Voice-Over-IP

Daniel Zappala

CS 460 Computer Networking
Brigham Young University

Coping with Best-Effort Service

- sample application
  - send a 160 byte UDP packet every 20ms
  - packet carries a voice sample plus a header
- *packet loss*
  - can send with TCP, but ...
    - extra delay due to retransmission and reordering may be unacceptable for a conversation
    - rate may become too slow
  - can use FEC
    - send redundant information in each packet
    - but if loss exceeds 10 to 20 percent, nothing you can do

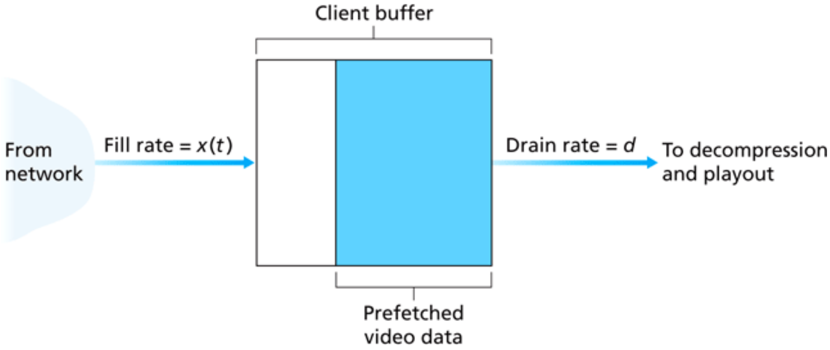
Coping with Best-Effort Service

- *packet delay*
  - need delays less than 400ms
  - consider any packets delayed excessively to be lost
  - nothing else an application can do
- *packet jitter*
  - variation in interpacket delays
  - can be larger or smaller than the original delay between packets
  - use buffering and then play out smoothly

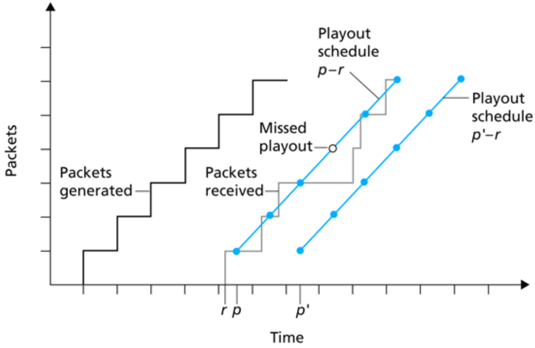
Removing Jitter

Removing Jitter

- include a timestamp on each packet of voice data
- use buffering and delay playout
    - 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



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 $p_i = t_i + d_i + Kv_i$
    - $d_i =$ estimate of delay (EWMA)
    - $v_i =$ estimate of delay variation (EWMA)
    - $K =$ constant, e.g. $\mathrm{K} = 4$
- 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

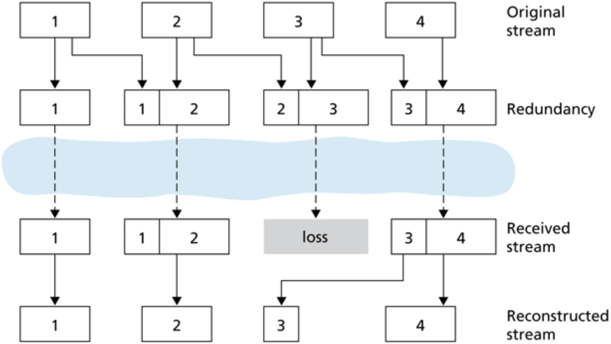
Recovering from Packet Loss

Forward Error Correction (FEC)

- **FEC with Redundant Data**
  - 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

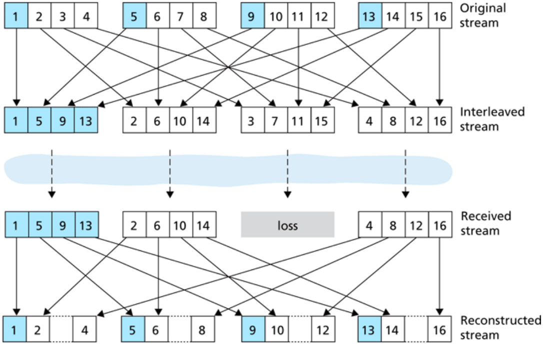
Forward Error Correction (FEC)

- **FEC With Lower Quality 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

Error Concealment

- alternatives
  - replay the last packet
  - interpolation
- usually good enough to fool the human ear for small loss rates and small amounts of lost data

VoIP with Skype

Skype

- voice and video
- codecs at various rates
  - 3 kbps to 1 Mbps video
  - voice sampled at 16,000 samples/s instead of 8,000
- transport
  - sent via UDP unless blocked by firewall
  - control packets over TCP
- FEC for loss recovery
- adapts encoding and FEC based on network conditions

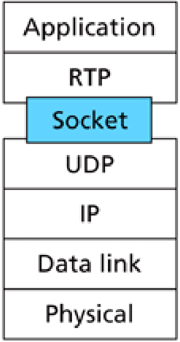
Skype P2P

- superpeers and ordinary peers
  - super peers keep an index mapping Skype username to IP address and ports
  - likely using a DHT
- both caller and callee may have NAT
  - superpeer arranges for a non-NATed super-peer to act as relay
  - caller sends data to callee through relay, and vice versa
- multiparty communication
  - everyone sends audio packets to person who started the call
  - this person combines all audio into a single stream, sends combined stream to everyone else
  - for video, everone sends packets to a relay, which sends out unaltered streams to everyone else – need higher uploading bandwidth

Real-Time Streaming Protocols

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



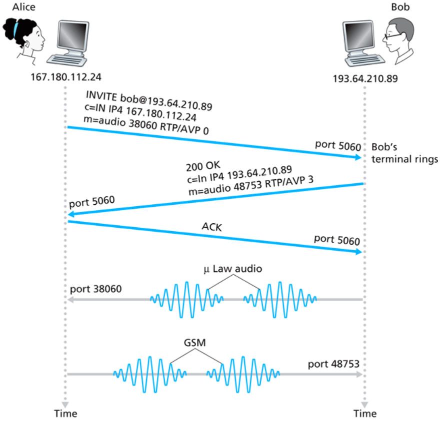
RTP Header

- 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

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