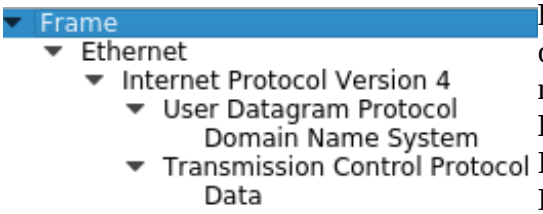
Name : Ankit Kumar Singh
Roll No. 150101086

Networks Lab_Assigment2
Date: 21st Feb'18

Data is collected at differnet times of day from Team Viewer App (on linux enviroment). Data can be viewed here <link>

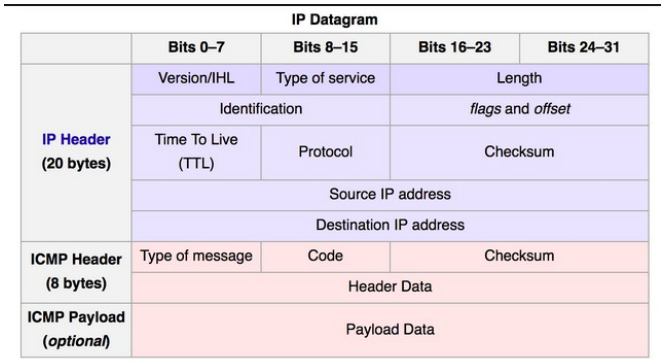
Sol1.**Protocol hierarchy** observed : (All the mentioned packet format can be seen as entries in ques 2. which are explained in much depth there.)



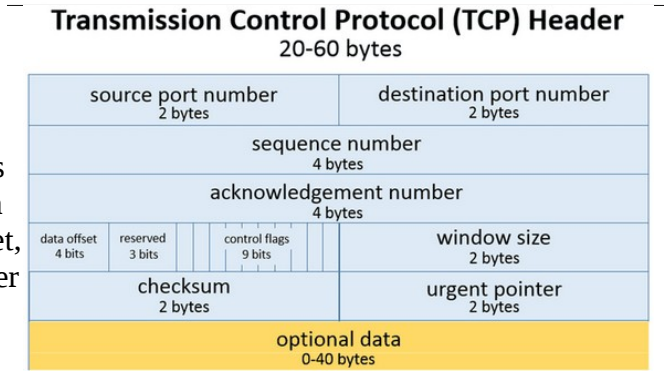
Physical layer: Ethernet frame is prefixed with a preamble(56 bits of alternating (0&1)'s) and SFD(start frame delimiter). But these are removed by Wireshark and hence not observed in the Frame.
Data Link Layer: **Ethernet II**
Ethernet frame here starts with Ethernet header which includes Dest. MAC address and Source MAC address.This is followed by Type/Length of payload data for any protocol like IP etc. This is followed by Payload from above layers. Then comes Frame check sequence(FCS) 32-bit cyclic redundancy check used to detect corruption in data.

Preamble(8B)	Dest. Address(6B)	Source Address(6B)	Type/Length(2B)	User Data(46-1500B)	FCS(4 B)

Network Layer :**IPv4** It is the backbone of TCP/IP suite. It is responsible for identifying hosts based on their logical address. It uses 32-bit address space to uniquely identify the hosts on network and makes best effort to deliver packets to destined host.Packet contains IP header , ICMP Header , ICMP payload, IP Packet Header consists of IP Version , IHL(IP header length),Total length, TTL(time to leave), Identification number, flags and offset, source and destination IP address. ICMP header contains message type, code and checksum. ICMP payload contains data from above layer



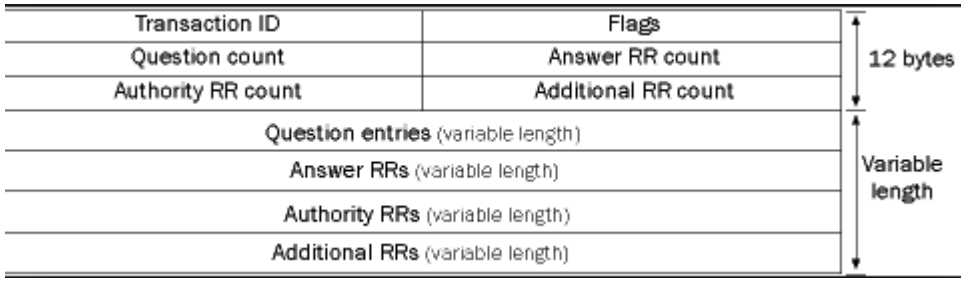
Transport Layer: TCP It provides the reliable, ordered and error-checked delivery of a stream of octets between application running on hosts communicating by an IP address. Its Segment consists of 2 parts : Segment Header and a data section. Header is shown in the figure. It consists of source port number, destination port number,sequence number,acknowlegment number, data offset, reserved bits, control flags, window size, checksum, urgent pointer and optional data. Data part cointains IP packet/data from above layer..



Transport Layer: UDP It is a message oriented protocol that provides no guarantee to upper layer protocol for message delivery and do not retain state of UDP once message is sent.Its packet contains(Header and Data).Its header contains 4 fields each of 2 Bytes, namely, Source port, Destination port, length and UDP checksum.

Source Port	Destination Port
Length	UDP Checksum
Data	

Application Layer: DNS DNS is a decentralized domain name resolving system for various services connected to WAN or LAN. It involves query asked by some client and IP resolved by DNS server. The process is recursive(i.e if one server doesnt know the answer it asks some other dns server as defined).In short



it provides a mapping from domain name to IP addresses.Packet format,as shown, conatins Transaction ID, Flags, Questions/Answer/Authorised nameserver/Additional info counts , Questions/Answer/Authorised nameserver/Additional info data.
Apart from this protocols very few other packets of other protocol were observed: **DCERPC(3/50k) and ESP(1/50k)** from the same IP address. But their purpose is unclear since they were not observed in large quantity and their occurrence was random.Also **NTP** packets were found but not always and source unknown.

Sol2.
(observed in raw_data5.pcapng)
1. Ethernet II : Data Link Layer

```
Ethernet II, Src: ChiconyE_2c:23:87 (4c:bb:58:2c:23:87), Dst: 66:db:43:51:c3:73
▸ Destination: 66:db:43:51:c3:73 (66:db:43:51:c3:73)
▸ Source: ChiconyE_2c:23:87 (4c:bb:58:2c:23:87)
  Type: IPv4 (0x0800)
```

Destination : Gives the 6 bytes MAC address of the destination NIC/device adapater.
Source : Gives the 6 bytes MAC address of the source NIC/device adapter.
Type : 2 byte field to indicate upper layer protocol to be used.

2. Internet Protocol Version 4(IPv4): Network Layer

```
Internet Protocol Version 4, Src: 192.168.43.10, Dst: 217.146.11.2
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 145
  Identification: 0x4d26 (19750)
▸ Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0x1cfa [validation disabled]
  [Header checksum status: Unverified]
  Source: 192.168.43.10
  Destination: 217.146.11.2
  [Source GeoIP: Unknown]
▸ [Destination GeoIP: AS42473 ANEXIA Internetdienstleistungs GmbH, India, Mumbai, 16, 18.975000, 72.825798]
```

Version : Version of IP protocol used.
Header Length : Size of the entire IP header.
Differentiated Services Field : Contains 2 fields. DSCP(Differentiated Services Code Point) packet header value that can be used to indicate the level of service requested for traffic, such as high priority or best effort delivery and ECN(Explicit Congestion Notification).
Total length: Size of IP packet(Header+ Payload).
Identification : packet number so that framented part can be uniquely identified.
Flags : Indicates that packet is fragmented or not.
Fragment Offset : if fragmented tells the poisition of frament in all fragments.
Time to live(TTL) : puts a limit on time(hops it passes) a packet can be valid/exists on network.
Protocol : Upper level protocol used in the payload.
Header Checksum : checks for error and attempts to correct the header
Source : IP address of my PC(device used for connection). Dynamic IP allocated by Hotspot.
Destination : IP address of the address of destination server.

3. UDP (User Datagram Protocol): **Transport layer**

```
User Datagram Protocol, Src Port: 53, Dst Port: 46895
  Source Port: 53
  Destination Port: 46895
  Length: 437
  Checksum: 0xcb4d [unverified]
  [Checksum Status: Unverified]
  [Stream index: 2]
```

Source Port : Port of the client/source used to connect to the reciever/dest.
Destination Port : Port of the destination/ reciever used for connection.
Length : Size of UDP datagram (Header + Payload)
Checksum : Used for error detection and correction

4. TCP(Transmission Control Protocol) : Transport Layer

```
Transmission Control Protocol, Src Port: 46606, Dst Port: 5938, Seq: 17725, Ack: 17944, Len: 93
   Source Port: 46606
   Destination Port: 5938
   [Stream index: 1]
   [TCP Segment Len: 93]
   Sequence number: 17725    (relative sequence number)
   [Next sequence number: 17818    (relative sequence number)]
   Acknowledgment number: 17944    (relative ack number)
   Header Length: 32 bytes
▸  Flags: 0x018 (PSH, ACK)
   Window size value: 584
   [Calculated window size: 74752]
   [Window size scaling factor: 128]
   Checksum: 0x7560 [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
▸  Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
▸  [SEQ/ACK analysis]
```

Source Port and Destination port as in UDP.
Sequence Number: Data from above layer is divided into IP packets, indicates relative sequence amonst all.
Ack Number : acknowedgment of the packet recieved at other end.(indicates Piggybacking).
Header Length : length of TCP Header.
Flags: Ack: Acknowledges received data and PSH push data immediately.
Window Size value: buffer space available for recieving packets.
Checksum : Error detection and correcting code.
Urgent Point : Point to report when reciever needs some imp. Info.
Options : Other imp info like timestamps etc. Used in tcp communication.

5. DNS(Domain Name System): Application layer

```
Domain Name System (query)
   [Response In: 11]
   Transaction ID: 0x8268
▸  Flags: 0x0100 Standard query
   Questions: 1
   Answer RRs: 0
   Authority RRs: 0
   Additional RRs: 0
▾  Queries
   ▸  ping3.teamviewer.com: type A, class IN
```

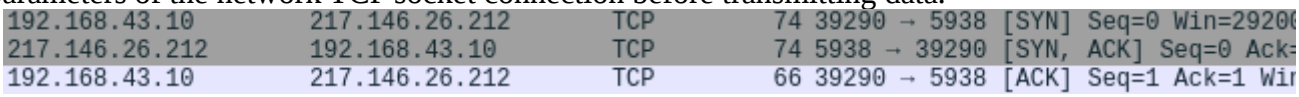
Transaction ID: 2 Byte identification field generated by device/application which created DNS query.
Flags: To indicate one of the following :query, response, opcode, recursion query needed, non-autenticated etc.
Questions/Answer/Authority/Additional: Number of queries/answer/autority records/additional records.
Queries/Answer/Authoritative nameservers/Additional records: Details/data related to repective values.

Sol3.
When the team viewer app is opened it sends DNS query for ping3.teamviewer.com(most probably to check whether a connection to team viewer server can be made) for both Ipv4 and Ipv6 results.

```
192.168.43.10       192.168.43.1        DNS        80 Standard query 0x1e75 A ping3.teamviewer.com
192.168.43.1        192.168.43.10       DNS       144 Standard query response 0x1e75 A ping3.teamviewer.com A
192.168.43.10       192.168.43.1        DNS        80 Standard query 0x198d AAAA ping3.teamviewer.com
192.168.43.1        192.168.43.10       DNS       192 Standard query response 0x198d AAAA ping3.teamviewer.com
```

This is followed by TCP 3-way handshake as shown. First client app sends SYN packet to host, host sends SYN-ACK to the client. Client replies with ACK. Once this ACK is received by the host , connection is established. This method is reffered to as TCP SYN-SYN-ACK handshaking method based on the sequence of

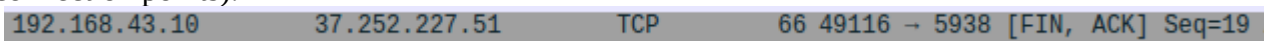
method used.This mechanism was designed so that two computers attempting to communicate can negotiate the parameters of the network TCP socket connection before transmitting data.



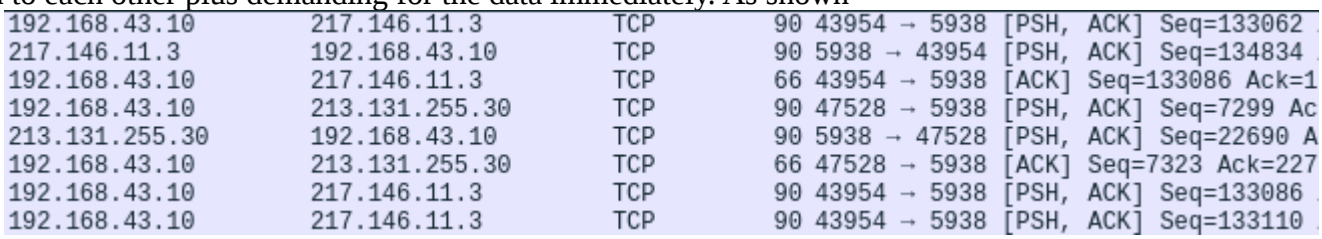
After this DNS query(Name: master6.teamviewer.com) is perfomed followed by TCP handshake(to register the current instance of the Team Viewer app). Again a query(Name: server19403.teamviewer.com , kept changing) is performed. After this TCP handshake is again performed and connection to fixed IP is established.(This is used as mapping if other peer wants to connect to this instance).

Along with the TCP handshake their is another imp message sequense that takes place i.e [PSH ACK] and [ACK]. Here PSH is an indication by the sender that, if the receiving machine's TCP implementation has not yet provided the data it's received to the code that's reading the data (program, or library used by a program), it should do so at that point.There is a coupling between the push function and the use of buffers of data that cross the TCP/user interface. Each time a PUSH flag is associated with data placed into the receiving user's buffer, the buffer is returned to the user for processing even if the buffer is not filled. If data arrives that fills the user's buffer before a PUSH is seen, the data is passed to the user in buffer size units.

The TCP handshake/stream is finished using the [FIN ACK] tcp packet.(probably it helps to free some connection points).



Also when another device is used to connect the TV instance using the ID, similar queries and handshakes take place in order to establish connection between the two peers. Once a connection is established, it is followed by large sequence (too many) of [PSH ACK] and [ACK] packets exhange in which indicates both end sending data to each other plus demanding for the data immediately. As shown



Sol4.

Importance of Protocols used by team viewer(wrt app itself):

TCP: Team Viewer involves fetching and transfering data from both ends live i.e continuous and quickly. Its bandwidth probing and congestion control will attempt to use all of the available bandwidth between peers transfering content as quick as possible while being friendly to other (TCP) traffic on the same link. This is the foremost requirement of an Remote Desktop sharing app. Also it allows prefetching of data helping in maitaining the smooth(nearly smooth) view of the shared desktop and continous input being given.

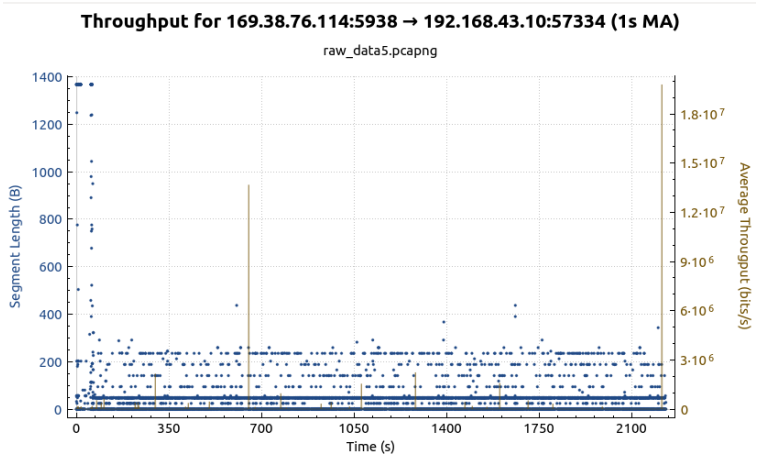
Also TCP is reliable, hence the frames tramitted are not lost preventing the ambigous beaviour in commands given by remote client or data send by shared desktop.

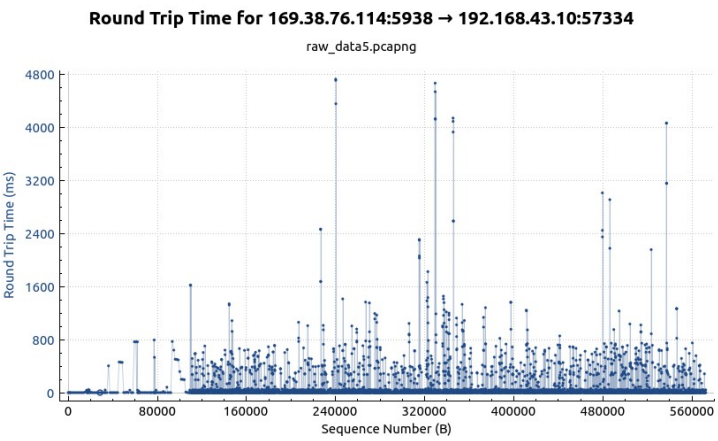
UDP: It is not used as such for data transfer since it is not relaible as it doesnt maintain the state of connection and doesnot guarantee or make attempts that packet will reach the destination. It is only used for DNS queries.

DNS: Used at various stage to resolve the domain name. Apart from this it also has a major advantage. Suppose the IP address of a server gets changed or server supports multiple ip mappings. Then all this Ips neednt be hardcoded in the client app only the domain name can be used which can be mapped to any one of the IPs available.

IPv4 : Used for logically determining the location of source and destination in the network. It also provides error detection and correction techniques.
Ethernet II : Used to encode the information about source and destination physical(i.e MAC) address. Also it contains information about error detection and correction.
(Also **NTP** was invoked at sometimes but the source invoking it is not clear. Maybe invoked by OS or TV app for time synchronization.)

Sol5. (graphs for raw_data5.pcapng has been shown here. Also the images are corresponding to this data. Rest values are given in tabular format)





Graph shows the **throughput and RTT starting from connection establishment,** then sequence of transaction between two peer devices.
Total packets and bytes captured and Average packet size.(shown below). Packets: 52352 , Avg. Packet Size: 153.5 Bytes.

Measurement	Captured	Displayed
Packets	52352	52352 (100.0%)
Time span, s	2303.396	2303.396
Average pps	22.7	22.7
Average packet size, B	153.5	153.5
Bytes	8045562	8045562 (100.0%)
Average bytes/s	3492	3492
Average bits/s	27 k	27 k

Packet Loss: 600(1.1%) shown below

Packets: 52352 · Displayed: 600 (1.1%)

No. of UDP and TCP packets are shown in the figure

Protocol Name	Percent packet	Packets
▸ User Datagram Protocol	0.6	326
▸ Transmission Control Protocol	98.9	51789

Sample Number	Sample1 (raw_data5)	Sample2(raw_data3	Sample3 (raw_data4)
Time of the day	**3 pm**	**4 am**	**6 am**
Host A	192.168.43.10:57334	192.168.43.10:46606	192.168.43.10:43954
Host B	169.38.76.114:5938	217.146.11.2:5938	217.146.11.3:5938
Throughput from A to B (bps)	21 k	46 k	7163
Throughput from B to A (bps)	730 k	12 k	8250
Avg. packet size (bytes)	153.5	509.5	831.59

RTT(ms)	83	91	108
No. of Packets lost	600	1	0
No. of TCP packets from A to B	30.497	1575	402
No. of TCP packets from B to A	20,973	1264	317
No. of UDP packets	331(none between A and B)	73 (none between A and B)	24 (none between A and B)
No .of Responses/request	0.67(because more data is sent then recieved)	1.25	1.27

Sol6. Multiple IPs were detected to be used as shown in the following images.(Infact almost every new connection tried resulted in the new IP address).But the IP address didnt changed once a connection is established. Changing IP could be because of the various possible servers used by the Team Viewer App, which might be changing with the time of day or every new route being establsihed to Server. In P2P connection the shorter route that can be esablished between 2 peers more faster will the data transfer.Also large number of routes/server are needed for faster, countinous and large data transfer reasoning the change at every new connection.Remote Desktop Control app requires transfer of live data and commands which requires large and contigous transfer of data, also there can be many connection (Some IPs are:
213.131.255.40,
217.146.11.3,
217.146.11.2 etc.)

Images showing various IP used (in differnet connection attempts).