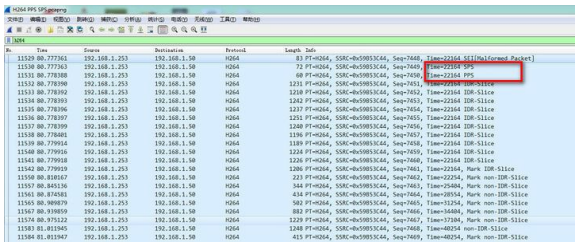
H264码流中SPS PPS详解

DaveBobo
关注互联网发展

- 1 SPS和PPS从何处而来?
- 2 SPS和PPS中的每个参数起什么作用?
- 3 如何解析SDP中包含的H.264的SPS和PPS串?

1 客户端抓包

在做客户端视频解码时，一般都会使用Wireshark抓包工具对接收的H264码流进行分析，如下所示:



在这里我们可以看到对解码视频起关键作用的SPS和PPS。

双击SPS内容如下:

```
H.264
  NAL unit header or first byte of the payload
  H264 NAL Unit Payload
      0100 0010 = Profile_idc: Baseline profile (66)
      1... .... = Constraint_set0_flag: 1
      .0.. .... = Constraint_set1_flag: 0
      ..0. .... = Constraint_set2_flag: 0
      ...0 .... = Constraint_set3_flag: 0
      .... 0... = Constraint_set4_flag: 0
      .... .0.. = Constraint_set5_flag: 0
      .... ..00 = Reserved_zero_2bits: 0
      0001 0101 = Level_id: 21 [Level 2.1 4 Mb/s]
      1... .... = seq_parameter_set_id: 0
      .001 01.. = log2_max_frame_num_minus4: 4
      .... ..1. = pic_order_cnt_type: 0
      .... ...0  0101 .... = log2_max_pic_order_cnt_lsb_minus4: 4
      .... 010. = num_ref_frames: 1
      .... ...0 = gaps_in_frame_num_value_allowed_flag: 0
      0000 1010  0... .... = pic_width_in_mbs_minus1: 19
      .000 1111 = pic_height_in_map_units_minus1: 14
      1... .... = frame_mbs_only_flag: 1
      .1.. .... = direct_8x8_inference_flag: 1
      ..0. .... = frame_cropping_flag: 0
      ...1 .... = vui_parameters_present_flag: 1
      .... 0... = aspect_ratio_info_present_flag: 0
      .... .0.. = overscan_info_present_flag: 0
      .... ..0. = video_signal_type_present_flag: 0
      .... ...0 = chroma_loc_info_present_flag: 0
      1... .... = timing_info_present_flag: 1
      .000 0000  0000 0000  0001 0011  1000 1000  0... .... = num_units_in_tick: 10000
      .000 0000  0000 0010  0100 1001  1111 0000  0... .... = time_scale: 300000
      .1.. .... = fixed_frame_rate_flag: 1
      ..0. .... = nal_hrd_parameters_present_flag: 0
      ...0 .... = vcl_hrd_parameters_present_flag: 0
      .... 0... = pic_struct_present_flag: 0
      .... .0.. = bitstream_restriction_flag: 0
      .... ..1. = rbsp_stop_bit: 1
      .... ...0 = rbsp_trailing_bits: 0
```

双击PPS内容如下:

```
H.264
  NAL unit header or first byte of the payload
  H264 NAL Unit Payload
      1... .... = pic_parameter_set_id: 0
      .1.. .... = seq_parameter_set_id: 0
      ..0. .... = entropy_coding_mode_flag: 0
      ...0 .... = pic_order_present_flag: 0
      .... 1... = num_slice_groups_minus1: 0
      .... .1.. = num_ref_idx_l0_active_minus1: 0
      .... ..1. = num_ref_idx_l1_active_minus1: 0
      .... ...0 = weighted_pred_flag: 0
      00.. .... = weighted_bipred_idc: 0
      ..00 0110  0... .... = pic_init_qp_minus26(se(v)): -6
      .000 1100 = pic_init_qs_minus26(se(v)): -6
      1... .... = chroma_qp_index_offset: 0
      .0.. .... = deblocking_filter_control_present_flag: 0
      ..0. .... = constrained_intra_pred_flag: 0
      ...0 .... = redundant_pic_cnt_present_flag: 0
      .... 1... = rbsp_stop_bit: 1
      .... .000 = rbsp_trailing_bits: 0
```

那么从上面的sps中我们知道图像的宽，高。

宽=（19+1）*16=320

高=（14+1）*16=240

为什么？参考下面

2 SPS PPS详解

2.1 SPS语法元素及其含义

在H.264标准协议中规定了多种不同的NAL
Unit类型，其中类型7表示该NAL Unit内保存的数据为Sequence Paramater
Set。在H.264的各种语法元素中，SPS中的信息至关重要。如果其中的数据丢失或出现错误，那么

解码过程很可能会失败。SPS及后续将要讲述的图像参数集PPS在某些平台的视频处理框架（比如iOS的VideoToolBox等）还通常作为解码器实例的初始化信息使用。

SPS即Sequence Paramater Set，又称作序列参数集。SPS中保存了一组编码视频序列(Coded video
sequence)的全局参数。所谓的编码视频序列即原始视频的一帧一帧的像素数据经过编码之后的结构组成的序列。而每一帧的编码后数据所依赖的参数保存于图像参数集中。一般情况SPS和PPS的NAL
Unit通常位于整个码流的起始位置。但在某些特殊情况下，在码流中间也可能出现这两种结构，主要原因可能为：

- 解码器需要在码流中间开始解码；
- 编码器在编码的过程中改变了码流的参数（如图像分辨率等）；

在做视频播放器时，为了让后续的解码过程可以使用SPS中包含的参数，必须对其中的数据进行解析。其中H.264标准协议中规定的SPS格式位于文档的7.3.2.1.1部分，如下图所示：

seq_parameter_set_data( ) {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
constraint_set4_flag	0	u(1)
constraint_set5_flag	0	u(1)
reserved_zero_2bits /* equal to 0 */	0	u(2)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 || profile_idc == 110 || profile_idc == 122 || profile_idc == 244 || profile_idc == 44 || profile_idc == 83 || profile_idc == 86 || profile_idc == 118 || profile_idc == 128) {		
chroma_format_idc	0	ue(v)
if(chroma_format_idc == 3)		
separate_colour_plane_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)
qpprime_y_zero_transform_bypass_flag	0	u(1)
seq_scaling_matrix_present_flag	0	u(1)
if(seq_scaling_matrix_present_flag)		
for(i = 0; i < (chroma_format_idc != 3) ? 8 : 12); i++) {		
seq_scaling_list_present_flag[i]	0	u(1)
if(seq_scaling_list_present_flag[i])		
if(i < 6)		
scaling_list(ScalingList4x4[i], 16, UseDefaultScalingMatrix4x4Flag[i])	0	
else		
scaling_list(ScalingList8x8[i − 6], 64, UseDefaultScalingMatrix8x8Flag[i − 6])	0	
}		
}		
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if(pic_order_cnt_type == 0)		
log2_max_pic_order_cnt_lsb_minus4	0	ue(v)
else if(pic_order_cnt_type == 1) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
offset_for_top_to_bottom_field	0	se(v)
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)		
offset_for_ref_frame[i]	0	se(v)
}		
max_num_ref_frames	0	ue(v)
gaps_in_frame_num_value_allowed_flag	0	u(1)
pic_width_in_mbs_minus1	0	ue(v)
pic_height_in_map_units_minus1	0	ue(v)
frame_mbs_only_flag	0	u(1)
if(!frame_mbs_only_flag)		
mb_adaptive_frame_field_flag	0	u(1)
direct_8x8_inference_flag	0	u(1)
frame_cropping_flag	0	u(1)
if(frame_cropping_flag) {		
frame_crop_left_offset	0	ue(v)
frame_crop_right_offset	0	ue(v)
frame_crop_top_offset	0	ue(v)
frame_crop_bottom_offset	0	ue(v)
}		
vui_parameters_present_flag	0	u(1)
if(vui_parameters_present_flag)		
vui_parameters( )	0	
}		

其中的每一个语法元素及其含义如下：

(1) profile_idc：

标识当前H.264码流的profile。我们知道，H.264中定义了三种常用的档次profile：

基准档次：baseline profile;

主要档次：main profile;

扩展档次：extended profile;

在H.264的SPS中，第一个字节表示profile_idc，根据profile_idc的值可以确定码流符合哪一种档次。判断规律为：

profile_idc = 66 → baseline profile;

profile_idc = 77 → main profile;

profile_idc = 88 → extended profile;

在新版的标准中，还包括了High、High 10、High 4:2:2、High 4:4:4、High 10 Intra、High 4:2:2 Intra、High 4:4:4 Intra、CAVLC 4:4:4 Intra等，每一种都由不同的profile_idc表示。

另外，constraint_set0_flag ~ constraint_set5_flag是在编码的档次方面对码流增加的其他一些额外限制性条件。

在我们实验码流中，profile_idc = 0x42 = 66，因此码流的档次为baseline profile。

(2) level_idc

标识当前码流的Level。编码的Level定义了某种条件下的最大视频分辨率、最大视频帧率等参数，

码流所遵从的level由level_idc指定。

当前码流中，level_idc = 0x1e = 30，因此码流的级别为3。

(3) seq_parameter_set_id

表示当前的序列参数集的id。通过该id值，图像参数集pps可以引用其代表的sps中的参数。

(4) log2_max_frame_num_minus4

用于计算MaxFrameNum的值。计算公式为MaxFrameNum = 2^(log2_max_frame_num_minus4 + 4)。MaxFrameNum是frame_num的上限值，frame_num是图像序号的一种表示方法，在帧间编码中常用作一种参考帧标记的手段。

(5) pic_order_cnt_type

表示解码picture order count(POC)的方法。POC是另一种计量图像序号的方式，与frame_num有着不同的计算方法。该语法元素的取值为0、1或2。

(6) log2_max_pic_order_cnt_lsb_minus4

用于计算MaxPicOrderCntLsb的值，该值表示POC的上限。计算方法为MaxPicOrderCntLsb = 2^(log2_max_pic_order_cnt_lsb_minus4 + 4)。

(7) max_num_ref_frames

用于表示参考帧的最大数目。

(8) gaps_in_frame_num_value_allowed_flag

标识位，说明frame_num中是否允许不连续的值。

(9) pic_width_in_mbs_minus1

用于计算图像的宽度。单位为宏块个数，因此图像的实际宽度为:

frame_width = 16 × (pic_width_in_mbs_minus1 + 1);

(10) pic_height_in_map_units_minus1

使用PicHeightInMapUnits来度量视频中一帧图像的高度。PicHeightInMapUnits并非图像明确的以像素或宏块为单位的高度，而需要考虑该宏块是帧编码或场编码。PicHeightInMapUnits的计算方式为：

PicHeightInMapUnits = pic_height_in_map_units_minus1 + 1;

(11) frame_mbs_only_flag

标识位，说明宏块的编码方式。当该标识位为0时，宏块可能为帧编码或场编码；该标识位为1时，所有宏块都采用帧编码。根据该标识位取值不同，PicHeightInMapUnits的含义也不同，为0时表示一场数据按宏块计算的高度，为1时表示一帧数据按宏块计算的高度。

按照宏块计算的图像实际高度FrameHeightInMbs的计算方法为：

FrameHeightInMbs = (2 − frame_mbs_only_flag) * PicHeightInMapUnits

(12) mb_adaptive_frame_field_flag

标识位，说明是否采用了宏块级的帧场自适应编码。当该标识位为0时，不存在帧编码和场编码之间的切换；当标识位为1时，宏块可能在帧编码和场编码模式之间进行选择。

(13) direct_8x8_inference_flag

标识位，用于B_Skip、B_Direct模式运动矢量的推导计算。

(14) frame_cropping_flag

标识位，说明是否需要对输出的图像帧进行裁剪。

(15) vui_parameters_present_flag

标识位，说明SPS中是否存在VUI信息。

2.2 PPS语法元素及其含义

除了序列参数集SPS之外，H.264中另一重要的参数集合为图像参数集Picture Paramater Set(PPS)。通常情况下，PPS类似于SPS，在H.264的裸码流中单独保存在一个NAL Unit中，只是PPS NAL Unit的nal_unit_type值为8；而在封装格式中，PPS通常与SPS一起，保存在视频文件的文件头中。

在H.264的协议文档中，PPS的结构定义在7.3.2.2节中，具体的结构如下表所示：

其中的每一个语法元素及其含义如下：

(1) pic_parameter_set_id

表示当前PPS的id。某个PPS在码流中会被相应的slice引用，slice引用PPS的方式就是在Slice header中保存PPS的id值。该值的取值范围为[0,255]。

(2) seq_parameter_set_id

表示当前PPS所引用的激活的SPS的id。通过这种方式，PPS中也可以取到对应SPS中的参数。该值的取值范围为[0,31]。

(3) entropy_coding_mode_flag

熵编码模式标识，该标识位表示码流中熵编码/解码选择的算法。对于部分语法元素，在不同的编码配置下，选择的熵编码方式不同。例如在一个宏块语法元素中，宏块类型mb_type的语法元素描述符为“ue(v)
| ae(v)”，在baseline profile等设置下采用指数哥伦布编码，在main profile等设置下采用CABAC编码。

标识位entropy_coding_mode_flag的作用就是控制这种算法选择。当该值为0时，选择左边的算法，通常为指数哥伦布编码或者CAVLC；当该值为1时，选择右边的算法，通常为CABAC。

(4) bottom_field_pic_order_in_frame_present_flag

标识位，用于表示另外条带头中的两个语法元素delta_pic_order_cnt_bottom和delta_pic_order_cn是否存在的标识。这两个语法元素表示了某一帧的底场的POC的计算方法。

(5) num_slice_groups_minus1

表示某一帧中slice group的个数。当该值为0时，一帧中所有的slice都属于一个slice group。slice group是一帧中宏块的组合方式，定义在协议文档的3.141部分。

(6) num_ref_idx_l0_default_active_minus1、num_ref_idx_l0_default_active_minus1

表示当Slice Header中的num_ref_idx_active_override_flag标识位为0时，P/SP/B slice的语法元素num_ref_idx_l0_active_minus1和num_ref_idx_l1_active_minus1的默认值。

(7) weighted_pred_flag

标识位，表示在P/SP slice中是否开启加权预测。

(8) weighted_bipred_idc

表示在B Slice中加权预测的方法，取值范围为[0,2]。0表示默认加权预测，1表示显式加权预测，2表示隐式加权预测。

(9) pic_init_qp_minus26和pic_init_qs_minus26

表示初始的量化参数。实际的量化参数由该参数、slice header中的slice_qp_delta/slice_qs_delta计算得到。

(10) chroma_qp_index_offset

用于计算色度分量的量化参数，取值范围为[-12,12]。

(11) deblocking_filter_control_present_flag

标识位，用于表示Slice header中是否存在用于去块滤波器控制的信息。当该标志位为1时，slice header中包含去块滤波相应的信息；当该标识位为0时，slice header中没有相应的信息。

(12) constrained_intra_pred_flag

若该标识为1，表示I宏块在进行帧内预测时只能使用来自I和SI类型宏块的信息；若该标识位0，表示I宏块可以使用来自Inter类型宏块的信息。

(13) redundant_pic_cnt_present_flag

标识位，用于表示Slice header中是否存在redundant_pic_cnt语法元素。当该标志位为1时，slice header中包含redundant_pic_cnt；当该标识位为0时，slice header中没有相应的信息。

3 解析SDP中包含的H.264的SPS和PPS串

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 (SPS)
和Picture Parameter Set
(PPS)需要用到,那么这两项从哪里获取呢?答案是从H264码流中获取.在H264码流中,都是以"0x00
0x00 0x01"或者"0x00
0x00 0x00 0x01"为开始码的,找到开始码之后,使用开始码之后的第一个字节的低5位判断是否为
7(sps)或者8(pps),
及data[4] & 0x1f == 7 ||
data[4] & 0x1f ==
8.然后对获取的nal去掉开始码之后进行base64编码,得到的信息就可以用于sdp.sps和pps需要用逗号分隔开来.

SDP中的H.264的SPS和PPS串，包含了初始化H.264解码器所需要的信息参数，包括编码所用的profile，level，图像的宽和高，deblock滤波器等。

由于SDP中的SPS和PPS都是BASE64编码形式的，不容易理解，有一个工具软件可以对SDP中的SPS和PPS进行解析，下载地址：download.csdn.net/downl...。

用法是在命令行中输入：

spsparser sps.txt pps.txt output.txt

例如sps.txt中的内容为：

Z0LgFNoFglE=

pps.txt中的内容为：

aM4wpIA=

最终解析得到的结果为：

这里需要特别提一下这两个参数

pic_width_in_mbs_minus1 = 21

pic_height_in_mbs_minus1 = 17

分别表示图像的宽和高，以宏块（16x16）为单位的值减1

因此，实际的宽为 (21+1)*16 = 352 高为 (17+1)*16 = 288

到这里应该知道第一部分客户端抓包计算图像宽高遗留下来的问题了吧。

Reference:

cnblogs.com/lidabo/p/65...

blog.csdn.net/heanyu/ar...

blog.csdn.net/shaqoneal...

blog.csdn.net/shaqoneal...

编辑于 2020-09-17 08:43

H.264 (MPEG-4 高级视频编码)

写下你的评论...

14 条评论

默认

最新

遥知不是雪

很好的文章，收获很多:-)，虽然3处parameter写成了paramater，但是问题不大，十分感谢。

2019-09-14

回复 2

刘彬

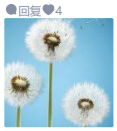
最近在做rtmp分离h264视频流，用的librtmp。分离出来FLV数据第一帧是包头，第二帧就是I帧

后面就是P帧了，没有收到SPS\PPS，请问这个是什么情况
2018-04-19
回复 喜欢

krislyy
flv中有专门的tag封装了这两个信息
2018-05-31
回复 1

不知道的大海
封装到video Tag Data结构
2022-06-22
回复 喜欢

otjiang
sdp中不包含sps和pps吧，sdp是回话描述的文本协议；sps和pps是在h264的NALU码流中的
2019-02-22
回复 4



松阳人
应该是sdp里面用的一些参数值是从sps和pps解析获得的
2021-04-01
回复 1

chris liu
h264码流怎么计算出帧率呢
2019-07-05
回复 喜欢

海森堡
先检测SPS中timing_info_present_flag标签的值是否为1，如果是表示SPS中存在
num_units_in_tick，time_scale和fixed_frame_rate_flag这三个标签，至此可以计算出帧率:
fps = time_scale / num_units_in_tick;
if (fixed_frame_rate_flag == 1) {
fps = fps / 2;
}
2020-07-08
回复 8

enjoy
讲得很好，很多了解的人，大多写不出来。楼主棒棒
2019-05-27
回复 喜欢

周嘉华
谢谢博主，理解良多
2021-10-12
回复 喜欢

自成
感谢分享，学习了
2018-02-26
回复 喜欢

雲淡風輕
謝謝 很有幫助
2017-08-28
回复 喜欢

DaveBobo
作者
交流学习
2017-09-04
回复 喜欢

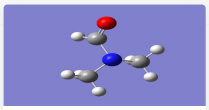
不用吃睡的哑巴
通过 pic_width_in_mbs_minus1计算frame_width，16只是宏块中的亮度宽度，还有两个色度宽度要考虑进去
2022-03-02
回复 喜欢

写下你的评论...

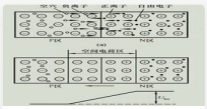
文章被以下专栏收录

DaveBobo的多媒体编程
视音频多媒体编程技术学习交流

推荐阅读



判断DMF中N原子是sp2杂化还是sp3杂化（NBO简单应...
唯理计算 发表于计算教程



模电2-浅谈PN结
凉亭

模拟电路——PN结
一、何为半导体？导体对电信号有良好的导通性，绝缘体对电信号起阻断作用。而半导体的导电能力介于导体和绝缘体之间。 半导体的导电能力随温度、光照和掺杂等因素发生显著变化，这些特点使...
竹知雨

PN结番外篇一——正反向电流和载流子浓度
西瓜骑士敢... 发表于芯路

×
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 131 14 条评论 分享 喜欢 收藏 申请转载 ···