RTSP'

Henning Schulzrinne
Dept. of Computer Science
Columbia University
schulzrinne@cs.columbia.edu

IETF San Jose, California
MMUSIC WG

December 9, 1996

RTSP Prime (RTSP')

- semantics very similar to RTSP-00
- textual encoding; but: RTSP' would work text or binary
- works with TCP and UDP
- similar to HTTP, but commands in both directions
- re-use of HTTP authentication, encryption, servers, ...
- integrated with SIP/2.0
- allow evolution of session description
- syntax is strawman – don't take too seriously

Media Server Operation

"Traditional" media-on-demand server – live or stored media:

1. RTSP', HTTP: client retrieves session description, including multicast address (if applicable) ➠ `rtsp://movies.com/twister`
2. client picks desirable stream combination, indicates unicast port
3. RTSP': client sets port, speed, ...
4. RTSP': client issues PLAY, ... requests for individual media or session
5. server can announce new media (for live sessions)
6. RTSP': client or server says BYE

Media Server in Conference

1. SIP: conference participant (CP) invites server
   e.g.: play audio into existing multicast session
   session establishment outside scope (e.g., H.323)
   session description in invitation
   server joins group and waits for commands
2. RTSP': GET description (language, content, but not address)
3. RTSP': include conference id parameter
4. RTSP': CP sets parameters, controls playout
5. "passing the remote": social protocol, floor control

Open issue: which stored media stream for which conference session?

Naming

- URL-like name for session and individual components: e.g., host/session/media
- names in session description relative (appended to base URL) or absolute (different media on different servers)
- control each named entity
- dynamically generated session id to distinguish between different instances
- define absolute positioning URL (#)?

`blockbuster.com/twister/audio_en/1797#11:12`

Transport

- TCP connection $\neq$ application association
- one stream, several connections (HTTP/1.0)
- several streams, one connection

RTSP' over UDP

- several commands in one packet
- sequence number (per packet) for reliability
  - fourth parameter on method line (for UDP only?)
  - `Sequence:` header
- worry about late ACKs after restart?
- RTT estimation with retransmission ➠ different seq. #

Data interleaving

- numbered in session description
- trade-off between compactness and parsing
- coordinate all control streams on single TCP

e.g.: `$` *stream length data* (all binary) or IMAP: `{1044}CRLF` *data*

Example: join a broadcast

```
C → S   GET twister RTSP/1.0
        Accept: application/sdp, application/sdf
S → C   RTSP/1.0 200 OK
        Content-Type: application/sd

        (media (id 1345) (m audio 224.2.0.1 3456) (id audio_en))
C → S   PLAY twister/1345/audio_en RTSP/1.0
        Range: smpte=0:10:20-
```

smpte: SMPTE time code = standard for (pro) video editing min:sec:frame

clock: absolute time (e.g., meeting recording)

Example: unicast media server

```
C → S   GET twister RTSP/1.0
S → C   RTSP/1.0 200 OK
        ...
C → S   SET_PARAMETER twister/audio_en
C → S   PLAY twister/audio_en
```

denial-of-service attack ➠ only to address of request

Recording

- choice of name ➠ like `POST`
- name generator command?
- cueing – reasons?

```
C → S   RECORD scratch/AA12345
        Conference: 199612081531.AA@bar.com
```


Updating an on-going session

- server can "push" new media description to client
- differential or replace?
- `SESSION` command

Camera Control

➠ just more commands
difficulty: naming (several cameras attached to host)

```
SET_PARAMETER camera1
Angle: 125
```


Requirements for session description

- names (URLs) for overall session and media, either absolute or relative
- 8-bit stream numbers for muxed streams
- assemble network description
- model: server offers streams, client picks
- need more negotiation? capability description?

Other RTSP Issues

- parameter settings: advantage of second level not clear
- separate issue: who gets to control ➠ floor control or social protocol
- request for QOS report should be in RTCP (see AVT slides)
- retransmission should be in RTCP ➠ already there for H.261!
- difference between PAUSE and STOP? RSVP?

Session Description

- current SDP is compact, but not good for structure
- ➠ re-use much of SDP coding, but add structure
- easy inheritance of parameters

par: play all list elements concurrently (in parallel)

one: pick one the list elements

seq: play list elements sequentially

Parameters: at any level; globally unique names?

SDF

```
(session (v 0)
  (o mhandley 2890844526 2890842807 IN IP4 126.16.64.4)
  (s Sd seminar)
  (i A seminar on the session description protocol)
  (u http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.01.ps)
  (e M.Handley@cs.ucl.ac.uk (Mark Handley))
  (c IN IP4 224.2.17.12/127)
    (t 2873397496 2873404696)(a recvonly)
  (par
    (media (m audio 3456 RTP (
      (one
        ((enc PCMU) (id Foo))
        ((enc DVI)  (id Bar))
     )
    (media (m video 2232 RTP H261) (id Pix))
    (media (m whiteboard 32416 UDP WB)(orient portrait))
  ))
```