1.

Source port number 2 bytes			Destination port number 2 bytes
Sequence number 4 bytes			
Acknowledgement number 4 bytes			
Data offset 4 bits	Reserved 3 bits	Control flags 9 bits	Window size 2 bytes
Checksum 2 bytes			Urgent pointer 2 bytes
Optional data 0 - 40 bytes			

Source port number 65500			Destination port number 8080
Sequence number 34 (2728724858 raw)			
Acknowledgement number 2 (907036626 raw)			
Data offset 0101	Reserved 000	Control flags 0 0001 0000	Window size 8212
Checksum 0xb565 [unverified]			Urgent pointer 0
Optional data			

Source Port: this is the port used by the pc sending the TPC segment.

Destination Port: this is the port used by the pc receiving the TPC segment.

Sequence Number: this helps the software on both sides to keep track of how much data has been transferred and put the data in the correct order.

Data Offset: indicates the number of bytes in the TCP packet where the data can be found.

Window Size: is the number of octets in the header.

Checksum: is used by the receiver to verify the integrity of the data.

Urgent Pointer: is the end of an “urgent” data in the packet.

Optional Data: contains the data from the user application.

2.

Source Port	Destination Port
Length	Checksum
Data	

Source Port 54915	Destination Port 54915
Length 271	Checksum 0x49df [unverified]
Data 263	

Source Port: is the port of the device sending the data.

Destination Port: is the port of the device receiving the data.

Length: is the number of bytes comprising the UDP header and UDP payload data.

Checksum: is used by the receiver to verify the integrity of the header and payload data.

3.

Checksum = 0x0000a520

Source address = 192.168.0.101

Destination address = 192.168.0.10

Protocol = 17

Length = 16

Source Port = 49905

Destination Port = 50950000

Payload Length (in Hex) =

{

01 00

01 00

00 0d

00 08

}

11000000.10101000

\+ 00000000.01100101

11000001.00001101 (source)

11000000.10101000

\+ 00000000.00001010

11000000.10110010 (destination)

11000001.00001101

\+ 11000000.10110010

10000001.11000000 (source and destinatiom)

10000001.11000000

\+ 00000000.00010001

10000001.11010001 (protocol)

10000001.11010001

\+ 00000000.00010000

10000001.11100001 (length of packet)

10000001.11100001

\+ 11000010.11110001

01000100.11010011 (source port)

01000100.11010011

\+ 00010011.11100111

01011000.10111010 (destination port)

01011000.10111010

\+ 00000000.00010000

01011000.11001010 (length of packet)

01011000.11001010

\+ 00000001.00000000

01011001.11001010

01011001.11001010

\+ 00000001.00000000

01011010.11001010

01011010.11001010

\+ 00000000.00001101

01011010.11010111

01011010.11010111

\+ 00000000.00001000

01011010.11011111

01011010.11011111

10100101.00100000

HEX VALUE = A520

4.

I captured data from YouTube. I noticed that for live videos the UDP is used as it is faster than TCP as it doesn't have as much error checking but the live was slightly of a lower quality. And for videos that weren't live it used the TCP and as this didn't have to receive data constantly it could do all the error checking necessary and the video overall was of better quality.

5.

A three-way handshake is a method used in TCP/IP network to create a connection between a local host/client and server. It's a 3-step process that requires both the client and server to exchange packets before the real data communication process starts.

Step 1 (SYN): In the first step, the client wants to establish a connection with a server, so it sends a segment with Synchronize Sequence Number which informs the server that the client is likely to start communication and with what sequence number it starts segments with

Step 2 (SYN + ACK): Server responds to the client request with SYN-ACK signal bits set. Acknowledgement (ACK) signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with

Step 3 (ACK): In the final part client acknowledges the response of the server and they both establish a reliable connection with which they will start the actual data transfer

6.

A TCP-4 way teardown happens after the TCP 3-way handshake, when the server wants to finish the connection. The connection is terminated after the server send the FIN flag.

Step 1: Send SYN from the client to the server.

Step 2: send SYN/ACK from server to client.

Step 3: send ACK from client to server.

Step 4: the server sets the FIN flags and ends the connection.