



音视频协议-RTP协议打包

零声Github分享官

目录

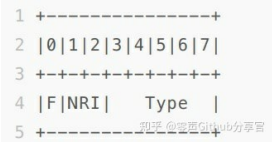
- 1. H264打包RTP的方法
- 2. RTP打包AAC

1. H264打包RTP的方法

RTP的特点不仅仅支持承载在UDP上，这样利于低延迟音视频数据的传输，另外一个特点是它允许通过其它协议接收端和发送端协商音视频数据的封装和编解码格式，这样固定头的payload type 字段就比较灵活。

H.264标准协议定义了两种不同的类型：一种是VCL即Video Coding Layer，一种是NAL即Network Abstraction Layer。其中前者就是编码器吐出来的原始编码数据，没有考虑传输和存储问题。后面这种就是为了展现H.264的网络亲和性，对VCL输出的slice片数据进行了封装为NALUs(NAL Units)，然后再封装为RTP包进行传输。

NALU的基本格式是：NALU Header + NALU Data,其中NALU的头由一个字节组成如下所示：



F (1 bit)	如果是环帧，则置1，其余H.264固定为0。
NRI (2 bit)	用来指示该NALU 的重要性等级。值越大，表示当前NALU越重要。具体大于0 时取何值，没有具体规定。 例如：如果是00，则表示此帧即使丢失了，也不影响解码；其他值则表示此帧如果丢失了，会影响解码。这个字段指明了该NALU的重要性，但是实际我们不太关心这个字段。
Nalu_Type (5 bit)	NAL Unit的类型。这个值指明了NALU的类型。其中NALU的类型见下表

Nalu_Type	NALU内容	备注
0	未指定	
1	非IDR图像编码的slice	比如普通I、P、B帧
2	编码slice数据划分A	2类型时，只传递片中最重要信息，如片头，片中宏块的预测模式等；一般不会用到；
3	编码slice数据划分B	3类型是只传输残差；一般不会用到；
4	编码slice数据划分C	4时则只可以传输残差中的AC系数；一般不会用到；
5	IDR图像中的编码slice	IDR帧，IDR一定是I帧但是I帧不一定是IDR帧。
6	SEI补充增强信息单元	可以存一些私有数据等；
7	SPS 序列参数集	编码的参数配置
8	PPS 图像参数集	编码的参数配置
9	接入单元定界符	
10	序列结束	
11	码流结束	
12	填充数据	
24	STAP-A Single-time aggregation packet	单一时间聚合包模式，意味着一个RTP包可以传输多个NALU，但是这些NALU的编码时间要一样才能聚合到一个RTP。
25	STAP-B Single-time aggregation packet	单一时间聚合包模式，比STAP-B多一个DON
26	MTAP 16 Muti-time aggregation packet	多个时间聚合包模式：意味着一个RTP包可以传输多个NALU，但是这些NALU的编码时间有可能不一样。
27	MTAP 24 Muti-time aggregation packet	多个时间的聚合包模式
28	FU-A Fragmentation unit	分包模式：当一个RTP容纳不下一个NALU时，就需要FUs这种格式。
29	FU-B Fragmention unit	分包模式
30-31	未指定，保留	

相关视频推荐：

音视频流媒体高级开发/FFmpeg/webRTC/rtmp/hls/rtsp/ff...
e⁹ www.bilibili.com/video/BV1NZ4y1m7dS/

【文章福利】：音视频面试题、学习资料、教学视频和学习路线图资料（资料包括C/C++，Linux，FFmpeg webRTC rtmp hls rtp ffmpeg srs等），qun994289133免费分享，有需要的可以加群领取哦！~学习交流裙994289133加入领取资料

- king-《tcpip详解卷一》：150行代码拉开协议栈实现的篇章-11.19.mp4
- FFmpeg进阶学习-播放器-录制-录屏-RTMP推流拉流.mp4
- dpdk如何支持千万级副并发，c10m的极限瓶颈在哪里?.mp4
- dpdk的前世今生，cc++程序员的未来方向.mp4
- darren-直播技术架构分析-让你的直播秒开-2.24.mp4
- darren-直播低延迟播放技术-变速实现-9.19.mp4
- darren-怎么更快速地学习FFmpeg-12.4.mp4
- darren-音视频难点-该怎么学习流媒体服务器-11.28.mp4
- Darren-音视频面试题-直播延迟分析和优化-12.4.mp4
- Darren-音视频面试烧不开的直播延迟分析-11.4.mp4
- Darren-音视频面试-流媒体服务器开发原理分析-rtmp-hls-httpflv.mp4
- darren-音视频面试必问的直播延迟分析-4.26.mp4
- darren-音视频面试必问-RTSPRTMP推流的各种坑分析.mp4
- darren-音视频面试必问-RTSP、RTMP推流的各种坑分析-3.6.mp4
- darren-音视频开发系列-音视频入门核心知识精讲-1.6.mp4
- darren-音视频开发系列-快速掌握音视频开发基础知识-12.6.mp4
- darren-音视频开发如何快速掌握FFmpeg-2.18.mp4
- darren-音视频开发进阶-图文并茂分析H264编码原理1.20.mp4
- darren-音视频开发基础和识别进阶剖析-3.9.mp4
- Darren-音视频开发第一个项目-播放器设计-10.29.mp4
- Darren-音视频开发第一个项目-ffmpeg播放器升变2.7.mp4

企鹅君羊994289133领取资料



一、

1-11就是NALU的单个包类型，但是一个NALU的大小是不一样的，如果是非视频数据的

SPS PPS才十几个字节，对于IDR帧，则有可能几十KB。

这样把NALU打包到RTP方式就很多，分为：

一个RTP包承载一个NALU；

多个NALU合并到一个RTP；

一个大的NALU切分成多个RTP。

二、

同时由于时间戳的问题，就有了24-29几种类型。

但是对于发送端RTP包的一方来说，尽可能找简单的打包方式。对于接受端则需要适配各种发送

端的打包方式，因为无法决定输入源的打包方式。

(打包的时候不要搞太复杂的模式)

我们对于NALU的长度<=1400 (rtp payload size) 的则采用的是单一NALU打包到单一RTP包中；

我们对于NALU的长度>1400的则采用了FU-A的方式进行了打包，这种就是把一个大的NALU进行了切分，最后接收方则进行了合并，把多个RTP包合并成一个完整的NALU即可；

为什么NALU的长度大于1400字节就要进行FU-A切片，是因为底层MTU大小值一般为1500，从传输效率讲，使用1400作为切分条件。

RTP最大数据包 (包含RTP头部) = MTU - UDP首部 - IP 报文首部

FU indication:
十六进制: 0x7C
二进制: 0111 1100
FU header:
十六进制: 0x85
二进制: 1000 0101
这里的SE是10, 则说明该RTP包承载的NALU的第一个切片。

1.2.4、第一个IDR帧的NALU第二个切片

Wireshark packet capture showing RTP payload for the second slice of the first IDR frame NALU. The packet list shows a sequence of RTP packets. The packet details pane shows the FU header with SE set to 10. The packet bytes pane shows the raw data with the SE field highlighted in red.

FU indication:
十六进制: 0x7C
二进制: 0111 1100
FU header:
十六进制: 0x05
二进制: 0000 0101
这里的SE是00, 则说明该RTP包承载的NALU的中间切片。

1.2.5、第一个IDR帧的NALU最后一个切片

Wireshark packet capture showing RTP payload for the last slice of the first IDR frame NALU. The packet list shows a sequence of RTP packets. The packet details pane shows the FU header with SE set to 11. The packet bytes pane shows the raw data with the SE field highlighted in red.

FU indication:
十六进制: 0x7C
二进制: 0111 1100
FU header:
十六进制: 0x45
二进制: 0100 0101
这里的SE是01, 则说明该RTP包承载的NALU的最后一个切片。

2、RTP打包AAC

过程:

1. 需要将aac的前7个(或9个)字节的ADTS去掉, 即是跳过adts header.
2. 添加RTP Header.
3. 添加AU_HEADER_LENGTH.
4. 添加AU_HEADER.
5. 添加AU (去掉ADTS的aac数据) 数据.

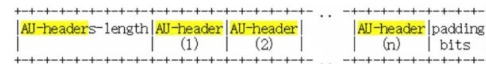


Figure 2: The AU Header Section

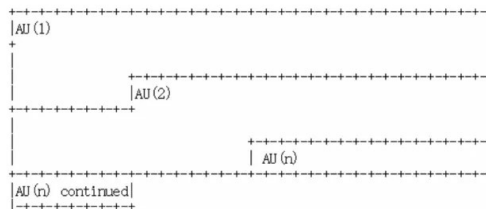


Figure 5: Access Unit Data Section; each AU is 1500 bytes

注意: 一个RTP包中可以有一个AU-header-length 和 n个AU-header和 n个AU (AU每包实际音频数据流)

5.1、AU-header-length

头两个字节表示au-header的长度, 单位是bit. 一个AU-header长度是两个字节 (16bit) 因为可以有多个

个au-header所以AU-header-length的值是 16的倍数, 一般音频都是单个音频数据流的发送, 所以

AU-header-length的值是16

//AU_HEADER_LENGTH

bytes[12] = 0x00; //高位

bytes[13] = 0x10; //低位 只有一个AU_HEADER

因为单位是bit, 除以8就是auHeader的字节长度; 又因为单个auheader字节长度2字节, 所以再除以2就

是auheader的个数。

(注意: AU-header长度并不是固定为2字节, 具体要看SDP)

5.2、AU-header

au-header的高13个bits就是一个au 的字节长度:

//AU_HEADER

bytes[14] = (byte)((len & 0x1fe0) >> 5); //高位

bytes[15] = (byte)((len & 0x1f) << 3); //低位

(注意: AU-header长度并不是固定为2字节, 具体要看SDP)

5.3、AU

音频实际数据 (去掉ADTS的aac数据)

5.4、RTSP/SDP中AAC相关配置

```
1 v=0
2 o=- 16128587303007558182 16128587303007558182 IN IP4 WINDOWS-751D
  U9Q
3 s=Unnamed
4 i=N/A
5 c=IN IP4 0.0.0.0
6 t=0 0
7 a=tool:vlc 3.0.5
8 a=recvonly
9 a=type:broadcast
10 a=charset:UTF-8
11 a=control:rtsp://192.168.2.195:8554/
12 m=audio 0 RTP/AVP 96
13 b=AS:128
14 b=RR:0
15 a=rtpmap:96 mpeg4-generic/22050
16 a=fmtp:96 streamtype=5; profile-level-id=15; mode=AAC-hbr; config
  =138856e500; sizeLength=13; indexLength=3; indexDeltaLength=3; Pr
  ofile=1;
17 a=control:rtsp://192.168.2.195:8554/trackID=4
18 m=video 0 RTP/AVP 96
19 b=AS:800
20 b=RR:0
21 a=rtpmap:96 H264/90000
22 a=fmtp:96 packetization-mode=1;profile-level-id=42c01e;sprop-para
  meter-sets=Z0LAhtCQKeX/8CgAJ/EAADAZAAAF2qPF16gA==,aM43IA==;
23 a=control:rtsp://192.168.2.195:8554/trackID=5
```

streamtype对于AAC, 固定为5。

profile-level-id固定为1。

config, SizeLength, IndexLength, IndexDeltaLength作用:

config是16进制的, 前两个字节 1388 , 表示采样率为22050, 1个channel。

前两个字节的为ios-14996-3中定义的AudioSpecificConfig, 前13个bits的格式为:

字段	比特数	说明
audioObjectType	5	aac的profile, 通常情况是1, 或者2
samplingFrequencyIndex	4	aac的采样频率的索引, 见下表
channelConfiguration	4	aac的通道数, 1-6表示为相应的通道数量, 7表 示非标准配置

samplingFrequencyIndex的取值:

值	表示的采样频率
0	96000,
1	88200,
2	64000,
3	48000,
4	44100,
5	32000,
6	24000,
7	22050,
8	16000,
9	12000,
10	11025,
11	8000,
12	7350
15	表示自定义的采样频率, 不会出现在标准中

1388 转换成2进制为 0001 0011 1000 1000

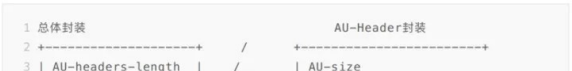
audioObjectType为 00010, 即 2

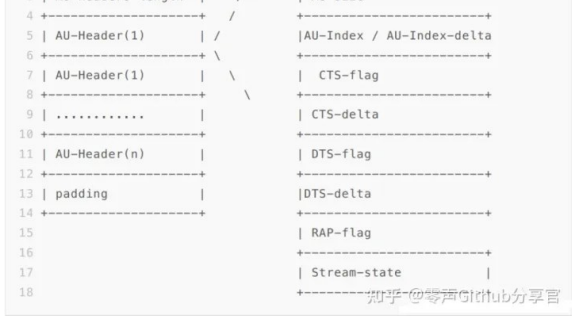
samplingFrequencyIndex为 0111, 即 7, 对应的采样频率为 22050

channelConfiguration为 0001, 表示channel数量为1。

sizeLength=13; indexLength=3; indexDeltaLength=3涉及到音频的AU Header。

5.5、AU-Header数据段的格式





字段	比特数	说明
AU-headers-length	0或16	如果没有AU-Header, 即fmtpp中不出现SizeLength等参数时, 此字段不存在. 否则, 使用两个字节来存储所有AU-Header所占用的比特数 (不包含自己, 不包含padding字段), 网络字节序
AU-size	由sizeLength决定	表示本段音频数据占用的字节数
AU-index	由indexLength决定	表示本段的序号, 通常0开始
AU-index-delta	由indexDeltaLength决定	表示本段的序号与上段序号的差值

其它的值都是可选的, 如果sdp中没有出现相关的参数(或者为0), 则表示它们不出现。

以最简单的情况举例, 假设aac数据长度为200字节, 只有一个au-header。

200 的二进制为 0000011001000。(补足为13 bits)

AU-headers-length 值为16, 因为只有一个au-header, au-header中

只有AU-size和AU-Index, 共占用16bits。

整个au-header数据段的内容为

0000 0000 0000 1000 0000011001000 000

如果一个rtsp中只有一个aac包, 不需要加AU-Header, 那么sdp中的aac参数可以简化为

a=fmtpp:96 streamtype=5; profile-level-id=1; mode=AAC-hbr; config=1 38856e500;

编辑于 2022-05-18 20:29

RTP Ffmpeg 音视频开发

写下你的评论...



发布



还没有评论, 发表第一个评论吧

推荐阅读

音视频通信协议--RTSP

Fengg... 发表于Media...

RTP协议封装音视频媒体数据详解

RTP协议对媒体数据(包括音频和视频)的封装是由指定的协议文档规定。1. RTP封装H.264视频编码数据 1.1 H.264 基本流的结构 H.264的基本流 (elementary stream,ES) 的结构分为两层, 包...

逆旅过客 发表于基于RTP...

音视频基础知识---协议相关 RTSP RTMP HLS

lucas 发表于移动音视频...

rtsp协议详解 流媒体传输协议

最详细的音视频流媒体传输协议 - rtsp协议详解

零声Github分享官

