Multimedia protocols
INF 3190
Michael Welzl
Multimedia

• Combined use of more content forms: text, graphics, audio, video, ...
  – Networks context: multimedia usually means that audio and/or video are used

• Only real-time multimedia of interest
  – Downloading a movie is not much different from downloading a large piece of software (but, note: it’s large)
  – Here, “Real-time” means soft real-time

• Requirements differ:
  – one-way streaming media: compensate network fluctuations by buffering; buffer size → initial delay + time lag (can be bad for live TV broadcasts...)
  – interactive application: buffer size → delay during usage
  – Often, timely is more important than reliable delivery → avoid retransmissions
Characterizing multimedia streams

No constraints, like file transfer

"adaptive"

Hard real-time constraints

Soft real-time constraints
Quality of Service (QoS)

• How to support multimedia bandwidth / delay requirements:
  – use special network mechanisms that can do it (QoS)
  – or dimension the network accordingly

• Both approaches cost money
  – Dimensioning: usually less. It’s also less risky...
  – Internet QoS was once a big thing (because of notion: “value-added services” = more money), but is now a history lesson
  – So we end it here 😊 and assume a non-QoS-Internet from now on
  – Note: perfectly dimensioned networks are also not assumed: not very interesting (and not always possible – e.g. WiFi)
    • Remember, multimedia content is large; there is never a “good enough”
Transmission modes

1 Sender
2 Receivers

Unicast

Broadcast

Overlay Multicast

IP Multicast
Multicast issues

- Required for applications with multiple receivers only
  - video conferences, real-time stream transmission (e.g. radio, TV), ..

- Issues:
  - group management
    - protocol required to dynamically join / leave group: Internet Group Management Protocol (IGMP)
    - state in routers: hard / soft (lost unless refreshed)?
    - who initiates / controls group membership?
  - congestion control
    - scalability (ACK implosion), dealing with receiver heterogeneity, fairness

- Multicast congestion control mechanism classification:
  - sender- vs. receiver-based, single-rate vs. multi-rate (layered),
  - reliable vs. unreliable, end-to-end vs. network-supported
Multimedia content fluctuates

• This is natural: sometimes we talk, sometimes we don’t, sometimes we move, sometimes we don’t.
  – exploited by compression schemes
  – Necessary to cope with size of multimedia content

• Typical values:
  Uncompressed
  • video: 140 – 216 Mbit/s; audio (CD): 1.4 Mbit/s; speech: 64 Kbit/s
  – Compressed audio & video:
    • VOD: down to 1.2 – 4 Mbit/s; Conf.: down to 128 Kbit/s
  – Compressed speech: down to 6.2 Kbit/s
Example: MPEG-1

- International Standard: Moving Pictures Expert Group
  - Compression of audio and video for playback (1.5 Mbit/s), real-time decoding
- Sequence of I-, P-, and B-Frames

I-Frames “intra-coded”
B-Frames bi-directionally coded
P-Frames predictive coded
Matching stream and network rates

- Works if lucky, and buffer large enough
- Large buffer $\Leftrightarrow$ interactivity
Matching stream and network rates /2

- Ideal case
- Realistic?
Matching stream and network rates /3

- "Adaptive Multimedia Application"
- Smoother network bandwidth would facilitate matching
Adaptive multimedia: the user experience

- Studied by several research groups
  - Automatically evaluate "user experience" by judging received content based on knowledge about users
  - Study heartbeat etc. of users who test adaptive multimedia; surveys

- Consistent result: users do not like fluctuations

<table>
<thead>
<tr>
<th>RAP (TCP-&quot;friendly&quot;)</th>
<th>5 different BG traffic levels</th>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>3 short movies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smooth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reactive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intro</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- \( \alpha = 1, \beta = 0.5 \)
- \( \alpha = 0.31, \beta = 0.875 \)
- \( \alpha = 1.3125, \beta = 0.125 \)
Resulting transport layer problem

- How to be fair towards TCP (“TCP-friendly”) and have a relatively stable (“smooth”) rate
  - Several ways to do this
  - Well known example: TCP-Friendly Rate Control (TFRC)
  - Determines sending rate by calculating how much TCP would send under similar conditions
  - Note: TFRC is not a protocol (only a congestion control mechanism)

\[ T = \frac{s}{R \sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1 + 32p^2)} \]

- \( s \): packet size
- \( R \): rtt
- \( t_{RTO} \): TCP retransmit timeout
- \( p \): steady-state loss event rate
Datagram Congestion Control Protocol (DCCP)

• Motivation: provide unreliable, timely delivery
  – e.g. VoIP: significant delay = 😞, but some noise = 😊
  – UDP: no congestion control
    • unresponsive applications endanger others (congestion collapse) and may hinder themselves (queuing delay, loss, ..)

• DCCP realizes congestion control in the OS, where it belongs
DCCP /2

• Roughly:
  – DCCP = TCP – (bytestream semantics, reliability)
    = UDP + (congestion control w/ ECN, handshakes, ACKs)

• Main specification does not contain congestion control mechanisms
  – CCID definitions (e.g. TCP-like, TFRC, TFRC for VoIP)

• IETF standard – but not used much (up to now ?)
One-way streaming over TCP

- Assumption: buffering (delay) doesn’t matter ⇒ no need for a smooth rate!
- Little loss case: TCP retransmissions won’t hurt
- Heavy loss case:
  - DCCP: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10...
  - TCP: (assume window = 3): 1, 2, 3, 2, 3, 4, 3, 4, 5, 4...
    - Application would detect: 4 out of 10 expected packets arrived ⇒ should reduce rate
    - Is receiving 1, 4, 7, 10 instead of 1, 2, 3, 4 really such a big benefit? Or is it just a matter of properly reacting?
In RealPlayer and MediaPlayer, TCP can be used for streaming... seems to work well (also in YouTube!)
Real-time Transport Protocol (RTP)

• Designed for requirements of (soft!) real-time data transport
  – NOT a transport protocol
  – Two Components: RTP and RTP Control Protocol (RTCP)

• Provides several important functions
  – sequencing and loss detection (sequence numbers)
  – synchronization (timestamps)
  – payload identification (RTP profiles)
  – (via RTCP) QoS feedback and session information
  – scalable multicast support (…)
  – mixers and translators to adapt to bandwidth limitations
  – support for changing codecs on the fly, encryption
RTP Packet Format

• Relatively long header (>40 bytes)
  – overhead carrying possibly small payload
  – header compression
  – other means to reduce bandwidth (e.g. silence suppression)

• Header extensions for payload specific fields possible
  – Specific codecs
  – Error recovery mechanisms

• RTP can be used over any transport protocol – usually UDP
Profiles and Payload Types

• Profiles define codecs used to encode the payload data and their mapping to payload format codes ("Payload Type" header field)

• Each profile is accompanied by several payload format specifications
  – e.g. audio: G.711, G.723, G.726, G.729, GSM, QCELP, MP3, DTMF etc., and video: H.261, H.263, H.264, MPEG

• A complete specification of RTP for a particular application usage requires a profile and/or payload format specification(s)
Example profiles

• Profile for Audio and video conferences with minimal control defines
  – a set of static payload type assignments
  – mechanism for mapping between payload formats
  – and a payload type identifier (in header) using the Session Description Protocol (SDP)
    • mapping can be dynamic, i.e. per-session

• Secure Real-time Transport Protocol (SRTP) = profile that provides cryptographic services for the transfer of payload
RTP Control Protocol (RTCP)

- Monitoring
  - of QoS / application performance

- Feedback to members of a group about delivery quality, loss, etc.
  - Sources may adjust data rate
  - Receivers can determine if QoS problems are local or network-wide

- Loose session control
  - Convey information about participants and session relationships

- Automatic adjustment to overhead
  - report frequency based on RTP sending rate and participant count
RTCP Sender / Receiver Reports

• Sender report
  – Sender Information
    • Timestamps, Packet Count, Byte Count
  – List of statistics per source

• Receiver report
  – For each source
    • Loss statistics
    • Inter-arrival jitter
    • Timestamp of last SR
    • Delay between reception of last SR and sending of RR

• Analysis of reports
  – Cumulative counts for short and long time measurements
  – NTP timestamp for encoding- and profile independent monitoring
RTP Quality Adaptation

- Component interoperations for control of quality
- Evaluation of sender and receiver reports
- Modification of encoding schemes and parameters
- Adaptation of transmission rates
- Hook for possible retransmissions (outside RTP)
**RTP Mixer**

- Reconstructs constant spacing generated by sender
- Translates audio encoding to a lower-bandwidth
- Mixes reconstructed audio streams into a single stream
- Resynchronizes incoming audio packets
  - New synchronization source value (SSRC) stored in packet
  - Incoming SSRCs are copied into the contributing synchronization source list (CSRC)
- Forwards the mixed packet stream
- Useful in conference bridges
RTP Translator

- Translation between protocols
  - e.g., between IP and ST-2
  - Two types of translators are installed
- Translation between encoding of data
  - e.g. for reduction of bandwidth without adapting sources
- No resynchronization in translators
  - SSRC and CSRC remain unchanged
Signaling Protocols

• Control of media delivery by sender or receiver
  – Sender and receiver “meet” before media delivery

• Signaling should reflect different needs
  – Media-on-demand
    • Receiver controlled delivery of content; explicit session setup
  – Internet telephony and conferences:
    • Bi-directional data flow, live sources; (mostly) explicit session setup, mostly persons at both ends
  – Internet broadcast
    • Sender announces multicast stream; no explicit session setup
Real-Time Streaming Protocol (RTSP)

• Internet media-on-demand
  – Select and playback streaming media from server
  – Similar to VCR (start, stop, pause, ..), but
    • Potentially new functionality
    • Integration with Web
    • Security
    • Varying quality

• RTSP is also usable for
  – Near video-on-demand (multicast)
  – Live broadcasts (multicast, restricted control functionality)
  – ...

RTSP Approach

• In line with established Internet protocols
  – Similar to HTTP 1.1 in style
  – Range definitions
  – Proxy usage
  – Expiration dates for RTSP DESCRIBE responses
  – Other referenced protocols from Internet (RTP, SDP)

• Functional differences from HTTP
  – Data transfer is separate from RTSP connection; typically via RTP
  – Server maintains state – setup and teardown messages
  – Server as well as clients can send requests
RTSP Features

• Rough synchronization
  – Media description in DESCRIBE response
  – Timing description in SETUP response
  – Fine-grained through RTP sender reports

• Aggregate and separate control of streams possible

• Virtual presentations: synchronized streams from multiple servers
  – Server controls timing for aggregate sessions
  – RTSP Server may control several data (RTP) servers

• Load balancing through redirect at connect time
  – Use REDIRECT at connect time

• Caching
  – Only RTSP caching so far
## RTSP Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIONS</td>
<td>C → S</td>
<td>determine capabilities of server/client</td>
</tr>
<tr>
<td></td>
<td>C ← S</td>
<td></td>
</tr>
<tr>
<td>DESCRIBE</td>
<td>C → S</td>
<td>get description of media stream</td>
</tr>
<tr>
<td>ANNOUNCE</td>
<td>C ← S</td>
<td>announce new session description</td>
</tr>
<tr>
<td>SETUP</td>
<td>C → S</td>
<td>create media session</td>
</tr>
<tr>
<td>RECORD</td>
<td>C → S</td>
<td>start media recording</td>
</tr>
<tr>
<td>PLAY</td>
<td>C → S</td>
<td>start media delivery</td>
</tr>
<tr>
<td>PAUSE</td>
<td>C → S</td>
<td>pause media delivery</td>
</tr>
<tr>
<td>REDIRECT</td>
<td>C ← S</td>
<td>redirection to another server</td>
</tr>
<tr>
<td>TEARDOWN</td>
<td>C → S</td>
<td>immediate teardown</td>
</tr>
<tr>
<td>SET_PARAMETER</td>
<td>C ← S</td>
<td>change server/client parameter</td>
</tr>
<tr>
<td>GET_PARAMETER</td>
<td>C ← S</td>
<td>read server/client parameter</td>
</tr>
</tbody>
</table>
Session Initiation Protocol (SIP)

• Lightweight generic signaling protocol

• Internet telephony and conferencing
  – call: association between number of participants
  – signaling association as signaling state at endpoints (no network resources)

• Several “services” needed
  – Name translation, user location, feature negotiation, call control
SIP Basics

• Establish calls between users
  – directly or forwarding (manual and automatic)
  – re-negotiate call parameters
  – terminate and transfer calls

• Supports personal mobility (change of terminal)
  – through proxies or redirection

• Control, location and media description (via SDP)

• Extensible
  – IMS – Internet Multimedia Subsystem – the next generation of telecoms’ service gateways
SIP – Methods

• Basic Methods:
  – INVITE: session setup – like RTSP SETUP and DESCRIBE in one
  – ACK: like RTSP ACK
  – OPTIONS: like RTSP OPTIONS
  – BYE: end a session
  – CANCEL: terminate an ongoing session setup operation
  – REGISTER: register a user in a location server, update location, ...

• Additional Methods (partially standardized):
  – INFO: carry information between User Agents
  – REFER: ask someone to send an INVITE to another participant
  – SUBSCRIBE: request to be notified of specific event
  – NOTIFY: notification of specific event
SIP Operation – Proxy Mode

- Proxy forwards requests
  - possibly in parallel to several hosts
  - cannot accept or reject call
  - useful to hide location of callee
SIP Operation – Redirect Mode

1. Invite u@domain1

2. Where?

3. domain2

4. Moved temporarily
   Location: user@domain2

5. ACK u@domain1

6. Invite user@domain2

7. “Ring”

8. ACK user@domain2

Site A

User with “symbolic name” calls another

Site B

location server

Redirect Mode

UNIVERSITY OF OSLO
PSTN: SS7 / SIGTRAN

(PSTN = Public Switched Telephone Network)

- SS7: telephony signaling protocols
  - mainly call setup and teardown
  - international standard + national variants
  - services such as call forwarding (busy and no answer), voice mail, call waiting, conference calling, calling name and number display, ...

- SIGTRAN: IETF standards, most importantly SCTP
  - **efficiently** transferring such data over the Internet
SCTP services: SoA TCP + extras

<table>
<thead>
<tr>
<th>Services/Features</th>
<th>SCTP</th>
<th>TCP</th>
<th>UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-duplex data transmission</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Connection-oriented</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Reliable data transfer</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Unreliable data transfer</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Partially reliable data transfer</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Ordered data delivery</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Unordered data delivery</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Flow and Congestion Control</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>ECN support</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Selective acks</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Preservation of message boundaries (ALF)</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>PMTUD</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Application data fragmentation</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Multistreaming</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Multihoming</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Protection against SYN flooding attack</td>
<td>yes</td>
<td>no</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Application Level Framing (ALF)

- Concept applied in RTP and SCTP
  - Byte stream (TCP) inefficient when packets are lost
  - Application may want logical data units ("chunks")

- ALF: app chooses packet size = chunk size
  packet 2 lost: no unnecessary data in packet 1,
  use chunks 3 and 4 before retrans. 2 arrives

- 1 ADU (Application Data Unit) = multiple chunks \ ALF still more efficient!
Unordered delivery & multistreaming

• Decoupling of reliable and ordered delivery
  – Unordered delivery: eliminate Head-Of-Line blocking delay

TCP receiver buffer

| Chunk 2 | Chunk 3 | Chunk 4 | Chunk 1 |

App waits in vain!

• Support for multiple data streams
  (per-stream ordered delivery)
  - Stream sequence number (SSN) preserves order within streams
  - no order preserved between streams
Multihoming

• ...at transport layer! (i.e. transparent for apps, such as FTP)

• TCP connection ⇔ SCTP association
  – 2 IP addresses, 2 port numbers ⇔ 2 sets of IP addresses, 2 port numbers

• Goal: robustness (*not load balancing – yet?)
  – automatically switch hosts upon failure
  – eliminates effect of long routing reconvergence time

• TCP: no “keepalive“ messages when connection idle

• SCTP monitors reachability via ACKs of data chunks and heartbeat chunks
References


• INF3190 2009 slides by Carsten Griwodz