Multimedia Streaming

Jennifer Rexford
COS 461: Computer Networks
Lectures: MW 10-10:50am in Architecture N101

http://www.cs.princeton.edu/courses/archive/spr12/cos461/

Digital Audio and Video Data

2

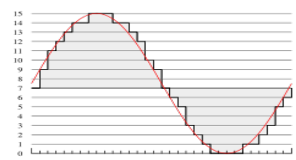
Challenges for Media Streaming

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

3

Digital Audio

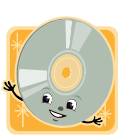
- Sampling the analog signal
  – Sample at some fixed rate
  – Each sample is an arbitrary real number
- Quantizing each sample
  – Round each sample to one of a finite # of values
  – Represent each sample in a fixed number of bits



4

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/sec
  – Sample size: 16 bits per sample
  – Rate: 705.6 kbps for mono,
  1.411 Mbps for stereo

5

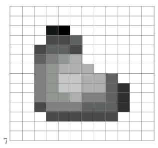
Audio Compression

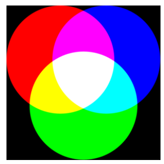
- Audio data requires too much bandwidth
  – Speech: 64 kbps is too high for some connections
  – Stereo music: 1.411 Mbps exceeds most access rates
- Compression to reduce the size
  – Remove redundancy, and details user don't perceive
- Example audio formats
  – Speech: GSM (13 kbps), G.729 (8 kbps), 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

6

Digital Video

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

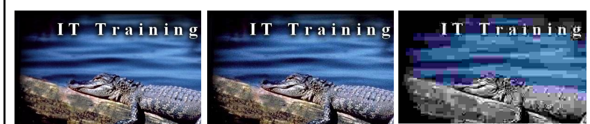




7

Video Compression: Within Image

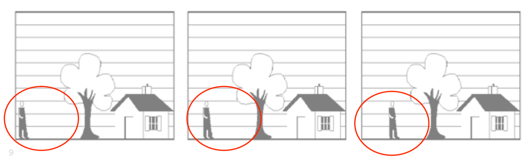
- Image compression
  - Exploit spatial redundancy (e.g., regions of same color)
  - Exploit aspects humans tend not to notice
- Common image compression formats
  - Joint Pictures Expert Group (JPEG)
  - Graphical Interchange Format (GIF)



Uncompressed: 167 KBGood 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)



9

Streaming Over the Internet

10

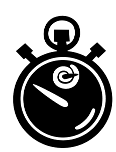
Transferring Audio and Video Data

- 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

11

Streaming Stored Audio and Video

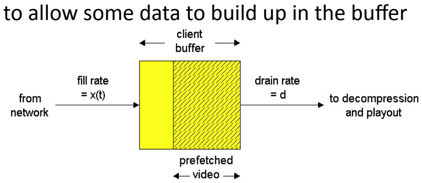
- Client-server system
  - Server stores the audio and video files
  - Clients request files, play them as they download, and perform VCR-like functions (e.g., rewind, pause)
- Playing data at the right time
  - Server divides the data into segments
  - … and labels each segment with frame id
- Avoiding starvation at the client
  - The data must arrive quickly enough



12

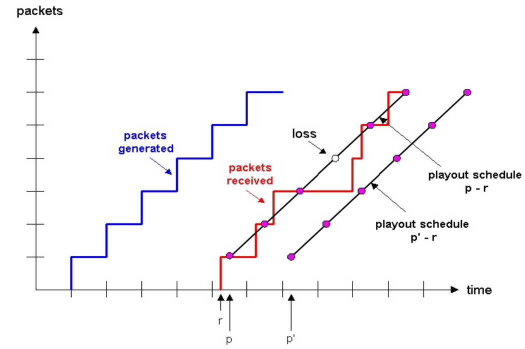
Playout Buffer

- 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



13

Influence of Playout Delay



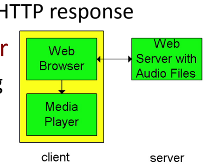
Requirements for Data Transport

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

15

Streaming From Web Servers

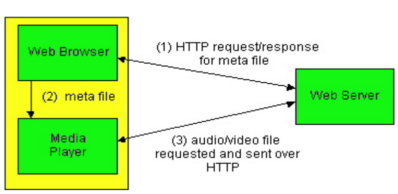
- Data stored in a file
  - Audio: an audio file
  - Video: interleaving of audio and images in a file
- HTTP request-response
  - TCP connection between client and server
  - Client HTTP request and server HTTP response
- Client invokes the media player
  - Content-type indicates encoding
  - Browser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



16

Initiating Streams from Web Servers

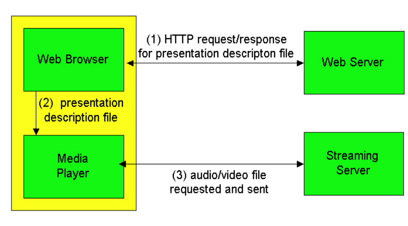
- Avoid passing all data through the Web browser
  - Web server returns a meta file describing the object
  - Browser launches media player and passes meta file
  - Player sets up its own connection to the Web server



17

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



18

TCP is Not a Good Fit

- Reliable delivery
  - Retransmission of lost packets may not be useful
- Adapting the sending rate
  - Slowing down after loss may cause starve client
- Protocol overhead
  - 20-byte TCP header is large for audio samples
  - ACKing every other packet is a lot of overhead

19

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
  - Whether to retransmit lost data
  - Whether to adapt the sending rate
  - … or adapt quality of the audio/video encoding

20

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
- 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

21

YouTube: HTTP, TCP, and Flash

- Flash videos
  - All uploaded videos converted to Flash format
  - Nearly every browser has a Flash plug-in
  - … avoids need for users to install players
- HTTP/TCP
  - Implemented in every browser
  - Easily gets through most firewalls
- Keep It Simple, Stupid
  - Simplicity more important than video quality

22

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

23

Quality of Interactive Applications

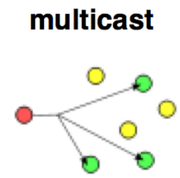
- The application can help
  - Good audio compression algorithms
  - Forward error correction
  - Adaptation to the available bandwidth
- But, ultimately the network is a major factor
  - Long propagation delay?
  - High congestion?
  - Disruptions during routing changes?

24

Multicast

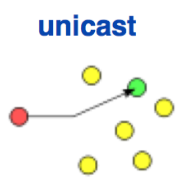
Multicast

- Many receivers
  - Receiving the same content
- Applications
  - Video conferencing
  - Online gaming
  - IP television (IPTV)
  - Financial data feeds



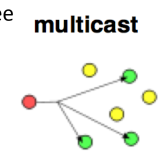
Iterated Unicast

- Unicast message to each recipient
- Advantages
  - Simple to implement
  - No modifications to network
- Disadvantages
  - High overhead on sender
  - Redundant packets on links
  - Sender must maintain list of receivers

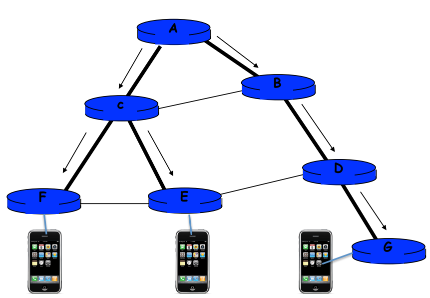


IP Multicast

- Embed receiver-driven tree in network layer
  - Sender sends a single packet to the group
  - Receivers “join” and “leave” the tree
- Advantages
  - Low overhead on the sender
  - Avoids redundant network traffic
- Disadvantages
  - Control-plane protocols for multicast groups
  - Overhead of duplicating packets in the routers



Multicast Tree



Single vs. Multiple Senders

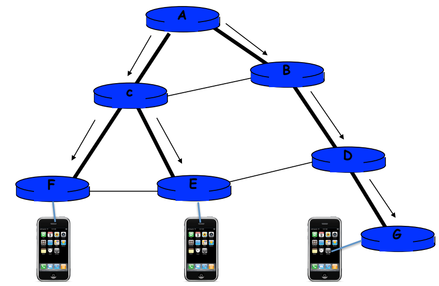
- Source-based tree
  - Separate tree for each sender
  - Tree is optimized for that sender
  - But, requires multiple trees for multiple senders
- Shared tree
  - One common tree
  - Spanning tree that reaches all participants
  - Single tree may be inefficient
  - But, avoids having many different trees

Multicast Addresses

- Multicast "group" defined by IP address
  - Multicast addresses look like unicast addresses
  - 224.0.0.0 to 239.255.255.255
- Using multicast IP addresses
  - Sender sends to the IP address
  - Receivers join the group based on IP address
  - Network sends packets along the tree

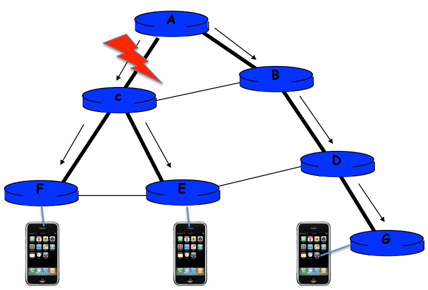
Example Multicast Protocol

- Receiver sends a "join" messages to the sender
  - And grafts to the tree at the nearest point



IP Multicast is Best Effort

- Sender sends packet to IP multicast address
  - Loss may affect multiple receivers



Challenges for Reliable Multicast

- Send an ACK, much like TCP?
  - ACK-implosion if all destinations ACK at once
  - Source does not know # of destinations
- How to retransmit?
  - To all? One bad link effects entire group
  - Only where losses? Loss near sender makes retransmission as inefficient as replicated unicast
- Negative acknowledgments more common

Scalable Reliable Multicast

- Data packets sent via IP multicast
  - Data includes sequence numbers
- Upon packet failure
  - If failures relatively rare, use Negative ACKs (NAKs) instead: "Did not receive expected packet"
  - Sender issues heartbeats if no real traffic. Receiver knows when to expect (and thus NAK)

Handling Failure in SRM

- Receiver multicasts a NAK
  - Or send NAK to sender, who multicasts confirmation
- Scale through NAK suppression
  - If received a NAK or NCF, don't NAK yourself
  - Add random delays before NAK'ing
- Repair through packet retransmission
  - From initial sender
  - From designated local repairer

Conclusions

- Digital audio and video
  - Increasingly popular media on the Internet
  - Video on demand, VoIP, online gaming, IPTV…
- Many challenges
  - Best-effort network vs. real-time applications
  - Unicast routing vs. multi-party applications
- Friday's precept
  - Hashing and partitioning to balance load