Streaming Multimedia Applications
Multimedia Networking
Multimedia Applications?

• **What are they?**
  – An application that deals with one of more of the following data types:
    • Text
    • Images
    • Audio
    • Video

• **Most common scenario today:**
  – Transmission, processing, and rendering multimedia information over the network.
Some Example Applications

- Common multimedia applications on the Internet:
  - Streaming stored audio and video.
  - Streaming live audio and video.
  - Real-time interactive audio and video.
- All the above have common characteristics:
  - Delay sensitivity.
    - End-to-end packet delay.
    - Delay jitter :: variability of packet delay within the same packet stream.
– Can tolerate packet losses.
  • Occasional packet losses cause minor disturbances during playback.

• Requirement is just the reverse as compared to normal data transmission.
  – Cannot tolerate losses.
  – Can tolerate delay variations.
Application QoS Categories

• **Hard QoS:**
  – The application may malfunction if the QoS constraints cannot be met.
  – Typical examples:
    • Critical patient monitoring systems.
    • Missile control systems.

• **Soft QoS:**
  – Functionally application performs correctly.
  – Typical examples:
    • Most multimedia applications.
Streaming Stored Multimedia

- **Basic concept:**
  - The basic media file is stored at the source.
  - The file is transmitted to the client when requested.
  - The client starts playing the media before the whole of it is transferred.
- **Central concept to streaming.**
  - Minimum continuous rate of transfer to be maintained for jitter-free playback.
Client playing a part of the video, and sever sending the later part, are carried out in overlapped fashion.
• **Typical client functionality:**
  – Pause, fast forward, play, rewind, etc., just like normal media players.
  – An initial delay (5-10 sec) for the client to get resynchronized with the origin server.
Streaming Live Multimedia

• **Basic concept:**
  – Multimedia content not stored anywhere a priori.
    • Generated on the fly and broadcasted.
  – **Typical examples:**
    • Live news feed.
    • Live cricket match over the Internet.
  – **Client usually has a playback buffer.**
    • Content buffered during transmission.
    • Allows rewind (but no fast forward).

• **Other constraints:**
  – Depending on the latency of the path, the live stream may play on the desktop after an appreciable delay (10-20 sec).
  – **Timing constraint for jitter-less playback is still present.**
### Real-time Interactive Multimedia

- **Basic concept:**
  - Interactive in the sense that the content to be transmitted is decided by the end parties dynamically.
  - Typical examples:
    - IP telephony
    - Video conferencing
    - On-line games
  - End-to-end delay requirements are important.
    - Includes application-level and also network delays.
    - About 200 msec considered to be good enough for audio.
    - Beyond 500 msec, audio may be unacceptable.
How Internet Handles Multimedia Today?

• Internet is driven by TCP/UDP/IP.
  – Multimedia transport takes place on top of these only.
  – No guarantee on throughput, losses, etc.

• Internet multimedia applications use application-level techniques to get the best out of the underlying service.

• Next generation Internet can handle this much better.
Streaming Multimedia
Multimedia on Internet

• **The simple approach:**
  – Multimedia object stored as a file on the web server.
  – File transferred to client as HTTP object.
  – Client receives the whole file and stores it in a buffer.
  – Client invokes the media player to play the received file.

• **Basically:**
  – No streaming, no pipelining, long delay.
• **The streaming approach:**
  – Browser requests for a “metafile” from the web server.
  – Browser launches the media player, and passes the “metafile” to it.
  – The media player directly contacts the web server using HTTP.
  – Server streams audio/video object in its HTTP response to the media player.
  – Usually considered unsatisfactory:
    • Little control, non-interactive.
• **Using a separate streaming server:**
  
  – Provides the best performance.
  – This architecture can use non-HTTP (may be proprietary) protocols between server and media player.
  – Can also use UDP instead of TCP.
    • For better response.
Web Browser

Web Server

Media Player

Streaming Server
Some Issues in Streaming

- Use of client buffering
  - Allows to compensate for network-added delay and delay jitter.

- Whether to use TCP or UDP ....
  - TCP
    - Transfer rate fluctuates due to TCP congestion control.
    - Better quality because no packets are lost.
    - More delay variations due to retransmission.
  - UDP
    - Server sends data at rate appropriate for client. Does not depend on network congestion.
    - Send rate = encoding rate = constant rate
    - Short playout delay (2-5 seconds) to compensate for network delay jitter.
• **Variability in client rates**
  – How to handle variations in client receive rate capabilities?
    • 33 Kbps dialup
    • 2 Mbps leased line
    • 100 Mbps Ethernet
  – **Common solution:**
    • Server stores multiple copies of the content (say, video), that have been encoded at different rates.
Real Time Streaming Protocol

- **RTSP**
  - Gives user much better control over streaming media.
- **RTSP is .....**
  - A client-server application-layer protocol.
  - Provides control to the user:
    - Pause, play, rewind, forward, repositioning, etc.
- **What RTSP is not ....**
  - Does not specify how the media is encoded and compressed.
  - Does not restrict the transport layer protocol.
    * Can be TCP or UDP.
  - Details about client-side buffering is also not specified.
How RTSP works?

- Like the FTP protocol, RTSP also uses “out-of-band” control.
  - RTSP control messages uses different port number than the media stream.
    - Port 554.
  - The media stream is considered “in-band”.
- Typical scenario:
  - The “metafile” is first sent to the web browser over HTTP.
  - The browser launches media player.
  - The media player sets up an RTSP control connection, and a data connection to the streaming server.
- Two servers:
  - A web server
  - A streaming server
A Typical Metafile

<title>Trailer</title>
<session>
  <group language=en>
    <switch>
      <track type=audio
        e="PCMU/8000/1"
        src = "rtsp://stream.com/trailer/audio.en/lofi">
      <track type=audio
        e="DVI4/16000/2" pt="90 DVI4/8000/1"
        src="rtsp://stream.com/trailer/audio.en/hifi">
      </switch>
    <track type="video/jpeg"
      src="rtsp://stream.com/trailer/video.en">
  </group>
</session>
RTSP Operation

- **Web Browser** to **Web Server** via HTTP GET
- **Presentation description**
- **Media Player** to **Media Server**
- **SETUP**
- **PLAY**
- **MEDIA STREAM**
- **PAUSE**
- **TEARDOWN**
RTSP Exchange Example

C: SETUP rtsp://stream.com/trailer/audio RTSP/1.0
    Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 OK
    Session 4231

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

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

C: TEARDOWN rtsp://stream.com/trailer/audio.en/lofi RTSP/1.0  Session: 4231

S: RTSP/1.0 200 OK
Internet Telephony
Introduction

• A classic application of interactive multimedia over Internet.
• Voice chat (PC-to-PC, say) over the Internet.
• How is basically works?
  – The speaker speaks into the microphone connected to the PC.
    • Alternating talk spurts and silent periods.
    • Data rate of 8000 bytes per second generated during each talk spurt (64 Kbps).
  – Packets get generated only during the talk spurts,
    • Every 20 msec, the sender gathers the data into chunks => 160 bytes per chunk (maximum).
  – Application-layer header is added to each chunk.
  – The data chunk and the header is encapsulated into a UDP packet.
  – The UDP packets are transmitted.
• **To summarize:**
  – An UDP packet gets transmitted every 20 msec during a talk spurt.
  – No packets generate during idle periods.
Packet Loss Analysis

- Two main reasons for quality loss:
  - Normal packet loss
    - Some IP packets are lost, and are not delivered at the destination.
  - Loss due to excessive delay
    - An IP packet arrives, but too late to be played.
    - Delays < 150 msec are normally not detected. Delays > 400 msec can be annoying.
- Depending on encoding technique, packet loss rate of up to 20% can be tolerated.
Handling Jitters

• Variable end-to-end delays in consecutive packets can cause jitters.

• How to handle / remove jitters?
  – Use sequence number with each packet.
    • Out-of-order playback can be avoided.
  – Timestamps in the packet header.
    • Works similar to sequence number.
  – Delayed playout
    • The playout of packets is delayed long enough so that most of the packets are received before their playout times.
    • Delay can be adaptive.
Protocols Used

• Two widely used protocols:
  – Session Initiation Protocol (SIP)
  – ITU standard H.323
Session Initiation Protocol (SIP)
Basic Idea

• SIP is an application layer protocol.
  – Used to establish, manage and terminate multimedia sessions.
  – Various types of sessions are possible:
    • Two-party, multi-party, multi-cast
• SIP can run on either TCP or UDP.
SIP Messages

- Six message types are defined in SIP:
  - INVITE – caller initializes a session
  - ACK – caller confirms after callee answers the call
  - BYE – used to terminate a session
  - OPTIONS – used to know the capabilities of a host
  - CANCEL – an ongoing initialization process can be aborted
  - REGISTER – to make a connection when the callee is not available
Sender / Receiver Addressing

• **SIP is flexible in specifying the address.**
  – Typically uses the IP address, email address, and the telephone number to identify the sender and the receiver.
  – Must be specified in a standard SIP format.
Simple SIP Session

• Three parts:
  – Establishing a session
    • Uses a 3-way handshake protocol.
  – Communication
    • Caller and callee uses two temporary ports for the purpose.
  – Terminating the session
    • Either party can initiate this.
Exchange of voice packets

INVITE

OK

ACK

BYE
The H.323 Standard
Basic Idea

• A standard that allows telephones on the public network to talk to computers on the Internet.

• Uses a gateway:
  – Connects the telephone network to the Internet.
  – Translates messages from one protocol stack to another.
Various Protocols Used

- **H.323 uses a number of protocols:**
  - **G.71 or G723.1**
    - Used for compression.
  - **H.245**
    - Allows parties to negotiate the compression method.
  - **Q.931**
    - For establishment and termination of connections.
  - **H.225**
    - Used for registration with the gatekeeper.
Typical Operation

- **Step 1:** Host sends a broadcast message; gatekeeper responds with its IP address.
- **Step 2:** Using H.225, the host and gatekeeper negotiate bandwidth required.
- **Step 3:** Host, gatekeeper, gateway, and telephone communicate using Q.931 for connection setup.
- **Step 4:** All the four use H.245 to negotiate the compression method to be used.
- **Step 5:** The host and the telephone exchange audio through the gateway using RTP & RTCP protocols.
- **Step 6:** All the four use Q.931 to terminate the connection.
Real Time Protocol (RTP)
Introduction

- Real-time Transport Protocol (RTP) is used to handle real-time traffic over the Internet.
- RTP does not have an inherent mechanism to deliver packets.
  - Uses UDP for the purpose.
  - RTP basically performs sequencing, time-stamping, mixing, etc. for real-time traffic requirements.
Transport Layer

IP

UDP

RTP
• **Typical multimedia sessions:**
  – Relay on Real-time Transport Protocol (RTP) for transmitting data.
  – Relay on Real-time Transport Control Protocol (RTCP) for transmitting control information.
Some Problems

• A typical complex session today:
  – Number of entities involved in a multimedia session is large.
  – In asymmetric heterogeneous broadcast environments the RTCP protocol becomes ineffective (e.g. satellite networks).

• So we must:
  – Extend RTCP to address scalability, and its inability to operate effectively in asymmetric broadcast environments.
RTP/RTCP: Introduction

- **Real-time Transport Protocol (RTP):**
  - This is an Internet standard for sending real-time data over the network.
    - Examples: Internet telephony, interactive audio/video.
  - It consists of a data and control (RTCP) component that work together.
    - **Data:**
      - Provides support for streaming data.
      - Timing reconstruction, loss detection, etc.
– Real-time Transport Control Protocol (RTCP):
  • This is the control part of RTP, and provides the following functions:
    Data delivery monitoring
    Source identification
    Allow session member to calculate the rate to send status messages
• **Which port numbers do they use?**
  – RTP or RTCP are not assigned any well-known port number.
  – The port numbers are assigned on demand.
  – **Restriction:**
    • For RTP, port number must be even.
    • For RTCP, port number must be odd.
RTP Packet Header

V, P, X, CC, M, PT  Sequence number

Timestamp

Synchronization source (SSRC) identifier

Contributing source (SSRC) identifiers

32 bits

V: version,  P: padding,  X: extension,  CC: CSRC count,  M: maker,  PT: payload type
RTCP Status Messages

• Typical information sent:
  – Time Stamps:
    • Used to correlate time stamp of a given session to wall-clock time.
    • Can be used to make rough estimate of round-trip propagation time between receivers.
  – Fraction of packets lost.
  – Total number of packets sent.
  – Sender ID of the Status message.
Design Choice: TCP or RTP

• Typical requirement of multimedia applications:
  – Constant data rate is more important, rather than having a guarantee of receiving all packets, and that too in order.
  – Examples: streaming audio, video.

• TCP:
  – Good for applications that need guarantee on delivery and delivery order.
    • The resend protocol of TCP can cause unacceptable delays in real-time data streaming applications.
• **RTP:**
  – Specifically addresses this issue.
  – The protocol is designed to focus on:
    • Supplying applications with constant data rate.
    • Giving applications feedback on the quality of a link (can help adapt to changing link conditions).
RTP: Some Assumptions Made

• **Assumption 1**
  – The entities are fully connected to each other. This allows a feedback path for control/status information between them.
    • Entities broadcast its control/status information to all other entities.
    • To limit the total amount of control traffic, the amount of network bandwidth allocated for control information is controlled.

• **Assumption 2**
  – All entities are considered to be equal.
    • Constant rate for all entities to receive control information.
    • No variation among entities is assumed.
How to Broadcast Control Info?

- Consider an asymmetric/ unicast environment like satellite networks:
  - A single entity needs to broadcast to all other entities.
  - These receiver entities are usually not directly connected to each other.
    - Via the satellite.
  - Therefore there is no back channel for control/status information to be sent to all entities at once.
  - Can use the satellite for broadcast.
    - A signal repeater in the sky.
• Basically …
  – Instead of having an entity broadcast to all other entities.
    • Individual entity sends control/status information to the source (i.e. satellite).
    • Source (satellite) broadcast information to all entity receivers (Reflection).
Satellite

E1 sends to satellite
Satellite broadcasts

Satellite

E1  E2  E3  En
Summarization

• An alternative to blind broadcast over unicast channel.
  – Some of the information sent in control status messages can be only important to the Source.
  – Summarization:
    • Source gathers report packets from many receiver entities.
    • Source processes this data and broadcasts a summary report which is of a much smaller size than pure Reflection.