MultiMedia Networking

Created by: Omar Alkhateeb

For more MindMaps Check our Bitlo Website https://bitlo.devexple.com/

Image Representation

- **multimedia includes**
  - images
  - videos
  - text
  - sounds
- **Still Images** — generated by computers in 2 ways
  - **bitmaps**
    - photo-realistic images
    - complex drawings
    - fine detail
  - **vector-drawn objects**
    - lines
    - boxes
    - circles
    - polygons
    - other graphic shapes — expressed mathematically in
      - angles
      - coordinates
      - distances
    - used by
      - architects
      - engineers
      - graphic artists
- **Vector**
  - **definition** — a line that is described by the location of its two endpoints
  - **drawing**
    - to represent a vector we use **cartesian coordinates** — (x, y, z)
    - to draw we use Bezier curves or paths — to mathematically represent a curve
  - can be stored in a file called — **SVG** (Scaled Vector Graphics)
    - can be saved in a small amount of memory
    - and because they are scalable without distortion
  - **Advantages & Disadvantages**
    - low memory
    - faster downloads
    - Same quality for different resolutions
    - Refresh time lower with higher drawn objects
- **Bitmaps**
  - **definition**
    - The two-dimensional array of pixel values — that represents the graphics/image data
    - Image are broken up into a grid — recorded individually as a sequence of pixels
  - **Advantages & Disadvantages**
    - Works well for complex variations
    - Not flexible
    - Might need a lot of memory
    - Problems when scaled
  - famous formats
    - BMP
    - GIF
    - JPEG
    - TIFF
  - bitmaps as **matrices**
    - if still pictures aren't compressed — they are represented as bitmaps — a grid of pixels
    - **Resolution** — width x height
    - **monochrome images**
      - aka binary image
      - where each pixel is represented by 1 bit only — som it only shows B&W
    - **GrayScale images** — 8 bit per pixel — 255 level of gray
    - **Color Image**
      - 24-bit true color
        - 8 bit — for **R**ed
        - 8 bit — for **G**reen
        - 8 bit — for **B**lue
        - note — they are stored as 32 bit file — the extra 8 bit for **alpha** — which represents transparency
      - 8-bit pseudo color
- **Colors**
  - **Additive** Colors — Form a color by adding amounts of three primaries
    - Red
    - Green
    - Blue
  - **Subtractive** Colors — Form a color by filtering white light with
    - Cyan
    - Magenta
    - Yellow
  - **YUV** Color Space
    - **Y** — **luminance** — Y = 0.3R + 0.6G + 0.1B
    - **UV** — **chrominance**
      - **Cb** — **Chroma blue** — Cr = (B-Y)/2) + 0.5
      - **Cr** — **Chroma red** — Cb = (R-Y)/1.6) + 0.5
- **8-bit** Color Images
  - make use of 8 bits of color information — 256 colors
    - in producing a screen image
    - use the concept of a **lookup table**
      - to store color information
      - to store only the index — for each pixel
  - **Color picker** — here — each block of the color picker corresponds to one row of the color LUT
- **Popular File Formats**
  - **8-bit GIF**
    - first image type recognized by net browsers
    - consists of multiple images
    - has **color map** — the actual length of the table equals $2^{\text{pixel}+1}$
  - **JPEG**
    - currently the most important common file format
    - famous becomes it takes advantage of limitation of the human vision — so it achieves high levels of compression
  - **PNG**
    - stands for **P**ortable **N**etwork **G**raphics
    - Support for up to 48 bits of color information
    - Files may contain gamma-correction information
  - **TIFF**
    - stands for **T**agged **I**mage **F**ile **F**ormat
    - support for attachment of additional information
    - was lossless — but now can have jpeg compression

Image Compression

- **Introduction to Compression**
  - we reduce the volume of data to be exchanged — For Storage & Transmission
  - because image pixels are highly correlated — that leads to redundant information — Compression is about removing redundancy
- **Concepts**
  - **Bitrate** — bits/pixel
  - **Compression Ratio (CR)**
    - number of bits to represent the original image
    - number of bits in compressed bit stream
  - **Lossless** — original image can be exactly reconstructed
  - **Nearly lossless** — reconstructed image nearly (visually) lossless
  - **Lossy** — reconstructed image with loss of quality
- **Lossless Compression**
  - the integrity of the data is **preserved** — because the compression and decompression algorithms are exact **inverses** of each other — no part of the data is lost in the process
  - **Lossless compression methods**
    - **Run-Length Encoding (RLE)**
      - simplest method of removing redundancy
      - it replaces a repeated sequence — with count & the symbol
    - **Dictionary coding**
      - based on creation of a dictionary (array) of strings in the text
      - how it works
        - The dictionary is created as the message is scanned
        - and if a sequence of characters that is an entry in the dictionary is found in the message
        - the code (index) of that entry is sent instead of the sequence
    - **Huffman coding**
      - we encode data as binary patterns
      - we consider the frequency of symbols and the probability of their occurrence in the message
        - assigns shorter codes to symbols that occur more frequently
        - and longer codes to those that occur less frequently
      - **Huffman Tree**
        - is a tree in which the leaves of the tree are the symbols
          - It is made so that the most frequent symbol is the closest to the root
          - and the least frequent symbol is the farthest from the root
        - after the tree has been made — we can create a table that shows how each character can be encoded and decoded

Symbol	Code	Symbol	Code	Symbol	Code
A	00	C	011	E	11
B	010	D	10		

  - **Variable Word Length Coding** — assigns the shortest possible code words to the most probable gray levels
- **Transform-Based Compression**
  - Input image ($M \times N$) → Construct $n \times n$ subimages → Forward transform → Quantizer → Symbol encoder → Compressed image
    - in forward transform there is a **mapper** — transforms the input data — into a format designed to reduce interpixel redundancies in the input image
    - **quantizer** — reduces the accuracy of the mapper's output — This stage reduces the psychovisual redundancies of the input image — note: This operation is irreversible
    - **The symbol coder**
      - creates a fixed- or variable-length code to represent the quantizer output
      - and maps the output in accordance with the code
  - Compressed image → Symbol decoder → Inverse transform → Merge $n \times n$ subimages → Decompressed image
  - **Compression (encoder)**
  - **Decompression (Decoder)**
  - example — Three phases of JPEG: Blocked image → DCT → Quantization → Data compression → 0011101……000001 Compressed image
    - **JPEG** (Joint Photographic Expert Group)
    - Discrete Cosine Transform (DCT)
      - multiplication of a column matrix F — source
      - by a
      - square matrix T — DCT Coefficient
      - result is matrix M — transformed data
      - $$F(u) = \frac{C(u)}{2}\sum_{x=0}^{N-1} f(x)\cos\left(\frac{(2i+1)u\pi}{2N}\right)$$
      - If 1D
      - $$F(u,v) = \frac{2C(u)C(v)}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)v\pi}{2N}$$
      - a piece of an image — g(x, u, y, v) = g1(x,u).g2(y,v)
      - If 2D
    - **Quantization in DCT** — Uniform quantization is achieved by dividing DCT coefficients by N and round the result

Digital Audio

- **Characteristics of Sound**
  - Amplitude
    - low is quiet
    - high is loud
  - Wavelength
  - Frequency
    - low is low pitch
    - high is a high pitch
    - measured in hertz
  - Timbre
- **Digital Representation of Audio**
  - quantization error — quantized value of analog value — analog signal — audio signal amplitude — time — sampling rate (N sample/sec)
    - Sample
    - Quantize
    - Compress
  - Must convert wave form to digital
  - examples on sampling
    - if analog sample at constant rate
      - telephone — 8000 samples per second
      - CD music — 44100
    - each sample quantized (rounded)
  - according to **nyquist**
    - we need double highest rate freq of samples — so we lose nothing
    - max data rate R = 2 B log2 V bits/second
      - B = bandwidth (in Hz)
      - V = discrete levels (bits per signal change)
  - If we cannot sample fast enough
    - Reduce signal frequency to half of maximum sampling frequency
    - use low-pass filter — to remove higher frequencies
  - Continuous-time, continuous amplitude (analog) input signal → **PAM sampler** → Discrete-time continuous amplitude signal (PAM pulses) → **Quantizer** → Discrete-time discrete amplitude signal (PCM pulses) → **Encoder** → Digital bit stream output signal
    - **PCM** (Pulse Code Modulation)
    - Analog signal is quantized into a number of discrete levels
  - **SNR** = $10\log_{10}\left(\frac{\text{Signal energy}}{\text{Noise energy}}\right)$ — where signal energy is $A^2/2$
  - Data rate = sample rate * quantization * channel

Video Compression

- **Video**
  - sequence of images displayed at constant rate — Video file requires a high transmission rate
  - **CBR** (Constant Bit Rate) — video encoding rate fixed
  - **VBR** (Variable Bit Rate) — video encoding rate changes
  - **Temporal Compression** — Most of the frame is the same as the previous one
- **Chroma sub-sampling**
  - YUV 4:4:4 — 24 bits per pixel
  - YUV 4:2:2 — 16 bits/pixel
  - YUV 4:1:1 — 12 bits/pixel
- **Moving Picture Experts Group (MPEG)**
  - a method to compress video
  - defines exactly format — Luminance and two chrominance components — (2:1:1) — Pixel precision is 8 bits for each component
  - Each image is divided into **macro-blocks**
    - 16x16 pixels for luminance
    - 8x8 for each chrominance component
- **Spatial Compression**
  - Each frame is a picture that can be independently compressed
  - redundant frames are removed
- **Temporal Compression** — I B B P B B P B B I — MPEG divides frames into 3 types
  - **I-Frame** — intra frames — typically about 12 frames between I frames
  - **P-Frame** — Predictive frames — encode from previous I or P reference frame
  - **B-Frame** — Bi-directional frames — encode from previous and future I or P frames

Multimedia Streaming over IP Networks

- **3 application types** of media networking
  - **streaming, stored**
    - **streaming** — can begin playout before downloading entire file
    - **stored** — can transmit faster than audio/video will be rendered
  - **conversational** — voice/video over IP
  - **streaming live** — audio, video
- **challenges** — continuous playout constraint — may lead to **jitter**
  - Streaming stored video
  - network delays — so we will need client side buffer
- **DASH** (**D**ynamic, **A**daptive **S**treaming over **H**TTP)
  - Streaming Multimedia over HTTP
  - multimedia file retrieved via HTTP GET
  - send at maximum possible rate under TCP
  - **server**
    - divides video file into multiple chunks
    - each chunk stored — encoded at different rates
    - manifest file — provides URLs for different chunks
  - **client**
    - periodically measures server-to-client bandwidth
    - consulting manifest — requests one chunk at a time
      - chooses maximum coding rate sustainable given current bandwidth
      - can choose different coding rates at different points in time — (depending on available bandwidth at time
  - DASH can often achieve **continuous playout** at the best possible quality level without frame freezing or skipping
- **CDN** (Content Distribution Network)
  - **reason behind it** — how to **stream** content (selected from millions of videos) to hundreds of thousands of **simultaneous users?**
  - our options
    - single large **mega server**
      - single point of failure
      - point of network congestion
      - long path to distant clients
      - multiple copies of video sent over outgoing link
      - but it doesn't **scale!**
    - store/serve multiple copies of videos at multiple geographically distributed sites **"CDN"** — 2 approaches
      - enter deep — push CDN servers deep into many access networks
      - bring home — smaller number (10's) of larger clusters in POPs near (but not within) access networks
  - another challenge
    - how does CDN DNS select "good" CDN node to stream to client?
    - solution — let client decide - give client a list of several CDN servers — like netflix

Bitrates & Video Streaming

- this lecture IS Summerize it using **notebookLM**
- **Video Fundamentals**
  - pixel array — Digital Image
  - image sequence at a constant rate — video
  - coding (Redundancy)
    - within single image — spatial
    - between frames — temporal
- **Bitrate Factors**
  - Frames per second
  - Resolution
  - Compression degree
  - bandwidth restriction
- **CBR** (Constant Bit Rate)
  - Fixed encoding rate
  - predictable & reliable
  - faster encoding
  - best for live streaming
  - weaknesses
    - lower quality than VBR
    - inefficient bandwidth use
    - transient quality drops
- **VBR** (Variable Bit Rate)
  - Dynamic bit allocation
  - higher overall quality
  - efficient bandwidth use
  - best for video on-demand
  - subtypes
    - single pass — Quality VBR
    - multi-pass — unconstrained VBR
    - maximum bitrate — constrained VBR
  - weaknesses
    - slower encoding
    - higher latency
    - limited compatibility
- **Reducing File size**
  - new gen codecs
    - HEVC
    - VP9
    - AV1
  - transcoding tool presets
  - lowering resolution
  - reducing motion (tripods)
  - reducing framerate

QoS (Quality of Service)

- **IPTV** (Internet Protocol Television)
  - transmitting over **IP** networks — which is Digital television
  - **definition** — over IP based networks managed to provide the required level of — is multimedia services of delivering television/audio/text/graphics/data
    - Quality of Service (QoS)
      - **Bandwidth** — in order to manage bandwidth usage — we use some compression techniques — like MPEG2 & MPEG 4
      - **Packet Loss**
        - IP datagram lost due to network congestion — Network loss
        - IP datagram arrives too late for playout at receiver — Delay loss
        - from loss **distance** and loss **duration** — **PLR** (Packet Loss Ratio)
          - distance means spacing between consecutive packet loss
          - duration is the duration of a loss
      - **Delay**
      - **Jitter** — introduced in real-time data by the delay between packets — to prevent it we use timestamp
    - Quality of Experience (QoE)
      - Freeze
      - Blur
      - Noise
      - Color
      - Channel Change Time
    - security
    - interactivity
    - reliability
  - **Main Features**
    - Triple play
      - VoIP — voice
      - IPTV — video
      - internet services
    - interactive TV
    - Time Shifting
    - Personalization
    - Accessible on multiple devices
    - Quality of Services/Experience
    - & data protection — security
- managed vs unmanaged networks
  - **managed**
    - bandwidth allocation
    - contention ratio
    - control over
    - content
  - **unmanaged**
    - Variable bandwidth
    - Higher contention ratio
    - Less control over content
- **QoS mechanisms**
  - **at network level**
    - like buffering & scheduling — includes traffic management mechanisms
    - Checking that network resource are available before serving the use — admission control
  - At other levels other than the transport include
    - loss concealment
    - Forward Error Correction (FEC)
  - **at application level**
    - Checking that application resource are available before serving the user — Admission Control
    - Adding redundancy to data for FEC — FEC
    - Retransmitting lost data — RTP Retransmission
- Dimensioning **best effort** networks
  - partition traffic into classes
  - network treats different classes of traffic differently
  - discard policy
    - if packet arrives to full queue
    - drop arriving packet — **Tail drop**
    - drop/remove on priority basis — **priority** — who to discard?
    - drop/remove randomly — **random**
  - same as in OS — scheduling policies

Protocols for real-time streaming

- **Multimedia key characteristics**
  - uses extremely high data rates
  - requires real-time playback
  - Delay sensitive
    - end-to-end delay
    - delay jitter
  - Loss tolerant — infrequent losses cause minor glitches
- **Streaming stored video**
  - **challenges**
    - continuous playout constraint — once client playout begins, playback must match original timing — so you need client side buffer
    - client interactivity
      - pause
      - fast-forward
      - rewind
      - jump through video
    - video packets may be lost & retransmitted
  - **approaches**
    - **First Approach** — received in entirety at client then passed to player — long delays until playout
      - Using a **Web Server**
    - **Second Approach** — using metafile — player contact server and the streaming process is started
      - Using a Web Server with a **Metafile**
    - **Third Approach** — allows for non-HTTP protocol between server & the media player — UDP or TCP for step (5)
      - Using a **Media Server**
    - **Fourth Approach**
      - Using a **Media Server** and **RTSP**
  - **UDP or TCP**
    - **UDP**
      - server sends at rate appropriate for client — send rate = encoding rate = constant rate — fill rate = constant rate - packet loss
    - **TCP**
      - send at maximum possible rate under TCP
      - fill rate fluctuates due to TCP congestion control
      - larger playout delay — smooth TCP delivery rate
      - HTTP/TCP passes more easily through firewalls
- **RTSP** (Real-Time Streaming Protocol)
  - **Selt**
    - does not target multimedia content
    - no commands for fast forward
    - client-server application layer protocol — in RTSP: RFC 2326
    - user control
      - rewind
      - fast forward
      - pause
      - resume
      - repositioning
  - what it **doesn't** do
    - doesn't define how audio/video is encapsulated for streaming over network
    - doesn't restrict how streamed media is transported (UDP or TCP possible)
    - doesn't specify how media player buffers audio/video
  - **out of band** control
    - **FTP** (File Transfer Protocol) uses an "out-of-band" control channel
      - file transferred over one TCP connection
      - control info
        - directory changes
        - file deletion
        - rename
      - sent over separate TCP connection
    - port 554 — use different port numbers
    - media stream is considered in-band
- Real-Time interactive applications examples
  - multi-user network games
  - streaming stored video clips
  - social network
  - Voice over IP
  - Internet TV
  - IPTV
  - real-time video conference
- Why real-time data can't use TCP?
  - TCP forces the receiver application to wait for retransmission
  - TCP cannot support multicast
  - TCP congestion control mechanisms decreases the congestion window when packet losses are detected — slow start
  - TCP headers are larger than a UDP header
  - TCP doesn't contain the necessary timestamp — encoding needed by the receiving application
  - TCP doesn't allow packet loss
- **Translation & Mixing**
  - **Translation** — means changing the encoding of a payload to a lower quality to match the bandwidth of the receiving network
  - **Mixing** — means combining several streams of traffic into one stream
- **RTP** (Real-time Transport Protocol)
  - is the protocol designed to handle real-time traffic on the Internet
  - does not have a delivery mechanism
  - so it extends it — must be used with UDP
  - stands between UDP and the application program
  - The main contributions of RTP are
    - time-stamping
    - sequencing
    - mixing facilities
  - **Packet Format**

Ver	P	X	Contr. count	M	Payload type	Sequence number
Time stamp						
Synchronization source identifier						
Contributor identifier						
⋮						
Contributor identifier						

  - User Control of Streaming Media

Voice-over-IP Networks

- **VoIP** (Voice over Internet Protocol)
  - technology that enables one to make and receive phone calls through the Internet — instead of using the traditional analog PSTN lines
  - needed to maintain **conversational** aspect — VoIP **end-end-delay requirement**
    - impair interactivity — higher delays noticeable
    - good — < 150 msec
    - bad — > 400 msec
  - includes **application-level** network **delays** — like
    - packetization
    - playout
  - **Audio**
    - analog audio signal sampled at constant rate — each sample quantized (rounded)
    - 64 kbps during this
  - **characteristics**
    - speaker's audio — alternating talk spurts
      - packets only generate during these
      - 20 msec chunks at 8 Kbytes/sec: 160 bytes of data
    - silent periods
    - application-layer header added to each chunk
    - chunk+header encapsulated into UDP or TCP segment
    - application sends segment into socket every 20 msec during talk spurt
  - **packet loss and delay**
    - **network loss** — IP datagram lost due to network — router buffer overflow
    - **delay loss** — IP datagram arrives too late for playout at receiver — anything more than 400 ms is discarded
    - **loss tolerance** — packet loss rates between 1% and 10% can be tolerated

	Loss	Delay
Multimedia	tolerant	sensitive
Data	intolerant	tolerant

    - between media & data
  - **Jitter**
    - **definition** — time from when a packet is generated at the source until it is received at the receiver can **fluctuate** from packet to packet
    - How to prevent it?
      - To prevent jitter, we can time-stamp the packets and separate the arrival time from the playback time
      - loss packet loss — large q
      - better interactive experience — small q
      - receiver attempts to playout each chunk exactly q msecs after chunk was generated
      - so estimate the network delay and the variance of the network delay — but we need to make the playout **adaptive** instead of fixed
      - $$\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\left(\sum_{i=1}^{n-1} x_i + x_n\right) = \frac{1}{n}\left(\frac{n-1}{n-1}\sum_{i=1}^{n-1} x_i + x_n\right)$$
      - $$= \frac{n-1}{n}\mu_{n-1} + \frac{1}{n}x_n = (1-\alpha)\mu_{n-1} + \alpha x_n$$
      - which can be computed like this
      - note problems and its calculations will be written in another file (if I had the time to do it)
      - EstimatedRTT = (1 - a)*EstimatedRTT + a*SampleRTT