Classes of multimedia Applications

- Streaming Stored Audio and Video
- Streaming Live Audio and Video
- Real-Time Interactive Audio and Video
- Others
The multimedia content has been prerecorded and stored on a server.

User may pause, rewind, forward, etc...

The time between the initial request and display start can be 1 to 10 seconds.

**Constraint:** after display start, the playout must be continuous.
Class: Streaming Live Audio and Video

- Similar to traditional broadcast TV/radio, but delivery on the Internet
- Non-interactive just view/listen
  - Can not pause or rewind
- Often combined with multicast
- The time between the initial request and display start can be up to 10 seconds
- **Constraint:** like stored streaming, after display start, the playout must be continuous

From Xavier Appé, modified by C. Pham for educational purpose only
Class: Real-Time Interactive Audio and Video

- Phone conversation/Video conferencing
- **Constraint**: delay between initial request and display start must be small
  - Video: <150 ms acceptable
  - Audio: <150 ms not perceived, <400 ms acceptable
- **Constraint**: after display start, the playout must be continuous
Class: Others

- Multimedia sharing applications
  - Download-and-then-play applications
  - E.g. Napster, Gnutella, Freenet
- Distance learning applications
  - Coordinate video, audio and data
  - Typically distributed on CDs
Outlines

- Classes of multimedia applications
  - Requirements/Constraints
- Problems with today’s Internet and solutions
- Common multimedia protocols
  - RTP, RTCP
- Accessing multimedia data through a web server
Challenge

- TCP/UDP/IP suite provides best-effort, no guarantees on expectation or variance of packet delay

- Performance deteriorate if links are congested (transoceanic)

- Most router implementations use only First-Come-First-Serve (FCFS) packet processing and transmission scheduling
Problems and solutions

- Limited bandwidth
  - Solution: Compression

- Packet Jitter
  - Solution: Fixed/adaptive playout delay
    for Audio (example: phone over IP)

- Packet loss
  - Solution: FEC, Interleaving
Problem: Limited bandwidth

Intro: Digitalization

- Audio
  - x samples every second \((x=\text{frequency})\)
  - The value of each sample is rounded to a finite number of values (for example 256). This is called quantization

- Video
  - Each pixel has a color
  - Each color has a value
Problem: Limited bandwidth
Need for compression

- **Audio**
  - CD quality: 44100 samples per second with 16 bits per sample, stereo sound
  - $44100 \times 16 \times 2 = 1.411$ Mbps
  - For a 3-minute song: $1.441 \times 180 = 254$ Mb = 31.75 MB

- **Video**
  - For 320*240 images with 24-bit colors
  - $320 \times 240 \times 24 = 230$KB/image
  - 15 frames/sec: $15 \times 230$KB = 3.456MB
  - 3 minutes of video: $3.456 \times 180 = 622$MB

From Xavier Appé, modified by C. Pham for educational purpose only
Audio compression

- Several techniques
  - GSM (13 kbps), G.729(8 kbps), G723.3(6.4 and 5.3kbps)
  - MPEG 1 layer 3 (also known as MP3)
    - Typical compress rates 96kbps, 128kbps, 160kbps
    - Very little sound degradation
    - If file is broken up, each piece is still playable
    - Complex (psychoacoustic masking, redundancy reduction, and bit reservoir buffering)
    - 3-minute song (128kbps) : 2.8MB

From Xavier Appé, modified by C. Pham for educational purpose only
Image compression: JPEG

- Divide digitized image in 8x8 pixel blocks
- Pixel blocks are transformed into frequency blocks using DCT (Discrete Cosine Transform). This is similar to FFT (Fast Fourier Transform)
- The quantization phase limits the precision of the frequency coefficient.
- The encoding phase packs this information in a dense fashion
JPEG Compression
Video compression

- Popular techniques
  - MPEG 1 for CD-ROM quality video (1.5Mbps)
  - MPEG 2 for high quality DVD video (3-6 Mbps)
  - MPEG 4 for object-oriented video compression
Video Compression: MPEG

- MPEG uses inter-frame encoding
  - Exploits the similarity between consecutive frames
- Three frame types
  - I frame: independent encoding of the frame (JPEG)
  - P frame: encodes difference relative to I-frame (predicted)
  - B frame: encodes difference relative to interpolated frame
  - Note that frames will have different sizes
- Complex encoding, e.g. motion of pixel blocks, scene changes, ...
  - Decoding is easier then encoding
- MPEG often uses fixed-rate encoding
MPEG Compression (cont.)
MPEG System Streams

- Combine MPEG video and audio streams in a single synchronized stream
- Consists of a hierarchy with meta data at every level describing the data
  - System level contains synchronization information
  - Video level is organized as a stream of group of pictures
  - Group of pictures consists of pictures
  - Pictures are organized in slices
  - ...

From Xavier Appé, modified by C. Pham for educational purpose only
MPEG System Streams (cont.)

ISO 11172 Stream

Pack 1
- Pack Header
- System Header
- Audio Packet
- Video Packet
- Video Packet
- Video Packet

Pack 2
- Pack Header
- Video Packet
- Audio Packet
- Video Packet
- Video Packet
- Video Packet
- End Code

From Xavier Appé, modified by C. Pham for educational purpose only
MPEG System Streams (cont.)

From Xavier Appé, modified by C. Pham for educational purpose only
Problem: Packet Jitter

- Jitter: Variation in delay

Sender

| No jitter | 6 | 5 | 4 | 3 | 2 | 1 |

Receiver

| Jitter    | 5 | 6 | 4 | 3 | 2 | 1 |

- Example

pkt 6

pkt 5

From Xavier Appé, modified by C. Pham for educational purpose only
Dealing with packet jitter

- How does Phone over IP applications limit the effect of jitter?
  - A sequence number is added to each packet
  - A timestamp is added to each packet
  - Playout is delayed
Dealing with packet jitter

Fixed playout delay

- Fixed playout delay
Dealing with packet jitter
Adaptive playout delay

Objective is to use a value for $p-r$ that tracks the network delay performance as it varies during a transfer. The following formulas are used:

$$d_i = (1-u)d_{i-1} + u(r_i - t_i)$$
$$\nu_i = (1-u)\nu_{i-1} + u|r_i - t_i - d_i|$$

Where

- $t_i$ is the timestamp of the $i$th packet (the time $pkt_i$ is sent)
- $r_i$ is the time packet $i$ is received
- $p_i$ is the time packet $i$ is played
- $d_i$ is an estimate of the average network delay
- $\nu_i$ is an estimate of the average deviation of the delay from the estimated average delay
Problem: Packet loss

- Loss is in a broader sense: packet never arrives or arrives later than its scheduled playout time.
- Since retransmission is inappropriate for Real Time applications, FEC or Interleaving are used to reduce loss impact.
Recovering from packet loss
Forward Error Correction

- Send redundant encoded chunk every \( n \) chunks (XOR original \( n \) chunks)
  - If 1 packet in this group lost, can reconstruct
  - If >1 packets lost, cannot recover

- Disadvantages
  - The smaller the group size, the larger the overhead
  - Playout delay increased

From Xavier Appé, modified by C. Pham for educational purpose only
Recovering from packet loss

Piggybacking Lo-fi stream

- With one redundant low quality chunk per chunk, scheme can recover from single packet losses
Recovering from packet loss

Interleaving

- Divide 20 msec of audio data into smaller units of 5 msec each and interleave
- Upon loss, have a set of partially filled chunks
Recovering from packet loss
Receiver-based Repair

- The simplest form: Packet repetition
  - Replaces lost packets with copies of the packets that arrived immediately before the loss

- A more computationally intensive form: Interpolation
  - Uses Audio before and after the loss to interpolate a suitable packet to cover the loss

From Xavier Appé, modified by C. Pham for educational purpose only
Movie Time
Outlines

- Difference with classic applications
- Classes of multimedia applications
  - Requirements/Constraints
- Problems with today’s Internet and solutions
- Common multimedia protocols
  - RTP, RTCP
- Accessing multimedia data through a web server

From Xavier Appé, modified by C. Pham for educational purpose only
Real Time Protocol (RTP)

- RTP logically extends UDP
  - Sits between UDP and application
  - Implemented as an application library

- What does it do?
  - Framing
  - Multiplexing
  - Synchronization
  - Feedback (RTCP)
RTP packet format

- **Payload Type**: 7 bits, providing 128 possible different types of encoding; eg PCM, MPEG2 video, etc.

- **Sequence Number**: 16 bits; used to detect packet loss
RTP packet format (cont)

- **Timestamp**: 32 bytes; gives the sampling instant of the first audio/video byte in the packet; used to remove jitter introduced by the network

- **Synchronization Source identifier (SSRC)**: 32 bits; an id for the source of a stream; assigned randomly by the source
Timestamp vs. Sequence No

- Timestamps relates packets to real time
  - Timestamp value sampled from a media specific clock
- Sequence number relates packets to other packets
Audio silence example

- Consider audio data type
  - What do you want to send during silence?
    - Not sending anything
  - Why might this cause problems?
    - Other side needs to distinguish between loss and silence
  - Receiver uses Timestamps and sequence No. to figure out what happened
RTP Control Protocol (RTCP)

- Used in conjunction with RTP. Used to exchange control information between the sender and the receiver.
- Three reports are defined: Receiver reception, Sender, and Source description.
- Reports contain statistics such as the number of packets sent, number of packets lost, inter-arrival jitter.
- Typically, limit the RTCP bandwidth to 5%. Approximately one sender report for three receiver reports.

From Xavier Appé, modified by C. Pham for educational purpose only.
Outlines

- Difference with classic applications
- Classes of multimedia applications
  - Requirements/Constraints
- Problems with today’s Internet and solutions
- Common multimedia protocols
  - RTP, RTCP
- Accessing multimedia data through a web server
Streaming Stored Multimedia Example

- Audio/Video file is segmented and sent over either TCP or UDP, public segmentation protocol: **Real-Time Protocol (RTP)**

- User interactive control is provided, e.g. the public protocol **Real Time Streaming Protocol (RTSP)**
Streaming Stored Multimedia Example

- **Helper Application**: displays content, which is typically requested via a Web browser; e.g. RealPlayer; typical functions:
  - Decompression
  - Jitter removal
  - Error correction: use redundant packets to be used for reconstruction of original stream
  - GUI for user control
Streaming from Web Servers

- Audio: in files sent as HTTP objects
- Video (interleaved audio and images in one file, or two separate files and client synchronizes the display) sent as HTTP object(s)

A simple architecture is to have the Browser request the object(s) and after their reception pass them to the player for display
- No pipelining
Streaming from a Web Server (cont)

- Alternative: set up connection between server and player, then download
- Web browser requests and receives a **Meta File** (a file describing the object) instead of receiving the file itself;
- Browser launches the appropriate Player and passes it the Meta File;
- Player sets up a TCP connection with a streaming server Server and downloads the file

From Xavier Appé, modified by C. Pham for educational purpose only
Using a Streaming Server

1. HTTP request/response for presentation description file
2. Presentation description file
3. Audio/video file requested and sent

From Xavier Appé, modified by C. Pham for educational purpose only
Options when using a streaming server

- Use UDP, and Server sends at a rate (Compression and Transmission) appropriate for client; to reduce jitter, Player buffers initially for 2-5 seconds, then starts display
- Use TCP, and sender sends at maximum possible rate under TCP; retransmit when error is encountered; Player uses a much large buffer to smooth delivery rate of TCP
Real Time Streaming Protocol (RTSP)

- For user to control display: rewind, fast forward, pause, resume, etc...
- Out-of-band protocol (uses two connections, one for control messages (Port 554) and one for media stream)
- RFC 2326 permits use of either TCP or UDP for the control messages connection, sometimes called the RTSP Channel
- As before, meta file is communicated to web browser which then launches the Player; Player sets up an RTSP connection for control messages in addition to the connection for the streaming media
Meta File Example

<title>Twister</title>

<session>
  <group language=en lipsync>
    <switch>
      <track type=audio
            e="PCMU/8000/1"
            src="rtsp://audio.example.com/twister/audio.en/lofi">
        <track type=audio
              e="DVI4/16000/2" pt="90 DVI4/8000/1"
              src="rtsp://audio.example.com/twister/audio.en/hifi">
      </switch>
    </group>
  </group>
</session>

From Xavier Appé, modified by C. Pham for educational purpose only
**RTSP Operations**

```
S: RTSP/1.0 200 1 OK
Session 4231
```