09. Multimedia Networking

9.1 Multimedia Networking Applications

A multimedia networking application is any application that employs audio or video.

Video applications dominate the internet traffic volume, so it's important to do what we can to reduce the bandwidth used. Video is extremely expensive because it's essentially a stream of images that are being sent across the network. We usually compress video.

Audio and images are substantially cheaper, but those also depend on how they're transferred, and if they're compressed or not.

Media may be compressed at different levels to reduce bandwidth usage. We may also want to compress multiple times at once for clients that are on lower bandwidth connections such as mobile viewers of an online conference.

There are also trade-offs to be considered depending on what media is being transmitted. Hiccups in the video are typically alright, but if audio keeps cutting it becomes a huge problem.

9.2 Streaming Stored Video

Streaming video is preferably done at a rate matching what the client is using to watch the video. There are a couple of ideas possible for streaming stored video:

- UDP Streaming: stream over UDP where video might be lost, but that can be made up for because of a video buffer. You also need a media control server to control interactivity (pause, play) between client and server. Additionally, a lot of firewalls block UDP traffic by default, making UDP a bad candidate for streaming.
- HTTP Streaming: stores the video on disk, allowing the client to access it through a URL. Server serves the video as quickly as TCP congestion flow allows it to. The player buffers the frames, allowing you to build the buffer if the transport rate is higher than the consumption rate. Most services uses this (YouTube, Netflix)

Pausing is also solved with HTTP streaming with the Content-Range HTTP header, allowing the client to quickly resume at any given "point" in the video file.

Dynamic Adaptive Streaming over HTTP (DASH) is a convention to solve sending video to lower bandwidth clients. The server has a manifest file that describes which versions of the video file are available, then the client can decide which one to use based on their bandwidth.

If everything is hosted from one server you will see a bottleneck at the server, which is why content delivery networks (CDNs) are used. The same files are available on all CDN servers, also reducing the distance between the video file from the client (as CDN servers are typically spread across multiple geographical areas)

9.3 Voice-over-IP

Voice-over-IP (VoIP) is real-time audio calls over the IP protocol. Services like Skype used VoIP (before they died). First, the parties establish a connection and encoding algorithm to be used, then they're able to send data to each other.

VoIP is very end-to-end delay-sensitive, as packet loss and network delays can hugely affect the quality of the call. Therefore anything less than 150ms is considered a great delay, while anything above 400ms becomes unintelligible. VoIP can tolerate between 1-20% packet loss. There are two primary ways to remove jitter at the receiver:

1. Fixed playout delay: all packets are played exactly q milliseconds after the chunk is generated. This means that there's a fixed delay of q milliseconds at the receiver end, but it allows for the network transmission to have some delays. Anything arriving after q will be discarded.
2. Adaptive playout delay: instead of using a fixed playout delay, you estimate the network delay, and determine an average packet delay. This only runs where there's a spurt of audio being transmitted, meaning different spurts may use different delays.

Content is typically chopped into segments before transmission. These chunks are error corrected with forwards error correction, creating a redundant chunk by XORing all n original chunks. This method can re-create all chunks given not more than 1 full chunk is lost, but there's a tiny overhead in decoding of course.

To conceal packet loss, chunks can be spread and deduplicated across multiple packets, meaning if a packet gets lost, the receiver will still have most of every chunk. (Interleaving)