**Rtsp协议小结**

**命令交互**

**OPTIONS**

**c -> s**

OPTIONS rtsp://192.168.31.115:8554/live RTSP/1.0\r\n

CSeq: 2\r\n

\r\n

**s -> c**

RTSP/1.0 200 OK\r\n

CSeq: 2\r\n

Public: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY\r\n

\r\n

**DESCRIBE**

**c -> s**

DESCRIBE rtsp://192.168.31.115:8554/live RTSP/1.0\r\n

CSeq: 3\r\n

Accept: application/sdp\r\n

\r\n

**s -> c**

RTSP/1.0 200 OK\r\n

CSeq: 3\r\n

Content-length: 146\r\n

Content-type: application/sdp\r\n

\r\n

v=0\r\n

o=- 91565340853 1 in IP4 192.168.31.115\r\n

t=0 0\r\n

a=contol:\*\r\n

m=video 0 RTP/AVP 96\r\n

a=rtpmap:96 H264/90000\r\n

a=framerate:25\r\n

a=control:track0\r\n

**v=0**

表示sdp的版本  
**o=- 91565340853 1 IN IP4 192.168.31.115**  
格式为 o=<用户名> <会话id> <会话版本> <网络类型><地址类型> <地址>  
用户名：-  
会话id：91565340853，表示rtsp://192.168.31.115:8554/live请求中的live这个会话  
会话版本：1  
网络类型：IN，表示internet  
地址类型：IP4，表示ipv4  
地址：192.168.31.115，表示服务器的地址

**m=video 0 RTP/AVP 96\r\n**

格式为 m=<媒体类型> <端口号> <传输协议> <媒体格式 >  
媒体类型：video

端口号：0，为什么是0？因为上面在SETUP过程会告知端口号，所以这里就不需要了

传输协议：RTP/AVP，表示RTP OVER UDP，如果是RTP/AVP/TCP，表示RTP OVER TCP

媒体格式：表示负载类型(payload type)，一般使用96表示H.264

**a=rtpmap:96 H264/90000**

格式为a=rtpmap:<媒体格式><编码格式>/<时钟频率>

**a=framerate:25**

表示帧率

**a=control:track0**

表示这路视频流在这个会话中的编号

**SETUP**

**c -> s**

SETUP rtsp://192.168.31.115:8554/live/track0 RTSP/1.0\r\n

CSeq: 4\r\n

Transport: RTP/AVP;unicast;client\_port=54492-54493\r\n

\r\n

客户端发送建立请求，请求建立连接会话，准备接收音视频数据

解析一下Transport: RTP/AVP;unicast;client\_port=54492-54493\r\n

**RTP/AVP**：表示RTP通过UDP发送，如果是**RTP/AVP/TCP**则表示RTP通过TCP发送

**unicast**：表示单播，如果是**multicast**则表示多播

**client\_port=54492-54493**：由于这里希望采用的是RTP OVER UDP，所以客户端发送了两个用于传输数据的端口，客户端已经将这两个端口绑定到两个udp套接字上，54492表示是RTP端口，54493表示RTCP端口(RTP端口为某个偶数，RTCP端口为RTP端口+1)

**s -> c**

RTSP/1.0 200 OK\r\n

CSeq: 4\r\n

Transport: RTP/AVP;unicast;client\_port=54492-54493;server\_port=56400-56401\r\n

Session: 66334873\r\n

\r\n

服务端接收到请求之后，得知客户端要求采用RTP OVER UDP发送数据，单播，客户端用于传输RTP数据的端口为54492，RTCP的端口为54493

服务器也有两个udp套接字，绑定好两个端口，一个用于传输RTP，一个用于传输RTCP，这里的端口号为56400-56401

之后客户端会使用54492-54493这两端口和服务器通过udp传输数据，服务器会使用56400-56401这两端口和这个客户端传输数据

**PLAY**

**c -> s**

PLAY rtsp://192.168.31.115:8554/live RTSP/1.0\r\n

CSeq: 5\r\n

Session: 66334873\r\n

Range: npt=0.000-\r\n

\r\n

**s -> c**

RTSP/1.0 200 OK\r\n

CSeq: 5\r\n

Range: npt=0.000-\r\n

Session: 66334873; timeout=60\r\n

\r\n

**TEARDOWN**

**c -> s**

TEARDOWN rtsp://192.168.31.115:8554/live RTSP/1.0\r\n

CSeq: 6\r\n

Session: 66334873\r\n

\r\n

**s -> c**

RTSP/1.0 200 OK\r\n

CSeq: 6\r\n

\r\n

**RTP**

**RTP头部**



**版本号(V)**：2Bit，用来标志使用RTP版本

​ **填充位§**：1Bit，如果该位置位，则该RTP包的尾部就包含填充的附加字节

​ **扩展位(X)**：1Bit，如果该位置位，则该RTP包的固定头部后面就跟着一个扩展头部

​ **CSRC技术器(CC)**：4Bit，含有固定头部后面跟着的CSRC的数据

​ **标记位(M)**：1Bit，该位的解释由配置文档来承担

​ **载荷类型(PT)**：7Bit，标识了RTP载荷的类型

​ **序列号(SN)**：16Bit，发送方在每发送完一个RTP包后就将该域的值增加1，可以由该域检测包的丢失及恢复

​ 包的序列。序列号的初始值是随机的

​ **时间戳**：32比特，记录了该包中数据的第一个字节的采样时刻

​ **同步源标识符(SSRC)**：32比特，同步源就是RTP包源的来源。在同一个RTP会话中不能有两个相同的SSRC值

​ **贡献源列表(CSRC List)**：0-15项，每项32比特，这个不常用

**RTP OVER TCP**

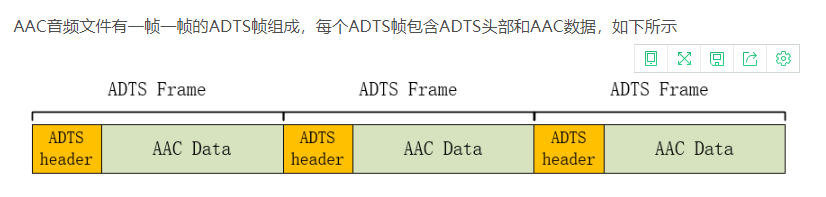
RTP默认是采用UDP发送的，格式为RTP头+RTP载荷，如果是使用TCP，那么需要在RTP头之前再加上四个字节

第一个字节：$，辨识符

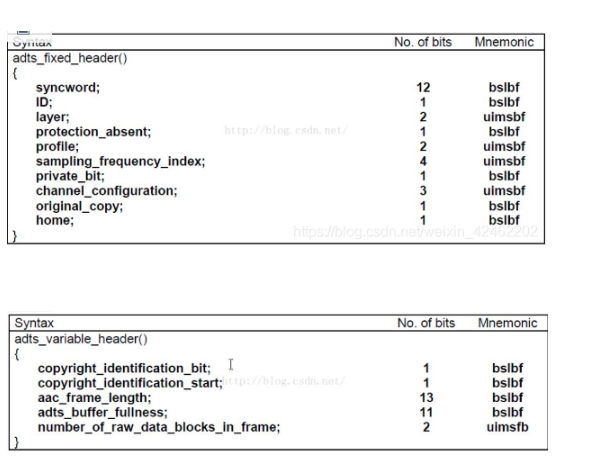
第二个字节：通道，在SETUP的过程中获取

第三第四个字节： RTP包的大小，最多只能12位，第三个字节保存高4位，第四个字节保存低8位

AAC格式



ADTS头部的大小通常为7个字节，包含着这一帧数据的信息，内容如下



各字段的意思如下

* syncword

总是0xFFF, 代表一个ADTS帧的开始, 用于同步.

* ID

MPEG Version: 0 for MPEG-4，1 for MPEG-2

* Layer

always: ‘00’

* protection\_absent

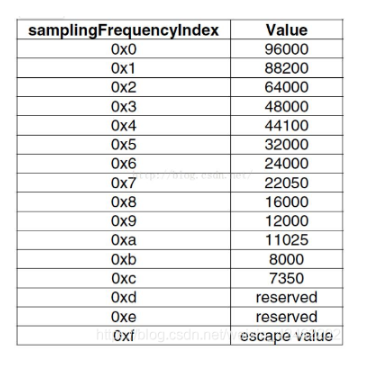
Warning, set to 1 if there is no CRC and 0 if there is CRC

* profile

表示使用哪个级别的AAC，如01 Low Complexity(LC) – AAC LC

* sampling\_frequency\_index

采样率的下标



* aac\_frame\_length

一个ADTS帧的长度包括ADTS头和AAC原始流

* adts\_buffer\_fullness

0x7FF 说明是码率可变的码流

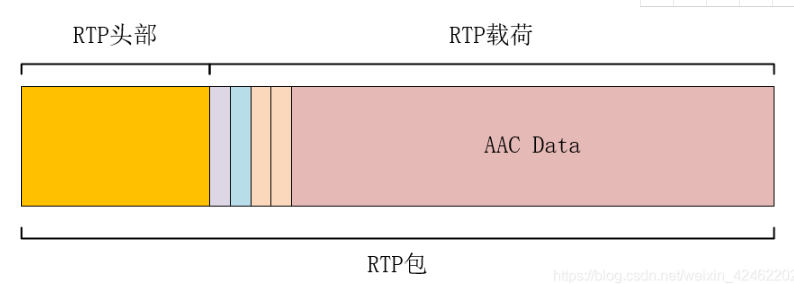
* number\_of\_raw\_data\_blocks\_in\_frame

表示ADTS帧中有number\_of\_raw\_data\_blocks\_in\_frame + 1个AAC原始帧

这里主要记住ADTS头部通常为7个字节，并且头部包含aac\_frame\_length，表示ADTS帧的大小

AAC的RTP打包方式就是将ADTS帧取出ADTS头部，取出AAC数据，每帧数据封装成一个RTP包

需要注意的是，并不是将AAC数据直接拷贝到RTP的载荷中。AAC封装成RTP包，在RTP载荷中的前四个字节是有特殊含义的，然后再是AAC数据，如下图所示



其中RTP载荷的一个字节为0x00，第二个字节为0x10

第三个字节和第四个字节保存AAC Data的大小，最多只能保存13bit，第三个字节保存数据大小的高八位，第四个字节的高5位保存数据大小的低5位

### 2.3 AAC RTP包的时间戳计算

假设音频的采样率位44100，即每秒钟采样44100次

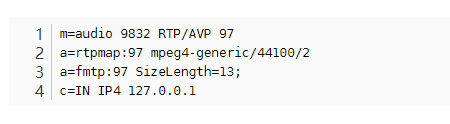
AAC一般将1024次采样编码成一帧，所以一秒就有44100/1024=43帧

RTP包发送的每一帧数据的时间增量为44100/43=1025

每一帧数据的时间间隔为1000/43=23m

## AAC的sdp媒体描述

下面给出AAC的媒体描述信息



这个一个媒体级的sdp描述，关于sdp文件描述详情可看[从零开始写一个RTSP服务器（一）不一样的RTSP协议讲解](https://blog.csdn.net/weixin_42462202/article/details/98986535)

* \*\*m=audio 9832 RTP/AVP 97 \*\*

格式为 m=<媒体类型> <端口号> <传输协议> <媒体格式 >  
媒体类型：audio，表示这是一个音频流

端口号：9832，**表示UDP发送的目的端口为9832**

传输协议：RTP/AVP，表示RTP OVER UDP，通过UDP发送RTP包

媒体格式：表示负载类型(payload type)，一般使用97表示AAC

* **a=rtpmap:97 mpeg4-generic/44100/2**

格式为a=rtpmap:<媒体格式><编码格式>/<时钟频率> /[channel]

mpeg4-generic表示编码，44100表示时钟频率，2表示双通道

* **c=IN IP4 127.0.0.1**

IN：表示internet

IP4：表示IPV4

127.0.0.1：**表示UDP发送的目的地址为127.0.0.1**

**特别注意**：这段sdp文件描述的udp发送的目的IP为127.0.0.1，目的端口为9832