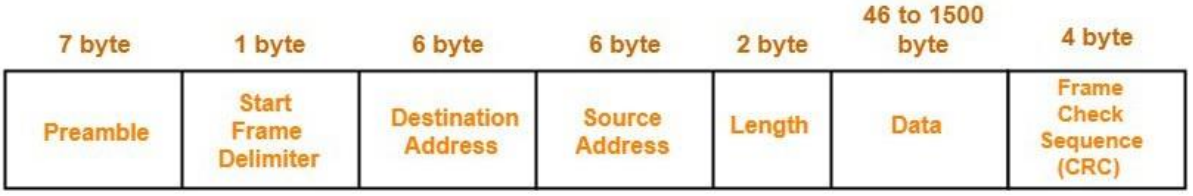
CS 342 Computer Networks Lab - Assignment 2

Group : 34

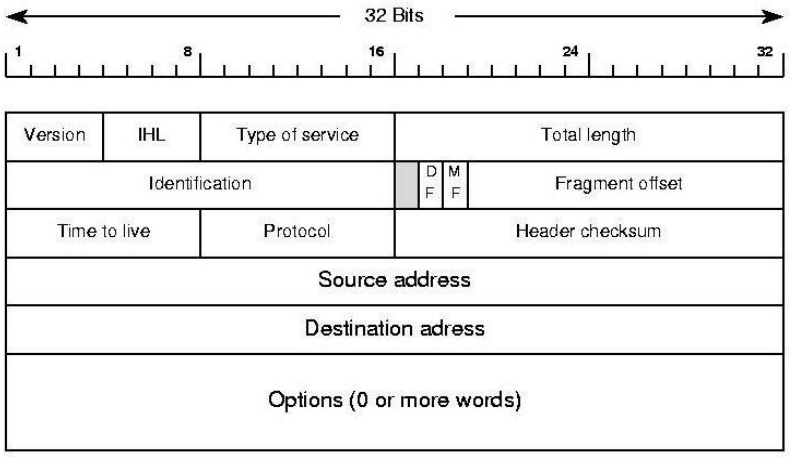
- Application - Online Game
- Game Played : Ball in the Hole
- PC was connected to Hoxx VPN (England) while taking 2 of the readings.
- Traces - TaceLink

Ans 1. List of Protocols Used at Different Layers: The various protocols used by the online game website for various layers were -

Link Layer (Ethernet) - Ethernet frame starts with 7-Bytes Preamble. This is pattern of alternative 0's and 1's which indicates starting of the frame and allow sender and receiver to establish bit synchronization. SFD is a 1-Byte field which is always set to 10101011 and signifies the start of frame. The destination and source address are 6-byte fields which contains the MAC address of the source machine and the machine for which the message is destined. Length gives the length of the frame. Data field contains the actual data which is also known as Payload. The Frame Check Sequence field contains 32-bits hash code of data generated over the remaining fields. If the checksum computed by destination is not same as sent checksum value, data received is corrupted.

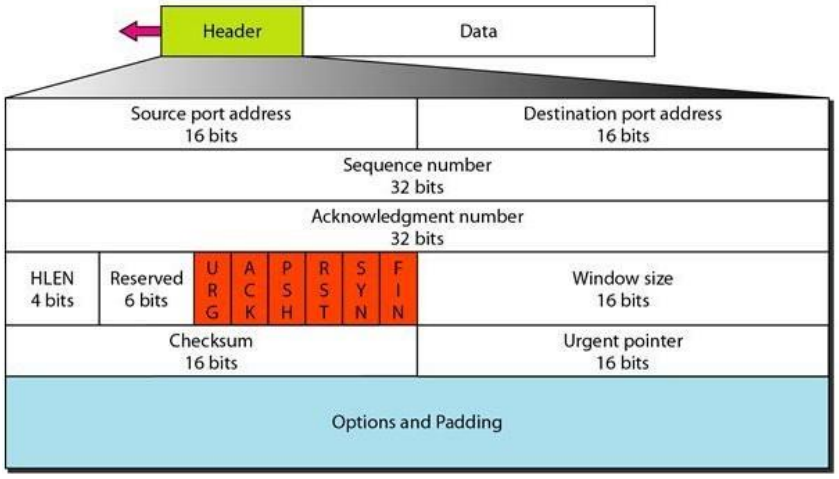


Network Layer (Internet Protocol) - Version gives the version number of the IP packet header. Internet Header Length (IHL) specifies the length of the IP packet header in 32bit words. The Type of Service field is is often ignored by current routers but is meant to allow traffic to be prioritised. Total length is the length of the entire datagram including header and data. The Identification field is used to identify the original IP packet in case of fragmentation during the transmission. If an IP Packet is too large to handle, the 3-bit Flag field tells if it can be fragmented or not. Fragment offset tells the exact position of the fragment in the original IP Packet. Time to Live or TTL is a value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded. The Protocol field identifies the transport-layer protocol which will interpret the Data section. The Header Checksum field is used to keep checksum value of entire header which is then used to check if the packet is received error-free. The source and destination address are 32-bit addresses of the sender (or source) and the receiver (or destination) of the packet. Options is an

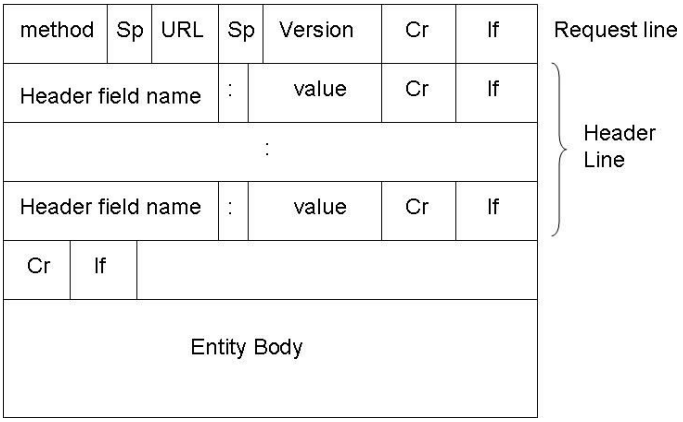


optional field, which is used if the value of IHL is greater than 5 and can be used for security or debugging.

Transport Layer (TCP)- The Source and Destination port address give the source and destination port numbers. The sequence number of the first data octet in this segment is the next field. The acknowledgment field contains the value of the next sequence number the sender of the segment is expecting to receive. HLEN stores the total size of a TCP header in multiples of four bytes. The Reserved field is reserved for future use and must be zero. Window size is the number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept. The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header and text. The urgent pointer points to the sequence number of the octet following the urgent data. Optional TCP data can be used to include support for special acknowledgment and window scaling algorithms.



Application Layer (HTTP) - An *HTTP method*, a verb (like GET, PUT or POST) or a noun (like HEAD or OPTIONS), that describes the action to be performed. The *request target*, usually a URL, or the absolute path of the protocol, port, and domain are usually characterized by the request context. The *HTTP version*, which defines the structure of the remaining message. There are numerous request headers available. They can be divided in several groups: *General headers*, like Via, apply to the message as a whole. *Request headers*, like User-Agent, Accept-Type (accepted body encoding), modify the request by specifying it further (like Accept-Language), by giving context (like Referer), or by conditionally restricting it (like If-None) *Entity headers*, like Content-Length which apply to the body of the request. Obviously, there is no such header transmitted if there is no body in the request. Bodies can be broadly divided into two categories: Single-resource bodies, consisting of one single file, defined by the two headers: Content-Type and Content-Length. Multiple-resource bodies, consisting of a multipart body, each containing a different bit of information. This is typically associated with HTML Forms.



Ans 2. Observed Values for Protocol Fields: Several of the fields have been explained in the previous answer, so I will explain only theadditional ones here.

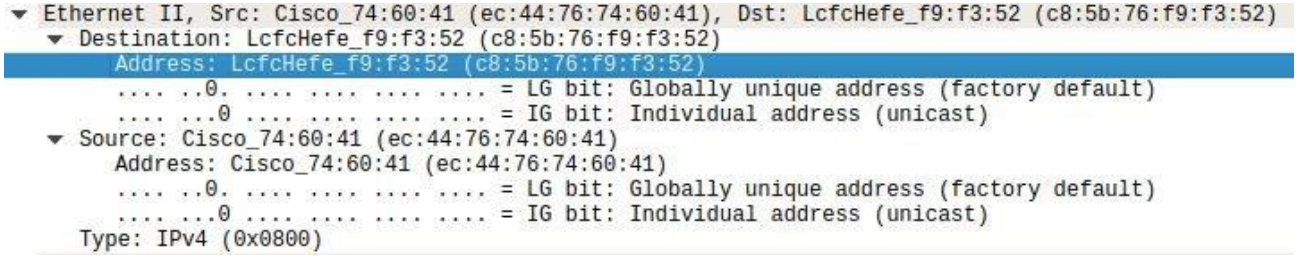
- > Ethernet:
  - Source and destination MAC addresses.
  - Type field for IPv4.

- > IPv4:
  - Version: IPv4.
  - Header length: 20 bytes.
  - DSCP (Differentiated Services Code Point).
  - ECN (Explicit Congestion Notification).
  - TTL: 64.
  - Don't fragment flag set.
  - Protocol field: TCP.
  - Header Checksum: 0x6422.
  - Source and destination IP addresses.

- >TCP:
  - Source and destination port numbers.
  - Header length: 32 bytes.
  - PSH and ACK flags set.
  - Checksum: 0x018c.
  - Window size scaling factor.
  - Urgent pointer not set.

- > HTTP:
  - HTTP method: GET.
  - Request target: addictinggames.com.
  - HTTP version: 1.1.
  - Various request headers (User-Agent, Accept, etc.).

Ethernet -



The values of the source and destination MAC addresses can be seen. Source and Destination both are unicast here meaning point-to-point communication. The type of upper layer protocol used is IPv4.

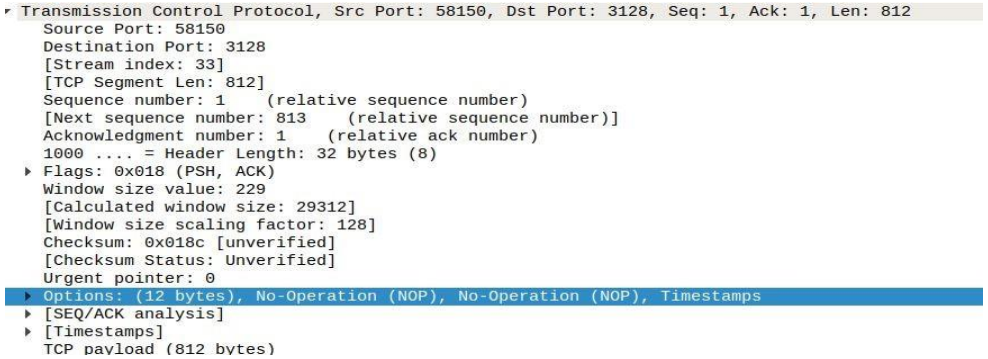
IPv4 -

Version is IPv4, thus leading 4 bits are 0100. Header length is 20 bytes while total datagram size is 864 bytes. Differentiated Services Code Point (DSCP) is a packet header value that can be used to indicate the level of service requested for traffic, such as high priority or best effort delivery. ECN is Explicit Congestion Notification. TTL is 64. Don't fragment flag is set due to which a router which

normally would fragment a packet larger than MTU (and potentially deliver it out of order), instead will drop the packet. Upper layer protocol is TCP. Checksum is value 0x6422. Source and destination IP addresses are also specified.

TCP -

The TCP packet contains the values of the source and destination ports, the sequence and acknowledgement and header length (which is 32 bytes in this case). The PSH and ACK flags are set.



The Push flag tells the receiver's network stack to "push" the data straight to the receiving socket, and not to wait for any more packets before doing so. The ACK flag, which stands for “Acknowledgment”, is used to acknowledge the successful receipt of a packet. Checksum value is 0x018c. The window size scaling factor shows the number of leftward bit shifts that should be used for an advertised window size. The urgent pointer is not set.

HTTP -

The request is a GET request to addictinggames.com using HTTP version 1.1. Proxy connection Keep Alive instructs the server to maintain the TCP connection even after sending response. Cache-Control is used to specify directives that must be obeyed by all caching mechanisms along the request-response chain. Upgrade-Insecure-Requests tells the server that the client would prefer redirection to HTTPS. Pragma is an implementation - specific field that may have various effects anywhere along the request-response chain. Proxy Authorization is added as this reading was taken while using IITG Proxy. The User-Agent request header contains a characteristic string that allows the network protocol peers to identify the application type. Accept lists the acceptable response data types. Accept-encoding and Accept-language give the content encoding, usually a compression algorithm, and the language that the client can understand. A cookie for the website containing information like recent game played and device id is also present.

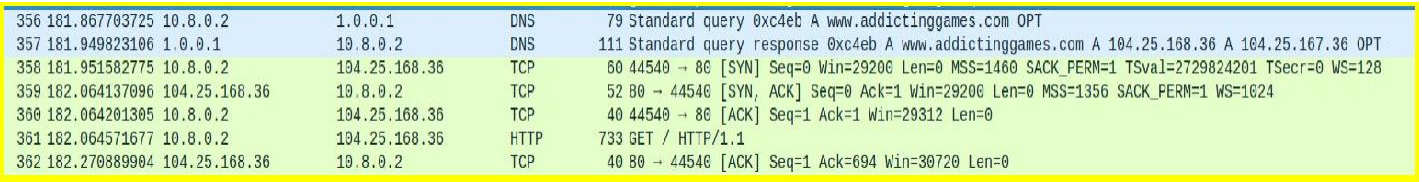
Ans 3: Sequence of Messages and Handshaking

1. DNS query to resolve www.addictinggames.com.
2. 3-way TCP handshake to establish a connection.
3. HTTP GET request to retrieve web content.
4. Server sends requested data over TCP.
5. TCP connection termination with a 3-way handshake.

3-way Handshake:
- SYN (Client) ➡ SYN + ACK (Server) ➡ ACK (Client)

Connection Termination:
- FIN (Client) ➡ ACK (Server) ➡ FIN (Server) ➡ ACK (Client)

Ans 3.



The following exchange of messages (as can be seen in the picture) was observed on visiting the online games website addictinggames.com -

1. A DNS query was sent to get the IP Address corresponding to the domain www.addictinggames.com.
2. A TCP connection is initiated by performing a 3-way handshake with the destination server.
3. An HTTP GET request is sent to get the web page and other attachments.
4. The requested data is sent by the server over the established TCP connection.
5. Connection termination on closing browser by sending FIN TCP packets.

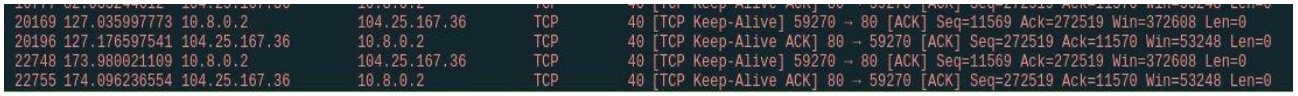
3-way handshake -

Step 1 (SYN) **:** In the first step, client wants to establish a connection with server, so it sends a segment

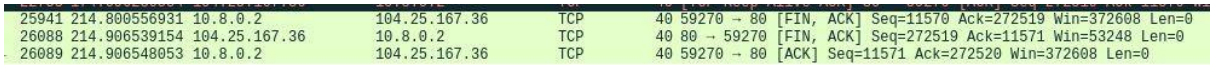
with SYN (Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts its' segments.
Step 2 (SYN + ACK) : Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start its' segments..
Step 3 (ACK) **:** In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer.
Data Transfer - The client using a GET request requests the HTML content After that data exchange happens using TCP protocol with the use of various ACKs and [PSH,ACK]s. TCP's push capability

accomplishes two things: The sending application informs TCP that data should be sent immediately. The PSH flag in the TCP header informs the receiving host that the data should be pushed up to the receiving application immediately. So the server tells the client at various intervals that it has no more data to send and requests an acknowledgement immediately to which the client also replies with an ACK.

After the game has loaded, no live exchange of data occurs. Thus any actions like completing a level, pausing or restarting the game and muting the audio generates no data packets. This is due to the fact that the game I choose was quite basic and was controlled by a single Javascript file. Thus once all the resources are received from the server, no further communication with the server is required. **The client only sends TCP Keep Alive packets so that the server maintains the connection.**



Connection termination -



Step 1 (FIN From Client) **–** Suppose that the client application decides it wants to close the connection. (Note that the server could also choose to close the connection). This causes the client send a TCP segment with the **FIN** bit set to **1** to server and to enter the

Step 2 (ACK From Server) **–** When Server received FIN bit segment from the client, server immediately sends acknowledgement (ACK) segment to the client.

Step 3 (Client waiting) **–** The client waits for a TCP segment from the server with an acknowledgment. When it receives this segment the client waits for another segment from the server with the FIN bit set to 1.

Step 4 (FIN from Server) **–** Server sends FIN bit segment to the Sender(Client) after some time when Server send the ACK segment (because of some closing process in the Server).

Step 5 (ACK from Client) **–** When Client receive FIN bit segment from the Server, the client acknowledges the server's segment. The connection formally closes and all resources on the client side (including port numbers and buffer data) are released.

Ans 4. HTTP (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web. It is based on the client-server paradigm and uses requests and responses for data transfer. As can be seen in the traces, our website uses HTTP version 1.1. It uses it to deliver the website's HTML code, the game icons, javascript and audio to the client.

Transmission Control Protocol (TCP) – By looking at the traces we infer that TCP is responsible for the Transport Layer. Due to the 3-way handshake approach used it ensures that the receiver is ready to receive the resource. If a packet is lost in transit then a retransmission is requested using the system of acknowledgement and sequence numbers. Thus inorder and reliable data transfer is guaranteed. Connection termination too takes place with a sequence of 3 packets in our case. Generally games prefer to use UDP as a faster response is required in a live game. But here very no data is being communicated to the server as all the game resources are loaded initially. Thus TCP seems more viable as it guarantees packet delivery.

IPv4 - The IP has the job of delivering packets using the IP headers from the source to the destination.

Existing in the network layer, IPv4 connection is a hop to hop connection.
Ethernet - Ethernet is the most widely used way of connecting computers together in a local area network or LAN. Details regarding the MAC address of the source and destination are found in Ethernet header. Ethernet lying in data link layer is also responsible for error detection and correction along with flow control.
Ans 5.

Task 5: Caching Mechanisms

- The website uses a Content Distribution Network (CDN) hosted by Cloudfare.
- CDN caches and serves static content like game icons and JavaScript.
- End users receive content from the nearest CDN server, improving performance.

Ans 6.:

	Sample 1	Sample 2	Sample 3
Time	4 P.M	11 PM	5 AM
Network	LAN (Lab)	WIFI through VPN (Room)	LAN (Room)
Throughput	29000	8591	16000
Average RTT	12 ms	2 ms	1 ms
Avg. Packet Size	634 bytes	655 bytes	297 bytes
Total Packets	23092	7305	24253
Packets Lost	22	14	7
No. of TCP Packets	16075	7225	7785
No. of UDP Packets	7014	0	1843
Number of Responses wrt one request send	1.43	1.19	1.25

Name	Type	TTL	Section	IPAddress
www.addictinggames.com	AAAA	0	Answer	2606:4700:20::ac43:475b
www.addictinggames.com	AAAA	0	Answer	2606:4700:20::681a:9a1
www.addictinggames.com	AAAA	0	Answer	2606:4700:20::681a:8a1
www.addictinggames.com	A	0	Answer	172.67.71.91
www.addictinggames.com	A	0	Answer	104.26.9.161
www.addictinggames.com	A	0	Answer	104.26.8.161
dina.ns.cloudflare.com	A	0	Answer	108.162.192.107
dina.ns.cloudflare.com	A	0	Answer	172.64.32.107
dina.ns.cloudflare.com	A	0	Answer	173.245.58.107
dina.ns.cloudflare.com	AAAA	0	Answer	2606:4700:50::adf5:3a6b
dina.ns.cloudflare.com	AAAA	0	Answer	2803:f800:50::6ca2:c06b
dina.ns.cloudflare.com	AAAA	0	Answer	2a06:98c1:50::ac40:206b
glen.ns.cloudflare.com	A	0	Answer	172.64.33.169
glen.ns.cloudflare.com	A	0	Answer	173.245.59.169
glen.ns.cloudflare.com	A	0	Answer	108.162.193.169
glen.ns.cloudflare.com	AAAA	0	Answer	2a06:98c1:50::ac40:21a9
glen.ns.cloudflare.com	AAAA	0	Answer	2606:4700:58::adf5:3ba9
glen.ns.cloudflare.com	AAAA	0	Answer	2803:f800:50::6ca2:c1a9
dina.ns.cloudflare.com	A	0	Answer	173.245.58.107
dina.ns.cloudflare.com	A	0	Answer	108.162.192.107
dina.ns.cloudflare.com	A	0	Answer	172.64.32.107
dina.ns.cloudflare.com	AAAA	0	Answer	2a06:98c1:50::ac40:206b
dina.ns.cloudflare.com	AAAA	0	Answer	2606:4700:50::adf5:3a6b
dina.ns.cloudflare.com	AAAA	0	Answer	2803:f800:50::6ca2:c06b
glen.ns.cloudflare.com	A	0	Answer	108.162.193.169
glen.ns.cloudflare.com	A	0	Answer	172.64.33.169
glen.ns.cloudflare.com	A	0	Answer	173.245.59.169
glen.ns.cloudflare.com	AAAA	0	Answer	2803:f800:50::6ca2:c1a9
glen.ns.cloudflare.com	AAAA	0	Answer	2a06:98c1:50::ac40:21a9
glen.ns.cloudflare.com	AAAA	0	Answer	2606:4700:58::adf5:3ba9

Yes, the whole content is being sent from multiple locations. The website addictinggames.com (172.67.71.91) uses a Content Distribution Network hosted by Cloudfare having domain cdn.addictinggames.com (172.67.71.91). The base ‘index.html’ file and logo image were retrieved from addictinggames.com, while the other game media (including game icons, javascript) were retrieved from the CDN. The latter data is the more crucial both wrt size and frequency of use. So it is critical that it be delivered reliably and quickly. Requested content is cached (pre-saved) by a CDN’s servers meaning end users will get that content by connecting to the nearest CDN server rather than waiting for their request to go directly to the origin server. This results in a significantly better performance for the client. Load balancing on multiple servers leads to reduced workload and reliability in case a server fails.