Streaming Audio and Video

Multimedia on the Internet

Daniel Zappala

Brigham Young University
Computer Science Department
1 Introduction
2 Stored Media
3 CDNs
4 Streaming Protocols
5 Real Time Media
   - Overview
   - Delay
   - Loss
6 Real-Time Protocols
   - RTP
   - RTCP
   - SIP
7 Take-Home Points
Question of the Day

- How can we stream audio and video over the Internet?
Multimedia Applications

- **streaming stored audio and video**
  - content pre-recorded and stored at server
  - user begins playback before entire file received
  - content played continuously, at same rate as original recording
  - user can pause, rewind, fast-forward, index content

- **streaming live audio and video**
  - content is sent live, rather than pre-recorded, still continuous
  - higher expectation that delays are short, no pauses unless user requests it

- **real-time interactive audio and video**
  - user interaction - conversations
  - highest expectation for low delay – must be below human thresholds for interactive conversation to be possible
  - delay $< 150\text{ms}$ not perceptible, $150 - 400\text{ms}$ acceptable, $> 400\text{ms}$ not tolerable
Challenge

- the Internet offers only best-effort service
  - no guarantee on delay (affects live and interactive streaming)
  - no guarantee on throughput (affects video streaming)
Quality of Service

- network service that provides applications with performance guarantees
  - bandwidth
  - delay
  - jitter: variability of inter-packet delays

- multimedia applications are typically delay sensitive but loss tolerant

- data applications are typically loss sensitive but delay tolerant
Internet Support for QoS

- how will the Internet support QoS?
- do nothing
  - use application-level techniques to mitigate effects of delay and jitter
  - add more bandwidth as needed
  - use CDNs for stored content
  - use application-layer multicast for live content
- integrated services
  - per-flow virtual circuits to guarantee bandwidth, delay
  - must upgrade all routers and hosts
- differentiated services
  - per-class virtual circuits to guarantee bandwidth, delay to classes of traffic from an organization
  - must upgrade all routers and hosts
Audio Compression

1. analog signal sampled at constant rate
   - telephone: 8,000 samples/second
   - CD music: 44,100 samples/second

2. each sample quantized (rounded)
   - for example 256 possible values

3. each quantized value represented by bits
   - for example 8 bits for 256 values
   - 9,000 samples/second, 256 quantized values: 64,000 bps

4. receiver converts digital data back to analog signal, some quality reduction

- typical rates
  - CD: 1.411 Mbps
  - MP3: 96, 128, 160 kbps
  - VoIP: 14 kbps
**Video Compression**

- Video is a sequence of images displayed at a constant rate
  - For example, 24 frames/second
- Each frame is divided into an array of pixels
- Each pixel is represented by some bits
- Can eliminate spatial and temporal redundancy in and between frames
- Example video rates
  - MPEG 1 (CD-ROM): 1.5 Mbps
  - MPEG 2 (DVD): 3 - 6 Mbps
  - MPEG 4 (Internet): < 1 Mbps
- Layered video
  - Hierarchical encoding into layers
  - Additional layers produce increasing resolution
  - Send number of layers that path to receiver can handle
Streaming Stored Audio and Video

- applications make the best out of best-effort service
  - client buffering
  - use UDP instead of TCP
  - use multiple encodings and select the best that can be delivered
- media player responsibilities
  - remove jitter
  - decompress stream
  - conceal errors
  - user controls: pause, rewind, fast forward
Typical Streaming Architecture

2. Presentation description file is sent to media player.
3. Stream server sends multimedia file requested by multimedia player.
Client Buffering

- playout delay compensates for delay, jitter
- need adequate buffer size to prevent draining the buffer
UDP vs TCP

- **UDP**
  - server sends at rate appropriate for client (often ignores congestion control)
    - send rate = encoding rate = constant rate
    - fill rate = constant rate - packet loss
  - short playout delay (2-5 seconds) to compensate for network delay jitter
  - recover errors if time permits

- **TCP**
  - send at maximum possible rate under TCP: congestion control
  - fill rate fluctuates due to TCP congestion control
  - need a larger playout delay to smooth TCP delivery rate
  - HTTP/TCP passes more easily through firewalls
Content Distribution Networks

- serving stored video
  - high bandwidth costs
  - potential for delay and loss the greater the separation between client and server
- a CDN replicates content at hundreds of servers
  - servers located in lower-tier ISPs, clients
  - for each client CDN delivers content from the “best” server, e.g. one in the same ISP
CDN Example

- origin server links to CDN servers for objects it wants CDN to distribute
- CDN name server returns IP address of CDN server closest to the requesting host: uses a network “map”
- in this example, using CDN to distribute images
Provisioning and Dimensioning

- provide bandwidth to eliminate queueing delay and loss
  - **bandwidth provisioning**: given a topology, determine the capacity required on all links to meet performance requirements
  - **network dimensioning**: design a topology (place routers, create links, assign bandwidth to links) to meet performance requirements

- provisioning process
  - model traffic between network end points (e.g. arrival rate of users, arrival rate of packets for a given application)
  - define performance requirements
  - develop a model to predict performance given a workload
  - find a minimal cost bandwidth allocation that will meet all user requirements

- who will pay the ISPs to install sufficient bandwidth?
- will ISPs cooperate to ensure the entire path is properly provisioned?
RTSP: User Control

- RFC 2326
- client-server, application layer protocol
- user can pause, rewind, fast forward, index, etc
- does not define
  - how media is streamed over network: packet format
  - transport protocol used
  - client buffering
- messages are *out-of-band* with respect to the streaming media: port 554
RTSP Example

Client

Web browser

HTTP GET

Presentation description file

Servers

Web server

Setup

Play

Media stream

Pause

Teardown

Media player

Media server
<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">
          <track type="video/jpeg"
src="rtsp://video.example.com/twister/video">
        </switch>
      </track type=audio>  
    </group>
  </session>
RTSP Protocol Example

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
Transport: rtp/udp; compression; port=3056; mode=PLAY
S: RTSP/1.0 200 1 OK
   Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
   Session: 4231
   Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
   Session: 4231
   Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
   Session: 4231

S: 200 3 OK
Internet Phone Example

- streaming audio: alternating talk spurts, silent periods
  - packets generated only during talk spurts
  - talk spurt = 64 kbps
  - sample every 20 ms: 160 bytes per packet
- send packets using UDP
- ideally, during a talk spurt a packet arrives every 20ms and you play it
  - loss
  - varying delay
- current applications must make the best of best-effort service
Effects of Loss and Delay

- **loss**
  - can be tolerated, based on encoding and concealing methods (e.g. FEC)
  - excessive loss (10 to 20%) is rare, but deadly

- **delay**
  - must be less than 400 ms
  - less than 150 ms is ideal

- **jitter**
  - packets experience different amounts of queueing delay
  - can be eliminated with buffering, but this increases delay
Coping With Delay

- use buffering to compensate for delay, jitter
  - play packets from the buffer at a certain rate
  - want to prevent draining the buffer too early
  - want to prevent delaying the packets too much: 400 ms maximum, but less is better

![Diagram of buffer system](image.png)
Fixed Playout Delay

- play packet $i$ at $t_i + q_i$
  - $t_i$: time the packet was generated
  - small $q$: better real-time interaction
  - large $q$: fewer missed playouts
Adaptive Playout Delay

- **goal:** minimize playout delay, infrequent missed playouts
- for start of talk spurt, play packet $i$ at $p_i = t_i + d_i + K v_i$
  - $d_i = \text{estimate of delay (EWMA)}$
  - $v_i = \text{estimate of delay variation (EWMA)}$
  - $K = \text{constant, e.g. } K = 4$
- for other packets, $p_j = t_j + p_i - t_i$
  - use the same offset from $t$ as beginning of talk spurt
  - application may denote start of talk spurt
- because playout delay is adaptive, silent periods may be compressed or elongated
Coping With Loss

- **Forward Error Correction (FEC):** add some redundant data to the stream
  - for every group of \( n \) packets, create a redundant packet: exclusive-OR of the \( n \) original packets
  - send \( n + 1 \) packets, increasing bandwidth by \( 1/n \)
- receiver can reconstruct the stream with any \( n \) packets
  - must wait for \( n \) packets before playout
- **trade-offs**
  - larger \( n \): less bandwidth
  - smaller \( n \): shorter playout delay, smaller chance of two packets out of \( n \) being lost
- **simple version of FEC:** see RFC 2733
add lower-resolution audio to each packet
substitute lower quality when needed: Free Phone, RAT
many other more complex kinds of FEC available
Interleaving

- Interleave smaller (5 ms) pieces among packets
- If a packet is lost, still have most of the stream
- No redundancy overhead, but added playout delay
RTP: Real-Time Protocol

- specifies packet structure for audio and video streams
  - payload identification
  - sequence number
  - timestamp
- used for interoperability between multimedia applications
- runs on top of UDP
- RFC 1889
RTP Header

- **payload type (7 bits):** type of encoding, e.g. PCM, GSM, JPEG, MPEG audio, MPEG video
- **sequence number (16 bits):** detect packet loss and order packets
- **timestamp (32 bits):** sampling time of first byte
- **SSRC (32 bits):** source of RTP stream - allows multiple sources per session
RTCP: Real-Time Control Protocol

- works with RTP
- each participant in RTP session periodically transmits RTCP control packets to other participants
  - statistics from sender and receiver on session quality
- use feedback to control performance, e.g. congestion control, type of encoding
- packets sent using same mechanism as data (unicast or multicast)
RTCP Packets

- **source description:** sender’s name, host name, email address, SSRC of associated RTP stream
- **sender report:** SSRC of RTP stream, current time, number of packets sent, number of bytes sent
- **receiver report:** fraction of packets lost, last sequence number, average jitter
RTCP Extras

- synchronization of different media streams in an RTP session
  - timestamps in audio and video streams tied to sampling clocks, not wall-clock time
  - each RTCP sender report contains timestamp of most recent RTP packet and wall-clock time of when it was created

- bandwidth scaling
  - RTCP will limit control traffic to 5%
  - e.g. if sender transmits video at 2 Mbps, then RTCP tries to use 100 kbps
  - 75% of rate for receivers, 25% to sender
  - receivers try to share 75% equally
  - receiver determines RTCP transmission period by dividing average RTCP packet size by allocated rate
SIP: Session Initiation Protocol

- vision
  - all telephone and video conference calls take place over the Internet
  - people are identified by names and email addresses rather than by phone numbers
  - you can reach the person you are calling wherever she is on the Internet
- call setup
  - start and end call
  - agree on media type and encoding
- map name and email to IP address
- call management
  - add new streams during call
  - change encoding during call
  - invite other users to call
  - transfer and hold calls
SIP Call Setup

- example assumes Alice knows Bob’s IP address
- Alice provides port number, IP address, preferred encoding
- Bob responds with new port number, IP address, preferred encoding
- SIP messages can use TCP or UDP, default port 5060
encoding negotiation
  - may not have requested encoder
  - may prefer a different one
  - respond with **606 Not Acceptable** and list encoders
  - sender can send a new invitation with new encoder

can reject a call
  - *busy, gone, it payment required, forbidden*

media can use RTP or any other protocol

syntax is similar to HTTP
Name Translation and User Location

- need to map user name or email address to IP address
  - mobility
  - changing IP addresses due to DHCP
  - many different IP devices per user
  - call forwarding
- **SIP registrar**: clients register to provide current location and IP address (similar to instant messaging)
- **SIP proxy**: find callee on behalf of caller (similar to DNS server)
Session Initiation Example

1. SIP client 217.123.56.89 initiates a connection.
2. SIP client 217.123.56.89 contacts SIP registrar upenn.edu.
3. SIP client 217.123.56.89 requests service through SIP proxy umass.edu.
4. SIP registrar eurcom.fr responds to SIP client 217.123.56.89.
5. SIP proxy umass.edu relays the response.
6. SIP client 197.87.54.21 answers the request.
7. SIP registrar eurcom.fr confirms the connection.
8. SIP client 217.123.56.89 establishes the session with SIP client 197.87.54.21.
9. SIP client 217.123.56.89 and SIP client 197.87.54.21 exchange media streams.

SIP registrar
upenn.edu

SIP proxy
umass.edu

SIP registrar
eurcom.fr

SIP client
217.123.56.89

SIP client
197.87.54.21
Take-Home Points

- **stored versus real-time streaming**
  - stored: lower delay expectation
  - real-time: very low delay required for interactivity
  - use CDN, provisioning to provide low enough loss, high enough bandwidth for good quality streaming

- **real-time applications are typically delay sensitive but loss tolerant**
  - use buffering and adaptive playout delay to conceal delay, jitter
  - use FEC to conceal loss

- **functionality provided by RTP - RTCP - SIP**
  - RTP: packet format for compatibility
  - RTCP: sender and receiver reports
  - SIP: session initiation