Chapter 8: Multimedia

Network Nirvana

Illusion of sufficient capacity

Goal: Deploy enough link capacity so that congestion does not occur and multimedia traffic flows without delay or loss

Pros/Cons?

- Low complexity of network mechanisms but high bandwidth costs

Bandwidth provisioning

- Deploy the *right* amount of bandwidth
- How much bandwidth is "enough?"
- Depends on:
  - Traffic demands (aggregate)

Network dimensioning

- Deploy bandwidth in the right places
- How much faster should the *core* be with respect to the *edge*?
- Depends on
  - Network topology
  - Traffic demands (specific)
  - Communication patterns

Multimedia delivery

Voice

- Encoded by sampling at a constant rate
- 8kHz, 8-bit quantization implies 64Kbps

Video

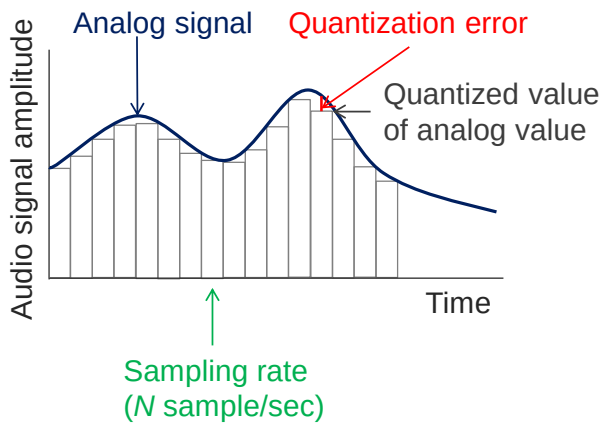
- PAL video format: 640x480 pixels, 24-bit quantization, 25 fps implies 184 Mbps

Requirements

- Need compression
- Face-to-face applications need end-to-end delay < 60 ms (compression and decompression included)
- Need random access to stored data

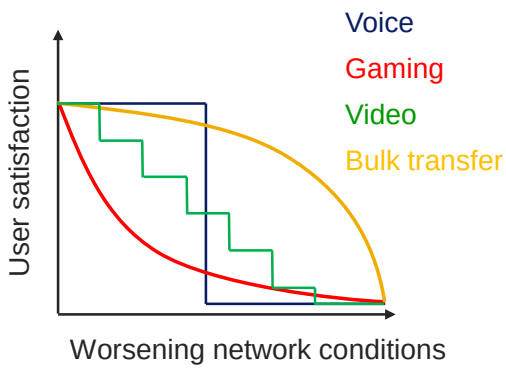
Audio stream encoding

- Analog audio signal sampled at constant rate
  - telephone: 8,000 samples/sec
  - CD music: 44,100 samples/sec
- Each sample quantized, i.e., rounded
  - e.g., 28=256 possible quantized values
  - Each quantized value represented by bits, e.g., 8 bits for 256 values
- Receiver converts bits back to analog signal with some losses
- Example rates
  - CD: 1.411 Mbps
  - MP3: 96, 128, 160 kbps
  - Internet telephony: 5.3 kbps and up



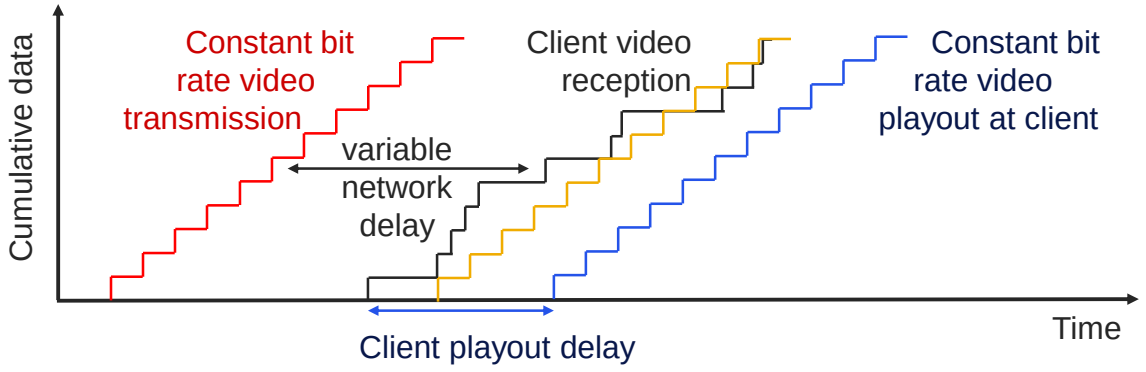
Quality of User Experience (QoE)

- User satisfaction



- Threats to quality
  - Latency
    - 5ms - cybesickness
    - 20ms – impact on game scores
    - 80ms – shortens Skype sessions
    - 200ms – speech turnaround interval
  - Jitter
  - Loss
  - Congestion

Effect of jitter on playout



What would happen if the yellow line was the playout schedule?

Client-side buffering and playout delay compensate for jitter

Compression Techniques

- Entropy Coding
  - Lossless encoding
  - Examples: run-length coding, Huffman coding, arithmetic coding
- Source Coding
  - Lossy encoding
  - Examples: content prediction technique – Difference PCM, Delta Modulation
- Hybrid Coding
  - Combine entropy coding with source coding
  - Examples: JPEG, JPEG-2000, H. 264, MPEG-2, MPEG-4, MPEG-7, MPEG-21

Entropy Encoding

- Run-length Encoding (RLE)
  - Multiple occurring bytes are grouped together as
    Number-Occurence Special-Character CompressedByte

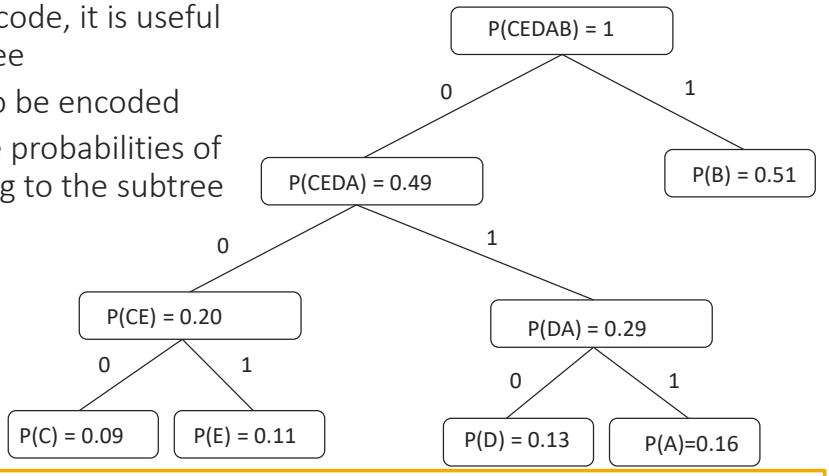
What is the compression ratio of AAAAAAAABBBCCCCCCCDD?

- Fixed-length encoding
  - Encode a message using N symbols.
  - Use binary numbers of equal length to represent each symbol by L bits ($L \geq \log_2(N)$) bits per symbol
  - What is L for $N = 5$ symbols?

Which encoding yields better compression, run-length coding or fixed-length coding for the string AAAAAAAABBBCCCCCCCDD?

Huffman Encoding

- Statistical encoding
- To determine Huffman code, it is useful to construct a binary tree
- Leaves are characters to be encoded
- Nodes carry occurrence probabilities of the characters belonging to the subtree



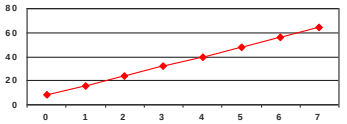
Huffman Code
Symbol A = 011
Symbol B = 1
Symbol C = 000
Symbol D = 010
Symbol E = 001

Encode AAAAAAAABBBCCCCCCCDD with a Huffman encoding

FFT and DCT

- DCT converts the information contained in a block(8x8) of pixels from ***spatial*** domain to the ***frequency*** domain
- A simple analogy:
  - Consider an unsorted list of 12 numbers between 0 and 3 -> (*2, 3, 1, 2, 2, 0, 1, 1, 0, 1, 0, 0*).
  - Consider a transformation of the list involving two steps (1.) sort the list (2.) Count the frequency of occurrence of each of the numbers ->(*4,4,3,1*).
  - Through this transformation we lost the spatial information but captured the frequency information
- Fast Fourier transform (FFT), Discrete Cosine Transform (DCT) retain spatial information to convert back and forth between spatial and frequency domains

Source



Transmission

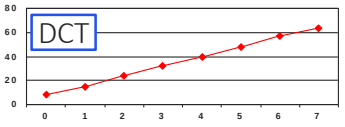
DCT

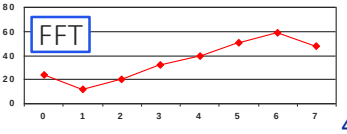
100	-52	0	-5

FFT

36	10	10	6

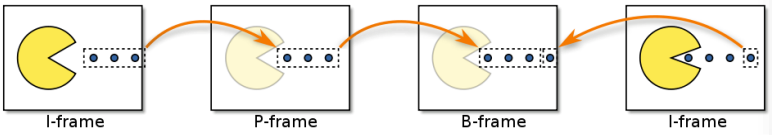
Decoded



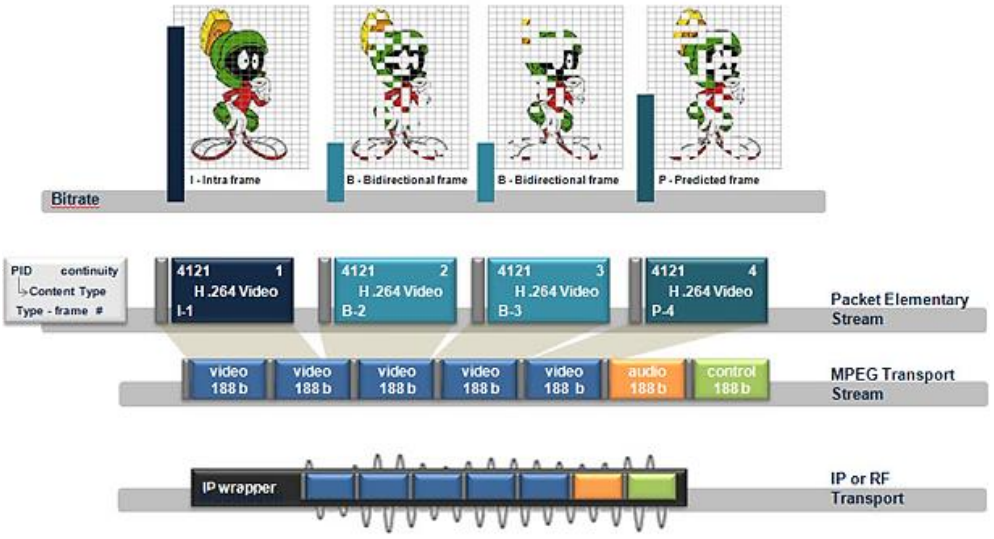


Video stream encoding

- Video: sequence of images displayed at constant rate
- Digital image: array of pixels (point color representations)
- Video encoding exploits redundancy between frames
  - Spatial – within an image
  - Temporal – between frames
- Variable rate based on image complexity and changes
  - MPEG2 (DVD) 3-6 Mbps
  - MPEG4 (< 1 Mbps)
- Frame types
  - **I**-frames – don't need other frames to display
  - **P**-frames – encode changes from previous frames
  - **B**-frames – can references forward changes

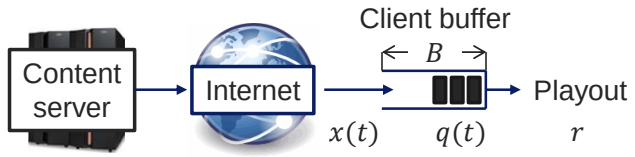


MPEG



Video playout

- Playout rate r is constant
- Streaming rate $x(t)$ varies over time
- Playout delayed until $q(t)$ exceeds some threshold
- What happens when?
  - $\bar{x} < r$
  - $\bar{x} \geq r$



- DASH: Dynamic Adaptive Streaming over HTTP
  - Video sent in chunks
  - Server advertises encoding rates and chunk `urls` through a *manifest*
  - Client selects chunk with encoding rate based on exponential moving average of $x(t)$