

RTP封装h264



ai_believe

关注

❤️ 0.505

2017-07-18 09:49:59

字数 916

阅读 7,431

网络抽象层单元类型 (NALU):

NALU头由一个字节组成,它的语法如下:



F: 1个比特, forbidden_zero_bit. 在 H.264 规范中规定了这一位必须为 0.
NR: 2个比特, nal_ref_idc. 取00~11,似乎指示这个NALU的重要性.如00的NALU解码器可以丢弃它而不影响图像的回放.
Type: 5个比特, nal_unit_type. 这个NALU单元的类型 简述如下:



h264仅用1-23,24以后的用在RTP H264负载类型头中

不同类型的NALU的重要性指示如下表所示:

nal_unit_type	NAL 类型	nal_reference_bit
0	未使用	0
1	非I帧的片	此片属于参考帧，则不等于0，不属于参考帧，则等于0
2	片数据4分区	同上
3	片数据2分区	同上
4	片数据3分区	同上
5	IP帧的片	5
6	补充图像信息单元 (SEI)	0
7	序列参数集	非0
8	图像参数集	非0
9	分片数据	0
10	序列结束	0
11	码流结束	0
12	填充	0
13-23	保留	0
24-31	未定义	0

RTP 头的结构:



V: RTP协议的版本号, 占2bits. 当前协议版本号为2
P: 填充标志, 占1bit. 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.
X: 扩展标志, 占1bit. 如果X=1, 则在RTP报头后面有一个扩展报头
CC: CSRC计数器, 占4位. 指示CSRC 标识符的个数
M: 1bit. 标记解帧由设置定义. 目的在于允许重要事件在包流中标记出来. 如不同的有效载荷有不同的含义. 对于视频, 标记一帧的开始; 对于音频, 标记会话的开始.
负载类型 Payload type(P): 7bits
注: rfc里面对一些早期的格式定义了这个payload type, 但是后来的, 如h264并没有分配, 那就用96来代替. 因此现在96以上都不表示特定的格式. 具体表示什么要用sdp或者其他协议来协商.
序列号 Sequence number(SN): 16bits. 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. 序列号的初始值是随机产生的. 可以用于检查丢包以及进行数据包排序.
时间戳 Timestamp: 32bits. 必须使用90kHz时钟频率.
同步信源(SSRC)标识符: 32bits. 用于标识同步信源. 该标识符是随机随机产生的. 参加同一视频会议的两个同步信源不能有相同的SSRC.
特约信源(CSRC)标识符: 每个CSRC标识符占32bits. 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.

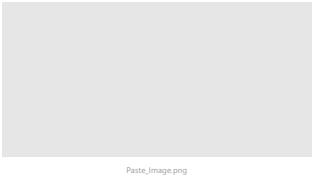
上面介绍了NALU和RTP header的基本结构. 下面介绍的全部都是RTP Payload的部分
Rtp负载第一个字节的结构如下,它和H264的NALU头结构一致, 可以把它认为是RTP h264负载类型字节, 完全是多增加的一个字节. 不影响后面的NALU结构



封包介绍:

单一NAL单元模式

对于 NALU 的长度小于 MTU 大小的包, 一般采用单一 NAL 单元模式. 对于一个原始的 H.264 NALU 单元常由 [Start Code] [NALU Header] [NALU Payload] 三部分组成. 其中 Start Code 用于标示这是一个 NALU 单元的开始, 必须是 "00 00 00 01" 或 "00 00 01". NALU 头仅一个字节, 其后都是 NALU 单元内容. 打包时去除 "00 00 01" 或 "00 00 01" 的开始码, 把其他数据打包的 RTP 包即可.



例: 如有一个 H.264 的 NALU 是这样的:
[00 00 00 01 67 42 A0 1E 23 56 0E 2F ...]



ai_believe

总用户 119

GCD中的dispatch_sync, dispatch_sync 分别... 组合执行小实验

阅读 276

Android主线程不能访问网络异常解决办法

阅读 597

热门文章

一介平民白手起家建立起黑金王朝

恋爱脑的我离婚后疯狂打脸渣男

怎么办？我的女友涉嫌谋杀

我死以后的一年，定要找出凶手

我被警察枪伤了老公！必须复仇

推荐阅读

网络层首部记录着超多信息

阅读 33

iOS音频AudioStreamBasicDescription 设置

阅读 253

09. 基本数据类型

阅读 217

Python编程练习18：Python语言串口操作入门

阅读 49

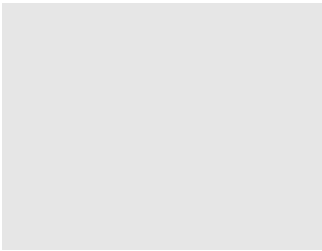
HTTP协议

阅读 45

这是一个序列参数集 NAL 单元 [00 00 00 01] 是四个字节的开始码, 67 是 NALU 头, 42 开始的数据是 NALU 内容.
封装成 RTP 包得如下:
[RTP Header] [67 42 A0 1E 23 56 0E 2F]
即只要去掉 4 个字节的开始码就可以了.

组合封装模式

其次, 当 NALU 的长度特别小时, 可以把几个 NALU 单元封在一个 RTP 包中.



这里只介绍STAP-A模式. 如果是STAP-8的话会多加入一个DON域. 另外还有MTAP16, MTAP24. 具体不介绍. 可以看rfc文档. 文章尾贴一个链接可以去看.
转载的话注明一下作者 :jwybobo2007 出处 :<http://blog.csdn.net/jwybobo2007/article/details/7054140>
例:

如有一个 H.264 的 NALU 是这样的:
[00 00 00 01 67 42 A0 1E 23 56 0E 2F ...]
[00 00 00 01 68 42 B0 12 58 6A D4 FF ...]
封装成 RTP 包得如下:
[RTP Header] [78 (STAP-A头, 占用1个字节) [第一个NALU长度 (占用两个字节)] [67 42 A0 1E 23 56 0E 2F] [第二个NALU长度 (占用两个字节)] [68 42 B0 12 58 6A D4 FF ...]

FU-A

当NALU的长度超过MTU时,就必须对NALU单元进行分片,也称为Fragmentation Units(FUs).
本荷载类型允许分片一个NAL单元到几个RTP包中. 下图 表示FU-A的RTP荷载格式. FU-A由1字节的分片单元指示, 1字节的分片单元头, 和分片单元荷载组成.

□

FU指示字节有以下格式:

□

FU指示字节的类型域的2, 29表示FU-A和FU-B. F的使用在5, 3描述. NRI域的值必须根据分片NAL单元的NRI域的值设置.

注意:这是第一个字节FU indicator, NRI为 优先级程度 00 可以去 , 11不能去. F一般设置0. 这一个字节用来表示当前包为分片FU-A包.

FU头的格式如下:

□

S: 1 bit
当设置成1, 开始位指示分片NAL单元的. 当跟随的FU荷载不是分片NAL单元荷载的开始, 开始位设为0.
E: 1 bit
当设置成1, 结束位指示分片NAL单元的. 即, 荷载的最后字节也是分片NAL单元的最后一个字节. 当跟随的FU荷载不是分片NAL单元的最后分片, 结束位设置为0.
R: 1 bit
保留位必须设置为0. 接收者必须忽略该位.
Type: 5 bits
NAL单元荷载类型定义在[1]的表7-1.

注意:这是第二个字节, 用来表示开始结束和NAL头的类型.

示例代码:

```
1 if (ilen > isize) { //超过MTU
2     const unsigned char s_e_r_start = 0x00;
3     const unsigned char s_e_r_mid = 0x00;
4     const unsigned char s_e_r_end = 0x00;
5     //0取NAL数据 1byte
6     unsigned char nalulType = *((unsigned char *) pcData) & 0x1f; //0取NALU的5bit 帧类型
7
8     unsigned char nal_ref_idc = *((unsigned char *) pcData) & 0x00; //0取NALU的2bit 帧重要程度 00 可以去 11不能去
9     //nal_ref_idc = 0x00;
10    //0取FU-A的1byte
11    unsigned char f_nri_type = nal_ref_idc + 28; //F为0 1bit, nri上面位取到2bit, 28为FU-A的分片类型5bit
12    unsigned char s_e_r_type = nalulType;
13    bool bFirst = true;
14    bool mark = false;
15    int offset = 1;
16    while (mark) {
17        if (ilen < nOffset + isize) { //是否拆分结束
18            isize = ilen - nOffset;
19            mark = true;
20            s_e_r_type = s_e_r_end + nalulType;
21        } else {
22            if (bFirst == true) {
23                s_e_r_type = s_e_r_start + nalulType;
24                bFirst = false;
25            } else {
26                s_e_r_type = s_e_r_mid + nalulType;
27            }
28        }
29        memcpy(mucSectionBuf, &f_nri_type, 1);
30        memcpy(mucSectionBuf + 1, &s_e_r_type, 1);
31        memcpy(mucSectionBuf + 2, (unsigned char *) pcData + nOffset, isize);
32        nOffset += isize;
33        makeH264Rtp(mucSectionBuf, isize + 2, mark, u1Stamp);
34    } else {
35        makeH264Rtp(pcData, ilen, true, u1Stamp);
36    }
```

9人点赞 > 网络编程 ...

更多精彩内容, 就在简书APP

"小礼物走一走, 来简书关注我"

赞赏支持

还没有人赞赏, 支持一下



al__believe

专注webrtc视频, H264/265, ffmpeg, rtmp等相关技术,
技术交流:1...

总资产119 共写了4.8W字 获得152个赞 共269个粉丝

关注

视频编码(H264概述)

视频压缩编码的目标1)保证压缩比例2)保证恢复的质量3)易实现、低成本、可靠性 压缩的出发点(可行性1))时间相关...

 rogenu1228
 阅读 3,775
 评论 0
 赞 11

基于ffmpeg的rtsp拉流代码分析流水账

概念 RTP: Real-time Transport Protocol实时传输协议。一般用于多媒体数据的传输。...

 糖后张
 阅读 12,028
 评论 1
 赞 10

H264中的sps pps

使用RTP传输H264的时候,需要用到sdp协议描述,其中有两项:Sequence Parameter Sets ...

 rogenu1228
 阅读 3,419
 评论 0
 赞 8

原来爱上你(十九):天作之合

原来爱上你目录上一章:原来爱上你(十八)::爱的苦楚竟“小白,走了吗?”康伟走到许小白身旁。“马上就好了。”许...

 沐阿洋
 阅读 365
 评论 0
 赞 0

YC创业课第四课:创建产品,聆听用户增长

一、主要内容回顾: 1.如何创建产品 1)从现实中需要解决的问题出发,然后选择一个行业。需求和阿选择? 创业者要...

 成长践行
 阅读 158
 评论 0
 赞 0



2015.8.5

西游记与企业管理的研究至今还在延续着,我也很有投入感地参与其中。但冷静下来想想,娱乐性与严谨性的融合好像已经成为我...

 王家坝,尉
 阅读 112
 评论 0
 赞 1

写下你的评论...

评论0
 赞9
 ...