

User Control of Streaming Media: RTSP

HTTP

- Does not target multimedia content
- No commands for fast forward, etc.

RTSP: RFC 2326

- Client-server application layer protocol.
- For user to control display: rewind, fast forward, pause, resume, repositioning, etc...

RFC 7826

- 2016 RTSP 2.0

What it doesn't do:

- does not define how audio/video is encapsulated for streaming over network
- does not restrict how streamed media is transported; it can be transported over UDP or TCP
- does not specify how the media player buffers audio/video

RTSP: out of band control

FTP uses an “out-of-band” control channel:

- A file is transferred over one TCP connection.
- Control information (directory changes, file deletion, file renaming, etc.) is sent over a separate TCP connection.
- The “out-of-band” and “in-band” channels use different port numbers.

RTSP messages are also sent out-of-band:

- RTSP control messages use different port numbers than the media stream: out-of-band.
 - Port 554
- The media stream is considered “in-band”.

RTSP Example

Scenario:

- metafile communicated to web browser
- browser launches player
- player sets up an RTSP control connection, data connection to streaming server

Metafile Example

```
<title>Twister</title>
```

```
<session>
```

```
  <group language=en lipsync>
```

```
    <switch>
```

```
      <track type=audio
```

```
        e="PCMU/8000/1"
```

```
        src = "rtsp://audio.example.com/twister/audio.en/lofi">
```

```
      <track type=audio
```

```
        e="DVI4/16000/2" pt="90 DVI4/8000/1"
```

```
        src="rtsp://audio.example.com/twister/audio.en/hifi">
```

```
    </switch>
```

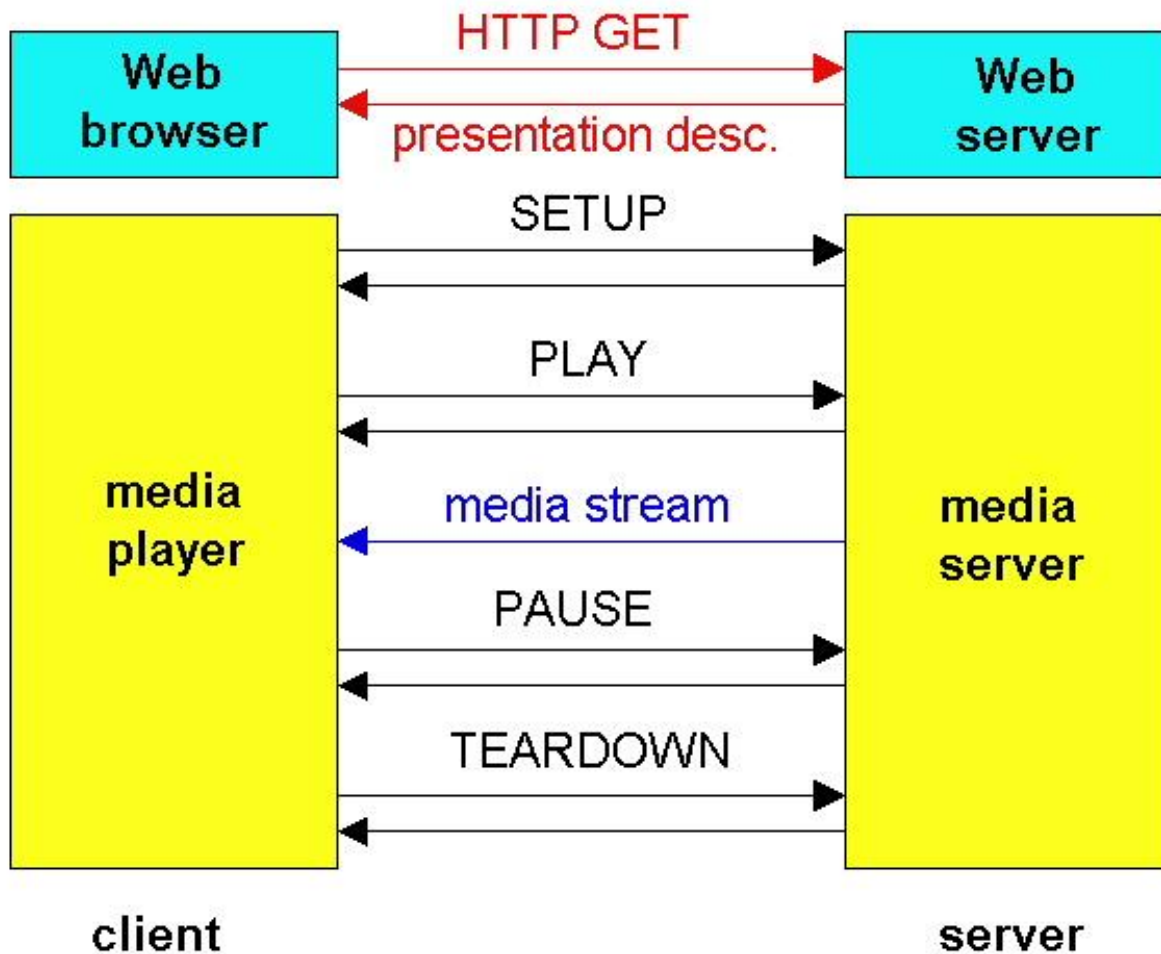
```
  <track type="video/jpeg"
```

```
    src="rtsp://video.example.com/twister/video">
```

```
</group>
```

```
</session>
```

RTSP Operation



RTSP Exchange Example

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231
Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Session: 4231

S: 200 3 OK

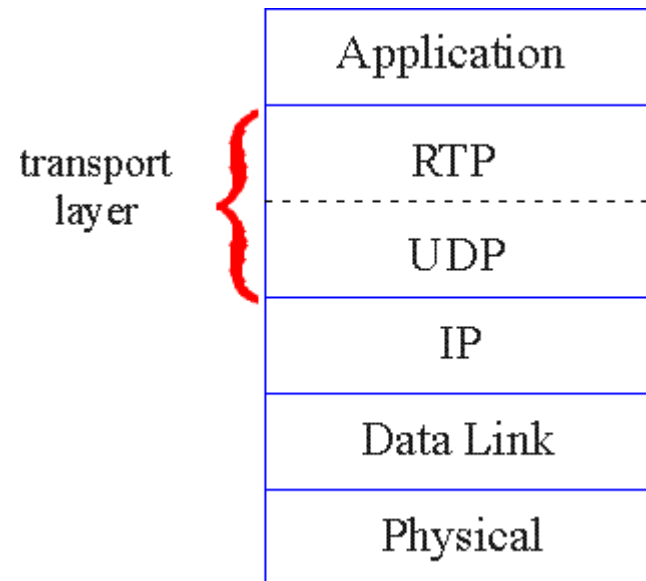
Real-Time Protocol (RTP)

- RTP specifies a packet structure for packets carrying audio and video data
- RFC 1889 (RFC 3550)
- RTP packet provides
 - payload type identification
 - packet sequence numbering
 - timestamping
- RTP runs in the end systems.
- RTP packets are encapsulated in UDP segments
- Interoperability: If two Internet phone applications run RTP, then they may be able to work together

RTP runs on top of UDP

RTP libraries provide a transport-layer interface that extend UDP:

- port numbers, IP addresses
- payload type identification
- packet sequence numbering
- time-stamping



RTP Example

- Consider sending 64 kbps PCM-encoded voice over RTP.
- Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk.
- The audio chunk along with the RTP header form the RTP packet, which is encapsulated into a UDP segment.
- RTP header indicates type of audio encoding in each packet
 - sender can change encoding during a conference.
- RTP header also contains sequence numbers and timestamps.

RTP and QoS

- RTP does **not** provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- RTP encapsulation is only seen at the end systems: it is not seen by intermediate routers.
 - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.

RTP Header



RTP Header

Payload Type (7 bits): Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs the receiver through this payload type field.

- Payload type 0: PCM mu-law, 64 kbps
- Payload type 3, GSM, 13 kbps
- Payload type 7, LPC, 2.4 kbps
- Payload type 26, Motion JPEG
- Payload type 31. H.261
- Payload type 33, MPEG2 video

Sequence Number (16 bits): Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence.

RTP Header (2)

- **Timestamp field (32 bits long).** Reflects the sampling instant of the first byte in the RTP data packet.
 - For audio, timestamp clock typically increments by one for each sampling period (for example, each 125 usecs for a 8 KHz sampling clock)
 - if application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active. Timestamp clock continues to increase at constant rate when source is inactive.
- **SSRC field (32 bits long).** Identifies the source of the RTP stream. Each stream in a RTP session should have a distinct SSRC.

RTSP/RTP Programming

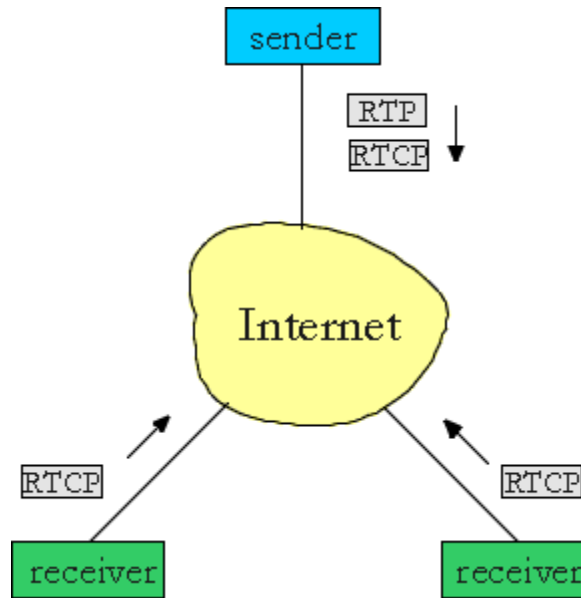
Assignment

- Build a server that encapsulates stored video frames into RTP packets
 - grab video frame, add RTP headers, create UDP segments, send segments to UDP socket
 - include seq numbers and time stamps
 - client RTP provided for you
- Also write the client side of RTSP
 - issue play and pause commands
 - server RTSP provided for you

Real-Time Control Protocol (RTCP)

- Works in conjunction with RTP.
- Each participant in RTP session periodically transmits RTCP control packets to all other participants.
- Each RTCP packet contains sender and/or receiver reports
 - report statistics useful to application
- Statistics include number of packets sent, number of packets lost, interarrival jitter, etc.
- Feedback can be used to control performance
 - Sender may modify its transmissions based on feedback

RTCP - Continued



- For an RTP session there is typically a single multicast address; all RTP and RTCP packets belonging to the session use the multicast address.
- RTP and RTCP packets are distinguished from each other through the use of distinct port numbers.
- To limit traffic, each participant reduces his RTCP traffic as the number of conference participants increases.

RTCP Packets

Receiver report packets:

- fraction of packets lost, last sequence number, average interarrival jitter.

Sender report packets:

- SSRC of the RTP stream, the current time, the number of packets sent, and the number of bytes sent.

Source description packets:

- e-mail address of sender, sender's name, SSRC of associated RTP stream.
- Provide mapping between the SSRC and the user/host name.

Synchronization of Streams

- RTCP can synchronize different media streams within a RTP session.
- Consider videoconferencing app for which each sender generates one RTP stream for video and one for audio.
- Timestamps in RTP packets tied to the video and audio sampling clocks
 - not tied to the wall-clock time
- Each RTCP sender-report packet contains (for the most recently generated packet in the associated RTP stream):
 - timestamp of the RTP packet
 - wall-clock time for when packet was created.
- Receivers can use this association to synchronize the playout of audio and video.

RTCP Bandwidth Scaling

- RTCP attempts to limit its traffic to 5% of the session bandwidth.

Example

- Suppose one sender, sending video at a rate of 2 Mbps. Then RTCP attempts to limit its traffic to 100 Kbps.
- RTCP gives 75% of this rate to the receivers; remaining 25% to the sender
- The 75 kbps is equally shared among receivers:
 - With R receivers, each receiver gets to send RTCP traffic at $75/R$ kbps.
- Sender gets to send RTCP traffic at 25 kbps.
- Participant determines RTCP packet transmission period by calculating avg RTCP packet size (across the entire session) and dividing by allocated rate.