CS525u
Multimedia Computing

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

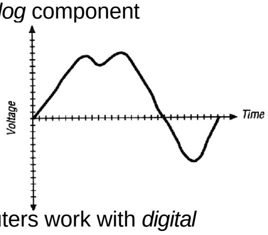
- Let’s get started!
- Consider audio or video on a computer
  - Examples you have seen, or
  - Guess how it might look
- What are two conditions that degrade quality?
  - Giving technical name is ok
  - Describing appearance is ok

Introduction Outline

- Background
  - ***<u>Digitial Audio (Linux MM, Ch2)</u>***
  - Graphics and Video (Linux MM, Ch4)
  - Multimedia Networking (Kurose, Ch6)
- Audio Voice Detection (Rabiner)
- MPEG (Le Gall)
- Misc

Digital Audio

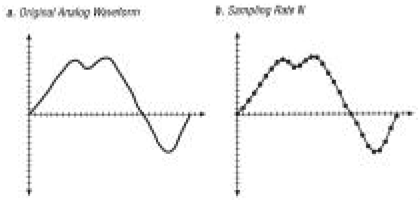
- Sound produced by variations in air pressure
  - Can take any continuous value
  - *Analog* component

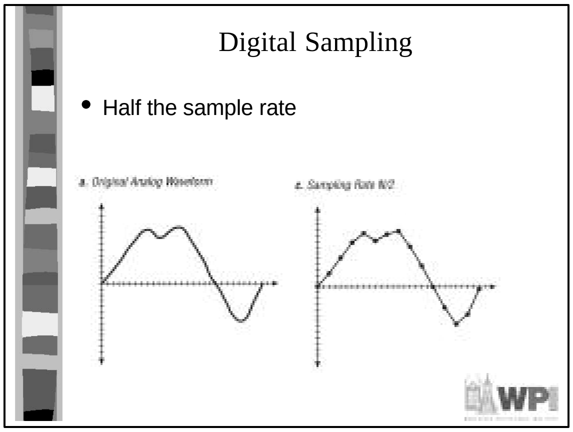


- Computers work with *digital*
  - Must convert analog to digital
  - Use *sampling* to get discrete values

Digital Sampling

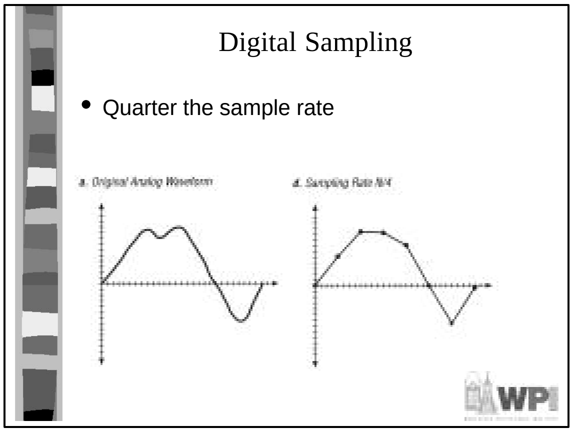
- *Sample rate* determines number of discrete values





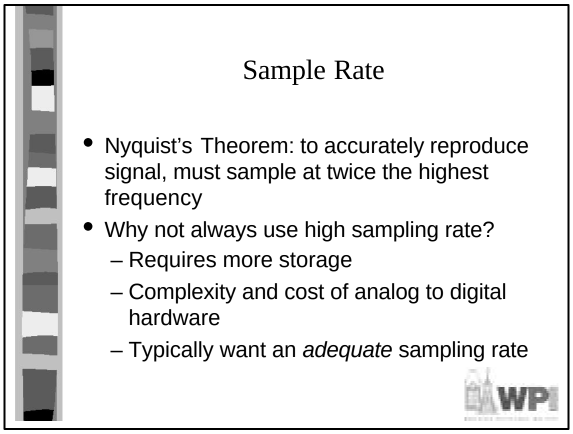
Digital Sampling

- Half the sample rate



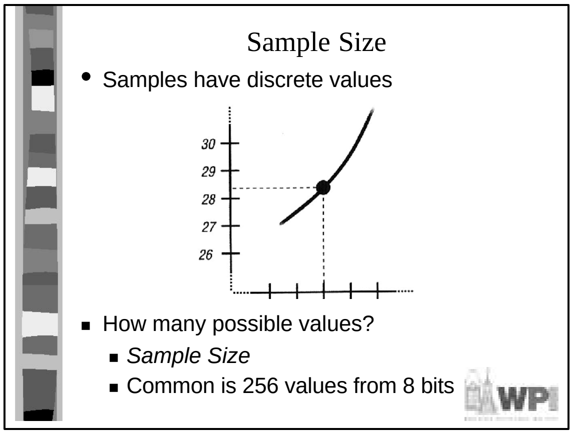
Digital Sampling

- Quarter the sample rate



Sample Rate

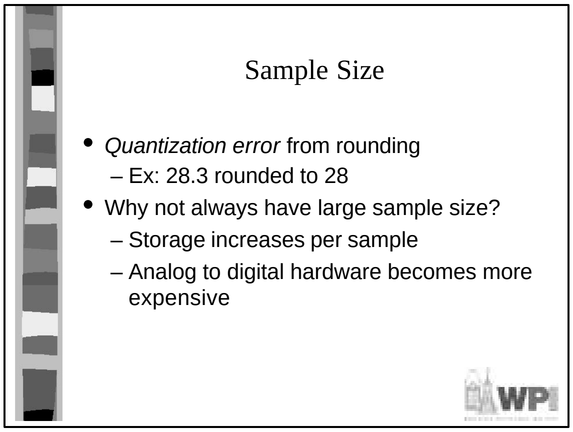
- Nyquist's Theorem: to accurately reproduce signal, must sample at twice the highest frequency
- Why not always use high sampling rate?
  - Requires more storage
  - Complexity and cost of analog to digital hardware
  - Typically want an *adequate* sampling rate



Sample Size

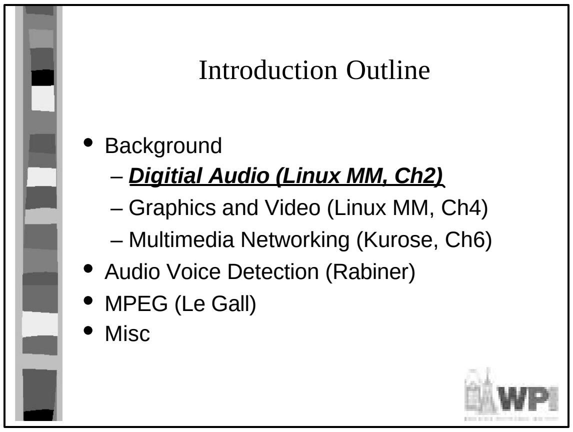
- Samples have discrete values

- 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



Introduction Outline

- Background
  - ***<u>Digitial Audio (Linux MM, Ch2)</u>***
  - Graphics and Video (Linux MM, Ch4)
  - Multimedia Networking (Kurose, Ch6)
- Audio Voice Detection (Rabiner)
- MPEG (Le Gall)
- Misc

Review

- What is the relationship between samples and fidelity?
  - Why not always have a high sample frequency?
  - Why not always have a large sample size?

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?

Back of the Envelope Calculations

- Telephones typically carry digitized voice
- 8 KHz (8000 samples per second)
- 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 on floppy
- Fine for voice, but what about music?

More Back of the Envelope Calculations

- 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 or about 600 Mbytes
- Can use *compression* to reduce

Audio Compression

- Above sampling assumed linear scale with respect to intensity
- Human ear not keen at very loud or very quiet
- *Companding* uses modified logarithmic scale to greater range of values with smaller sample size
  - µ-law effectively stores 12 bits of data in 8-bit sample
  - Used in U.S. telephones
  - Used in Sun computer audio
  - MP3 for music

MIDI

- Musical Instrument Digital Interface
  - Protocol for controlling electronic musical instruments
- MIDI message
  - Which device
  - Key press or key release
  - Which key
  - How hard (controls volume)
- MIDI file can play ‘song’ to MIDI device

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

Example Sound Files

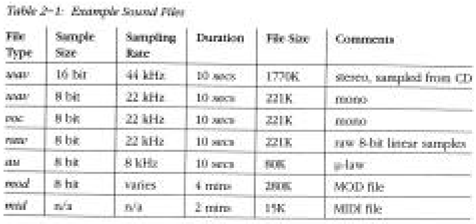


Table 2–1: Example Sound Files

File Type	Sample Size	Sampling Rate	Duration	File Size	Comments
wav	16 bit	44 kHz	10 secs	1770K	stereo, sampled from CD
wav	8 bit	22 kHz	10 secs	221K	mono
voc	8 bit	22 kHz	10 secs	221K	mono
raw	8 bit	22 kHz	10 secs	221K	raw 8-bit linear samples
au	8 bit	8 kHz	10 secs	80K	µ-law
mod	8 bit	varies	4 mins	280K	MOD file
mid	n/a	n/a	2 mins	15K	MIDI file

Outline

- Introduction
  - – Digital Audio (Linux MM, Ch2)
  - – ***Graphics and Video (Linux MM, Ch4)***
  - – Multimedia Networking (Kurose, Ch6)
- Audio Voice Detection (Rabiner)
- MPEG (Le Gall)
- Misc

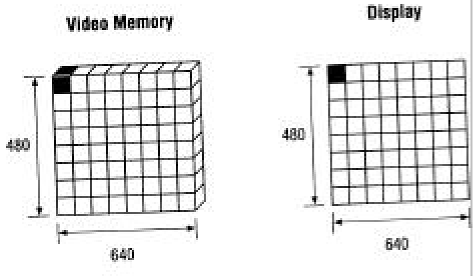
Graphics and Video
“A Picture is Worth a Thousand Words”

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

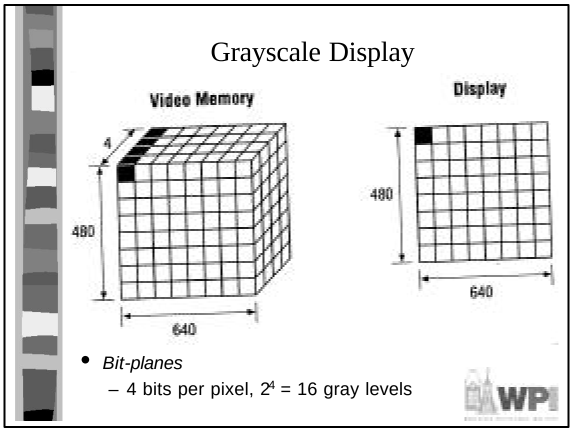
Graphics Basics

- 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

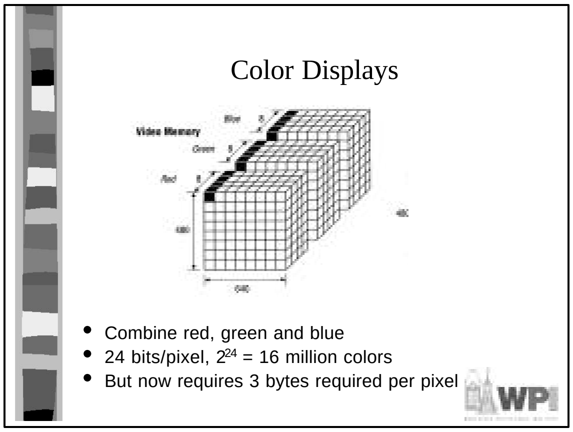


- Pixels are on (black) or off (white)
  - – *Dithering* can appear gray



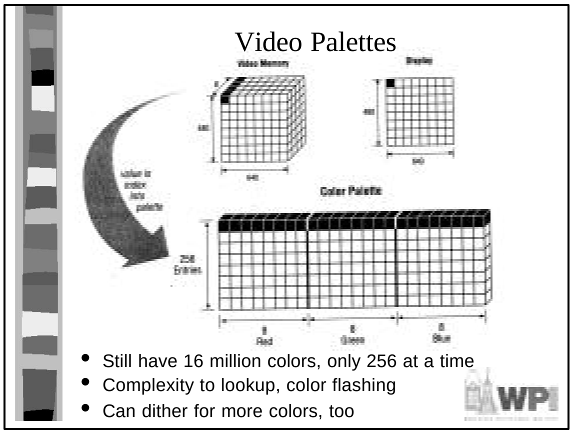
Grayscale Display

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



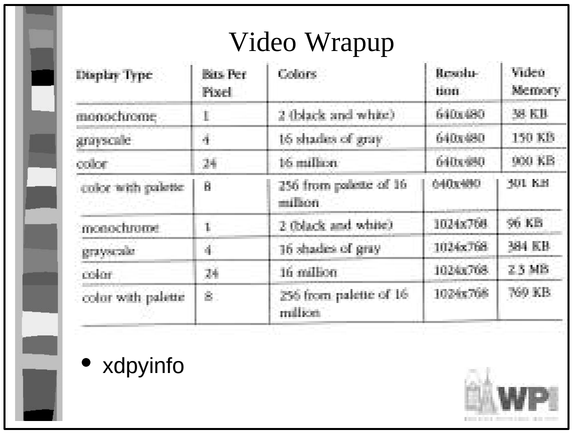
Color Displays

- Combine red, green and blue
- 24 bits/pixel, 2^{24} = 16 million colors
- But now requires 3 bytes required per pixel



Video Palettes

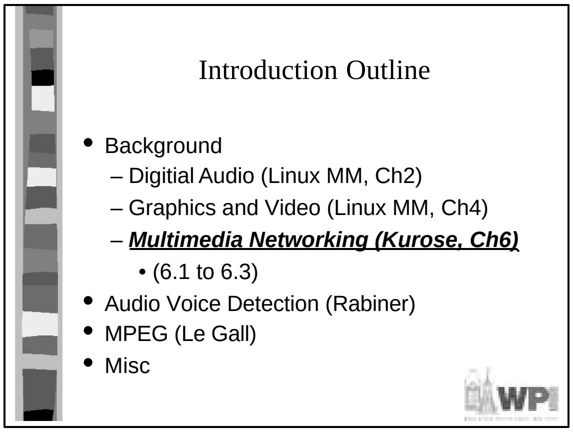
- Still have 16 million colors, only 256 at a time
- Complexity to lookup, color flashing
- Can dither for more colors, too



Video Wrapup

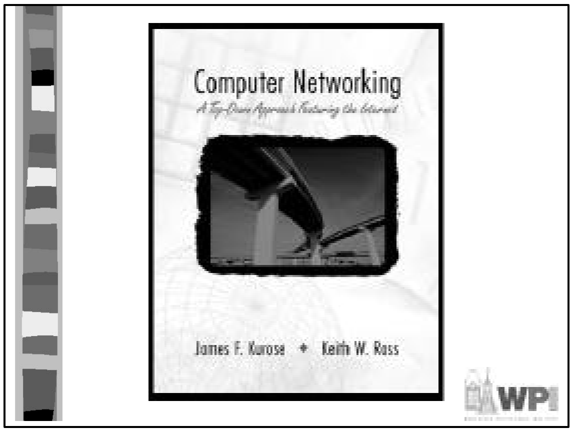
Display Type	Bits Per Pixel	Colors	Resolu-tion	Video Memory
monochrome	1	2 (black and white)	640x480	38 KB
grayscale	4	16 shades of gray	640x480	150 KB
color	24	16 million	640x480	900 KB
color with palette	8	256 from palette of 16 million	640x480	301 KB
monochrome	1	2 (black and white)	1024x768	96 KB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



Introduction Outline

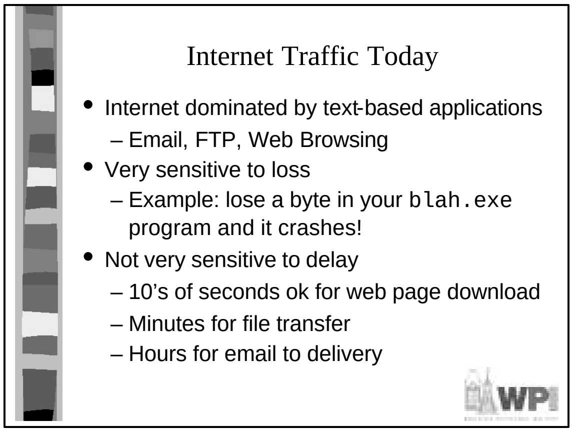
- Background
  - Digitial Audio (Linux MM, Ch2)
  - Graphics and Video (Linux MM, Ch4)
  - ***Multimedia Networking (Kurose, Ch6)***
    - (6.1 to 6.3)
- Audio Voice Detection (Rabiner)
- MPEG (Le Gall)
- Misc



Computer Networking

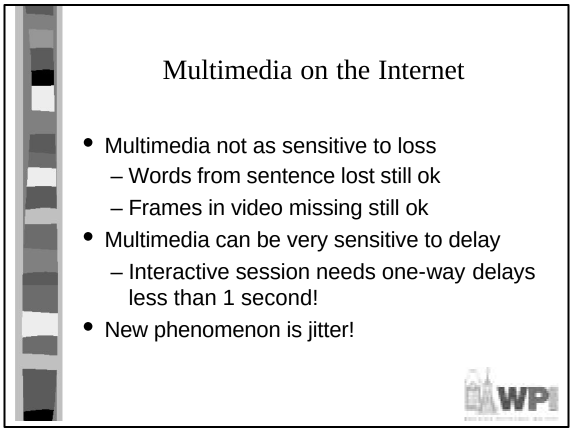
A Top-Down Approach Featuring the Internet

James F. Kurose ✦ Keith W. Ross



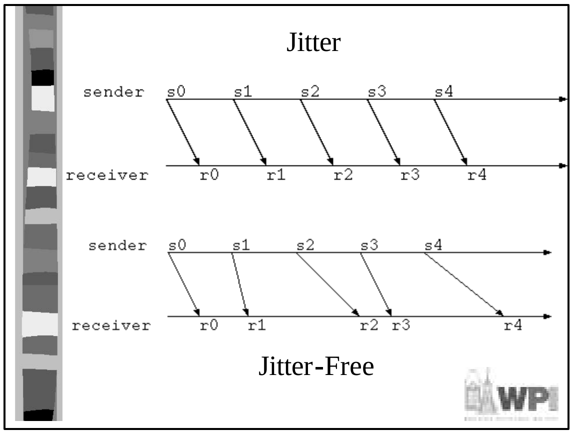
Internet Traffic Today

- 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



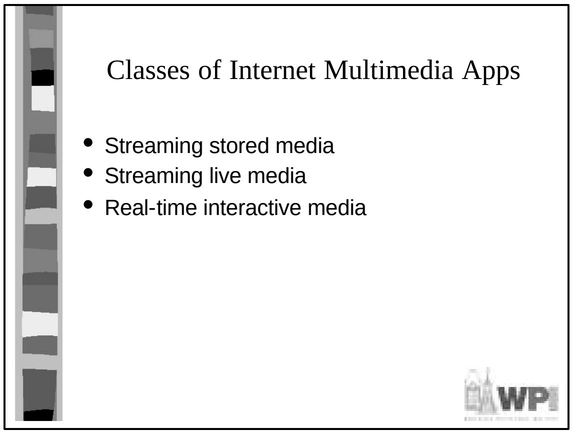
Multimedia on the Internet

- 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!



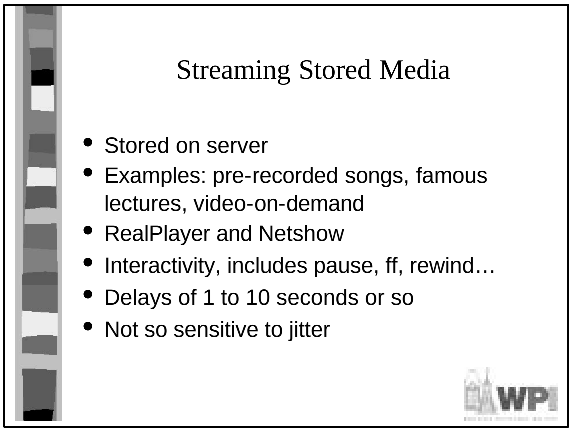
Jitter

Jitter-Free



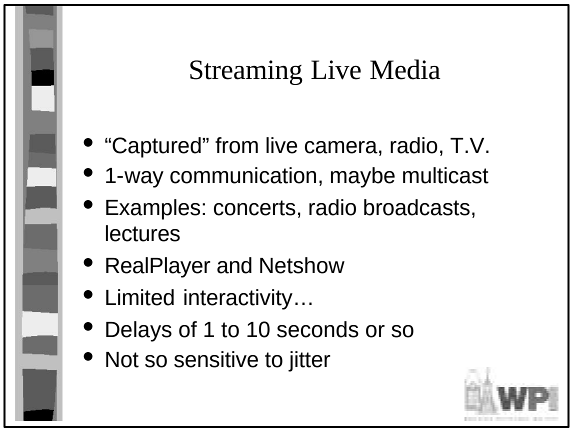
Classes of Internet Multimedia Apps

- 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 and Netshow
- 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 and Netshow
- 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 lousy

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!

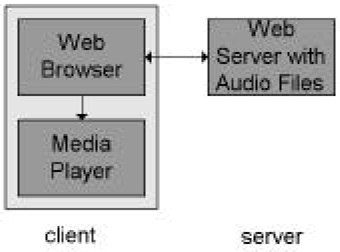
Multimedia on the Internet

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

The Media Player

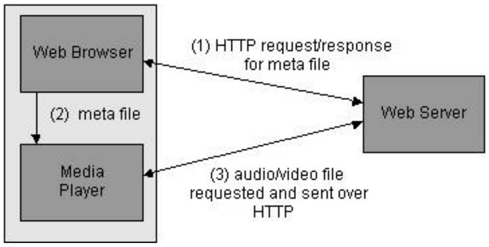
- End-host application
  - Real Player, Windows Media Player
- Needs to be pretty smart
- Decompression (MPEG)
- Jitter-removal (Buffering)
- Error correction (Repair, as a topic)
- GUI with controls (HCI issues)
  - Volume, pause/play, sliders for jumps

Streaming through a Web Browser

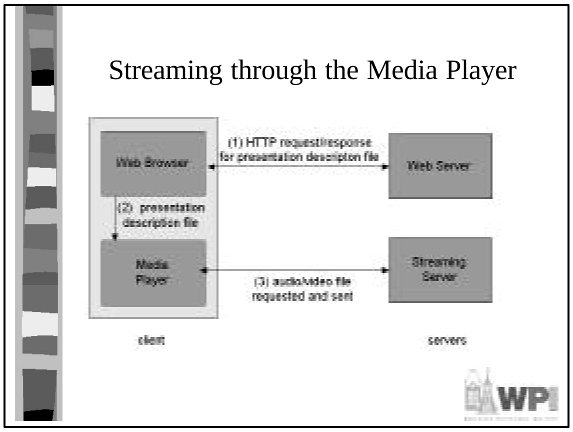


Must download whole file first!

Streaming through a Plug-In

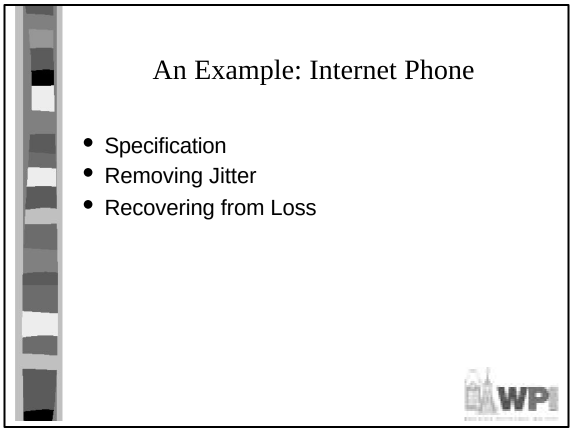

Web Browser
(1) HTTP request/response for meta file
(2) meta file
Web Server
Media Player
(3) audio/video file requested and sent over HTTP

Must still use TCP!



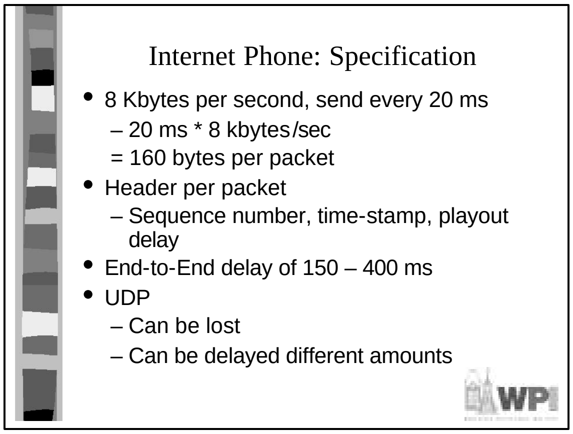
Streaming through the Media Player

Web Browser
(1) HTTP request/response for presentation description file
Web Server
(2) presentation description file
Media Player
(3) audio/video file requested and sent
Streaming Server
client
servers



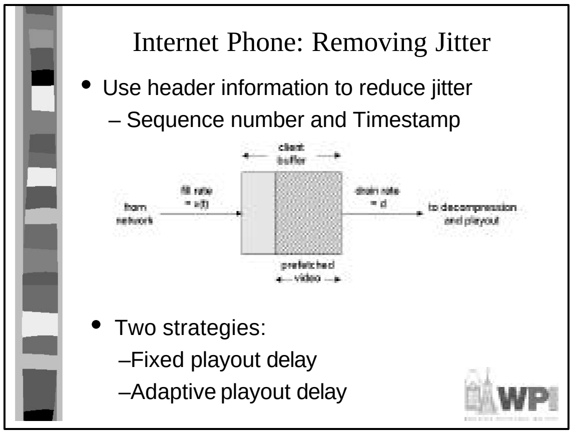
An Example: Internet Phone

- Specification
- Removing Jitter
- Recovering from Loss



Internet Phone: Specification

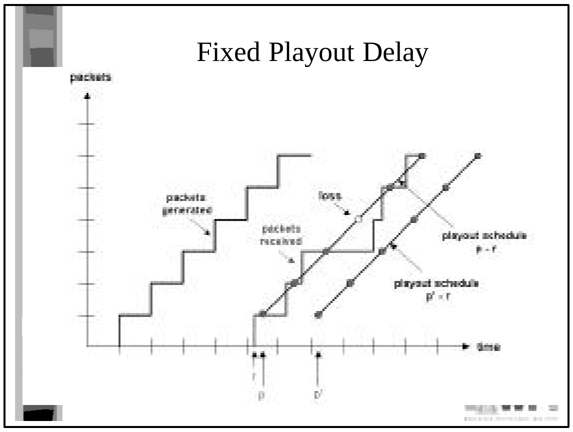
- 8 Kbytes per second, send every 20 ms
  - 20 ms * 8 kbytes/sec
  = 160 bytes per packet
- Header per packet
  - Sequence number, time-stamp, playout delay
- End-to-End delay of 150 – 400 ms
- UDP
  - Can be lost
  - Can be delayed different amounts



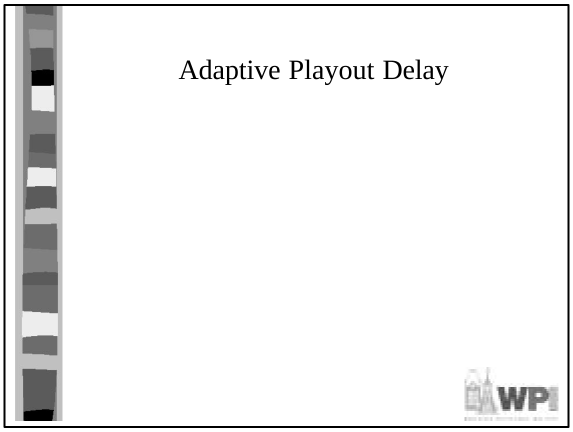
Internet Phone: Removing Jitter

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

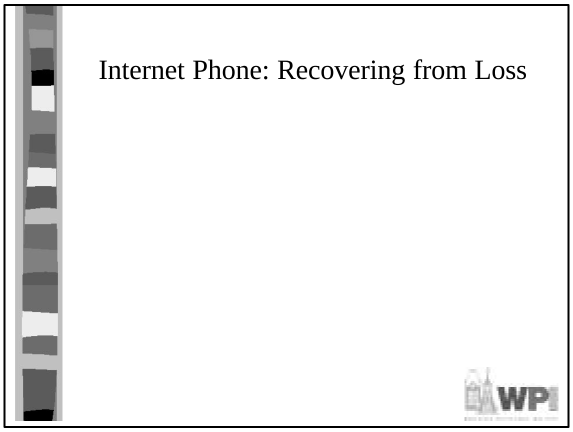
- Two strategies:
  –Fixed playout delay
  –Adaptive playout delay



Fixed Playout Delay



Adaptive Playout Delay



Internet Phone: Recovering from Loss