# Multimedia Networking

**Reading: Sections 3.1.2, 3.3, 4.5, and 6.5**

CS-375: Computer Networks

Dr. Thomas C. Bressoud



# Digital Audio and Video Data

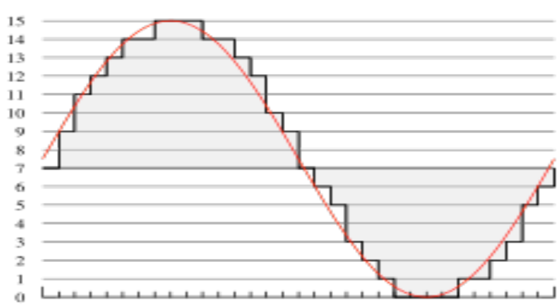
# Challenges for Media Streaming

- Large volume of data
  - Each sample is a sound or an image
  - Many samples per second
- Volume of data may vary over time
  - Due to compression of the data
- Cannot tolerate much variation in delay
  - Once playout starts, need to keep playing
- Sometimes cannot tolerate much delay, period
  - For interactive applications (e.g., VoIP and gaming)
- Though some loss is acceptable



# Digital Audio

- Sampling the analog signal
  - Sample at some fixed rate
  - Each sample is an arbitrary real number
- Quantizing each sample
  - Round each sample to one of a finite number of values
  - Represent each sample in a fixed number of bits




**4 bit representation (values 0-15)**

## Audio Examples

- Speech
  - Sampling rate: 8000 samples/second
  - Sample size: 8 bits per sample
  - Rate: 64 kbps

- Compact Disc (CD)
  - Sampling rate: 44,100 samples/second
  - Sample size: 16 bits per sample
  - Rate: 705.6 kbps for mono,
    1.411 Mbps for stereo

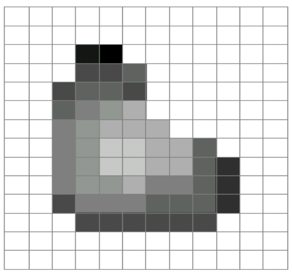


## Audio Compression

- Audio data requires too much bandwidth
  - Speech: 64 kbps is too high for a dial-up modem user
  - Stereo music: 1.411 Mbps exceeds most access rates
- Compression to reduce the size
  - Remove redundancy
  - Remove details that human tend not to perceive
- Example audio formats
  - Speech: GSM (13 kbps), G.729 (8 kbps), and G.723.3 (6.4 and 5.3 kbps)
  - Stereo music: MPEG 1 layer 3 (MP3) at 96 kbps, 128 kbps, and 160 kbps

# Digital Video

- Sampling the analog signal
  - Sample at some fixed rate (e.g., 24 or 30 times per sec)
  - Each sample is an image
- Quantizing each sample
  - Representing an image as an array of picture elements
  - Each pixel is a mixture of colors (red, green, and blue)
  - E.g., 24 bits, with 8 bits per color





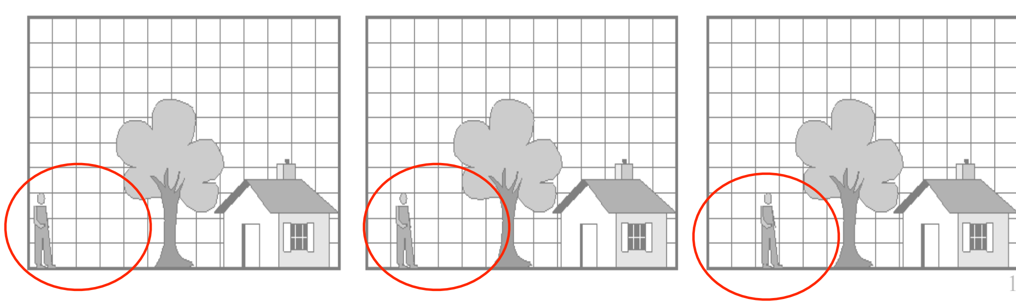
## Video Compression: Within an Image

- Image compression
  - Exploit spatial redundancy (e.g., regions of same color)
  - Exploit aspects humans tend not to notice
- Common image compression formats
  - Joint Pictures Expert Group (JPEG)
  - Graphical Interchange Format (GIF)


Uncompressed: 167 KB    Good quality: 46 KB    Poor quality: 9 KB

## Video Compression: Across Images

- Compression across images
  - Exploit temporal redundancy across images
- Common video compression formats
  - MPEG 1: CD-ROM quality video (1.5 Mbps)
  - MPEG 2: high-quality DVD video (3-6 Mbps)
  - Proprietary protocols like QuickTime and RealNetworks



## Streaming Over the Internet



## Transferring Audio and Video Data

- Simplest case: just like any other file
  - Audio and video data stored in a file
  - File downloaded using conventional protocol
  - Playback does not overlap with data transfer
- A variety of more interesting scenarios
  - Live vs. pre-recorded content
  - Interactive vs. non-interactive
  - Single receiver vs. multiple receivers
- Examples
  - Streaming audio and video data from a server
  - Interactive audio in a phone call

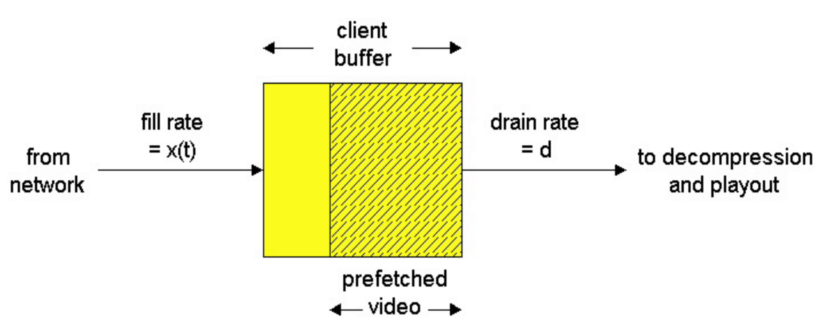
# Streaming Stored Audio and Video

- Client-server system
  - Server stores the audio and video files
  - Clients request files, play them as they download, and perform VCR-like functions (e.g., rewind and pause)
- Playing data at the right time
  - Server divides the data into segments
  - … and labels each segment with timestamp or frame id
  - … so the client knows when to play the data
- Avoiding starvation at the client
  - The data must arrive quickly enough
  - … otherwise the client cannot keep playing

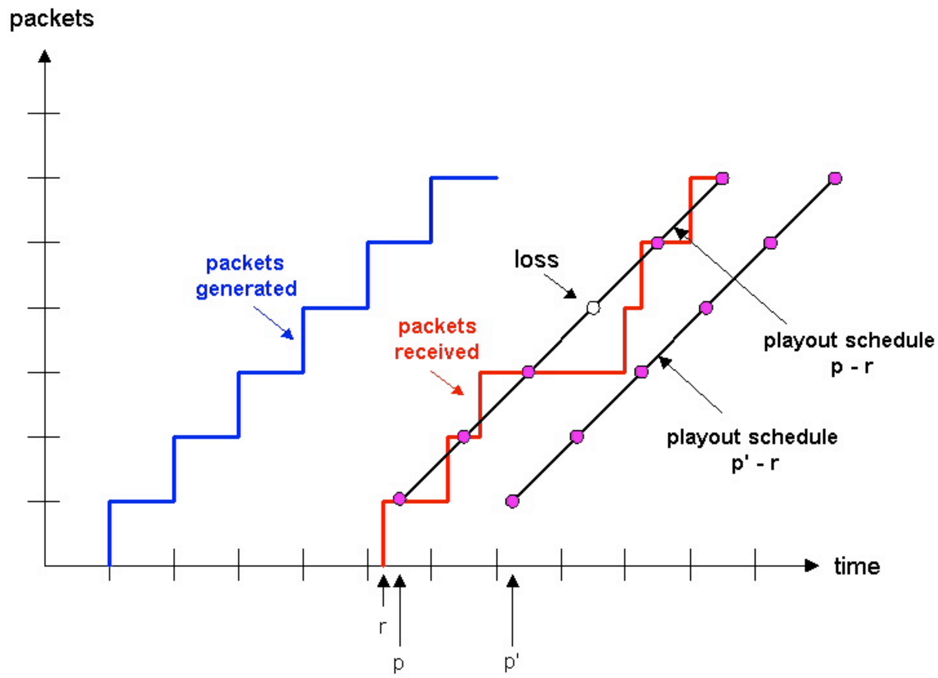


# Playout Buffer

- Client buffer
  - Store the data as it arrives from the server
  - Play data for the user in a continuous fashion
- Playout delay
  - Client typically waits a few seconds to start playing
  - … to allow some data to build up in the buffer
  - … to help tolerate some delays down the road



# Influence of Playout Delay

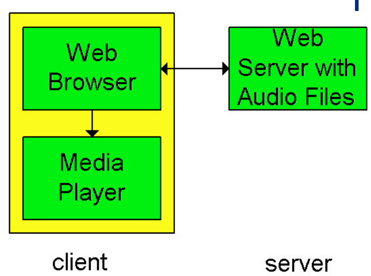



# Requirements for Data Transport

- Delay
  - Some small delay at the beginning is acceptable
  - E.g., start-up delays of a few seconds are okay
- Jitter
  - Variability of packet delay within the same packet stream
  - Client cannot tolerate high variation if the buffer starves
- Loss
  - Small amount of missing data does not disrupt playback
  - Retransmitting a lost packet might take too long anyway

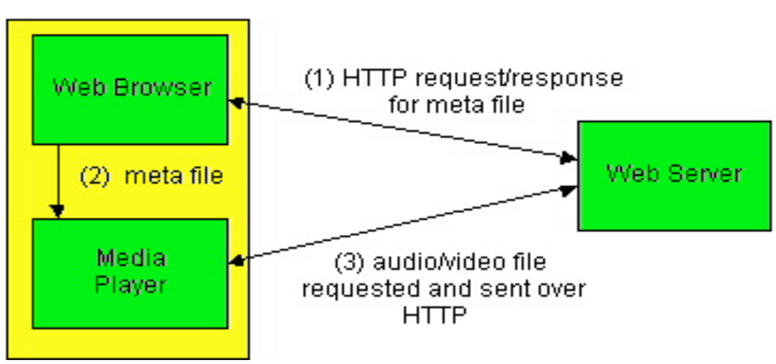
## Streaming From Web Servers

- Data stored in a file
  - Audio: an audio file
  - Video: interleaving of audio and images in a single file
- HTTP request-response
  - TCP connection between client and server
  - Client HTTP request and server HTTP response
- Client invokes the media player
  - Content-type indicates the encoding
  - Browser launches the media player
  - Media player then renders the file



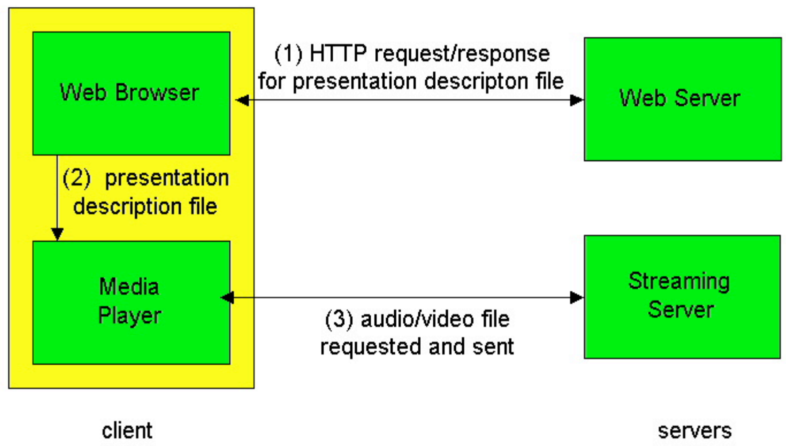

## Initiating Streams from Web Servers

- Avoid passing all data through the Web browser
  - Web server returns a meta file describing the object
  - Browser launches media player and passes the meta file
  - The player sets up its own connection to the Web server



## Using a Streaming Server

- Avoiding the use of HTTP (and perhaps TCP, too)
  - Web server returns a meta file describing the object
  - Player requests the data using a different protocol






## TCP is Not a Good Fit

- Reliable delivery
  - Retransmission of lost packets
  - … even though retransmission may not be useful
- Adapting the sending rate
  - Slowing down after a packet loss
  - … even though it may cause starvation at the client
- Protocol overhead
  - TCP header of 20 bytes in every packet
  - … which is large for sending audio samples
  - Sending ACKs for every other packet
  - … which may be more feedback than needed

## Better Ways of Transporting Data

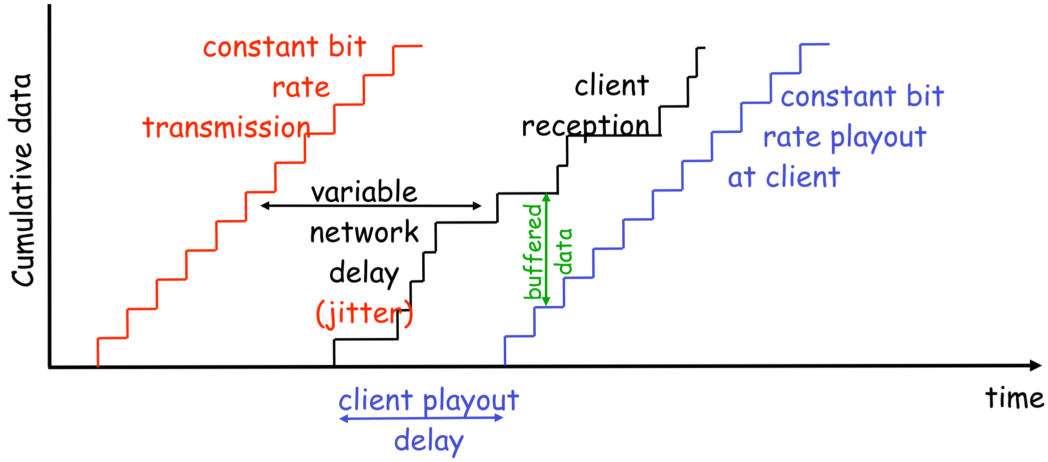
- User Datagram Protocol (UDP)
  - No automatic retransmission of lost packets
  - No automatic adaptation of sending rate
  - Smaller packet header
- UDP leaves many things to the application
  - When to transmit the data
  - How to encapsulate the data
  - Whether to retransmit lost data
  - Whether to adapt the sending rate
  - … or adapt quality of the audio/video encoding



## Recovering From Packet Loss

- Loss is defined in a broader sense
  - Does a packet arrive in time for playback?
  - A packet that arrives late is as good as lost
  - Retransmission is not useful if deadline passed
- Selective retransmission
  - Sometimes retransmission is acceptable
  - E.g., if client has not already started playing data
  - Data can be retransmitted within time constraint
- Could do Forward Error Correction (FEC)
  - Send redundant info so receiver can reconstruct

## Delay Jitter



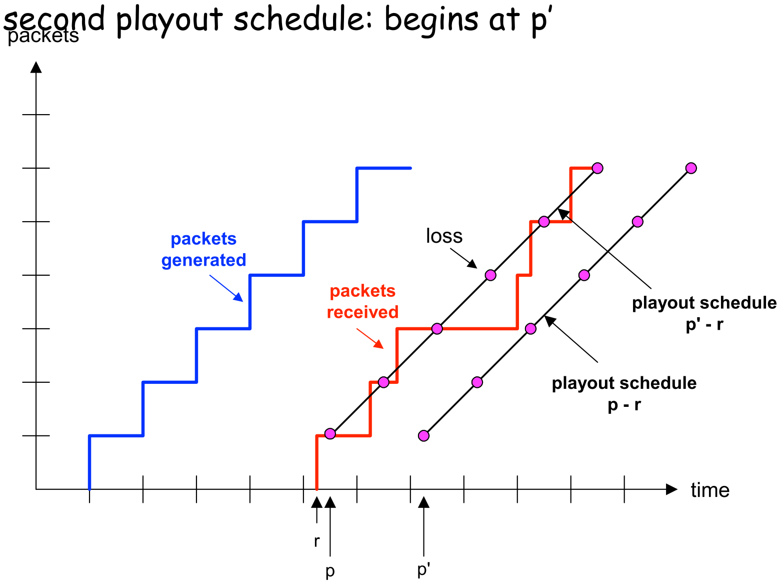

- consider end-to-end delays of two consecutive packets: difference can be more or less than 20 msec (transmission time difference)

## Internet Phone: Fixed Playout Delay

- receiver attempts to playout each chunk exactly q msecs after chunk was generated.
  - chunk has time stamp t: play out chunk at t+q .
  - chunk arrives after t+q: data arrives too late for playout, data "lost"
- tradeoff in choosing q:
  - large q: less packet loss
  - small q: better interactive experience

# Fixed Playout Delay

- sender generates packets every 20 msec during talk spurt.
- first packet received at time r
- first playout schedule: begins at p
- second playout schedule: begins at p'




# Adaptive Playout Delay (1)

- Goal: minimize playout delay, keeping late loss rate low
- Approach: adaptive playout delay adjustment:
  - estimate network delay, adjust playout delay at beginning of each talk spurt.
  - silent periods compressed and elongated.

$t_i$ = timestamp of the ith packet

$r_i$ = the time packet i is received by receiver

$p_i$ = the time packet i is played at receiver

$r_i - t_i$ = network delay for ith packet

$d_i$ = estimate of average network delay after receiving ith packet

dynamic estimate of average delay at receiver:

$$d_i = (1-u)d_{i-1} + u(r_i - t_i)$$

where u is a fixed constant (e.g., u = .01).

## Adaptive playout delay (2)

- also useful to estimate average deviation of delay, $v_i$ :

  $$v_i = (1-u)v_{i-1} + u\,|r_i - t_i - d_i|$$

- estimates $d_i$ , $v_i$ calculated for every received packet (but used only at start of talk spurt

- for first packet in talk spurt, playout time is:

  $$p_i = t_i + d_i + Kv_i$$

  where K is positive constant

- remaining packets in talkspurt are played out periodically

## Adaptive Playout (3)

Q: How does receiver determine whether packet is first in a talkspurt?

- if no loss, receiver looks at successive timestamps.
  - difference of successive stamps > 20 msec -->talk spurt begins.
- with loss possible, receiver must look at both time stamps and sequence numbers.
  - difference of successive stamps > 20 msec and sequence numbers without gaps --> talk spurt begins.

# Recovery from packet loss (1)

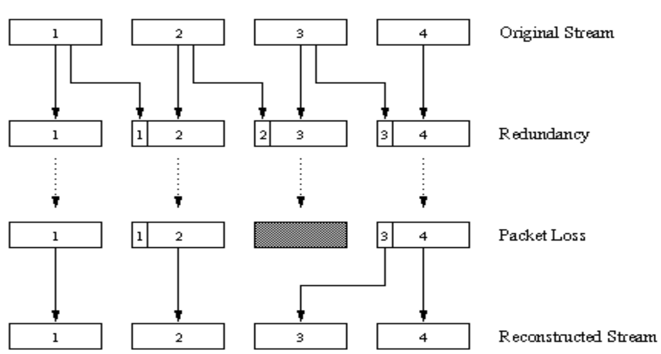
## Forward Error Correction (FEC): simple scheme

- for every group of n chunks create redundant chunk by exclusive OR-ing n original chunks
- send out n+1 chunks, increasing bandwidth by factor 1/n.
- can reconstruct original n chunks if at most one lost chunk from n+1 chunks
- playout delay: enough time to receive all n+1 packets
- tradeoff:
  - increase n, less bandwidth waste
  - increase n, longer playout delay
  - increase n, higher probability that 2 or more chunks will be lost

# Recovery from packet loss (2)

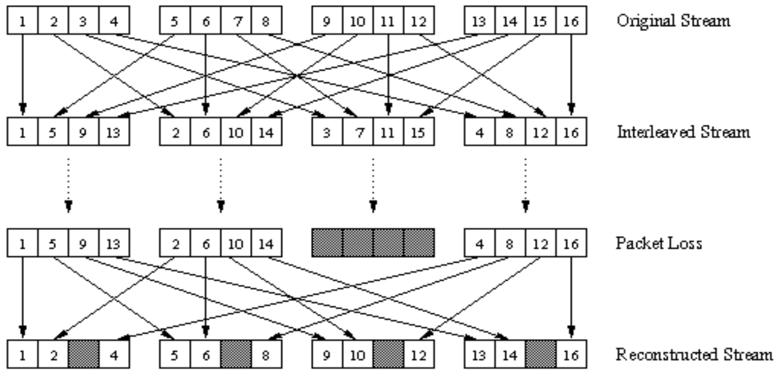
## 2nd FEC scheme

- "piggyback lower quality stream"
- send lower resolution audio stream as redundant information
- e.g., nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps.




- whenever there is non-consecutive loss, receiver can conceal the loss.
- can also append (n-1)st and (n-2)nd low-bit rate chunk

## Recovery from packet loss (3)




Interleaving

- chunks divided into smaller units
- for example, four 5 msec units per chunk
- packet contains small units

- if packet lost, still have most of every chunk
- no redundancy overhead, but increases playout delay

## Interactive Audio and Video

- Two or more users interacting
  - Telephone call
  - Video conference
  - Video game
- Strict delay constraints
  - Delays over 150-200 msec are very noticeable
  - … delays over 400 msec are a disaster for voice
- Much harder than streaming applications
  - Receiver cannot introduce much playout delay
  - Difficult if network doesn’t guarantee performance

# Conclusions

- Digital audio and video
  - Increasingly popular media on the Internet
  - Video on demand, VoIP, online gaming, IPTV…
- Interaction with the network
  - Adapt to delivering data over best-effort network
  - Adapt network to offer better quality-of-service