

## Basic Streaming Technology and RTSP Protocol



<http://www.cswl.com/whiteppr/tech/streamingtechnology.html>

### Index

- [Streaming Technology](#)
  - [Definition](#)
  - [The Need for Video on the internet](#)
  - [How does it work](#)
  - [Key Words in Streaming Technology](#)
  - [Protocols Used in Streaming Technology](#)
  - [Example](#)
  - [Streaming Video](#)
  - [Streaming Media Sample](#)
  - [Streaming Media Delivery Methods](#)
  - [Choosing A Delivery Method](#)
- [Real Time Streaming Protocol](#)
  - [Working of RTSP](#)
  - [Functions of RTSP](#)
  - [Differences from HTTP](#)
  - [RTSP Data Packet Formats](#)
  - [RTSP Methods](#)
  - [RTSP Operations](#)
  - [RTSP Request](#)
  - [RTSP Response](#)
  - [How to use RTSP Methods](#)
  - [Request Headers in RTSP](#)
  - [Response Headers in RTSP](#)
  - [Protocol States](#)
- [Advantages of Streaming](#)
- [Disadvantages of Streaming](#)
- [References](#)

### Streaming Technology

*Definition :*

Streaming is the process of playing a file while it is still downloading. Streaming technology, also known as streaming media, lets a user view and hear digitized content —

video, sound and animation — as it is being downloaded. Using a World Wide Web browser plug-in, streamed sounds and images can arrive within seconds of a user's click.

*The need for video on the internet :*

Various media are used in Internet, not only for the obvious purpose of entertainment applications, but for other applications such as video conferencing, video archives and libraries, remote learning, multimedia presentations, and of course, video on demand. Here is a list of the common applications:

- Internet Multimedia.
- Interactive Video Games.
- Personal Communications.
- Video Storage.
- Multimedia Email.
- Database Services and Archives.
- Video Surveillance.
- Emergency Systems.
- Broadcasting.

How Does it Work ?

A web designer can add links that reference a media hosting network, to a website. When a viewer clicks on one of the links, the media network streams the clip directly to the viewer's PC as though it was coming from the original website. Because the media files are hosted on a separate network, the streaming has no impact on any existing web servers.

When a user logs into a hosted streaming network for instance, he or she is automatically fed content from the nearest available server on the Internet. This content is thus delivered at the best quality possible for that connection.

*Key Words in Streaming Technology :*

- *Buffering*  
Receiving and storing data before playing it back.
- *Codec*  
Coder/decoder. Codecs convert data between uncompressed and compressed formats, thereby reducing the bandwidth a clip consumes.
- *Encoding*  
Converting a file into a compressed, streaming format. For example, you can encode WAV files as RealAudio.
- *SMIL*  
Synchronized Multimedia Integration Language. A mark-up language for

specifying how and when each clip plays. SMIL files use the `.smil` or `.smi` extension.

- *Stream*  
A flow of a single type of data, measured in Kilobits per second (Kbps).
- *Multicast*  
Used for broadcasting large events over the Internet. Allows a single computer to create the content (film, etc.) and many computers to play the same single stream simultaneously.
- *Streaming*  
When a large media file (audio, video, etc.) is broken into smaller pieces so it can be viewed or heard immediately. This avoids the wait for the whole file to be downloaded first.
- *Firewall*  
Security devices used to protect companies from unauthorized access to their servers. A firewall, using either proxy services or packet filtering, ensures that all communication between an organization's network and the Internet, conforms to the organization's security policies.
- *On-Demand*  
A type of streaming in which a clip plays from start to finish when a user clicks a link. Most clips are streamed this way.
- *Request headers*  
Request headers are used in client requests to communicate information about the client.
- *Response Headers*  
Response headers are used in server responses to communicate information about the server and how it may handle requests.
- *AVI*  
AVI, or Audio Video Interleave, is the most common format for audio/video data on the PC. It is a special case of the RIFF (Resource Interchange File Format), and is defined by Microsoft.
- *MPEG*  
MPEG (pronounced M-peg) stands for Moving Picture Experts Group. MPEG is the name given to the family of standards used for coding audio-visual information (e.g., movies, video, music) in a digitally compressed format.
- *SMPTE*  
SMPTE is a time code synchronization protocol originally developed for use in the television and motion picture industry where it was used to handle video tape technology.
- *ASF*  
ASF (Advanced Streaming Format) is a file format that stores audio and video information and is specially designed to run over networks like the Internet.

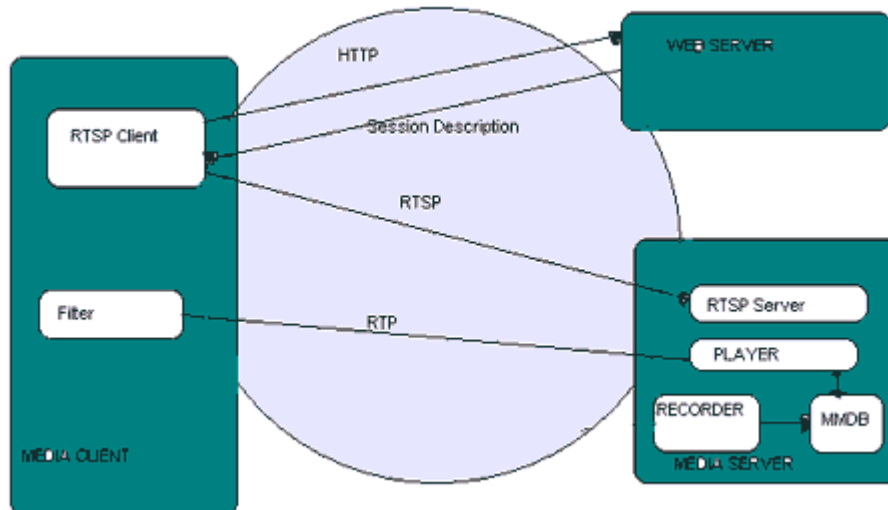
*Protocols Used in Streaming Technology :*

Protocols are the rules implemented for a particular technology. Protocols in streaming technology are used to carry message packets, and communication takes place only through them. Some of the protocols used in streaming technology are:

- *Session Description Protocol (SDP)*  
A media description format intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation.
- *Real Time Transport Protocol (RTP)*  
A UDP packet format and set of conventions that provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services.
- *Real-time Control Protocol (RTCP)*  
RTCP is the control protocol that works in conjunction with RTP. RTCP control packets are periodically transmitted by each participant in an RTP session to all other participants. RTCP is used to control performance and for diagnostic purposes.
- *Hypertext Transfer Protocol (HTTP)*  
An application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol that can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods.
- *Real Time Streaming Protocol (RTSP)*  
An application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video, using the Transmission Control Protocol (TCP) or the User Data Protocol (UDP).

*Example :*

For example, the following diagram depicts how protocols are used to communicate between the components of RTSP system.



### *Streaming Video :*

Streaming video is a sequence of "moving images" that are sent in compressed form over the Internet and are seen by the viewer as they arrive.

### Video Streaming System

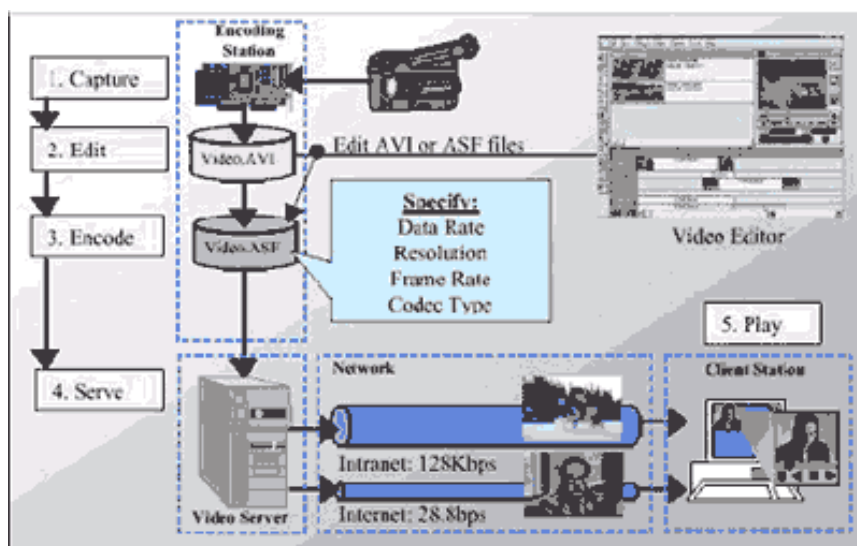
A complete video-streaming system involves all of the basic elements of creating, delivering, and ultimately playing the video content. The main components of a complete video streaming system used to accomplish this—Encoding Station, Video Server, Network Infrastructure, and Playback Client—are illustrated in the following diagram.

- **Step 1. Capture:**  
As this diagram shows, the first step in the process of creating streaming video is to "capture" the video from an analog source such as a camcorder or VHS tape, digitize it and store it to disk. This is usually accomplished with an add-in analog video capture card and the appropriate capture software. Newer digital video sources such as digital video camcorders can be captured straight to disk with a "Firewire" capture board without the analog-to-digital conversion step. The capture card may also support the delivery of "live" video in addition to "stored" video.
- **Step 2. Edit/Author :**  
Once the video is converted to digital and is stored on disk it can be edited using a variety of non-linear editing tools. At this stage, as described below, an authoring tool may also be used to integrate the video with other multimedia into a presentation, entertainment, or training format.
- **Step 3. Encode:**  
After the video is edited and is integrated with other media it may be encoded to the appropriate streaming file format. This generally involves using the encoding software from the video-streaming vendor and specifying the desired output resolution, frame rate, and data rate for the streaming video file. When multiple

data rates need to be supported, multiple files may be produced corresponding to each data rate. As an alternative, newer video streaming technologies create one file that has "dynamic bandwidth adjustment" to the needed client data rate.

- Step 4. Serve:

The video server manages the delivery of video to clients using the appropriate network transport protocols over the network connection. The video server consists of a hardware platform that has been optimally configured for the delivery of real-time video plus video server software that runs under an operating system such as Microsoft Windows NT that acts as a "traffic cop" for the delivery of video streams. Video server software is generally licensed by the "number of streams." If more streams are requested than the server is licensed for, the software rejects the request.



- Step 5. Play:

Finally, at the client station the video player receives and buffers the video stream and plays it in the appropriate size window using a VCR-like user interface. The player generally supports such functions as play, pause, stop, rewind, seek, and fast forward. Client players can run stand-alone or can be ActiveX controls or browser plug-ins. They can decode video using software or using hardware add-in decoder boards.

*Streaming Media Sample :*

- *NETSHOW*

Microsoft WindowsNT® Server NetShow™ Services enable Internet Providers and organizations to deliver the highest-quality audio and video across the Internet or enterprise networks. NetShow Services allow users to receive audio

and video broadcasts from their personal computers. NetShow Services consist of server and tools components for streaming audio, video, illustrated audio, animations, and other multimedia file types over networks.

The streaming media components of the Windows Media Technologies provide a complete solution for integrating audio and video into online applications, bringing the vibrant power of networked multimedia to the Internet and corporate intranets. The Windows Media Player continuously decompresses and plays the content in real time.

The NetShow 3.0 system includes the player, the server, the software developer's kit (SDK), and other tools.

- *Real Media*

RealServer streams audio, video, images, animation, text, and other data types to client computers. This newest version of RealServer has been designed to keep pace with your multimedia needs as they continue to change.

RealServer is server software that streams both live and prerecorded media over a network. The streamed data can originate either on the Internet or within an intranet. The client receives the media in real time, and without having to wait for clips to be downloaded.

- Working of Real Server :

RealServer streams media to clients over networks and the Internet. It's usually employed in conjunction with a Web server. Note that you can use some RealServer features with third-party products to create specialized functions, such as report analysis.

- Channels and Protocols :

RealServer uses two connections, known as channels, to communicate with clients: one for communication with the client, and one for actual data. The communication channel is known as the control channel, as it is over this line that RealServer requests and receives passwords and clients send instructions such as fast-forward, pause, and stop.

Media clips themselves, on the other hand, are actually streamed over a separate data channel. Every link to content begins with a protocol identifier, such as rtsp, pnm, or http.

RealServer uses two main protocols to communicate with clients: Real Time Streaming Protocol (RTSP) and Progressive Networks Audio (PNA). Occasionally, RealServer will use HTTP for metafiles that point to RealServer content, and for HTML pages that it serves (such as the Web-

based RealSystem Administrator). It may also be used to deliver clips to clients that are located behind firewalls. Within these channels, RealServer uses two other protocols for sending instructions and data:

- i. Transport Control Protocol (TCP), for sending commands from the client (such as "start" and "pause") and sending commands from RealServer to clients for specific information (such as the clips' titles).
- ii. User Datagram Protocol (UDP), for sending actual streamed content.

### *Streaming Media Delivery Methods :*

#### **Delivery Methods :**

There are two main ways that users access and experience media clips:

- i. On-demand, as with renting a video at a 24-hour video store, a clip is available to a given user whenever she wants it. The user can fast-forward, rewind, or pause the clip, and MediaServer will send the right part of it. This type of clip is prerecorded or preassembled.
- ii. Live, as with a live telecast of the Olympic Games, a user can tune in to the action that is happening at any given time. Note that a user can not fast-forward or rewind through the clip, because the event is happening in real time. Of course, the delivery of content as a live event requires an actual live event, and it's possible only if you or the content creator have the software and hardware needed to capture the content and convert it to a media format that MediaServer can broadcast.

There's also a third, less common method, which uses on-demand clips but delivers them as if they were live:

- iii. Simulated live: Just as television broadcasts sometimes record live events and then broadcast them later, such as Olympic sports that wouldn't be seen live everywhere because of time-zone differences, simulated live broadcasts take prerecorded events and broadcast them as live ones. Thus, although the content is prerecorded, users view the events as if they were live.

#### *Choosing a Delivery Method :*

Once you've determined how you want the user to experience a given clip (on-demand or live)

### I. On-demand

Prerecorded clips are delivered, or streamed, to users upon request. A user who clicks a link to an on-demand clip watches the clip from the beginning. The user can fast-forward, rewind, or pause the clip.

### II. Live

There are three ways to deliver the clip: unicasting, splitting, or multicasting.

#### i. *Unicasting*

This is the simplest and most popular method of live broadcasting, as it requires little or no configuration.

#### ii. *Splitting*

Splitting is the term used to describe how one MediaServer can share its live media streams with other MediaServers. Clients connect to these other RealServers, called splitters, rather than to the main MediaServer where the streams originate. Splitting reduces the traffic load on the source MediaServer, enabling it to distribute other broadcasts simultaneously.

#### iii. *Multicasting*

Multicasting is a standardized method for delivering presentations to large numbers of users over a network or the Internet.

### III. Simulated Live

The same delivery options are available as for live broadcasting: unicasting, splitting, and multicasting. The only difference is that, with simulated live broadcasting, the event has already been recorded, and no connection to a production tool or encoder is needed.

## **Real-Time Streaming Protocol**

The main protocol that is used in Streaming is RTSP Protocol.

Real-time Streaming Protocol is an application-level protocol that aims to provide a robust protocol for streaming multimedia in one-to-many applications over unicast and multicast, and to support interoperability between clients and servers from different vendors. RTSP is considered more of a framework than a protocol. RTSP is designed to work on top of RTP to both control and deliver real-time content.

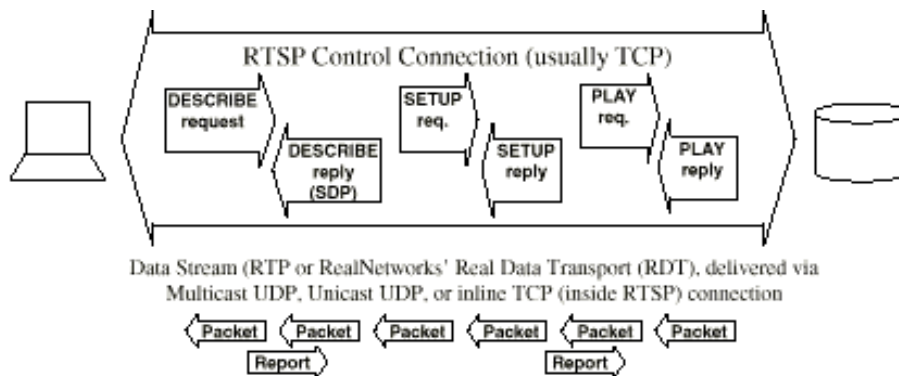
### How does RTSP Work?

RTSP takes advantage of streaming which breaks data into packets sized according to the bandwidth available between client and server. When the client has received enough packets, the user's software can be playing one packet, decompressing another, and downloading the third. This enables the user to listen or view the real-time file almost immediately, and without downloading the entire media file. This applies to live data feeds as well as stored clips.

### *Functions of RTSP :*

- Provides for on-demand access of multimedia items such as stored real-time audio/video files, live real-time feeds, or stored non-real-time items.
- Allows interoperability between client-server multimedia products from multiple vendors.
- Provides for control and delivery of real-time media and associated events between a media server and large numbers of media clients.
- Addresses key concerns of Internet content-providers and users - quality of service, efficiency of delivery, rights management, and measurement. It also provides a underpinning for developing the richest possible streaming multimedia applications.

An example of how RTSP works with other protocols is shown below.



### *Differences from HTTP :*

- An RTSP server needs to maintain state by default in almost all cases.
- Both an RTSP server and client can issue requests.
- The Request-URI always contains the absolute URI

### *RTSP Data Packet Formats :*

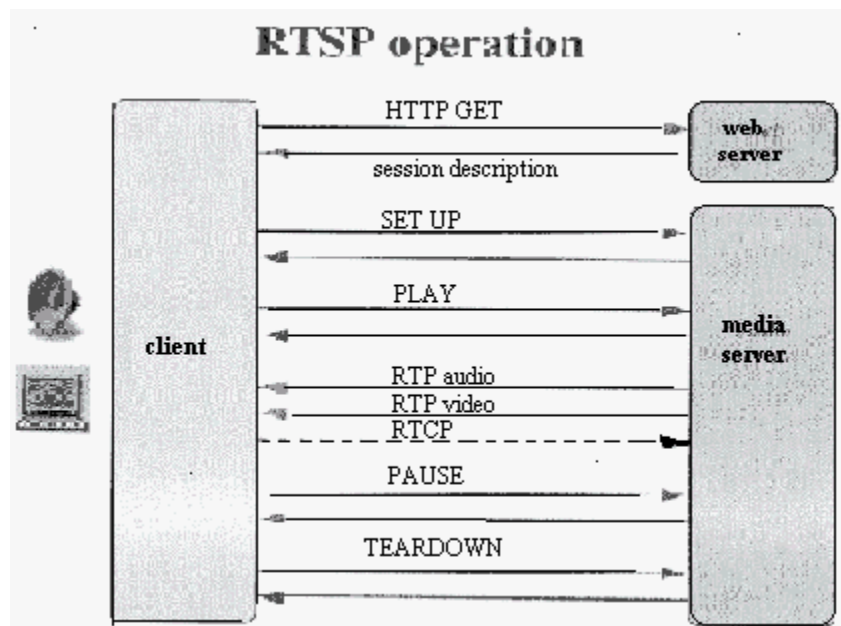
RealServer uses one of two packet formats for sending media data to an RTSP client:

- Standard Real Time Transport Protocol, RTP
- RealNetworks' Real Data Transport, RDT

### *RTSP Methods :*

Method	Description
Options	Get available methods
SETUP	Establish transport
ANNOUNCE	Change description of media object
DESCRIBE	Get description of media object
PLAY	Start playback, reposition
RECORD	Start recording
REDIRECT	Redirect client to new server
PAUSE	Halt delivery, but keep state
SET-PARAMETER	Device or encoding control
TEARDOWN	Remove state

*RTSP Operation :*



Putting all the methods together, to send a control request, the client constructs a line consisting of the method, the request URL, and the protocol version number. Then, the client includes a general header, a request header and possibly an entity header, as for the http protocol. This is sent to the server, which executes the request if possible. It then

returns a response containing a status-line and general response and entity headers. The status line contains the protocol version, the numeric status code, and a textual description. The media streams are left unspecified by RTSP. These could be RTP streams, or any other form of media transmission. RTSP only specifies the control and its up to the client and server software to maintain the mapping between the control channel and the media streams.

A key concept in RTSP is the notion of a session. RTSP works by first requesting a presentation to be started by a server, receiving in return a session identifier which it then uses in all subsequent controls. Eventually, the client can request the teardown of session, which releases the associated resources. The session identifier represents the shared state between the client and server. If the state is lost, for example through one of the machines being rebooted, then the protocol relies on the transport of the media stopping automatically, eg. through not receiving RTCP messages if using RTP, or the implementation using the GET\_PARAMETER method below as a keep-alive.

The control requests and responses may be sent over either TCP or UDP. Since the order of the requests matters, the requests are sequenced, so if any requests are lost, they must be retransmitted. Using UDP thus requires the construction of retransmission mechanisms, so there are very few occasions when the application can get away with using UDP.

The most obvious additions to the request header fields are a Cseq field to contain the sequence numbers of requests generated by the client, and a Session field to both the request and response headers to identify the session. Session identifiers are generated in response to a SETUP request, and must be used in all stateful methods. The Transport field allows the client and server to negotiate and set parameters for the sending of the media stream. In particular, it allows the client and server to set ports and multicast addresses for the RTP streams.

#### *RTSP Request :*

A request message from a client to a server or vice versa includes, within the first line of that message, the method to be applied to the resource, the identifier of the resource, and the protocol version in use.

Request-Line = Method SP

Request-URI SP RTSP-Version CRLF

Request = Request-Line \*

(general-header | request-header | entity-header )

CRLF [ message-body ]

Method = "DESCRIBE" | "ANNOUNCE" | "GET\_PARAMETER" |

"OPTIONS" | "PAUSE" | "PLAY" | "RECORD" | "REDIRECT" |

SETUP" | "SET\_PARAMETER" | "TEARDOWN" | extension-method

extension-method = token  
Request-URI = "\*" | absolute\_URI  
RTSP-Version = "RTSP" "/" 1\*DIGIT "." 1\*DIGIT

#### *RTSP Response :*

After receiving and interpreting a request message, the recipient responds with an RTSP response message.

Response = Status-Line  
\*( general-header | response-header | entity-header )  
CRLF [ message-body ]

The first line of a Response message is the Status-Line, consisting of the protocol version followed by a numeric status code; and the textual phrase associated with the status code, with each element separated by SP characters. No CR or LF is allowed except in the final CRLF sequence.

Status-Line = RTSP-Version  
SP Status-Code SP Reason-Phrase CRLF

The Status-Code element is a 3-digit integer result code of the attempt to understand and satisfy the request. The Reason-Phrase is intended to give a short textual description of the Status-Code. The Status-Code is intended for use by automata and the Reason-Phrase is intended for the human user. The client is not required to examine or display the Reason-Phrase.

#### *How to use RTSP Methods :*

In this section, the most important RTSP methods will be described.

#### *DESCRIBE*

The DESCRIBE method retrieves the description of a presentation or media object identified by the request URL from a server. The DESCRIBE reply-response pair constitutes the media initialization phase of RTSP.

Example:

```
C->S: DESCRIBE
rtsp://server.example.com/fizzle/foo RTSP/1.0
CSeq: 312
Accept: application/sdp,
application/rtsp, application/mpeg
```

S->C: RTSP/1.0 200 OK  
CSeq: 312  
Date: 23 Jan 1997 15:35:06 GMT  
Content-Type: application/sdp  
Content-Length: 376  
v=0  
o=mhandley 2890844526  
2890842807 IN IP4 126.16.64.4  
s=SDP Seminar  
i=A Seminar on the session  
description protocol  
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps e=mjh@iis.edu (Mark Handley)  
c=IN IP4 224.2.17.12/127  
t=2873397496 2873404696  
a=recvonly  
m=audio 3456 RTP/AVP 0  
m=video 2232 RTP/AVP 31  
m=whiteboard 32416 UDP WB  
a=orient:portrait

### *SETUP*

The SETUP request for a URI specifies the transport mechanism to be used for the streamed media. The Transport header specifies the transport parameters acceptable to the client for data transmission; the response will contain the transport parameters selected by the server.

Example:

C->S: SETUP rtsp://example.com/foo/bar/baz.rm RTSP/1.0 CSeq: 302  
Transport: RTP/AVP;unicast;client\_port=4588-4589

S->C: RTSP/1.0 200 OK  
CSeq: 302  
Date: 23 Jan 1997 15:35:06 GMT  
Session: 47112344  
Transport: RTP/AVP;unicast;  
client\_port=4588-4589;  
server\_port=6256-6257

### *PLAY*

The PLAY method tells the server to start sending data via the mechanism specified in

SETUP. A client MUST NOT issue a PLAY request until any outstanding SETUP requests have been acknowledged as successful. The PLAY request positions the normal playtime to the beginning of the range specified and delivers stream data until the end of the range is reached.

Example:

```
C->S: PLAY rtsp://audio.example.com/twister.en RTSP/1.0
CSeq: 833
Session: 12345678
Range: smpte=0:10:20-;time=19970123T153600Z
```

```
S->C: RTSP/1.0 200 OK
CSeq: 833
Date: 23 Jan 1997 15:35:06 GMT
Range: smpte=0:10:22-;time=19970123T153600Z
```

#### *PAUSE*

The PAUSE request causes the stream delivery to be interrupted (halted) temporarily. If the request URL names a stream, only playback and recording of that stream is halted. For example, for audio, this is equivalent to muting. If the request URL names a presentation or group of streams, delivery of all currently active streams within the presentation or group is halted. After resuming playback or recording, synchronization of the tracks MUST be maintained. Any server resources are kept, though servers MAY close the session and free resources after being paused for the duration specified with the timeout parameter of the Session header in the SETUP message.

Example:

```
C->S: PAUSE rtsp://example.com/fizzle/foo RTSP/1.0
CSeq: 834
Session: 12345678
```

```
S->C: RTSP/1.0 200 OK
CSeq: 834
Date: 23 Jan 1997 15:35:06 GMT
```

#### *TEARDOWN*

The TEARDOWN request stops the stream delivery for the given URI, freeing the resources associated with it. If the URI is the presentation URI for this presentation, any RTSP session identifier associated with the session is no longer valid. Unless all transport

parameters are defined by the session description, a SETUP request has to be issued before the session can be played again.

Example:

C->S: TEARDOWN rtsp://example.com/fizzle/foo RTSP/1.0  
CSeq: 892  
Session: 12345678

S->C: RTSP/1.0 200 OK  
CSeq: 892

*Request Headers in RTSP :*

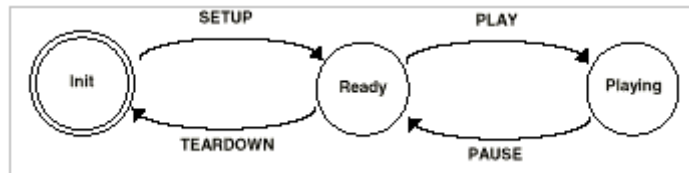
ACCEPT	Media description format
ACCEPT-ENCODING	Encoding of media format
ACCEPT-LANGUAGE	Human language
AUTHORIZATION	Authentication
BANDWIDTH	Client bandwidth available
CONFERENCE	Conference identifier
FROM	Name of requestor
If-Modified since	Conditional retrieval
RANGE	Time range to play
Referrer	How did we get here?
SPEED	Speed-up Delivery
User Agent	Software

*Response Headers of RTSP :*

Location	Redirection
Proxy Authentication	Authenticate to proxy
Public	Methods supported
Retry-After	Busy, comeback later
Server	Server software
Vary	Cache
Www-authenticate	Request authorization

*Protocol States :*

Unlike HTTP, RTSP does have a certain amount of state associated with it. Here is a simple state machine that describes the minimal state in RTSP:



**Advantages of Streaming**

- Audio and video begins playing soon after the stream begins. The user doesn't have to download the whole file in order to listen music or watch video.
- Sound quality is good.
- Artists and publishers can control distribution and protect copyright because the user never gets a copy of the audio file.

**Disadvantages of Streaming**

- High cost of server software.
- Sound quality and stream may be affected by low speed or inconsistent Internet connections.
- Requires a preconfigured server.

**References**

1. Transferring real-time video on the Internet  
<http://www.tml.hut.fi/Opinnot/Tik-110.551/1997/iwsem.html>
2. Streaming products -- comparison  
<http://streamingmedialand.com/industry.html>
3. Optimize RealServer for real-time delivery of multimedia files  
<http://www.service.real.com/help/library/guides/server8/htmlfiles/intro.htm>

4. RealServer Administration Guide

<http://service.real.com/help/library/index.html>

5. For creating a multimedia extravaganza with video, audio, animation, images, and text.

<http://service.real.com/help/library/guides/productiong261/htmlfiles/intro.htm>