1. RTP封装



**RTP头的结构体**

struct RtpHeader

{

/\* byte 0 \*/

uint8\_t csrcLen : 4;//CSRC计数器，占4位，指示CSRC 标识符的个数。

uint8\_t extension : 1;//占1位，如果X=1，则在RTP报头后跟有一个扩展报头。

uint8\_t padding : 1;//填充标志，占1位，如果P=1，则在该报文的尾部填充一个或多个额外的八位组，它们不是有效载荷的一部分。

uint8\_t version : 2;//RTP协议的版本号，占2位，当前协议版本号为2。

/\* byte 1 \*/

uint8\_t payloadType : 7;//有效载荷类型，占7位，用于说明RTP报文中有效载荷的类型，如GSM音频、JPEM图像等。

uint8\_t marker : 1;//标记，占1位，不同的有效载荷有不同的含义，对于视频，标记一帧的结束；对于音频，标记会话的开始。

/\* bytes 2,3 \*/

uint16\_t seq;//占16位，用于标识发送者所发送的RTP报文的序列号，每发送一个报文，序列号增1。接收者通过序列号来检测报文丢失情况，重新排序报文，恢复数据。

/\* bytes 4-7 \*/

uint32\_t timestamp;//占32位，时戳反映了该RTP报文的第一个八位组的采样时刻。接收者使用时戳来计算延迟和延迟抖动，并进行同步控制。

/\* bytes 8-11 \*/

uint32\_t ssrc;//占32位，用于标识同步信源。该标识符是随机选择的，参加同一视频会议的两个同步信源不能有相同的SSRC。

/\*标准的RTP Header 还可能存在 0-15个特约信源(CSRC)标识符

每个CSRC标识符占32位，可以有0～15个。每个CSRC标识了包含在该RTP报文有效载荷中的所有特约信源

\*/

};

**RTP包的结构体**

struct RtpPacket

{

struct RtpHeader rtpHeader;

uint8\_t payload[0];

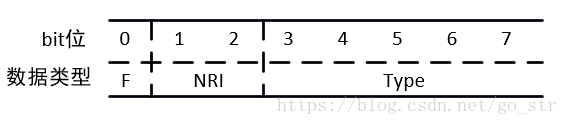
};

// 包含一个RTP头部和RTP载荷

1. H264码流进行RTP封装

**（1）**H.264由一个一个的NALU组成，每个NALU之间使用00 00 00 01或00 00 01分隔开，每个NALU的第一次字节都有特殊的含义,





1，F(forbiden):禁止位，占用NAL头的第一个位，当禁止位值为1时表示语法错误；

2，NRI:参考级别，占用NAL头的第二到第三个位；值越大，该NAL越重要。

3，Type:Nal单元数据类型，也就是标识该NAL单元的数据类型是哪种，占用NAL头的第四到第8个位；

常用Nalu\_type：

0x06 (0 00 00110) SEI type = 6

0x67 (0 11 00111) SPS type = 7

0x68 (0 11 01000) PPS type = 8

0x65 (0 11 00101) IDR type = 5

0x65 (0 10 00101) IDR type = 5

0x65 (0 01 00101) IDR type = 5

0x65 (0 00 00101) IDR type = 5

0x61 (0 11 00001) I帧 type = 1

0x41 (0 10 00001) P帧 type = 1

0x01 (0 00 00001) B帧 type = 1

对于H.264格式了解这些就够了，我们的目的是想从一个H.264的文件中将一个一个的NALU提取出来，然后封装成RTP包，下面介绍如何将NALU封装成RTP包。

**（2）** H.264可以由三种RTP打包方式

（2-1）单NALU打包： 一个RTP包包含一个完整的NALU

（2-2）聚合打包：对于较小的NALU，一个RTP包可包含多个完整的NALU

（2-3）分片打包：对于较大的NALU，一个NALU可以分为多个RTP包发送

注意：这里要区分好概念，每一个RTP包都包含一个RTP头部和RTP荷载，这是固定的。而H.264发送数据可支持三种RTP打包方式

比较常用的是单NALU打包和分片打包，这里只介绍两种

单NALU打包

所谓单NALU打包就是将一整个NALU的数据放入RTP包的载荷中，这是最简单的一种方式。

分片打包

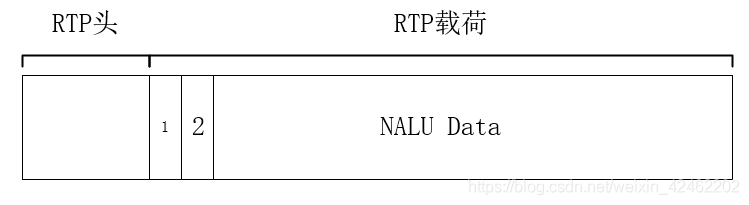
每个RTP包都有大小限制的，因为RTP一般都是使用UDP发送，UDP没有流量控制，所以要限制每一次发送的大小，所以如果一个NALU的太大，就需要分成多个RTP包发送，至于如何分成多个RTP包，如下：

首先要明确，RTP包的格式是绝不会变的，永远多是RTP头+RTP载荷

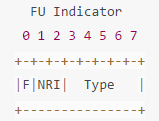


RTP头部是固定的，那么只能在RTP载荷中去添加额外信息来说明这个RTP包是表示同一个NALU

如果是分片打包的话，那么在RTP载荷开始有两个字节的信息，然后再是NALU的内容



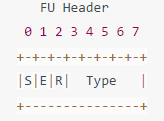
**第一个字节位FU Indicator，其格式如下**



高三位：与NALU第一个字节的高三位相同

Type：28，表示该RTP包一个分片，为什么是28？因为H.264的规范中定义的，此外还有许多其他Type，这里不详讲

**第二个字节位FU Header，其格式如下**



S：标记该分片打包的第一个RTP包

E：比较该分片打包的最后一个RTP包

Type：NALU的Type