Today: Video over Internet

- “Isochronous” transmission: transmit a continuous data over the Internet
    - Two PCs each with a microphone
    - Microphone: vibration -> voltage going up and down
    - Sampling: Voltage -> discrete samples
        - Why digital? Digital can be corrected.
        - Take discrete measurements of the analog signal. If the sampling rate is greater than twice of the largest frequency (if the signal occupies 0 to largest frequency), the samples can reconstruct the original signal perfectly.
    - Quantization: set a limited number of options and round off each sample to the nearest option -> a sequence of integers
    - Typical sampling rate for sound: 48 kSa/s (48000 sample / s)
    - Human ears’ dynamic range ~90 dB -> Typical quantization 16 bits (65, 536 levels)
    - Bitrate: 48 kSa /s * 2 byte / Sa = 96 kByte/s
- One design: 48,000 packets/second and each packet = 2 byte sample + 8 byte UDP header + 20 byte IP header + Ethernet header + preamble. A lot of overhead
- Another design: 48 packets/second and each packet = 2000 byte payload
    - The penalty for droppint a packet becomes higher
    - Each packet takes longer to transmit
    - Take 100 samples before sending a packet
    - **Latency Increases**
- 50 packets / second, each of 2 KB.
- Each packet contains 20ms of audio. Ideally, the receiver receives a new packet, starts to play the audio and when it is about to run out of the last packet, it receives a new one.
    - In other words, this is an “elasticity buffer”
- In a perfect world where the time it takes for a packet to travel from one end to the other is the same, this works. But, there is a variance in travel time == **“Jitter”**
- How to solve jitters?
    - Wait longer before it starts playing the audio. (Just like the receiver of the elasticity buffer waits for more bytes to start draining)
    - But again, **Latency Increases**
- In live video, there is a microphone and a camera attached to each end.
    - How does contiguous video become discrete data?
        - Originally By Eadweard Muybridge: continuous video -> frames of photos
    - Camera: continuous light coming in -> frames of snapshot of light
        - For human eyes, each frame is around 1920 columns and 1080 rows of pel/”picture element”/pixel
        - Each picture element, in practice, is 3 illuminants. Each illuminant is 8 bits.
        - You need a sample rate of 30 Hz
        - This is 1.5 Gbit/s. (This is a lot)
    - In practice, the frames are compressed

  - Intraframe compression (JPEG) “I” - “I pictures”
  - Predictive frames: utilize the similarity between neighboring frames. “P” - “P pictures”
  - These compression techniques get 1.5 Gbit/s => 3 Mbit/s
- The price:
  - More compute (hardware encoder)
  - Less accurate to reality
  - Less predictable
    - The compression outcome size varies based on whether neighboring frames are similar or not (e.g. for a large action scene, the outcome would be bigger)
    - Larger variance => Larger elasticity buffer
- What if a frame is missing:
  - If a P frame is missing, the next P frame and the next next P frame can’t be decoded
  - So you have to retransmit that P frame (this takes a while to realize a retransmission is needed)
  - Or restart on a new I frame (I frame is larger than P frame)
- How the upper-level application changes its behavior to the lower-level networking performance affects the overall performance of the system.
- We have been talking about interfaces, layers, and modularity across this lecture. But choosing the right level of modularity and the right interface is very important.
- These labs have been (hopefully) making a lot of sense, since they have been used in the real world for ~40 years, and through that process, people gradually found out the right modularity and interface of the networking stack. However, for newer applications (e.g. video) we don’t know yet.