

## 第一部分:RTSP协议

### 一、RTSP协议概述

RTSP(Real-TimeStream Protocol )是一种基于文本的应用层协议，在语法及一些消息参数等方面，RTSP协议与HTTP协议类似。

RTSP被用于建立的控制媒体流的传输，它为多媒体服务扮演“网络远程控制”的角色。尽管有时可以把RTSP控制信息和媒体数据流交织在一起传送，但一般情况RTSP本身并不用于转送媒体流数据。媒体数据的传送可通过RTP/RTCP等协议来完成。

一次基本的RTSP操作过程是:首先，客户端连接到流服务器并发送一个RTSP描述命令（DESCRIBE）。流服务器通过一个SDP描述来进行反馈，反馈信息包括流数量、媒体类型等信息。客户端再分析该SDP描述，并为会话中的每一个流发送一个RTSP建立命令(SETUP)，RTSP建立命令告诉服务器客户端用于接收媒体数据的端口。流媒体连接建立完成后，客户端发送一个播放命令(PLAY)，服务器就开始在UDP上传送媒体流（RTP包）到客户端。在播放过程中客户端还可以向服务器发送命令来控制快进、快退和暂停等。最后，客户端可发送一个终止命令(TEARDOWN)来结束流媒体会话

### 二、RTSP协议与HTTP协议区别

1. RTSP引入了几种新的方法，比如DESCRIBE、PLAY、SETUP等，并且有不同的协议标识符，RTSP为rtsp 1.0,HTTP为http 1.1；
2. HTTP是无状态的协议，而RTSP为每个会话保持状态；
3. RTSP协议的客户端和服务端都可以发送Request请求，而在HTTP协议中，只有客户端能发送Request请求。
4. 在RTSP协议中，载荷数据一般是通过带外方式来传送的(除了交织的情况)，及通过RTP协议在不同的通道中来传送载荷数据。而HTTP协议的载荷数据都是通过带内方式传送的，比如请求的网页数据是在回应的消息体中携带的。
5. 使用ISO10646(UTF-8)而不是ISO 8859-1，以配合当前HTML的国际化；
6. RTSP使用URI请求时包含绝对URI。而由于历史原因造成的向后兼容性问题，HTTP/1.1只在请求中包含绝对路径，把主机名放入单独的标题域中；

### 三、RTSP重要术语

#### 1. 集合控制(Aggregatecontrol):

对多个流的同时控制。对音频/视频来讲，客户端仅需发送一条播放或者暂停消息就可同时控制音频流和视频流。

#### 2. 实体(Entity):

作为请求或者回应的有效负荷传输的信息。由以实体标题域(entity-header field)形式存在的元信息和以实体主体(entity body)形式存在的内容组成

#### 3. 容器文件(Containerfile):

可以容纳多个媒体流的文件。RTSP服务器可以为这些容器文件提供集合控制。

#### 4. RTSP会话(RTSP session):

RTSP交互的全过程。对一个电影的观看过程,会话(session)包括由客户端建立媒体流传输机制(SETUP)，使用播放(PLAY)或录制(RECORD)开始传送流，用停止(TEARDOWN)关闭流。

### 四、RTSP请求消息

#### 1. 消息格式:

方法 URI RTSP 版本 CR LF

消息头 CRLF CRLF

消息体 CR LF

其中方法包括OPTIONS、DESCRIBE、SETUP、PLAY、TEARDOWN等,URI是接受方的地址,例如:rtsp://192.168.0.1/video1.3gp。

RTSP版本一般都是 RTSP/1.0。每行后面的CR LF表示回车换行，需要接受端有相应的解析，最后一个消息头需要有两个CR LF

消息体是可选的，有的Request消息并不带消息体。

### 五、RTSP回应消息

## 1. 消息格式：

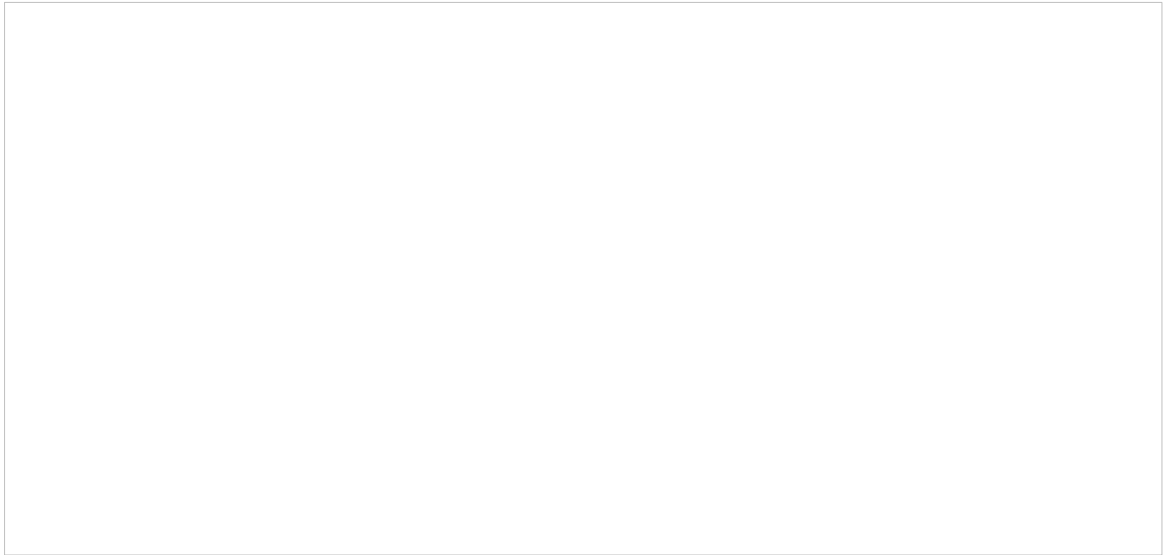
*RTSP 版本 状态码 解释 CR LF*

*消息头 CR LF      CR LF*

*消息体 CR LF*

其中RTSP版本一般都是RTSP/1.0,状态码是一个数值，用于表示Request消息的执行结果,比如200表示成功,解释是与状态码对应的文本解释.

## 六、RTSP 重要方法



### 1. OPTIONS:

用于得到服务器提供的可用方法；

如:

*OPTIONS rtsp://192.168.20.136:5000/xxx666 RTSP/1.0*

*CSeq: 1*

服务器的回应信息会在Public字段列出提供的方法。如:

*RTSP/1.0 200 OK*

*CSeq: 1      // 每个回应消息的cseq 数值和请求消息的cseq相对应*

*Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE*

### 2. DESCRIBE：

客户端向服务器端发送DESCRIBE，用于得到URI所指定的媒体描述信息，一般是SDP信息。客户端通过Accept头指定客户端可以接受的媒体述信息类型。

如：

*C->S: DESCRIBE rtsp://server.example.com/fizzle/fooRTSP/1.0*

*CSeq: 312*

*Accept: application/sdp, application/rtsp,application/mhég)*

服务器回应URI指定媒体的描述信息:

如:

*S->C: RTSP/1.0 200 OK*

*CSeq: 312*

*Date: 23 Jan 1997 15:35:06 GMT*

*Content-Type: application/sdp // 表示回应为SDP 信息*

Content-Length: 376

// 这里为一个空行

// 以下为具体的SDP 信息

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@isi.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

媒体初始化是任何基于RTSP系统的必要条件，但RTSP规范并没有规定它必须通过DESCRIBE方法完成。RTSP客户端可以通过以下方法来接收媒体描述信息：

- a) 通过DESCRIBE方法；
- b) 其它一些协议（HTTP，email附件，等）；
- c) 通过命令行或标准输入设备

### 3. SETUP：

用于确定传输机制，建立RTSP会话。客户端能够发出一个SETUP请求为正在播放的媒体流改变传输参数，服务器可能同意这些参数的改变。若是不同意，它必须响应错误"455 Method Not Valid In This State"。

Request中的Transport头字段指定了客户端可接受的数据传输参数；Response中的Transport 头字段包含了由服务器选出的传输参数。

如:

C->S: SETUP rtsp://example.com/foo/bar/baz.rm RTSP/1.0

CSeq: 302

Transport: RTP/AVP;unicast;client\_port=4588-4589

服务器端对SETUPRequest产生一个Session Identifiers。

如:

S->C: RTSP/1.0 200 OK

CSeq: 302

Date: 23 Jan 1997 15:35:06 GMT

Session: 47112344 // 产生一个SessionID

Transport: RTP/AVP;unicast;

client\_port=4588-4589;server\_port=6256-6257

### 4. PLAY：

PLAY方法告知服务器通过SETUP中指定的机制开始发送数据。在尚未收到SETUP请求的成功应答之前，客户端不可以发出PLAY请求。

PLAY请求将正常播放时间(normal play time)定位到指定范围的起始处，并且传输数据流直到播放范围结束。PLAY请求可能被管道化(pipeline

d)，即放入队列中（queued）；服务器必须将PLAY请求放到队列中有序执行。也就是说，后一个PLAY请求需要等待前一个PLAY请求完成才能得到执行。

比如，在下例中，不管到达的两个PLAY请求之间有多紧凑，服务器首先play第10到15秒，然后立即第20到25秒，最后是第30秒直到结束。

```
C->S: PLAY rtsp://audio.example.com/audio RTSP/1.0
```

```
CSeq: 835
```

```
Session: 12345678
```

```
Range: npt=10-15
```

```
C->S: PLAY rtsp://audio.example.com/audio RTSP/1.0
```

```
CSeq: 836
```

```
Session: 12345678
```

```
Range: npt=20-25
```

```
C->S: PLAY rtsp://audio.example.com/audio RTSP/1.0
```

```
CSeq: 837
```

```
Session: 12345678
```

```
Range: npt=30-
```

Range头可能包含一个时间参数。该参数以UTC格式指定了播放开始的时间。如果在这个指定时间后收到消息，那么播放立即开始。时间参数可能用来帮助同步从不同数据源获取的数据流。

不含Range头的PLAY请求也是合法的。它从媒体流开头开始播放，直到媒体流被暂停。如果媒体流通过PAUSE暂停，媒体流传输将在暂停点（the pause point）重新开始。

如果媒体流正在播放，那么这样一个PLAY请求将不起更多的作用，只是客户端可以用此来测试服务器是否存活。

## 5. PAUSE：

PAUSE请求引起媒体流传输的暂时中断。如果请求URL中指定了具体的媒体流，那么只有该媒体流的播放和记录被暂停（halt）。比如，指定暂停音频，播放将会无声。如果请求URL指定了一组流，那么在该组中的所有流的传输将被暂停。如：

```
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
```

PAUSE请求中可能包含一个Range头用来指定何时媒体流暂停，我们称这个时刻为暂停点（pause point）。该头必须包含一个精确的值，而不是一个时间范围。媒体流的正常播放时间设置成暂停点。当服务器遇到在任何当前挂起（pending）的PLAY请求中指定的时间点后，暂停请求生效。如果Range头指定了一个时间超出了任何一个当前挂起的PLAY请求，将返回错误“457 Invalid Range”。如果一个媒体单元（比如一个音频或视频帧）正好在一个暂停点开始，那么表示将不会被播放或记录。如果Range头缺失，那么在收到暂停消息后媒体流传输立即中断，并且暂停点设置成当前正常播放时间。

## 6. TEARDOWN：

TEARDOWN请求终止了给定URI的媒体流传输，并释放了与该媒体流相关的资源。如：

```
C->S: TEARDOWN rtsp://example.com/fizzle/foo RTSP/1.0
```

```
CSeq: 892
```

```
Session: 12345678
```

```
S->C: RTSP/1.0 200 OK
```

```
CSeq: 892
```

## 七、RTSP重要头字段参数

### 1. Accept:

用于指定客户端可以接受的媒体描述信息类型。比如:

```
Accept: application/rtsl, application/sdp;level=2
```

### 2. Bandwidth:

用于描述客户端可用的带宽值。

### 3. CSeq :

指定了RTSP请求回应的序列号, 在每个请求或回应中都必须包括这个头字段。对每个包含一个给定序列号的请求消息, 都会有一个相同序列号的回应消息。

### 4. Rang :

用于指定一个时间范围, 可以使用SMPTE、NTP或clock时间单元。

### 5. Session:

Session头字段标识了一个RTSP会话。Session ID 是由服务器在SETUP的回应中选择的, 客户端一旦得到Session ID后, 在以后的对Session 的操作请求消息中都要包含Session ID。

### 6. Transport:

Transport头字段包含客户端可以接受的传输选项列表, 包括传输协议, 地址端口, TTL等。服务器端也通过这个头字段返回实际选择的具体选项。如:

```
Transport: RTP/AVP;multicast;tli=127;mode="PLAY",
```

```
RTP/AVP;unicast;client_port=3456-3457;mode="PLAY"
```

## 八、简单的RTSP消息交互过程

C表示RTSP客户端,S表示RTSP服务端

### 1. 第一步：查询服务器端可用方法

1.C->S:OPTIONrequest //询问S有哪些方法可用

1.S->C:OPTIONresponse //S回应信息的public头字段中包括提供的所有可用方法

### 2. 第二步：得到媒体描述信息

2.C->S:DESCRIBE request //要求得到S提供的媒体描述信息

2.S->C:DESCRIBE response //S回应媒体描述信息, 一般是sdp信息

### 3. 第三步：建立RTSP会话

3.C->S:SETUPrequest //通过Transport头字段列出可接受的传输选项, 请求S建立会话

3.S->C:SETUPresponse //S建立会话, 通过Transport头字段返回选择的具体传输选项, 并返回建立的Session ID;

### 4. 第四步：请求开始传送数据

4.C->S:PLAY request //C请求S开始发送数据

4.S->C:PLAYresponse //S回应该请求的信息

### 5. 第五步：数据传送播放中

S->C:发送流媒体数据 // 通过RTP协议传送数据

### 6. 第六步：关闭会话，退出

6.C->S:TEARDOWN request //C请求关闭会话

6.S->C:TEARDOWN response //S回应该请求

上述的过程只是标准的、友好的rtsp流程, 但实际的需求中并不一定按此过程。

其中第三和第四步是必需的! 第一步, 只要服务器客户端约定好, 有哪些方法可用, 则option请求可以不要。第二步, 如果我们有其他途径得到媒体初始化描述信息(比如http请求等等), 则我们也不需要通过rtsp中的describe请求来完成。

## 第二部分:SDP协议

### 一、SDP协议概述

SDP(SessionDescription Protocol )会话描述协议,用于描述多媒体会话,它为会话通知、会话初始和其它形式的多媒体会话初始等操作提供服务。

SDP的设计宗旨是通用性协议,所有它可以应用于很大范围的网络环境和应用程序,但 SDP 不支持会话内容或媒体编码的协商操作。

SDP信息包括:

- 会话名称和目标;
- 会话活动时间;
- 构成会话的媒体;
- 有关接收媒体的信息、地址等。

### 二、SDP格式

SDP 信息是文本信息,UTF-8 编码采用 ISO 10646 字符设置。SDP 会话描述如下(标注\*符号的表示可选字段):

- v= (协议版本)
- o= (所有者/创建者和会话标识符)
- s= (会话名称)
- i=\* (会话信息)
- u=\* (URI 描述)
- e=\* (Email 地址)
- p=\* (电话号码)
- c=\* (连接信息 — 如果包含在所有媒体中,则不需要该字段)
- b=\* (带宽信息)

一个或更多时间描述(如下所示):

- z=\* (时间区域调整)
- k=\* (加密密钥)
- a=\* (0个或多个会话属性线路)
- 0个或多个媒体描述(如下所示)

时间描述

- t= (会话活动时间)
- r=\* (0或多次重复次数)

媒体描述

- m= (媒体名称和传输地址)
- i=\* (媒体标题)
- c=\* (连接信息 — 如果包含在会话层则该字段可选)
- b=\* (带宽信息)
- k=\* (加密密钥)
- a=\* (0个或多个会话属性线路)

### 三、SDP示例

*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@isi.edu (Mark Handley)*

*c=IN IP4 224.2.17.12/127*

*t=2873397496 2873404696*

*a=recvonly*

*m=audio 49170 RTP/AVP 0*

*m=video 51372 RTP/AVP 31*

*m=application 32416 udp wb*

*a=orient:portrait*

//字段解释

V=0 ; **V**ersion 给定了SDP协议的版本

o=<username><session id> <version> <network type> <address type>

<address> ; **O** rigin ,给定了会话的发起者信息

s=<sessionname> ;给定了 **S** ession Name

i=<sessiondescription> ; **I** nformation 关于Session的一些信息

u=<URI> ; **U** RI

e=<emailaddress> ; **E** mail

c=<networktype> <address type> <connection address> ; **C** onnect Data包含连接数据

t=<start time><stop time> ; **T** ime

a=<attribute> ; **A** ttribute

a=<attribute>:<value>

m=<media><port> <transport> <fmt list> ; **M** ediaAnnouncements

文章标签 : [rtsp](#) [协议](#) [学习笔记](#)

个人分类 : [IPTV](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com