

16-04 TS协议解析

1 ts文件分层

1.1 ts 层：Transport Stream

1.1.1 ts header

1.1.2 adaptation field

PAT 格式如下图

PMT 格式如下图

1.2 pes 层：Packet Elemental Stream

pes 层格式如下图：

pes 层内容如下图：

1.3 es 层：Elementary Stream

1.3.1 h.264 视频

1.3.2 aac音频

零声学院收集整理

腾讯课堂 《FFmpeg/WebRTC/RTMP音视频流媒体高级开发》 <https://ke.qq.com/course/468797?tuin=137bb271>

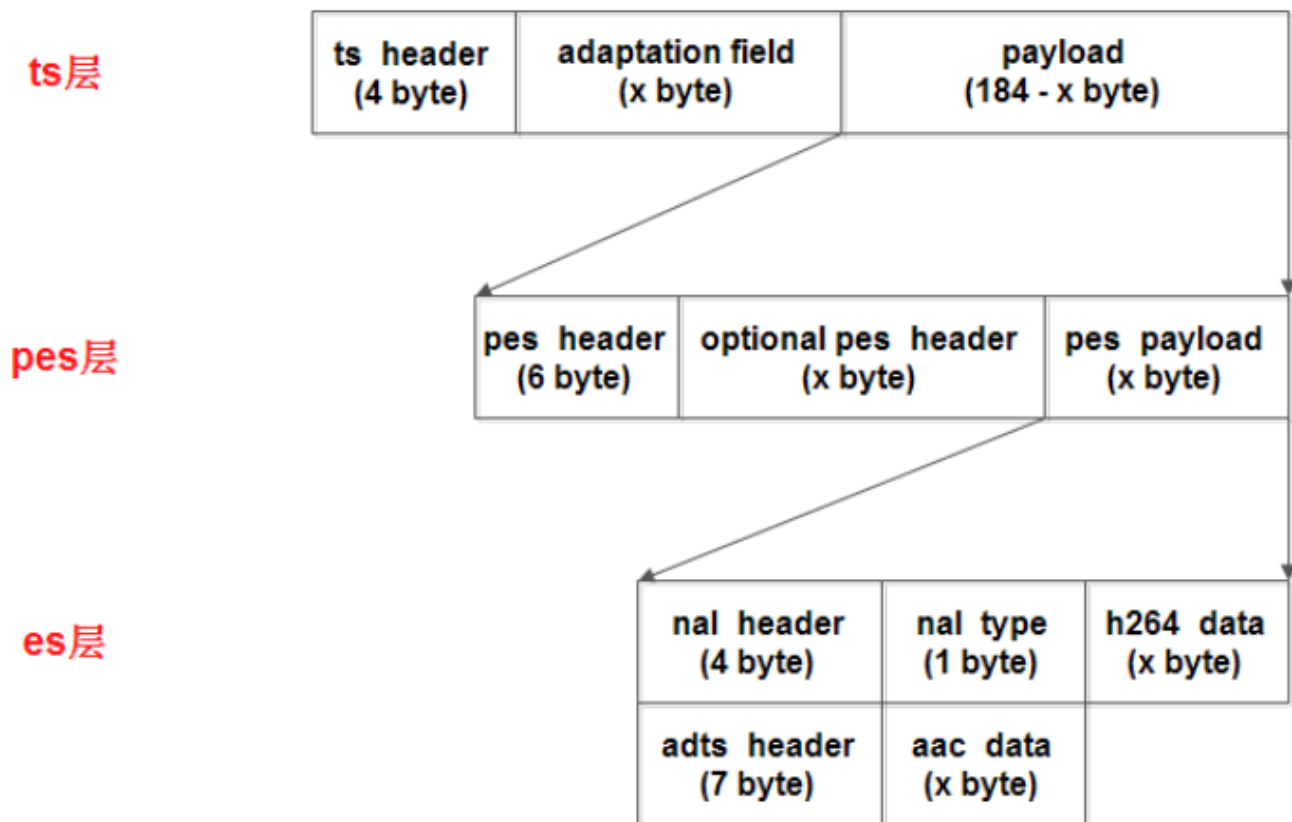
1 ts文件分层

ts 文件为传输流文件，视频编码主要格式为 H264/MPEG4，音频为 AAC/MP3。

ts 文件分为三层：

- ts 层：Transport Stream，是在 pes 层的基础上加入数据流的识别和传输必须的信息。
- pes 层：Packet Elemental Stream，是在音视频数据上加了时间戳等对数据帧的说明信息。

- es 层：Elementary Stream，即音视频数据。



1.1 ts 层：Transport Stream

ts 包大小固定为 188 字节，ts 层分为三个部分：ts header、adaptation field、payload。ts header 固定 4 个字节；adaptation field 可能存在也可能不存在，主要作用是给不足 188 字节的数据做填充；payload 是 pes 数据。

1.1.1 ts header

sync_byte	8b	同步字节，固定为0x47
transport_error_indicator	1b	传输错误指示符，表明在ts头的adapt域后由一个无用字节，通常都为0，这个字节算在adapt域长度内
payload_unit_start_indicator	1b	负载单元起始标示符，一个完整的数据包开始时标记为1
transport_priority	1b	传输优先级，0为低优先级，1为高优先级，通常取0
pid	13b	pid值
transport_scrambling_control	2b	传输加扰控制，00表示未加密
adaptation_field_control	2b	是否包含自适应区，‘00’保留；‘01’为无自适应域，仅含有效负载；‘10’为仅含自适应域，无有效负载；‘11’为同时带有自适应域

		和有效负载。
continuity_counter	4b	递增计数器，从0-f，起始值不一定取0，但必须是连续的

ts 层的内容是通过 PID 值来标识的，主要内容包括：PAT 表、PMT 表、音频流、视频流。解析 ts 流要先找到 PAT 表，只要找到 PAT 就可以找到 PMT，然后就可以找到音视频流了。PAT 表的和 PMT 表需要定期插入 ts 流，因为用户随时可能加入 ts 流，这个间隔比较小，通常每隔几个视频帧就要加入 PAT 和 PMT。PAT 和 PMT 表是必须的，还可以加入其它表如 SDT（业务描述表）等，不过 hls 流只要有 PAT 和 PMT 就可以播放了。

- PAT 表：主要的作用就是指明了 PMT 表的 PID 值。
- PMT 表：主要的作用就是指明了音视频流的 PID 值。
- 音频流/视频流：承载音视频内容。

1.1.2 adaptation field

adaptation_field_length	1B	自适应域长度，后面的字节数
flag	1B	取0x50表示包含PCR或0x40表示不包含PCR
PCR	5B	Program Clock Reference，节目时钟参考，用于恢复出与编码端一致的系统时序时钟STC（System Time Clock）。
stuffing_bytes	xB	填充字节，取值0xff

自适应区的长度要包含传输错误指示符标识的一个字节。pcr 是节目时钟参考，pcr、dts、pts 都是对同一个系统时钟的采样值，pcr 是递增的，因此可以将其设置为 dts 值，音频数据不需要 pcr。如果没有字段，ipad 是可以播放的，但 vlc 无法播放。打包 ts 流时 PAT 和 PMT 表是没有 adaptation field 的，不够的长度直接补 0xff 即可。视频流和音频流都需要加 adaptation field，通常加在一个帧的第一个 ts 包和最后一个 ts 包里，中间的 ts 包不加。如下图所示：

TSHead	PAT/PMT	Stuffing Bytes
--------	---------	----------------

TSHead	Adaptation Field	Pes1
TSHead	Pes2 - Pes(N-1)	
TSHead	Adaptation Field	PesN

PAT 格式如下图

table_id	8b	PAT表固定为0x00
section_syntax_indicator	1b	固定为1

zero	1b	固定为0
reserved	2b	固定为11
section_length	12b	后面数据的长度
transport_stream_id	16b	传输流ID, 固定为0x0001
reserved	2b	固定为11
version_number	5b	版本号, 固定为00000, 如果PAT有变化则版本号加1
current_next_indicator	1b	固定为1, 表示这个PAT表可以用, 如果为0则要等待下一个PAT表
section_number	8b	固定为0x00
last_section_number	8b	固定为0x00
开始循环		
program_number	16b	节目号为0x0000时表示这是NIT, 节目号为0x0001时表示这是PMT
reserved	3b	固定为111
PID	13b	节目号对应内容的PID值
结束循环		
CRC32	32b	前面数据的CRC32校验码

PMT 格式如下图

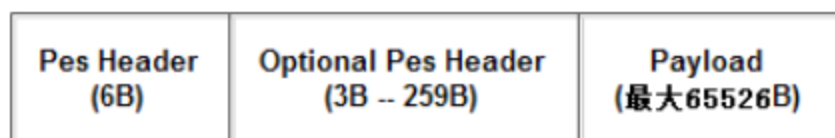
table_id	8b	PMT表取值随意, 0x02
section_syntax_indicator	1b	固定为1
zero	1b	固定为0
reserved	2b	固定为11
section_length	12b	后面数据的长度
program_number	16b	频道号码, 表示当前的PMT关联到的频道, 取值0x0001
reserved	2b	固定为11
version_number	5b	版本号, 固定为00000, 如果PAT有变化则版本号加1
current_next_indicator	1b	固定为1
section_number	8b	固定为0x00

last_section_number	8b	固定为0x00
reserved	3b	固定为111
PCR_PID	13b	PCR(节目参考时钟)所在TS分组的PID，指定为视频PID
reserved	4b	固定为1111
program_info_length	12b	节目描述信息，指定为0x000表示没有
开始循环		
stream_type	8b	流类型，标志是Video还是Audio还是其他数据，h.264编码对应0x1b，aac编码对应0x0f，mp3编码对应0x03
reserved	3b	固定为111
elementary_PID	13b	与stream_type对应的PID
reserved	4b	固定为1111
ES_info_length	12b	描述信息，指定为0x000表示没有
结束循环		
CRC32	32b	前面数据的CRC32校验码

1.2 pes 层：Packet Elemental Stream

pes 层是在每一个视频/音频帧上加入了时间戳等信息，pes 包内容很多，这里只留下最常用的。

pes 层格式如下图：



pes 层内容如下图：

pes start code	3B	开始码，固定为0x000001
stream id	1B	音频取值（0xc0–0xdf），通常为0xc0 视频取值（0xe0–0xef），通常为0xe0
pes packet length	2B	后面pes数据的长度，0表示长度不限制， 只有视频数据长度会超过0xffff
flag	1B	通常取值0x80，表示数据不加密、无优先级、备份的数据
flag	1B	取值0x80表示只含有pts，取值0xc0表示含有pts和dts
pes data length	1B	后面数据的长度，取值5或10

pts	5B	33bit值
pts	5B	33bit值

pts 是显示时间戳、dts 是解码时间戳，视频数据两种时间戳都需要，音频数据的 pts 和 dts 相同，所以只需要 pts。有 pts 和 dts 两种时间戳是 B 帧引起的，I 帧 和 P 帧的 pts 等于 dts。如果一个视频没有 B 帧，则 pts 永远和 dts 相同。从文件中顺序读取视频帧，取出的帧顺序和 dts 顺序相同。dts 算法比较简单，初始值 + 增量即可，pts 计算比较复杂，需要在 dts 的基础上加偏移量。

音频的 pes 中只有 pts（同 dts），视频的 I、P 帧两种时间戳都要有，视频 B 帧只要 pts（同 dts）。打包 pts 和 dts 就需要知道视频帧类型，但是通过容器格式我们是无法判断帧类型的，必须解析 h.264 内容才可以获取帧类型。

举例说明：

1 .	I	P	B	B	B	P
2 读取顺序：	1	2	3	4	5	6
3 dts 顺序：	1	2	3	4	5	6
4 pts 顺序：	1	5	3	2	4	6

点播视频 dts 算法：

$dts = \text{初始值} + 90000 / \text{video_frame_rate}$ ，初始值可以随便指定，但是最好不要取 0，video_frame_rate 就是帧率，比如 23、30。

pts 和 dts 是以 timestamp 为单位的，1s = 90000 time scale，一帧就应该是 90000/video_frame_rate 个 timescale。

用一帧的 timescale 除以采样频率就可以转换为一帧的播放时长。

点播音频 dts 算法：

$dts = \text{初始值} + (90000 * \text{audio_samples_per_frame}) / \text{audio_sample_rate}$ ，audio_samples_per_frame 这个值与编解码相关，aac 取值 1024，mp3 取值 1158，audio_sample_rate 是采样率，比如 24000、41000。AAC 一般解码出来是每声道 1024 个 sample，也就是说一帧的时长为 1024/sample_rate 秒。所以每一帧时间戳依次 0, 1024/sample_rate, ..., 1024*n/sample_rate 秒。

注：直播视频的 dts 和 pts 应该直接用直播数据流中的时间，不应该按公式计算。

1.3 es 层：Elementary Stream

es 层指的就是音视频数据。这里只介绍 h.264 视频和 aac 音频。

1.3.1 h.264 视频

打包 h.264 数据时必须给视频数据加上一个 nalu（Network Abstraction Layer Unit），nalu 包括 nalu header 和 nalu type，nalu header 固定为 0x00000001（帧开始）或 0x0000001（帧中）。h.264 的数据是由 slice 组成的，slice 的内容包括：视频、sps、pps 等。nalu type 决定了后面的 h.264 数据内容。

```

1  .
2      0 1 2 3 4 5 6 7
3      +--+--+--+--+--+--+
4      | F | NRI |  TYPE  |
5      +--+--+--+--+--+--+

```

- F: 1bit, forbidden_zero_bit, h.264 规定必须取 0。
- NRI: 2bits, nal_ref_idc, 取值为 0~3, 指示这个 nalu 的重要性, I 帧、sps、pps 通常取 3, P 帧通常取 2, B 帧通常取 0
- Type: 5bits, 取值如下表所示:

nal_unit_type	说明
0	未使用
1	非IDR图像片, IDR指关键帧
2	片分区A
3	片分区B
4	片分区C
5	IDR图像片, 即关键帧
6	补充增强信息单元(SEI)
7	SPS序列参数集
8	PPS图像参数集
9	分解符
10	序列结束
11	码流结束
12	填充
13~23	保留
24~31	未使用

打包 es 层数据时 pes 头和 es 数据之间要加入一个 type=9 的 nalu, 关键帧 slice 前必须要加入 type=7 和 type=8 的 nalu, 而且是紧邻的。如下图所示:

Pes Head	nalu (0x09)	随意 (1B)	nalu (其他)	内容	nalu (0x67)	内容	nalu (0x68)	内容	nalu (0x65)	内容
----------	----------------	------------	--------------	----	----------------	----	----------------	----	----------------	----

Pes Head	nalu (0x09)	随意 (1B)	nalu (其他)	内容	nalu (0x41)	内容
----------	----------------	------------	--------------	----	----------------	----

1.3.2 aac音频

打包aac音频必须加上一个adts(Audio Data Transport Stream)头, 共7Byte, adts包括fixed_header和variable_header两部分, 各28bit。

fixed_header

syncword	12b	固定为0xfff
id	1b	0表示MPEG-4, 1表示MPEG-2
layer	2b	固定为00
protection_absent	1b	固定为1
profile	2b	取值0~3, 1表示aac
sampling_frequency_index	4b	表示采样率, 0: 96000 Hz, 1: 88200 Hz, 2: 64000 Hz, 3: 48000 Hz, 4: 44100 Hz, 5: 32000 Hz, 6: 24000 Hz, 7: 22050 Hz, 8: 16000 Hz, 9: 12000 Hz, 10: 11025 Hz, 11: 8000 Hz, 12: 7350 Hz
private_bit	1b	固定为0
channel_configuration	3b	取值0~7, 1: 1 channel: front-center, 2: 2 channels: front-left, front-right, 3: 3 channels: front-center, front-left, front-right, 4: 4 channels: front-center, front-left, front-right, back-center
original_copy	1b	固定为0
home	1b	固定为0

variable_header

copyright_identification_bit	1b	固定为0
copyright_identification_start	1b	固定为0
aac_frame_length	13b	包括adts头在内的音频数据总长度
adts_buffer_fullness	11b	固定为0x7ff
number_of_raw_data_blocks_in_frame	2b	固定为00