ASSIGNMENT -2

Name: Kartikeya Saxena

Roll Number: 180101034

Game: Fall Guys

Link to Capture Files:

https://drive.google.com/drive/folders/1pxWX-H9kvqiQ75YuWEVfGGSwlZRGNyda?usp=sharing

Ans. 1- The following protocols used by the game at different layers along with their packet formats:

a) **Link Layer: Ethernet II**
   i) Destination: MAC address of the receiving end (IntelCor_a6:ba:8f (0c:dd:24:a6:ba:8f))
   ii) Source: MAC address of the sending end (b8:c1:ac:91:e2:0d)
   iii) Type: Type of Network Layer Protocol (IPv4)

b) **Network Layer: IPv4**
   i) Source: IP address of the sending end (85.236.96.33)
   ii) Destination: IP address of the receiving end (192.168.1.13)
   iii) Header Checksum: Error detection bits of the datagram(0xc6fb)
   iv) Time To Live(TTL): Maximum number of hops allowed by the packet before reaching destination (59)
   v) Total Length: IP header length + TCP Header length + Application length (314 = 20 + 20 + 274)

c) **Transport Layer: UDP, TCP**
   TCP:
   i) Source Port: port at the sending end(443)
   ii) Destination Port: port at the receiving end(51077)
   iii) Sequence Number: byte number of the first byte of data in the TCP packet sent(4262)
   iv) Window size: how much data (in bytes) the receiving device is willing to receive at any point in time.(129)
   v) Checksum: Error detection bits of the segment(0x4c46 [correct])
   UDP:
   i) Source Port: (61662)
   ii) Destination Port: (7877)
   iii) Length: UDP Header length + Application length(24 = 8 + 16)
   iv) Checksum: (0x44de [correct])

d) **Session Layer: Secure Sockets Layer, TLSv1.2**
   i) Content Type: Type of content whether Application Data, Handshake etc. (Application Data)
   ii) Version: (TLS 1.2)
   iii) Length: Length of the data(78)
   iv) Encrypted Application Data

e) **Application Layer: data, HTTP, DNS**

Data:

i) Data: When Wireshark can't determine how part of a packet should be formatted, it marks that chunk as "Data"(0035be6e0f9900408000000000000000)

DNS:

i) Queries: DNS Queries for host name resolution

ii) Answers: Answer to DNS Queries

iii) Questions: Queries Count (1)

iv) Answer RRs: Answers Count (3)

f) Frame: It contains info about the transferred packet as a whole.

i) Interface id: interface used for the connection (0) means wlp2s0.

ii) Frame Length: (137)

iii) Arrival Time: (Sep 26, 2020 13:29:48.509149000 IST)

I am taking the data from **file_guys2.pcapng** packet number 2041, 2173, 2658 without any filters

Ans. 2- Functionalities of the application

a) Pause/End/Open:

Protocols:

i) Ethernet II: All packets use this protocol as it enabled collision-free interconnection of multiple devices via a common bus. It also gives error free transmission of packets.

ii) IPv4: It is required since the client needs to connect to the internet to connect to the application server.

iii) TCP: This is required because we need to establish a reliable connection to log the user in and connect to the game field server.

iii) UDP: It also uses UDP for DNS queries and for requesting cloud services from amazonaws and unity.

iv) TLSv1.2: This is used to encrypt the data (such as player login and privacy data) before transferring it. These are mainly used by akamai.net which provides the security for the application.

v) DNS: This is used in the beginning when the application is opened for host address resolution and uses UDP below it.

b) Start/Resume:

Protocols:

i) Ethernet II: All packets use this protocol as it enabled collision-free interconnection of multiple devices via a common bus. It also gives error free transmission of packets.

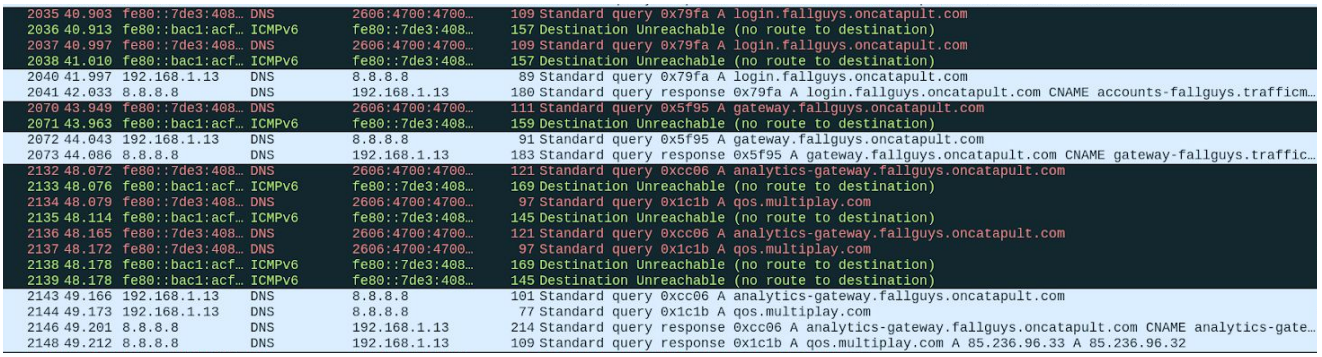
ii) IPv4: It is required since the client needs to connect to the internet to connect to the application server.

iii) UDP: It is required since it is a time sensitive and Loss-tolerant application and requires constant data flow which requires fastness over reliability.

iv) Data: When Wireshark can't determine how part of a packet should be formatted, it marks that chunk as "Data". The "Data" is a protocol that Wireshark doesn't support.

Ans.3- a) Pause/End/Open:

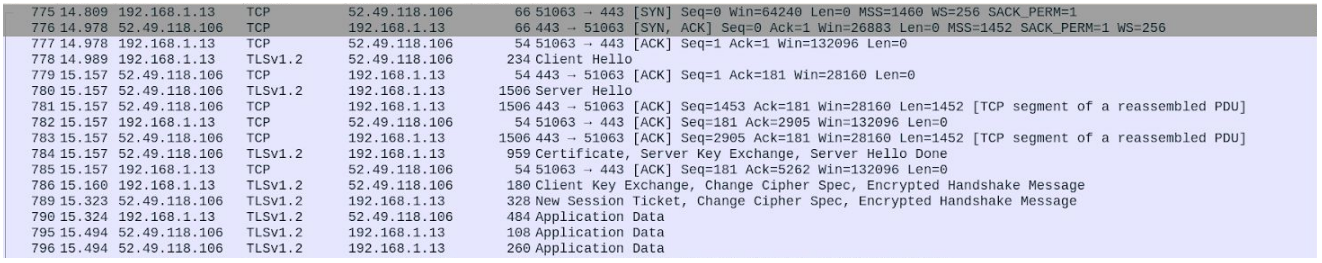
DNS



2035	40.903	fe80::7de3:408…	DNS	2606:4700:4700…	109	Standard query 0x79fa A login.fallguys.oncatapult.com
2036	40.913	fe80::bac1:acf…	ICMPv6	fe80::7de3:408…	157	Destination Unreachable (no route to destination)
2037	40.997	fe80::7de3:408…	DNS	2606:4700:4700…	109	Standard query 0x79fa A login.fallguys.oncatapult.com
2038	41.010	fe80::bac1:acf…	ICMPv6	fe80::7de3:408…	157	Destination Unreachable (no route to destination)
2040	41.997	192.168.1.13	DNS	8.8.8.8	89	Standard query 0x79fa A login.fallguys.oncatapult.com
2041	42.033	8.8.8.8	DNS	192.168.1.13	180	Standard query response 0x79fa A login.fallguys.oncatapult.com CNAME accounts-fallguys.traffic…
2070	43.949	fe80::7de3:408…	DNS	2606:4700:4700…	111	Standard query 0x5f95 A gateway.fallguys.oncatapult.com
2071	43.963	fe80::bac1:acf…	ICMPv6	fe80::7de3:408…	159	Destination Unreachable (no route to destination)
2072	44.043	192.168.1.13	DNS	8.8.8.8	91	Standard query 0x5f95 A gateway.fallguys.oncatapult.com
2073	44.086	8.8.8.8	DNS	192.168.1.13	183	Standard query response 0x5f95 A gateway.fallguys.oncatapult.com CNAME gateway-fallguys.traffic…
2132	48.072	fe80::7de3:408…	DNS	2606:4700:4700…	121	Standard query 0xcc06 A analytics-gateway.fallguys.oncatapult.com
2133	48.076	fe80::bac1:acf…	ICMPv6	fe80::7de3:408…	169	Destination Unreachable (no route to destination)
2134	48.079	fe80::7de3:408…	DNS	2606:4700:4700…	97	Standard query 0x1c1b A qos.multiplay.com
2135	48.114	fe80::bac1:acf…	ICMPv6	fe80::7de3:408…	145	Destination Unreachable (no route to destination)
2136	48.165	fe80::7de3:408…	DNS	2606:4700:4700…	121	Standard query 0xcc06 A analytics-gateway.fallguys.oncatapult.com
2137	48.172	fe80::7de3:408…	DNS	2606:4700:4700…	97	Standard query 0x1c1b A qos.multiplay.com
2138	48.178	fe80::bac1:acf…	ICMPv6	fe80::7de3:408…	169	Destination Unreachable (no route to destination)
2139	48.178	fe80::bac1:acf…	ICMPv6	fe80::7de3:408…	145	Destination Unreachable (no route to destination)
2143	49.166	192.168.1.13	DNS	8.8.8.8	101	Standard query 0xcc06 A analytics-gateway.fallguys.oncatapult.com
2144	49.173	192.168.1.13	DNS	8.8.8.8	77	Standard query 0x1c1b A qos.multiplay.com
2146	49.201	8.8.8.8	DNS	192.168.1.13	214	Standard query response 0xcc06 A analytics-gateway.fallguys.oncatapult.com CNAME analytics-gate…
2148	49.212	8.8.8.8	DNS	192.168.1.13	109	Standard query response 0x1c1b A qos.multiplay.com A 85.236.96.33 A 85.236.96.32

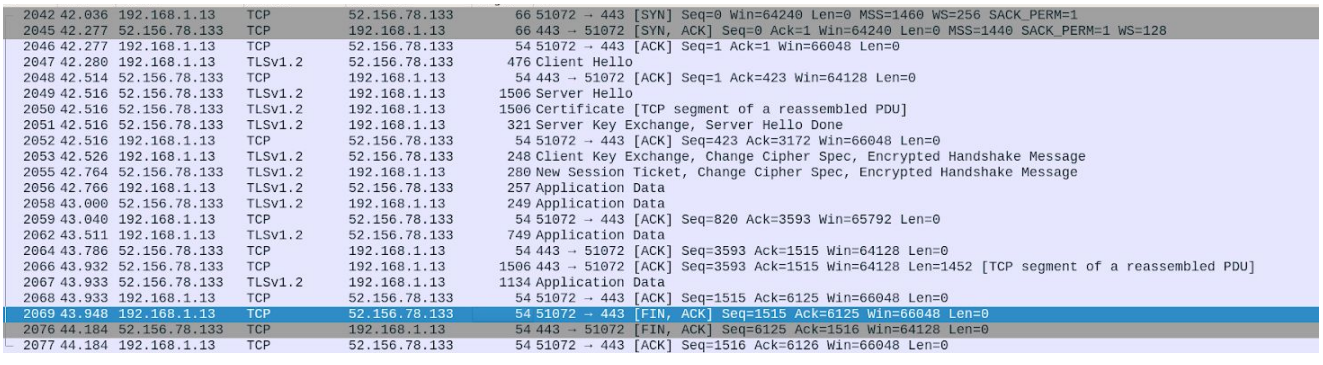
Whenever an application is opened, multiple DNS queries are performed to multiple DNS servers, which are answered to resolve the address for game gateway and login (gateway-prod.fallguys.oncatapult.com(52.175.249.150) and login-prod.fallguys.oncatapult.com(52.156.78.133))
No handshaking is done.

TCP



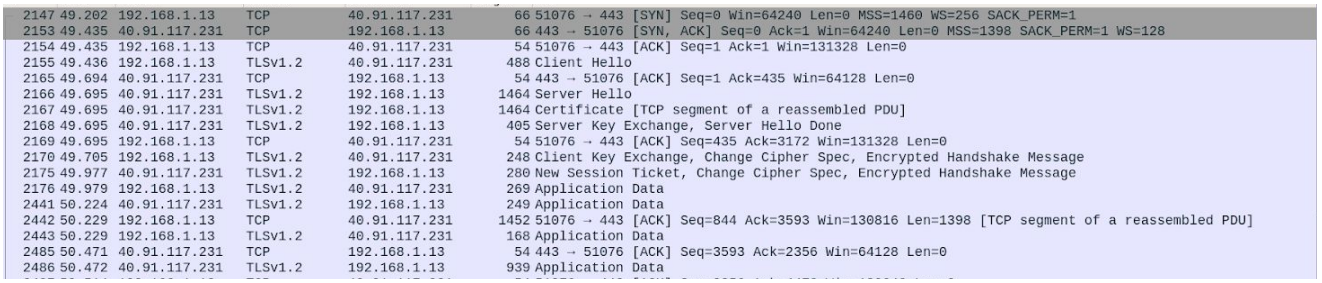
775	14.809	192.168.1.13	TCP	52.49.118.106	66	51063 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
776	14.978	52.49.118.106	TCP	192.168.1.13	66	443 → 51063 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1452 SACK_PERM=1 WS=256
777	14.978	192.168.1.13	TCP	52.49.118.106	54	51063 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
778	14.989	192.168.1.13	TLSv1.2	52.49.118.106	234	Client Hello
779	15.157	52.49.118.106	TCP	192.168.1.13	54	443 → 51063 [ACK] Seq=1 Ack=181 Win=28160 Len=0
780	15.157	52.49.118.106	TLSv1.2	192.168.1.13	1506	Server Hello
781	15.157	52.49.118.106	TCP	192.168.1.13	1506	443 → 51063 [ACK] Seq=1453 Ack=181 Win=28160 Len=1452 [TCP segment of a reassembled PDU]
782	15.157	192.168.1.13	TCP	52.49.118.106	54	51063 → 443 [ACK] Seq=181 Ack=2905 Win=132096 Len=0
783	15.157	52.49.118.106	TCP	192.168.1.13	1506	443 → 51063 [ACK] Seq=2905 Ack=181 Win=28160 Len=1452 [TCP segment of a reassembled PDU]
784	15.157	52.49.118.106	TLSv1.2	192.168.1.13	959	Certificate, Server Key Exchange, Server Hello Done
785	15.157	192.168.1.13	TCP	52.49.118.106	54	51063 → 443 [ACK] Seq=181 Ack=5262 Win=132096 Len=0
786	15.160	192.168.1.13	TLSv1.2	52.49.118.106	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
789	15.323	52.49.118.106	TLSv1.2	192.168.1.13	328	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
790	15.324	192.168.1.13	TLSv1.2	52.49.118.106	484	Application Data
795	15.494	52.49.118.106	TLSv1.2	192.168.1.13	108	Application Data
796	15.494	52.49.118.106	TLSv1.2	192.168.1.13	260	Application Data

Then we request for some cloud services from gamesec-gossip-eu-lb-prod-1789504702.eu-west-1.elb.amazonaws.co(52.49.118.106)
3 Way Handshake in the line 775, 776, 777. The client sends a message to initiate the connection by sending a SYN, the server accepts it by sending an ACK and requests the client for connection by sending SYN along with the previous ACK. Finally the client sends back an ACK to the server to accept the connection.
The conversation starts by “Client Hello” which basically will include which TLS version the client supports and the cipher suites supported to which server replies with “Server Hello” which contains the SSL Certificates and cipher suites it is going to use. Then the encryption keys are exchanged with encrypted handshake messages and then messages are exchanged using TLSv1.2. Finally FIN is used by client to close the connection.



2042	42.036	192.168.1.13	TCP	52.156.78.133	66	51072 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2045	42.277	52.156.78.133	TCP	192.168.1.13	66	443 → 51072 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1440 SACK_PERM=1 WS=128
2046	42.277	192.168.1.13	TCP	52.156.78.133	54	51072 → 443 [ACK] Seq=1 Ack=1 Win=66048 Len=0
2047	42.280	192.168.1.13	TLSv1.2	52.156.78.133	476	Client Hello
2048	42.514	52.156.78.133	TCP	192.168.1.13	54	443 → 51072 [ACK] Seq=1 Ack=423 Win=64128 Len=0
2049	42.516	52.156.78.133	TLSv1.2	192.168.1.13	1506	Server Hello
2050	42.516	52.156.78.133	TLSv1.2	192.168.1.13	1506	Certificate [TCP segment of a reassembled PDU]
2051	42.516	52.156.78.133	TLSv1.2	192.168.1.13	321	Server Key Exchange, Server Hello Done
2052	42.516	192.168.1.13	TCP	52.156.78.133	54	51072 → 443 [ACK] Seq=423 Ack=3172 Win=66048 Len=0
2053	42.526	192.168.1.13	TLSv1.2	52.156.78.133	248	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2055	42.764	52.156.78.133	TLSv1.2	192.168.1.13	280	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2056	42.766	192.168.1.13	TLSv1.2	52.156.78.133	257	Application Data
2058	43.000	52.156.78.133	TLSv1.2	192.168.1.13	249	Application Data
2059	43.040	192.168.1.13	TCP	52.156.78.133	54	51072 → 443 [ACK] Seq=820 Ack=3593 Win=65792 Len=0
2062	43.511	192.168.1.13	TLSv1.2	52.156.78.133	749	Application Data
2064	43.786	52.156.78.133	TCP	192.168.1.13	54	443 → 51072 [ACK] Seq=3593 Ack=1515 Win=64128 Len=0
2066	43.932	52.156.78.133	TCP	192.168.1.13	1506	443 → 51072 [ACK] Seq=3593 Ack=1515 Win=64128 Len=1452 [TCP segment of a reassembled PDU]
2067	43.933	52.156.78.133	TLSv1.2	192.168.1.13	1134	Application Data
2068	43.933	192.168.1.13	TCP	52.156.78.133	54	51072 → 443 [ACK] Seq=1515 Ack=6125 Win=66048 Len=0
2069	43.948	192.168.1.13	TCP	52.156.78.133	54	51072 → 443 [FIN, ACK] Seq=1515 Ack=6125 Win=66048 Len=0
2076	44.184	52.156.78.133	TCP	192.168.1.13	54	443 → 51072 [FIN, ACK] Seq=6125 Ack=1516 Win=64128 Len=0
2077	44.184	192.168.1.13	TCP	52.156.78.133	54	51072 → 443 [ACK] Seq=1516 Ack=6126 Win=66048 Len=0

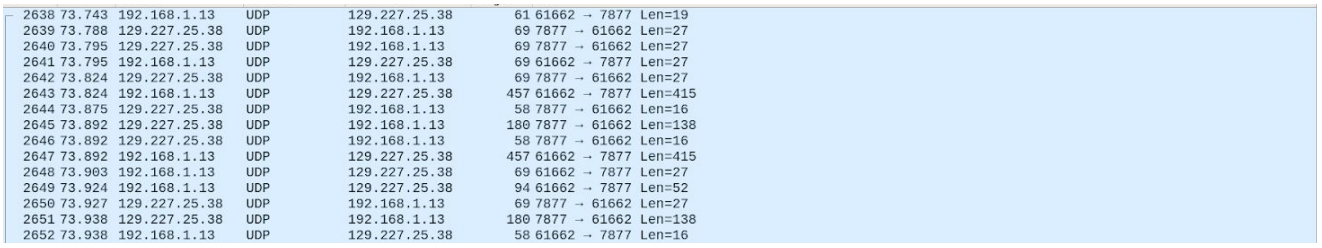
Then we request to login as our player from login-prod.fallguys.oncatapult.com(52.156.78.133)
3 Way Handshake in the line 2042, 2045, 2046. The procedure is the same as above.



2147	49.202	192.168.1.13	TCP	40.91.117.231	66	51076 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2153	49.435	40.91.117.231	TCP	192.168.1.13	66	443 → 51076 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1398 SACK_PERM=1 WS=128
2154	49.435	192.168.1.13	TCP	40.91.117.231	54	51076 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
2155	49.436	192.168.1.13	TLSv1.2	40.91.117.231	488	Client Hello
2165	49.694	40.91.117.231	TCP	192.168.1.13	54	443 → 51076 [ACK] Seq=1 Ack=435 Win=64128 Len=0
2166	49.695	40.91.117.231	TLSv1.2	192.168.1.13	1464	Server Hello
2167	49.695	40.91.117.231	TLSv1.2	192.168.1.13	1464	Certificate [TCP segment of a reassembled PDU]
2168	49.695	40.91.117.231	TLSv1.2	192.168.1.13	405	Server Key Exchange, Server Hello Done
2169	49.695	192.168.1.13	TCP	40.91.117.231	54	51076 → 443 [ACK] Seq=435 Ack=3172 Win=131328 Len=0
2170	49.705	192.168.1.13	TLSv1.2	40.91.117.231	248	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2175	49.977	40.91.117.231	TLSv1.2	192.168.1.13	280	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2176	49.979	192.168.1.13	TLSv1.2	40.91.117.231	269	Application Data
2441	50.224	40.91.117.231	TLSv1.2	192.168.1.13	249	Application Data
2442	50.229	192.168.1.13	TCP	40.91.117.231	1452	51076 → 443 [ACK] Seq=844 Ack=3593 Win=130816 Len=1398 [TCP segment of a reassembled PDU]
2443	50.229	192.168.1.13	TLSv1.2	40.91.117.231	168	Application Data
2485	50.471	40.91.117.231	TCP	192.168.1.13	54	443 → 51076 [ACK] Seq=3593 Ack=2356 Win=64128 Len=0
2486	50.472	40.91.117.231	TLSv1.2	192.168.1.13	939	Application Data

Then we establish a connection through the game gateway through which we will exchange data while in pause, end or whenever we are on the main page of the game.
3 Way Handshake in the line 2147, 2153, 2154. The procedure is the same as above.

b) Start:
UDP



2638	73.743	192.168.1.13	UDP	129.227.25.38	61	61662 → 7877 Len=19
2639	73.788	129.227.25.38	UDP	192.168.1.13	69	7877 → 61662 Len=27
2640	73.795	129.227.25.38	UDP	192.168.1.13	69	7877 → 61662 Len=27
2641	73.795	192.168.1.13	UDP	129.227.25.38	69	61662 → 7877 Len=27
2642	73.824	129.227.25.38	UDP	192.168.1.13	69	7877 → 61662 Len=27
2643	73.824	192.168.1.13	UDP	129.227.25.38	457	61662 → 7877 Len=415
2644	73.875	129.227.25.38	UDP	192.168.1.13	58	7877 → 61662 Len=16
2645	73.892	129.227.25.38	UDP	192.168.1.13	180	7877 → 61662 Len=138
2646	73.892	129.227.25.38	UDP	192.168.1.13	58	7877 → 61662 Len=16
2647	73.892	192.168.1.13	UDP	129.227.25.38	457	61662 → 7877 Len=415
2648	73.903	192.168.1.13	UDP	129.227.25.38	69	61662 → 7877 Len=27
2649	73.924	192.168.1.13	UDP	129.227.25.38	94	61662 → 7877 Len=52
2650	73.927	129.227.25.38	UDP	192.168.1.13	69	7877 → 61662 Len=27
2651	73.938	129.227.25.38	UDP	192.168.1.13	180	7877 → 61662 Len=138
2652	73.938	192.168.1.13	UDP	129.227.25.38	58	61662 → 7877 Len=16

The entire duration of the game we exchange UDP segments from (129.227.25.38). There are no sequence numbers and acknowledgement numbers.
No handshaking is done.

Ans.4-
I applied the ip filter to remove the ARP packets captured from chatter in my wifi network. All the values are rough estimates since some background application might be running whose packets would influence the results.

Statistics	Morning	Afternoon	Night
Throughput	146k bits/s	150k bits/s	137k bits/s
RTT	0.06268 seconds	0.13496 seconds	0.12492 seconds
Packet size	491 Bytes	405 Bytes	391 Bytes
No. of Packets Lost	1	3	1
UDP Packets	6384	14456	10816
TCP Packets	3979	2269	426
No. of responses / request	6408/3943	11508/5153	7897/3146

Morning: fall_guys7.pcapng
Afternoon: fall_guys2.pcapng
Night: fall_guys5.pcapng

Ans. 5- Yes, data is fetched from multiple destinations at different times.
Morning: 129.227.201.158

Afternoon: 129.227.25.38
Night: 129.227.20.2
These multiple IP exist due to the load balancer of the game server. Multiple game servers are running parallelly and each require exactly 60 players to begin the game, so the load balancer(reverse proxy) cleverly distributes the load to the server which needs players at that time. Hence different IPs at different times.