The Real-Time Streaming Protocol (RTSP)

RFC 2326

Reading

- RTSP is defined in RFC 2326
  - Read sections 1 through 4, 9, and 14
- Good web site maintained by one of the authors of the protocol*:
  - **http://www.cs.columbia.edu/~hgs/rtsp/rtsp.html**
- Where it all comes together: white paper describing the relationship between IP Multicast, RTP, RTSP, RSVP, etc.:
  - **http://www.ipmulticast.com/community/whitepapers/highprot.html**

* These notes are based in part on a 1997 presentation by H. Schulzrinne.

What is RTSP?

- The Real-Time Streaming Protocol (RTSP) is a control protocol intended for:
  - establishment of one or more synchronized, continuous-media streams
  - control of such streams
- RTSP can be seen as a “network remote control”
- RTSP is not used to deliver the streams
  - use RTP or similar for that

Protocol History

- Co-authored by Columbia University, Real Networks, Netscape and the IETF MMUSIC.
- Submitted to the IETF in 1996 with support from about 40 media companies.
- RFC 2326 came out in 1998.
- Implemented in many products, including RealPlayer.
- Toolkits and free source code available.

Operations Supported by RTSP

- Retrieval of media from a media server
  - both already-established multicast sessions and unicast sessions
- Invitation of a media server to a conference
  - to contribute a stream to the conference
  - to record part or all of the conference
- Addition of media to an existing conference
  - media can become available dynamically

Differences between RTSP and HTTP

- The RTSP design is based on HTTP, with the following differences:
  - new methods; different protocol identifier
  - RTSP servers need to keep state while HTTP servers do not
  - Both RTSP servers and clients can issue requests
  - Data is carried by an external protocol (typically but not necessarily RTP)
  - RTSP uses UTF-8 instead of ISO 8859-1 character set
  - RTSP uses absolute request URIs
  - RTSP defines an extension mechanism

Protocol Properties

- ***Extendable***: new methods can be added in a compatible way
- ***Easy to parse***: all protocol messages are text-based and HTTP/MIME parsers can be re-used
- ***Security***: all the HTTP security mechanisms can be employed, as well as transport and network layer security
- ***Transport independent***: RTSP implements application-layer reliability and can run on top of TCP, UDP, or any other protocol. Standardized ports for RTSP:

```
rtsp         554/tcp    Real Time Streaming Control Protocol
rtsp         554/udp    Real Time Streaming Control Protocol
rtsp-alt     8554/tcp   RTSP Alternate (see port 554)
rtsp-alt     8554/udp   RTSP Alternate (see port 554)
```

Protocol Properties (cont.)

- ***Multi-Server capable***: each media in a stream can reside in a different server; the client establishes multiple concurrent connections to different servers.
- ***Control of recording devices***: the protocol is capable of controlling both playback and recording devices.
- ***Separation of stream control and conference initiation***: the protocol establishes independence between these two functions. H.323 or SIP can be used to invite a server.
- ***Frame-accurate***: the protocol uses SMPTE time stamps to allow for remote digital editing.
- ***Presentation description neutral***: any description format can be used (typically SDP); RTSP will identify the format.

Protocol Properties (cont.)

- ***Proxy and firewall friendly***: the protocol can be easily handled by firewalls (no source IP address dependencies).
- ***HTTP friendly***: the protocol re-uses many of the same concepts and procedures as HTTP, and most of the infrastructure is the same.
- ***Server control***: the client is able to start and stop the stream.
- ***Transport negotiation***: the client can negotiate the transport method prior to streaming.
- ***Capability negotiation***: the server and client can negotiate capabilities prior to start of the session; for example, whether or not seeking into the stream is possible.

Ways to extend RTSP

- Add new parameters to existing methods
  - recipients not implementing these methods should safely ignore them.
- Add new methods
  - recipients can respond with a "not implemented" response to unknown methods; sender must not use them again.
- Define a new protocol, keeping the the format of the version field the same.

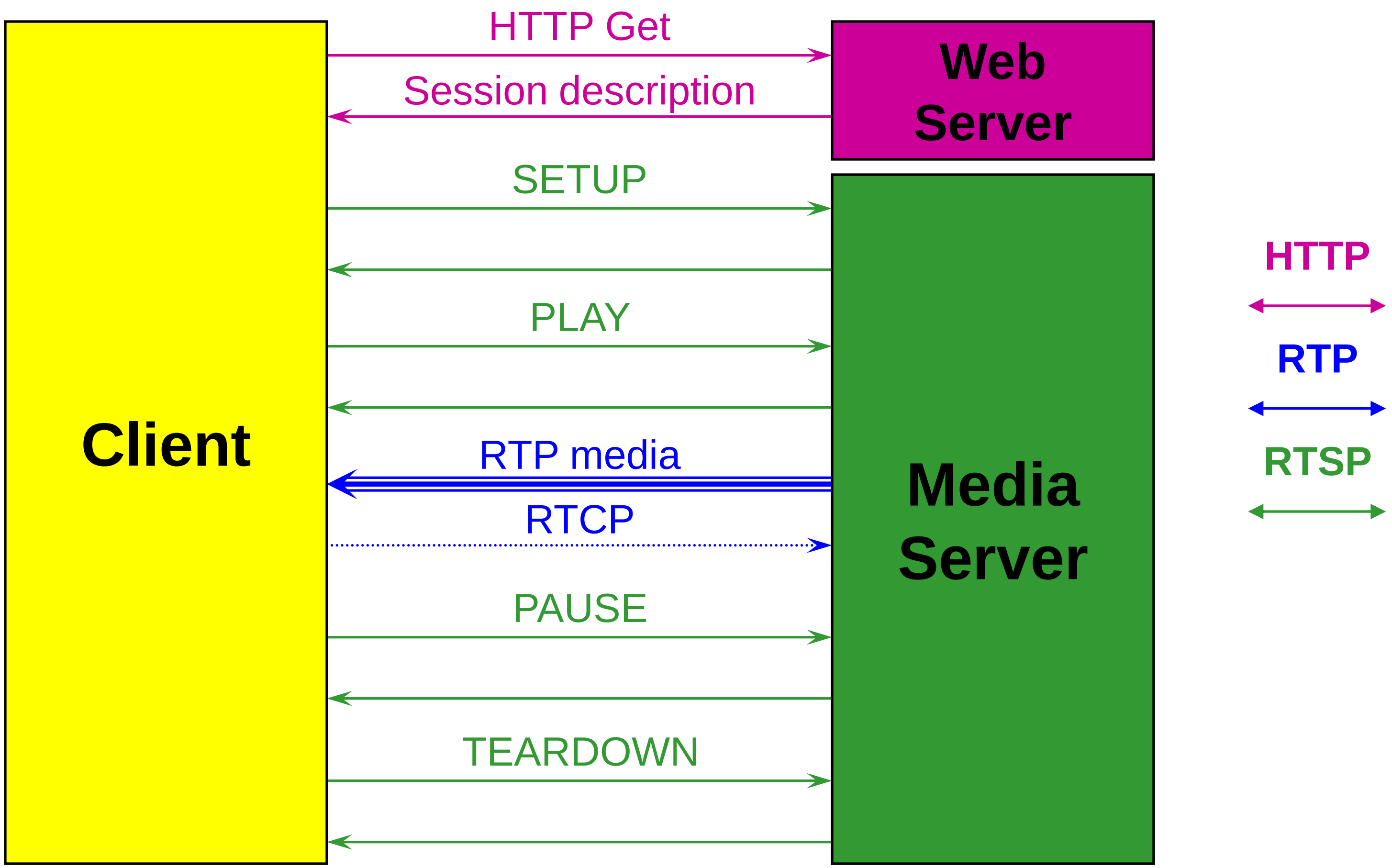
Basic Protocol Operation

- **Step 1**: the client acquires a session description file through some means (HTTP, for example). This session file contains one or more RTSP URLs for the available media.
- **Step 2**: the client contacts the RTSP server(s) and media starts streaming. Addressing can be:
  - Unicast: the media is sent to the source of the RTSP request, on a client-selected port.
  - Multicast, server-selected address: this is the case for live transmission.
  - Multicast, client-selected address: typically the case when the server is invited to a conference.

Primary RTSP Methods

- **SETUP**
  - Server allocates resources and starts an RTSP session
- **RECORD and PLAY**
  - Start the media stream allocated by SETUP
- **PAUSE**
  - Temporarily stops the stream without freeing resources
- **TEARDOWN**
  - Frees the resources and terminates the session

RTSP Example



The RTSP URL

- RTSP defines two standard URLs:
  - **`rtsp://`** for TCP-based transport
  - **`rtspu://`** for UDP-based transport
- URLs can refer to the presentation as a whole or parts of it:
  - Whole stream:
    - **`rtsp://stream.stanford.edu:554/ee384b`**
  - Just the sound:
    - **`rtsp://stream.stanford.edu:554/ee384b/sound`**
- No filesystem is assumed; paths are opaque to the client. Notation only indicates media hierarchy.

Transport Details

- If RTSP is transported over TCP, it makes use of the TCP reliability features.
- RTSP servers must support TCP; support for UDP is optional.
- If RTSP is transported over UDP, unacknowledged requests are retransmitted after a RTT (defaults to 500 ms).
- Multiple RTSP requests may be packed into a single PDU.

RTSP Time

- Normal Play Time (NPT)
  - Time, in seconds (floating point), since the beginning of the stream.
  - Special constant “now” defined for live streams.
- Absolute Time (GMT)
  - Specified to a fraction of second
  - Used to synchronize playback between servers
- SMPTE Relative Timestamps
  - Industry standard for time codes
  - Relative to the beginning of the clip
  - Expressed in hours:minutes:seconds:frames:subframes

Detailed Example: Unicast Session

Client C wants a movie whose description is stored in web server W, with the video track in server V and audio track in server A. Client C wants to skip the first 10 minutes of the movie.

Client Requests Description from Web Server

```
C→W: GET /twister.sdp HTTP/1.1
     Host: www.example.com
     Accept: application/sdp

W→C: HTTP/1.0 200 OK
     Content-Type: application/sdp

     v=0
     o=- 2890844526 2890842807 IN IP4 192.16.24.202
     s=RTSP Session
     m=audio 0 RTP/AVP 0
     a=control:rtsp://audio.example.com/twister/audio.en
     m=video 0 RTP/AVP 31
     a=control:rtsp://video.example.com/twister/video
```

Client Sets Up Audio Session

```
C→A: SETUP rtsp://audio.example.com/twister/audio.en
     RTSP/1.0
     CSeq: 1
     Transport: RTP/AVP/UDP;
            unicast;
            client_port=3056-3057


A→C: RTSP/1.0 200 OK
     CSeq: 1
     Session: 12345678
     Transport: RTP/AVP/UDP;
            unicast;
            client_port=3056-3057;
            server_port=5000-5001
```

Client Sets Up Video Session

```
C→V: SETUP rtsp://video.example.com/twister/video RTSP/1.0
     CSeq: 1
     Transport: RTP/AVP/UDP;
            unicast;
            client_port=3058-3059


V→C: RTSP/1.0 200 OK
     CSeq: 1
     Session: 23456789
     Transport: RTP/AVP/UDP;
            unicast;
            client_port=3058-3059;
            server_port=5002-5003
```

Client Starts Video Playback

```
C→V: PLAY rtsp://video.example.com/twister/video RTSP/1.0
     CSeq: 2
     Session: 23456789
     Range: smpte=0:10:00-

V→C: RTSP/1.0 200 OK
     CSeq: 2
     Session: 23456789
     Range: smpte=0:10:00-0:20:00
     RTP-Info: url=rtsp://video.example.com/twister/video;
           seq=12312232;
           rtptime=78712811
```

Client Starts Audio Playback

```
C→A: PLAY rtsp://audio.example.com/twister/audio.en
     RTSP/1.0
     CSeq: 2
     Session: 12345678
     Range: smpte=0:10:00-

A→C: RTSP/1.0 200 OK
     CSeq: 2
     Session: 12345678
     Range: smpte=0:10:00-0:20:00
     RTP-Info:
            url=rtsp://audio.example.com/twister/audio.en;
            seq=876655;
            rtptime=1032181
```

Client is Finished

```
C→A: TEARDOWN rtsp://audio.example.com/twister/audio.en
     RTSP/1.0
     CSeq: 3
     Session: 12345678

A→C: RTSP/1.0 200 OK
     CSeq: 3

C→V: TEARDOWN rtsp://video.example.com/twister/video
     RTSP/1.0
     CSeq: 3
     Session: 23456789

V→C: RTSP/1.0 200 OK
     CSeq: 3
```

Other Functions

- REDIRECT:
  - Redirects the client to another URL
  - Requires a TEARDOWN and SETUP
  - Used for load balancing
- Caching:
  - Caching of content, not RTSP responses
  - Live content can "cut-through"
- PLAY:
  - VCR control capabilities: rewind, fast-forward