CS342 Networks Lab Assignment 02

Name: Pooja Gajendra Bhagat
Roll Number: 180101057

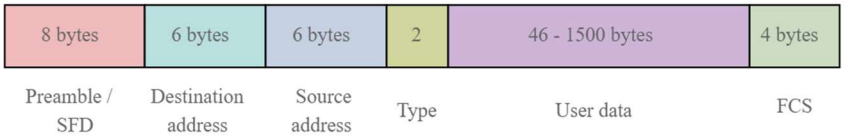
App Name:
Outlook client (Desktop App)

Traces Link: https://rb.gy/ucedcn

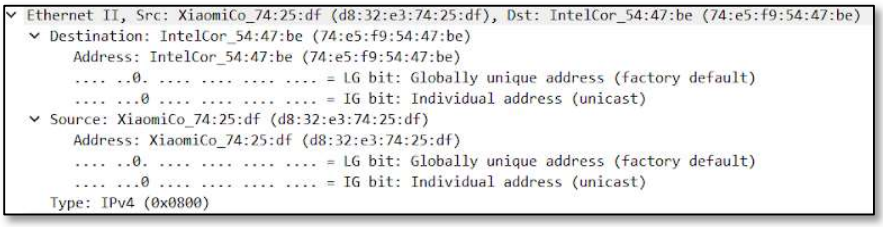
Question 1

Data Link Layer

Ethernet II



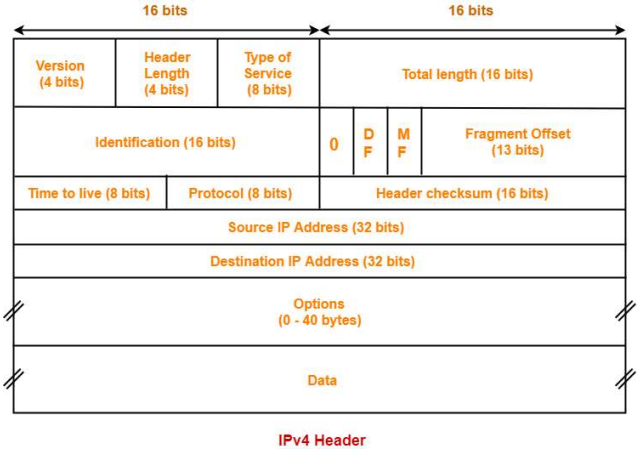
Ethernet is a widely used LAN technology. Ethernet packet starts with a preamble which enables the receiver to synchronise and know that a data frame is about to be sent. SFD (start frame delimiter) indicates the start of the frame(1 byte). Destination Address gives the station MAC address where the packet is to intended to be sent. The first bit indicates whether it is an individual address or a group address. The source address consists of six bytes, and it is used to identify the sending station. Type field is the one which differentiates between 802.3 and Ethernet II (Type of Ethernet). User Data block contains the payload data to be sent and it may be up to 1500 bytes long. FCS contains Cyclic Redundancy Check (CRC) for error detection and analysis.



The Src and Dst field contain the source and destination endpoint MAC addresses whose values are d8:32:e3:74:25:df and 74:e5::f9:54:47:be respectively. Both source and destination contain a LG bit which distinguishes vendor assigned (0) and administratively assigned MAC addresses (1) and IG bit which specifies whether the MAC address is unicast (0) or multicast (1). In this case both LG and IG bits are 0 for source and destination each indicating both are Globally unique addresses and unicast. Type indicates the upper layer protocol to be used which is IPv4 in this case.

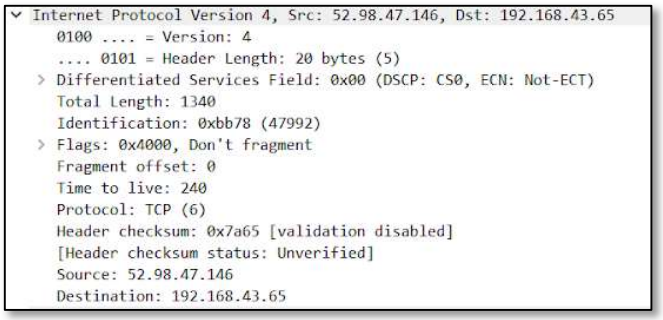
Network Layer

Internet protocol version 4



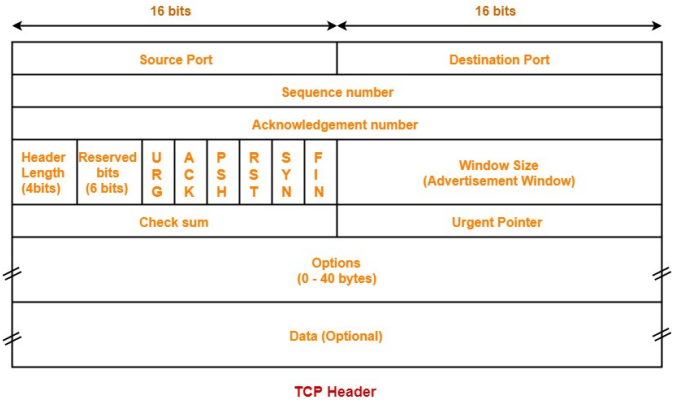
IP is responsible to deliver data packets from the source host to the destination host. The packet header contains various fields. Version indicates the version of IP used in this case, 4. Header length specifies the length of IP header. Total length specifies the total length of the datagram (in bytes). DF bit stands for Do Not Fragment and MF stands for More Fragment bits. Time to live indicates the maximum number of hops a datagram can take to reach the destination. Protocol field tells the next level protocol to the network layer at the destination side. Source/Destination Address contains the logical address of sender and receiver datagram. Options are used for Record route, Source routing, Padding etc.

We can observe from the given data that the version of IP protocol used is 4 i.e. IPv4. The header length is 20 bytes and the total length of packet is 1340 bytes. The DF bit is set which indicates that packet should not be fragmented. Time to live is 240 which means that packet can make at most 240 hops. The next level protocol to be used is TCP and source and destination addresses are 52.98.47.146 and 192.168.43.65 respectively.



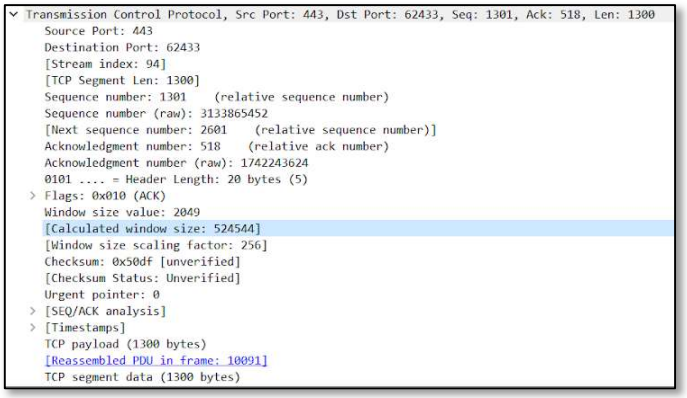
Transport Layer

Transmission Control Protocol

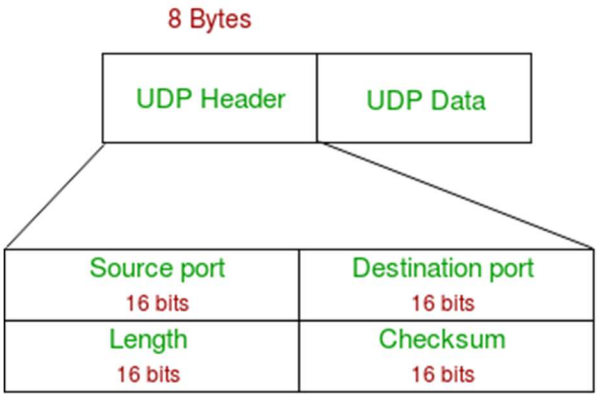


TCP is used for organizing data in a way that ensures the secure transmission between the server and client. Source Port and Destination Port indicates the port of the sending and receiving application. Sequence Number contains the sequence number of the first data byte. Acknowledgement Number contains the sequence number of the data byte that receiver expects to receive next from the sender. Header Length specifies the length of the TCP header. There is a total of 6 types of Flags of 1 bit each. Some of them are ACK, PSH and SYN. Checksum is used to verify the integrity of data in the TCP payload. Window Size contains the size of the receiving window of the sender. It advertises how much data (in bytes) the sender can receive without acknowledgement. Urgent Pointer indicates how much data in the current segment counting from the first data byte is urgent. Options are used for different purposes like timestamp, window size extension, parameter negotiation, padding.

The Source port and destination port in this case is 443 and 62433 respectively between which the communication is happening. Sequence number is 1301 and acknowledgement number is 518. Only ACK flag is set which means that the machine sending the packet is acknowledging data that is received from another end. Header length is 20 bytes. Window size is 2049 and Urgent Pointer is 0 which means that no further bytes are urgent.



User Datagram Protocol



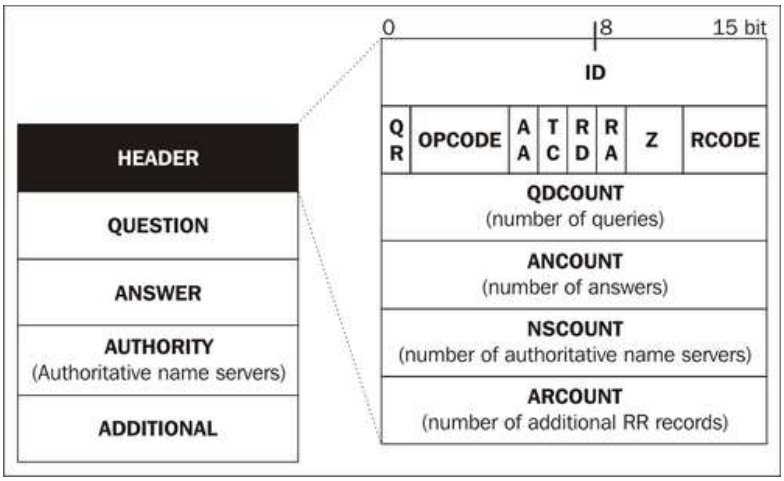
UDP (User Datagram Protocol) is a communications protocol that is primarily used for establishing low-latency and loss-tolerating connections between applications on the internet. Source Port is a 16-bit field that identifies the port of the sending application. Destination Port is a 16-bit field which identifies the port of the sending application. Length specifies the combined length of UDP Header and Encapsulated data. Checksum is used for error control and calculated on UDP Header, encapsulated data and IP pseudo-header.

The fields which can be observed from the following information are the source port and the destination port which are 53 and 64530 respectively. The length of the packet is 148 and the checksum status is unverified. Stream index is an internal Wireshark mapping to [IP address A, TCP port A, IP address B, TCP port B].

```
User Datagram Protocol, Src Port: 53, Dst Port: 64530
    Source Port: 53
    Destination Port: 64530
    Length: 148
    Checksum: 0x25a2 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 36]
  [Timestamps]
      [Time since first frame: 0.063084000 seconds]
      [Time since previous frame: 0.063084000 seconds]
```

Application Layer

Domain Name System



The DNS protocol works when your computer sends out a DNS query to a name server to resolve a domain. The first 12 bytes is the header section. ID is a 16-bit identifier assigned by the program that generates any kind of query. QR is one-bit field that specifies whether this message is a query (0), or a response (1). OPCODE specifies the kind of query in the message. RCODE or response code for messages specifies error condition. QDCOUNT, ANCOUNT, NSCOUNT, ARCOUNT, specify number of, entries in the question section, resource records in the answer section, name server resource records in the authority records section resource records in the additional records section respectively. QUESTION section contains information about the query that is being made. ANSWER section contains the resource records for the name that was originally queried. AUTHORITY section contains records of other authoritative servers.

```
Domain Name System (response)
    Transaction ID: 0xafe5
  Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 5
    Authority RRs: 0
    Additional RRs: 0
  Queries
    > acdc-direct.office.com: type A, class IN
  Answers
    > acdc-direct.office.com: type CNAME, class IN, cname outlook.ms-acdc.office.com
    > outlook.ms-acdc.office.com: type CNAME, class IN, cname BOM-efz.ms-acdc.office.com
    > BOM-efz.ms-acdc.office.com: type A, class IN, addr 40.100.140.18
    > BOM-efz.ms-acdc.office.com: type A, class IN, addr 52.98.88.66
    > BOM-efz.ms-acdc.office.com: type A, class IN, addr 40.100.140.242
    [Request In: 10203]
    [Time: 0.063084000 seconds]
```

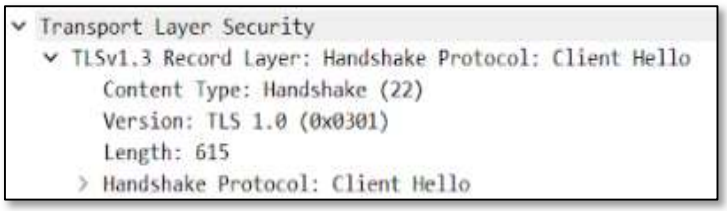
Transaction ID of this DNS request is 0xafe5 (16-bit). There are many flag values specified of which Opcode specifies that the query type is standard and the first bit (QR) indicates that the type is response. QDCOUNT or Questions in the pictures gives the number of queries which is 1. Answer RR(ANCOUNT) is 5, Authority RR(NSCOUNT) is 0 and Addition RR(ARCOUNT) is also 0. The next two sections list out the queries and answers.

Transport Layer Security

Byte	+0	+1	+2	+3
0	Content type			
1..4	Version		Length	
5..n	*Payload*			
n..m	MAC			
m..p	Padding (block ciphers only)			

TLS provides security in communication by encrypting the application data. The basic unit of data in SSL (Secure Socket Layer) is a record. Each record consists of a five-byte record header, followed by data. Content Type specifies whether the content is Handshake, Application Data, Alert, Change Cipher Spec etc. Version field specifies the version of TLS being used. Length gives the length of packet including header. Payload contains the data of the packet and MAC is the Message authentication code.

In this case we can observe that the TLS Version being used is 1.3 (TLSv1.3) and content type is handshake. The TLS Handshake Protocol is used to authenticate the participants of the communication and negotiate an encryption algorithm. The length of the packet is 615 including header.



Question 2

Reasons for protocols being used

As outlook is an email application user data security is one of the major concerns. Also, for email clients, reliability is import as data loss when transferring emails may lead to bigger issues.

Functionalities of Application

1. Launching Application
2. Login
3. Refresh Inbox
4. Save Draft
5. Send Mail
6. Create Group
7. Create Folder
8. Move email from one folder to another
9. Download Attachment
10. Delete Mail
11. Logout
12. Closing Application

Ethernet II: Ethernet is the most widely used data link layer protocol. It is preferred over other protocols because of its reliable data transfer, high speed and security. It involves proper error handling and flow control mechanisms for error handling along with CRC for error detection and preamble for synchronization.

TCP: Majorly TCP protocol is used for data communications between the client and the outlook server because:

- Transmission control protocol is a reliable protocol i.e. it ensures data which is sent reaches the destination successfully. When a sender doesn't get an acknowledgement after a certain period of time, it will assume that the packet got lost on its way. So, it will send it again.
- TCP ensures the ordered delivery of packets. Although packets may come out of order, TCP rearranges them before sending them to application.
- TCP also ensures proper error handling and flow control mechanisms to minimize the error loss rate as we cannot afford any data loss for communication over email.

IPv4: Internet Protocol version 4 is a connection less protocol which enables data communication over packet switched networks like the internet. It is generally used with the TCP protocol as TCP is only compatible with IP at network Layer. IP is neither reliable nor guarantees ordered data transfer therefore TCP is needed to supplement these shortcomings of IP protocol.

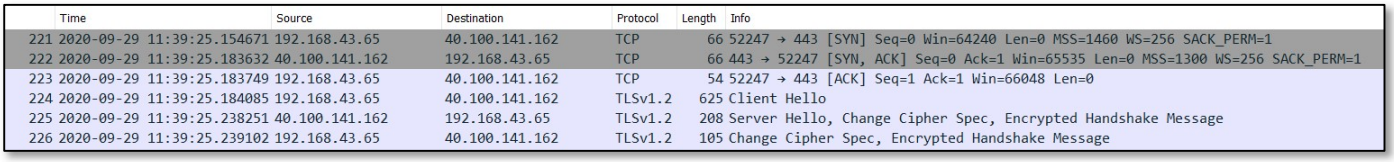
TLS: Outlook is secured by an TLS certificate. TLSv1.3 is a Secure Sockets Layer (SSL) protocol which encrypts the application data and prevents hackers who snoop packets to get any important information and prevent unwanted eavesdropping. So, all the login credentials (e.g. your email and password) are safely transmitted from client to server in encrypted format so that no hacker can get access of user data.

UDP: UDP protocol is rarely used by outlook only in association with a DNS request. UDP is a connectionless protocol which provides faster but low reliability data transfer. It is useful when individual packets have to be sent across the network. UDP packets contain smaller headers which makes the packets lightweight and hence faster data transfer. DNS query to fetch the IP address of outlook uses UDP as the underlying protocol.

DNS: DNS protocol is used to map domain names to IP addresses, which indicate the server address to which the client has to connect. It allows users to have human-readable domains while ensuring a mapping to the IP addresses corresponding to the domain names.

Question 3

Launching the application



	Time	Source	Destination	Protocol	Length	Info
221	2020-09-29 11:39:25.154671	192.168.43.65	40.100.141.162	TCP	66	52247 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
222	2020-09-29 11:39:25.183632	40.100.141.162	192.168.43.65	TCP	66	443 → 52247 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 WS=256 SACK_PERM=1
223	2020-09-29 11:39:25.183749	192.168.43.65	40.100.141.162	TCP	54	52247 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=0
224	2020-09-29 11:39:25.184085	192.168.43.65	40.100.141.162	TLSv1.2	625	Client Hello
225	2020-09-29 11:39:25.238251	40.100.141.162	192.168.43.65	TLSv1.2	208	Server Hello, Change Cipher Spec, Encrypted Handshake Message
226	2020-09-29 11:39:25.239102	192.168.43.65	40.100.141.162	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message

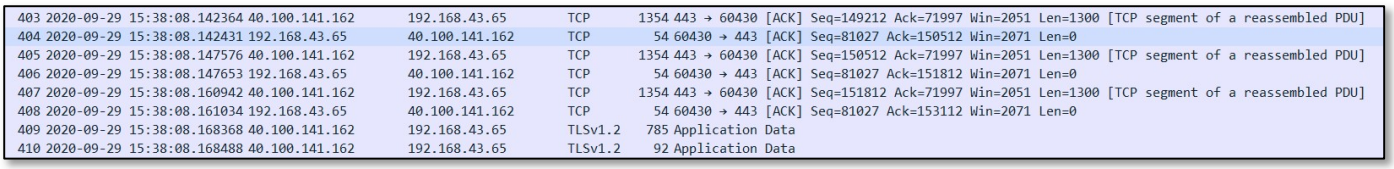
When the application is launched two handshaking procedure takes place, first 3-way TCP handshaking (first 3 messages) and then TLS Handshaking (last 3 messages).

TCP 3-way Handshake: This process is used to make a connection between the client (PC) and the server (outlook.office.com) in a TCP/IP network. It is a 3-step process between port 52247 of client and port 443 of the server.

- *Step 1*: **SYN** : The client sends a segment to the server with SYN (Synchronize Sequence Number) which is the initial sequence number it plans to use and informs the server that client is likely to start communication.
- *Step 2*: **SYN, ACK** : The server sends a response with SYN-ACK bit set. ACK (Acknowledgement) indicates that the server has acknowledged the client's sequence number and SYN signifies server's sequence number with which it is likely to start with.
- *Step 3*: **ACK** : The client then sends a message with ACK bit set, and acknowledges the server's response. The client and server establish a reliable connection for actual data transfer.

TLS Handshake: TLS handshake is used to make the connection secure. First the TLS protocol sends 'Client Hello' message (line 4) to initiate a session with the server. The server responds with a 'Server Hello' message (line 5) containing the server certificate which is used primarily for authentication, cipher suite requirements, and randomly generated data for creating session keys. The client responds with a client key (line 6) and a secure connection is established between the client and the server.

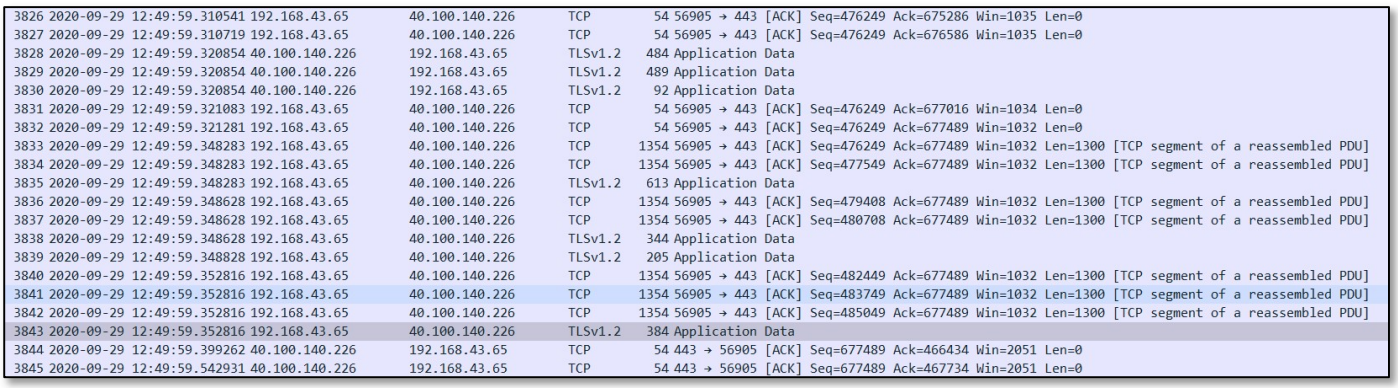
Refresh Inbox



403	2020-09-29 15:38:08.142364	40.100.141.162	192.168.43.65	TCP	1354	443 → 60430 [ACK] Seq=149212 Ack=71997 Win=2051 Len=1300 [TCP segment of a reassembled PDU]
404	2020-09-29 15:38:08.142431	192.168.43.65	40.100.141.162	TCP	54	60430 → 443 [ACK] Seq=81027 Ack=150512 Win=2071 Len=0
405	2020-09-29 15:38:08.147576	40.100.141.162	192.168.43.65	TCP	1354	443 → 60430 [ACK] Seq=150512 Ack=71997 Win=2051 Len=1300 [TCP segment of a reassembled PDU]
406	2020-09-29 15:38:08.147653	192.168.43.65	40.100.141.162	TCP	54	60430 → 443 [ACK] Seq=81027 Ack=151812 Win=2071 Len=0
407	2020-09-29 15:38:08.160942	40.100.141.162	192.168.43.65	TCP	1354	443 → 60430 [ACK] Seq=151812 Ack=71997 Win=2051 Len=1300 [TCP segment of a reassembled PDU]
408	2020-09-29 15:38:08.161034	192.168.43.65	40.100.141.162	TCP	54	60430 → 443 [ACK] Seq=81027 Ack=153112 Win=2071 Len=0
409	2020-09-29 15:38:08.168368	40.100.141.162	192.168.43.65	TLSv1.2	785	Application Data
410	2020-09-29 15:38:08.168488	40.100.141.162	192.168.43.65	TLSv1.2	92	Application Data

When inbox is refreshed server sends the data to the outlook client. Application data with a with TCP and IP headers make up a segment. The client sends Acknowledgement packets of length = 0, indicating successful reception of data from the server. The packets may arrive out of order as the they may take different path from source to destination due to network congestion and load balancing. TCP examines the segments and reassembles the PDU (Protocol Data Unit) in frame. Once all these segments arrive, they are re-assembled and then data is sent to the application layer.

Send Mail



3826	2020-09-29 12:49:59.310541	192.168.43.65	40.100.140.226	TCP	54	56905 → 443 [ACK] Seq=476249 Ack=675286 Win=1035 Len=0
3827	2020-09-29 12:49:59.310719	192.168.43.65	40.100.140.226	TCP	54	56905 → 443 [ACK] Seq=476249 Ack=676586 Win=1035 Len=0
3828	2020-09-29 12:49:59.320854	40.100.140.226	192.168.43.65	TLSv1.2	484	Application Data
3829	2020-09-29 12:49:59.320854	40.100.140.226	192.168.43.65	TLSv1.2	489	Application Data
3830	2020-09-29 12:49:59.320854	40.100.140.226	192.168.43.65	TLSv1.2	92	Application Data
3831	2020-09-29 12:49:59.321083	192.168.43.65	40.100.140.226	TCP	54	56905 → 443 [ACK] Seq=476249 Ack=677016 Win=1034 Len=0
3832	2020-09-29 12:49:59.321281	192.168.43.65	40.100.140.226	TCP	54	56905 → 443 [ACK] Seq=476249 Ack=677489 Win=1032 Len=0
3833	2020-09-29 12:49:59.348283	192.168.43.65	40.100.140.226	TCP	1354	56905 → 443 [ACK] Seq=476249 Ack=677489 Win=1032 Len=1300 [TCP segment of a reassembled PDU]
3834	2020-09-29 12:49:59.348283	192.168.43.65	40.100.140.226	TCP	1354	56905 → 443 [ACK] Seq=477549 Ack=677489 Win=1032 Len=1300 [TCP segment of a reassembled PDU]
3835	2020-09-29 12:49:59.348283	192.168.43.65	40.100.140.226	TLSv1.2	613	Application Data
3836	2020-09-29 12:49:59.348628	192.168.43.65	40.100.140.226	TCP	1354	56905 → 443 [ACK] Seq=479408 Ack=677489 Win=1032 Len=1300 [TCP segment of a reassembled PDU]
3837	2020-09-29 12:49:59.348628	192.168.43.65	40.100.140.226	TCP	1354	56905 → 443 [ACK] Seq=480708 Ack=677489 Win=1032 Len=1300 [TCP segment of a reassembled PDU]
3838	2020-09-29 12:49:59.348628	192.168.43.65	40.100.140.226	TLSv1.2	344	Application Data
3839	2020-09-29 12:49:59.348828	192.168.43.65	40.100.140.226	TLSv1.2	205	Application Data
3840	2020-09-29 12:49:59.352816	192.168.43.65	40.100.140.226	TCP	1354	56905 → 443 [ACK] Seq=482449 Ack=677489 Win=1032 Len=1300 [TCP segment of a reassembled PDU]
3841	2020-09-29 12:49:59.352816	192.168.43.65	40.100.140.226	TCP	1354	56905 → 443 [ACK] Seq=483749 Ack=677489 Win=1032 Len=1300 [TCP segment of a reassembled PDU]
3842	2020-09-29 12:49:59.352816	192.168.43.65	40.100.140.226	TCP	1354	56905 → 443 [ACK] Seq=485049 Ack=677489 Win=1032 Len=1300 [TCP segment of a reassembled PDU]
3843	2020-09-29 12:49:59.352816	192.168.43.65	40.100.140.226	TLSv1.2	384	Application Data
3844	2020-09-29 12:49:59.399262	40.100.140.226	192.168.43.65	TCP	54	443 → 56905 [ACK] Seq=677489 Ack=466434 Win=2051 Len=0
3845	2020-09-29 12:49:59.542931	40.100.140.226	192.168.43.65	TCP	54	443 → 56905 [ACK] Seq=677489 Ack=467734 Win=2051 Len=0

When a mail is to be sent, the client first tries to send the data to the server in the form of TCP packets. All the data in the packets are encrypted using a shared secret key which are acknowledged by the server by an ACK message.

TCP packets are used to transfer all types of data be it, text or a file. There are TLS packets in between the TCP packets which ensures successful transfer of data between in encrypted format between client and server. These TCP packets may take different routes to travel from source to destination, subject to network congestion and load balancing, and may arrive at the destination out of order, but TCP protocol ensures the rearrangement of the packets to the original order.

Handshaking

The handshaking process takes place in order to establish rules for communication when a computer attempts to communicate with another device/server. Signals are usually exchanged between two devices to establish a communication link and formulate the protocols for the communication.
Handshaking takes place only when the connection is established with a particular server IP. There were no handshaking messages in case of any other functionality except for launching the application as the protocol for communication are only configured when the connection is established.
In very rare case handshaking sequence were observed when the application was running. This is due to the reason that the connection to particular server had to be closed possibly due to network congestion or overloading and another connection was established to a new server IP.

Question 4

	Morning	**Afternoon**	**Night**
Throughput (kilobytes/sec)	58	69	138
Average RTT (s)	0.098	0.085	0.023
No. of Packets Lost	0	0	0
Avg. Packet Size (bytes)	627	615.35	642
Count of TCP Packets	7253	9656	4064
Count of UDP Packets	132	141	89
No. of Response per Request	1.22	1.04	1.23

Question 5

The content is being shared between different source/destination IP during the three different times of the day. The time and IPs are mentioned below

Time	**Morning**	**Afternoon**	**Night**
IP address	13.107.18.11	40.100.140.226	40.100.141.162

The reason for this happening is that because there are multiple servers catering to outlook.office.com website and the requests are directed to a particular host depending on the network traffic. Multiple servers are deployed to make the site faster by load balancing which reduces the network congestion. It also increases network reliability as even if a server fails, the load can be distributed. Hence there is no single point of failure and the server can provide uninterrupted service to the users.