CS525z
Multimedia Networking

Introduction

Introduction Purpose

- Brief introduction to:
  - Digital Audio
  - Digital Video
  - Perceptual Quality
  - Network Issues
  - The "Science" (or lack of) in "Computer Science"
- Get you ready for research papers!
- Introduction to:
  - Silence detection (for project 1)

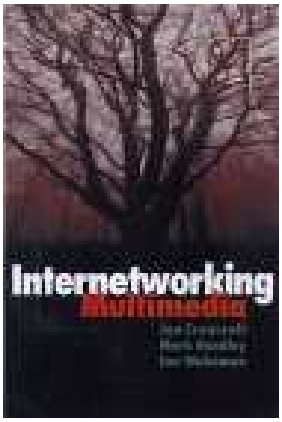
Groupwork

- Let's get started!
- Consider audio or video on a computer
  - Examples you have seen, or
  - Guess how it might look
- What are two conditions that degrade quality?
  - Giving technical name is ok
  - Describing appearance is ok

Introduction Outline

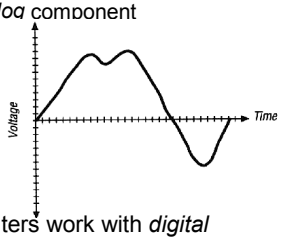
- Background
  - ***Internetworking Multimedia (Ch 4)***
  - Graphics and Video (Linux MM, Ch 4)
  - Multimedia Networking (Kurose, Ch 6)
- Audio Voice Detection (Rabiner)
- MPEG
  - Fitzek and Reisslein intro
  - Le Gall
- Misc

(These Slides)



Digital Audio

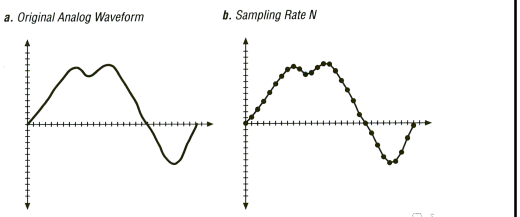
- Sound produced by variations in air pressure
  - Can take any continuous value
  - *Analog* component



- Computers work with *digital*
  - Must convert analog to digital
  - Use *sampling* to get discrete values

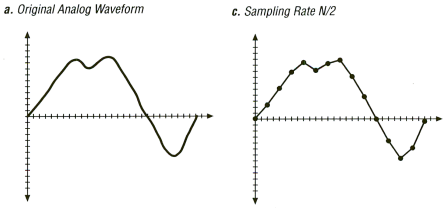
Digital Sampling

- *Sample rate* determines number of discrete values



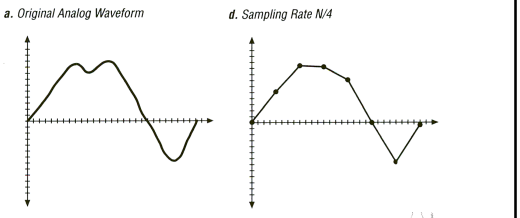
Digital Sampling

- Half the sample rate



Digital Sampling

- Quarter the sample rate

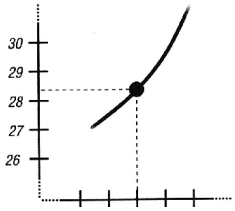


Sample Rate

- Nyquist's Theorem: to accurately reproduce signal, must sample at twice the highest frequency
- Why not always use high sampling rate?
  - Requires more storage
  - Complexity and cost of analog to digital hardware
  - Human's can't always perceive
    - Dog whistle
  - Typically want an *adequate* sampling rate

Sample Size

- Samples have discrete values



- How many possible values?
  - *Sample Size*
  - Common is 256 values from 8 bits

Sample Size

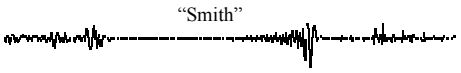
- *Quantization error* from rounding
  - Ex: 28.3 rounded to 28
- Why not always have large sample size?
  - Storage increases per sample
  - Analog to digital hardware becomes more expensive

Groupwork

- Think of as many uses of computer audio as you can
- Which require a high sample rate and large sample size? Which do not? Why?

Audio

- Encode/decode devices are called *codecs*
  - Compression is the complicated part
- For voice compression, can take advantage of speech:



- Many similarities between adjacent samples
  - Send differences (μ-law)
  - Adapt to signal (ADPCM)
- Use understanding of speech
  - Can 'predict' (CELP)

Audio by People

- Sound by breathing air past vocal cords
  - Use mouth and tongue to shape vocal tract
- Speech made up of phonemes
  - Smallest unit of distinguishable sound
  - Language specific
- Majority of speech sound from 60-8000 Hz
  - Music up to 20,000 Hz
- Hearing sensitive to about 20,000 Hz
  - Stereo important, especially at high frequency
  - Lose frequency sensitivity as age

Typical Encoding of Voice

- Today, telephones carry digitized voice
- 4 KHz (8000 samples per second)
  - Adequate for most voice communication
- 8-bit sample size
- For 10 seconds of speech:
  - 10 sec x 8000 samp/sec x 8 bits/samp
    = 640,000 bits or 80 Kbytes
  - Fit 3 minutes of speech on a floppy disk
  - Fit 2 weeks of sound on typical hard disk
- Fine for voice, but what about music?

Typical Encoding of Audio

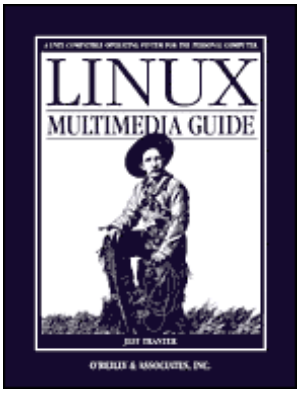
- Can only represent 4 KHz frequencies (why?)
- Human ear can perceive 10-20 KHz
  - Used in music
- CD quality audio:
  - sample rate of 44,100 samples/sec
  - sample size of 16-bits
  - 60 min x 60 secs/min x 44,100 samp/sec
    x 2 bytes/samples x 2 channels
    = 635,040,000, about 600 Mbytes (typical CD)
- Can use *compression* to reduce
  - mp3, RealAudio

Sound File Formats

- Raw data has samples (interleaved w/stereo)
- Need way to 'parse' raw audio file
- Typically a header
  - Sample rate
  - Sample size
  - Number of channels
  - Coding format
  - …
- Examples:
  - .au for Sun μ-law, .wav for IBM/Microsoft

Introduction Outline

- Background
  - Internetworking Multimedia (Ch 4)
  - ***Graphics and Video (Linux MM, Ch 4)***
  - Multimedia Networking (Kurose, Ch 6)
- Audio Voice Detection (Rabiner)
- MPEG
  - Fitzek and Reisslein intro
  - Le Gall
- Misc

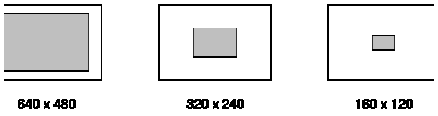


Graphics and Video
"A Picture is Worth a Thousand Words"

- People are visual by nature
- Many concepts hard to explain or draw
- Pictures to the rescue!
- Sequences of pictures can depict motion
  - Video!

Video Images

- Television about 6000 lines, 4:3 aspect ratio
  - 833x625 (PAL), 700x525 (NTSC)
- Digital video smaller
  - 352x288 (H.261), 176x144 (QCIF)



- Monitors higher resolution than T.V.
  - 1200x1000 pixels not uncommon
- Computer video often called "Postage Stamp"

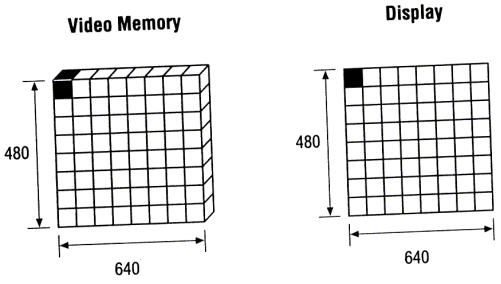
Video Image Components

- Luminance (Y) and Chrominance: Hue (U) and Intensity (V)
  - Human eye less sensitive to color than luminance, so those sampled at less resolution
- YUV is for backward compatibility with BW televisions (only had Luminance)
  - Monitors are typically RGB

Graphics Basics

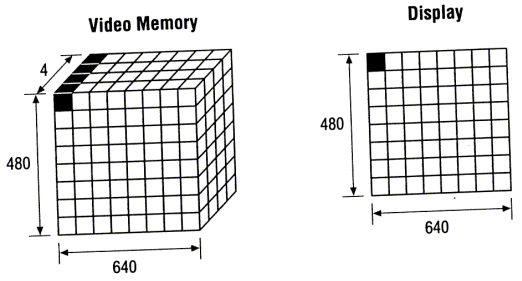
- Display images with graphics hardware
- Computer graphics (pictures) made up of pixels
  - Each pixel corresponds to region of memory
  - Called *video memory* or *frame buffer*
- Write to video memory
  - monitor displays with raster cannon

Monochrome Display



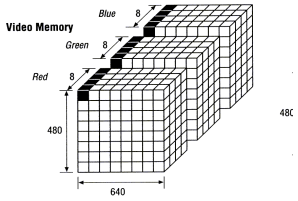
- Pixels are on (black) or off (white)
  - *Dithering* can appear gray

Grayscale Display



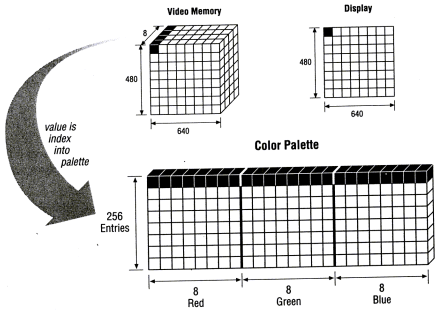
- *Bit-planes*
  - 4 bits per pixel, 2^4 = 16 gray levels

Color Displays



- Humans can perceive far more colors than grayscales
  - Cones (color) and Rods (gray) in eyes
- All colors seen as combination of red, green and blue
- Max needed
  - 24 bits/pixel, 2^{24} = 16 million colors (true color)
- But now requires 3 bytes required per pixel

Video Palettes



- Still have 16 million colors, only 256 at a time
- Complexity to lookup, color flashing
- Can dither for more colors, too

Graphics Summary

Display Type	Bits Per Pixel	Colors	Resolution	Video Memory
monochrome	1	2 (black and white)	640x480	38 KB
grayscale	4	16 shades of gray	640x480	150 KB
color	24	16 million	640x480	900 KB
color with palette	8	256 from palette of 16 million	640x480	301 KB
monochrome	1	2 (black and white)	1024x768	96 KB
grayscale	4	16 shades of gray	1024x768	384 KB
color	24	16 million	1024x768	2.3 MB
color with palette	8	256 from palette of 16 million	1024x768	769 KB

- xdpyinfo, display→settings

Moving Video Images (Guidelines)

- Unit is Frames Per Second (fps)
- 24-30 fps: full-motion video
- 15 fps: full-motion video approximation
- 7 fps: choppy
- 3 fps: very choppy
- Less than 3 fps: slide show

Moving Video Images

- Series of frames with changes appear as motion (say, 30 fps)

Time:Size	640x480	320x240	160x120
1sec	27Mb	6.75Mb	1.68Mb
1min	1.6Gb	400Mb	100Mb
1hour	97Gb	24Gb	6Gb
1000hours	97Tb	24Tb	6Tb

Uncompressed video is enormous!

Video Compression

Time v. Scale	None	3:1	25:1 (JPEG)	100:1 (MPEG)
1 sec	27 Mb	9 Mb	1.1 Mb	270 Kb
1 min	1.6 Gb	540 Mb	65 Mb	16 Mb
1 hour	97 Gb	32 Gb	3.9 Gb	970 Mb

640x480

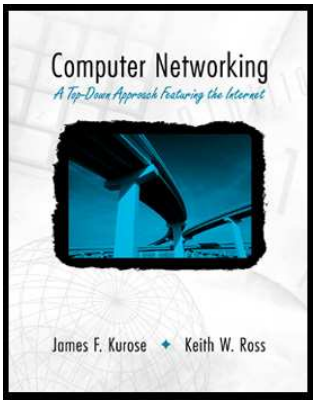
Time v. Scale	None	3:1	25:1 (JPEG)	100:1 (MPEG)
1 sec	6.75 Mb	2.25 Mb	270 Kb	68 Kb
1 min	400 Mb	133 Mb	16 Mb	4 Mb
1 hour	24 Gb	8 Gb	1 Gb	240 Mb

320x240

- Lossless or Lossy
- Intracoded or Intercoded
  - Take advantage of dependencies between frames
    - Motion
- (More on MPEG later)

Introduction Outline

- Background
  - Internetworking Multimedia (Ch 4)
  - Graphics and Video (Linux MM, Ch 4)
  - ***Multimedia Networking (Kurose, Ch 6)***
- Audio Voice Detection (Rabiner)
- MPEG
  - Fitzek and Reisslein intro
  - Le Gall
- Misc



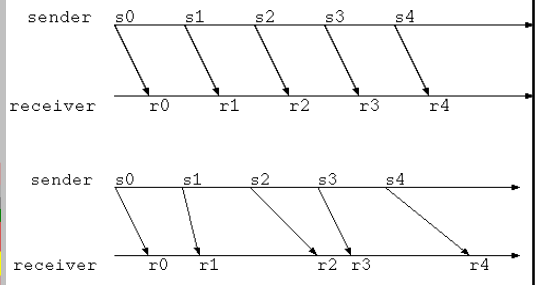
Internet Traffic Today

- Internet dominated by text-based applications
  - Email, FTP, Web Browsing
- Very sensitive to loss
  - Example: lose a byte in your `blah.exe` program and it crashes!
- Not very sensitive to delay
  - 10's of seconds ok for web page download
  - Minutes for file transfer
  - Hours for email to delivery

Multimedia on the Internet

- Multimedia not as sensitive to loss
  - Words from sentence lost still ok
  - Frames in video missing still ok
- Multimedia can be very sensitive to delay
  - Interactive session needs one-way delays less than 1 second!
- New phenomenon is jitter!

Jitter



Jitter-Free

Classes of Internet Multimedia Apps

- Streaming stored media
- Streaming live media
- Real-time interactive media

Streaming Stored Media

- Stored on server
- Examples: pre-recorded songs, famous lectures, video-on-demand
- *RealPlayer, Media Player and Quicktime*
- Interactivity, includes pause, ff, rewind…
- Delays of 1 to 10 seconds or so
- Not so sensitive to jitter

Streaming Live Media

- "Captured" from live camera, radio, T.V.
- 1-way communication, maybe multicast
- Examples: concerts, radio broadcasts, lectures
- *RealPlayer, Media Player* and *Quicktime*
- Limited interactivity…
- Delays of 1 to 10 seconds or so
- Not so sensitive to jitter

Real-Time Interactive Media

- 2-way communication
- Examples: Internet phone, video conference
- Very sensitive to delay
  - < 150ms very good
  - < 400ms ok
  - > 400ms crappy

Hurdles for Multimedia on the Internet

- IP is best-effort
  - No delivery guarantees
  - No bandwidth guarantees
  - No timing guarantees
- So … how do we do it?
  - Not too well for now
  - This class is largely about techniques to make it better!

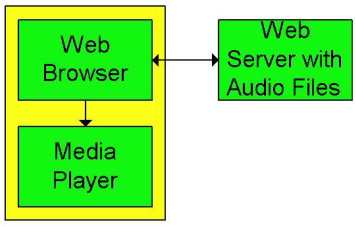
Multimedia on the Internet

- The Media Player
- Streaming through the Web
- The Internet Phone Example

The Media Player

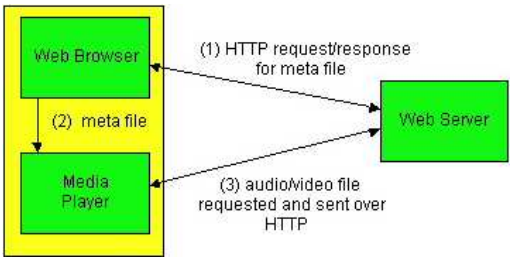
- End-host application
  - Real Player, Windows Media Player
- Needs to be pretty smart
- Decompression (MPEG)
- Jitter-removal (Buffering)
- Error correction (Repair)
- GUI with controls (HCI issues)
  - Volume, pause/play, sliders for jumps

Streaming through a Web Browser



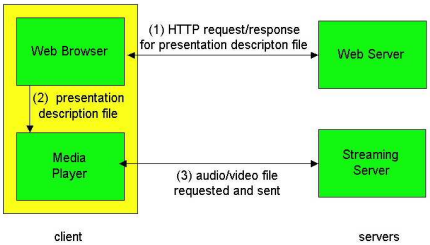
Must download whole file first!

Streaming through a Plug-In



Must still use TCP!

Streaming through the Media Player



An Example: Internet Phone

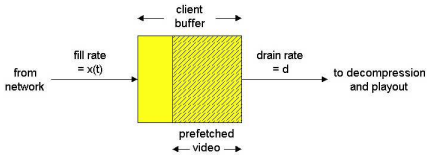
- Specification
- Removing Jitter
- Recovering from Loss

Internet Phone: Specification

- 8 Kbytes per second, send every 20 ms
  - 20 ms * 8 kbytes/sec
  - = 160 bytes per packet
- Header per packet
  - Sequence number, time-stamp, playout delay
- End-to-End delay of 150 – 400 ms
  - (So, why isn't TCP effective?)
- UDP
  - Can be delayed different amounts (Need to remove Jitter)
  - Can be lost (Need to recover from Loss)

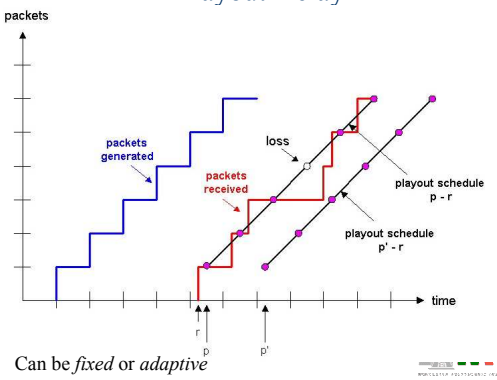
Internet Phone: Removing Jitter

- Use header information to reduce jitter
  - Sequence number and Timestamp



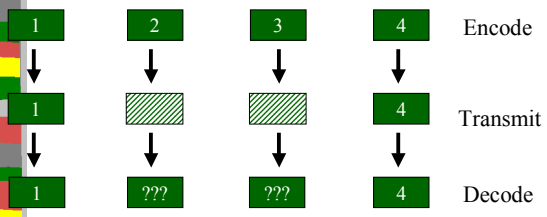
- Strategy:
  - –Playout delay (Delay Buffer)

Playout Delay



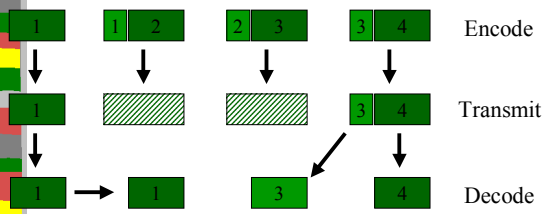
Can be *fixed* or *adaptive*

Internet Phone: Loss



What do you do with the missing packets?

Internet Phone: Recovering from Loss



Projects

- Project 1:
  - Read and Playback from audio device
  - Detect Speech and Silence
  - Evaluate (1a)
- Project 2:
  - Build an Internet Phone application
  - Evaluate (2b)
- Project 3:
  - Multi-person Internet Phone via multicast
  - Evaluate (3b)