

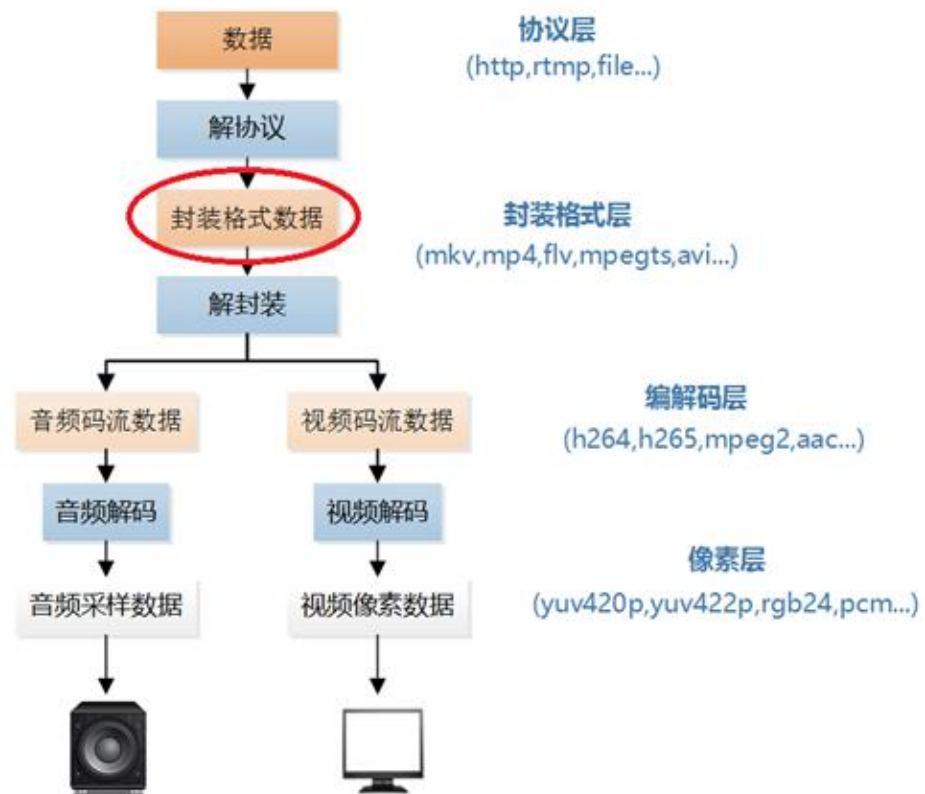
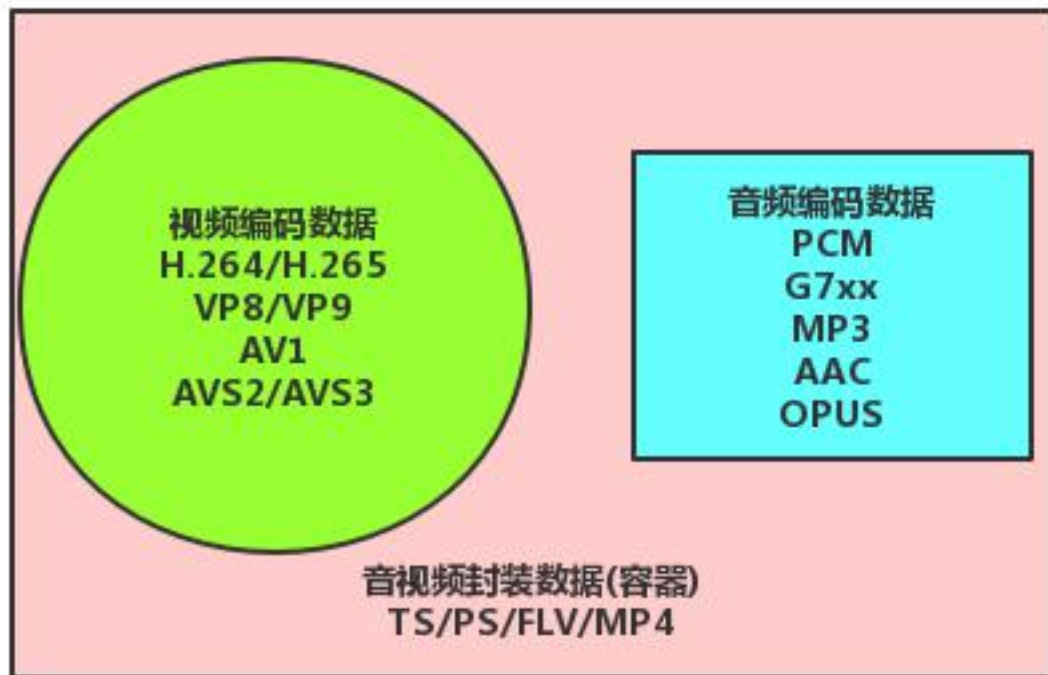
音视频封装面面观

X-Lab:田野
2019/12/17

概要

- 音视频基础
- 音视频封装格式

封装基础



封装基础

- 0.该封装格式推出机构和Spe
- 1.该封装格式支持的音视频编码格式以及怎么封装的
- 2.封装格式的复杂性、容错性，灵活性
- 3.封装格式对流化和网络传输协议的支持
- 4.封装格式对码流高标清切换，对Trick-play播放的支持
- 5.封装格式里面时间戳信息是怎么体现的
- 6.不同封装格式对使用场景的支持

封装总结

名称	推出机构 时间	支持视频编码格式	支持音频编 码格式	传输流媒体协议	概括性描述	目前使用领域
TS MPEG2-TS Transport Stream	MPEG组织 1994年	MPEG2 MPEG4H. 263 H. 264	G7xx AAC	HLS (TCP)	VCD/DVD时代的封装格式，逐渐推出历史舞台，得益于苹果还能存活一段时间。	广电 数字电视 苹果系列产品安防
PS MPEG2-PS Program Stream	MPEG组织 1994年	MPEG2 MPEG4H. 263 H. 264	G7xx AAC	RTP (TCP UDP) RTSP	VCD/DVD时代的封装格式，基本已经退出历史舞台，安防还有少许使用。	DVD电影 安防
FLV Flash Video	Adobe	H. 264	MP3 AAC	HTTP (TCP) RTMP (TCP)	Adobe目前已经不再更新标准，CDN存量比较大，还有一定的市场，但是长远看也会退出历史舞台。	互联网视频网站点 播视频网站CDN
MP4 MOV FMP4 F4V MPEG-4 Part 14	MPEG组织 基于 QuickTime格式。	MPEG2 MPEG4 H. 263 H. 264 H. 265 AV1	MP3 AAC OPUS	HTTP (TCP) HLS (TCP) DASH (TCP)	持续更新相关标准，基本跟上高清这一波潮流，发展动力强劲。	互联网视频网站点 播视频网站短视频
RAW	登虹	H. 264 H. 265 AV1	G711 AAC OPUS	Tcpbuffer (TCP)	公司私有个数数据，仅仅局限登虹自己产品。	安防IoT

MPEG是运动图像专家组 (Moving Picture Experts Group) 的简称，其实质上的名称为国际标准化组织（ISO）和国际电工委员会（IEC）联合技术委员会（JTC）1的第29 分委员会的第11工作组，即ISO/IEC JTC1/SC29/WG11，成立于1988年。

MEPG2基本概念

基本概念：

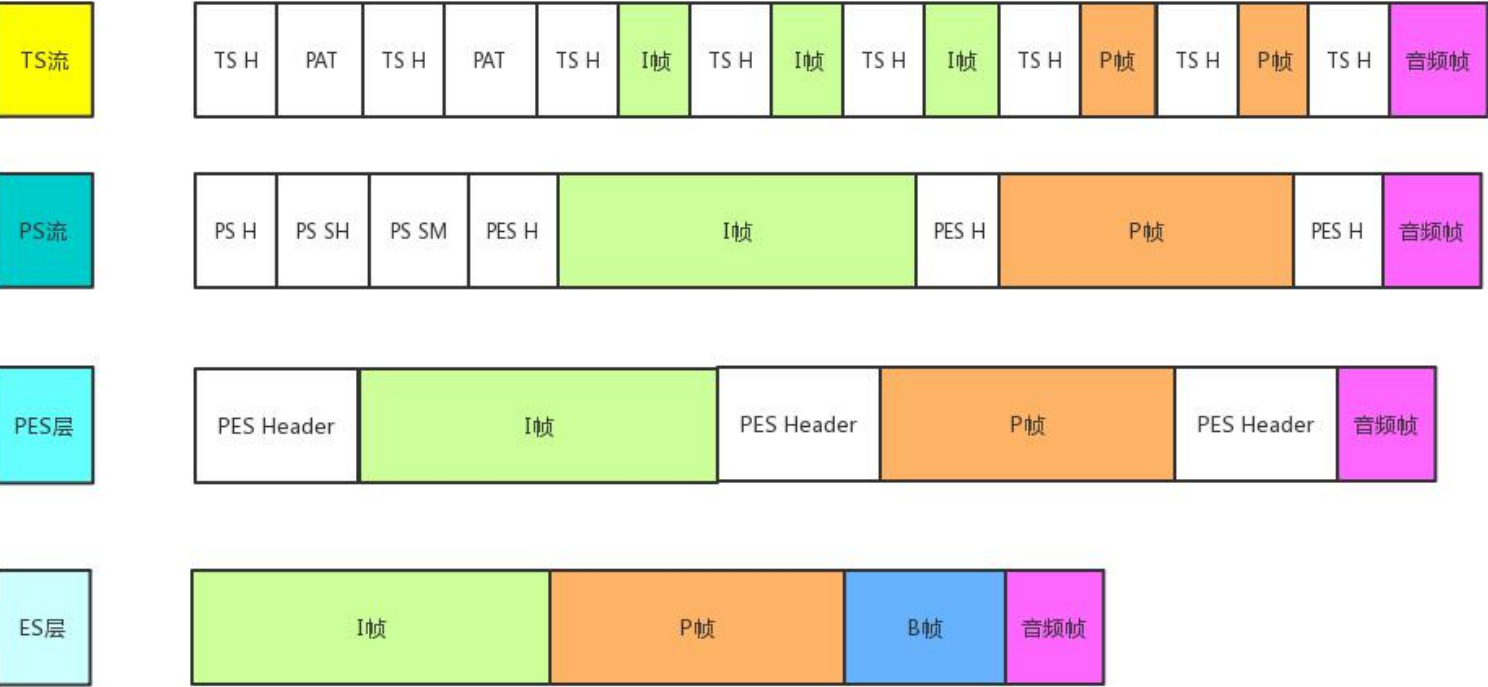
ES即Elementary Stream:指只包含1个信源编码器的数据流，就是编码后的原始视频数据或者音频数据；

PES即Packetized Elementary Stream:将ES流分组打包后，在每个包前面加PES包头就构成了PES,视频PES一般一帧一个包，音频PES一般一个包的数据量不超过64KB；

PS流(Program Stream):节目流, 将具有共同时间基准的一个或多个PES组合（复合）而成的单一数据流（用于播放或编辑系统，如m2p）；

TS流(Transport Stream):传输流，将具有共同时间基准或独立时间基准的一个或多个PES组合（复合）而成的单一数据流（用于数据传输）；

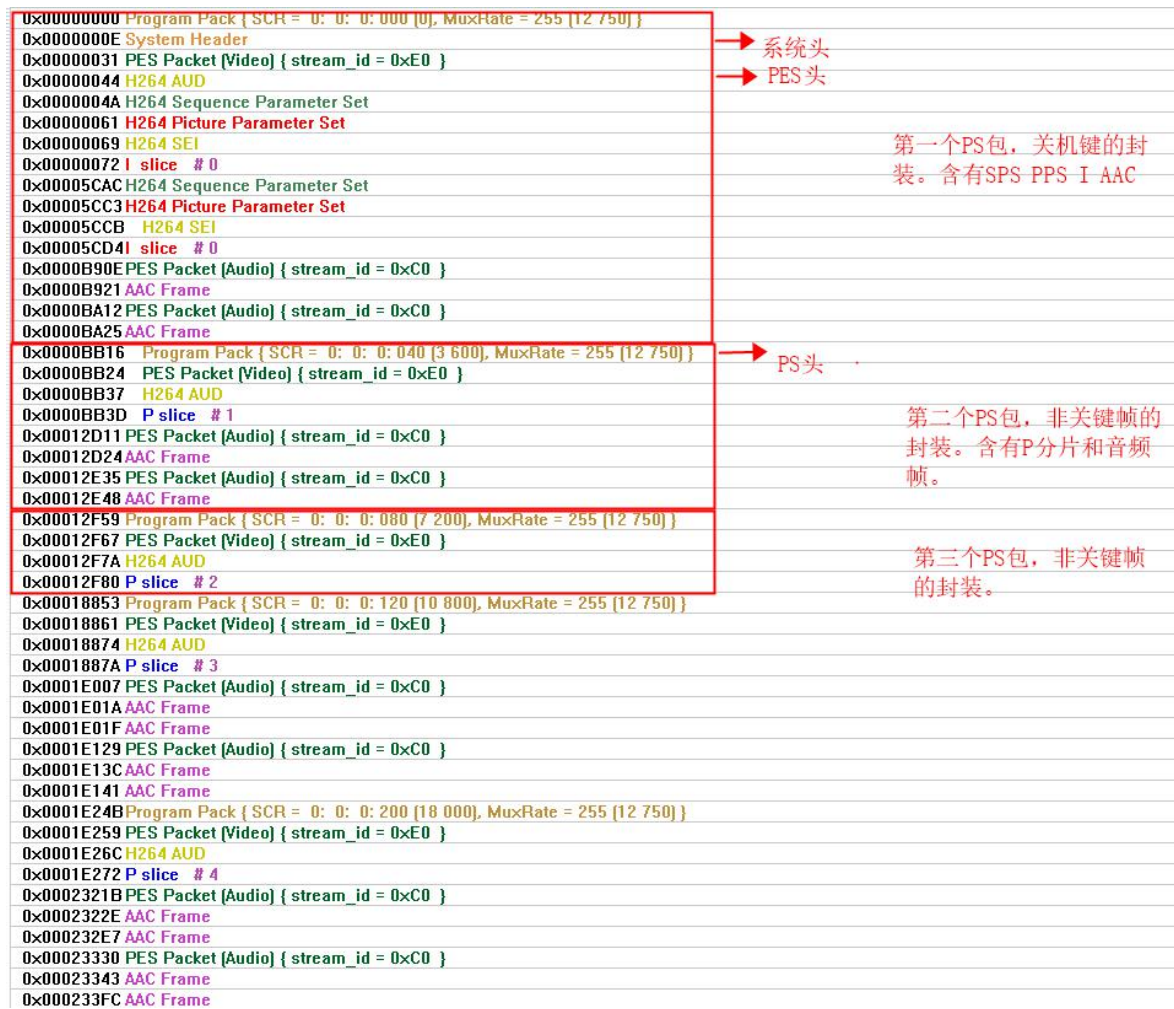
MPEG2码流层次结构



- TS S: Transport Stream Header**
- PAT:Program Association Table** 节目关联表
- PMT:Program Map Table** 节目映射表
- PS H: Program Stream Header**
- PS SH:Program Ststem System Header**
- PS SM:Program Stream System Map**

PS 封装格式解析

PS码流分析示意图



1. IDR关键帧封装:

每个IDR NALU 前一般都会包含SPS、PPS 等NALU，因此将SPS、PPS、IDR 的NALU 封装为一个PS 包:

PSheader| PS system header | PS system Map | PES header | h264 raw data。

2. 非关键帧封装:

PS包从外到内的顺序是:

PSheader| PES header | h264 raw data。

3. 含有音频的PS封装:

PS头|PES(video)|PES(audio)

PSheader| PES header | h264 raw data|PES header|AAC raw data

PS Header

语法	位数	助记符
pack_header() {		
pack_start_code	32	bslbf
'01'	2	bslbf
system_clock_reference_base[32..30]	3	bslbf
marker_bit	1	bslbf
system_clock_reference_base[29..15]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_base[14..0]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_extension	9	uimsbf
marker_bit	1	bslbf
program_mux_rate	22	uimsbf
marker_bit	1	bslbf

marker_bit	1	bslbf
reserved	5	bslbf
pack_stuffing_length	3	uimsbf
for (i=0;i<pack_stuffing_length;i++){		
stuffing_byte	8	bslbf
}		
if (nextbits()==system_header_start_code)		
{		
system_header()		
}		
}		

0x00000000	00 01 BA 44 00 04 00 04 01 00 03 FF F8 00 00	0123456789ABCDEF
0x00000010	01 BB 00 09 81 86 A1 05 E1 7F E0 E8 00 00 00 01	
0x00000020	BC 00 0E E0 FF 00 00 00 04 1B E0 00 00 43 80 9B	
0x00000030	58 00 00 01 E0 B8 D7 80 C0 0A 31 00 01 00 01 11	
0x00000040	00 01 00 01 00 00 00 01 09 30 00 00 00 01 67 42	
0x00000050	00 1F 66 38 14 01 2F 24 00 00 1F 24 00 05 7F 40	

总结：

1. ps头无论是什么样的包都是要有的，00 00 01 BA是关键，其它字

段基本可以忽略不计；

2. 无扩展字段是直接读十四个字节跳过解析即可，但是当

pack_stuffing_length不为0时，需要解析出来，加上该长度跳过。该

字段位于第十四字节的后面三bit位；

3. 也可以根据该值进行打包和封装，基本对应位都是用默认值；

PS System Header

语 法	位数	助记符
system_header() {		
system_header_start_code	32	bslbf
header_length	16	uimsbf
marker_bit	1	bslbf
rate_bound	22	uimsbf
marker_bit	1	bslbf
audio_bound	6	uimsbf
fixed_flag	1	bslbf
CSPS_flag	1	bslbf
system_audio_lock_flag	1	bslbf
system_video_lock_flag	1	bslbf
marker_bit	1	bslbf
vedio_bound	5	uimsbf
packet_rate_restriction_flag	1	bslbf
reserved_bits	7	bslbf
while (nextbits()!='1') {		
stream_id	8	uimsbf
'11'	2	bslbf
P-STD_buffer_bound_scale	1	bslbf
P-STD_buffer_size_bound	13	uimsbf
}		
}		

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
0x00000000	00	00	01	BA	44	00	04	00	04	01	00	03	FF	F8	00	00	...D.....
0x00000010	01	BB	00	09	81	86	A1	05	E1	7F	E0	E8	00	00	00	01[].....
0x00000020	BC	00	0E	E0	FF	00	00	00	04	1B	E0	00	00	43	80	9BC..
0x00000030	58	00	00	01	E0	B8	D7	80	C0	0A	31	00	01	00	01	11	X.....l.....
0x00000040	00	01	00	01	00	00	00	01	09	30	00	00	00	01	67	420....gB
0x00000050	00	1F	95	A8	14	01	6E	84	00	00	1F	F4	00	05	7E	40n.....~@
0x00000060	10	00	00	00	01	68	CE	3C	80	00	00	00	01	06	E5	01h.<.....
0x00000070	2F	80	00	00	00	01	65	B8	00	00	6B	94	7C	46	1E	00	/.....e....k. F..
0x00000080	01	00	09	07	00	1A	4C	0B	28	75	82	30	1B	64	0E	B2L.{u.0.d..
0x00000090	A7	C1	48	70	86	84	F0	64	03	41	F6	5F	FD	BF	F8	20	..Hp...d.A._...
0x000000A0	FC	2F	3F	FB	F5	79	DA	3F	FF	FA	78	53	29	F7	FF		./?...y.?./...xS)..
0x000000B0	D7	DA	23	56	8F	CD	FF	FD	3E	F1	12	09	33	77	AB	C5	..#V....>...3w..
0x000000C0	DE	2E	7C	5D	E7	C5	3A	BC	5D	E2	99	F5	AB	C5	CF	89	..]...:.].....
0x000000D0	3E	A7	CD	EA	F1	77	89	0F	3E	27	EA	7C	2F	40	F7	A9	>....w...>'./@..
0x000000E0	F1	27	D4	F8	5E	82	A7	CD	EA	F1	21	E7	C4	9F	53	E1	..'...^.....!...S.
0x000000F0	70	A0	AB	C1	A8	90	1F	06	A2	40	F3	E0	D4	48	1E	7C	p.....@...H.
0x0000100	1A	84	82	A7	C1	A8	48	2F	3E	0D	42	41	57	83	50	90H/>..BAW.P.
0x0000110	3C	1A	89	01	F0	6A	12	0B	DE	0D	44	81	E0	B2	24	0F	<....j....D...\$.
0x0000120	83	50	90	3D	E0	AA	28	43	C1	A8	48	0F	83	50	90	74	.P.=..(C..H..P.t
0x0000130	F8	35	09	07	5E	0D	49	01	F0	5D	0A	10	3B	A2	35	F4	.S...^..I..]...;5.
0x0000140	FD	AF	BE	FB	EF	BE	F9	F0	61	03	40	6F	3E	0C	83	41a.@o>..A

总结：

对于系统头部的解析，我们一般只要先首先判断是否存在系统头（根据系统头的起始码

0x000001BB），然后我们读取系统头的头部长度,即PS SYSTEM HEADER LENGTH部分，

然后根据头部的长度，跳过PS系统头。进入下一个部分，即PS 节目流映射头。

PS System Map Header

语 法	位数	助记符
program_stream_map() {		
packet_start_code_prefix	24	bslbf
map_stream_id	8	uimsbf
program_stream_map_length	16	uimsbf
current_next_indicator	1	bslbf
reserved	2	bslbf
program_stream_map_version	5	uimsbf
reserved	7	bslbf
marker_bit	1	bslbf
program_stream_info_length	16	uimsbf
for (i=0;i<N;i++){		
descriptor()		
}		
elementary_stream_map_length	16	uimsbf
for (i=0;i<N1;i++){		
stream_type	8	uimsbf
elementary_stream_id	8	uimsbf
elementary_stream_info_length	16	uimsbf
for (i=0;i<N2;i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Stream Type:
1、MPEG-4 视频流： 0x10 ;
2、H.264 视频流： 0x1B
3、SVAC 视频流： 0x80
4、G.711 音频流： 0x90
5、G.722.1 音频流： 0x92
6、G.723.1 音频流： 0x93
7、G.729 音频流： 0x99
8、SVAC音频流： 0x9B

4字节描述PS基本流的
音视频编码类型

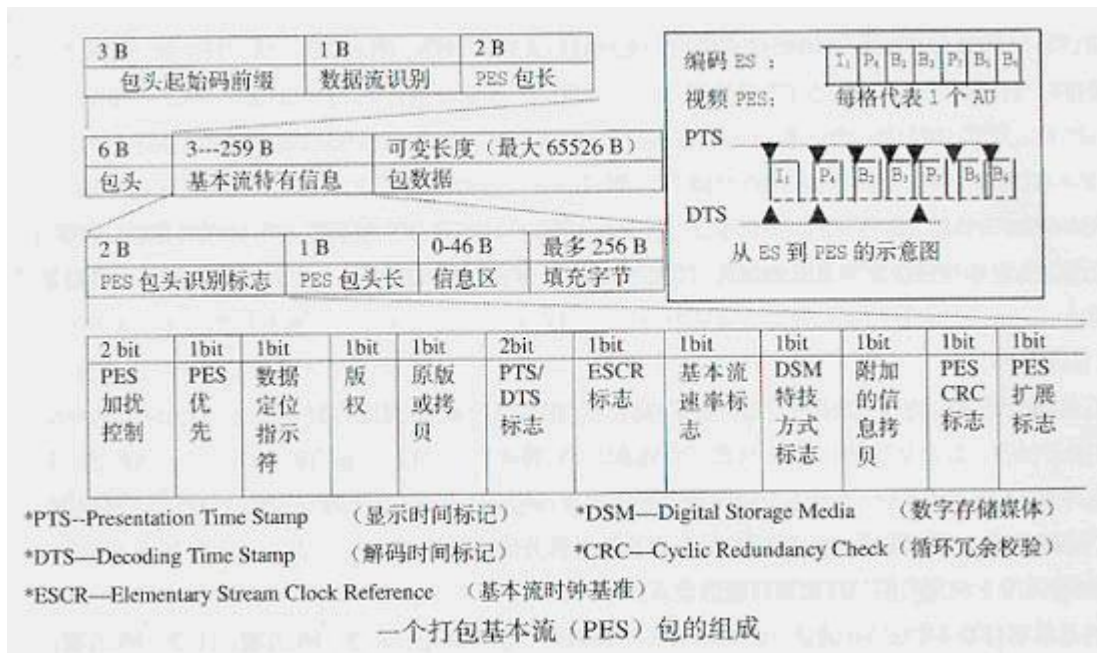
elementary_sream_id:
这个字段的定义,其0x(C0~DF)
指音频, 0x(E0~EF)为视频
该基本流所在PES分组的PES
分组标题中stream_id字段的
值。

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
0x00000000	00	00	01	BA	44	00	04	00	04	01	00	03	FF	F8	00	00D.....
0x00000010	01	BB	00	09	81	86	A1	05	E1	7F	E0	E8	00	00	00	01[].....
0x00000020	BC	00	0E	E0	FF	00	00	00	04	1B	E0	00	00	43	80	9BC..
0x00000030	58	00	00	01	E0	B8	D7	80	C0	0A	31	00	01	00	01	11	X.....1....
0x00000040	00	01	00	01	00	00	00	01	09	30	00	00	00	01	67	420....gB
0x00000050	00	1F	95	A8	14	01	6E	84	00	00	1F	F4	00	05	7E	40n.....~@
0x00000060	10	00	00	00	01	68	CE	3C	80	00	00	00	01	06	E5	01h.<.....
0x00000070	2F	80	00	00	00	01	65	B8	00	00	6B	94	7C	46	1E	00	/.....e...k. F..
0x00000080	01	00	09	07	00	1A	4C	0B	28	75	82	30	1B	64	0E	B2L.(u.0.d..
0x00000090	A7	C1	48	70	86	84	F0	64	03	41	F6	5F	FD	BF	F8	20	..Hp...d.A._...
0x000000A0	FC	2F	3F	FB	F5	79	DA	3F	2F	FF	FA	78	53	29	F7	FF	./?...y.?/..xS)..
0x000000B0	D7	DA	23	56	8F	CD	FF	FD	3E	F1	12	09	33	77	AB	C5	..#V....>...3w..
0x000000C0	DE	2E	7C	5D	E7	C5	3A	BC	5D	E2	99	F5	AB	C5	CF	89	..]...:].....
0x000000D0	3E	A7	CD	EA	F1	77	89	0F	3E	27	EA	7C	2F	40	F7	A9	>....w...>'./@..
0x000000E0	F1	27	D4	F8	5E	82	A7	CD	EA	F1	21	E7	C4	9F	53	E1	.'...^.....!...S.
0x000000F0	70	A0	AB	C1	A8	90	1F	06	A2	40	F3	E0	D4	48	1E	7C	p.....@...H.
0x00000100	1A	84	82	A7	C1	A8	48	2F	3E	0D	42	41	57	83	50	90H/>.BAW.P.
0x00000110	3C	1A	89	01	F0	6A	12	0B	DE	0D	44	81	E0	B2	24	0F	<....j....D...\$.
0x00000120	83	50	90	3D	E0	AA	28	43	C1	A8	48	0F	83	50	90	74	.P.=..(C..H..P.t
0x00000130	F8	35	09	07	5E	0D	49	01	F0	5D	0A	10	3B	A2	35	F4	.5..^..I..]...;5.
0x00000140	FD	AF	BE	FB	EF	BE	F9	F0	61	03	40	6F	3E	0C	83	41a.@o>..A
0x00000150	79	F0	3C	80	34	17	9F	06	40	D0	54	F8	07	D0	83	9C	y.<.4...@.T.....
0x00000160	AB	C2	00	83	AC	78	2A	43	44	7C	12	24	03	A8	D7	81x*CD .\$.

对于这个字段的解析，我们需要取值0x000001BC的位串，指出节目流映射的开始，暂时不需要处理，读取Header Length直接跳过即可，如果需要解析流编码类型，必须详细解析这个字段。

实际上我们需要在解析节目映射流头字段时，需要解析基本映射流长度字段,这里面包含这是否含有音视频以及音视频对应的编码格式。

PES Header



重点字段

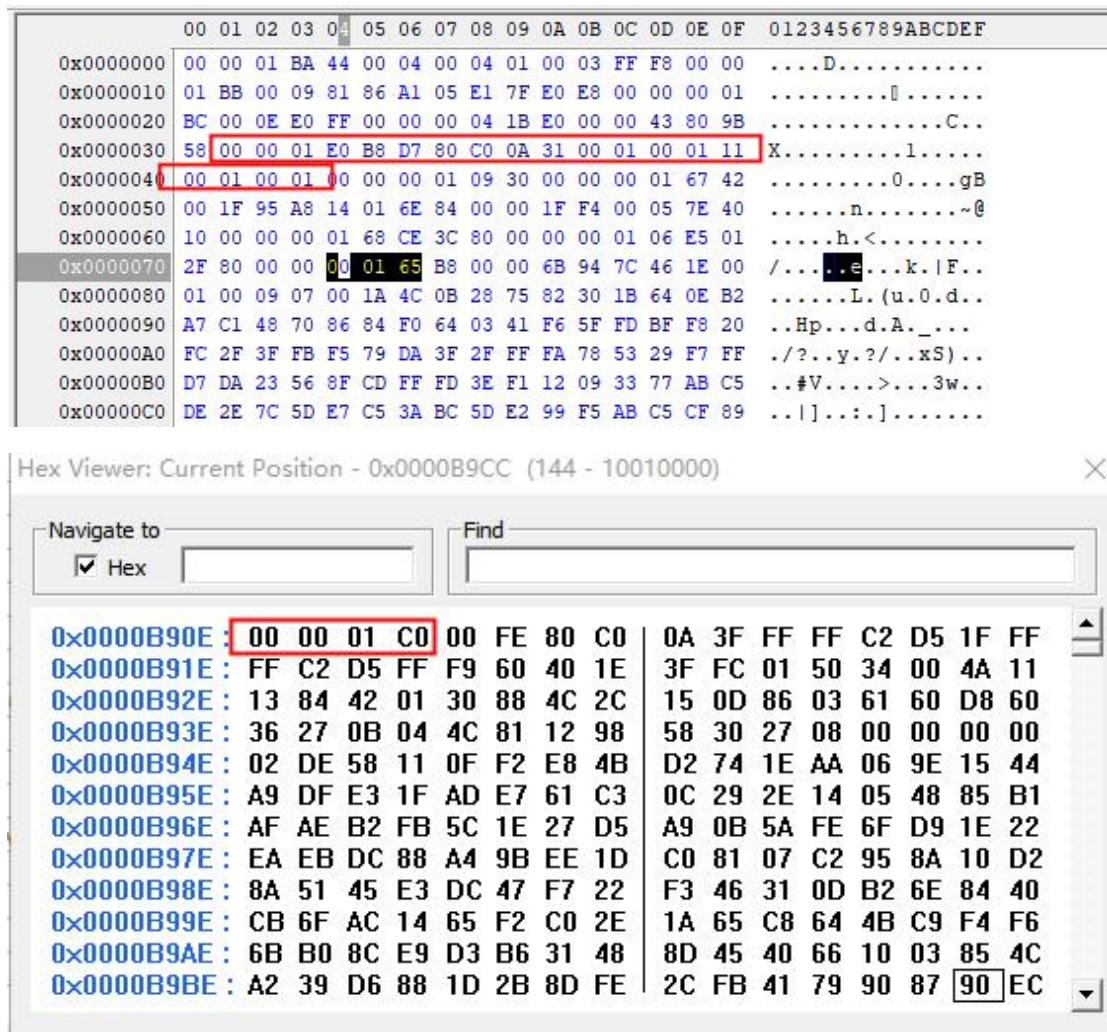
1. Stream_id :区分视频还是音频以及私有数据的关键；
2. PTS_DTS_flags :2位字段。当值为'10'时，PTS字段应出现在PES分组标题中；当值为'11'时，PTS字段和DTS字段都应出现在PES分组标题中；当值为'00'时，PTS字段和DTS字段都不出现在PES分组标题中。值'01'是不允许的；

总结：

PES显然是里面所有头最复杂的，但是很多字段到底有没有定义，需要看第八个字节的六大标记，如果没有此标记解析起来相对简单。

- 1.先解析头的前缀四字节，根据stream_id判断是音频还是视频；
- 2.紧接着长度，则为下一个ps包的起始位置；
- 3.判断PTS_DTS_flags PTS DTS标记；
- 4.解析PTS和DTS值即可；

PES Header



总结:

PES显然是里面所有头最复杂的, 但是很多字段到底有没有定义, 需要

看第八个C0字节的六大标记, 如果没有此标记解析起来相当简单。特

别是PES扩展字段标记和特技方式字段。再后面是长度0A, 后面的字

节则就是PES重要信息PTS和DTS。当然有没有DTS还是要看前面的标

记。一种时间戳占五个字节。

1.先解析头的前缀四字节, 根据stream_id判断是音频还是视频;

2.紧接着长度, 则为下一个ps包的起始位置;

3.判断PTS_DTS_flags PTS DTS标记, 解析PTS和DTS值即可;

4.stream_id里面含有一些私有数据0xBD, 实现是需要去掉;

国标PS流

总结:

国标PS流虽然非常复杂，每个包头涉及字段很多。但是实际解析起来可以忽略大量字段,同时如果是封装，对于不关心字段也就填充默认值来处理:

1.PS头即一般就是14字节，直接识别跳过即可;

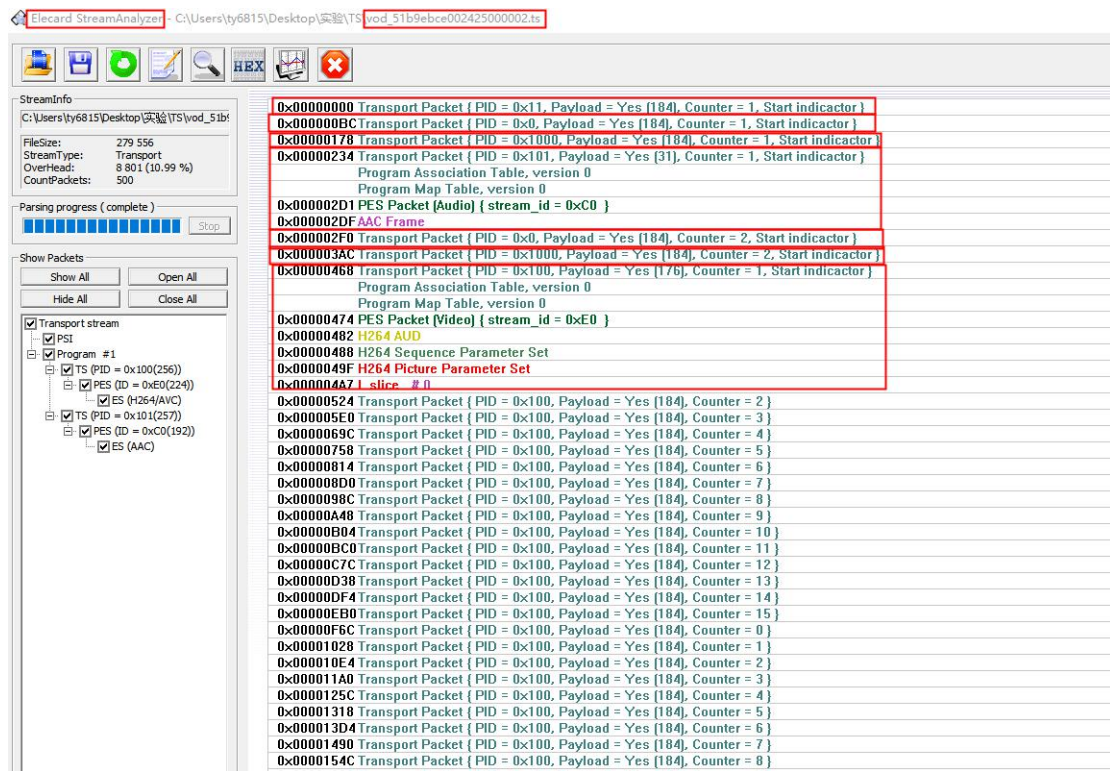
2.系统标题头也是识别前缀，直接读取长度跳过即可;

3.节目映射头只有当PES里面承载的ES是IDR帧时才会存在，一般要解析是否还有音视频，同时了解他们的编码格式;

4.PES头虽然复杂，但是我们只解析里面的PTS和DTS，里面的六大标记字段只有是0，解析起来也是非常简单的，其中第7和第8字节是关键;

TS 封装格式解析

TS流分析



总结:

1.TS流由一个个TS包组成，正常情况下每个TS包固定大小是188字节；

2.TS包类型有PAT、PMT、视频包、音频包等，其中PAT是分析TS的起始位置和关键；

3.TS流是下面这种形式包:

PAT PMT PMT PMT DATA DATAPAT PMT PMT PMT

DATA DATA.....

4.TS流和PS流在PES层面是统一的，可以自由相互转换；

5.一个完整的PES包由整数个TS包组成，否则需要对最后一个TS进行填充；

PAT Program Association Table:节目关联映射表

PMT Program Map Table: 节目映射表

TS Header

The screenshot displays the MPEG-2 TS packet analyser 2.4.6.0 interface. The 'TS header' section is highlighted with a red box, showing fields: Sync byte (0x47), Transport error (0), Payload unit start (1), Transport priority (0), PID (0), Transport scrambling (0), Adaptation field control (1), and Continuity counter (1). The 'Adaptation' section is also highlighted, showing Adaptation byte count (0) and 'No adaptation field, payload only'. The 'TS file' section shows the path 'C:\Users\ty6815\Desktop\实验\TS\vod_51b9ebce002425000002.ts' and info '273 Kb, 1,487 188-byte packets, offset 0, pre ts header 0'. The 'TS packet 2' section shows a hex dump of the packet data, with the first few bytes (47 40 00 11 00) highlighted. The 'Table' section on the right shows the 'program_association_section' with fields: table_id (0x0), section_length (13), transport_stream_id (1), version_number (0), current_next (True), section_number (0), last_section_number (0), program_number (1), and program_map_PID (4096). The 'Section CRC' is also shown as 2A B1 04 B2.

MPEG-2 TS packet analyser 2.4.6.0

File Tools Options Help

Filter

☐ Payload start 1

☐ PID

TS header

Sync byte 0x47

Transport error 0

Payload unit start 1

Transport priority 0

PID 0

Transport scrambling 0

Adaptation field control 1

Continuity counter 1

Adaptation

Adaptation byte count 0

No adaptation field, payload only

TS file

Path C:\Users\ty6815\Desktop\实验\TS\vod_51b9ebce002425000002.ts

Info 273 Kb, 1,487 188-byte packets, offset 0, pre ts header 0

TS packet 2

0 47 40 00 11 00 00 B0 0D 00 01 C1 00 00 00 01 F0

16 00 2A B1 04 B2 FF FF FF FF FF FF FF FF FF FF

32 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

48 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

64 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

80 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

96 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

112 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

128 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

144 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

160 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

176 FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Table

table_id: 0x0

(program_association_section)

section_length: 13

transport_stream_id: 1

version_number: 0

current_next: True

section_number: 0

last_section_number: 0

program_number: 1

program_map_PID: 4096

Section CRC: 2A B1 04 B2

File position: Ready.

重点字段解析：

Payload unit start:表示这个TS包是不是所承载内容的第一个TS包标识承载内容的开始和结束；

Continuity Counter:根据分别承载内容的不同，分别计数0-15之间循环；

PID：代表当前TS包承载内容类型一般第一个TS包的PID是0x00代表PAT；

PAT

字段	字段含义	占位	属于第几字节	当前数据和分析
table_id	表ID	8	第一字节	0x00: 标识一个TS PSI分段的内容是节目关联分段, 条件访问分段还是节目映射分段。 对于PAT置为0x00
section_syntax_indication	段同步标识	1	第二字节	1:
"0"	0值	1	第二字节	0:
reserved	保留值	2	第二字节	11:
section_length	分段长度	12	第三字节	0000 0000 1100: 0x0D: 十三字节。分段长度字段, 这个值是包括该字段在内到CRC_32校验字段的字节数, 其值不超过1021
transport_stream_id	传输流标识	16	第四、五字节	0x00 0x01: 该字节充当标签, 标识网络内此传输流有别于任何其他路复用流。其值由用户规定。
reserved	保留值	2		11:
version_number	PAT的版本号	5	第六字节	00 000: PAT的版本号, 如果PAT有变, 则版本号加1

current_next_indicator	标识	1	第六字节	1: 置为0时, 表明该传送的段不能使用, 下一个表分段才能有效, 一般默认值用1
section_number	分段号	8	第七字节	0x00表明该TS包属于PAT的第几个分段, 分段号从0开始。因为PAT可以描述很多PMT信息, 所以长度可能比较长。
last_section_number	最后一个分段号	8	第八字节	0x00表明该PAT的最大分段数目, 一般情况都是一个PAT表由一个TS包传输。
program_number (循环开始, N从0开始)	节目的编号	16	N+2字节	0x00 0x01
reserved	保留值	3	N+3字节	111:
network_PID	NIT表的PID值	13	N+3 N+4字节	节目号为0则用此值;
program_map_PID	PMT的ID值	13	N+4字节	1 0000 0000 0000: 0x10 0x004096 其它时, 则填充此值;
crc_32	CRC校验	32	最后载荷四个字节	CRC校验

PAT

总结：

1. 表格灰色部分是个循环，整个占用四字节，那到底有几个循环是怎么算出来的？

是根据`section_length-6-4`字节/4算来的，减去6就是`section_length`到`last_section_number`字段和，

减去4是因为还有CRC_32占用的四字节，总长度是0x0d即 $(13 - 6 - 4)/4$ ；

2. 表格红色部分，是if-else if关系。取决于前面字段节目号是否从0开始；

3. PAT表我们主要就是解析怎么获取到后面的PMT的表格ID；

PMT

MPEG-2 TS packet analyser 2.4.6.0

File Tools Options Help

Filter

☐ Payload start 1

☐ PID

TS file

Path C:\Users\ty6815\Desktop\实验\TS\vod_51b9ebce002425000002.ts

Info 273 Kb, 1,487 188-byte packets, offset 0, pre ts header 0

TS header

Sync byte 0x47

Transport error 0

Payload unit start 1

Transport priority 0

PID 4096

Transport scrambling 0

Adaptation field control 1

Continuity counter 1

Adaptation

Adaptation byte count 0

No adaptation field, payload only

TS packet 3

0	47 50 00 11 00 02 B0 17 00 01 C1 00 00 E1 00 F0
16	00 1B E1 00 F0 00 0F E1 01 F0 00 2F 44 B9 9B FF
32	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
48	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
64	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
80	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
96	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
112	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
128	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
144	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
160	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
176	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

File position: Ready.

1. 当前频道中包含的Video数据的PID；
2. 当前频道中包含的Audio数据的PID；
3. 和当前频道关联在一起的其它数据PID；

PMT

字段	字段含义	占位	属于第几字节	当前数据和分析
table_id	表ID	8	第一字节	0x02: 标识一个TS PSI分段的內容是节目关联分段，条件访问分段还是节目映射分段。对于PMT置为0x02
section_syntax_indication	段同步标识	1	第二字节	1: 对于PMT该字段置为1
“0”	0值	1	第二字节	0:
reserved	保留值	2	第二字节	11:
section_length	分段长度	12	第二、三字节	0000 0001 0111: 0x17: 23: 二十三字节。分段长度字段，前两位置00，这个值是包括该字段在内到CRC_32校验字段的字节数，起值不超过1021
program_number	传输流标识	16	第四、五字节	0x00 0x01: 对应于PAT中的program_number
reserved	保留值	2	第六字节	11:
version_number	PMT的版本号	5	第六字节	00 000: PMT的版本号，如果字段中有关信息有变，则版本号以32为模加1。版本号是对一个节目的定义。
current_next_indicator	标识	1	第六字节	1: 该字段置为1时，表示当前传送的program_map_section可用。置为0时，表明该传送的段不能使用，下一个表分段才能有效；

section_number	分段号	8	第七字节	0x00该字段一般总是置为0x00
last_section_number	最后一个分段号	8	第八字节	0x00该字段一般总是置为0x00
reserved	保留号	3	第九字节	1110 0001 0000 0000
PCR_PID	PCR值	13	第九字节第十字节	0 0001 0000 00000x100256该字段指示TS包的PID值，该TS含有该PCR字段，而PCR值对应于有节目号指定的节目。
reserved	保留值	4	第十一字节	1111
program_info_length不为0，后面进行循环层1	节目信息长度	12	第十一字节第十二字节	0000 0000 000: 表明跟随其后的对节目信息描述的字节数，也就是第一个N loop descriptors的字节数。这里是0则表示第一层循环略过；

PMT

stream_type开始循环层2	流类型	8	N+1字节 (N=0)	0x1B:表明这个流是h264编码格式; 表示PES流的类型。
reserved	保留值	3	N+2字节 (N=0)	111:
elementary_pid	负载该PES流的TS包的PID值	13	N+2字节N+3字节 (N=0)	0 0001 0000 0000: 0x100: 256: 表明负载该PES流的PID值
reserved	保留值	4	N+4字节 (N=0)	1111
es_info_length不为0, 循环层3	Es流描述相关的字节数	12	N+5字节 (N=0)	0000 0000 0000: 表明跟随其后描述相关节目元素的字节数; 否则为第二个循环的第二层循环;
stream_type	流类型	8	N+1字节 (N=1)	0x0F
reserved	保留值	3	N+2字节 (N=1)	111
elementary_pid	负载该PES流的TS包的PID值	13	N+2字节N+3字节 (N=1)	0 0001 0000 0001:0x101257:
reserved	保留值	4	N+4字节 (N=1)	1111:
es_info_length	Es流描述相关的字节数	12	N+5字节 (N=1)	0000 00000000: 表明跟随其后描述相关节目元素的字节数; 否则为第二个循环的第二层循环; 默认一般为0
crc_32	CRC校验	32	载荷最后四字节	0x2F 0x44 0xB9 0x9B CRC校验

总结:

1. program_info_length如果不为0, 则有多少字节, 则后面要跟多少字节对节目信息进行描述。
2. stream_type到es_info_length是另外一层循环, 在这里面有可能还存在一层循环, 就是es_info_length, 不为0时要将此字节计算在之内, 如果为0则一次循环需要五字节。
3. 图表中显示了红色和黄色循环了两次。
4. 这里面定义的真实码流视频和音频的PID, 所以PMT是定义每路节目的音视频类型TYPE和编号PID的关键。

音频TS

MPEG-2 TS packet analyser 2.4.6.0

File Tools Options Help



Filter

☐ Payload start 1
☐ PID

TS header

Synco byte 0x47
Transport error 0
Payload unit start 1
Transport priority 0
PID 257
Transport scrambling 0
Adaptation field control 3
Continuity counter 1

Adaptation

Adaptation byte count 153
Adaptation field followed by payload

TS file

Path C:\Users\ty6815\Desktop\实验\TS\vod_51b9ebce002425000002.ts
Info 273 Kb, 1,487 188-byte packets, offset 0, pre ts header 0

TS packet 4

0	47	41	01	31	98	00	FF	FF	FF	FF	FF	FF	FF	FF	FF
16	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
32	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
48	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
64	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
80	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
96	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
112	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
128	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
144	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00
160	C0	00	19	80	80	05	21	00	63	43	41	FF	F1	6C	40
176	BF	FC	00	FA	00	AC	21	C2	CD	00	00	0E			

PES start code found at byte 157
audio stream 192

Adaptation field length: 152
discontinuity indicator: False
random_access_indicator: False
ES_priority_indicator: False
PCR_flag: False
OPCR_flag: False
splicing_point_flag: False
transport_private_data_flag: False
adaptation_field_extension_flag: False

PES header

stream_id: C0 (audio stream 192)
PES_packet_length: 25
PES_scrambling: 0
PES_priority: False
data_alignment: False
copyright: False
original_or_copy: False
PTS_flag: True
DTS_flag: False
ESCR_flag: False
ES_rate_flag: False
DSM_trick_mode_flag: False
additional_copy_info_flag: False
PES_CRC_flag: False
PES_extension_flag: False
PES_header_data_length: 5
PTS: 1614240

0x000002D1 PES Packet (Audio) { stream_id = 0xC0 }

packet_length = 25
PES_scrambling_control = 0
PES_priority = 0
data_alignment_indicator = 0
copyright = 0
original_or_copy = 0
PTS_DTS_flags = 2
ESCR_flag = 0
ES_rate_flag = 0
DSM_trick_mode_flag = 0
additional_copy_info_flag = 0
PES_CRC_flag = 0
PES_extension_flag = 0
PES_header_data_length = 5
PTS = 0: 0: 17: 936 (1 614 240)

0x000002DF AAC Frame

id = 0
layer = 0
protection_absent = 1
profile = 1
sf_index = 11
private_bit = 0
channel_configuration = 1
original = 0
home = 0
copyright_identification_bit = 0
copyright_identification_start = 0
aac_frame_length = 17
adts_buffer_fullness = 2047
no_raw_data_blocks_in_frame = 0
SamplingRate = 8000
Channels = 1
Duration = 206258

file position: Ready.

视频TS

MPEG-2 TS packet analyser 2.4.6.0

File Tools Options Help

Filter

☐ Payload start 1

☐ PID

TS header

Sync byte	0x47
Transport error	0
Payload unit start	1
Transport priority	0
PID	256
Transport scrambling	0
Adaptation field control	3
Continuity counter	1

Adaptation

Adaptation byte count 8

Adaptation field followed by payload

TS file

Path C:\Users\ty6815\Desktop\实验\TS\vod_51b9ebce002425000002.ts

Info 273 Kb, 1,487 188-byte packets, offset 0, pre ts header 0

TS packet 7

0	47 41 00 31 07 50 00 0C 5B 5C 7E 00 00 00 01 E0
16	00 00 80 80 05 21 00 63 6D 71 00 00 00 01 09 F0
32	00 00 00 01 67 4D 00 1F 95 A8 14 01 6E 84 00 00
48	1C 20 00 05 7E 40 10 00 00 00 01 68 EE 3C 80 00
64	00 00 01 65 B8 00 00 0C 08 50 57 FC EF 27 51 85
80	1A D1 51 C7 23 70 E0 67 77 35 E8 57 0C CD A2 B7
96	D3 24 35 A4 D0 47 4B 24 55 D6 DD EF 49 01 7E CC
112	FF 88 61 45 EB A4 B1 B7 A0 25 35 3C 83 00 3A F6
128	89 10 47 2C 63 EF E7 F5 D6 6C D7 71 8A 16 5C 9E
144	FA 20 D8 30 18 6C E6 FA 33 A8 2B E6 DC 3B 8E 9A
160	9E FC 07 06 B7 4E 85 88 7D E9 3E B0 2D 0D 33 90
176	08 B7 F2 81 D8 BE F1 C4 65 7D DF 2E

PES start code found at byte 12

video stream 224

截图(Alt + A)

PES header

stream_id: E0 (video stream 224)

PES_packet_length: 0 (undefined)

PES_scrambling: 0

PES_priority: False

data_alignment: False

copyright: False

original_or_copy: False

PTS_flag: True

DTS_flag: False

ESCR_flag: False

ES_rate_flag: False

DSM_trick_mode_flag: False

additional_copy_info_flag: False

PES_CRC_flag: False

PES_extension_flag: False

PES_header_data_length: 5

PTS: 1619640

Video sequence

Sequence header code not found in this packet

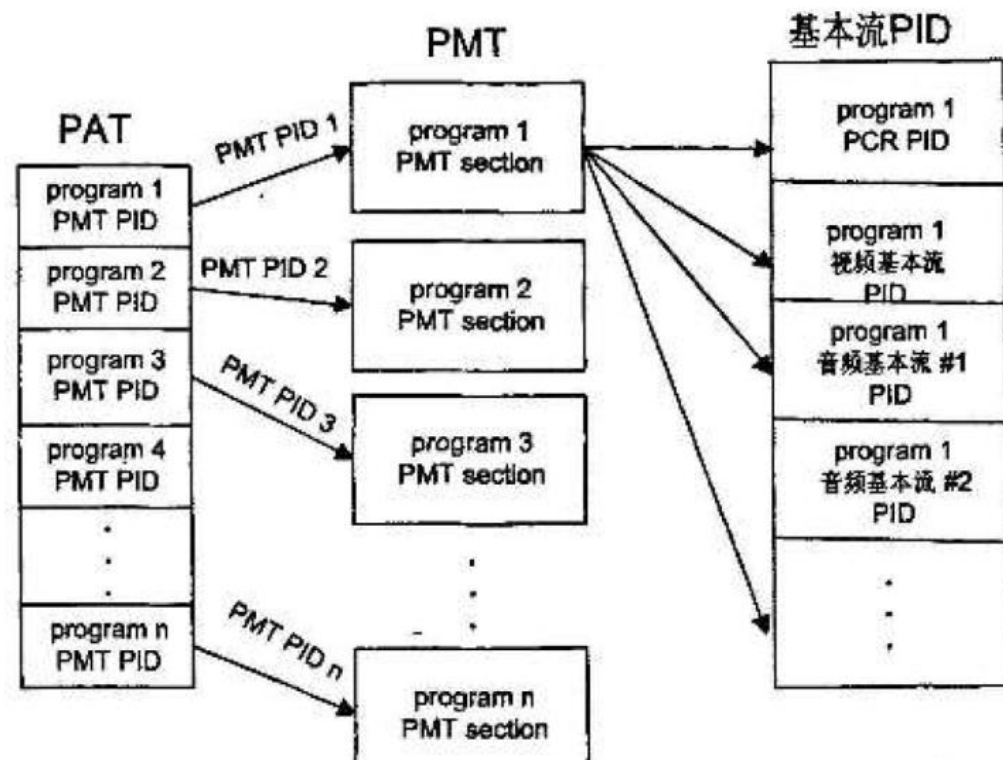
AFD not found in this packet

TS流总结

总结：

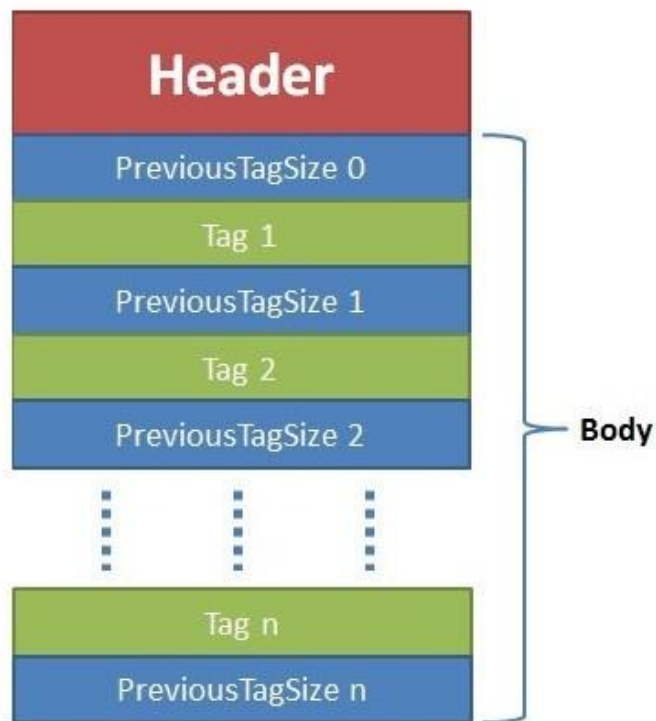
基本上从TS流分析H264码流这块，简单的方法就是先分析PID为0x00的PAT，然后分析PMT里面对音频和视频的PID定义。然后以定义PID分别过滤音频，然后再过滤视频，这样就完成了从TS文件到音视频数据的过滤。

- 1.第一步找PAT,注重分析PMT的表的PID；
- 2.第二步找PMT，分析里面是否含有音视频，音视频编码格式，音视频的PID；
- 3.第三步根据音视频的PID过滤音频TS包和视频TS包
- 4.拿掉TS头，PES头，后即可得到音频数据和视频数据的裸码流；



FLV 封装格式解析

FLV整体结构



Flv Header	Signature (3 字节) 为文件标识, 总为“FLV”, (0x46, 0x4c, 0x66)	
	Version (1 字节) 为版本, 目前为 0x01	
	Flags (1 字节) 前 5 位保留, 必须为 0。第 6 位表示是否存在音频 Tag。第 7 位保留, 必须为 0。第 8 位表示是否存在视频 Tag。	
	Headersize (4 字节) 为从 File Header 开始到 File Body 开始的字节数, 版本 1 中总为 9。	
Flv Body	Previous Tag Size #0 (4 字节) 表示前一个 Tag 的长度	
	Tag #1	Type (1 字节) 表示 Tag 类型, 包括音频 (0x08), 视频 (0x09) 和 script data (0x12), 其他类型值被保留
		Datasize (3 字节) 表示该 Tag Ddata 部分的大小
		Timestamp (3 字节) 表示该 Tag 的时间戳
		Timestamp_ex (1 字节) 表示时间戳的扩展字节, 当 24 位数值不够时, 该字节最为最高位将时间戳扩展为 32 位数值
		StreamID (3 字节) 表示 stream id 总是 0
	Tag Data	不同类型 Tag 的 data 部分结构各不相同, 当 header 的结构是相同的
	Previous Tag size #1 即 Tag #1 的大小 (11 + Datasize)	
	Tag #2	
	Previous Tag size #2	
	
	Tag #N	
	Previous Tag size #N	

flv header + flv body

flv header + previous size0 + tag1 + previous size1 + tag2 ++ prvious sizen+ tagn+1

flv header + previous size0 + tag1 header + tag1 data1 ++ previous sizen + tagn header+tagn data

previous tag sizen 表示的前一个tag的整体大小

每个Tag数据大小在Tag Header头里面的Data size字段里面；

实例分析

FlvParse

FLV structure

- File Header [0x00000000]
 - signature: FLV
 - version: 1
 - has audio: 1
 - has video: 1
 - header size: 9
- First Tag Size [0x00000009]: 0
- Metadata Tag [0x0000000D]
 - Tag Header [0x0000000D]
 - Tag Data [0x00000018]
- Pre Tag Size [0x00000144]: 311
- Video Tag1 [0x00000148]
 - Tag Header [0x00000148]
 - Tag Data [0x00000153]
- Pre Tag Size [0x0000017E]: 55
- Audio Tag2 [0x00000183]
 - Tag Header [0x00000183]
 - Tag Data [0x0000018E]
- Pre Tag Size [0x00000192]: 15
- Video Tag3 [0x00000196]
 - Tag Header [0x00000196]
 - Tag Data [0x000001A1]
- Pre Tag Size [0x00000399]: 515
- Audio Tag4 [0x0000039D]
 - Tag Header [0x0000039D]
 - Tag Data [0x000003A8]
- Pre Tag Size [0x000003B3]: 22
- Audio Tag5 [0x000003B7]
 - Tag Header [0x000003B7]
 - Tag Data [0x000003C2]
- Pre Tag Size [0x000003CD]: 22

选择要显示的信息类型: ☐ 高速模式

file header info, metadata info, tag info

分析时间:0s 更新界面时间:167s

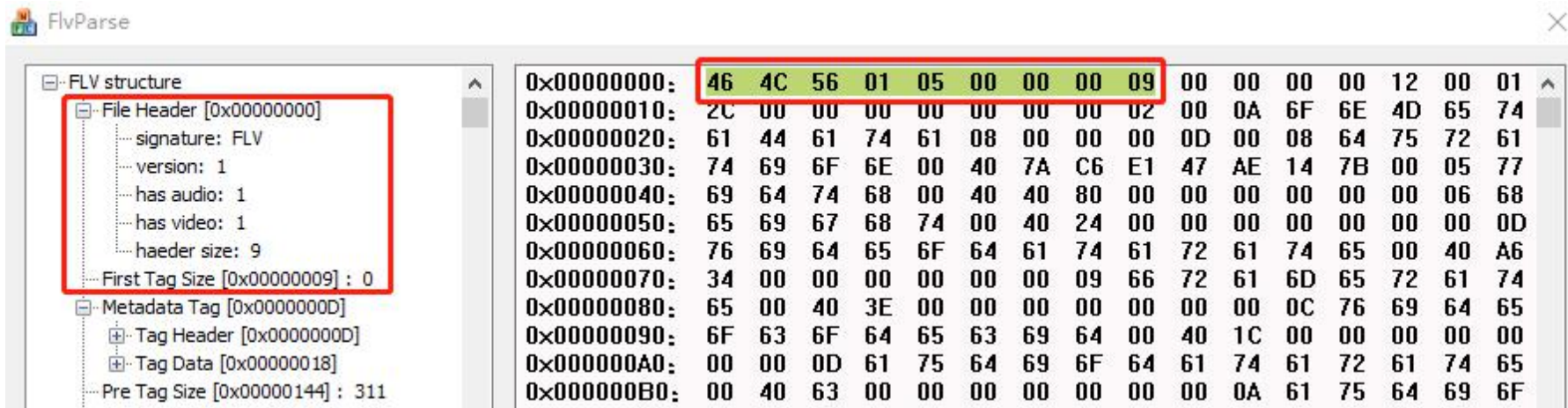
D:\Test_File\测试素材\flv文件\FLV_1920X1080_AVC_AAC_30_High@L4_Lonely.flv

浏览...

确定 取消

0x00000000:	46	4C	56	01	05	00	00	00	09	00	00	00	12	00	01
0x00000010:	2C	00	00	00	00	00	00	02	00	0A	6F	6E	4D	65	74
0x00000020:	61	44	61	74	61	08	00	00	0D	00	08	64	75	72	61
0x00000030:	74	69	6F	6E	00	40	7A	C6	E1	47	AE	14	7B	00	05
0x00000040:	69	64	74	68	00	40	80	00	00	00	00	00	00	06	68
0x00000050:	65	69	67	68	74	00	40	24	00	00	00	00	00	00	0D
0x00000060:	76	69	64	65	6F	64	61	74	61	72	61	74	65	00	40
0x00000070:	34	00	00	00	00	00	09	66	72	61	6D	65	72	61	74
0x00000080:	65	00	40	3E	00	00	00	00	00	00	0C	76	69	64	65
0x00000090:	6F	63	6F	64	65	63	69	64	00	40	1C	00	00	00	00
0x000000A0:	00	00	0D	61	75	64	69	6F	64	61	74	61	72	61	74
0x000000B0:	00	40	63	00	00	00	00	00	00	0A	61	75	64	69	6F
0x000000C0:	64	65	6C	61	79	00	00	00	00	00	00	00	00	00	0C
0x000000D0:	61	75	64	69	6F	63	6F	64	65	63	69	64	00	40	24
0x000000E0:	00	00	00	00	00	0C	63	61	6E	53	65	65	6B	54	6F
0x000000F0:	45	6E	64	01	01	00	0D	6C	61	73	74	74	69	6D	65
0x00000100:	74	61	6D	70	00	40	7A	C6	E1	47	AE	14	7B	00	0F
0x00000110:	65	74	61	64	61	74	61	63	72	65	61	74	6F	72	02
0x00000120:	0D	4D	6F	79	65	61	20	46	4C	56	20	4C	69	62	00
0x00000130:	63	75	65	50	6F	69	6E	74	73	08	00	00	00	00	00
0x00000140:	09	00	00	09	00	00	01	37	09	00	00	2C	00	00	00
0x00000150:	00	00	00	17	00	00	00	01	64	00	28	FF	E1	00	18
0x00000160:	67	64	00	28	AC	24	88	07	80	22	7E	58	40	00	FA
0x00000170:	40	00	3A	98	03	C6	0C	A8	01	00	04	68	EE	3C	B0
0x00000180:	00	00	37	08	00	00	04	00	00	00	00	00	00	00	AF
0x00000190:	12	10	00	00	00	0F	09	00	01	F8	00	00	00	00	00
0x000001A0:	00	17	01	00	00	00	00	01	EF	65	88	80	80	0B	FF
0x000001B0:	F1	02	73	FC	40	20	3B	F9	A2	B0	DD	3E	E7	3F	5B
0x000001C0:	B8	8F	A4	0C	9C	5D	3B	AB	6B	4B	80	00	00	03	00
0x000001D0:	23	54	CB	EE	00	00	04	01	63	84	1A	00	07	DF	B9
0x000001E0:	17	85	F0	D1	D8	90	00	04	93	27	5C	C0	33	9A	B0
0x000001F0:	00	DF	35	4C	C6	D0	00	66	03	D0	0A	80	07	D4	D5
0x00000200:	00	61	80	00	00	71	00	00	03	00	7F	00	00	03	00
0x00000210:	00	00	03	01	0D	00	00	03	02	02	00	00	03	03	14
0x00000220:	00	07	60	00	00	03	00	00	03	00	00	03	00	03	00

FLV Header



The screenshot shows the FlvParse application interface. On the left, the 'FLV structure' tree is expanded, showing the 'File Header [0x00000000]' with fields: signature: FLV, version: 1, has audio: 1, has video: 1, haeder size: 9, and First Tag Size [0x00000009]: 0. The 'Metadata Tag' section is also visible. On the right, the hex data is displayed in a table. The first row of data is highlighted with a green box, showing the first 9 bytes of the header: 46 4C 56 01 05 00 00 00 09.

Address	Hex Data
0x00000000:	46 4C 56 01 05 00 00 00 09 00 00 00 00 12 00 01
0x00000010:	2C 00 00 00 00 00 00 00 02 00 0A 6F 6E 4D 65 74
0x00000020:	61 44 61 74 61 08 00 00 00 0D 00 08 64 75 72 61
0x00000030:	74 69 6F 6E 00 40 7A C6 E1 47 AE 14 7B 00 05 77
0x00000040:	69 64 74 68 00 40 40 80 00 00 00 00 00 00 06 68
0x00000050:	65 69 67 68 74 00 40 24 00 00 00 00 00 00 0D
0x00000060:	76 69 64 65 6F 64 61 74 61 72 61 74 65 00 40 A6
0x00000070:	34 00 00 00 00 00 00 09 66 72 61 6D 65 72 61 74
0x00000080:	65 00 40 3E 00 00 00 00 00 00 00 0C 76 69 64 65
0x00000090:	6F 63 6F 64 65 63 69 64 00 40 1C 00 00 00 00 00
0x000000A0:	00 00 0D 61 75 64 69 6F 64 61 74 61 72 61 74 65
0x000000B0:	00 40 63 00 00 00 00 00 00 00 0A 61 75 64 69 6F

总结:

1. Flv header 的前三个字节是固定的FLV的 ASCII 码的值0x46 0x4C 0x56;
2. 接下来的一个字节表示 FLV 的版本号,例如 0x01 代表 FLV 版本号为 1;
3. 第 5 个字节中的第0位和第2位分别表示video和audio的存在情况（1表示存在，0 表示不存在）其余6位必须为0.最后的4字节表示FLV Header的长度，对于version 1，此处为9;
4. 一般判断格式是不是flv，先从收到数据的第一字节连续查找flv三个字符，如果找到一般可以判断是flv封装格式;
5. Header头数据一般是9字节但是不绝对，所以需要读最后的长度字段来解析;

FLV TAG Header

字段	占位	备注
Tag类型(TagType)	1 bytes	1-2bit位 ：必须为0，保留位； 第3bit位 ：0表示未加密，1表示加密，一般默认0； 4-8bit位 ：8：音频、9：视频、18：script数据；
数据大小(DataSize)	3 bytes	数据字段的长度，是Tag Data的长度，不包括11字节的Tag Header长度；
时间戳(Timestamp)	3 bytes	毫秒为单位，第一个tag时，该值总是0，单位是 毫秒 ，则意味着要将时间戳单位关系换算好；
时间戳扩展(TimeStampExtended)	1 bytes	时间戳扩展为4bytes, 代表高8位，很少用到；
流ID (Stream ID)	3bytes	总是0，暂时未用到，因为flv只封装一路音视频，但是对于TS和PS则有特殊含义；
数据(Data)	音频、视频或script	数据实体，音频、视频和脚本数据；

总结：

- 1.无论那种类型的tag，tag头字节都是11字节，要解析里面的音频帧，视频帧或者元数据需要读取tag头里面的data长度字段；
- 2.时间戳很关键，播放过程中，FLV tag的时间信息完全依赖于 FLV 时间戳，内置的其他时间信息都被忽略掉,一般非音视频的tag，时间戳就是0即可；
- 3.注意计算好时间戳大小，这里的单位是毫秒，所以一定要根据采样率和视频帧率，音频帧采样多少计算好时间戳，然后还要换算成毫秒；
- 4.Tag头解析完后，Tag数据部分不是视频帧和音频帧，还要根据H264和AAC的打包方案解析才能得到真实的音视频裸数据；

Script TAG 脚本元数据TAG

FlvParse

Metadata Tag [0x0000000D]

Tag Header [0x0000000D]

- type: 18
- data size: 300
- time stamp: 0
- stream id: 0

Tag Data [0x00000018]

- amf 1 type: 2
- string size: 10
- string: onMetaData
- amf 2 type: 8
- array size: 13
- metadata array data
 - duration: 428.43
 - width: 33.00
 - height: 10.00
 - videodatarate: 2842.00
 - framerate: 30.00
 - videocodecid: 7.00
 - audiodatarate: 152.00
 - audiodelay: 0.00
 - audiocodecid: 10.00
 - canSeekToEnd: 1
 - lasttimestamp: 428.43
 - metadatacreator: Moyea FLV

Pre Tag Size [0x00000144]: 311

选择要显示的信息类型: ☐ 高速模式

file header info, metadata info, tag info

分析时间: 0s 更新界面时间: 167s

D:\Test_File\测试素材\flv文件\FLV_1920X1080_AVC_AAC_30_High@L4_Lonely.flv

确定 取消

0x00000000: 46 4C 56 01 05 00 00 00 09 00 00 00 12 00 01
0x00000010: 2C 00 00 00 00 00 00 00 02 00 0A 6F 6E 4D 65 74
0x00000020: 61 44 61 74 61 08 00 00 00 0D 00 08 64 75 72 61
0x00000030: 74 69 6F 6E 00 40 7A C6 E1 47 AE 14 7B 00 05 77
0x00000040: 69 64 74 68 00 40 40 80 00 00 00 00 00 00 06 68
0x00000050: 65 69 67 68 74 00 40 24 00 00 00 00 00 00 00 0D
0x00000060: 76 69 64 65 6F 64 61 74 61 72 61 74 65 00 40 A6
0x00000070: 34 00 00 00 00 00 00 09 66 72 61 6D 65 72 61 74
0x00000080: 65 00 40 3E 00 00 00 00 00 00 0C 76 69 64 65
0x00000090: 6F 63 6F 64 65 63 69 64 00 40 1C 00 00 00 00 00
0x000000A0: 00 00 0D 61 75 64 69 6F 64 61 74 61 72 61 74 65
0x000000B0: 00 40 63 00 00 00 00 00 00 00 0A 61 75 64 69 6F
0x000000C0: 64 65 6C 61 79 00 00 00 00 00 00 00 00 00 0C
0x000000D0: 61 75 64 69 6F 63 6F 64 65 63 69 64 00 40 24 00
0x000000E0: 00 00 00 00 00 00 0C 63 61 6E 53 65 65 6B 54 6F
0x000000F0: 45 6E 64 01 01 00 0D 6C 61 73 74 74 69 6D 65 73
0x00000100: 74 61 6D 70 00 40 7A C6 E1 47 AE 14 7B 00 0F 6D
0x00000110: 65 74 61 64 61 74 61 63 72 65 61 74 6F 72 02 00
0x00000120: 0D 4D 6F 79 65 61 20 46 4C 56 20 4C 69 62 00 09
0x00000130: 63 75 65 50 6F 69 6E 74 73 08 00 00 00 00 00 00
0x00000140: 09 00 00 09 00 00 01 37 09 00 00 2C 00 00 00 00
0x00000150: 00 00 00 17 00 00 00 00 01 64 00 28 FF E1 00 18
0x00000160: 67 64 00 28 AC 24 88 07 80 22 7E 58 40 00 00 FA
0x00000170: 40 00 3A 98 03 C6 0C A8 01 00 04 68 EE 3C B0 00
0x00000180: 00 00 37 08 00 00 04 00 00 00 00 00 00 00 AF 00
0x00000190: 12 10 00 00 00 0F 09 00 01 F8 00 00 00 00 00 00
0x000001A0: 00 17 01 00 00 00 00 00 01 EF 65 88 80 80 0B FF
0x000001B0: F1 02 73 FC 40 20 3B F9 A2 B0 DD 3E E7 3F 5B 6D
0x000001C0: B8 8F A4 0C 9C 5D 3B AB 6B 4B 80 00 00 03 00 03
0x000001D0: 23 54 CB EE 00 00 04 01 63 84 1A 00 07 DF B9 98
0x000001E0: 17 85 F0 D1 D8 90 00 04 93 27 5C C0 33 9A B0 A0
0x000001F0: 00 DF 35 4C C6 D0 00 66 03 D0 0A 80 07 D4 D5 80
0x00000200: 00 61 80 00 00 71 00 00 03 00 7F 00 00 03 00 99
0x00000210: 00 00 03 01 0D 00 00 03 02 02 00 00 03 03 14 00
0x00000220: 00 07 60 00 00 03 00 00 03 00 00 03 00 03 00

1. Metadata Tag有下面两种包组成:

AMF1{"onMetaData"}|AMF2{"width height"}

字段	占位	备注
AMF Type	1byte 字节	固定值为0x08
AMF 长度	4byte 字节	数组元素的个数: 0x00 00 00 0D
AMF value值	计算得 到字节	因为在该tag的头已经说明该tag的data 有300字节, 则用该值减去AMF1长度13 字节和AMF2头的5字节即为剩余AMF2 的Value值大小: 300 - 13 - 5 = 283字 节

值	含义
duration	时长
width	视频宽度
height	视频高度
videodatarate	视频码率
framerate	视频帧率
videocodecid	视频编码方式
audiosamplerate	音频采样率
audiosamplesize	音频采样精度
stereo	是否为立体声
audiocodecid	音频编码方式
filesize	文件大小

Script TAG 脚本元数据TAG

MediaArea.net/MediaInfo - C:\Users\ty6815\Desktop\welcometobj.flv

文件(F) 查看(V) 选项(X) 调试(W) 帮助(H) 语言(U) | 有新版可用(S)

C:\Users\ty6815\Desktop\welcometobj.flv

概要

- 完整名称: C:\Users\ty6815\Desktop\welcometobj.flv
- 格式: Flash Video
- 文件大小: 9.60 MiB
- 持续时间: 1 分 49 秒
- 平均混合码率: 738 kb/s
- 编码程序: Lavf58.28.102
- date: 07/29/12 07:51:34
- album: Yinyuetai
- artist: yinyuetai.com
- comment: Yinyuetai Fukai
- liveXML: <?xml begin="" id="W5M0MpCehiHzreSzNTczkc9d">

视频

- 格式: AVC
- 格式/信息: Advanced Video Codec
- 格式概况: Main@L2.1
- 格式设置: CABAC / 8 Ref Frames
- 格式设置: CABAC: 是
- 格式设置: 参考帧: 8 帧
- 编解码器ID: 7
- 持续时间: 1 分 49 秒
- 码率: 680 kb/s
- 宽度: 448 像素
- 高度: 336 像素
- 画面比例: 4:3
- 帧率模式: 恒定
- 帧率: 15.000 帧/秒
- 色彩空间: YUV
- 色度抽样: 4:2:0
- 位深: 8 位
- 扫描方式: 逐行扫描
- 数据密度【码率/(像素*帧率)】: 0.301
- 流大小: 8.84 MiB (92%)
- 编解码器: x264 core 115 r1995 c1e60b9
- 编解码设置: cabac=1 / ref=8 / deblock=1:-1:-1 / analyse=0x1:0x131 / me=umh / subme=9 / psy=1 / psy_rd=1.00:0.15 /

音频

- 格式: AAC LC SBR
- 格式/信息: Advanced Audio Codec Low Complexity with Spectral Band Replication
- 格式名称: HE-AAC
- 格式设置: Explicit
- 编解码器ID: 10-2
- 持续时间: 1 分 49 秒
- 码率: 62.5 kb/s
- 声道: 2 声道
- ChannelLayout: L R
- 采样率: 24.0 kHz
- 帧率: 11.719 帧/秒 (2048 SPF)
- 压缩模式: 有损压缩
- 流大小: 832 KiB (8%)

FlvParse

Tag Data [0x00000018]

- amf 1 type: 2
- string size: 10
- string: onMetaData
- amf 2 type: 8
- array size: 20

metadata array data

- duration: 109.13
- width: 448.00
- height: 336.00
- videodatarate: 663.21
- framerate: 15.00
- videocodecid: 7.00
- audiodatarate: 62.51
- audiosamplerate: 24000.00
- audiosamplesize: 16.00
- stereo: 1
- audiocodecid: 10.00
- major_brand: isom
- minor_version: 1
- compatible_brands: isomavc1
- date: 07/29/12 07:51:34
- album: Yinyuetai
- artist: yinyuetai.com
- comment: Yinyuetai Fukai
- encoder: Lavf58.28.102
- filesize: 10059461.00

选择要显示的信息类型: ☐ 高速模式

file header info, metadata info, tag info

分析时间:0s 更新界面时间:30s

0x00000010:	CA	00	00	00	00	00	00	00	02	00	0A	6F	6E	4D	65	74
0x00000020:	61	44	61	74	61	08	00	00	00	14	00	08	64	75	72	61
0x00000030:	74	69	6F	6E	00	40	5B	48	93	74	BC	6A	7F	00	05	77
0x00000040:	69	64	74	68	00	40	7C	00	00	00	00	00	00	00	06	68
0x00000050:	65	69	67	68	74	00	40	75	00	00	00	00	00	00	00	0D
0x00000060:	76	69	64	65	6F	64	61	74	61	72	61	74	65	00	40	84
0x00000070:	B9	AE	00	00	00	00	00	09	66	72	61	6D	65	72	61	74
0x00000080:	65	00	40	2E	00	00	00	00	00	00	0C	76	69	64	65	
0x00000090:	6F	63	6F	64	65	63	69	64	00	40	1C	00	00	00	00	00
0x000000A0:	00	00	0D	61	75	64	69	6F	64	61	74	61	72	61	74	65
0x000000B0:	00	40	4F	41	40	00	00	00	00	0F	61	75	64	69	6F	
0x000000C0:	73	61	6D	70	6C	65	72	61	74	65	00	40	D7	70	00	00
0x000000D0:	00	00	00	00	0F	61	75	64	69	6F	73	61	6D	70	6C	65
0x000000E0:	73	69	7A	65	00	40	30	00	00	00	00	00	00	00	06	73
0x000000F0:	74	65	72	65	6F	01	01	00	0C	61	75	64	69	6F	63	6F
0x00000100:	64	65	63	69	64	00	40	24	00	00	00	00	00	00	00	0B
0x00000110:	6D	61	6A	6F	72	5F	62	72	61	6E	64	02	00	04	69	73
0x00000120:	6F	6D	00	0D	6D	69	6E	6F	72	5F	76	65	72	73	69	6F
0x00000130:	6E	02	00	01	31	00	11	63	6F	6D	70	61	74	69	62	6C
0x00000140:	65	5F	62	72	61	6E	64	73	02	00	08	69	73	6F	6D	61
0x00000150:	76	63	31	00	04	64	61	74	65	02	00	11	30	37	2F	32
0x00000160:	39	2F	31	32	20	30	37	3A	35	31	3A	33	34	00	05	61
0x00000170:	6C	62	75	6D	02	00	09	59	69	6E	79	75	65	74	61	69
0x00000180:	00	06	61	72	74	69	73	74	02	00	0D	79	69	6E	79	75
0x00000190:	65	74	61	69	2E	63	6F	6D	00	07	63	6F	6D	6D	65	6E
0x000001A0:	74	02	00	0F	59	69	6E	79	75	65	74	61	69	20	46	75
0x000001B0:	6B	61	69	00	07	65	6E	63	6F	64	65	72	02	00	0D	4C
0x000001C0:	61	76	66	35	38	2E	32	38	2E	31	30	32	00	08	66	69
0x000001D0:	6C	65	73	69	7A	65	00	41	63	2F	D8	A0	00	00	00	00
0x000001E0:	00	09	00	00	01	D5	12	00	0C	B7	00	00	00	00	00	00
0x000001F0:	00	02	00	09	6F	6E	58	4D	50	44	61	74	61	08	00	00
0x00000200:	00	01	00	07	6C	69	76	65	58	4D	4C	02	0C	97	3C	3F
0x00000210:	78	70	61	63	6B	65	74	20	62	65	67	69	6E	3D	22	EF
0x00000220:	BB	BF	22	20	69	64	3D	22	57	35	4D	30	4D	70	43	65
0x00000230:	68	69	48	7A	72	65	53	7A	4E	54	63	7A	6B	63	39	64

C:\Users\ty6815\Desktop\welcometobj.flv

浏览...

确定 取消

Video TAG 视频TAG

FlvParse

FLV structure

- File Header [0x00000000]
 - First Tag Size [0x00000009]: 0
- Metadata Tag [0x00000000]
 - Pre Tag Size [0x00000144]: 311
- Video Tag1 [0x00000148]
 - Tag Header [0x00000148]
 - type: 9
 - data size: 44
 - time stamp: 0
 - stream id: 0
 - Tag Data [0x00000153]
 - frame type: 16
 - codec id: 7
 - packet addr: 0x13460154
 - Pre Tag Size [0x0000017F]: 55
- Audio Tag2 [0x00000183]
 - Pre Tag Size [0x00000192]: 15
- Video Tag3 [0x00000196]
 - Tag Header [0x00000196]
 - type: 9
 - data size: 504
 - time stamp: 0
 - stream id: 0
 - Tag Data [0x000001A1]
 - frame type: 16
 - codec id: 7
 - packet addr: 0x134601A2

选择要显示的信息类型: ☐ 高速模式

file header info, metadata info, tag info

分析时间: 0s 更新界面时间: 167s

D:\Test_File\测试素材\flv文件\FLV_1920X1080_AVC_AAC_30_High@L4_Lonely.flv

确定 取消

0x00000000:	46	4C	56	01	05	00	00	00	09	00	00	00	00	12	00	01
0x00000010:	2C	00	00	00	00	00	00	02	00	0A	6F	6E	4D	65	74	
0x00000020:	61	44	61	74	61	08	00	00	0D	00	08	64	75	72	61	
0x00000030:	74	69	6F	6E	00	40	7A	C6	E1	47	AE	14	7B	00	05	77
0x00000040:	69	64	74	68	00	40	80	00	00	00	00	00	06	68		
0x00000050:	65	69	67	68	74	00	40	24	00	00	00	00	00	00	0D	
0x00000060:	76	69	64	65	6F	64	61	74	61	72	61	74	65	00	40	A6
0x00000070:	34	00	00	00	00	00	09	66	72	61	6D	65	72	61	74	
0x00000080:	65	00	40	3E	00	00	00	00	00	00	0C	76	69	64	65	
0x00000090:	6F	63	6F	64	65	63	69	64	00	40	1C	00	00	00	00	
0x000000A0:	00	00	0D	61	75	64	69	6F	64	61	74	61	72	61	74	65
0x000000B0:	00	40	63	00	00	00	00	00	00	0A	61	75	64	69	6F	
0x000000C0:	64	65	6C	61	79	00	00	00	00	00	00	00	00	00	00	0C
0x000000D0:	61	75	64	69	6F	63	6F	64	65	63	69	64	00	40	24	00
0x000000E0:	00	00	00	00	00	00	0C	63	61	6E	53	65	65	6B	54	6F
0x000000F0:	45	6E	64	01	01	00	0D	6C	61	73	74	74	69	6D	65	73
0x00000100:	74	61	6D	70	00	40	7A	C6	E1	47	AE	14	7B	00	0F	6D
0x00000110:	65	74	61	64	61	74	61	63	72	65	61	74	6F	72	02	00
0x00000120:	0D	4D	6F	79	65	61	2D	46	4C	56	20	4C	69	62	00	09
0x00000130:	63	75	65	50	6F	69	6E	74	73	08	00	00	00	00	00	00
0x00000140:	09	00	00	09	00	00	01	37	09	00	00	2C	00	00	00	00
0x00000150:	00	00	00	17	00	00	00	00	01	64	00	28	FF	E1	00	18
0x00000160:	67	64	00	28	AC	24	88	07	80	22	7E	58	40	00	00	FA
0x00000170:	40	00	3A	98	03	C6	0C	A8	01	00	04	68	EE	3C	B0	00
0x00000180:	00	00	37	08	00	00	04	00	00	00	00	00	00	00	AF	00
0x00000190:	12	10	00	00	0F	09	00	01	F8	00	00	00	00	00	00	00
0x000001A0:	00	17	01	00	00	00	00	00	01	EF	65	88	80	80	0B	FF
0x000001B0:	F1	02	73	FC	40	20	3B	F9	A2	B0	DD	3E	E7	3F	5B	6D
0x000001C0:	B8	8F	A4	0C	9C	5D	3B	AB	6B	4B	80	00	00	03	00	03
0x000001D0:	23	54	CB	EE	00	00	04	01	63	84	1A	00	07	DF	B9	98
0x000001E0:	17	85	F0	D1	D8	90	00	04	93	27	5C	C0	33	9A	B0	A0
0x000001F0:	00	DF	35	4C	C6	D0	00	66	03	D0	0A	80	07	D4	D5	80
0x00000200:	00	61	80	00	00	71	00	00	03	00	7F	00	00	03	00	99
0x00000210:	00	00	03	01	0D	00	00	03	02	02	00	00	03	03	14	00
0x00000220:	00	07	60	00	00	03	00	00	03	00	00	03	00	00	03	00

Video TAG Header

FlvParse

FLV structure

- File Header [0x00000000]
 - First Tag Size [0x00000009]: 0
- Metadata Tag [0x00000000]
 - Pre Tag Size [0x00000144]: 311
- Video Tag1 [0x00000148]
 - Tag Header [0x00000148]
 - frame type: 16
 - codec id: 7
 - packet addr: 0x10A30154
 - Pre Tag Size [0x0000017F]: 55
- Audio Tag2 [0x00000183]
 - Pre Tag Size [0x00000192]: 15
- Video Tag3 [0x00000196]
 - Pre Tag Size [0x00000399]: 515
- Audio Tag4 [0x0000039D]
 - Pre Tag Size [0x000003B3]: 22
- Audio Tag5 [0x000003B7]
 - Pre Tag Size [0x000003CD]: 22
- Video Tag6 [0x000003D1]
 - Pre Tag Size [0x0000042F]: 94
- Audio Tag7 [0x00000433]
 - Pre Tag Size [0x00000449]: 22
- Video Tag8 [0x0000044D]
 - Pre Tag Size [0x000004A8]: 91
- Audio Tag9 [0x000004AC]
 - Pre Tag Size [0x000004C7]: 22

选择要显示的信息类型: ☐ 高速模式

file header info, metadata info, tag info

分析时间: 0s 更新界面时间: 166s

0x00000000:	46	4C	56	01	05	00	00	00	09	00	00	00	00	12	00	01
0x00000010:	2C	00	00	00	00	00	00	02	00	0A	6F	6E	4D	65	74	
0x00000020:	61	44	61	74	61	08	00	00	0D	00	08	64	75	72	61	
0x00000030:	74	69	6F	6E	00	40	7A	C6	E1	47	AE	14	7B	00	05	77
0x00000040:	69	64	74	68	00	40	80	00	00	00	00	00	00	00	06	68
0x00000050:	65	69	67	68	74	00	40	24	00	00	00	00	00	00	00	0D
0x00000060:	76	69	64	65	6F	64	61	74	61	72	61	74	65	00	40	A6
0x00000070:	34	00	00	00	00	00	09	66	72	61	6D	65	72	61	74	
0x00000080:	65	00	40	3E	00	00	00	00	00	00	0C	76	69	64	65	
0x00000090:	6F	63	6F	64	65	63	69	64	00	40	1C	00	00	00	00	00
0x000000A0:	00	00	0D	61	75	64	69	6F	64	61	74	61	72	61	74	65
0x000000B0:	00	40	63	00	00	00	00	00	00	0A	61	75	64	69	6F	
0x000000C0:	64	65	6C	61	79	00	00	00	00	00	00	00	00	00	00	0C
0x000000D0:	61	75	64	69	6F	63	6F	64	65	63	69	64	00	40	24	00
0x000000E0:	00	00	00	00	00	00	0C	63	61	6E	53	65	65	6B	54	6F
0x000000F0:	45	6E	64	01	01	00	0D	6C	61	73	74	74	69	6D	65	73
0x00000100:	74	61	6D	70	00	40	7A	C6	E1	47	AE	14	7B	00	0F	6D
0x00000110:	65	74	61	64	61	74	61	63	72	65	61	74	6F	72	02	00
0x00000120:	0D	4D	6F	79	65	61	2D	46	4C	56	20	4C	69	62	00	09
0x00000130:	63	75	65	50	6F	69	6E	74	73	08	00	00	00	00	00	00
0x00000140:	09	00	00	09	00	00	01	37	09	00	00	2C	00	00	00	00
0x00000150:	00	00	00	17	00	00	00	00	01	64	00	28	FF	E1	00	18
0x00000160:	67	64	00	28	AC	24	88	07	80	22	7E	58	40	00	00	FA
0x00000170:	40	00	3A	98	03	C6	0C	A8	01	00	04	68	EE	3C	B0	00
0x00000180:	00	00	37	08	00	00	04	00	00	00	00	00	00	00	AF	00
0x00000190:	12	10	00	00	0F	09	00	01	F8	00	00	00	00	00	00	00
0x000001A0:	00	17	01	00	00	00	00	00	01	EF	65	88	80	80	0B	FF
0x000001B0:	F1	02	73	FC	40	20	3B	F9	A2	B0	DD	3E	E7	3F	5B	6D
0x000001C0:	B8	8F	A4	0C	9C	5D	3B	AB	6B	4B	80	00	00	03	00	03
0x000001D0:	23	54	CB	EE	00	00	04	01	63	84	1A	00	07	DF	B9	98
0x000001E0:	17	85	F0	D1	D8	90	00	04	93	27	5C	C0	33	9A	B0	A0
0x000001F0:	00	DF	35	4C	C6	D0	00	66	03	D0	0A	80	07	D4	D5	80
0x00000200:	00	61	80	00	00	71	00	00	03	00	7F	00	00	03	00	99
0x00000210:	00	00	03	01	0D	00	00	03	02	02	00	00	03	03	14	00
0x00000220:	00	07	60	00	00	03	00	00	03	00	00	03	00	00	03	00

Video TAG Data

Audio TAG 音频TAG

FlvParse

FLV structure

- File Header [0x00000000]
- First Tag Size [0x00000009] : 0
- Metadata Tag [0x00000000]
- Pre Tag Size [0x00000144] : 311
- Video Tag1 [0x00000148]
- Pre Tag Size [0x0000017F] : 55
- Audio Tag2 [0x00000183]
- Tag Header [0x00000183]
 - type: 8
 - data size: 4
 - time stamp: 0
 - stream id: 0
- Tag Data [0x0000018E]
 - format: 160
 - rate: 12
 - sample size: 2
 - type: 1
 - data addr: 0x10A3018F
- Pre Tag Size [0x00000192] : 15
- Video Tag3 [0x00000196]
- Pre Tag Size [0x00000399] : 515
- Audio Tag4 [0x0000039D]
- Pre Tag Size [0x000003B3] : 22
- Audio Tag5 [0x000003B7]
- Pre Tag Size [0x000003CD] : 22
- Video Tag6 [0x000003D1]
- Pre Tag Size [0x0000042F] : 94
- Audio Tag7 [0x00000433]
- Pre Tag Size [0x00000449] : 22
- Video Tag8 [0x0000044D]
- Pre Tag Size [0x000004A8] : 91

选择要显示的信息类型: ☐ 高速模式

file header info, metadata info, tag info

分析时间:0s 更新界面时间:166s

0x00000140:	09	00	00	09	00	00	01	37	09	00	00	2C	00	00	00	00
0x00000150:	00	00	00	17	00	00	00	01	64	00	28	FF	E1	00	18	
0x00000160:	67	64	00	28	AC	24	88	07	80	22	7E	58	40	00	FA	
0x00000170:	40	00	3A	98	03	C6	0C	A8	01	00	04	68	EE	3C	B0	00
0x00000180:	00	00	37	08	00	00	04	00	00	00	00	00	00	00	AF	00
0x00000190:	12	10	00	00	00	0F	09	00	01	F8	00	00	00	00	00	00
0x000001A0:	00	17	01	00	00	00	00	01	EF	65	88	80	80	0B	FF	
0x000001B0:	F1	02	73	FC	40	20	3B	F9	A2	B0	DD	3E	E7	3F	5B	6D
0x000001C0:	B8	8F	A4	0C	9C	5D	3B	AB	6B	4B	80	00	00	03	00	03
0x000001D0:	23	54	CB	EE	00	00	04	01	63	84	1A	00	07	DF	B9	98
0x000001E0:	17	85	F0	D1	D8	90	00	04	93	27	5C	C0	33	9A	B0	A0
0x000001F0:	00	DF	35	4C	C6	D0	00	66	03	D0	0A	80	07	D4	D5	80
0x00000200:	00	61	80	00	00	71	00	00	03	00	7F	00	00	03	00	99
0x00000210:	00	00	03	01	0D	00	00	03	02	02	00	00	03	03	14	00
0x00000220:	00	07	60	00	00	03	00	00	03	00	00	03	00	00	03	00
0x00000230:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000240:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000250:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x00000260:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000270:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000280:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x00000290:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x000002A0:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x000002B0:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x000002C0:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x000002D0:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x000002E0:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x000002F0:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000300:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000310:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x00000320:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000330:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000340:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x00000350:	00	07	97	B0	C7	1B	A6	2E	59	44	8A	11	0F	DC	9B	25
0x00000360:	FC	63	DF	F2	42	EB	FB	E4	D5	ED	CB	FD	2C	DD	72	DF

Audio TAG Header

FlvParse

FLV structure

- File Header [0x00000000]
- First Tag Size [0x00000009] : 0
- Metadata Tag [0x00000000]
- Pre Tag Size [0x00000144] : 311
- Video Tag1 [0x00000148]
- Pre Tag Size [0x0000017F] : 55
- Audio Tag2 [0x00000183]
- Pre Tag Size [0x00000192] : 15
- Video Tag3 [0x00000196]
- Pre Tag Size [0x00000399] : 515
- Audio Tag4 [0x0000039D]
- Tag Header [0x0000039D]
 - format: 160
 - rate: 12
 - sample size: 2
 - type: 1
 - data addr: 0x10A303A9
- Pre Tag Size [0x000003B3] : 22
- Audio Tag5 [0x000003B7]
- Pre Tag Size [0x000003CD] : 22
- Video Tag6 [0x000003D1]
- Pre Tag Size [0x0000042F] : 94
- Audio Tag7 [0x00000433]
- Pre Tag Size [0x00000449] : 22
- Video Tag8 [0x0000044D]
- Pre Tag Size [0x000004A8] : 91

选择要显示的信息类型: ☐ 高速模式

file header info, metadata info, tag info

分析时间:0s 更新界面时间:166s

0x00000390:	1D	1C	08	D8	E7	8E	A0	7E	29	00	00	02	03	08	00	00
0x000003A0:	0B	00	00	00	00	00	00	00	00	AF	01	21	00	49	90	02
0x000003B0:	00	23	80	00	00	00	16	08	00	00	0B	00	00	17	00	00
0x000003C0:	00	00	AF	01	21	00	49	90	02	19	00	23	80	00	00	00
0x000003D0:	16	09	00	00	53	00	00	21	00	00	00	00	27	01	00	00
0x000003E0:	00	00	00	00	4A	41	9A	04	0E	57	00	00	03	00	00	03
0x000003F0:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x00000400:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000410:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000420:	00	00	D3	DC	C9	6D	83	9D	AD	1D	40	51	EA	62	03	00
0x00000430:	00	00	5E	08	00	00	0B	00	00	2E	00	00	00	00	AF	01
0x00000440:	21	00	49	90	02	19	00	23	80	00	00	00	16	09	00	00
0x00000450:	50	00	00	43	00	00	00	00	27	01	00	00	00	00	00	00
0x00000460:	47	41	9A	08	16	57	00	00	03	00	00	03	00	00	03	00
0x00000470:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000480:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000490:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	C4	DE
0x000004A0:	9E	44	A9	1B	75	F4	97	E1	00	00	00	5B	08	00	00	0B
0x000004B0:	00	00	46	00	00	00	00	AF	01	21	00	49	90	02	19	00
0x000004C0:	23	80	00	00	16	08	00	00	0B	00	00	5D	00	00	00	00
0x000004D0:	00	AF	01	21	00	49	90	02	19	00	23	80	00	00	00	16
0x000004E0:	09	00	00	50	00	00	64	00	00	00	00	27	01	00	00	00
0x000004F0:	00	00	00	47	41	9A	0C	1E	57	00	00	03	00	00	03	00
0x00000500:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000510:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000520:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x00000530:	00	C4	E7	7C	00	07	D2	34	29	2F	C0	00	00	00	5B	08
0x00000540:	00	00	0B	00	00	74	00	00	00	00	AF	01	21	00	49	90
0x00000550:	02	19	00	23	80	00	00	00	16	09	00	00	00	49	00	85
0x00000560:	00	00	00	00	27	01	00	00	00	00	00	00	40	41	9A	10
0x00000570:	26	57	00	00	03	00	00	03	00	00	03	00	00	03	00	00
0x00000580:	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03
0x00000590:	00	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00
0x000005A0:	00	03	00	00	03	00	00	03	00	00	03	00	00	03	00	00

Audio TAG Data
AAC Sequence Header|AAC Raw Data

FLV总结

- 1.FLV就三种TAG，比较简单的封装格式，每个TAG都是由Tag Header和TAG Data组成；
- 2.FLV中对时间戳的处理？如果含有B帧时间戳怎么计算？特别是PTS和DTS时间戳；
- 3.你能从FLV这种封装格式中，借鉴到什么？加入让你设计一种私有封装格式，又该如何设计；
- 4.如果FLV里面扩展H.265视频编码格式，你觉得大概的思路是什么；
- 5.你觉得FLV的缺点有哪些；

参考文档

1. ISO/IEC 13818

2. ISO/IEC 14496

3. video_file_format_spec_v10_1.pdf