

6 RTSP协议

FFmpeg源码对应目录

RTSP协议概述

RTSP协议与HTTP协议区别

推流过程

推流详细过程

第一步：OPTION 查询服务器端可用方法

第二步：ANNOUNCE 发送媒体描述信息

第三步：SETUP建立RTSP会话

第四步：RECORD请求传送数据

第五步：RTP数据推送

第六步：TEARDOWN关闭会话，退出

总结

拉流过程

拉流详细过程

第一步：查询服务器端可用方法

第二步：DESCRIBE得到媒体描述信息

第三步：SETUP建立RTSP会话

第四步：PLAY请求开始传送数据

第五步：RTP数据传送播放中

第六步：TEARDOWN关闭会话，退出

提问

总结

腾讯课堂 零声学院

FFmpeg/WebRTC/RTMP 音视频流媒体高级开发 <https://ke.qq.com/course/468797?tuin=137bb271>

FFmpeg源码对应目录

rtsp.c 是总入口，推流拉流都是该文件

rtspenc.c：推流的时候使用

rtspdec.c：拉流的时候使用

RTSP协议概述

RTSP 很详细的英文文档 <https://blog.csdn.net/u012519333/article/details/52746375>

RTSP(Real-Time Stream 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

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 session的概念**；
3. RTSP协议的客户端和服务端都可以发送Request请求，而在HTTPF 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ISO 10646(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；

推流过程

推流详细过程

第一步：OPTION 查询服务器端可用方法

1.C->S:OPTION request //询问S有哪些方法可用

```

Real Time Streaming Protocol
  Request: OPTIONS rtsp://192.168.2.170:554/live/test RTSP/1.0\r\n
    Method: OPTIONS
    URL: rtsp://192.168.2.170:554/live/test
    CSeq: 1\r\n
    User-Agent: Lavf58.29.100\r\n
    \r\n

```

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

```

Response: RTSP/1.0 200 OK\r\n
  Status: 200
  CSeq: 1\r\n
  Date: Wed, Nov 04 2020 06:34:32 GMT\r\n
  Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, ANNOUNCE, RECORD, SET_PARAMETER, GET_PARAMETER\r\n
  Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
  \r\n

```

第二步：ANNOUNCE 发送媒体描述信息

2.C->S:ANNOUNCE request //客户端发送媒体描述信息给服务器

```

Request: ANNOUNCE rtsp://192.168.2.170:554/live/test RTSP/1.0\r\n
  Method: ANNOUNCE
  URL: rtsp://192.168.2.170:554/live/test
  Content-type: application/sdp
  CSeq: 2\r\n
  User-Agent: Lavf58.29.100\r\n
  Content-length: 494
  \r\n
Session Description Protocol
  Session Description Protocol Version (v): 0
  > Owner/Creator, Session Id (o): - 0 0 IN IP4 127.0.0.1
  Session Name (s): No Name
  > Connection Information (c): IN IP4 192.168.2.170
  > Time Description, active time (t): 0 0
  Session Start Time: 0
  Session Stop Time: 0
  > Session Attribute (a): tool:libavformat 58.29.100
  > Media Description, name and address (m): video 0 RTP/AVP 96
  > Media Attribute (a): rtpmap:96 H264/90000
  > Media Attribute (a): fmp:96 packetization-mode=1; sprop-parameter-sets=Z2QAHzQLax5ahAAAAADAEAAAAyDxYtIgA==,a0vjyyIA; profile-level-id=64001E
  > Media Attribute (a): control:streamid=0
  > Media Description, name and address (m): audio 0 RTP/AVP 97
  > Bandwidth Information (b): AS:128
  > Media Attribute (a): rtpmap:97 MPEG4-GENERIC/48000/2
  > Media Attribute (a): fmp:97 profile-level-id=1;mode=AAC-hbr;size-length=13;index-length=3;index-delta-length=3; config=119056E500
  > Media Attribute (a): control:streamid=1

```

2.S->C:ANNOUNCE response //S回应媒体描述信息，并返回了Session ID

```

Real Time Streaming Protocol
  Response: RTSP/1.0 200 OK\r\n
    Status: 200
    Content-Base: rtsp://192.168.2.170:554/live/test/\r\n
    CSeq: 2\r\n
    Date: Wed, Nov 04 2020 06:34:32 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: ditmIrW4Z20X
    \r\n

```

session id

第三步：SETUP建立RTSP会话

3.1C→S:SETUP request //通过Transport头字段列出可接受的传输选项，请求S建立会话

```
Real Time Streaming Protocol
  Request: SETUP rtsp://192.168.2.170:554/live/test/streamid=0 RTSP/1.0\r\n
    Method: SETUP
    URL: rtsp://192.168.2.170:554/live/test/streamid=0
    Transport: RTP/AVP/UDP;unicast;client_port=31590-31591;mode=record
    CSeq: 3\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: ditmIrW4Z20X
    \r\n
```

RTP:31590 RTCP:31591

3.1 S→C:SETUP response //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;

```
Real Time Streaming Protocol
  Response: RTSP/1.0 200 OK\r\n
    Status: 200
    CSeq: 3\r\n
    Date: Wed, Nov 04 2020 06:34:32 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: ditmIrW4Z20X
    Transport: RTP/AVP/UDP;unicast;client_port=31590-31591;mode=record;server_port=59472-59473;ssrc=00000000
    \r\n
```

C→S

RTP: 31590 -> 59472

RTCP: 31591 -> 59473

3.2C→S:SETUP request //通过Transport头字段列出可接受的传输选项，请求S建立会话

```
Real Time Streaming Protocol
  Request: SETUP rtsp://192.168.2.170:554/live/test/streamid=1 RTSP/1.0\r\n
    Method: SETUP
    URL: rtsp://192.168.2.170:554/live/test/streamid=1
    Transport: RTP/AVP/UDP;unicast;client_port=31592-31593;mode=record
    CSeq: 4\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: ditmIrW4Z20X
    \r\n
```

3.2 S→C:SETUP response //S建立会话，通过Transport头字段返回选择的具体传输选项

```

✓ Real Time Streaming Protocol
  ✓ Response: RTSP/1.0 200 OK\r\n
    Status: 200
    CSeq: 4\r\n
    Date: Wed, Nov 04 2020 06:34:32 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: ditmIrW4Z20X
    Transport: RTP/AVP/UDP;unicast,client_port=31592-31593;mode=record,server_port=53582-53583;ssrc=00000000\r\n

```

第四步： RECORD请求传送数据

4.C->S:RECORD request //C向S请求发送数据

```

✓ Real Time Streaming Protocol
  ✓ Request: RECORD rtsp://192.168.2.170:554/live/test RTSP/1.0\r\n
    Method: RECORD
    URL: rtsp://192.168.2.170:554/live/test
    Range: npt=0.000-\r\n
    CSeq: 5\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: ditmIrW4Z20X
    \r\n

```

4.S->C:RECORD response //S回应该允许的信息

```

✓ Real Time Streaming Protocol
  ✓ Response: RTSP/1.0 200 OK\r\n
    Status: 200
    CSeq: 5\r\n
    Date: Wed, Nov 04 2020 06:34:32 GMT\r\n
    RTP-Info: url=rtsp://192.168.2.170:554/live/test/streamid=0,url=rtsp://192.168.2.170:554/live/test/streamid=1\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: ditmIrW4Z20X
    \r\n

```

第五步： RTP数据推送

C->S:发送流媒体数据 // 通过RTP协议传送数据

视频

```

✓ User Datagram Protocol, Src Port: 31590, Dst Port: 59472
  Source Port: 31590
  Destination Port: 59472
  Length: 707
  Checksum: 0xfe18 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 13]
  > [Timestamps]
✓ Real-Time Transport Protocol
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 .... = Extension: False
  .... 0000 = Contributing source identifiers count: 0
  0... .... = Marker: False
  Payload type: DynamicRTP-Type-96 (96)
  Sequence number: 3497
  Timestamp: 2794369990
  Synchronization Source identifier: 0xd801e570 (3624002928)
  Payload: 0605ffffabdc45e9bde6d948b7962cd820d923eeef783236...

```

音频

```

✓ User Datagram Protocol, Src Port: 31592, Dst Port: 53582
  Source Port: 31592
  Destination Port: 53582
  Length: 1071
  Checksum: 0xbbe7 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 15]
  > [Timestamps]
✓ Real-Time Transport Protocol
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 .... = Extension: False
  .... 0000 = Contributing source identifiers count: 0
  1... .... = Marker: True
  Payload type: DynamicRTP-Type-97 (97)
  Sequence number: 2386
  Timestamp: 4216229973
  Synchronization Source identifier: 0x58e9b611 (1491711505)
  Payload: 00300a200bb00ac8de02004c61766335382e35342e313030...

```

还有一些RTCP的包，这里先不表，都是client->server

```

User Datagram Protocol, Src Port: 31591, Dst Port: 59473
  Source Port: 31591
  Destination Port: 59473
  Length: 36
  Checksum: 0x5eb7 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 12]
  > [Timestamps]
Real-time Transport Control Protocol (Sender Report)
  10... .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 0000 = Reception report count: 0
  Packet type: Sender Report (200)
  Length: 6 (28 bytes)
  Sender SSRC: 0xd801e570 (3624002928)
  Timestamp, MSW: 3813460481 (0xe34cca01)
  Timestamp, LSW: 55834574 (0x0353f7ce)
  [MSW and LSW as NTP timestamp: Nov  4, 2020 06:34:41.012999999 UTC]
  RTP timestamp: 2795018440
  Sender's packet count: 142
  Sender's octet count: 36794
  [RTCP frame length check: OK - 28 bytes]

```

视频的

第六步：TEARDOWN关闭会话，退出

6.C->S:TEARDOWN request //C请求关闭会话

```

Real Time Streaming Protocol
  Request: TEARDOWN rtsp://192.168.2.170:554/live/test RTSP/1.0\r\n
    Method: TEARDOWN
    URL: rtsp://192.168.2.170:554/live/test
    CSeq: 6\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: ditmIrW4Z20X
    \r\n

```

6.S->C:TEARDOWN response //S回应该请求

```

Real Time Streaming Protocol
  Response: RTSP/1.0 200 OK\r\n
    Status: 200
    CSeq: 6\r\n
    Date: Wed, Nov 04 2020 06:35:00 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: ditmIrW4Z20X
    \r\n

```

总结

Source	Destination	Protocol	Length	Info
192.168.2.170	192.168.2.223	RTSP		279 Reply: RTSP/1.0 200 OK
192.168.2.223	192.168.2.170	RTSP/SDP		548 ANNOUNCE rtsp://192.168.2.170:554/live/test RTSP/1.0
192.168.2.170	192.168.2.223	RTSP		248 Reply: RTSP/1.0 200 OK
192.168.2.223	192.168.2.170	RTSP		245 SETUP rtsp://192.168.2.170:554/live/test/streamid=0 RTSP/1.0
192.168.2.170	192.168.2.223	RTSP		303 Reply: RTSP/1.0 200 OK
192.168.2.223	192.168.2.170	RTSP		245 SETUP rtsp://192.168.2.170:554/live/test/streamid=1 RTSP/1.0
192.168.2.170	192.168.2.223	RTSP		303 Reply: RTSP/1.0 200 OK
192.168.2.223	192.168.2.170	RTSP		186 RECORD rtsp://192.168.2.170:554/live/test RTSP/1.0
192.168.2.170	192.168.2.223	RTSP		308 Reply: RTSP/1.0 200 OK
192.168.2.223	192.168.2.170	RTSP		169 TEARDOWN rtsp://192.168.2.170:554/live/test RTSP/1.0
192.168.2.170	192.168.2.223	RTSP		197 Reply: RTSP/1.0 200 OK

拉流过程

拉流详细过程

第一步：查询服务器端可用方法

1.C->S:OPTION request //询问S有哪些方法可用

```
Real Time Streaming Protocol
  Request: OPTIONS rtsp://192.168.2.170:554/live/test RTSP/1.0\r\n
    Method: OPTIONS
    URL: rtsp://192.168.2.170:554/live/test
    CSeq: 1\r\n
    User-Agent: Lavf58.29.100\r\n
    \r\n
```

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

```
Response: RTSP/1.0 200 OK\r\n
  Status: 200
  CSeq: 1\r\n
  Date: Wed, Nov 04 2020 06:34:32 GMT\r\n
  Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, ANNOUNCE, RECORD, SET_PARAMETER, GET_PARAMETER\r\n
  Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
  \r\n
```

第二步：DESCRIBE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


```

v Real Time Streaming Protocol
  v Request: DESCRIBE rtsp://192.168.2.170:554/live/test RTSP/1.0\r\n
    Method: DESCRIBE
    URL: rtsp://192.168.2.170:554/live/test
    Accept: application/sdp\r\n
    CSeq: 2\r\n
    User-Agent: Lavf58.29.100\r\n
    \r\n

```

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

```

v Real Time Streaming Protocol
  v Response: RTSP/1.0 200 OK\r\n
    Status: 200
    Content-Base: rtsp://192.168.2.170:554/live/test/\r\n
    Content-length: 544
    Content-type: application/sdp
    CSeq: 2\r\n
    Date: Wed, Nov 04 2020 07:18:41 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: prmc37lePTE7
    x-Accept-Dynamic-Rate: 1\r\n
    x-Accept-Retransmit: our-retransmit\r\n
    \r\n
  v Session Description Protocol
    Session Description Protocol Version (v): 0
    > Owner/Creator, Session Id (o): - 0 0 IN IP4 127.0.0.1
    > Connection Information (c): IN IP4 192.168.2.170
    v Time Description, active time (t): 0 0
      Session Start Time: 0
      Session Stop Time: 0
      Session Name (s): Streamed by ZLMediaKit-5.0(build in Oct 24 2020 00:20:46)
    > Session Attribute (a): tool:libavformat 58.29.100
    > Media Description, name and address (m): video 0 RTP/AVP 96
    > Media Attribute (a): fmp:96 packetization-mode=1; sprop-parameter-sets=Z2QAHqzZQLAX5ahAAAAADAEAAAAyDxYtIgA=,a0vjyyLA; profile-level-id=64001E
    > Media Attribute (a): rtpmap:96 H264/90000
    > Media Attribute (a): control:streamid=0
    > Media Description, name and address (m): audio 0 RTP/AVP 97
    > Bandwidth Information (b): AS:128
    > Media Attribute (a): fmp:97 profile-level-id=1;mode=AAC-hbr;sizelength=13;indexlength=3;indexdeltalength=3; config=119056E500
    > Media Attribute (a): rtpmap:97 MPEG4-GENERIC/48000/2
    > Media Attribute (a): control:streamid=1

```

第三步：SETUP建立RTSP会话

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

```

v Real Time Streaming Protocol
  v Request: SETUP rtsp://192.168.2.170:554/live/test/streamid=0 RTSP/1.0\r\n
    Method: SETUP
    URL: rtsp://192.168.2.170:554/live/test/streamid=0
    Transport: RTP/AVP/UDP;unicast client_port=21988-21989
    x-Dynamic-Rate: 0\r\n
    CSeq: 3\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: prmc37lePTE7
    \r\n

```

3.1 S->C:SETUP response //S建立会话，通过Transport头字段返回选择的具体传输选项

```

v Real Time Streaming Protocol
  v Response: RTSP/1.0 200 OK\r\n
    Status: 200
    CSeq: 3\r\n
    Date: Wed, Nov 04 2020 07:18:41 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: prmc37lePTE7
    Transport: RTP/AVP/UDP;unicast;client_port=21988-21989;server_port=39188-39189;ssrc=77EF5B7F\r\n

```

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

```

v Real Time Streaming Protocol
  v Request: SETUP rtsp://192.168.2.170:554/live/test/streamid=1 RTSP/1.0\r\n
    Method: SETUP
    URL: rtsp://192.168.2.170:554/live/test/streamid=1
    Transport: RTP/AVP/UDP;unicast;client_port=21990-21991
    x-Dynamic-Rate: 0\r\n
    CSeq: 4\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: prmc37lePTE7
    \r\n

```

3.2 S->C:SETUP response //S建立会话，通过Transport头字段返回选择的具体传输选项

```

v Real Time Streaming Protocol
  v Response: RTSP/1.0 200 OK\r\n
    Status: 200
    CSeq: 4\r\n
    Date: Wed, Nov 04 2020 07:18:41 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: prmc37lePTE7
    Transport: RTP/AVP/UDP;unicast;client_port=21990-21991;server_port=55182-55183;ssrc=29029EC5\r\n

```

第四步：PLAY请求开始传送数据

4.C->S:PLAY request //C请求S开始发送数据

```

v Real Time Streaming Protocol
  v Request: PLAY rtsp://192.168.2.170:554/live/test/ RTSP/1.0\r\n
    Method: PLAY
    URL: rtsp://192.168.2.170:554/live/test/
    Range: npt=0.000-\r\n
    CSeq: 5\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: prmc37lePTE7
    \r\n

```

4.S->C:PLAYresponse //S回应该请求的信息

```

v Response: RTSP/1.0 200 OK\r\n
  Status: 200
  CSeq: 5\r\n
  Date: Wed, Nov 04 2020 07:18:41 GMT\r\n
  Range: npt=6.72\r\n
  RTP-Info: url=rtsp://192.168.2.170:554/live/test/streamid=0;seq=4073;rtptime=633600,url=rtsp://192.168.2.170:554/live/test/streamid=1;seq=1068;rtptime=322560\r\n
  Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
  Session: prmc37lePTE7
\r\n

```

可能流已经播放了一段时间，npt=起始时间

```
url=rtsp://192.168.2.170:554/live/test/streamid=0;seq=4073;rtptime=633600,
```

```
url=rtsp://192.168.2.170:554/live/test/streamid=1;seq=1068;rtptime=322560\r\n
```

第五步： RTP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

视频 这里的ssrc 来自服务器

```

v Real-Time Transport Protocol
> [Stream setup by RTSP (frame 169)]
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 .... = Extension: False
  .... 0000 = Contributing source identifiers count: 0
  1... .... = Marker: True
  Payload type: DynamicRTP-Type-96 (96)
  Sequence number: 4072
  [Extended sequence number: 69608]
  Timestamp: 298181738
  Synchronization Source identifier: 0x77ef5b7f (2012175231)
  Payload: 019f2b6a457f00000414174cb1d95a713f10ee0d29fa37c1...

```

SSRC 就是在setup返回来的SSRC。

音频 这里的ssrc 来自服务器

```

Real-Time Transport Protocol
> [Stream setup by RTSP (frame 173)]
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 .... = Extension: False
  .... 0000 = Contributing source identifiers count: 0
  1... .... = Marker: True
  Payload type: DynamicRTP-Type-97 (97)
  Sequence number: 1069
  [Extended sequence number: 66605]
  Timestamp: 74801586
  Synchronization Source identifier: 0x29029ec5 (688037573)
  Payload: 003009e00af00ad021194acbfdba00246e8e2b0c9803a151...

```

第六步：TEARDOWN关闭会话，退出

6.C->S:TEARDOWN request //C请求关闭会话

```

▼ Real Time Streaming Protocol
  ▼ Request: TEARDOWN rtsp://192.168.2.170:554/live/test RTSP/1.0\r\n
    Method: TEARDOWN
    URL: rtsp://192.168.2.170:554/live/test
    CSeq: 6\r\n
    User-Agent: Lavf58.29.100\r\n
    Session: ditmIrW4Z20X
    \r\n

```

6.S->C:TEARDOWN response //S回应该请求

```

▼ Real Time Streaming Protocol
  ▼ Response: RTSP/1.0 200 OK\r\n
    Status: 200
    CSeq: 6\r\n
    Date: Wed, Nov 04 2020 06:35:00 GMT\r\n
    Server: ZLMediaKit-5.0(build in Oct 24 2020 00:20:23)\r\n
    Session: ditmIrW4Z20X
    \r\n

```

小总结：

推流拉流

第一步 option是一样的

第二步 有区别，推流：ANNOUNCE； 拉流：DESCRIBE

第三步：SETUP

第四步：推流：RECORD；拉流：PLAY

第五步：RTP传输，只是方向刚好相反

第六步：TEARDOWN

提问

- SSRC由谁来定义，是client还是server?
 - 推流是客户端自己定义的
 - 拉流的时候是服务器发送过来的
- session id由谁而来
 - 推流来自server，当C→S ANNOUNCE后，server reply时附带session id
- Transport字段的意义，比如

```
Transport: RTP/AVP/UDP;unicast;client_port=21990-21991;server_port=55182-55183;ssrc=29029EC5
```

- rtptime有什么作用
- rtp的时间戳问题
- 如何做音视频同步
- npt是什么

总结

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。

RTSP服务器默认端口是554，在客户端SETUP的时候会把自身的RTP和RTCP端口告知服务器。在RTSP的session建立后，会使用RTP/RTCP在约定好的端口上传输数据。

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/rtsp, 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请求回应对的序列号，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，客户端一当得

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;ttl=127;mode="PLAY",
RTP/AVP;unicast;client_port=3456-3457;mode="PLAY"