**How video transforms into IP packets**

The first step in transforming the video into IP packets is compressing the video using an encoder. After this step the video is now in the form of a stream. This stream is fed into a packetizer to make sure that the video can fit into the IP packets. The size of these packets is very important as there are pros and cons for making them smaller or bigger. For smaller packets lost packets are less harmful for the overall video because less of the video was held in the packets. Some video streaming sites can somewhat mask missing parts of the videos. Losing larger packets means that more information is lost making it is harder to mask. With longer packets you will need less bandwidth to send the video as it has less header information than shorter packets. After going through the packetizer the stream becomes a packetized stream with header information that contains information such as the source and time stamps. This is helpful for combining video and audio back together. These packetized streams are then combined with other video, audio, and/or data packetized streams using a multiplexer to make either program streams or transport streams.

When transporting the combined streams User Datagram Protocol (UDP), Transmission Control Protocol (TCP) and Real-time Transport Protocol (RTP) are the main protocols used. When the package is being sent it will be sent using the protocol that is best for it. UDP and TCP are both good for regular video playback but for any real-time videos RTP is the best. These protocols lie above IP in the TCP/IP reference model on the transport layer.

UDP can support high-speed information, it can not ensure that the data it sent was transferred correctly. This may make it seem like UDP would be bad for video streaming but since each image is displayed very quickly any type of error detection would slow the stream down delivery down. This makes UDP very quick and easy to set-up to start sending data.

TCP on the on the other hand requires that a connection be set-up between sender and receiver before any data is sent out. This means that TCP can do both flow control and make sure that every byte that was sent is received. However, this can slow down playback for videos as the information has to be resent if any data is lost making TCP bad for live streams.

Finally, there is RTP. This protocol is built on top of UDP and was made for real-time data transportation. RTP can detect congestion and let the receiver know but unlike TCP does not throttle down bandwidth or resend the packets which can lead to the receiver disabling video but keep displaying the audio. Finally, RTP allows for multicasting meaning that a single source can be received in destinations simultaneously.