CS349 ASSIGNMENT 2 (P2P- Teamviewer)

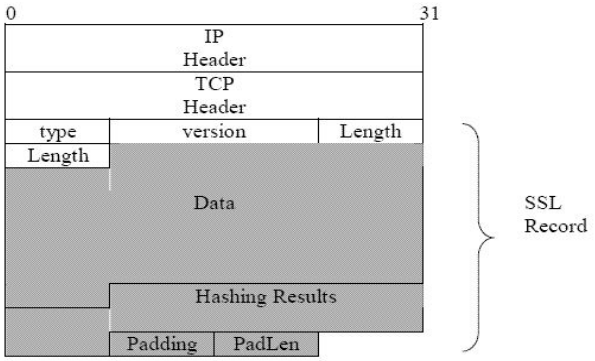
ARPIT GUPTA(160101015)

Wireshark traces: https://drive.google.com/open?id=1TRVqln-SqiQNmAk4Lsqg6qEIpvC8s0SH

1. PACKET FORMATS OF PROTOCOLS USED IN DIFFERENT LAYERS

A. Application layer

a.) HTTP - 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 have applicability for both request and response messages, Client Request-header have applicability only for request messages, Server Response-header have applicability only for response messages,Entity-header defines meta information about the entity-body. The header fields are - Connection general-header allows the sender to specify options that are desired for that particular connection and must not be communicated by proxies over further connections,Date, Authorization request-header field value consists of credentials containing the authentication information of the user agent,Cookie request-header field value contains a name/value pair of information stored for that URL, From request-header contains an Internet e-mail address for the human user who controls the requesting user agent, Host request-header is used to specify the Internet host and the port number of the resource being requested, Proxy-Authorization request-header allows the client to identify itself to a proxy which requires authentication, User-Agent request-header contains information about the user agent, Location response-header is used to redirect the recipient to a location other than the Request-URI for completion, Server response-header contains information about the software used by the origin server to handle the request.



b.) SSL/SSLv2 - SSL provides security in the communication between two hosts. It provides integrity, authentication and confidentiality. It can be used with any protocol that uses TCP as the transport layer. The basic unit of data in SSL is a record. Each record consists of a five-byte record header, followed by data. The header contains – Record Type can be of four types(Handshake,Change Cipher Spec, Alert, Application Data), Record Version is 16-byte value formatted in network order, Record Length is 16-byte value.

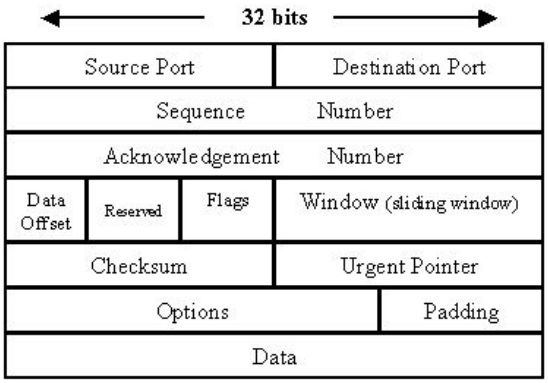
c.) Data - When Wireshark can't determine how part of a packet should be formatted, it marks that chunk as "Data". The "Data" is just the normal data payload.

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

Figure 4-8. TLS record structure

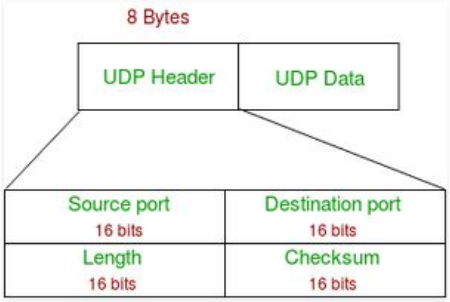
d.) TLSv1.3: Not unlike the IP or TCP layers below it, all data exchanged within a TLS session is also framed using a well-defined protocol.The TLS Record protocol is responsible for identifying different types of messages (handshake, alert, or data via the "Content Type" field), as well as securing and verifying the integrity of each message.Maximum TLS record size is 16 KB.Each record contains a 5-byte header, a MAC,and padding if a block cipher is used.To decrypt and verify the record, the entire record must be available.

B. Transport Layer



a.) TCP : Each TCP header has ten required fields totaling 20 bytes in size. They can also optionally include an additional data section up to 40 bytes in size. TCP headers has - Source and destination TCP ports which are the communication endpoints for sending and receiving devices, sequence and acknowledgement numbers to mark the ordering in a group of messages, data offset stores the total size of a TCP header in multiples of four bytes, Reserved data in TCP headers always has a value of zero, a set of six standard and three extended control flags (each an individual bit representing on or off) to manage data flow in specific situations, window size to regulate how much data sender sends to a receiver before requiring an acknowledgment in return, checksum for error detection, urgent pointer can be used as a data offset to mark a subset of a message which require priority processing. Optional TCP data can be used to include

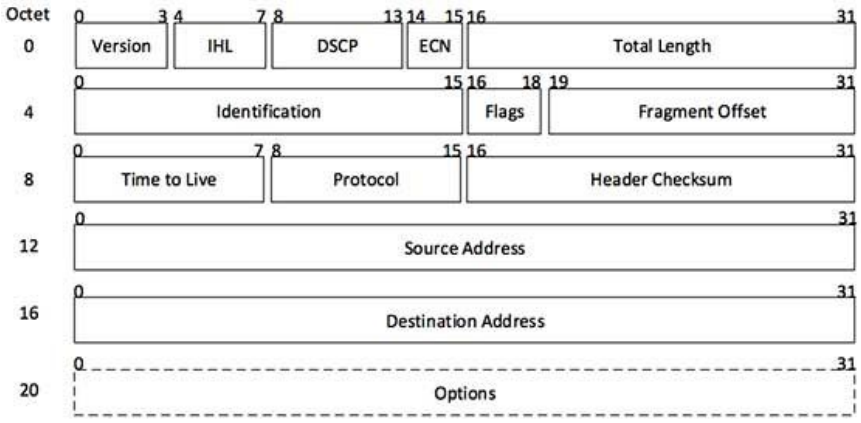
support for special acknowledgment and window scaling algorithms.Figure on the left shows a typical TCP packet header.



b.) UDP : UDP header is 8-bytes fixed and simple header.First 8 Bytes contains all necessary header information and remaining part consist of data. UDP port number fields are each 16 bits long, therefore range for port numbers defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or process.

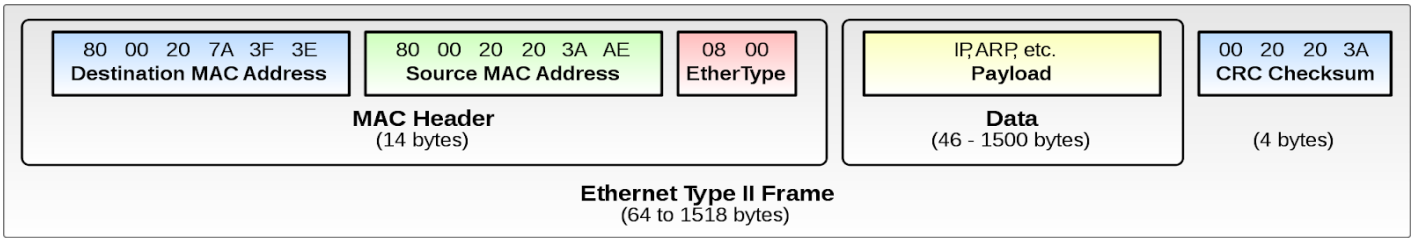
C. Network Layer

IPv4 is one of the core protocols of standards-based internetworking methods in the Internet. It is a connectionless protocol for use on packet-switched networks. The header consists of 14 fields, of which 13 are required. They are – Version is always equal to 4,Internet Header Length (IHL) has 4 bits which is the number of 32-bit words in header, Differentiated Services Code Point (DSCP) used in QoS, Explicit Congestion Notification (ECN) allows end-to-end notification of network congestion without dropping packets, Total Length is 16-bit field which defines the entire packet size in bytes, identification field is primarily used for uniquely identifying the group fragments of a single IP datagram, flags bit is used to control or identify fragments (0 th bit:Reserved and is always 0; 1 st bit: Don't Fragment (DF); 2 nd bit: More Fragments (MF)), Fragment Offset specifies the offset of a particular fragment, Time To Live (TTL) helps prevent datagrams from persisting on network forever,Protocol defines the protocol used in portion of the IP datagram, Header Checksum is used for error-checking of the header, Source address & Destination address is the IPv4 address of the sender and receiver of the packet respectively.



[Image: IP Header]

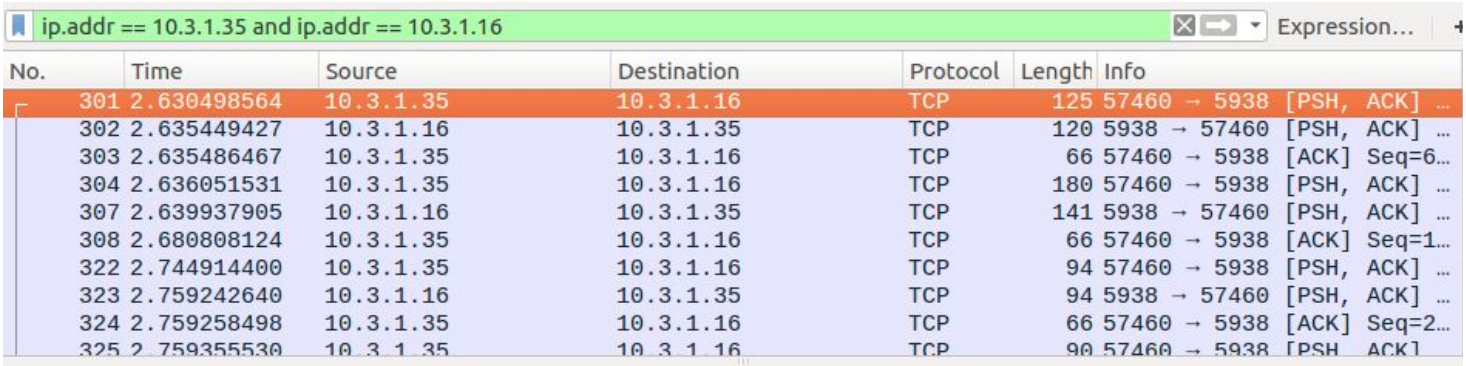
D. Link Layer



Ethernet(II) is the most common local area networking technology. It has Preamble which is 56 bits of alternating 1's and 0's, Destination MAC Address, Source MAC Address, Type that identifies an upper layer protocol encapsulated by the frame data, Length of frame and Frame Checksum for error detection.

2. EXPERIMENTAL OBSERVATIONS

When using Teamviewer through LAN, the following stream of packets were observed :-



When using Teamviewer through internet(IITG), the following stream of packets were observed :-

```
32312 138.988666140 10.3.1.35      202.141.80.20    TCP      1490 [TCP segment of a reassembled PDU]
32313 138.988839652 10.3.1.35      202.141.80.20    TCP      1514 [TCP segment of a reassembled PDU]
32314 138.988847021 10.3.1.35      202.141.80.20    TCP      1514 [TCP segment of a reassembled PDU]
32315 138.988891343 10.3.1.35      202.141.80.20    SSLv2    1514 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Data,…
32316 138.988907665 10.3.1.35      202.141.80.20    SSLv2    1514 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Data,…
32317 138.989098159 10.3.1.35      202.141.80.20    SSLv2    1514 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Data,…
32318 138.989232367 10.3.1.35      202.141.80.20    SSLv2    1514 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Data,…
32319 138.989310444 10.3.1.35      202.141.80.20    SSLv2    1514 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Data,…
32320 138.989479040 10.3.1.35      202.141.80.20    SSLv2    1514 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Data,…
32321 138.989560801 10.3.1.35      202.141.80.20    SSLv2    1298 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Data,…
32322 138.989819379 202.141.80.20  10.3.1.35        TCP        66 3128 → 40692 [ACK] Seq=21409 Ack=5506 Win=28288 Len=0 TSval=356…
32323 138.989839264 202.141.80.20  10.3.1.35        TCP        66 3128 → 40692 [ACK] Seq=21409 Ack=6954 Win=31104 Len=0 TSval=356…
```

In the both the above, time is the time elapsed since the starting of packets capture, source & destination are the sender & the reciever of the packets respectively, protocol is the protocol(of the highest layer) that the wireshark could identify, length is the length of the packet and info is a brief information contained in the packet decoded by wireshark.

- **SSLv2** – The HTTP part of the packet contains the hostname and the port, while the SSL part contains the Encrypted Data.(Note that both HTTP and SSLv2 are in application layer). WireShark is unable to determine the further headers of the SSL packet. It just identifies the version– 2 as seen in the following figure.

```
▼ Hypertext Transfer Protocol
    [Proxy-Connect-Hostname: IN-DEL-ANX-R003.teamviewer.com]
    [Proxy-Connect-Port: 443]
▼ Secure Sockets Layer
  ▼ SSLv2 Record Layer: Encrypted Data
      [Version: SSL 2.0 (0x0002)]
```

- **HTTP** – As seen in the figure, the Request method is connect to the request URI, which is the address of the teamviewer server. The version of HTTP used is 1.1. Since the connection is through IITG Proxy, Proxy Authorisation is also added. User agent is Mozilla by default in Wireshark. Wireshark also shows the previous request frame number and the frame number having response to this frame.

```
▼ Hypertext Transfer Protocol
  ▶ CONNECT IN-DEL-ANX-R003.teamviewer.com:443 HTTP/1.1\r\n
    Host: IN-DEL-ANX-R003.teamviewer.com:443\r\n
  ▶ Proxy-Authorization: Basic [redacted]
    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; DynGate)\r\n
    Proxy-Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: IN-DEL-ANX-R003.teamviewer.com:443]
    [HTTP request 2/2]
    [Prev request in frame: 32189]
    [Response in frame: 32215]
```

- **TCP** – The packet contains the Destination and the Source Port, TCP Stream index, sequence number and the acknowledgement number,Header length, Flags. In the following figure, the flags are set as 018 in HexaDecimal which corresponds to the Acknowledgement. Window size value is 296. Checksum is used for error detection. Wireshark is remembering the value of Window size scaling factor and presenting it again. Scaling factor shows the number of leftward bit shifts that should be used for an advertised window size.

```
▼ Transmission Control Protocol, Src Port: 40692, Dst Port: 3128, Seq: 185, Ack: 3739, Len: 241
    Source Port: 40692
    Destination Port: 3128
    [Stream index: 28]
    [TCP Segment Len: 241]
    Sequence number: 185    (relative sequence number)
    [Next sequence number: 426    (relative sequence number)]
    Acknowledgment number: 3739    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x018 (PSH, ACK)
    Window size value: 296
    [Calculated window size: 37888]
    [Window size scaling factor: 128]
    Checksum: 0x1a47 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]
    TCP payload (241 bytes)
```

- **IPv4** – Version header field is always 4 as we are using IPv4. Header length is 5 (which means 20 bytes because it counts in 4 Bytes word).Total length of the packet is 52 bytes. The flag set is don't fragment which instructs all the nodes

through which the packet passes to not fragment the packet. TTL is 63. The protocol contained in it is TCP. The packet is sent by the proxy(202.141.80.20) to my device(10.3.1.35).

```
▼ Internet Protocol Version 4, Src: 202.141.80.20, Dst: 10.3.1.35
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0xec00 (60416)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 63
    Protocol: TCP (6)
    Header checksum: 0x29fc [validation disabled]
    [Header checksum status: Unverified]
    Source: 202.141.80.20
    Destination: 10.3.1.35
```

- **Ethernet II** – It contains the physical MAC address of the devices communicating. Destination is my HP Device and the source is the Switch to which my device is connected. Source is always Unicast. Destination is Unicast in this case. In both of them, it is Globally Unique Adress and not a Local Address.

```
▼ Ethernet II, Src: Cisco_97:1e:ef (4c:4e:35:97:1e:ef), Dst: HewlettP_4c:86:ef (a0:8c:fd:4c:86:ef)
  ▼ Destination: HewlettP_4c:86:ef (a0:8c:fd:4c:86:ef)
      Address: HewlettP_4c:86:ef (a0:8c:fd:4c:86:ef)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▼ Source: Cisco_97:1e:ef (4c:4e:35:97:1e:ef)
      Address: Cisco_97:1e:ef (4c:4e:35:97:1e:ef)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

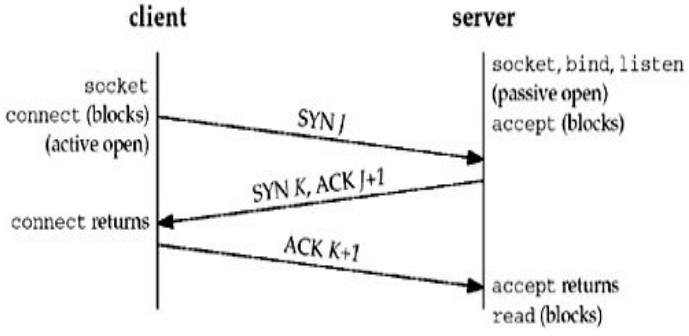
3. SEQUENCE OF MESSAGES

The application was tested under different conditions and the behaviour was observed in each. TeamViewer checks network connections to see what protocol it can efficiently deploy to make the communication between two peer computers efficient. If both the computers sit in the same LAN, there is no need to route data through HTTPS/SSL ports. But if the data is being sent through Teamviewer server over the internet, then the data is encrypted and sent through HTTPS/SSL ports so that not even the people at teamviewer can decipher and look at data.

The behaviour observed was similar when the two devices were connected directly through LAN and when connected through a switch. In both these scenarios, only TCP/IPv4 protocol packets were observed. The TCP three-way handshake (SYN SYN-ACK ACK) is the method used by TCP to set up a TCP/IP connection between the devices. After the connection setup, the devices exchange sequences of data and acknowledgements.The wireshark couldn't determine the application layer protocol and hence it states it as 'Data' protocol. During termination of the connection, another handshake is used(FIN, ACK, FIN, ACK). In both these cases, the source and destination of the packets either of the source or destination addresses and not any intermediate address.

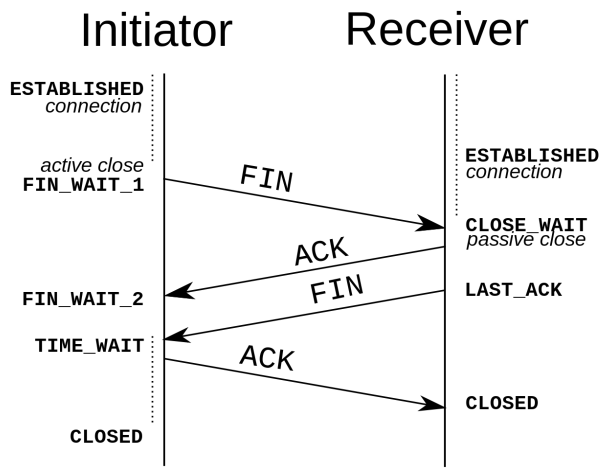
The behaviour was a little different when the two devices were connected through internet using teamviewer. Primarily, all the packets were being routed through the internet, so for both the devices the packets were being sent to and received from the proxy server(202.141.80.20). Here also, the TCP connection setup handshake and the termination handshake were observed. After the TCP connection, an HTTP CONNECT packet is sent to the teamviewer server and the HTTP connection is established.Unlike the previous scenario, here the data is sent using SSL (application layer) protocol. Hence, it is encrypted. All the acknowledgements are not encrypted and hence wireshark displays their protocol as TCP only. Finally on connection termination the TCP termination handshake occurs. Note that in this case, teamviewer.com does handshaking with regular TCP packets. But transmits data with SSL(encrypted) which is an application protocol which sits over TCP. SSL transmission is interlaced with plain TCP transmission with same source/destination IP pairs connected to the same ports. This indicates that the data is being transferred with SSL and handshake is happening with TCP.

Handshaking connections:



a.) TCP CONNECTION HANDSHAKE : To establish a connection, each device must send a SYN and receive an ACK for it from the other device. Thus, conceptually, we need to have four control messages pass between the devices. However, it's inefficient to send a SYN and an ACK in separate messages when one could communicate both simultaneously. Thus, in the normal sequence of events in connection establishment, one of the SYNs and one of the ACKs is sent together by setting both of the relevant bits (a message sometimes called a SYN+ACK). This makes a total of

three messages, and for this reason the connection procedure is called a three-way handshake.



b.) TCP TERMINATION HANDSHAKE : In the normal case, each side terminates its end of the connection by sending a special message with the FIN (finish) bit set. This message serves as a connection termination request to the other device. The device receiving the FIN responds with an acknowledgment to the FIN to indicate that it was received. The connection as a whole is not considered terminated until both sides have finished the shut down procedure by sending a FIN and receiving an ACK. Thus, termination isn't a three-way handshake like establishment: it is a pair of two-way handshakes. The states that the two devices in the connection move through during a normal connection shutdown are different because the device initiating the shutdown must behave differently than the one that receives the termination request. In particular, the TCP on the device receiving the initial termination request must inform its application process and wait for a signal that the process is ready to proceed. The initiating device doesn't need to do this, since the application is what started the ball rolling in the first place.

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

TCP: TeamViewer involves fetching and transferring 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 transferring 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 maintaining the smooth(nearly smooth) view of the shared desktop and continuous input being given.Also TCP is reliable, hence the frames transmitted are not lost preventing the ambiguous behaviour 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 reliable as it doesn't maintain the state of connection and does not guarantee or make attempts that packet will reach the destination. It is only used for DNS queries as observed in experiments. During the connection and termination phase, TCP handshake occurs as explained previously.Once connection is established for data transmission TCP is used. According to the teamviewer's site, once connection is established, the data exchange may use TCP or UDP depending on the requirements and conditions. But, I observed only TCP packets. I believe that this is so because IITG-Proxy server blocks UDP packets.

SSL: Since when connecting through internet, we want to avoid hackers snooping packets on wire, teamviewer goes for something like SSL.Application's requirement determines the selection of protocol. It is used so that even the people at the teamviewer's server are unable to view the data. Hence, SSL is used so that encrypted data is transferred.

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 needn't 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 TeamViewer app for time synchronization.)

5. STATISTICAL ANALYSIS

Time	Throughput (bytes/sec)	RTT (in ms)	Avg. packet size(Bytes)	No. of packets lost	UDP packets	TCP packets	Avg. Response wrt 1 request
6.00 pm (over lan)	9627	12.5	556	0	0	2953	0.6553
7.00 pm (iitg internet)	67k	16.24	642	0	0	105960 (run for	0.7238

						22 minutes)	
12.00am (wi-fi internet)	8015	20.06	379	0	52	4693	0.7258

6. CONTENT PROVIDERS

Teamviewer is a Peer-To-Peer Remote Desktop application. Hence, all the data is exchanged between two peers. TeamViewer site doesn't provide any content. As stated in earlier parts, **if trying to connect through LAN**, teamviewer application connects the devices directly through local LAN path. Hence, data is coming from only one IP and not multiple IP.

If the peers try to connect through IITG Internet, then the packets are sent through teamviewer servers, it routes the peer-peer data using standard HTTPS and SSL ports. In our case, all the traffic goes through IIT-Guwahati proxy server 202.140.80.20. Proxy server does NAT (Network Address Translation) because of which we only see proxy IP address in the packets we capture.

And while using connection through wifi hotspot internet,Multiple IPs were detected to be used.(Infact almost every new connection tried resulted in the new IP address).But the IP address didn't 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 established to Server. In P2P connection the shorter route that can be established between 2 peers more faster will the data transfer.Also large number of routes/server are needed for faster, continuous 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 contiguous transfer of data, also there can be many connection.Some such IPs found are:

- 213.227.184.134
- 188.172.208.137
- 223.187.184.146