

音视频 H264流媒体协议解析 RTP格式---用于网络发送

Linux百里

赞同 2

2 人赞同了该文章

RTP封装= 12字节固定RTP包头 + 载荷 (NALU)

补充：针对IP网络的RTP打包方式。为原始的NAL打包格式，就是开始的若干字节（1，2，4字节）是NAL的长度，而不是start_code,此时必须借助某个全局的数据来获得编码器的profile,level,PPS,SPS等信息才可以解码。（AVCC格式）？？？

（一）RTP包头

前12字节固定 + (0~15)个32位的CSRC标识符

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7																
V	P	X			CC		M																																								
时间戳																序号																															
同步源标识符(SSRC identifier)																																															
贡献源标识符 (CSRC identifiers)																																															
.....																																															

V (2bits): RTP协议的版本号, 当前协议版本号为2.
P (1bit): 填充标志, 如果设置填充位P=1, 在包尾将包含附加填充字节, 它不属于有效载荷. 填充的最后一个八进制包含应该忽略的八进制计数. 某些加密算法需要固定大小的填充字节, 或为在底层协议数据单元中携带几个RTP包.
X (1bit): 扩展标志, 如果X=1, 则在RTP报头后跟有一个扩展报头
CC(4bits): CSRC计数器, 指示CSRC 标识符的个数.

M (1bit): 标记位 (不同载荷含义不同, 视频标记一帧的最后一个分片slice则=1,其他=0)
PT (7bits): 载荷类型RTP_PAYLOAD_RTSP,记录后面资料使用哪种 Codec , receiver 端找出相应的 decoder 解码出来。例如H264=96
序列号(16bits): 用于标识发送者所发送的 RTP 报文的序列号 (初始值随机) , 每发送一个报文, 序号增加 1

时间戳(32bits): 时间戳反映了该 RTP 报文的第一个八位组的采样时刻。接受者使用时间戳来计算延迟和抖动, 并进行同步控制。
SSRC(32bits): 区分是在和谁通信。值随机选择, 参加同一视频会议的两个同步信源的SSRC要相同。

贡献源(CSRC)标识符(32bits): 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 协议实际上是由实时传输协议RTP (Real-time Transport Protocol) 和实时传输控制协议RTCP (Real-time Transport Control Protocol) 两部分组成。

RTP 协议基于多播或单播网络为用户提供连续媒体数据的实时传输服务;

RTCP 协议是 RTP 协议的控制部分, 用于实时监控数据传输质量, 为系统提供拥塞控制和流控制。

（二）回顾NALU类型

nal_unit(NumBytesInNALunit) {	C	Descriptor
forbidden_zero_bit	All	r(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)

F: 1 个比特

forbidden_zero_bit 在 H.264 规范中规定了这一位必须为 0.

NRI: 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. 不过一般情况下不太关心这个属性

Type: 5 个比特

nal_unit_type. 这个 NALU 单元的类型. 简述如下:

- 0 没有定义
- 1-23 NAL单元 单个 NAL 单元包.
- 24 STAP-A 单一时间的组合包
- 25 STAP-B 单一时间的组合包
- 26 MTAP16 多个时间的组合包
- 27 MTAP24 多个时间的组合包
- 28 FU-A 分片的单元
- 29 FU-B 分片的单元
- 30-31 没有定义

（三）打包模式： 拆包（1种） or 不拆包（2种）

RTP单次发送有上限 2种RTP打包：拆包or不拆包

在IP网络中, 当要传输的IP报文大小超过【最大传输单元MTU】时就会产生IP分片情况。（若交给底层协议拆包容易出问题→→→主动拆分NALU再打包成RTP包后发送）

1.SDP文件描述和封包的关联

H264的RTP中有三种不同的封装模式 (Single NAL,Non-interleaved,Interleaved) 通过SDP参数中指定, 如:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; packetization-mode=1; sprop-parameter-sets=Z0IACpZ
```

1、packetization-mode决定封包模式：

```
当 packetization-mode 的值为 0 时或不存在时，必须使用单一 NALU 单元模式。(无此字段时，缺省为0)
当 packetization-mode 的值为 1 时必须使用非交错(non-interleaved)封包模式。 FU-A
当 packetization-mode 的值为 2 时必须使用交错(interleaved)封包模式。 FU-B
```

2、sprop-parameter-sets: SPS,PPS

```
这个参数可以用于传输 H.264 的序列参数集和图像参数 NAL 单元。这个参数的值采用 Base64 进行编码。
//若不用Base64则可能会有数据丢失
```

3、profile-level-id:

这个参数用于指示 H.264 流的 profile 类型和级别. 由 Base16(十六进制) 表示的 3 个字节.
第一个字节表示 H.264 的 Profile 类型,
第三个字节表示 H.264 的 Profile 级别

每个打包方式允许的NAL单元类型总结(yes = 允许, no = 不允许, ig = 忽略)

Type	Packet Unit Mode	Single NAL Mode	Non-Interleaved Mode	Interleaved Mode
0	undefined	ig	ig	ig
1-23	NAL unit	yes	yes	no
24	STAP-A	no	yes	no
25	STAP-B	no	no	yes
26	MTAP16	no	no	yes
27	MTAP24	no	no	yes
28	FU-A	no	yes	yes
29	FU-B	no	no	yes
30-31	undefined	ig	ig	ig

知乎 @Linux百里

```
@single NAL unit packet 单包(1个RTP包:1个NALU)
@aggregation packets 聚合(组合)包(1个RTP包:多个NALU. 提高传输效率). 需要解包时在重组。
  @STAP (Single-time aggregation packet)
    STAP-A
    STAP-B
  @ MTAP (Multi-time aggregation packet)
    MTAP16
    MTAP24
@Fragmentation Unit 拆包处理【一个NALU->多包 NALU>最大传输单元MTU】
  FU-A //非交错模式
  FU-B //交错模式
```

2.单一NALU的RTP包

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 00 01" 的开始码, 把其他数据封包的 RTP 包即可.

如有一个 H.264 的 NALU 是这样的:

[00 00 00 01 67 42 A0 1E 23 56 0E 2F ...] 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 包将如下:

[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.

3.组合NALU的RTP包

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-B的话会多加入一个DON域, 另外还有MTAP16、MTAP24。

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 包将如下:

[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 ...]

4.分片NALU的RTP包：FU_indicator和FU_head （RTP分包时的包头）

当NALU的长度超过MTU时,就必须对NALU单元进行分片封包,也称为Fragmentation Units(FUs).

FU-A的分片格式：数据比较大的H264视频包，被RTP分片发送。12字节的RTP头后面跟随的就是FU-A分片

FU_indicator:

F 禁止位

NRI 重要标识位 即拆包的nalu自身的NRI ---- F与NRI 保存了 NALU的前3位

type RTP打包头类型, FU-A时type=28

FU_header:

S 开始位 1表示分片NAL单元的开始, 反之=0

E 结束位 1表示分片NAL单元的结束, 反之=0.

R 保留位 必须为0, 接收者必须忽略该位.

type NALU数据类型 NALU_header ---- 保存了 NALU 类型（即NALU的后5位）

例:

0x7C85=011111100 10000101 (开始包)

0x7C05=011111100 00000101 (中间包)

0x7C45=011111100 01000101 (结束包)

如有一个 H.264 的 NALU 是这样的:

[00 00 00 01 65 42 A0 1E 23 56 0E 2F ... 02 17 C8 FD F1 B9 C7 53 59 72 ... C8 FF FF F4 1A D5 C4 18 A8 ... F1 B9 C7 1D A5 FA 13 0B ...]

封装成 RTP 包将如下 (注意: 下面去掉了开始码和NALU头部, 但是在FU_header的type存放了NALU数据类型)

```
[ RTP Header ] [ 7C 85 42 A0 1E 23 56 0E 2F ... ]
[ RTP Header ] [ 7C 05 02 17 C8 FD F1 B9 C7 53 59 72 ... ]
[ RTP Header ] [ 7C 05 CB FF FF F4 1A D5 C4 18 A8 ... ]
[ RTP Header ] [ 7C 45 F1 B9 C7 1D A5 FA 13 0B ... ]
```

拆包和解包:

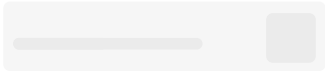
发送端—拆包: NAL_header与分片后的FU的单元头有如下关系:

NAL_header前三位为FU_indicator的前三位

NAL_header后五位为FU_header的后五位

接收端—解包: 将所有分片包组合还原成原始的NAL包

nal_unit_type = (fu_indicator & 0xe0) | (fu_header & 0x1f)



FFmpeg/WebRTC/RTMP/NDK/Android音视频流媒体高级开发 学习资料、教学视频和学习路线图 有需要的可以自行添加学习交流群

发布于 2022-03-01 17:37

网络发展 流媒体 RTP

写下你的评论...

🗨️ 🗨️ 🗨️

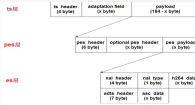


发布



还没有评论，发表第一个评论吧

推荐阅读



音视频流媒体开发系列 (12)
hls之m3u8、ts、h264、...

音视频流媒... 发表于音视频开发



【科普帖】音频格式大全

华志健au... 发表于音频砖家



音视频基础知识---封装格式

lucas 发表于移动音视频...



音视频编解码--M3U8文件格式

Fenng... 发表于Media...



× 登录即可查看 超5亿 专业优质内容
超 5 千万创作者的优质提问、专业回答、深度文章和精彩视频尽在知乎。
立即登录/注册

▲ 赞同 2 ▼ ● 添加评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...

