Name – Shashank Garewal

Roll no. - 150101067

Q1. Protocol used by vimeo application, while uploading and downloading.

OSI layer	Name	Protocol used
7 5	Application Session	HTTP TLSP
4	Transport	TCP
3	Network	IP or IPv4
2	Data-link	Ethernet II

There're other protocol too came while capturing but those isn't for my functionality of application.

Ethernet II protocol

Destination MAC address	**Source MAC address**	**EtherType**	**Payload**	**Frame Check Sequence**
6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes

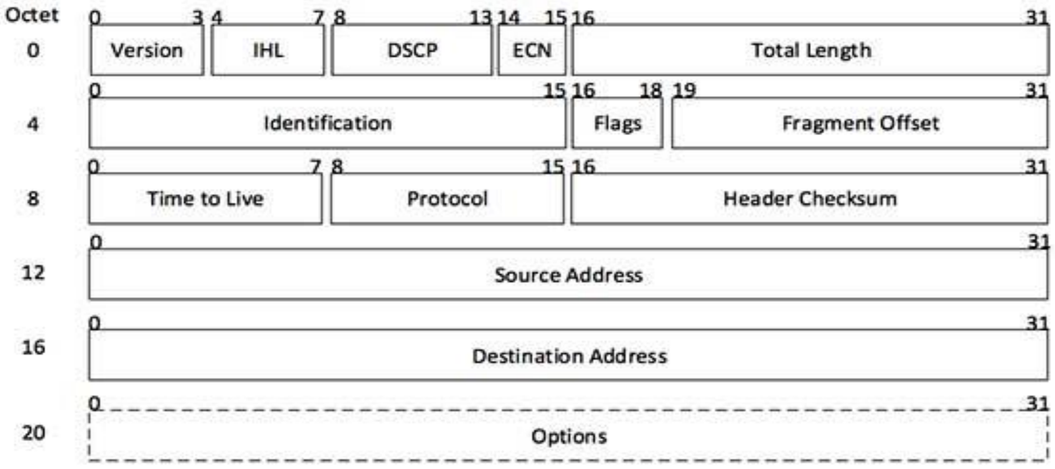
Source and **destination MAC address's :**original sender and final destination of packet.

EtherType : 2 bytes value which is used to determine Network protocol it have, like for Ethernet II, its 0x0800.

Payload – It's data which is from 46 – 1500 Bytes

FCQ or CRC **–** When the voltage on the wire returns to zero, the adapter checks the **last 4 bytes** it received against a checksum that it generates via a complex polynomial. If the calculated checksum != checksum on the frame, the frame is discarded and never reaches the memory buffers in the station.

IPv4 packet header consists of 20 bytes of data.



[Image: IP Header]

Version : 4, Version no. of Internet Protocol;

IP header length : 32-bit form word, they're usually five;

DSCP: Differentiated Services Code Point, 6 bit field informs a router how to queue packets while they are waiting to be forwarded, representing quality of service.

ECN: Explicit Congestion Notification, 2 bits

00 indicates the packet does not use ECN.

01 indicates the packet is a part of an ECN-capable transport flow.

10 indicates the packet is a part of an experimental ECN-capable transport flow.

11 indicates the packet has experienced congestion.

Size of Diagram : length of header + data (bytes);

Identification + **source address** identify packets;

Flags : control whether routers are allowed to fragment a packet.

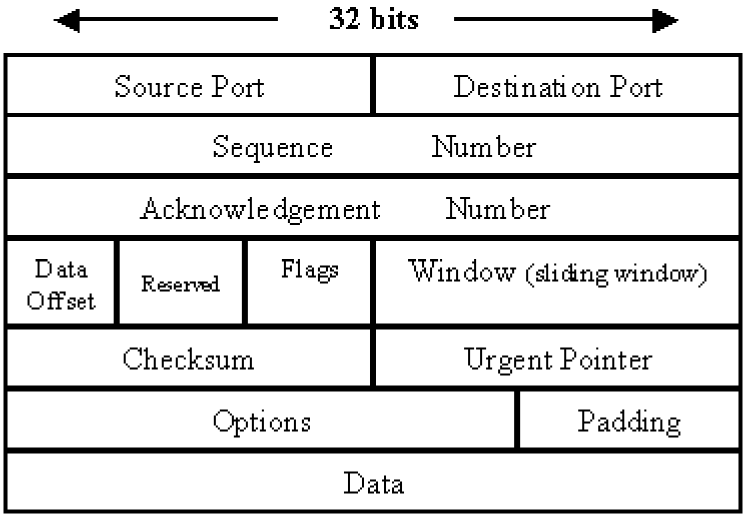
Fragmentation offset is count of from the start of the original sent packet, set by any router which performs fragmentation.

Time to live is number of hops a packet can live.

Protocol : indicate type of protocol its carrying out through transport layer.

Options are not generally used, but when its used, it increases IP header length.

The **TCP** packet format consists of these fields:



Source Port and **Destination Port fields** (16 bits each) identify the end points of the connection.

Sequence Number field (32 bits) specifies the number assigned to the first byte of data in the current message. Under certain circumstances, it can also be used to identify an initial sequence number to be used in the upcoming transmission.

Acknowledgement Number field (32 bits) contains the value of the next sequence number that the sender of the segment is expecting to receive, if the ACK control bit is set. Note that the sequence number refers to the stream flowing in the same direction as the segment, while the acknowledgement number refers to the stream flowing in the opposite direction from the segment.

Data Offset (a.k.a. Header Length) field (variable length) tells how many 32-bit words are contained in the TCP header. This information is needed because the Options field has variable length, so the header length is variable too.

Reserved field (6 bits) must be zero. This is for future use.

Flags field (6 bits) contains the various flags:

URG—Indicates that some urgent data has been placed.

ACK—Indicates that acknowledgement number is valid.

PSH—Indicates that data should be passed to the application as soon as possible.

RST—Resets the connection.

SYN—Synchronizes sequence numbers to initiate a connection.

FIN—Means that the sender of the flag has finished sending data.

Window field (16 bits) specifies the size of the sender's receive window (that is, buffer space available for incoming data).

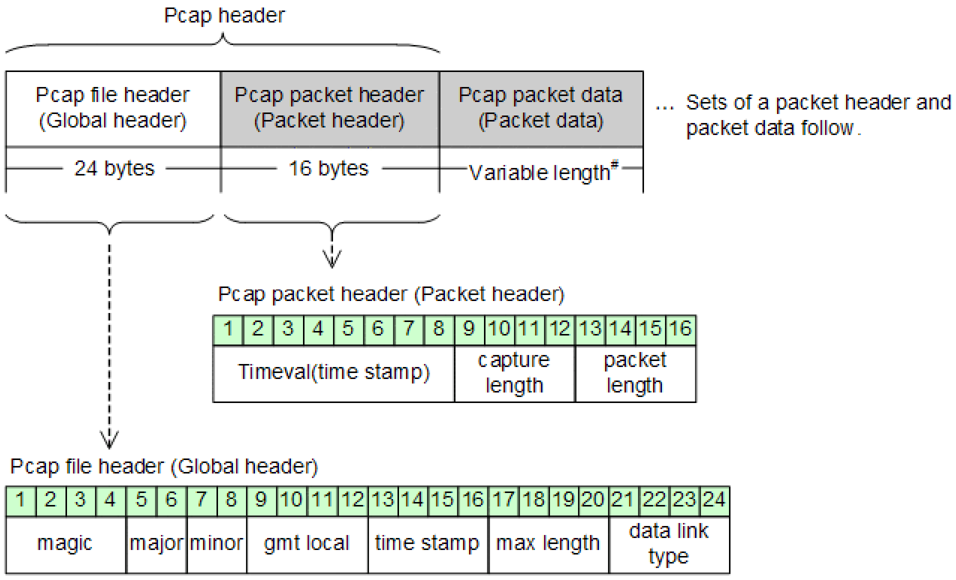
Checksum field (16 bits) indicates whether the header was damaged in transit.

Urgent pointer field (16 bits) points to the first urgent data byte in the packet.

Options field (variable length) specifies various TCP options.

Data field (variable length) contains upper-layer information.

The **HTTP** packet can't be directly input, a packet analyzer used which output Pcap Packet format.



In Pcap packet format, which contain Pcap header and Pcap data, where Pcap packet header creates Pcap global header in analyzer.

Q2. HTTP used in download functionality of vimeo-

In #354 packet, File Data: 1313 bytes- is amount of data does that http packet contains, Data: 0b0d788d...... - which is actual data that packet contains, and [length]: 1313bytes which represent length of data.

TCP Protocol used in download functionality of vimeo-

I have used room/12pm_1MB capture for both download and upload:

In #92 packet, a packet is coming from server, here source port - 3128 which use to transmit data to destination port – 44784 (my machine) both are epheremal port as they are above 1023 but as we're in proxy port-3128 is set for proxy server and my machine port is set randomly, SEQ no. - 4345 sequence of the tcp, this ensure that part of the data stream isn't missing from the entire packet, whereas ACK no. - 163 represent SEQ no. for next packet. After that flags – 0x010 ACK:set = 1 that says this packet include ACK, except that every flag is 0, which means its not a finishing neither a syncronize neither packet. Window size – 164 is size of the TCP receiver buffer, after that checksum - 0x8003 make sure data is actually intact and no error and legit after that there's urg – 0, urgent flag which says there's no urgent flag in packet which means no extra instruction like where to begin reading data in packet and other options. Len- 1448 is size of payload. TSval – 813064592, Timestamp Value field contains the current value of the timestamp clock of the TCP sending the option & TSecr – 821277168, Timestamp Echo Reply field echos a timestamp value that was sent by the remote TCP.

In #71 packet, contains a push flag PSH - 1 is used to inform the receiver that the sender has no further data to transmit.

#202 is blue greyish in background-color and the one with similar color codding means error occur while capturing the packet and the corresponding ACK to server is send in next packet I.e, #203.

#205 duplicate ACK packet, and it retransmit that packet for which it got all this duplicate ACK from.

IPv4 protocol for download functionality of vimeo-

In #6 Packet Src:10.3.1.31 is source IP address, Dst: 202.141.80.24 is destination IP address, version: 4 – IP version that is used(IPv4), Header length: 20 bytes – header length of IPv4 its mostly 20 bytes, Total Length: 52 – length of header + data, Identification: 1579 – number to identify actual packet, Flags: Don't fragment = 1 means no other sequence of packet left. Time to live: 64, 64 hops it can travel per sec which could help if packet goes in never ending loop, Protocol: TCP, protocol for #6 IP packet is TCP(protocol of below layer), Header checksum – to identify error, then again source and destination IP address, and at bottom shows Geo-logical location of source then destination which in #6 packet is for destination only.

Ethernet II Protocol for download functionality of vimeo- which contains MAC addresses of source and destination as in #6 packet, Src: 94:57:a5:db:ad:b8 and Dst: 4c:4e:35:97:1e:ef.

HTTP used in upload functionality of vimeo-

In #35 packet, Request Method: CONNECT- trying to connect the web directory,

Request URI: u.cloud.vimeo.com:443 - u.cloud.vimeo.com with port 443,

https. Request Version: HTTP/1.1 - http version 1.1

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:58.0) Gecko/20100101 Firefox/58.0\r\n - says what all kind of information my browser support and accept.

Proxy-Connection: keep-alive\r\n - it’s a proxy connection and keep it active, and looking to connect with proxy,

Proxy-Authorization: Basic cy5nYXJld2FsOmtva2IsYTEyMw==\r\n - where it got proxy authentication and bellow that you have Credentials: xxxxxxxxx:xxxxxxxx 😶😶.

#40 packet, HTTP/1.1 200 Connection established\r\n - 200 means connection established for request made in #35 packet.

TLSP used in upload functionality of vimeo-

In #42 packet, Version: TLS 1.0 (0x0301) - TLS 1.0 version and Length: 512 – length of data packet.

Inside Handshaking protocol, Random Bytes: be0e51ba1..... - can be use to create master key.

Cipher Suites (15 suites) - 1 of the 15 suites will be use to communicate with computer.

Signature Hash Algorithms (11 algorithms) - 11 different algorithm is made by server(u.cloud.vimeo.com),

In #51 packet, Handshake Type: Server Hello (2), random bytes again use to create master key, Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) - server picked that suite from suites list, then server send certification to source.

Session ID: c3944..... - session ticket and Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) - suite used and Handshake Protocol: Encrypted Handshake Message – happens.

In #53 packet, Encrypted Application Data: 000000000000000170df3e94....... - our credential which are shown as encrypted.

Q3. TCP Protocol – iam using room/12pm_1MB capture for both:

For download : #1 srl no., each side of session starts with relative SEQ=1 & ACK=1 from diagram, and server with #4 srl no. responds to the client with SEQ=1 & ACK=2, which indicate receipt of client's ACK flag. In #11 srl. no.is that first packet in the stream that carries actual payload with length 911 (download button data), thereby #12 srl. no.by server it send ACK=1 flag, and with #13 packet it send actual stream data(part of download), #14 send receipt for packet transmitted. Similarly for #15 to #18. #19-20 , #21-22 and so on, #18,20,22..... are packets in stream which contain payload(the client's HTTP request) - #19,21,23.... are receipt to upper layers process the HTTP request.

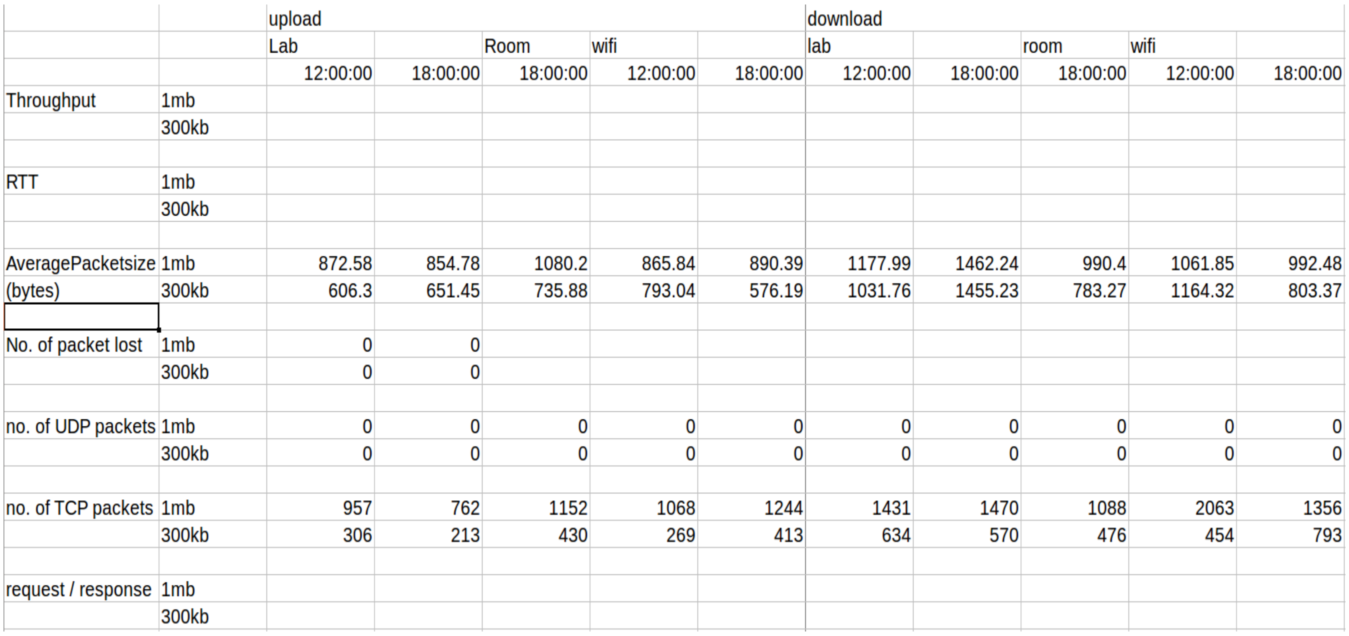
For Upload : #1 srl no., each side of session starts with relative SEQ=1 & ACK=1, #1 is first actual payload & #3 is the receipt for it with payload length=80. #1 and #2 are sent to service without looking for ACK, similarly for #5-6,#7-9,#8-10. **#11-13 is a 3-way handshaking**, #11 each side of a TCP session starts out with a (relative) sequence number of zero. #12 server responds to the client with a sequence number of zero, ACK=1 indicate receipt to the client SYN flag in #11. #13 client includes its own sequence number of 1 (incremented from zero because of the SYN).At this point, the sequence number for both hosts is 1. #14-15 normal payload transfer and receipt work.

Q4. HTTP is relevant in upload and download in vimeo, as its support proxy and it can uses dynamic ports of communication, it also supports pipelining, means client can ask for the next transfer already before the previous one has ended, which allows for multiple files to send or receive without round-trip delay. In upload, pieces of data forward to other layers they'll transmit it as bottom layer protocol says and it can use persisting path for transmission, and it also have some good compressing algorithm which helping in making it faster.

TCP is relevant in upload and download in vimeo, as its have seamless handshaking method for transmitting packets, plus it also have good error handling and failure recovery, so you don't need to re-upload or re-download whole media files while having error, low data overhead which leads to faster transfer as more data can be send in 1 packet.

TLSP is relevant in uploading and download in vimeo, as it have 2 type of video format- one is mature content and another is normal. So they need a credential from user so that they can encrypt mature content from normal content, and also for encrypting video while uploading as they're mature one or normal.

Q5. All the data are recorded which are directed to/from vimeo while upload/download function.



		upload					download				
		Lab		Room	wifi		lab		room	wifi	
		12:00:00	18:00:00	18:00:00	12:00:00	18:00:00	12:00:00	18:00:00	18:00:00	12:00:00	18:00:00
Throughput	1mb										
	300kb										
RTT	1mb										
	300kb										
AveragePacketsize	1mb	872.58	854.78	1080.2	865.84	890.39	1177.99	1462.24	990.4	1061.85	992.48
(bytes)	300kb	606.3	651.45	735.88	793.04	576.19	1031.76	1455.23	783.27	1164.32	803.37
No. of packet lost	1mb	0	0								
	300kb	0	0								
no. of UDP packets	1mb	0	0	0	0	0	0	0	0	0	0
	300kb	0	0	0	0	0	0	0	0	0	0
no. of TCP packets	1mb	957	762	1152	1068	1244	1431	1470	1088	2063	1356
	300kb	306	213	430	269	413	634	570	476	454	793
request / response	1mb										
	300kb										

Q6. Yeah, from my observation whole content for vimeo functionality (upload/download) connect with IITG proxy server i.e., 202.141.80.24 with port:3128 whenever used by protocol for communication.

All traces are in drive link

****drive link****