Multimedia Systems

CS-538 Advanced Operating Systems

(Slides include materials from *Operating System Concepts*, 7th ed., by Silbershatz, Galvin, & Gagne and from *Modern Operating Systems*, 2nd ed., by Tanenbaum)

Outline

- What is multi-media?
- Requirements and challenges for audio and video in computer systems
  - CD devices and formats
- Systems for multimedia
- Compression and bandwidth
- If time:–
  - Processor and disk scheduling
  - Network streaming and management

Silbershatz, Chapter 20
Tanenbaum, Chapter 7

What do we mean by “multimedia”

- Audio and video within a computer system
  - CD’s & DVD’s
  - Computer hard drive
  - Interactive
- Live broadcast & web casts
  - Radio stations, Network TV, Webcams, Skype, …
- Video on demand
  - Pause, fast forward, reverse, etc.
- Interactive activities involving audio, video, images
  - Meetings and presentations with 2-way audio
  - Teleconferencing
- Handheld devices
  - iPod, MP-3 players; personal video; mobile phones, …
- …

Multimedia Data and Delivery

- Stored in file system like ordinary data.
- Must be accessed with specific timing requirements.
- E.g., video *must* be displayed at 24-30 frames per second.
    - System must guarantee timely delivery

- *Continuous-media data*
    - Data with specific rate requirements

Multimedia Characteristics

- Multimedia files can be *very* large.
- Continuous media data may require *very* high data rates.
- Multimedia applications are usually sensitive to timing delays during playback.
- *Note:* human ear is more sensitive to *jitter* in audio than eye is to *jitter* in video!

Requirements

- "Smooth" audio and video
  - Deterioration in quality $\Rightarrow$ jerky playback
- Multiple concurrent streams
  - Video & multimedia servers
  - TiVo, etc.
- Wide range of network bandwidths
- Audio/video while PC is doing something else

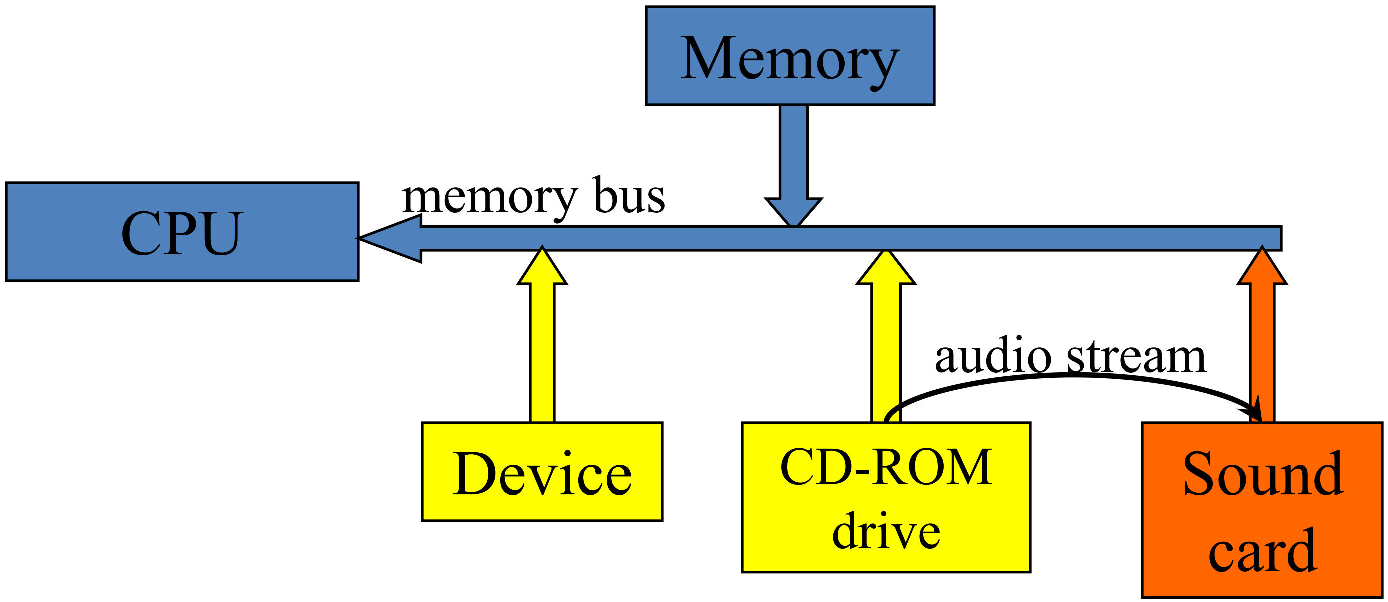
System and OS Challenges

- Bandwidths and Compression
- Jitter
- Processor Scheduling
- Disk Scheduling
- Network Streaming

Some System Architectures

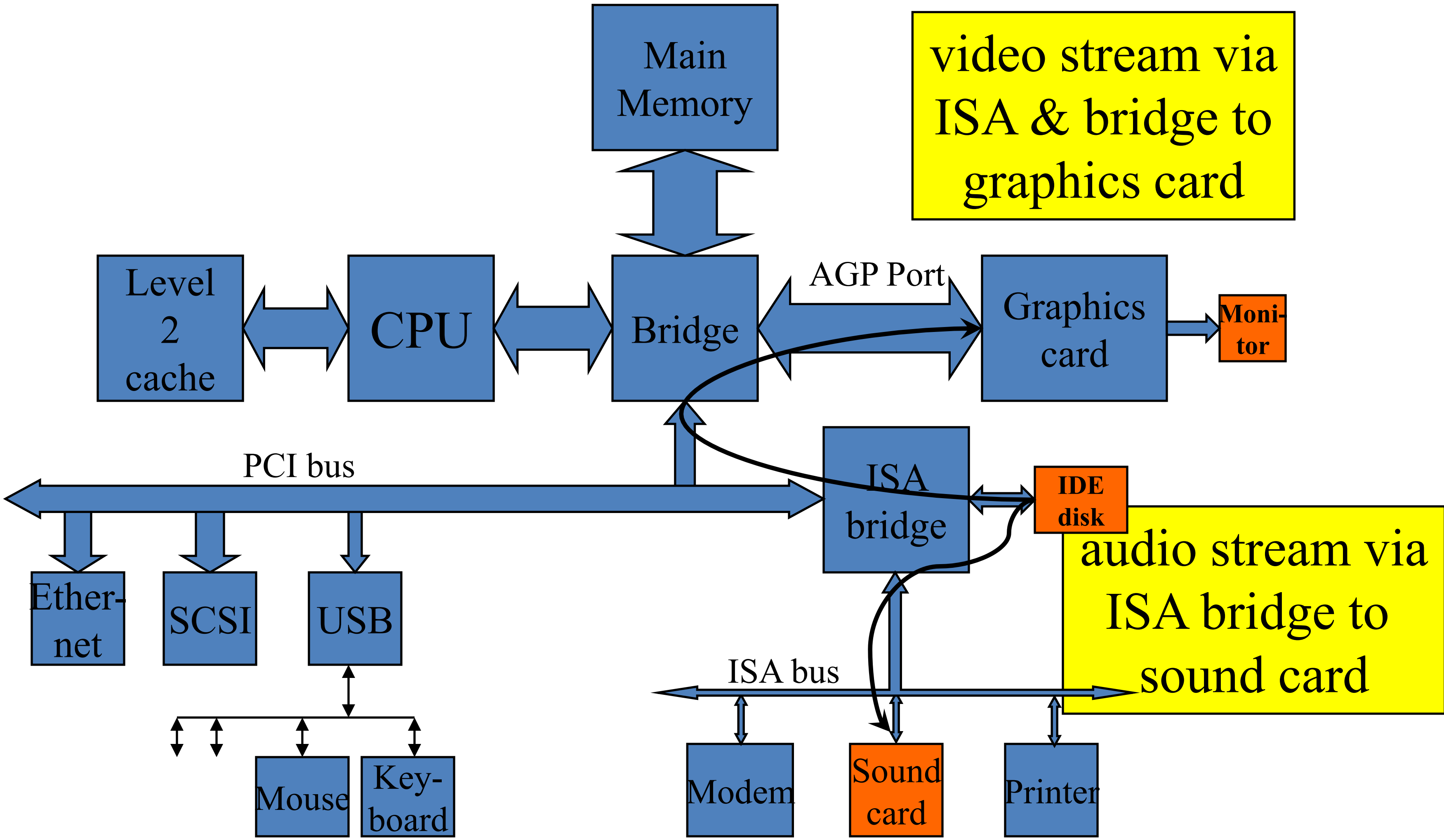
- Simple:
  - Data paths for audio/video that are separate from computational data paths
- Modern
  - Fast system bus, CPU, devices
- Video server
  - Disk farm and multiple streams

System Organization (a decade ago)



- Separate data path for audio stream
  - Or headphone jack and volume control on CD drive itself
- Main system bus and CPU were too busy/slow to handle real-time audio

System Organization (typical Pentium PC today)



DVD devices

- Similar in concept, different in details
- More data, higher bandwidth
  - *Track spacing:* 0.834µ *vs.* 1.6µ
  - *Bit spacing:* 0.4µ *vs.* 0.74µ
  - *Capacity:* 4.7 Gbytes *vs.* 650 Mbytes

Compression

An essential part of audio and video representation as files

Why Compression? – CD-quality audio

- 22,050 Hz $\Rightarrow$ 44,100 samples/sec
  - 16 bits per sample
- Two channels $\Rightarrow$ 176,400 bytes/sec
  $\cong$ 1.4 mbits/sec
  - Okay for a modern PC
  - Not okay for 56 kb/sec modem (speed) or iPod (space)!
- MP-3 $\cong$ 0.14 mbits/sec (10:1)
  - *Same* audio quality!
  - Compression ratio varies with type of music

Why Compression? – Video

- “Standard” TV frame = 640 $\times$ 480 pixels @ 25-30 frames/sec (fps) $\Rightarrow$
  - $\Rightarrow$ 9,216,000 pixels/sec = 27,648,000 bytes/sec
- Hollywood “standard” movie $\leq$ 133 minutes
  - $\Rightarrow$ approx. 210 gigabytes (standard resolution)!
- HDTV = 1280 $\times$ 720 pixels @ 30 fps
  - $\Rightarrow$ 82,944,000 bytes/sec (and rising!)
- DVD holds ~ 4.7 gigabytes
  - $\Rightarrow$ average of ~ 620 kilobytes/sec!
- “Standard” movie of 133 minutes requires serious compression just to *fit* onto DVD

Video Compression Requirements

- Compression ratio > 50:1
  - i.e., 210 gigabytes:4.7 gigabytes
- Visually indistinguishable from original
  - Even when paused
- Fast, cheap decoder
  - Slow encoder is okay
- VCR controls
  - Pause, fast forward, reverse

Video Compression Standards

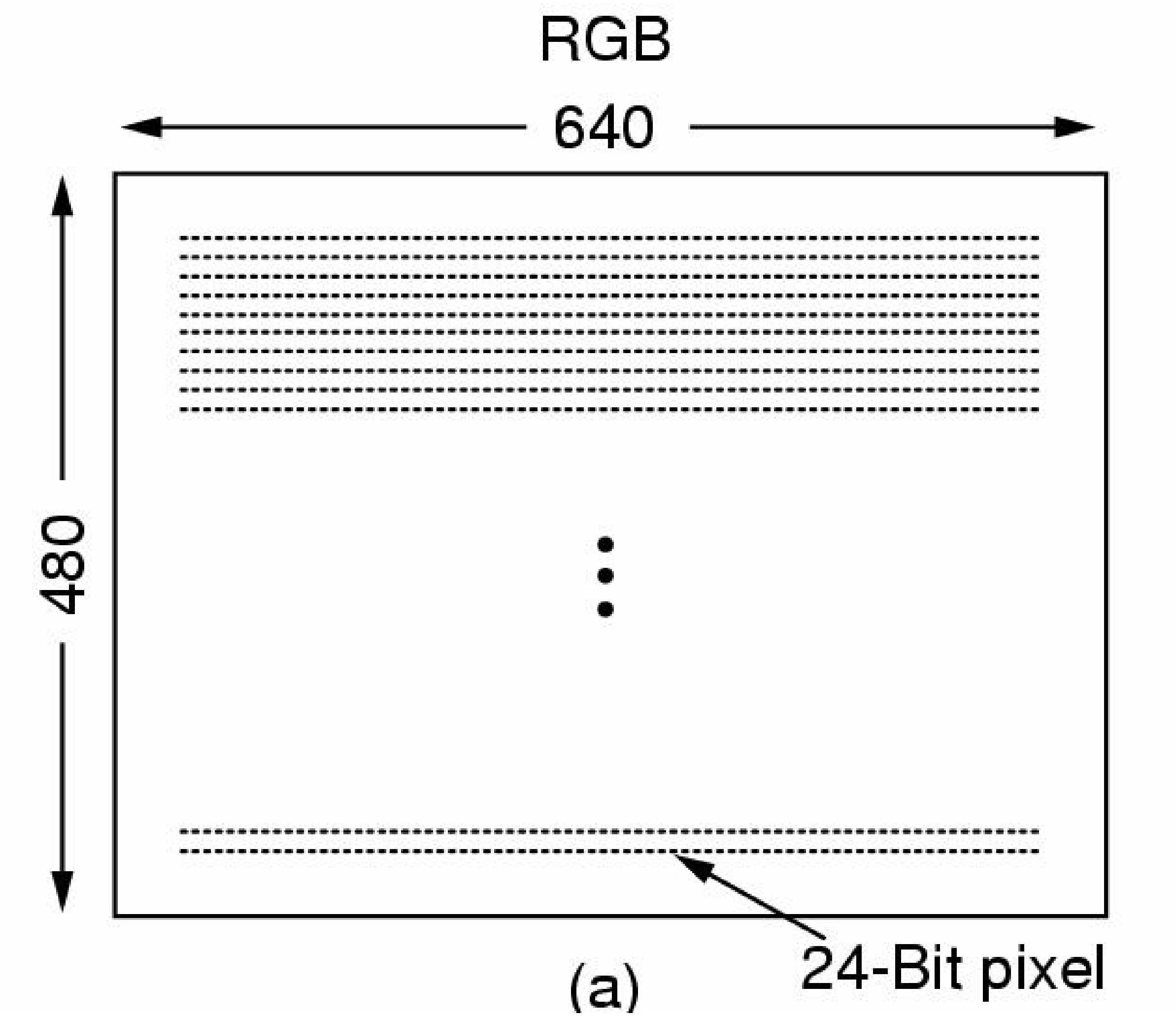
- MPEG (*Motion Picture Experts Group*)
  - Based on JPEG (*Joint Photographic Experts Group*)
  - Multi-layer
    - Layer 1 = system and timing information
    - Layer 2 = video stream
    - Layer 3 = audio and text streams
- Three standards
  - MPEG-1 – 352×240 frames; < 1.5 mb/sec (< VHS quality)
    - Layer 3 = MP3 Audio standard
    - Typical uses:– video clips on internet; video for handhelds
  - MPEG-2 – standard TV & HDTV; 1.5-15 mb/sec
    - DVD encoding
  - MPEG-4 – combined audio, video, interactive graphics
    - 2D & 3D animations

JPEG compression (single frame)

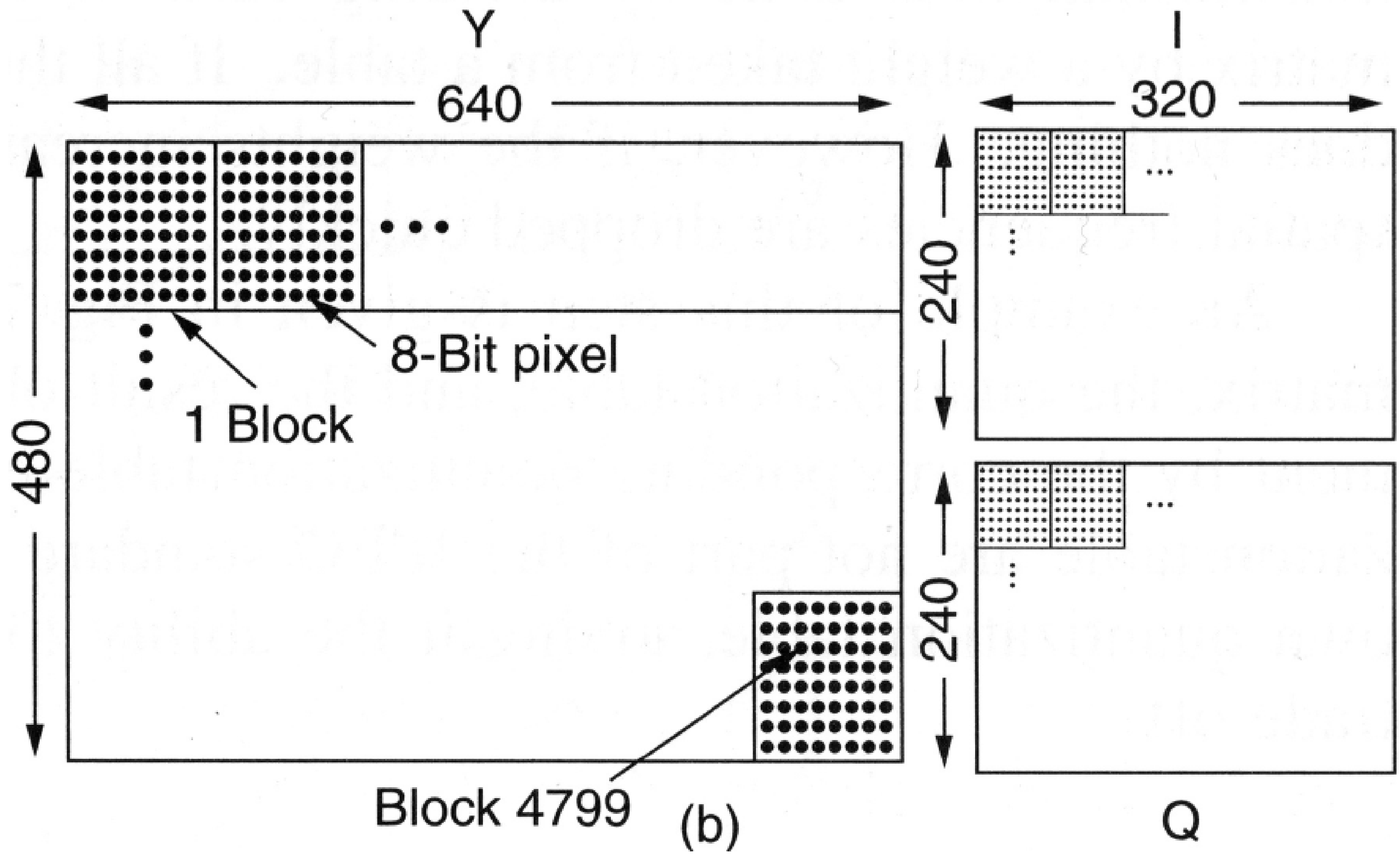
1. Convert RGB into YIQ
   - *Y* = *luminance* (i.e., brightness) ~ black-white TV
   - *I*, *Q* = *chrominance* (similar to *saturation* and *hue*)

   Reason: Human eye is more sensitive to luminance than to color (rods *vs.* cones)
2. Down-sample *I, Q* channels
   - i.e., average over 2×2 pixels to reduce resolution
   - lossy compression, but barely noticeable to eye
3. Partition each channel into 8×8 blocks
   - 4800 *Y* blocks, 1200 each *I* & *Q* blocks (for 640 × 480)

JPEG (continued)



JPEG (continued)



JPEG (continued)

4. Calculate *Discrete Cosine Transform* (DCT) of each 8×8 block
5. Divide 8×8 block of DCT values by *8×8 quantization table*
   - Effectively throwing away higher frequencies
6. Linearize 8×8 block, run-length encode, and apply a Huffman code to reduce to a small fraction of original size (in bytes)

JPEG (concluded)

7. Store or transmit 8×8 quantization table followed by list of compressed blocks

- Achieves 20:1 compression with good visual characteristics
  - Higher compression ratios possible with visible degradation
- JPEG algorithm executed backwards to recover image
  - Visually indistinguishable from original @ 20:1
- JPEG algorithm is symmetric
  - Same speed forwards and backwards

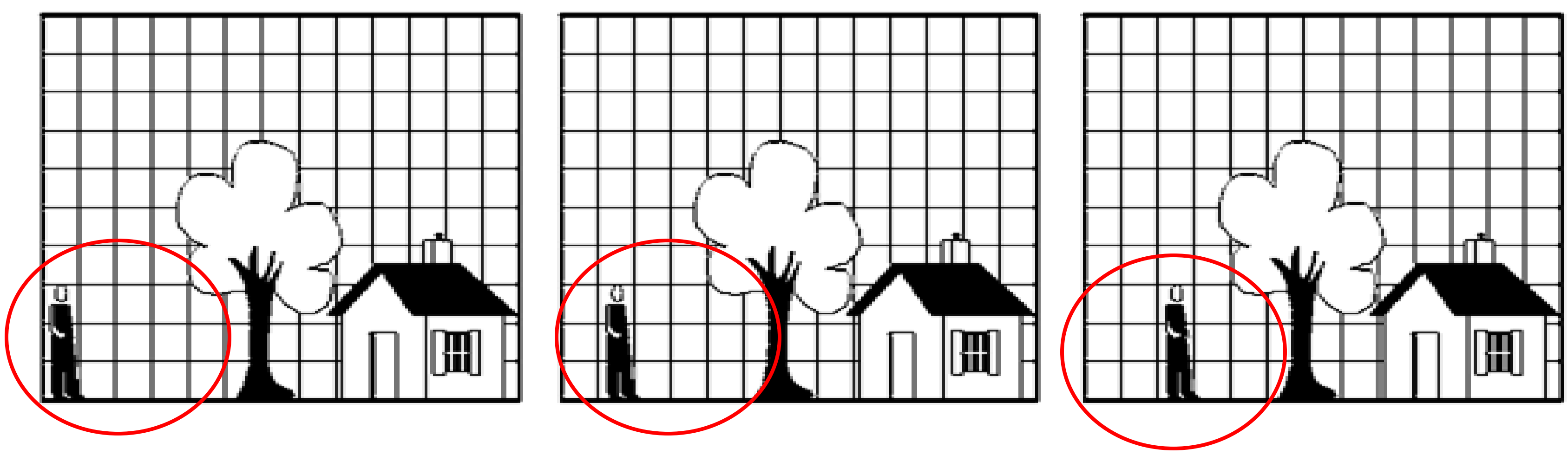
MPEG

- JPEG-like encoding of each frame
- Takes advantage of *temporal locality*
- I.e., each frame usually shares similarities with previous frame

  $\Rightarrow$ encode and transmit only differences
- Sometimes an object moves relative to background

  $\Rightarrow$ find object in previous frame, calculate difference, apply *motion vector*

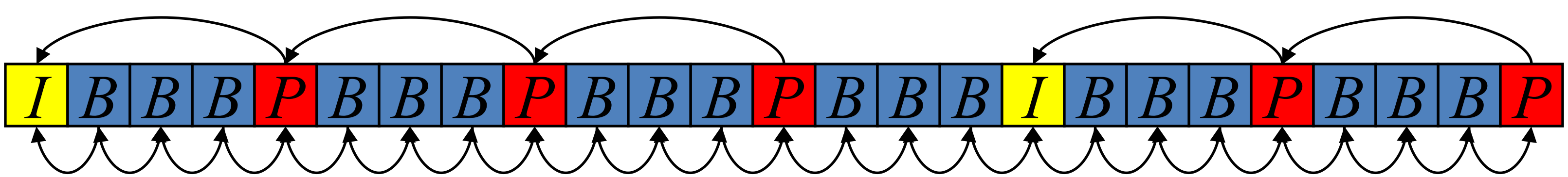
Temporal Locality (example)



Consecutive Video Frames

MPEG organization

- Three types of frames
  - *I-frame: Intracoded* or *Independent*.
    - Full JPEG-encoded frame
    - Occurs at intervals of a second or so
    - Also at start of every *scene*
  - *P-frame: Predictive* frame
    - Difference from previous frame
  - *B-frame: Bidirectional* frame
    - Like *p-frame* but difference from both *previous* and *next* frame



MPEG Characteristics

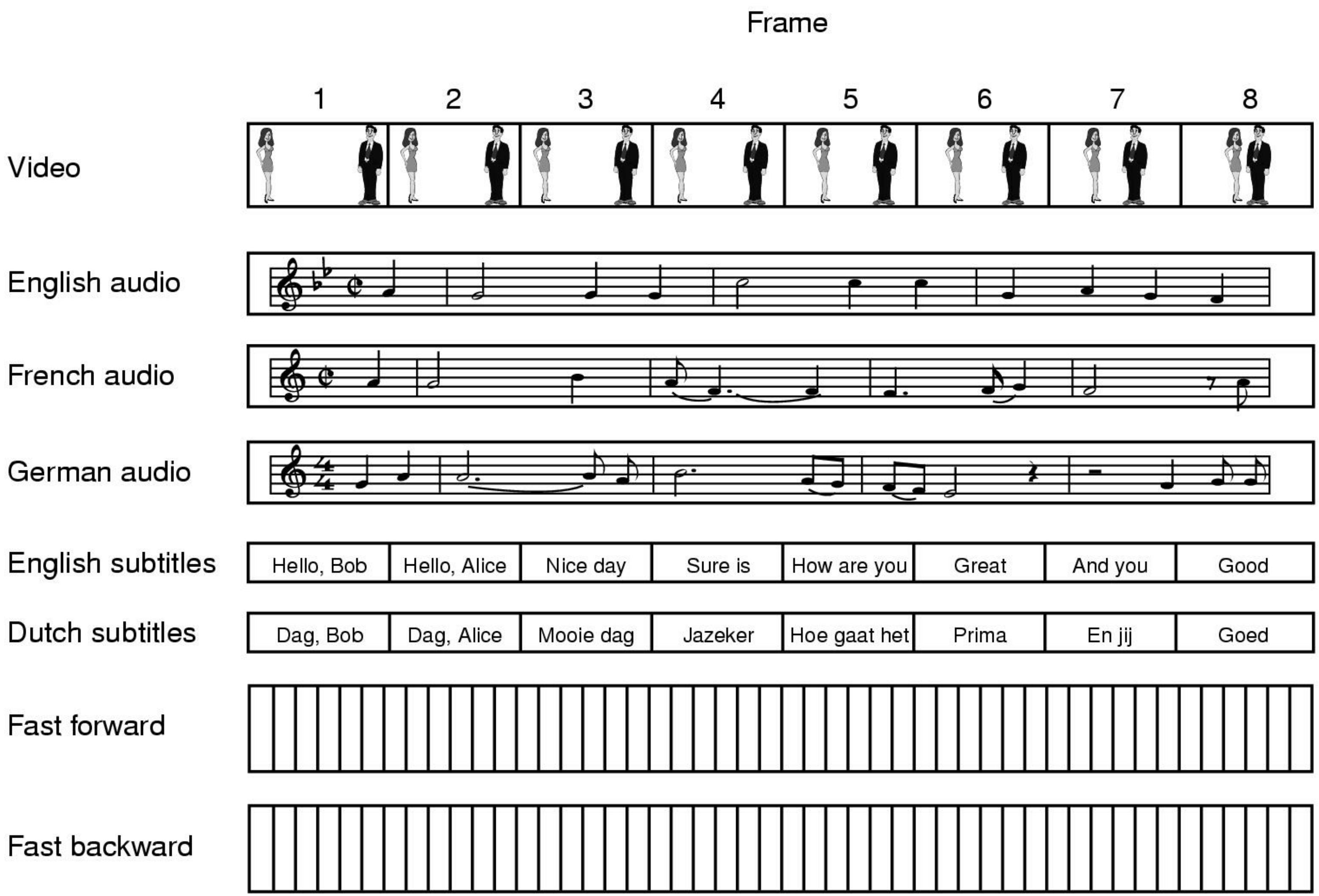
- Non-uniform data rate!
- Compression ratios of 50:1 – 80:1 are readily obtainable
- Asymmetric algorithm
  - Fast decode (like JPEG)
  - Encode requires image search and analysis to get high quality differences
- Cheap decoding chips available for
  - graphics cards
  - DVD players, etc.

MPEG Problem – Fast Forward/Reverse

- Cannot simply skip frames
  - Next desired frame might be *B* or *P* derived from a skipped frame
- Options:
  - Separate fast forward and fast reverse files
    - MPEG encoding of every *nth* frame
    - Often used in video-on-demand server
  - Display only *I* and *P* frames
    - If *B* frame is needed, derive from nearest *I* or *P*

“Movie” File Organization

- One MPEG-2 video stream
  - Possible fast forward, fast reverse sub-streams
- Multiple audio streams
  - Multiple languages
- Multiple text streams
  - Subtitles in multiple languages
- All interleaved
  - Possibly in multiple files!


Frame
1
2
3
4
5
6
7
8
Video
English audio
French audio
German audio
English subtitles
Hello, Bob
Hello, Alice
Nice day
Sure is
How are you
Great
And you
Good
Dutch subtitles
Dag, Bob
Dag, Alice
Mooie dag
Jazeker
Hoe gaat het
Prima
En jij
Goed
Fast forward
Fast backward

Operating System Challenge

- How to get the contents of a movie file from disk or DVD drive to video screen and speakers.
  - Fixed frame rate (25 or 30 fps)
  - Steady audio rate
  - Bounded *jitter*
- Such requirements are known as *Quality-of-Service* (*QoS*) guarantees.

- Classical problem in *real-time scheduling*
  - Obscure niche become mainstream!
  - See Silbershatz, §19.1–19.5

QoS Guarantees

- Building a system to guarantee QoS effects the following:–
  - CPU processing
  - Scheduling and interrupt handling
  - File systems
  - Network protocols

Requirement of Multimedia Operating Systems

- There are three levels of QoS
  - *Best-effort service:* the system makes a best effort with no attempt to guarantee
  - *Soft QoS:* allows different traffic streams to be prioritized, but no QoS guarantees are made
  - *Hard QoS:* system ensures QoS requirements are always met
    - Prioritization
    - Admission control
    - Bounded latency on interrupt handling, processing, etc.

Parameters Defining QoS

- *Throughput:* the total amount of work completed during a specific time interval
- *Delay:* the elapsed time from when request is first submitted to desired result
  Jitter: delays that occur during playback of a stream.
  Reliability: how errors are handled during transmission and processing of continuous media

Further QoS Issues

- QoS may be *negotiated* between the client and server.
- Operating systems may use an *admission control* algorithm
  - Admits a request for service only if the server has sufficient resources to satisfy the request

Two common methods

- Rate Monotonic Scheduling
- Earliest Deadline First

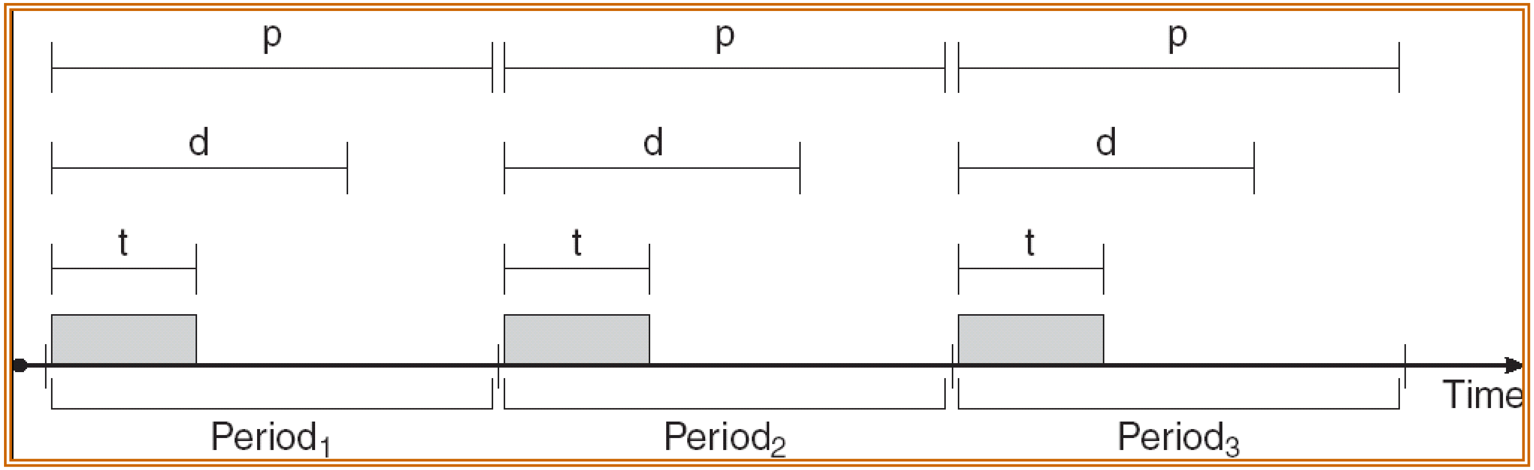
- Many variations
- Many analytic methods for proving QoS (Quality of Service)

Processor Scheduling for Real-Time
Rate Monotonic Scheduling (RMS)

- Assume *m* periodic processes
  - Process *i* requires t_i msec of processing time <u>every</u> p_i msec.
  - Equal processing every interval — like clockwork!

Example

- Periodic process *i* requires the CPU at specified intervals (periods)
- $\boldsymbol{p_i}$ is the duration of the period
- $\boldsymbol{t_i}$ is the processing time
- $\boldsymbol{d_i}$ is the deadline by when the process must be serviced
  - Often same as end of period



Processor Scheduling for Real-Time *Rate Monotonic Scheduling* (RMS)

- Assume *m* periodic processes
  - Process *i* requires t_i msec of processing time <u>every</u> p_i msec.
  - Equal processing every interval — like clockwork!
- Assume

$$\sum_{i=1}^{m} \frac{t_i}{p_i} \leq 1$$

- Assign priority of process *i* to be $\frac{1}{p_i}$
  - Statically assigned
- Let priority of non-real-time processes be *0*

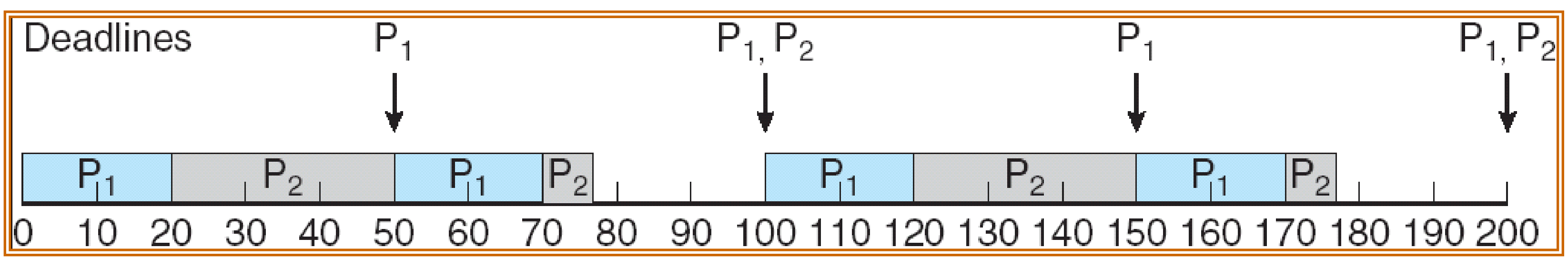
Rate Monotonic Scheduling (continued)

- Scheduler simply runs highest priority process that is ready
  - May pre-empt other real-time processes
  - Real-time processes become ready in time for each frame or sound interval
  - Non-real-time processes run only when no real-time process needs CPU

Example

- p_1 = *50* msec; t_1 = 20 msec
- p_2 = *100* msec; t_2 = 35 msec
- Priority(p_1) > Priority(p_2)

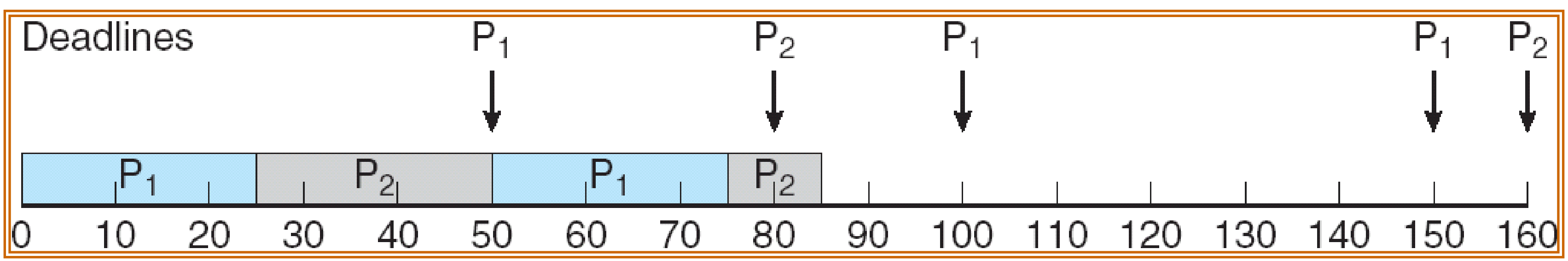
- Total compute load is 75 msec per every 100 msec.
- Both tasks complete within every period
  - 25 msec per 100 msec to spare



Example 2

- $p_1 = 50$ msec; $t_1 = 25$ msec
- $p_2 = 80$ msec; $t_2 = 35$ msec
- Priority(p_1) > Priority(p_2)

- Total compute load is ~ 94% of CPU.
- Cannot complete both tasks within some periods
  - Even though there is still CPU capacity to spare!



Processor Scheduling for Real-Time
Earliest Deadline First (EDF)

- When each process *i* become ready, it announces deadline D_i for its next *task*.
- Scheduler always assigns processor to process with earliest deadline.
  - May pre-empt other real-time processes

Earliest Deadline First Scheduling (continued)

- No assumption of periodicity
- No assumption of uniform processing times

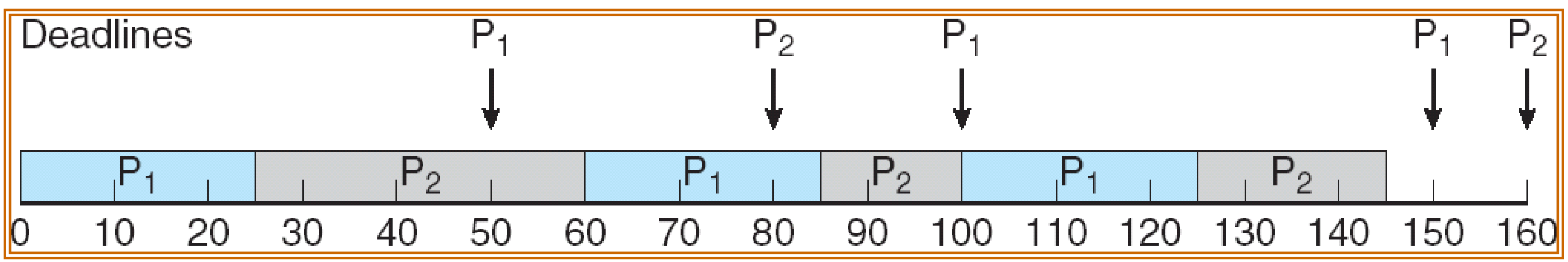
- **Theorem:** If *any* scheduling policy can satisfy QoS requirement for a sequence of real time tasks, then EDF can also satisfy it.
  - *Proof:* If *i* scheduled before *i+1*, but $D_{i+1} < D_i$, then *i* and *i+1* can be interchanged without affecting QoS guarantee to either one.

Earliest Deadline First Scheduling (continued)

- EDF is more complex scheduling algorithm
  - Priorities are dynamically calculated
  - Processes must know deadlines for tasks
- EDF can make higher use of processor than RMS
  - Up to 100%
- There is a large body of knowledge and theorems about EDF analysis

Example 2 (again)

- Priorities are assigned according to deadlines:
  - the earlier the deadline, the higher the priority;
  - the later the deadline, the lower the priority.



Network Management (continued)

- Broadcasting
  - Like cable TV
  - Fixed portion of network bandwidth dedicated to set of broadcast streams; inflexible
- Multicasting
  - Typical webcast
  - Multicast tree set up through internet to reach customers
  - Customers can come and go
- Unicast
  - Dedicated connection between server and client
  - Streaming version of familiar transport protocols

Streaming –
Delivery of Multimedia Data over Network

- Two types of streaming:–
  - *Progressive download:* client begins playback of multimedia file during delivery.
    - File is ultimately stored on client computer
    - (Hopefully) download speed > playback speed
  - *Real-time streaming:* multimedia file is delivered to, but *not* stored on, client computer
    - Played back at same speed as delivery
    - Limited amount of buffering to remove jitter

Two Types of Real-time Streaming

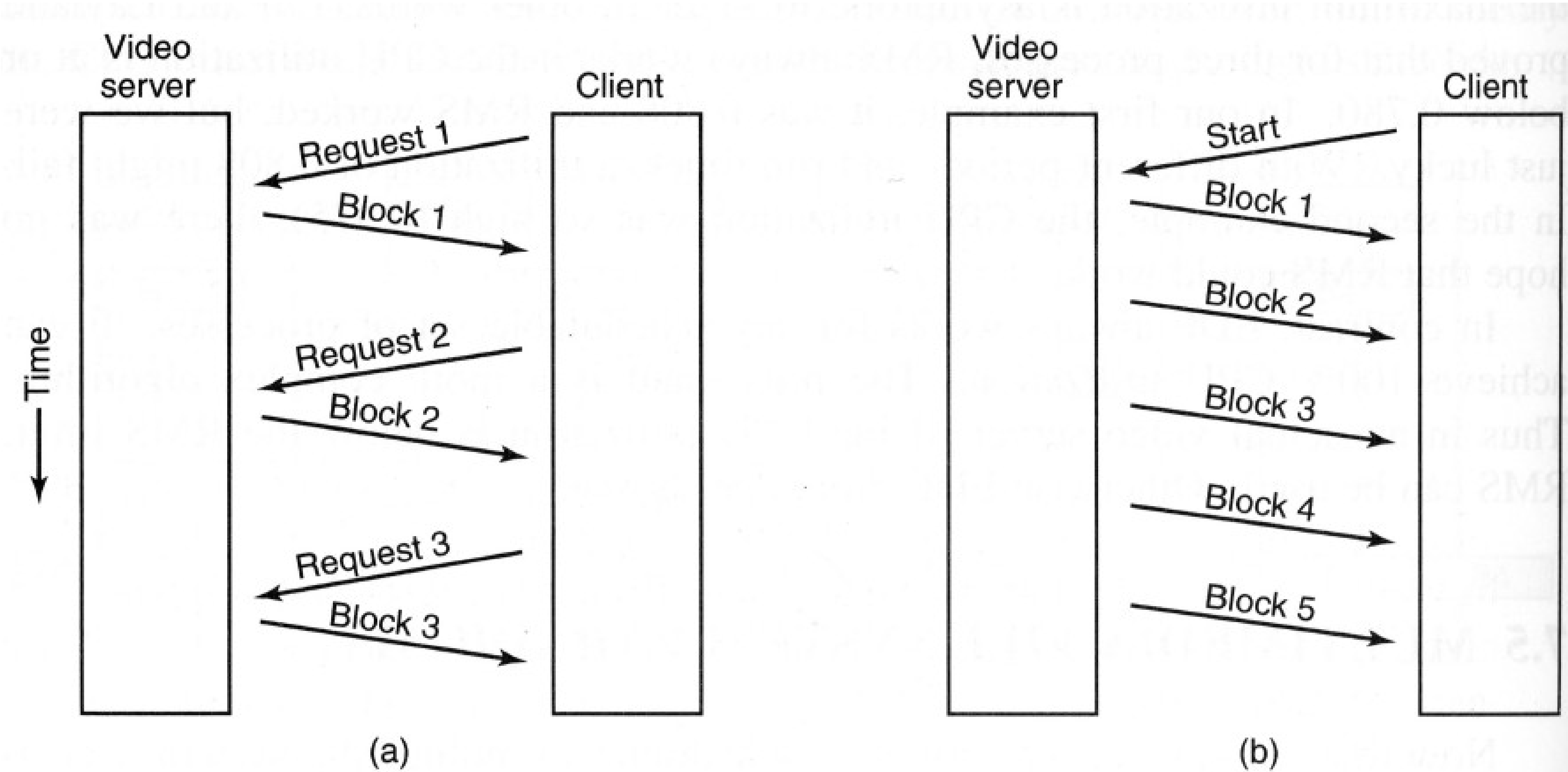
- *Live streaming:* to deliver a live event while it is occurring.
    - Broadcast or multicast

- *On-demand streaming:* to deliver archived media streams
    - Movies, lectures, old TV shows, etc.
    - Events *not* delivered when they occur.
    - Playback with *pause, fast forward, reverse, etc*
    - Unicast (usually)

Network Streaming

- Traditional HTTP
  - Stateless
  - Server responds to each request independently
- Real-Time Streaming Protocol (RTSP)
  - Client initiates a “push” request for stream
  - Server provides media stream at frame rate

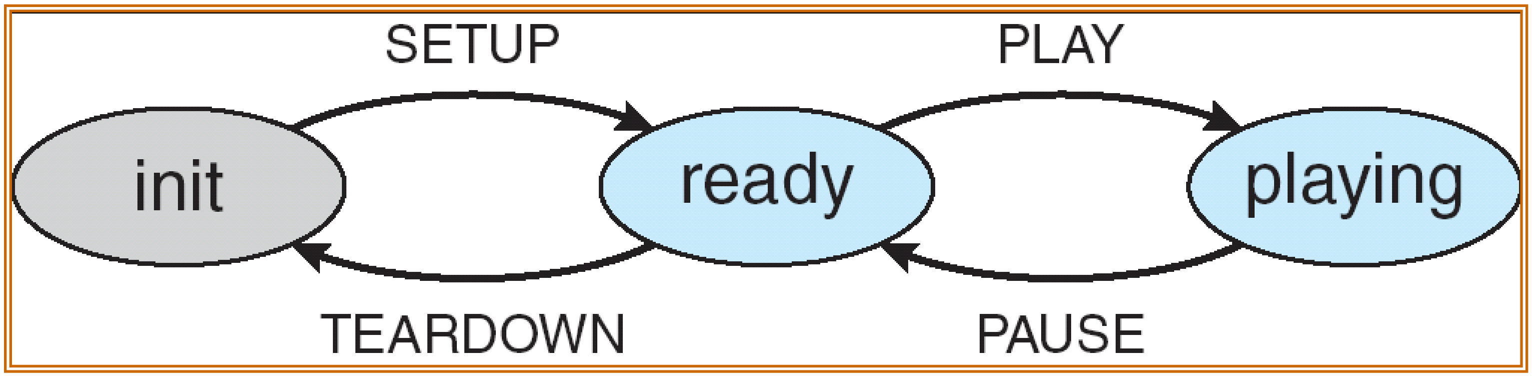
Pull (HTTP) *vs.* Push (RTSP) server



RTSP States

- *SETUP:* server allocates resources for client session.
- *PLAY:* server delivers stream to a client session.
- *PAUSE:* server suspends delivery of a stream.
- *TEARDOWN:* server releases resources and breaks down connection.

RTSP state machine



Bandwidth Negotiation

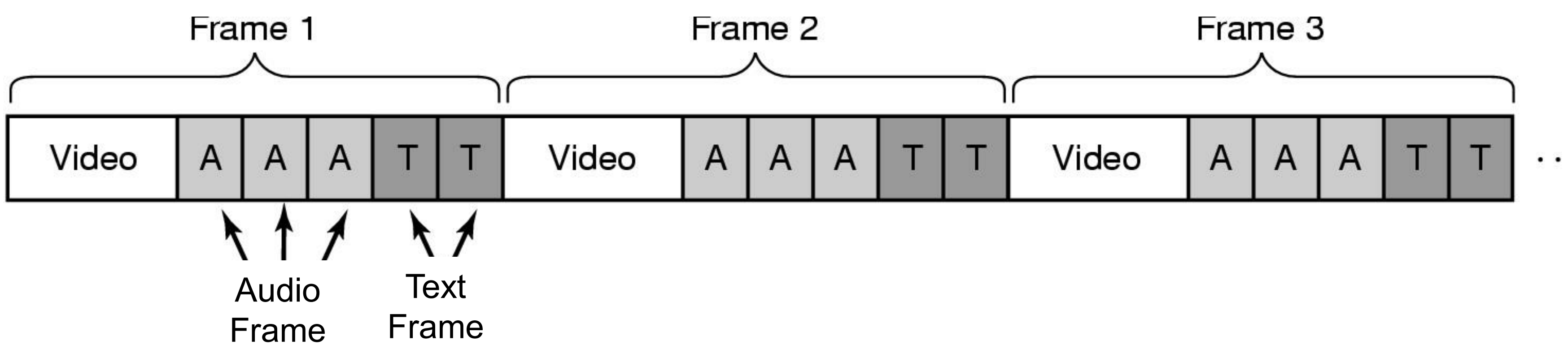
- Client application provides feedback to server to adjust bandwidth
- E.g.,
  - Windows Media Player
  - RealPlayer
  - Quicktime

Video Server

- Multiple CPUs
- Disk farm
  - 1000s of disks
- Multiple high-bandwidth network links
  - Cable TV
  - Video on demand
  - Internet

Multimedia File & Disk Management

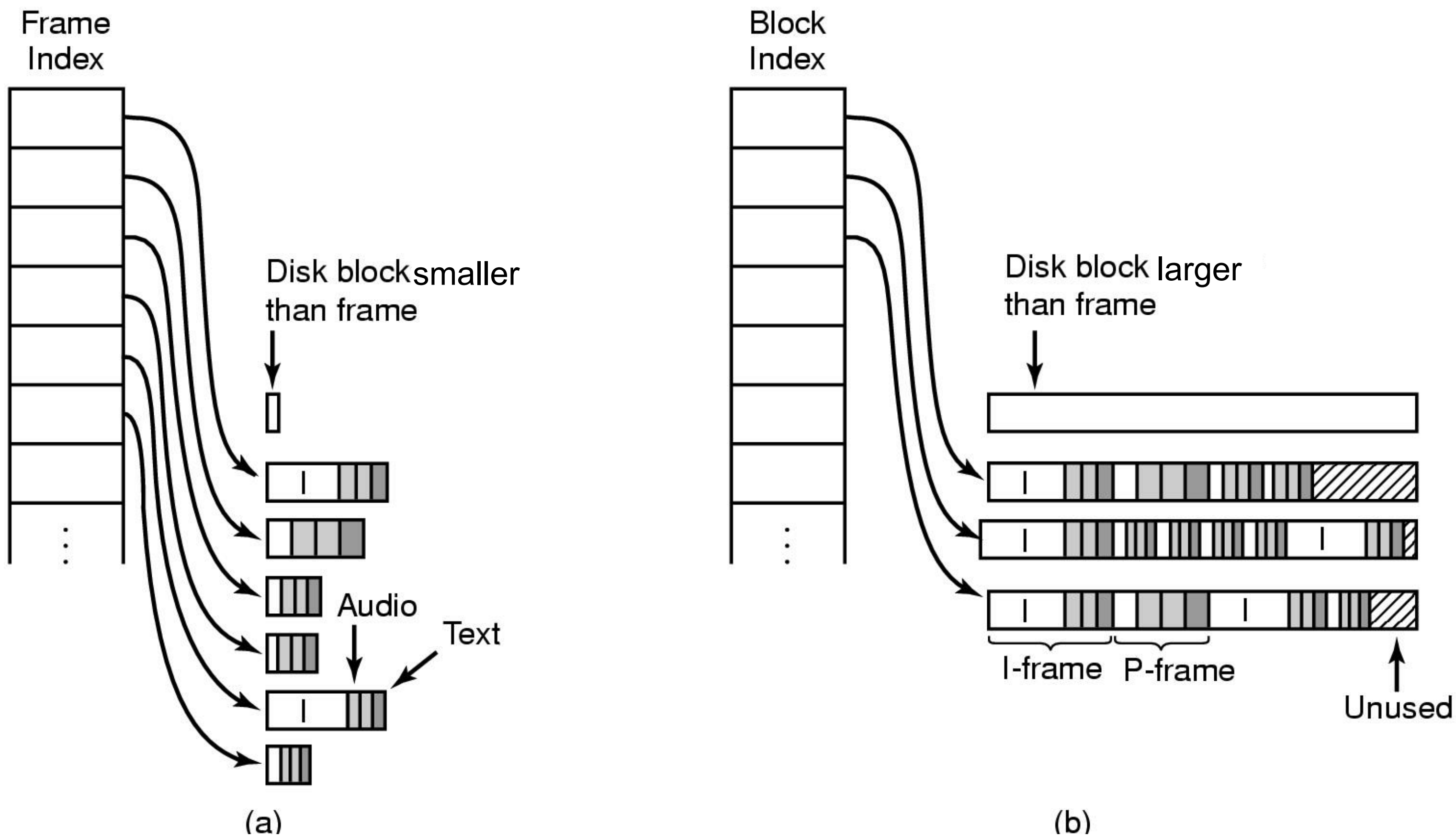
- Single movie or multimedia file on disk
  - Interleave audio, video, etc.
    - So temporally equivalent blocks are near each other
  - Attempt contiguous allocation
    - Avoid seeks within a frame



File organization – Frame *vs.* Block

- Frame organization
  - Small disk blocks (4-16 Kbytes)
  - Frame index entries point to starting block for each frame
  - Frames vary in size (MPEG)
  - Advantage: very little storage fragmentation
  - Disadvantage: large frame table in RAM
- Block organization
  - Large disk block (256 Kbytes or more)
  - Block index entries point to first *I-frame* of a sequence
  - Multiple frames per block
  - Advantage: much smaller block table in RAM
  - Disadvantage: large storage fragmentation on disk

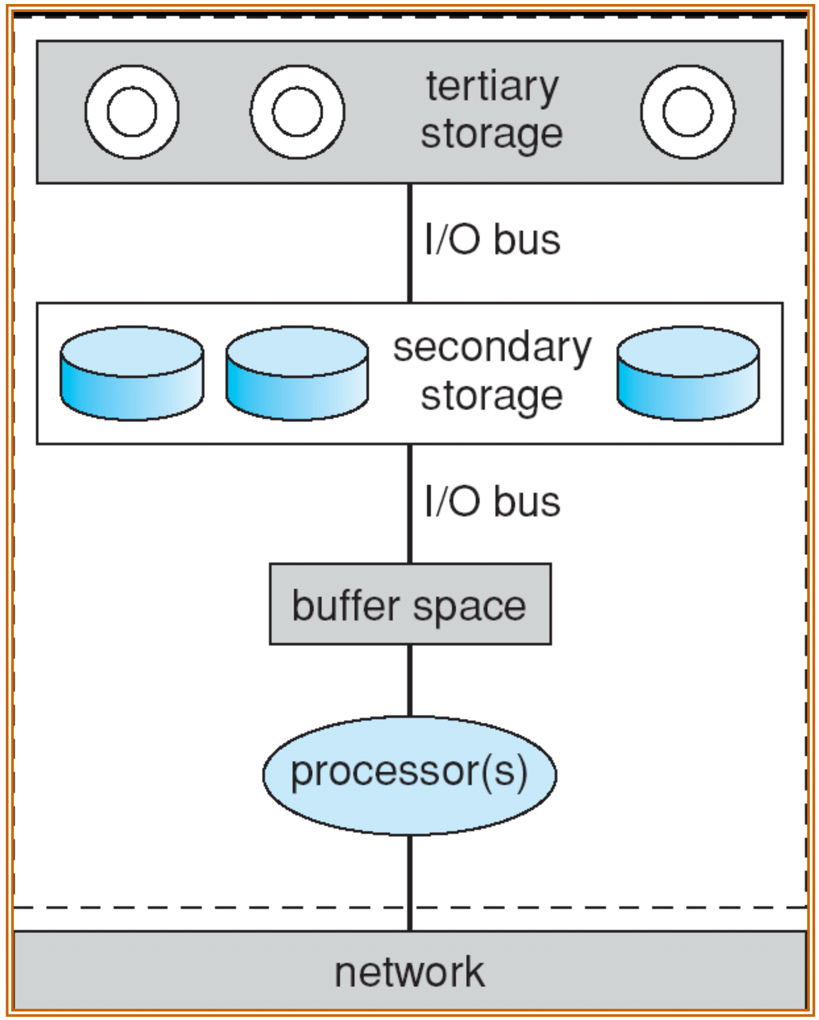
Frame *vs.* Block organization



File Placement on Server

- Random
- Striped
- “Organ pipe” allocation
  - Most popular video in center of disk
  - Next most popular on either side of it, etc.
  - Least popular at edges of disk
  - Minimizes seek distance

Resources on a file server



Conclusion

- Multimedia computing is challenging
- Possible with modern computers
    - Compression is essential, especially for video
- Real-time computing techniques move into mainstream
    - Processor and disk scheduling

- There is much more to this subject than fits into one class

Thank You……………………………………………

For your attentions