Streaming Media
- Classes of applications:
  - streaming stored video/audio
  - streaming live video/audio
  - real-time interactive video/audio
- Characteristics:
  - typically delay-sensitive
  - delay
  - jitter
  - loss tolerant:
    - drop/lose some ‘frames’
    - example: MPEG
  - user-specific quality needs:
    - image size, resolutions, color depth, frame rate, ...

Real-Time Streaming Protocol
HTTP
- does not target multimedia content
- no commands for fast forward, etc.
RTSP: RFC 2326
- client-server application layer protocol.
- for user to control display: rewind, fast forward, pause, resume, repositioning, etc...
What it doesn’t do:
- does not define how audio/video is encapsulated for streaming over network
- does not restrict how streamed media is transported; it can be transported over UDP or TCP
- does not specify how the media player buffers audio/video

Out of Band Control
FTP uses an "out-of-band" control channel:
- a file is transferred over one channel.
- control information (directory changes, file deletion, file renaming, etc.) is sent over a separate TCP connection.
- the "out-of-band" and "in-band" channels use different port numbers.
RTSP messages are also sent out-of-band:
- the RTSP control messages use different port numbers than the media stream, and are therefore sent out-of-band.
- the media stream, whose packet structure is not defined by RTSP, is considered "in-band".
- if the RTSP messages were to use the same port numbers as the media stream, then RTSP messages would be said to be "interleaved" with the media stream.

RTSP Operation

Meta File Example
<title>Twister</title>
<session>
  <group language=en lipsync>
    <switch>
      <track type=audio
        e="PCMUN8000/1"
        src="/http/audio.example.com/twister/audio.en/lofi/">
      </track>
    </switch>
    <track type=audio
      e="DV1416000/11"
      p="96 DV1416000/11"
      src="/http/audio.example.com/twister/audio.hifi/">
    </track>
    <track type=Video
      e="DV1416000/11"
      src="/http/video.example.com/twister/video/">
    </track>
  </group>
</session>
RTSP Exchange 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
RTSP/1.0
Session: 4231
Range: npt=0-
C: PAUSE
rtsp://audio.example.com/twister/audio
RTSP/1.0
Session: 4231
Range: npt=37
C: TEARDOWN
rtsp://audio.example.com/twister/audio
RTSP/1.0
Session: 4231
S: 200 3 OK

RTSP Streaming Caching

- Caching of RTSP response messages makes little sense.
- But desirable to cache media streams closer to client.
- Much of HTTP/1.1 cache control has been adopted by RTSP.
- Cache control headers can be put in RTSP SETUP requests and responses:
  - If-modified-since:
  - Expires:
  - Via:
  - Cache-Control:
- Proxy cache may hold only segments of a given media stream.
- Proxy cache may start serving a client from its local cache, and then have to connect to origin server and fill missing material, hopefully without introducing gaps at client.

Packet Loss

- **Network loss**: IP datagram lost due to network congestion (router buffer overflow)
- **Delay loss**: IP datagram arrives too late for playout at receiver
  - Delays: processing, queuing in network; end-system (sender, receiver) delays
  - Tolerable delay depends on the application
- **How can packet loss be handled?**

Receiver-Based Packet Loss Recovery

- Generate replacement packet
  - Packet repetition
  - Interpolation
  - Other sophisticated schemes
- Works when audio/video stream exhibits short-term self-similarity
- Works for relatively low loss rates (e.g., < 5%)
- Typically, breaks down on "bursty" losses

Forward Error Correction FEC

- For every group of n packets generate k redundant packets
- Send out n+k packets, increasing the bandwidth by factor k/n.
- Can reconstruct the original n packets provided at most k packets are lost from the group
- Works well at high loss rate (for a proper choice of k)
- Handles "bursty" packet losses
- Cost: increase in transmission cost (bandwidth)

FEC Example
Interleaving

Real-Time Protocol RTP
- RTP specifies the packet structure for packets carrying audio and video data: RFC 1889.
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - time-stamping
- RTP runs in the end systems.
- RTP packets are encapsulated in UDP segments

RTP Example
- Consider sending 64 kbps PCM-encoded voice over RTP.
- Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk.
- The audio chunk along with the RTP header form the RTP packet, which is encapsulated into a UDP segment.
- RTP header indicates type of audio encoding in each packet; senders can change encoding during a conference. RTP header also contains sequence numbers and timestamps.

RTP and QoS
- RTP does not provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- RTP encapsulation is only seen at the end systems -- it is not seen by intermediate routers.
- Routers providing the Internet's traditional best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.
- In order to provide QoS to an application, the Internet must provide a mechanism, such as RSVP, for the application to reserve network resources.

RTP Streams
- RTP allows each source (for example, a camera or a microphone) to be assigned its own independent RTP stream of packets.
- For example, for a videoconference between two participants, four RTP streams could be opened: two streams for transmitting the audio (one in each direction) and two streams for the video (again, one in each direction).
- However, some popular encoding techniques -- including MPEG1 and MPEG2 -- bundle the audio and video into a single stream during the encoding process. When the audio and video are bundled by the encoder, then only one RTP stream is generated in each direction.
- For a many-to-many multicast session, all of the senders and sources typically send their RTP streams into the same multicast tree with the same multicast address.

RTP Header
- Payload Type (7 bits): Used to indicate the type of encoding that is currently being used.
  - If a sender changes the encoding in the middle of a conference, the sender informs the receiver through this payload type field.
    - Payload type 0: PCM, 64 Kbps
    - Payload type 1: G.726, 16 Kbps
    - Payload type 2: G.728, 64 Kbps
    - Payload type 3: G.723.1, 56 Kbps
    - Payload type 4: H.323, 64 Kbps
    - Payload type 5: H.320, 64 Kbps
    - Payload type 6: H.261
    - Payload type 7: MPEG
- Sequence Number (16 bits): The sequence number increments by one for each RTP packet sent; may be used to detect packet loss and to restore packet sequence.
RTP Header
- Timestamp field (32 bytes long). Reflects the sampling instant of the first byte in the RTP data packet. The receiver can use the timestamps to remove packet jitter and provide synchronous playout. The timestamp is derived from a sampling clock at the sender.
  - As an example, for audio the timestamp clock increments by one for each sampling period (for example, each 125 usecs for a 8 KHz sampling clock); if the audio application generates chunks consisting of 160 encoded samples, then the timestamp increases by 160 for each RTP packet when the source is active. The timestamp clock continues to increase at a constant rate even the source is inactive.
- SSRC field (32 bits long). Identifies the source of the RTP stream. Each stream in a RTP session should have a distinct SSRC.

Real-Time Control Protocol RTCP
- Works in conjunction with RTP.
- Each participant in an RTP session periodically transmits RTCP control packets to all other participants. Each RTCP packet contains sender and/or receiver reports that report statistics useful to the application.
- Statistics include number of packets sent, number of packets lost, inter-arrival jitter, etc.
- This feedback of information to the application can be used to control performance and for diagnostic purposes.
  - The sender may modify its transmissions based on the feedback.

RTCP

<table>
<thead>
<tr>
<th>VER</th>
<th>P</th>
<th>RC</th>
<th>PTYPE</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA AREA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- VER (2 bits): version number
- P (1 bit): padding (length of padding stored in last octet of padding)
- RC (5 bits): "report count", number of reports following the header
- PTYPE (8 bits): packet/message type
- LENGTH (16 bits): total packet length
- DATA AREA: sequence of report records

RTCP Bandwidth Scaling
- RTCP attempts to limit its traffic to 5% of the session bandwidth.
  - For example, suppose there is one sender, sending video at a rate of 2 Mbps. Then RTCP attempts to limit its traffic to 100 Kbps.
  - The protocol gives 75% of this rate, or 75 kbps, to the receivers; it gives the remaining 25% of the rate, or 25 kbps, to the sender.
  - The 75 kbps devoted to the receivers is equally shared among the receivers. Thus, if there are R receivers, then each receiver gets to send RTCP traffic at a rate of 75/R kbps and the sender gets to send RTCP traffic at a rate of 25 kbps.
  - A participant (a sender or receiver) determines the RTCP packet transmission period by dynamically calculating the average RTCP packet size (across the entire session) and dividing the average RTCP packet size by its allocated rate.
**Synchronization of Streams**

- RTCP can be used to synchronize different media streams within a RTP session.
- Consider a videoconferencing application for which each sender generates one RTP stream for video and one for audio.
- The timestamps in these RTP packets are tied to the video and audio sampling clocks, and are not tied to the wall-clock time (i.e., to real time).
- Each RTCP sender-report packet contains, for the most recently generated packet in the associated RTP stream, the timestamp of the RTP packet and the wall-clock time for when the packet was created. Thus the RTCP sender-report packets associate the sampling clock to the real-time clock.
- Receivers can use this association to synchronize the playout of audio and video.

**RTSP, RTP, RTCP**

- Lots of multimedia tools use RTSP
- Libraries, sample tools (MP3 players, DVD players, etc), source code available
- Textbook: Chapter 28: link to author’s library

**SIP**

- Session Initiation Protocol
- Comes from IETF
- Leading Voice over IP protocol, replacing H.323

**SIP Services**

- Setting up a call
  - Provides mechanisms for caller to let callee know she wants to establish a call
  - Provides mechanisms so that caller and callee can agree on media type and encoding.
  - Provides mechanisms to end call.
- Determine current IP address of callee.
- Call management
  - Add new media streams during call
  - Change encoding during call
  - Invite others
  - Transfer and hold calls

**Comparison with H.323**

- H.323 is another signalling protocol for real-time, interactive streaming.
- H.323 is a complete, vertically integrated suite of protocols for multimedia conferencing: signalling, registration, admission control, transport and codecs.
- SIP is a single component. Works with RTP, but does not mandate it. Can be combined with other protocols and services.
- H.323 comes from the ITU (telephony).
- SIP comes from IETF: Borrows much of its concepts from HTTP. SIP has a Web flavor, whereas H.323 has a telephony flavor.
- SIP uses the KISS principle: Keep it simple stupid.

**H.323**

- Foundation for audio and video conferencing across IP networks.
  - Umbrella recommendation from the ITU
  - Implemented by applications such as NetMeeting, GnomeMeeting, etc.
- Very broad in scope:
  - stand-alone devices (e.g., Web phones,)
  - applications in PCs
  - point-to-point and multipoint conferences
- H.323 specification includes:
  - How endpoints make and receive calls.
  - How endpoints negotiate common audio/video encodings.
  - How audio and video chunks are encapsulated and sent over network.
  - How audio and video are synchronized (lipsync).
  - How endpoints communicate with their respective gatekeepers.
  - How Internet phones and PSTN/ISDN phones communicate.
H.323 Endpoints Must Support

- **G.711**: ITU standard for speech compression
- **RTP**: protocol for encapsulating media chunks into packets
- **H.245**: "Out-of-band" control protocol for controlling media between H.323 endpoints.
- **Q.931**: A signalling protocol for establishing and terminating calls.
- **RAS** (Registration/Admission/Status) channel protocol - Protocol for communicating with a gatekeeper (if gatekeeper is present).

Audio Source Encoding:
- e.g., G.711 or G.723.1
- RTP packet encapsulation
- UDP socket
- Internet or Gatekeeper

H.323 Encoding

Audio:
- H.323 endpoints must support G.711 standard for speech compression.
- Uses PCM, 8000 samples/second.
- G.711 transmits voice at 64 kbps.
- Optional: G.722, G.726, G.729

Video:
- Video capabilities for an H.323 endpoint are optional.
- Any video-enabled H.323 endpoint must support the QCIF H.261 (176x144 pixels).
- Optionally supports other H.261 schemes: CIF, 4CIF and 16CIF.
- H.264 is used with communication channels that are multiples of 64 kbps

Video Codecs

- **H.261** (1990)
  - ITU
  - resolutions: QCIF, CIF
- **H.263** (1995)
  - ITU, better performance, error recovery
  - resolutions: SQCIF, QCIF, CIF, 4CIF, 16CIF
- MPEG-4: AVC (aka H.264) (1998-)
  - MPEG-1, MPEG-2 plus new features: VRML, support for Digital Rights Management, animation support, ...