

RTSP请求报文的结构如下图所示

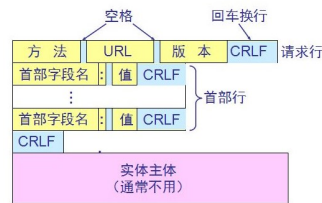


图2 RTSP请求报文的结构

RTSP请求报文的方法包括: OPTIONS、DESCRIBE、SETUP、TEARDOWN、PLAY、PAUSE、GET_PARAMETER和SET_PARAMETER。

一个请求消息 (a request message) 即可以由客户端向服务端发起也可以由服务端向客户端发起。请求消息的语法结构如下:

```
Request = Request-Line

1  *( general-header | request-header | entity-header)
2
3  CRLF
4
5  [message-body]
```

1. Request Line

请求消息的第一行的语法结构如下:

```
Request-Line = Method 空格 Request-URI 空格 RTSP-Version CRLF
```

其中在消息行中出现的第一个单词即是所使用的信令标志。目前已有的信息标志如下:

```
1  Method = "DESCRIBE"
2          | "ANNOUNCE"
3          | "GET_PARAMETER"
4          | "OPTIONS"
5          | "PAUSE"
6          | "PLAY"
7          | "RECORD"
8          | "REDIRECT"
9          | "SETUP"
10         | "SET_PARAMETER"
11         | "TEARDOWN"
```

例子:

```
DESCRIBE rtsp://211.94.164.227/3.3gp RTSP/1.0
```

2. Request Header Fields

在消息头中除了第一行的内容外, 还有一些需求提供附加信息。其中有些是一定要的, 后续我们会详细介绍经常用到的几个域的含义。

```
1  Request-header = Accept
2                  | Accept-Encoding
3                  | Accept-Language
4                  | Authorization
5                  | From
6                  | If-Modified-Since
7                  | Range
8                  | Referer
9                  | User-Agent
```

响应消息

响应报文的开始行是状态行, RTSP响应报文的结构如下图所示

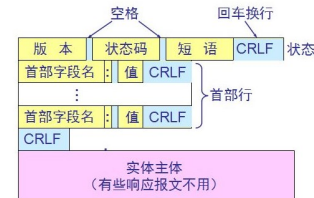


图3 RTSP响应报文的结构

响应消息的语法结构如下:

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

1. Status-Line

响应消息的第一行是状态行 (status-line), 每个元素之间用空格分开。除了最后的CRLF之外, 在此行的中间不得有CR或是LF的出现。它的语法格式如下,

```
Status-Line = RTSP-Version 空格 Status-Code 空格 Reason-Phrase CRLF
```

状态码 (Status-Code) 是一个三位数的整数, 用于描述接收方对所收到请求消息的执行结果

Status-Code的第一位数字指定了这个回复消息的种类, 一共有5类:

- [] 1XX: Informational – 请求被接收到, 继续处理
- [] 2XX: Success – 请求被成功的接收, 解析并接受
- [] 3XX: Redirection – 为完成请求需要更多的操作
- [] 4XX: Client Error – 请求消息中包含语法错误或是不能够被有效执行
- [] 5XX: Server Error – 服务器响应失败, 无法处理正确的有效的请求消息

我们在处理问题时经常会遇到的状态码有如下:

Status-Code	=
-	
-	
-	

2. Response Header Fields

在响应消息的域中存放的是无法放在Status-Line中,而又需要传递给请求者的一些附加信息。

```
1  Response-header = Location
2                  | Proxy-Authenticate
3                  | Public
```

4			Retry-After
5			Server
6			Vary
7			WWW-Authenticate

RTSP的主要方法：

方法	方向	对象	要求	含义
DESCRIBE	C->S	PS	推荐	检查演示或媒体对象的描述。也允许使用接收头指定用户理解的描述格式。DESCRIBE的答复-响应组成媒体RTSP初始阶段
ANNOUNCE	C->S S->C	PS 可选	可选	当从用户发往服务器时，ANNOUNCE将请求URL识别的演示或媒体对象描述发送给服务器；反之，ANNOUNCE实时更新连接描述。如新媒体流加入演示，整个演示描述再次发送，而不仅仅是附加组件，使组件能被删除
GET_PARAMETER	C->S S->C	PS 可选	可选	GET_PARAMETER请求检查RUL指定的演示与媒体的参数值。没有实体体时，GET_PARAMETER也许都用来测试用户与服务器的连接情况
OPTIONS	C->S S->C	PS 要求	要求	可在任意时刻发出OPTIONS请求，如用户打算尝试非标准请求，并不影响服务器状态
PAUSE	C->S	PS 推荐	推荐	PAUSE请求引起流发送临时中断。如请求URL命名一个流，仅回放和记录被停止；如请求URL命名一个演示或流组，演示或组中所有当前活动的流发送都停止。恢复回放或记录后，必须维持同步。在SETUP消息中连接头超时参数所指定时间段内被暂停后，尽管服务器可能关闭连接并释放资源，但服务器资源会被预订
PLAY	C->S	PS 要求	要求	PLAY告诉服务器以SETUP指定的机制开始发送数据；直到一些SETUP请求被成功响应，客户端才可发布PLAY请求。PLAY请求将正常播放时间设置在所指定范围的起始处，发送流数据直到范围的结束处。PLAY请求可排成队列，服务器将PLAY请求排成队列，顺序执行
RECORD	C->S	PS 可选	可选	该方法根据演示描述初始化媒体数据记录范围，时标反映开始和结束时间；如没有给出时间范围，使用演示描述提供的开始和结束时间。如连接已经启动，立即开始记录，服务器数据请求URL或其他URL决定是否存储记录的数据；如服务器没有使用URL请求，响应应为201（创建），并包含描述请求状态和参考新资源的实体与位置头。支持现场演示记录的媒体服务器必须支持时间范围格式，smpte格式没有意义
REDIRECT	S->C	PS 可选	可选	重定向请求通知客户端连接到另一服务器地址。它包含强制头地址，指示客户端发布URL请求；也可能包括参数范围，以指明重定向到何时生效。若客户端要继续发送或接收URL媒体，客户端必须对当前连接发送TEARDOWN请求，而对指定主机新连接发送SETUP请求
SETUP	C->S	S 要求	要求	对URL的SETUP请求指定用于流媒体的传输机制。客户端对正播放的流发布一个SETUP请求，以改变服务器允许的传输参数。如不允许这样做，响应错误为“455 Method Not Valid In This State”，为了通过防火墙，客户端必须指明传输参数，即使对这些参数没有影响
SET_PARAMETER	C->S S->C	PS 可选	可选	请求设置演示或URL指定的参数值。请求仅包含单个参数，允许客户端决定某个特殊请求为何失败。如请求包含多个参数，所有参数可成功设置，服务器必须只对该请求起作用。服务器必须允许参数可重设置或同时一道，也不允许改变数据。注意：当改变传输参数时必须用SETUP命令设置。将设置传输参数解释为SETUP有利于防火墙，将参数划分成规则排列形式，结果有更多有意义的错误指示
TEARDOWN	C->S	PS 要求	要求	TEARDOWN请求停止给定的URL流发送，释放相关资源。如URL是此演示URL，任何RTSP连接标识不再有效，除非全部传输参数是连接描述定义的，SETUP请求必须在连接可再次播放前发布

注：P—演示，C—客户端 S—服务器，S（对象栏）—流

指令指的是在Request-URI中指定的需要被接收者完成的操作。指令（The method）大小写敏感，不能以字符“\$”开始，并且一定是一个标记符。

RTSP重要头字段参数

- 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
- Bandwidth:
用于描述客户端可用的带宽值。
3. CSeq:
指定了RTSP请求应对应的序列号，在每个请求或响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响应消息。
4. Rang:
用于指定一个时间范围，可以使用SMPT、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的响应中选择的，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选项。如:
Transport: RTP/AVP;multicast;ttl=127;mode="PLAY",
RTP/AVP;unicast;client_port=3456-3457;mode="PLAY"

简单的RTSP消息交互过程

C表示RTSP客户端,S表示RTSP服务端

第一步：查询服务器端可用方法

C->S OPTION request //询问S有哪些方法可用
S->C OPTION response //S响应信息的public头字段中包括提供的所有可用方法

第二步：得到媒体描述信息

C->S DESCRIBE request //要求得到S提供的媒体描述信息
S->C DESCRIBE response //S回应媒体描述信息，一般是sdp信息

第三步：建立RTSP会话

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
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;

第四步：请求开始传送数据

C->S PLAY request //C请求S开始发送数据
S->C PLAY response //S回应应该请求的信息

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

第六步： 关闭会话，退出

C->S EARDOWN request //C请求关闭会话
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，只要服务器和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），则我们也不需要通过rtsp中的describe请求来完成。

RTSP的请求响应示例

其中C是客户端，S是服务器。

OPTIONS

1	C->S:	OPTION request //询问S有哪些方法可用
2	S->C:	OPTION response //S响应信息中包括提供的所有可用方法

```
OPTIONS rtsp://218.207.101.236:554/mobile/3/67A451E937422331 RTSP/1.0
Cseq: 1
```

服务端对OPTIONS的回应：（服务器的回应信息会在Public字段列出提供的方法。）

```
RTSP/1.0 200 OK
Server: PVSS/1.4.8 (Build/20090111; Platform/Win32; Release/StarValley;)
Cseq: 1
Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, OPTIONS, ANNOUNCE, RECORD
```

DESCRIBE

```
1 C->S: DESCRIBE request //要求得到S提供的媒体初始化描述信息
2 S->C: DESCRIBE response //S回应媒体初始化描述信息，主要是sdp
```

客户端到服务端的请求示例：（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。）

```
DESCRIBE rtsp://218.207.101.236:554/mobile/3/67A451E937422331/8jH5QPU5GWS07Ugn.sdp RTSP/1.0
Cseq: 2
```

服务端对DESCRIBE的回应：（服务器回应URI指定媒体的描述信息）

```
1 RTSP/1.0 200 OK
2 Server: PVSS/1.4.8 (Build/20090111; Platform/Win32; Release/StarValley;)
3 Cseq: 2
4 Content-length: 421
5 Date: Mon, 03 Aug 2009 08:21:33 GMT
6 Expires: Mon, 03 Aug 2009 08:21:33 GMT
7 Content-Type: application/sdp
8 x-Accept-Retransmit: our-retransmit
9 x-Accept-Dynamic-Rate: 1
10 Content-Base: rtsp://218.207.101.236:554/mobile/3/67A451E937422331/8jH5QPU5GWS07Ugn.sdp/
11
12
13 v=0
14 o=MediaBox 127992 137813 IN IP4 0.0.0.0
15 s=RTSP Session
16 i=Starv Box Live Cast
17 c=IN IP4 218.207.101.236
18 t=0 0
19 a=range:npt=now-
20 a=control:*
21 m=video 0 RTP/AVP 96
22 b=AS:20
23 a=rtmap:96 NP4V-ES/1000
24 a=fmtp:96 profile-level-id=8; config=000001b008000001b50900000100000012008440fa282c2090a31f; decode_buf=12586
25 a=range:npt=now-
26 a=framerate:5
27 a=framesize:96 176-144
28 a=cliprect:0,0,144,176
29 a=control:trackID=1
```

转载地址 <http://blog.csdn.net/xiejiaشو/article/details/5194804>



相关推荐

- [RTSP协议详解中文版](#) 01-10
- [实时流协议 \(RTSP\) 是应用层协议，控制实时数据的传送。RTSP提供了一个可扩展框架，使受控。按需传输实时数据（如音频与视频）成为可能。数据...](#)
- [RTSP视频流显示\(海康威视\) 热门推荐](#) 会飞的蜗牛 475+
- [RTSP视频流显示\(海康威视\) VLCSDK \(C++\) ffmpeg+Nginx 本文的主要是想在html上实时显示海康威视的摄像头数据。笔者尝试了如下三种方式: ...](#)
- [RTSP协议详解及实例分析_king_weng的博客_rtpsp](#) 3-18
- [RTSP协议详解及实例分析 1、RTSP简介 RTSP\(Real-TimeStreamProtocol\)协议是一个基于文本的多媒体播放控制协议,属于应用层。RTSP以客户端方式...](#)
- [RTSP协议解析](#) rongfengqian的专栏 375+
- [RTSP能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放在一起传送，但...](#)
- [RTSP协议学习](#) 菜鸡的学习笔记 475+
- [RTSP简介 RTSP和HTTP RTP/RTCP的关系 RTSP和HTTP RTSP和RTPT/RTCP RTSP消息 简单的rtsp交互过程 RTSP中常用方法 OPTION DESCRIBE SET...](#)
- [RTSP协议详解 最新发布](#) shui_36002055的博文 2937
- [参考：手册RTSP协议系列 Rtp基本流程 rtp协议简介 rtp，英文全称 Real Time Streaming Protocol，RFC3236，实时流传输协议。是TCP/IP协议体系...](#)
- [实时流协议—RTSP【详解】](#) 王木木 2472
- [一、RTSP协议介绍 RTSP（Real Time Streaming Protocol 实时流协议）是一个有效地在IP网络上传输流媒体数据的应用层协议。RTSP对流媒体提供了...](#)
- [流媒体传输协议介绍](#) machin的专栏 5109
- [流媒体传输协议介绍一、RTSP协议介绍什么是rtsp? RTSP协议以客户服务器方式工作，，如：暂停/继续、后退、前进等。它是一个多媒体播放控制协议...](#)
- [RTSP主要流程](#) huabaochen的博文 9325
- [流媒体推流流程 option -> Announce -> Setup -> Record 流媒体拉流流程 option -> Describe -> Setup -> Play 1、OPTIONS OPTIONS#rtsp/10.137.10.1...](#)
- [流媒体网络协议—RTSP](#) Ritchie_Lin的博文 3377
- [RTSP/RTSP介绍1、与HTTP协议的异同2、RTSP的特性RTSP原理1、RTSP会话交互过程2、RTSP客户端状态机：RTSP请求与回应1 RTSP方法一览：2 RT...](#)
- [RTSP协议](#) 天空的博文 2743
- [RTSP\(Real-Time Stream Protocol\)协议是一个基于文本的多媒体播放控制协议，属于应用层。RTSP以客户端方式工作，对流媒体提供播放、暂停、后退...](#)
- [RTSP协议实例分析](#) dosthing的博文 175+
- [1、前言互联网上关于RTSP的文章很多，但是大多数都是抽象的理论介绍。本文将从实际例子解说RTSP协议。不求面面俱到，但求简单易懂。RTSP（R...](#)
- [RTSP协议—图文解释](#) 熊乏的博文 9208
- [一、什么是RTSP RTSP（Real Time Streaming Protocol）实时流传输协议。是 TCP/IP 协议体系中的一个应用层协议。RTSP在体系结构上位于RTP和RT...](#)
- [流媒体传输协议系列之—RTSP协议详解](#) qinglongyeyue的博文 7660
- [转自http://www.mamimcode.com/info-detail-1444337.html 流媒体传输协议介绍 一、RTSP协议介绍 什么是rtsp? RTSP协议以客户服务器方式工作，，如：...](#)
- [RTSP协议讲解](#) 一叶知秋 246
- [写在前面 目前正在学习RTSP协议。偶然间发现在这篇文章非常好，故转载学习使用：RTSP协议讲解 一、什么是RTSP协议? RTSP是一个实时传输流协...](#)
- [rtsp协议笔记](#) 2481
- [RTSP协议学习笔记_1 第一部分 RTSP协议_3 一、 RTSP协议概述_3 二、 RTSP协议与HTTP协议区别_3 三、 RTSP重要术语_4 1. ...](#)

“相关推荐”对你有帮助么？

👍 非常有帮助 👏 有帮助 😐 一般 😞 没帮助 😡 非常没帮助

©2022 CSDN 皮肤主题：大白 设计师：CSDN官方博客 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

黑面狐 关注

3 评论 0 收藏 63 专栏目录