and hence a total of 5 words are present. Two flags ACK and PSH are present.ACK means that the machine sending the packet is acknowledging data that is received from another end. PSH is an indication to push the entire buffer immediately to the receiving application. The Window Size Value on packets from A to B indicates how much buffer space is available on A for receiving packets and its current value is 1561. The checksum is again used for error detection in the entire packet.

d)Link layer
Protocol used : Ethernet (II)

Ethernet(II) is the most common Local Area Network (LAN)technology. Some of the important fields of the header are:

Preamble/SFD:The starting of the frame and allows the sender and receiver to establish bit synchronization.
Source and Destination Address:MAC address of sender and reciever.
Data:Actual data and both IP header data is stored.
CRC:This is used to detect any in-transit corruption of data.



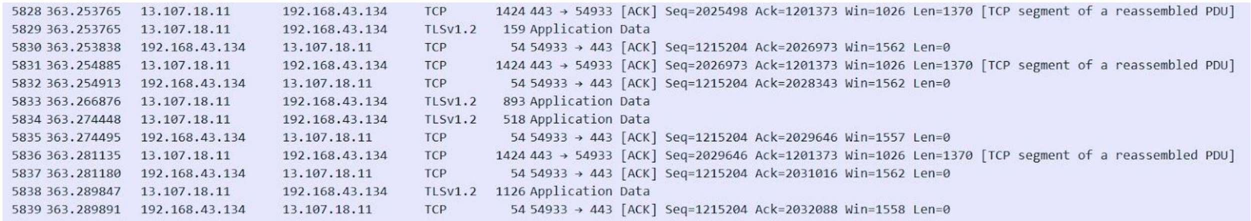
```
v Ethernet II, Src: IntelCor_5a:9d:d5 (30:24:32:5a:9d:d5), Dst: XiaomiCo_60:bd:5e (d8:32:e3:60:bd:5e)
  v Destination: XiaomiCo_60:bd:5e (d8:32:e3:60:bd:5e)
      Address: XiaomiCo_60:bd:5e (d8:32:e3:60:bd:5e)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: IntelCor_5a:9d:d5 (30:24:32:5a:9d:d5)
      Address: IntelCor_5a:9d:d5 (30:24:32:5a:9d:d5)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

Type indicates which protocol is encapsulated in the payload of the frame.Destination and source contain the MAC address of destination endpoint and sender’s endpoint. Both addresses are unicast and globally unique.

Q2)

Operations performed	Protocols used
Sending mail	TCP and TLSv1.2
Refresh Inbox	TCP and TLSv1.2
Creating and saving meeting	TCP and TLSv1.2
Deleting mail	TCP and TLSv1.2

Explanation:



```
5828 363.253765   13.107.18.11      192.168.43.134    TCP      1424 443 → 54933 [ACK] Seq=2025498 Ack=1201373 Win=1026 Len=1370 [TCP segment of a reassembled PDU]
5829 363.253765   13.107.18.11      192.168.43.134    TLSv1.2   159 Application Data
5830 363.253838   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2026973 Win=1562 Len=0
5831 363.254885   13.107.18.11      192.168.43.134    TCP      1424 443 → 54933 [ACK] Seq=2026973 Ack=1201373 Win=1026 Len=1370 [TCP segment of a reassembled PDU]
5832 363.254913   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2028343 Win=1562 Len=0
5833 363.266876   13.107.18.11      192.168.43.134    TLSv1.2   893 Application Data
5834 363.274448   13.107.18.11      192.168.43.134    TLSv1.2   518 Application Data
5835 363.274495   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2029646 Win=1557 Len=0
5836 363.281135   13.107.18.11      192.168.43.134    TCP      1424 443 → 54933 [ACK] Seq=2029646 Ack=1201373 Win=1026 Len=1370 [TCP segment of a reassembled PDU]
5837 363.281180   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2031016 Win=1562 Len=0
5838 363.289847   13.107.18.11      192.168.43.134    TLSv1.2  1126 Application Data
5839 363.289891   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2032088 Win=1558 Len=0
```

1)**TCP**
a) It was the only used protocol in this case i.e. outlook.office.com only uses TCP for transmission of data and not UDP. There are several reasons for the same.
b) TCP is more reliable than UDP. We can be sure that there will be no data loss.
c) TCP is used with TLSv which makes it more secure with reliable.
d) As mail information is quite confidential and we can not afford data loss and which is minimised here as handshaking is done at start.

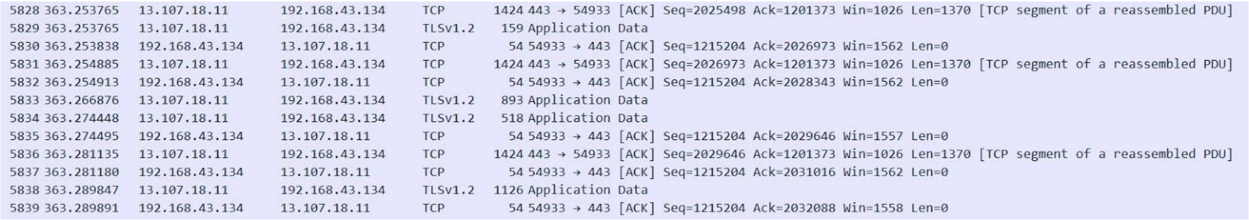
2) **TLSv1.2:**
a) Prevents eavesdropping and used for modification on internet traffic.
b) Protocol is highly secure and as mails contains private and confedential information it is very necessary to maintain privacy and security which is done using this protocol. Hackers or any external devices can not break and leak the data because of this.
c) During login or refreshing or sending mails data is kept safe. Particular in login password and credentials are safe because of this.

3)**Ethernet II**: Ethernet II is used at the link layer as it ensures reliable data transfer between two nodes and involves proper error handling and flow control mechanisms to minimize errors. This was seen in all functionalities.

Q3)
Operations Selected: -1) Refreshing the page
2)Sending mail

Refreshing the Page:

When we refresh the page and check if we have received a mail or not the required components are loaded from the server and required handshaking is done to establish the connection.



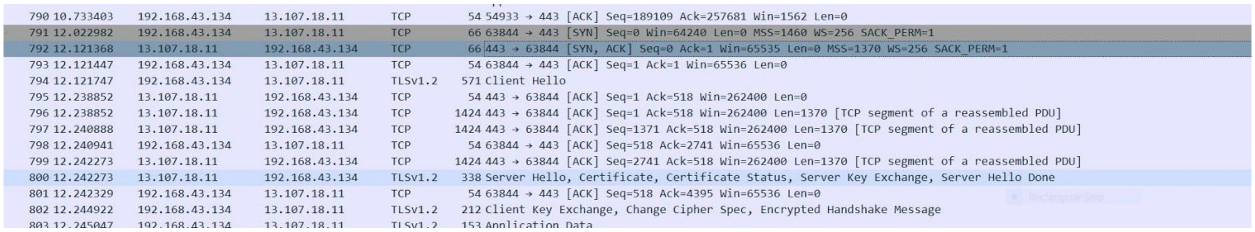
```
5828 363.253765   13.107.18.11      192.168.43.134    TCP      1424 443 → 54933 [ACK] Seq=2025498 Ack=1201373 Win=1026 Len=1370 [TCP segment of a reassembled PDU]
5829 363.253765   13.107.18.11      192.168.43.134    TLSv1.2   159 Application Data
5830 363.253838   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2026973 Win=1562 Len=0
5831 363.254885   13.107.18.11      192.168.43.134    TCP      1424 443 → 54933 [ACK] Seq=2026973 Ack=1201373 Win=1026 Len=1370 [TCP segment of a reassembled PDU]
5832 363.254913   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2028343 Win=1562 Len=0
5833 363.266876   13.107.18.11      192.168.43.134    TLSv1.2   893 Application Data
5834 363.274448   13.107.18.11      192.168.43.134    TLSv1.2   518 Application Data
5835 363.274495   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2029646 Win=1557 Len=0
5836 363.281135   13.107.18.11      192.168.43.134    TCP      1424 443 → 54933 [ACK] Seq=2029646 Ack=1201373 Win=1026 Len=1370 [TCP segment of a reassembled PDU]
5837 363.281180   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2031016 Win=1562 Len=0
5838 363.289847   13.107.18.11      192.168.43.134    TLSv1.2  1126 Application Data
5839 363.289891   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=1215204 Ack=2032088 Win=1558 Len=0
```

1)**TCP and TLS handshaking:**
The host sends a SYN packet to the destination which helps in synchronizing the sequence number. The destination then responds by sending an ACK packet which is an acknowledgement and another SYN packet which helps to synchronize with the source.In the end the source finally sends an ACK packet and the handshaking completes.

TLS v1.2 which is observed is a protocol in the Secure Socket Layer.Here we can see “Client Hello” “Server Hello” and Key Exchange.During this both the sides exchange messages to verify and establish the encryption method they use.Server Hello signifies that server can take requests from the client and hence TLS handshaking completes.

Sending the mail:

While sending a email the data that is sent from one host to other is passed through various hosts to reach the destination host.When the data sent is complete there is an acknowledgement from the server and there is exchange of message to break the TCP connection.The client firsts sends a FIN message to server and it acknowledges it by sending back a FIN-ACK message.The client on receiving this message ACK back to server and connection finally closes.



```
 790 10.733403   192.168.43.134    13.107.18.11      TCP        54 54933 → 443 [ACK] Seq=189109 Ack=257681 Win=1562 Len=0
 791 12.022982   192.168.43.134    13.107.18.11      TCP        66 63844 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
 792 12.121368   13.107.18.11      192.168.43.134    TCP        66 443 → 63844 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1370 WS=256 SACK_PERM=1
 793 12.121447   192.168.43.134    13.107.18.11      TCP        54 63844 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0
 794 12.121747   192.168.43.134    13.107.18.11      TLSv1.2   571 Client Hello
 795 12.238852   13.107.18.11      192.168.43.134    TCP        54 443 → 63844 [ACK] Seq=1 Ack=518 Win=262400 Len=0
 796 12.238852   13.107.18.11      192.168.43.134    TCP      1424 443 → 63844 [ACK] Seq=1 Ack=518 Win=262400 Len=1370 [TCP segment of a reassembled PDU]
 797 12.240888   13.107.18.11      192.168.43.134    TCP      1424 443 → 63844 [ACK] Seq=1371 Ack=518 Win=262400 Len=1370 [TCP segment of a reassembled PDU]
 798 12.240941   192.168.43.134    13.107.18.11      TCP        54 63844 → 443 [ACK] Seq=518 Ack=2741 Win=65536 Len=0
 799 12.242273   13.107.18.11      192.168.43.134    TCP      1424 443 → 63844 [ACK] Seq=2741 Ack=518 Win=262400 Len=1370 [TCP segment of a reassembled PDU]
 800 12.242273   13.107.18.11      192.168.43.134    TLSv1.2   338 Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
 801 12.242329   192.168.43.134    13.107.18.11      TCP        54 63844 → 443 [ACK] Seq=518 Ack=4395 Win=65536 Len=0
 802 12.244922   192.168.43.134    13.107.18.11      TLSv1.2   212 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
 803 12.245047   192.168.43.134    13.107.18.11      TLSv1.2   153 Application Data
```

Q4)

TIME	Throughput (Bytes/s)	RTT (ms)	Packet size (Bytes)	Packet Loss	No. of TCP & UDP	Responses Per Packet
1pm	53k	19ms	675	0%	TCP: 5620, UDP: 0	1.59
5pm	35k	55 ms	658	0%	TCP: 6017, UDP: 0	1.56
9pm	21k	25ms	657	0%	TCP: 2399, UDP: 0	1.62

No UDP packet was found

Q5)

The observation is that the destination IP address changes with time. During day-time, the destination address was **13.107.18.11** and during the night the observed IP was **40.100.140.114.**

Reasons:

a) Load balancing: Many websites have several servers set up across the world. This helps in load balancing i.e. if one particular server receives a lot of requests at one time, the next request is sent to some other server by the router. This helps to keep the network traffic stable.
b) Reliability: If a server fails due to some reasons, then there should be other servers which respond to the clients. The website crashes if the only server goes down. So using multiple servers helps to provide reliability.