



# RTSP Live Streaming

Frank Wang | Software Engineer

**ABSTRACT**—With the rapid evolution of the IP camera industry, almost every IP surveillance camera supports Real Time Streaming Protocol (RTSP) video streaming, which means the user can operate a media player to watch live views from anywhere. RTSP provides a way for users to control video and audio. Real-time Transport Protocol (RTP) does not actually provide for the transfer of video signals and audio signals; it controls how packages are delivered.

## I - Introduction

RTSP is a network control protocol that is designed for watching a live feed and controlling media sessions between end points. The transmission of streaming data itself is not a task of RTSP. Instead, RTSP uses a combination of reliable transmission over TCP (used for control) and best-efforts delivery over UDP (used for content) to stream content to users. Most RTSP servers use the Real-time Transport Protocol (RTP) in conjunction with Real-time Control Protocol (RTCP) for media stream delivery.

## II - Analysis Of Network Packet

Let's consider an interaction where the client and server will use a combination of TCP-based RTSP and UDP-based RTP and RTCP to deliver a video stream.

We capture the network traffic between our IP camera and clientside by Wireshark. The parameters of experimental scenario are shown in the table below.

### The Parameters of Experimental Scenario

| ITEM           | VALUE       |
|----------------|-------------|
| → Server IP    | 192.168.0.4 |
| → Media Server | LIVE555     |
| → Client IPC0  | 192.168.0.2 |
| → User-Agent   | VLC Player  |

## A. RTSP

RTSP is a protocol used for transferring real-time multimedia data (e.g., audio and video) between a server and a client. It is a streaming protocol; this means that RTSP attempts to facilitate scenarios in which the multimedia data is being simultaneously transferred and rendered (i.e., video is displayed, and audio is played).

The server needs to maintain a session state to be able to correlate RTSP requests with a stream. The state transitions are depicted in Fig.1.

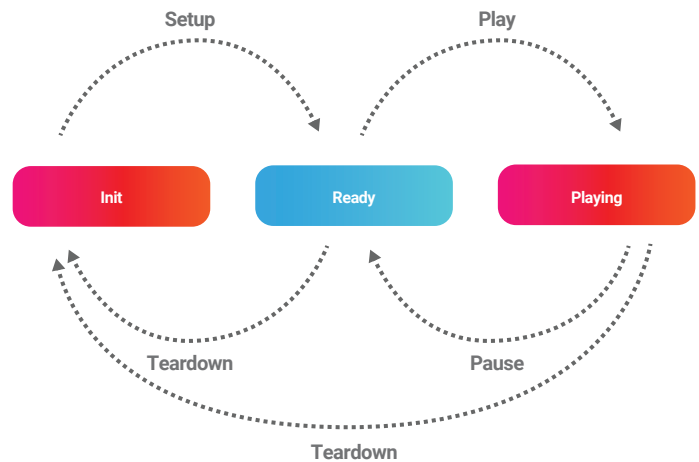


Fig.1. State machine transition diagram of RTSP server.



The basic RTSP requests are described as follows:

### 1. Options

The client will establish a TCP connection to port 554 on the server. An OPTIONS request returns the request types that the server will accept.

```
Client -> Server
Request: OPTIONS tsp://192.168.0.4:554/live/ch00_ORSP/1.0
```

```
Server -> Client
Response: RTSP/1.0 200 OK
Public: OPTIONS, DESCRIBE, SETUP, PAUSE, GET_PARAMETER, SET_PARAMETER
```

### 2. Describe

A DESCRIBE request includes an RTSP URL and the type of reply data that can be handled. This response includes the presentation description, typically in Session Description Protocol (SDP) format.

```
Client -> Server
Request: DESCRIBE rtsp://192.168.0.4:554/live/ch00_RTSP/1.0
User-Agent: LibVLC/2.2.4 (LIVE555 Streaming Media v2016.02.22)
Accept: application/sdp
```

```
Server -> Client
Response: RTSP/1.0 200 OK
Content-type: application/sdp
```

```
Media Description, name and address (m): audio RT/AVP 96
Connection Information (c): IN IP4 0.0.0.0.
Bandwidth Information (b): AS:96
Media Attribute (a): tmap: 96 MPEG4-GENERIC/8000
Media Attribute (a): fmp:96 streamtype=5; profile-level-id=1;mode=AAC
hbr;size-length=13;index-length=3;index-delta-length=3;config=858
Media Attribute (a): control=track1
```

```
Media Description, name and address (m): video RT/AVP 97
Connection Information (c): IN IP4 0.0.0.0.
Bandwidth Information (b): AS:8192
Media Attribute (a): tmap: 97 H264/9000
Media Attribute (a): fmp:97 packetization-mode=1;profile-level-id=4D401F
sprop-parameter-sets=Z01AH+dAAET8uAlkaau4AAK/IGAAADAQAaw
AAGrOw/+MAAADAgAAwAANWfh/8C,aP48gA==
Media Attribute (a): control=track2
```

### 3. Setup

SETUP request specifies how a single media stream must be transported. The request contains the media stream URL and a transport specifier.

This specifier typically includes a local port for receiving RTP data (audio or video), and another for RTCP data (meta information). The server reply usually confirms the chosen parameters, and fills in the missing parts, such as the server's chosen ports. track1 (audio):

```
Client -> Server
Request: SETUP rtsp://192.168.0.4/live/ch00_0/track1RTSP/1.0
Transport: RTP/AVP;unicast;client_port=59842-59843
```

```
Server -> Client
Response: RTSP/1.0 200 OK
Transport:
RTP/AVP;unicast;destination=192.168.0.2;source=192.168.0.4;client_port=59842-59843;server_port=6970-6971
Session: 0DB4F708;timeout=65
```

- Client RTP port is 59842, client RTCP port is 59843
- Server RTP port is 6970, server RTCP port is 6971

track2 (video):

```
Client -> Server
Request: SETUP rtsp://192.168.0.4/live/ch00_0/track2RTSP/1.0\r\n
Transport: RTP/AVP;unicast;client_port=59844-59845
Session: 0DB4F708
```

```
Server -> Client
Response: RTSP/1.0 200 OK
Transport:
RTP/AVP;unicast;destination=192.168.0.2;source=192.168.0.4;client_port=59844-59845;server_port=6972-6973
Session: 0DB4F708;timeout=65
```

- Client RTP port is 59844, client RTCP port is 59845
- Server RTP port is 6972, server RTCP port is 6973



#### 4. Play

A PLAY request will cause one or all media streams to be played.

|                                |  |
|--------------------------------|--|
| <b>Client</b> -> <b>Server</b> |  |
| Request:                       | S ETUP rtsp://192.168.0.4/live/ch00_0/RTSP/1.0                   |
| Session:                       | 0DB4F708   |
| <b>Server</b> -> <b>Client</b> |  |
| Response:                      | RTSP/1.0 200 OK  |
| Session:                       | 0DB4F708   |
| RTP-Info                       |  |
| url=                           | rtsp://192.168.0.4/live/ch00_0/rack1;seq=37454;tptime=1616571534 |
| url=                           | rtsp://192.168.0.4/live/ch00_0/rack2;seq=9004;tptime=4264431073  |

#### 5. Teardown

A TEARDOWN request is used to terminate the session. It stops all media streams and frees all session-related data on the server.

|                                |  |
|--------------------------------|--|
| <b>Client</b> -> <b>Server</b> |  |
| Request:                       | T TEARDOWN rtsp://192.168.0.4/live/ch00_0/RTSP/1.0 |
| Session:                       | 0DB4F708   |
| <b>Server</b> -> <b>Client</b> |  |
| Response:                      | RTSP/1.0 200 OK                                    |

#### B. RTP

RTP is designed to carry the encoded audio and video data. RTP adds a header to each packet, which is then passed to the UDP for further processing.

RTP also provides a time-stamping function that allows multiple streams from the same source to be synchronized. Each form of payload (i.e., video and audio) has a specific way of being mapped into RTP.

Each source inserts time stamps into outgoing packet headers which can be processed by the receiver to recover the stream's clock signal that is needed to correctly play video and audio clips.

As can be seen in Fig. 2, an entire frame can be identified as a sequence of packets ending with a packet having the RTP marker bit set.

|   |               |
|---|---------------|
| PT=DynamicRTP-TYPE-96, SSRC=0x54BCDF1B, Seq=37454, jme=1616571534, Mark | audio x#1     |
| PT=DynamicRTP-TYPE-96, SSRC=0x54BCDF1B, Seq=37455, jme=1616572558, Mark | audio x#1     |
| PT=DynamicRTP-TYPE-96, SSRC=0x54BCDF1B, Seq=37456, jme=1616573582, Mark | audio x#2     |
| PT=DynamicRTP-TYPE-96, SSRC=0x54BCDF1B, Seq=37457, jme=1616574606, Mark | audio x#3     |
| PT=DynamicRTP-TYPE-97, SSRC=0x39483036, Seq=9004, jme=4264431073        | video y (1/6) |
| PT=DynamicRTP-TYPE-97, SSRC=0x39483036, Seq=9005, jme=4264431073        | video y (2/6) |
| PT=DynamicRTP-TYPE-96, SSRC=0x54BCDF1B, Seq=37458, jme=1616575630, Mark | audio x#4     |
| PT=DynamicRTP-TYPE-97, SSRC=0x39483036, Seq=9006, jme=4264431073        | video y (3/6) |
| PT=DynamicRTP-TYPE-97, SSRC=0x39483036, Seq=9007, jme=4264431073        | video y (4/6) |
| PT=DynamicRTP-TYPE-97, SSRC=0x39483036, Seq=9008, jme=4264431073        | video y (5/6) |
| PT=DynamicRTP-TYPE-97, SSRC=0x39483036, Seq=9009, jme=4264431073, Mark  | video y (6/6) |

Fig. 2. Video and audio RTP packets.

For voice packets, the marker bit indicates the beginning of a talkspurt. Beginnings of talkspurts are good opportunities to adjust the playout delay at the receiver to compensate for differences between the sender and receiver clock rates as well as changes in the network delay jitter.

#### C. RTCP

RTCP is a bidirectional UDP-based mechanism to allow the client to communicate stream-quality information back to the server. This connection always uses the next incremental UDP port of the RTP source port. Fig. 3 shows how the three protocols work together in our experimental scenario.

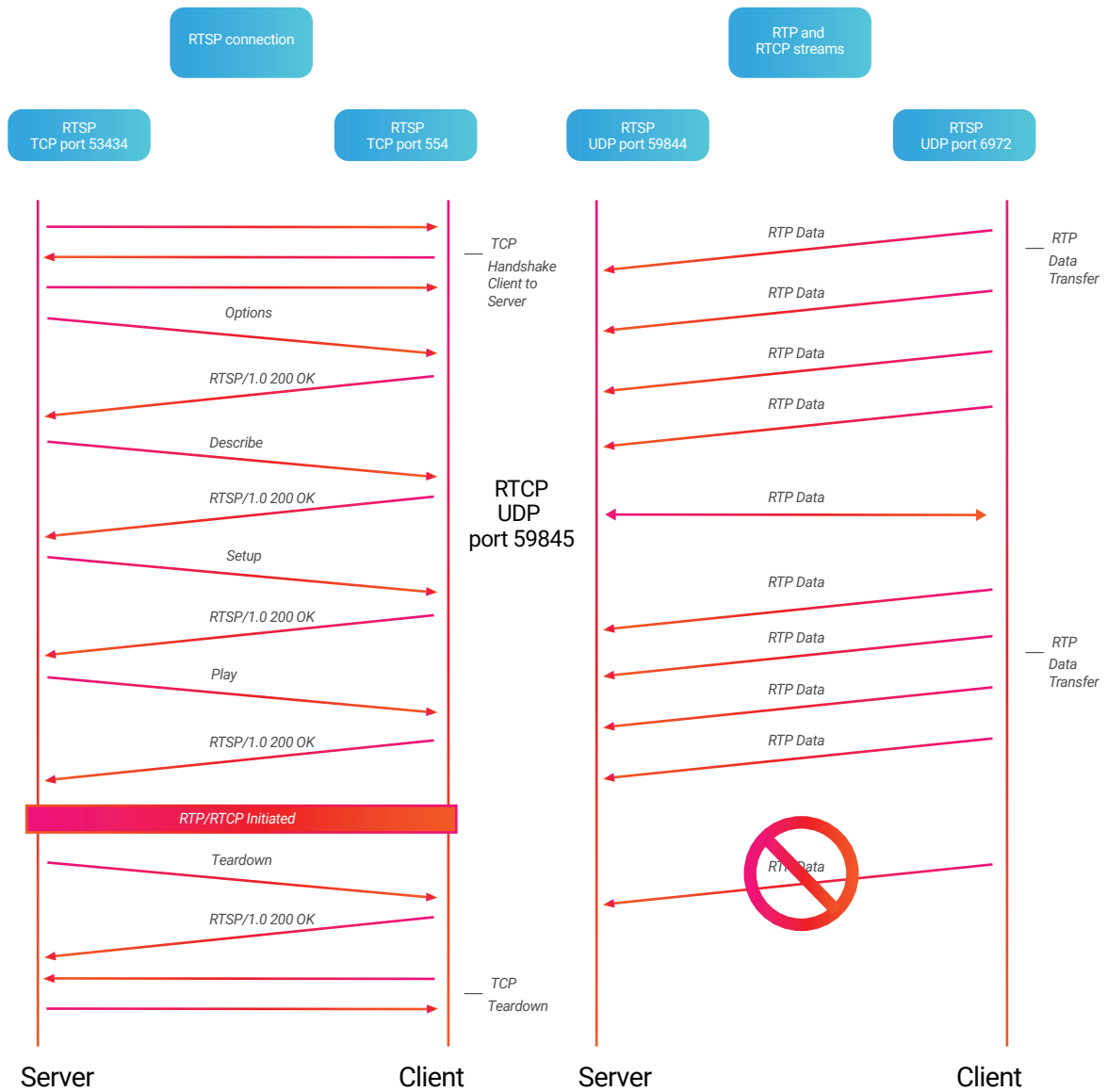


Fig. 3. The three main application protocols used in real-time streaming

### III - Conclusion

The RTSP provides a means for the user to control video and audio. RTSP does not actually provide for the transport of video signals and audio signals; instead, it allows these signals to be controlled by the user. Like a dispatcher for a delivery service, RTSP does not actually deliver packages; it controls when and how packages are delivered by other protocols, such as RTP.

### References

1. M. Syme and P. Goldie, **Optimizing Network Performance with Content Switching: Server, Firewall and Cache Load Balancing: Server, Firewall, and Cache Load Balancing**, Prentice Hall, 2003.