Real-Time Services and Multimedia

Outline

- Application requirements
- Interactive multimedia
- Streaming multimedia
Real time application

- **Example application (audio)**
  - sample voice once every 125us
  - each sample has a playback time
  - packets experience variable delay in network
  - playback time = point in time at which data is needed in the receiving host
    - data arriving after playback time is useless
  - playback point = constant offset added to playback time
    - delays vary in time, each packet can come with different delay
    - absorb variations by using a playback buffer
    - ok, as long as playback buffer does not drain

---

**Playback buffer**

- **Rate adaptation**
  - sender can adjust coding rate depending on level of congestion
- **Packet arrival**
- **Packet generation**
- **Network delay**
- **Buffer delay**
- **Delay adaptation**
  - receiver can adjust buffering delay depending on delay variation
Example distribution of delays

- One way delay of a path across Internet measured over a whole day
- If playback buffer delay (play back point) = 100 ms ⇒ 3% packets are lost
- Play back point of almost 200 ms needed to have all data arrive on time

Application performance issues

- Issues affecting real-time application performance
  - large delay prohibits conversation
  - variations in delay
    - can be smoothened by using buffers, but overall delay increases
    - resource reservation may help (note problems with reservations)
  - echo
    - needs to be removed when large delays
  - lost packets
    - replace by silence, extrapolation, or previous data
  - observing silence
    - can compress data more
  - address conversion: phone number ⇔ IP address
  - need control messages end-to-end
  - security
    - fire wall may add difficulties
Interactive vs. streaming

- Requirement: deliver data in “timely” manner
  - reliability: 100% reliability not always required

- Interactive multimedia: short end-end delay
  - e.g., IP telephony, tele/videoconferencing, Internet games, virtual worlds
  - excessive delay impairs human interaction
  - end-end delay requirements:
    - video: < 150 msec acceptable
    - audio: < 150 msec good, < 400 msec OK
    - includes application-level (packetization) and network delays

- Streaming (non-interactive) multimedia: sensitive to delay variation
  - data must arrive in time for “smooth” playout (e.g., RealPlayer, WindowsMediaPlayer)
  - late arriving data introduces gaps in rendered audio/video
    - media stored at source
    - transmitted to client
    - streaming: client playout begins before all data has arrived
    - timing constraint for still-to-be transmitted data: in time for playout

Different protocols

- For streaming multimedia, need protocols for
  - data transport: RTP, application specific transport over UDP (or TCP => overkill, unnecessary retransmits)
  - stream control: RTSP

- For interactive multimedia, need protocols for
  - data transport: RTP, application specific over UDP
  - connection control: RTCP
  - session and call control: H.323 (SDP, SIP,...)
Protocol roadmap for Internet multimedia

Outline

- Application requirements
- Interactive multimedia
- Streaming multimedia

Real-time Transport Protocol (RTP)

- RTP contains functionality that is specific to multimedia
  - development of RTP based on earlier work on “vat” conference application for MBone (Multicast backbone)
  - much of work on multimedia protocols for Internet based on MBone development

- Runs on top of transport-layer protocol (usually UDP)
  - called transport protocol as provides common end-to-end functions (difficulty in fitting to the strict protocol layer model)
  - pays attention to interaction between different applications (e.g., synchronization of audio and video streams)

RTP properties

- RTP allows similar applications to interoperate with each other
  - possible to communicate/negotiate encoding and compressing schemes
- Timing mechanism
  - RTP provides time stamping to enable the recipient of data stream to determine the timing relationship in data ⇒ data can be played back correctly
- Synchronization
  - RTP supports synchronization of multiple media in a conference
- Congestion control
  - RTP gives indications of packet losses (some multimedia applications can adjust to congestion, for example change coding)
  - up to application to react to these
- Frame boundary indication
  - frame boundaries can signify, e.g., talk/silence periods, application can use these to its advantage
- “User friendly” identification of users (user@domain.com)
- Supports multicast
RTP and RTCP

• RTP DOES NOT
  – reserve resources
  – guarantee QoS
  – guarantee timely delivery
  – guarantee reliable delivery (if needed)
  – relies on lower-layer services to do so

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
</tr>
<tr>
<td>UDP</td>
</tr>
<tr>
<td>IP</td>
</tr>
<tr>
<td>Subnet</td>
</tr>
</tbody>
</table>

• RTP standard has too parts
  – RTP to exchange multimedia data (packet format)
    • fields for frame indications, sequence numbers, timestamps …
  – RTP Control Protocol (RTCP) to monitor QoS and convey information about the participants in an ongoing session

• RTP and RTCP do not address session control

RTP Control Protocol - RTCP

• Based on periodic transmission of control packets to all participants in the session, using the same distribution mechanism as the data packets

• Performs 4 functions
  – provides feedback on the quality of the data distribution
    • related to flow and congestion control functions of other transport protocols
    • rate must be controlled in order to scale up to large number of participants
  – carries CNAME, persistent transport-level identifier for an RTP source
    • keeps track on each participant
  – correlation and synchronization of media streams
    • different streams may have different clocks
  – optional, minimal session control information

• RTCP is an outband control protocol
  – if RTP connection in UDP port N ⇒ RTCP in port N+1
RTCP (cont)

- Different packet types
  - sender reports (active senders report transmission and reception statistics)
  - receiver reports (as above, used by those who don’t send)
  - source descriptions (carry CNAME and other info)
  - application-specific control packets

- If a large session, control traffic may consume a lot of bandwidth
  - RTCP provides a method which tries to limit control traffic to 5%

Session Control and Call Control (H.323)

- Session control: how to make information available for holding a videoconference?
  - need to convey, for example, Time, Encoding, multicast IP address, sending data using RTP over UDP port number 4000
  - need protocols for session and call control
  - ITU-T umbrella standard for session and call control: H.323

- Protocols developed in IETF for this purpose
  - SDP (Session Description Protocol)
  - SAP (Session Announcement Protocol)
  - SIP (Session Initiation Protocol)
  - SCCP (Simple Conference Control Protocol)

- Compare:
  - want to announce a conference session (use SDP or SAP)
    • need to send information to a well-known address
  - make an internet phone call (use SDP and SIP)
    • locate users, announce desire to talk, negotiate encoding etc.
**Session Description Protocol (SDP)**

- For describing multimedia sessions to do session announcements, session invitations, other session initiations
- General purpose: does not support negotiation of session content or media encodings
- Contains: name and purpose of session, session time, media and address, required bandwidth, connection information
- Usage:
  - VoIP phone calls
  - multicast session description (session content)
    - SAP used for multicast announcements

**Session Initiation Protocol (SIP)**

- Application-layer protocol that can establish, modify and terminate multimedia sessions (conferences) or Internet telephony calls
  - also media encodings are negotiated
  - Request-response protocol, requests sent by clients and received by servers
  - Can invite participants to unicast or multicast sessions, media and participants can be added to existing sessions
  - Supports name mapping and redirection services, enable personal mobility (personal number)
  - ASCII protocol, transport on top of UDP
- Designed as a part of overall IETF multimedia data and control architecture (RSVP, RTP, RTSP, SAP, SDP)
  - functionality or operation does not depend on any these protocols
  - for example, any session description format can be used
  - in practice, SDP is used exclusively (SDP lists media types and the supported encodings)
- SIP does not
  - offer conference control services
  - allocate multicast addresses
  - reserve network resources (RSVP)
SIP methods

- INVITE: invites a user to a conference
- BYE: terminates a connection between two users in a conference
- OPTIONS: requests information about capability
- ACK: used for reliable message exchange
- REGISTER: conveys location info to a SIP server

VoIP example

1-8: session establishment via SIP
9: data transfer over RTP

VoIP example (cont)

- SIP INVITE message with embedded SDP session presentation
  - Alice calls Bob
  - Bob accepts audio but not video

INVITE sip:bob@macrosoft.com SIP/2.0
From: sip:alice@wonderland.com
To: sip:bob@macrosoft.com
Call-ID: 31415@wonderland.com
CSeq: 42 INVITE
Content-Type: application/sdp
v=0
o=user1 536 2337 IN IP4 h3.wonderland.com
c=IN IP4 h3.wonderland.com
m=audio 3456 RTP/AVP 0 1
m=video 4000 RTP/AVP 38 39

SIP/2.0 200 OK
From: sip:alice@wonderland.com
To: sip:bob@macrosoft.com
Call-ID: 31415@wonderland.com
CSeq: 42 INVITE
Content-Type: application/sdp
v=0
o=user1 535 687637 IN IP4 m.macrosoft.com
c=IN IP4 m.macrosoft.com
m=audio 1200 RTP/AVP 1
m=video 0 RTP/AVP

Outline

- Application requirements
- Interactive multimedia
- Streaming multimedia
Internet multimedia: simplest approach

- Audio or video stored in file
- Files transferred as HTTP object
  - received in entirety at client
  - then passed to player
- Audio, video not streamed:
  - no, “pipelining,” long delays until playout!

Internet multimedia: streaming approach

- Browser GETs metafile
  - metafile tells type of media (application) and URL of the actual media file
- Browser launches player, passing metafile
- Player contacts server
- Server streams audio/video to player
  - streaming by using HTTP over TCP => can not send stream control commands to server (rewind, stop, pause, …)
Streaming from a streaming server

This architecture allows for non-HTTP protocol between server and media player
- can have a richer control protocol for media streams
- Can also use UDP instead of TCP for data transport

Streaming multimedia: UDP or TCP?

- **UDP**
  - server sends at rate appropriate for client
    - with no regard to network congestion
  - short playout delay (2-5 seconds) to compensate for network delay jitter
  - error recover: time permitting

- **TCP**
  - send at maximum possible rate under TCP
  - congestion loss: retransmission, rate reductions
    - depending on playout delay, retransmissions may even be “too late” and thus unnecessary
  - needs larger playout delay to eliminate effects of sending rate fluctuations
**Streaming multimedia: client rates**

- Question: How to handle different client receive rate capabilities?
  - 28.8 Kbps dialup
  - 100Mbps Ethernet
- Answer: server stores, transmits multiple copies of video, encoded at different rates

**User control of streaming multimedia**

- Real Time Streaming Protocol (RTSP): RFC 2326
  - user control: rewind, FF, pause, resume, etc...
  - out-of-band protocol:
    - one port (544) for control msgs
    - one port for media stream
  - TCP or UDP for control msg connection
- Scenario:
  - metafile communicated to web browser
  - browser launches player
  - player sets up an RTSP control connection, data connection to server
RTSP operation

HTTP GET
presentation desc.
SETUP
PLAY
media stream
PAUSE
TEARDOWN

Web browser
Web server
Web server
media stream
media server

RTSP
media player
UDP/TCP

client
server