1. Making the Best of Best-Effort Service
   - Coping with Delay
   - Coping With Loss
   - Content Distribution Networks
   - Provisioning and Dimensioning

2. Protocols for Real-Time Interactive Applications
   - RTP
   - RTCP
   - SIP

3. Take-Home Points
Question of the Day

How can we stream audio and video over the Internet?
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
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
Making the Best of Best-Effort Service
Protocols for Real-Time Interactive Applications
Take-Home Points

Coping with Delay

Coping With Loss
Content Distribution Networks
Provisioning and Dimensioning

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 $p_i = t_i + d_i + K v_i$
  - $d_i =$ estimate of delay (EWMA)
  - $v_i =$ estimate of delay variation (EWMA)
  - $K =$ 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
FEC: Forward Error Correction

- 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
FEC With Lower Quality Redundant Data

- 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
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

Content Distribution Networks

Origin server in North America

CDN distribution node

CDN server in South America

CDN server in Europe

CDN server in Asia
Making the Best of Best-Effort Service
Protocols for Real-Time Interactive Applications
Take-Home Points
Coping with Delay
Coping With Loss
Content Distribution Networks
Provisioning and Dimensioning

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?
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
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
SIP Extras

- 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 sends a request to a SIP proxy at umass.edu.
2. SIP proxy forwards the request to the SIP registrar at upenn.edu.
3. SIP registrar at upenn.edu forwards the request to the SIP registrar at eucom.fr.
4. SIP registrar at eucom.fr forwards the request to the SIP client at 197.87.54.21.
5. SIP client at 197.87.54.21 responds with an acknowledgment.
6. SIP registrar at eucom.fr sends an acknowledgment to the SIP proxy at umass.edu.
7. SIP proxy forwards the acknowledgment to the SIP registrar at upenn.edu.
8. SIP registrar at upenn.edu forwards the acknowledgment to the SIP client at 217.123.56.89.
9. SIP client at 217.123.56.89 sends a final acknowledgment back to the SIP client at 197.87.54.21.
Take-Home Points

- use buffering and adaptive playout delay to conceal delay, jitter
- use FEC to conceal loss
- use provisioning to provide low enough loss, high enough bandwidth for good quality streaming – someone must pay
- functionality provided by RTP - RTCP - SIP
  - RTP: packet format for compatibility
  - RTCP: sender and receiver reports
  - SIP: session initiation