Media Streaming in the Internet

- Introduction to Media Streaming
- Real-time Streaming Protocol (RTSP)
Real-time Media Streaming

Retrieving content from a source where

▷ the content is continuous in nature (e.g. audio, video),

▷ the content is (potentially) presented to the user before it has been downloaded entirely, and

▷ there is no human-to-human interaction involved (i.e. latencies are acceptable to a certain degree),

▷ yet there may be a need for interactive streaming controls (possibly realized in a distributed fashion across sender and receiver)

Contrast: interactive, interpersonal communications
Two Types of Streaming

- **Broadcast streaming (non-interactive)**
  - Sender transmits media stream according to its own schedule
  - Receivers “tune into a media stream” of interested
  - Receivers have no means to influence the transmission
  - Suitable for multicast / broadcast networks

- **Interactive streaming**
  - Sender provides media stream to receivers “on demand”
  - Receivers may start / stop transmission
  - Receivers may invoke further operations
    - Fast forward, search, play offset, …
  - Suitable for P2P sessions or coordinated small groups
Architectural Components

- **Content Description**
  - Describe type of content, format, access methods, ...
  - SDP, SDPng, IMGs, MPEG tables, proprietary formats, ...

- **Content Description Delivery / Access Protocol**
  - Delivers Content Description
  - HTTP, SMTP, NNTP, SAP, proprietary protocols, ...

- **Content Access (= Media Streaming) Protocol**
  - Initiates, controls, and terminates media streams
  - RTSP, proprietary protocols, ...

- **Content Delivery (= Media Transport) Protocol**
  - Carries the actual content
  - RTP/RTCP, HTTP, proprietary protocols, ...
Conceptual Overview

1. Reference to Media Server

2. Content Description

Clients

Announcement Server

Web Server

Media Server

SAP

HTTP

RTSP
Conceptual Overview

Clients

- Announcement Server
- Web Server
- Media Server

- Control sessions
- Media sessions
Variants of Media Streaming

From a service provider

- Via a broadcast network
  - Broadcasting
  - Advanced multicast-based video-on-demand

- Specific support for the last mile
  - TV-over-DSL (and other Internet access links)

- Video-on-Demand
  - Integrated with the web
  - Using dedicated network links

In a private household

- From a server to one or more home devices

Community-based: Peer-to-peer

- Via the Internet between consumers
- Assisting service providers
Real-Time Streaming Protocol (RTSP)

- RFC 2326 ("buggy", "underspecified")
- draft-ietf-mmusic-rfc2326bis-19.txt

Interactive streaming control in the Internet
- Media servers provide media streams to users on demand
- Content described by presentation descriptions

"Network Remote Control" of a media server
- PLAY [and RECORD]
- Numerous options for media control
  - PAUSE, faster / slower playback, selection of ranges from a stream, ...
Protocol Characteristics

- Borrows heavily from HTTP
  - Syntax, quite a bit of semantics, parts of the architecture

- Important differences
  - Servers may issue requests, too!
    - Symmetric communication
  - Servers are stateful
  - Different methods
  - Different headers
    - But many HTTP headers re-used
  - Entities (=request/response bodies) only describe content
  - Content itself (=media) is carried out of band
    - e.g. in RTP; also support for interleaving of media with RTSP connection

- Transport: TCP [or UDP]
  - Reliability handled at the RTSP level
RTSP Components

Content Description

Media Server

Content Base

Video
Audio - 1
Audio - 2

/content
/movies/matrix
/audio/en
/audio/de
/video

rtsp://media-server.tkk.fi/movies/matrix/audio/en
RTSP URIs

- **Schemes:**
  - rtsp: reliable, connection-oriented (TCP)
  - rtspu: potentially unreliable, connectionless (UDP)
  - rtspu: secure, reliable, connection-oriented (TLS)

- **General scheme:**
  - rtsp:// host / local identifier

- **Host**
  - Should be DNS name
  - Support for IPv4; IPv6 now being added

- **Local Identifier**
  - Opaque; may be used for aggregate / non-aggregate control
Time in RTSP

9 SMPTE Timestamps
   - SMPTE = Society of Motion Picture Television Engineers
   - Measured in hours, minutes, seconds, frames, fractions (subframes)
     - 29.97 or 25 frames per second (default: 29.97)

9 Normal Play Time (NPT ≠ NTP)
   - Relative to beginning of stream
   - In seconds: SS.fff 10.74
   - In human readable time: HHH:MM:SS.fff 3:47:09.314159

9 Absolute Time
   - Using ISO 8601 format
   - 20021211T101435.89Z

9 (RTP Media Time)
   - Media-specific clock for the RTP timestamp
   - Synchronized with absolute time via RTCP
RTSP Sessions

- Shared state between RTSP client and server

- Establish by SETUP message

- Removed by TEARDOWN
  - Or due to some timeout

- Independent of underlying TCP connections
  - TCP connections may be closed and re-opened during a single RTSP session

- Typically bound to a single presentation
  - in case of SDP, valid for one SDP session (description)

- May contain several RTP sessions
  - e.g. one per media stream
RTSP Request Message

SETUP rtsp://ms.tkk.fi/movies/matrix RTSP/1.0
CSeq: 302
Date: 10 Dec 2002 15:35:06 GMT
Session: 47112344
Transport: RTP/AVP;unicast;
          client_port=4588-4589

[Optional Message Body]
RTSP Response Message

RTSP/1.0 200 OK
CSeq: 302
Date: 10 Dec 2002 15:35:07 GMT
Server: Matrix-Server 0.4.2
Session: 47112344
Transport: RTP/AVP;unicast;
    client_port=4588-4589;server_port=6256-6257
<CRLF>
[Optional Message Body]
RTSP Protocol Operation: DESCRIBE

- Obtain presentation description from server
  - e.g. SDP
- Media initialization
  - Contains information about all embedded media streams
  - Support for aggregate / non-aggregate control
  - Allows a client to determine suitability of content
  - Choose encoding if possible
- Optional: description may be obtained out-of-band

Client

Server

DESCRIBE

200 OK + SDP
RTSP Protocol Operation: ANNOUNCE

- Updates the presentation description actively from the server
  - e.g. add or remove media streams
- May be issued at any time
RTSP Protocol Operation: SETUP

- Initiate an RTSP session
- Reserve resources at the server
  - Server may redirect to other servers (e.g. if busy)
- Convey transport parameters for media sessions
  - Negotiate transport protocol
  - e.g. RTP/UDP vs. tunneling
  - Enable firewalls to open holes
RTSP Protocol Operation: PLAY

- Start streaming
- Allows to specify a variety of streaming operations
  - Range(s) to play
    - = seek operation
    - E.g. 10-20s; 30-45s; 60s-
  - Forward / backward
  - Speed
    - +3.0
    - - 2.5
RTSP Protocol Operation: PAUSE

- Interrupt streaming
  - But keep resources allocated
- May take effect
  - Immediately or
  - At a specified point in time
- PLAY may be used to resume streaming
RTSP Protocol Operation: TEARDOWN

- Stop streaming
- Terminate RTSP session
  - Free resources
- Takes effect immediately

Client

- DESCRIBE
- 200 OK + SDP
- SETUP + transport
- 200 OK + transport
- PLAY [range]
- 200 OK
- PAUSE [time]
- 200 OK [range]
- TEARDOWN
- 200 OK

Server
RTSP Methods

- OPTIONS
- DESCRIBE, ANNOUNCE
- SETUP, TEARDOWN
- PLAY, PAUSE
- REDIRECT
  - May be used by a server to refer a client to a different location

- GET_PARAMETER
  - Retrieve parameter value specified in the header (in the Session: context)
    - Returned in 200 OK response body as “Name: value” pairs
  - May be used for keep-alive purposes

- SET_PARAMETER
  - Set value of parameter(s) per response body (“Name: value” pairs)

- [RECORD]
  - Record a media stream at a server
  - Underspecified, not really supported, now removed from base spec
RTSP General Header Fields

(For reference only)

- Cache-Control:
- Connection:
- CSeq:
- Date:
- Timestamp:
- Via:
RTSP Request Header Fields

(For reference only)

- **Accept:**
- **Accept-Encoding:**
- **Accept-Language:**
- **Authorization:**
- **Bandwidth:**
- **Blocksize:**
- **From:**
- **If-Modified-Since:**
- **Require:**
- **Proxy-Require:**
- **Supported:**
- **Referer:**
- **Scale:**
- **Speed:**
- **Range:**
- **Session:**
- **Transport:**
- **User-Agent:**
Some Response Status Codes

- 100 Continue
- **200 OK** / 201 Created
- 300 Multiple Choices
- 301 Moved Permanently / 302 Moved Temporarily
- 304 Not Modified
- 305 Use Proxy
- 400 Bad Request
- 401 Unauthorized / 407 Proxy Authentication Required
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed / 406 Not Acceptable / 408 Request Timeout
- 451 Parameter Not Understood
- 454 Session Not Found
- 455 Method not valid in this State / 457 Invalid Range
- 461 Unsupported Transport
- 500 Internal Server Error / 501 Not Implemented / 551 Option not Supported
Response Header Fields

(For reference only)

- Accept-Ranges:
- Proxy-Authenticate: / WWW-Authenticate:
- Public:
- Location:
- Range: / Scale: / Speed:
- Retry-After:
- RTP-Info:
- Transport:
- Unsupported:
- Vary:
- Session:
Entities

- Entities contained in RTSP messages are typically presentation descriptions
  - e.g. an SDP message (Content-Type: application/sdp)
  - Should always fully specify the media stream(s)

- Header fields:
  - Content-Length:, Content-Type:, Content-Encoding:, Content-Base:, Content-Location:, Content-Language:
  - Allow:
  - Last-Modified:, Expires:
Interleaving

- RTSP should use RTP/UDP for media streaming
  - Not always feasible (e.g. firewall, see next slide)

- Interleaving of RTSP and media data
  - Escape binary data ("$")
  - Define multiple “channels”
  - Specify packet length in binary
  - Yields a four byte header:
    - Interleaved with RTSP messages
    - Starts right after previous message
    - Length used to determine how many bytes to skip / pass
RTSP 2.0

- Presently under development (well advanced)
- draft-ietf-mmusic-rfc2326bis-15.txt

- Tons of editorial changes (readability, coherence, …!)

- Better state machine descriptions

- Updated (more coherent) semantics for various header fields
  - Significant alignment with SIP based upon experience gained there

- RECORD disappeared from base spec
  - Was underspecified anyway

- Support for NAT traversal upcoming
  - draft-ietf-mmusic-rtsp-nat-05.txt
Firewall Friendliness

▶ Several means to support RTSP across firewalls
  ● Interleaving support
  ● Transport: header indicates port numbers, IP addresses, …
    Firewall logic does not need to parse SDP format
  ● SOCKS support

▶ Still may be insufficient
  ● Firewalls may block RTSP in the first place
  ● “Last resort”: HTTP tunneling
    Really bad (dubious!)
    Boils down to a competition between firewall vendors and application developers
    Defeats the purpose of a firewall in the first place
  ● Nevertheless: widely deployed (“HTTP streaming”)
    Apple, Microsoft, …
RTSP Real World Implementations

- **Server Implementation:**
  - Apple’s Darwin Media Server
  - Real Network’s Helix DNA Media Server
  - Live555 Media Server
  - VideoLAN
  - Microsoft Streaming Server

- **Client Implementation:**
  - vlc (uses live555 libraries)
  - Mplayer (uses live555 libraries)
  - Real player
  - Windows media player

- Youtube’s mobile version uses RTSP