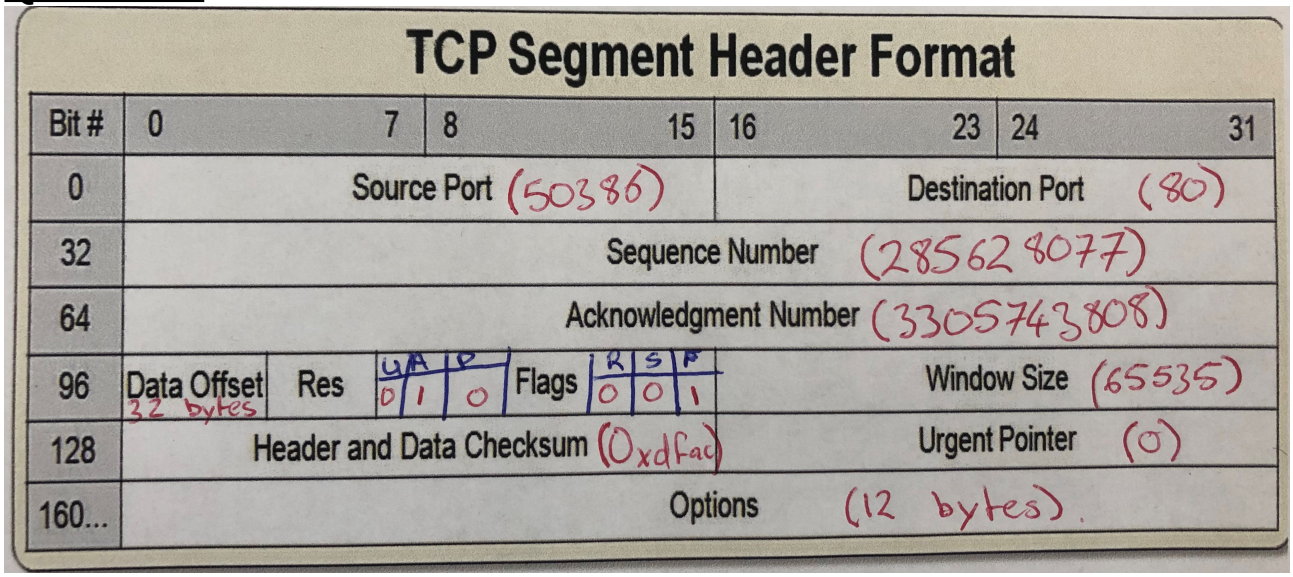
Lab 5

Josh Browne

C00224660

Question 1:



TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port (50386)		Destination Port (80)	
32	Sequence Number (285628077)			
64	Acknowledgment Number (3305743808)			
96	Data Offset (32 bytes)	Res	Flags (URG=0, ACK=1, PSH=0, RST=0, SYN=0, FIN=1)	Window Size (65535)
128	Header and Data Checksum (0xdfac)		Urgent Pointer (0)	
160...	Options (12 bytes)			

Source Port (2 bytes) (50386): This is the first parameter in the TCP header. This is the address of the sender application over TCP.

Destination Port (2 bytes) (80): Port number of thr destination user of recieving TCP. It is set by the user. The parameter is mandatory. Over a public internet, the port numbers also called called *well-known ports*. (e.g a website always runs over a default port 80)

Sequence Number (4 bytes) (285628077): TCP does the sequence control using this sequence number. This basically is how TCP correctly arranges the *data* on the recieving end so that its correctly arranged before being sent to the application.

Acknowledgement Number (4 bytes) (3305743808): TCP makes sure that a message sent to the remote layer has been recieved (reliable protocol). It does this by using this TCP *Ack* number. *Ack* number is set by reciever. The value signifies the expected next sequence number from the sender.

Data Offset (32 bytes): This field is for the purpose of giving the length of bits that is used for the TCP Header. The rest of the bits after the TCP header is the user data.

In my example, the Header length is 32 bytes. This means that after 32 bytes into this packet, is the data.

Reserved (): This field is unused and may be used in the future.

Flags (6 bits) (010001): A flag is a parameter which is only one bit long, so its value can only be zero or one. If a flags value is one then it is set and if its zero it means the flag is not set. TCP flags are a set of 6 bits (6 flags, each explained below).

- **URG:** If Urgent Pointer is valid, then this flag is set.
- **ACK:** This is set if there is an acknowledgement sequence number in the TCP header.
- **PSH:** Push request. Basically, when this flag is set, it delivers the pending bytes immediatly to the application.
- **RST:** Reset flag.
- **SYN:** If set, this segment is for the connection setup.
- **FIN:** If set, connection terminated.

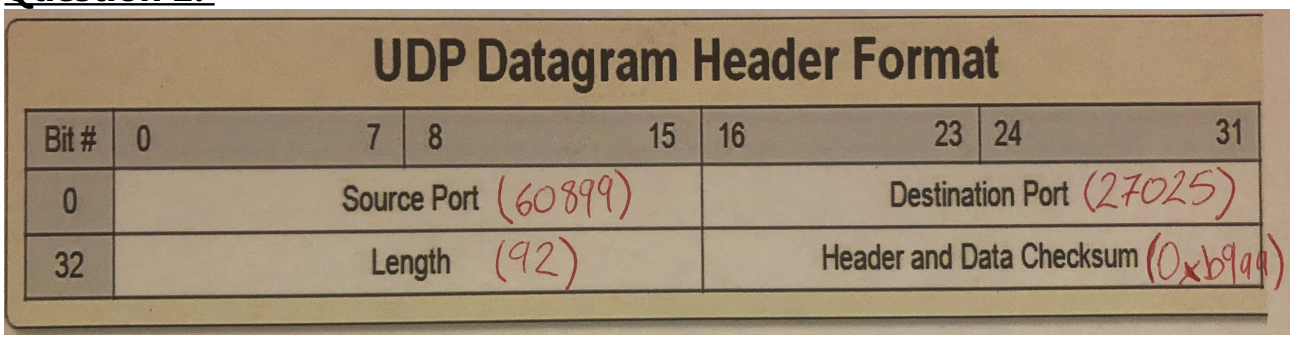
Window Size (32 bit) (65535): Has usage in flow control. Contains size of reciever window.

Header and Data Checksum (16 bit) (0xdfac): The sender computes and sets the checksum before sending it to the reciever. On the reciever side, the checksum is calculated again and matched. If the chechsum does not match, it means the segment is corrupted and discarded. The purpose of this is to make sure that the TCP segment is not altered over the network.

Urgent Pointer (0): As the name suggests, it is something that should process immediately. When the URG flag is set, the parameter tells how many bytes are urgent. The receiver side sends the urgent bytes fist to the application.

Options: These are optional parameters.

Question 2:



UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port (60899)		Destination Port (27025)	
32	Length (92)		Header and Data Checksum (0xb9aa)	

Source Port (16 bits) (60899): This field identifies the sender's port, when used, and should be assumed to be the port to reply to if needed.

Destination Port (16 bits) (27025): This field identifies the receiver's port and is required.

Length (16 bits) (92): This field specifies the length in bytes of the UDP header and UDP data.

Header and Data Checksum (16 bits) (0xb9aa): This field is used for error checking of the header and data. This field is optional in IPv4, and mandatory for Ipv6. If unused, the field is all zeros.

Question 3:

Verify the checksum using 16-bit One's Complement Sum algorithm.

Header binary values:

0000 0111 1101 0111
0000 0111 1101 0111
0000 0000 0001 1100
1110 1010 1111 0101

Addition:

0000 0111 1101 0111
0000 0111 1101 0111 +

0000 1111 1010 1110
0000 0000 0001 1100 +

0000 1111 1100 1010
1110 1010 1111 0101 +

1111 1010 1011 1111 = One's complement sum
0000 0101 0100 0000 = One's complement

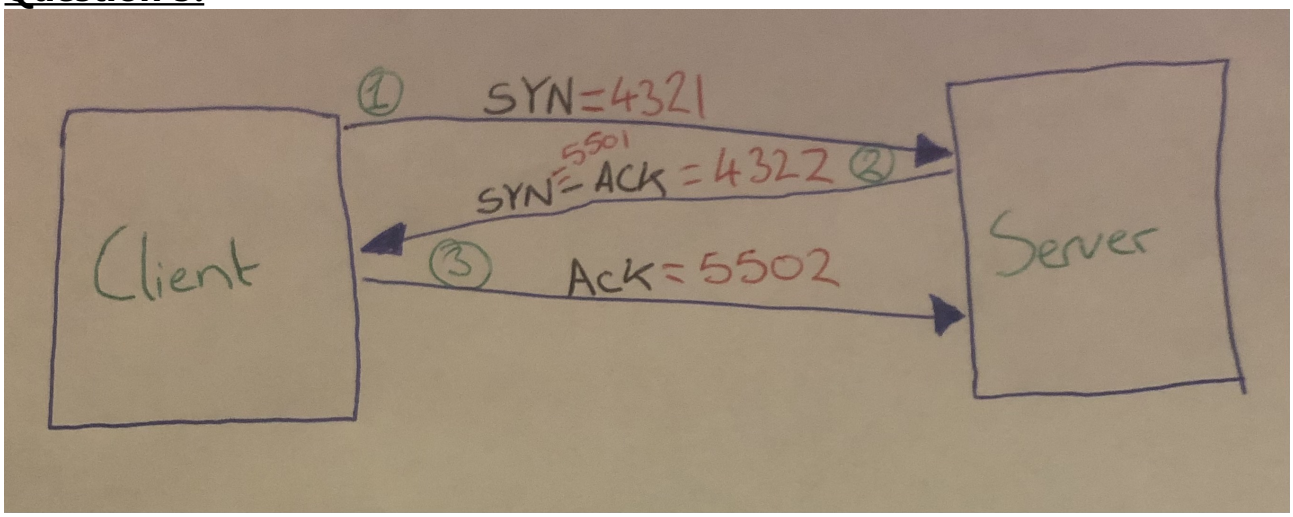
Question 4:

I captured packets from the Twitch streaming application. I found that the majority of packets where of TCP type. Although, there where a few UDP's captured.

UDP betters TCP for live video sharing services for the following reasons:

- UDP offers reduced latency over the TCP reliability
- In case of time sensitive applications, UDP is a faster protocol as it doesn't wait for acknowledgement from the client side and retransmission of lost packet.

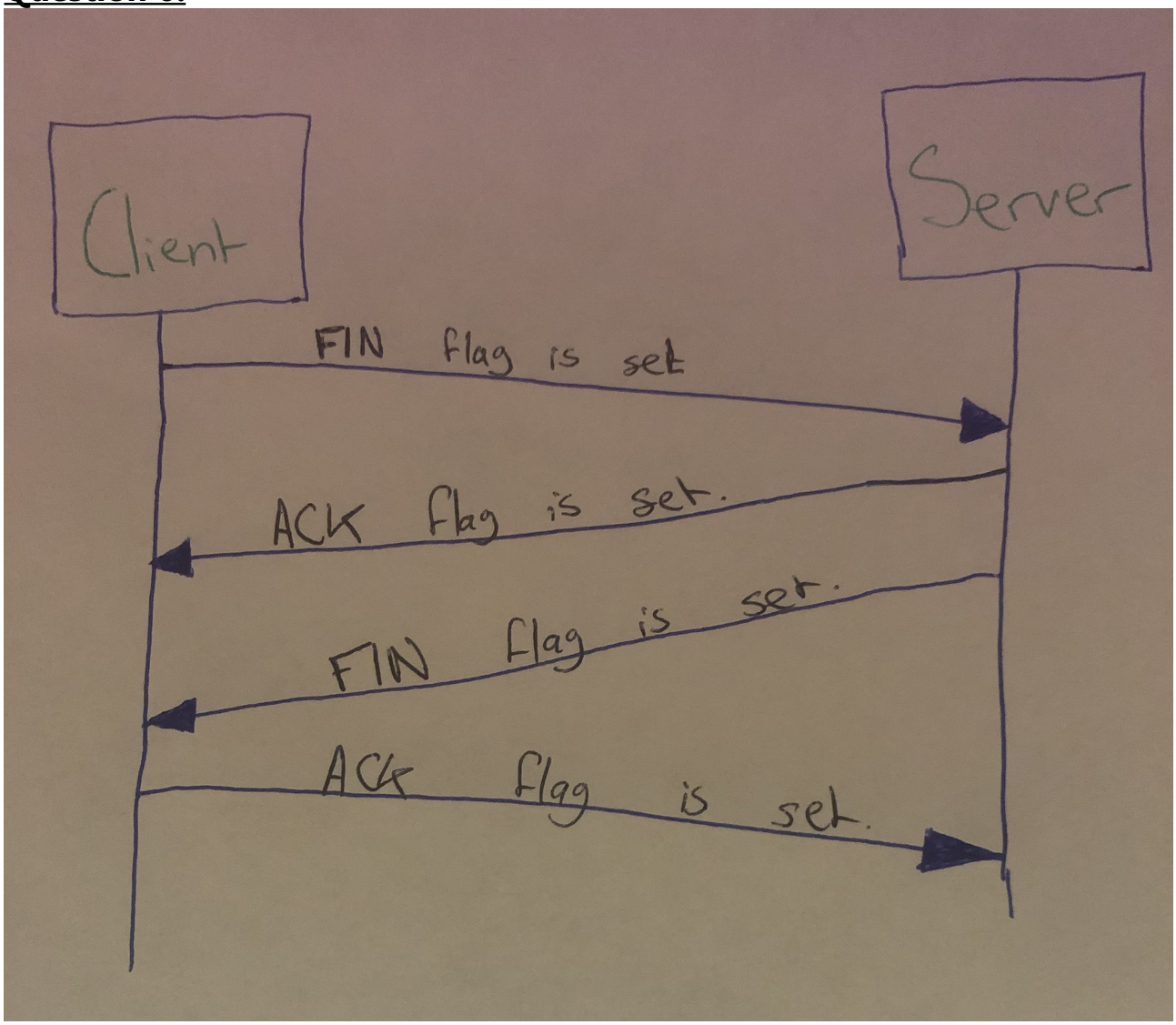
Question 5:



TCP 3-way handshake is a process which is used in a TCP/IP network to make a connection between server and client. The steps to this process are as follows:

1. The client establishes a connection with the server. It sends a packet with SYN set and this informs the server about the client and that they should start communication.
2. The server responds to the client's request with SYN & ACK flags set. The acknowledgement number that will be sent is the the client's initial Sequence number plus one. (e.g clients sequence num = 4321, servers acknowledgement num = 4322. (from diagram above))
3. In this final step, the client acknowledges the response of the Server, and they both create a stable connection will begin the actual data transfer process.

Question 6:



TCP connection tear-down is performed with a 4-way handshake. Specifically, in order for an established TCP connection to be terminated, the following 4 TCP packets are exchanged:

1. Host A → Host B: FIN flag set.
2. Host B → Host A: ACK flag set.
3. Host B → Host A: FIN flag set.
4. Host A → Host B: ACK flag set.

These 4 steps that terminate a TCP connection are described as a 4-way handshake.