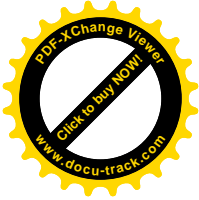
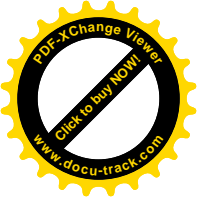


# TS 系统解码

- TS 系统解码.....1
  - 1 TS 流的内容.....2
    - 1.1 ES.....2
    - 1.2 PES.....2
    - 1.3 TS.....4
  - 2 TS 流的解码过程（只考虑本地文件，解码只涉及到系统层打包的解码，不涉及到编码的解码）.....5
    - 2.1 PAT 解码： .....5
    - 2.2 PMT 解码 .....6
    - 2.3 音视频解码.....6
  - 3 TS 流解码.....7
    - 3.1 总用例图.....7
    - 3.2 各解码类的 UML 图.....8
      - 3.2.1 解码主类.....8
      - 3.2.2 事件管理类.....9
      - 3.2.3 各类包解析的关系图.....10
      - 3.2.4 输入输出类.....11
      - 3.2.5 解析 PAT、PMT、私有数据等.....12
      - 3.2.6 解析 TS 包.....13
      - 3.2.7 解析 PES 包.....14
      - 3.2.8 PCR 时钟类.....15
      - 3.2.9 CRC 校验类.....15
    - 3.3 解码流程图.....16
      - 3.3.1 TS 包解析流程图.....16
      - 3.3.2 Section 解码流程图.....17
      - 3.3.3 PES 解码流程图.....18
  - 4 TS 流合成（编码）.....19
    - 4.1 各编码类的 UML 图.....19
      - 4.1.1 编码主类.....19
      - 4.1.2 合成各类之间的关系.....20
    - 4.2 编码主要流程.....21
      - 4.2.1 TS 编码流程图.....21
      - 4.2.2 负载（pes、section）打包过程.....22
  - 附表一： .....23
  - 附表二： .....25



# 1 TS 流的内容

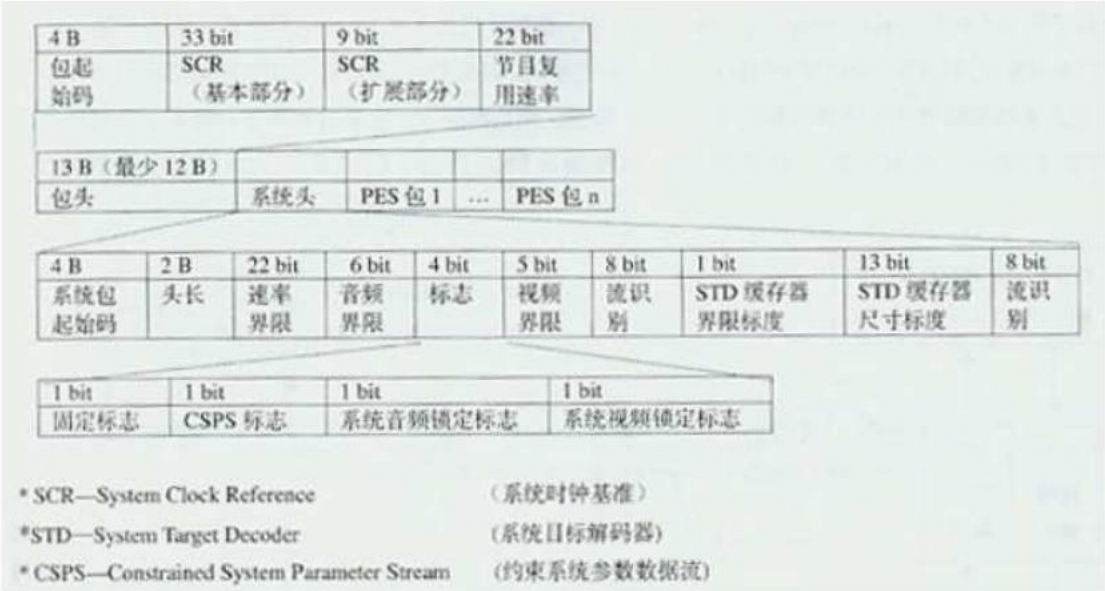
TS->PES->ES->NAL

## 1.1 ES

An elementary stream (ES) is defined by MPEG communication protocol is usually the output of an audio or video encoder.

## 1.2 PES

allows an Elementary stream to be divided into packets.



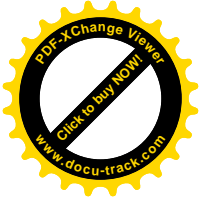
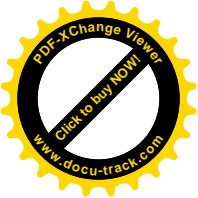
图一：PES 结构图

PES 结构各字段的具体含义见附表一。

由图一可见，1 个 PES 包是由包头、ES 特有信息和包数据 3 个部分组成。由于包头和 ES 特有信息二者可合成 1 个数据头，所以可认为 1 个 PES 包是由数据头和包数据（有效载荷）两个部分组成的。

包头由起始码前缀、数据流识别及 PES 包长信息 3 部分构成。包起始码前缀是用 23 个连续“0”和 1 个“1”构成的，用于表示有用信息种类的数据流识别，是 1 个 8 bit 的整数。由二者合成 1 个专用的包起始码，可用于识别数据包所属数据流（视频，音频，或其它）的性质及序号。例如：

比特序 1 1 0 xxxxx 是号码为 xxxxx 的 MPEG-2 音频数据流； 比特序 1 1 1 0 xxxxx 是号码



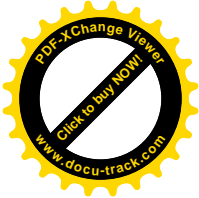
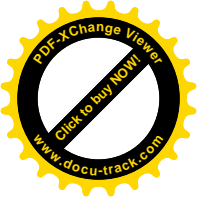
为xxxx的 MPEG-2 视频数据流。

PES 包长用于包长识别,表明在此字段后的字节数。如, PES 包长识别为 2 B, 即  $2 \times 8 = 16$  bit 字宽,包总长为  $216 - 1 = 65535$  B, 分给数据头 9 B (包头 6 B + ES 特有信息 3 B), 可变长度的包数据最大容量为 65526 B。尽管 PES 包最大长度可达  $(216 - 1) = 65535$  B(Byte), 但在通常的情况下是组成 ES 的若干个 AU 中的由头部和编码数据两部分组成的 1 个 AU 长度。1 个 AU 相当于编码的 1 幅视频图像或 1 个音频帧, 参见图一右上角从 ES 到 PES 的示意图。也可以说, 每个 AU 实际上是编码数据流的显示单元, 即相当于解码的 1 幅视频图像或 1 个音频帧的取样。

ES 特有信息是由 PES 包头识别标志、PES 包头长信息、信息区和用于调整信息区可变包长的填充字节 4 部分组成的 PES 包控制信息。其中, PES 包头识别标志由 12 个部分组成: PES 加扰控制信息、PES 优先级别指示、数据适配定位指示符、有否版权指示、原版或拷贝指示、有否显示时间标记 (PTS-Presentation Time Stamp) / 解码时间标记 (DTS-Decode Time Stamp) 标志、PES 包头有否基本流时钟基准 (ESCR-Elementary Stream Clock Reference) 信息标志、PES 包头有否基本流速率信息标志、有否数字存储媒体 (DSM) 特技方式信息标志、有否附加的拷贝信息标志、PES 包头有否循环冗余校验 (CRC-Cyclic Redundancy Check) 信息标志、有否 PES 扩展标志。有扩展标志, 表明还存在其它信息。如, 在有传输误码时, 通过数据包计数器, 使接收端能以准确的数据恢复数据流, 或借助计数器状态, 识别出传输时是否有数据包丢失。

其中, 有否 PTS/DTS 标志, 是解决视音频同步显示、防止解码器输入缓存器上溢或下溢的关键所在。因为, PTS 表明显示单元出现在系统目标解码器 (STD-System Target Decoder) 的时间, DTS 表明将存取单元全部字节从 STD 的 ES 解码缓存器移走的时刻。视频编码图像帧次序为 I1P4B2B3P7B5B6I10B8B9 的 ES, 加入 PTS/DTS 后, 打包成一个个视频 PES 包。每个 PES 包都有一个包头, 用于定义 PES 内的数据内容, 提供定时资料。每个 I、P、B 帧的包头都有一个 PTS 和 DTS, 但 PTS 与 DTS 对 B 帧都是一样的, 无须标出 B 帧的 DTS。对 I 帧和 P 帧, 显示前一定要存储于视频解码器的 重新排序缓存器 中, 经过延迟 (重新排序) 后再显示, 一定要分别标明 PTS 和 DTS。例如, 解码器输入的图像帧次序为 I1P4B2B3P7B5B6I10B8B9, 依解码器输出的帧次序, 应该 P4 比 B2、B3 在先, 但显示时 P4 一定要比 B2、B3 在后, 即 P4 要在提前插入数据流中的时间标志指引下, 经过缓存器重新排序, 以重建编码前视频帧次序 I1B2B3P4B5B6P7B8B9I10。显然, PTS/DTS 标志表明对确定事件或确定信息解码的专用时标的存在, 依靠专用时标解码器, 可知道该确定事件或确定信息开始解码或显示的時刻。例如, PTS/DTS 标志可用于确定编码、多路复用、解码、重建的时间。





### 1.3 TS

Transport stream 将具有共同时间基准或者具有独立时间基准的一个或者多个 PES 组成的单一数据流。

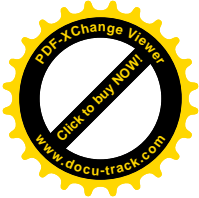
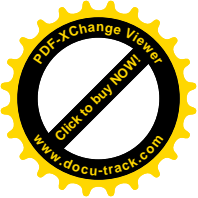


图二：TS 包结构

TS 结构各字段的具体含义见附表二。

由图二可见，TS包由包头、自适应区和包数据3部分组成。每个包长度为固定的188 B，包头长度占4 B，自适应区和包数据长度占184 B。184 B为有用信息空间，用于传送已编码的视音频数据流。当节目时钟基准（PCR-Program Clock Reference）存在时，包头还包括可变长度的自适应区，包头的长度就会大于4 B。考虑到与通信的关系，整个传输包固定长度应相当于4个ATM包。考虑到加密是按照8 B顺序加扰的，代表有用信息的自适应区和包数据的长度应该是8 B的整数倍，即自适应区和包数据为 $23 \times 8 \text{ B} = 184 \text{ B}$ 。

TS包的包头由如图所示的同步字节、传输误码指示符、有效载荷单元起始指示符、传输优先、包识别（PID-Packet Identification）、传输加扰控制、自适应区控制和连续计数器8个部分组成。其中，可用同步字节位串的自动相关特性，检测数据流中的包限制，建立包同步；传输误码指示符，是指有不能消除误码时，采用误码校正解码器可表示1bit 的误码，但无法校正；有效载荷单元起始指示符，表示该数据包是否存在确定的起始信息；传输优先，是给TS包分配优先权；PID值是由用户确定的，解码器根据PID将TS上从不同ES来的TS包区别出来，以重建原来的ES；传输加扰控制，可指示数据包内容是否加扰，但包头和自适应区永远不加扰；自适应区控制，用2 bit表示有否自适应区，即（01）表示有有用信息无自适应



应区，（10）表示无有用信息有自适应区，（11）表示有有用信息有自适应区，（00）无定义；连续计数器可对PID包传送顺序计数，据计数器读数，接收端可判断是否有包丢失及包传送顺序错误。显然，包头对TS包具有同步、识别、检错及加密功能。

TS包自适应区由自适应区长、各种标志指示符、与插入标志有关的信息和填充数据4部分组成。其中标志部分由间断指示符、随机存取指示符、ES优化指示符、PCR标志、接点标志、传输专用数据标志、原始PCR标志、自适应区扩展标志8部分组成。重要的是标志部分的PCR字段，可给编解码器的27MHz时钟提供同步资料，进行同步。其过程是，通过PLL，用解码时本地用PCR相位与输入的瞬时PCR相位锁相比较，确定解码过程是否同步，若不同步，则用这个瞬时PCR调整时钟频率。因为，数字图像采用了复杂而不同的压缩编码算法，造成每幅图像的数据各不相同，使直接从压缩编码图像数据的开始部分获取时钟信息成为不可能。为此，选择了某些（而非全部）TS包的自适应区来传送定时信息。于是，被选中的TS包的自适应区，可用于测定包信息的控制bit和重要的控制信息。自适应区无须伴随每个包都发送，发送多少主要由选中的TS包的传输专用时标参数决定。标志中的随机存取指示符和接点标志，在节目变动时，为随机进入I帧压缩的数据流提供随机进入点，也为插入当地节目提供方便。自适应区中的填充数据是由于PES包长不可能正好转为TS包的整数倍，最后的TS包保留一小部分有用容量，通过填充字节加以填补，这样可以防止缓存器下溢，保持总码率恒定不变。

## 2 TS 流的解码过程（只考虑本地文件，解码只涉及到系统层打

包的解码，不涉及到编码的解码）

TS 流的基本解码过程：PAT 节目关联表-》PMT 节目映射表-》PES 音频、视频包。

首先，从 PAT 获取 TS 流中所有节目映射表。

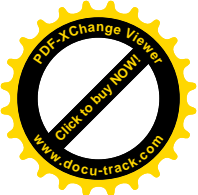
然后，从节目映射表中获取每个节目（本设计为本地文件，只含有一个 PMT）数据（视频和音频）的 PID。

最后，根据传输过来的数据 PID 对视频数据和音频数据进行系统层复用解码。系统层复用解码：循环：TS-》PES-》ES。

### 2.1 PAT 解码：

PAT 表携带以下信息：

- （1）TS 流 ID --- transport\_stream\_id，该 ID 标志唯一的流 ID 。
- （2）节目频道号-- program\_number，该号码标志 TS 流中的一个频道，该频道可以包含



很多的节目(即可以包含多个 Video PID 和 Audio PID)

(3) PMT 的 PID--- program\_map\_PID, 表示本频道使用的哪个 PID 做为 PMT 的, 因为 PID 可以有很多的频道, 因此 DVB 规定 PMT 的 PID 可以由用户自己定义.

## 2.2 PMT 解码

PMT 表中包含的数据如下:

- (1) 当前频道中包含的所有Video 数据的PID
- (2) 当前频道中包含的所有Audio 数据的PID
- (3) 和当前频道关联在一起的其他数据的PID(如数字广播, 数据通讯等使用的PID)

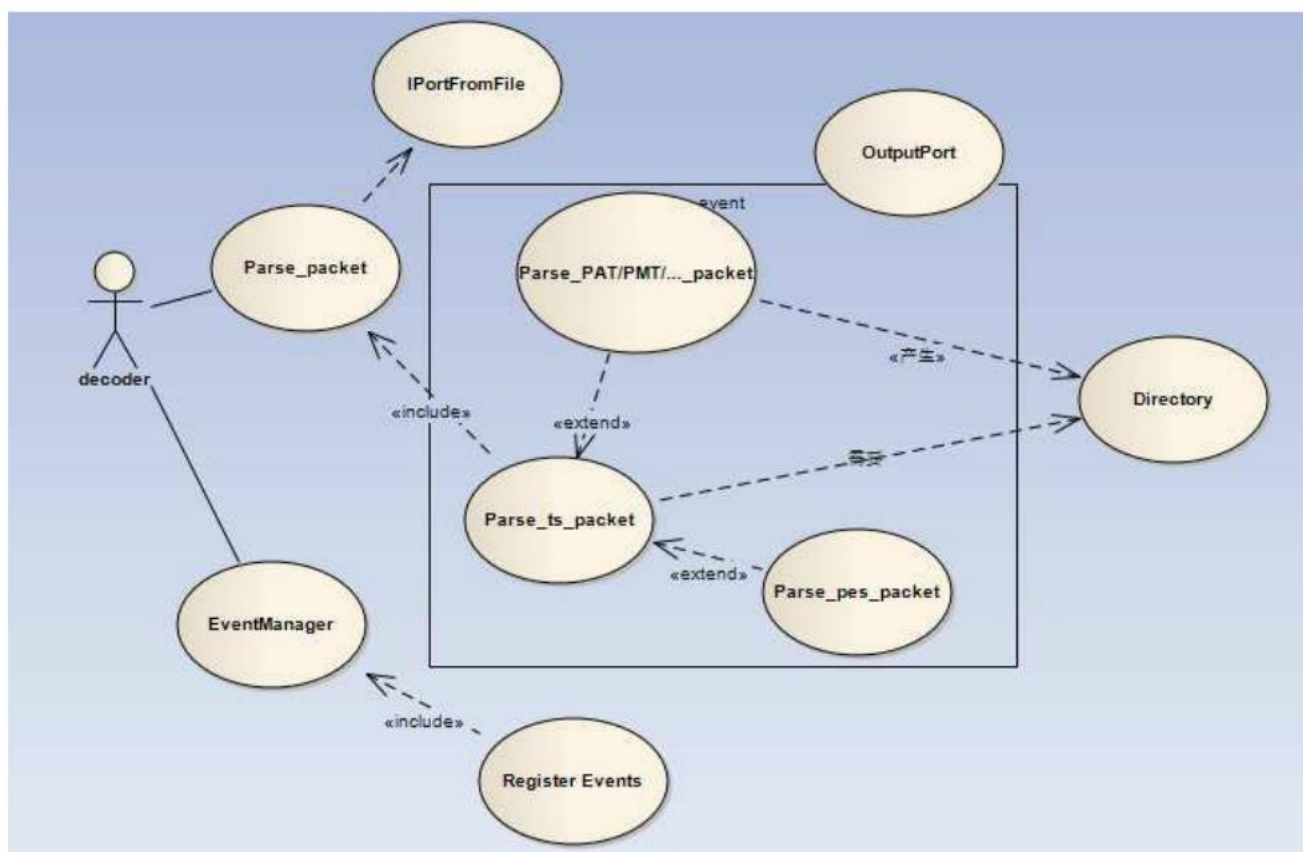
## 2.3 音视频解码

音视频解码的数据如下:

- (1) 根据音频PID解码音频数据到缓存区
- (2) 根据视频PID解码音频数据到缓存区
- (3) 和当前频道关联在一起的其他数据放到数据区中

## 3 TS 流解码

### 3.1 总用例图



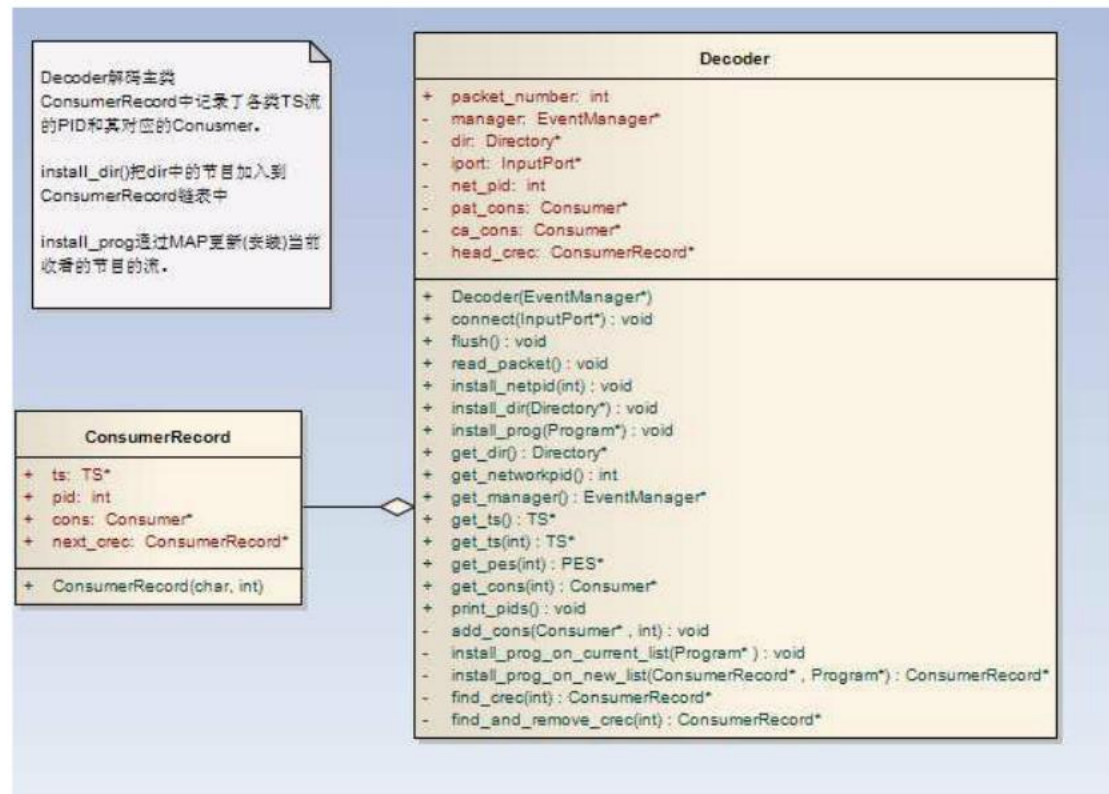
Decoder 首先调用事件管理器注册各类事件（事件类型见 3.2.2），然后调用 read\_packet 函数读取指定文件中的 TS 包进行分析，根据 PID 值不同分别对 TS 进行不同的处理。

Parse\_PAT/PMT/...\_packet 把 PAT(注：PAT 的 PID 为 0)和 PMT 的携带的相关节目流 PID 的信息(包括视频，音频，私有数据等)存入 Directory 中，之后 Parse\_ts\_packet 通过 Directory 中提供的 PID 信号分别对不同的 TS 分别处理。



## 3.2 各解码类的 UML 图

### 3.2.1 解码主类



The Decoder object manages such global information as the Network Table pid value, the current packet number and the number of programs and streams per program (as represented in the Directory structure).

主要函数介绍:

read\_packet () : Called to read a new transport packet.

install\_dir (Directory\*): install Directory, Directory 是一个节目单 (比如湖南卫视, 星空卫视, 东方卫视, CCTV)。

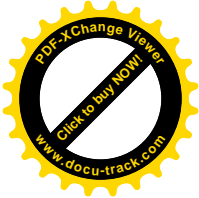
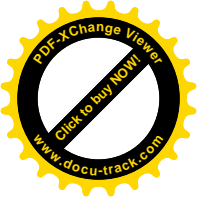
install\_prog (Program\*): install program(如湖南卫视) to Directory。

install\_netpid () : 表示网络信号的来源, 当这个更新时, 需要重新安装目录。

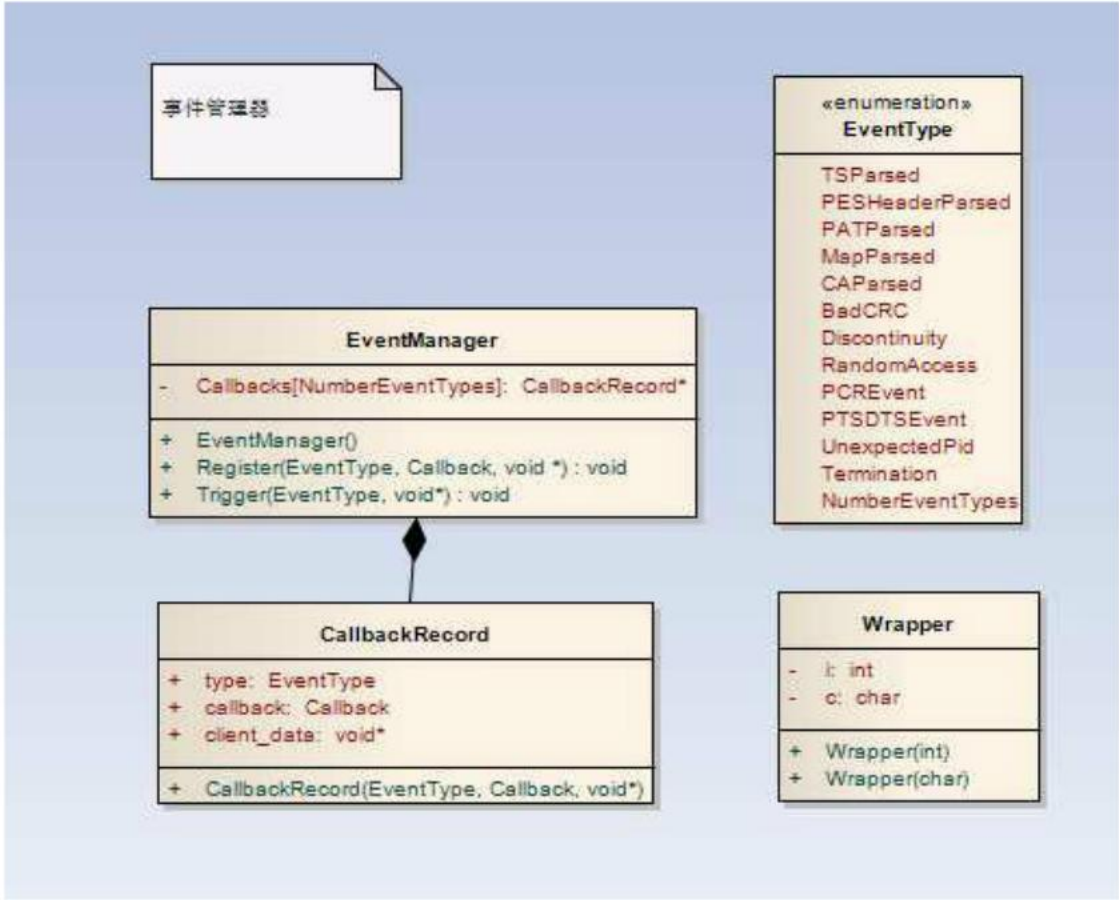
install\_prog\_on\_current\_list 安装节目到当前列表。

install\_prog\_on\_new\_list 安装节目到新的列表。





### 3.2.2 事件管理类

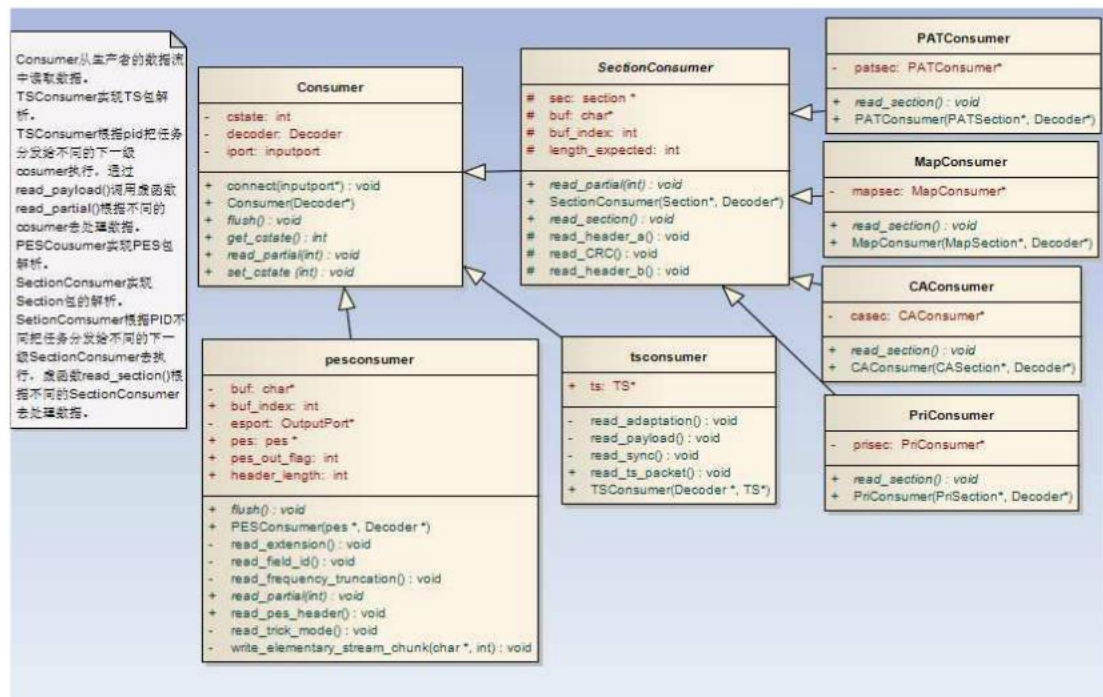


typedef void (\*Callback)(EventType, void\*, void\*): 回调函数模板。

void Register (EventType t, Callback f, void\* d): 注册事件。

void Trigger (EventType, void\*): 触发事件。

### 3.2.3 各类包解析的关系图



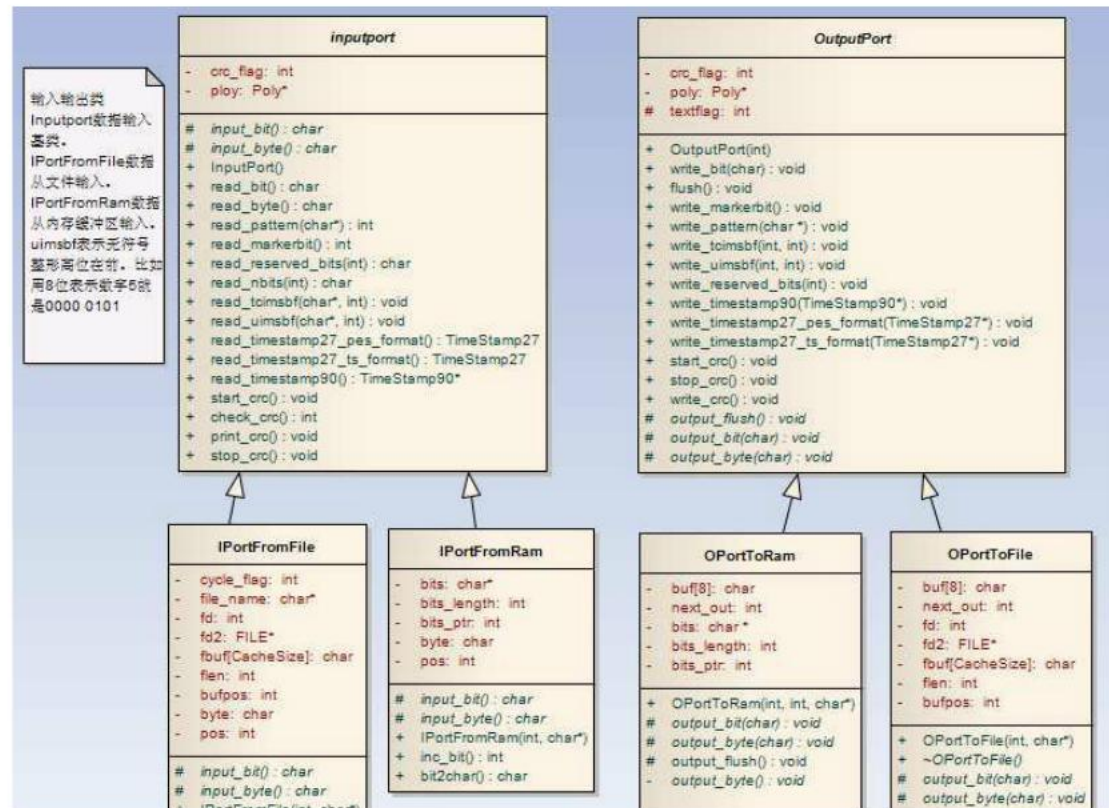
Consumer : connect (InputPort\*):连接到输入接口。set\_cstate、get\_cstate 表示设置和获取状态。

Sectionconsumer: read\_partial 表示读取 TS 的 payload。read\_header\_a、read\_header\_b 表示读取负载的头部分。

read\_section 根据读取的负载，更新节目。各个子类分别实现。

PES 同上。

### 3.2.4 输入输出类



按要求从缓冲区中读入规定的位，或者字节数。

比如函数 `read_bit()` 表示从缓冲区中读入一个 bit 的数据（注意缓冲区中以 byte 对齐），读入时生成 CRC。所有从缓冲区读出数据的方式（bit\byte\nbits）基函数都是这个。函数 `read_byte` 从缓冲区读取一个字节。`read_pattern` 表示读取指定字符串。`read_reserved_bits` 读取填充数据。

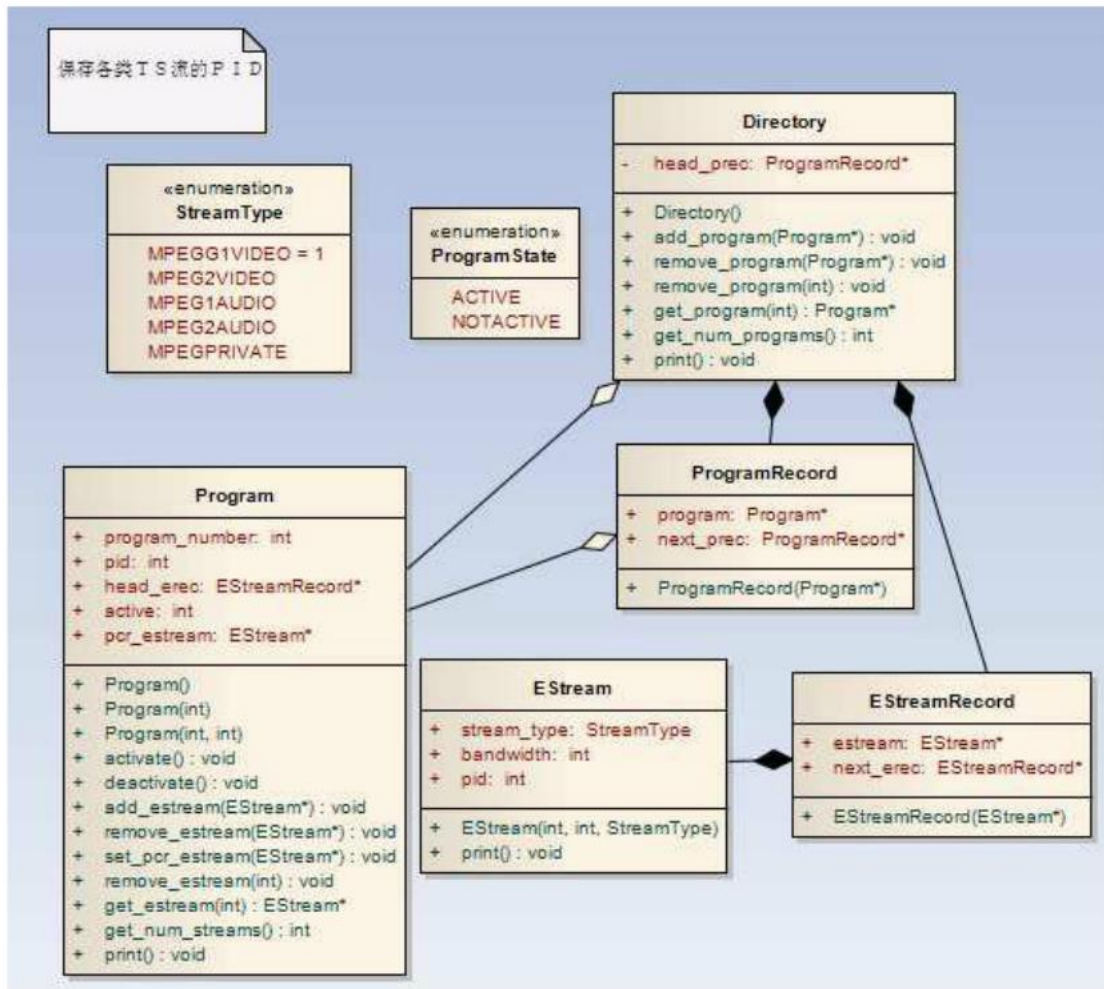
`read_uimsbf(int nbits, char* message)` 读取 n 位转化为无符号整形，高位在前。

`read_tcimsbf(int nbits, char* message)` 读取 n 位转化为有符号整形，高位在前。

`check_crc` 表示 CRC 校验。



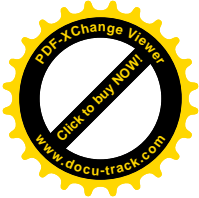
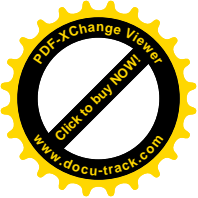
### 3.2.5 解析 PAT、PMT、私有数据等



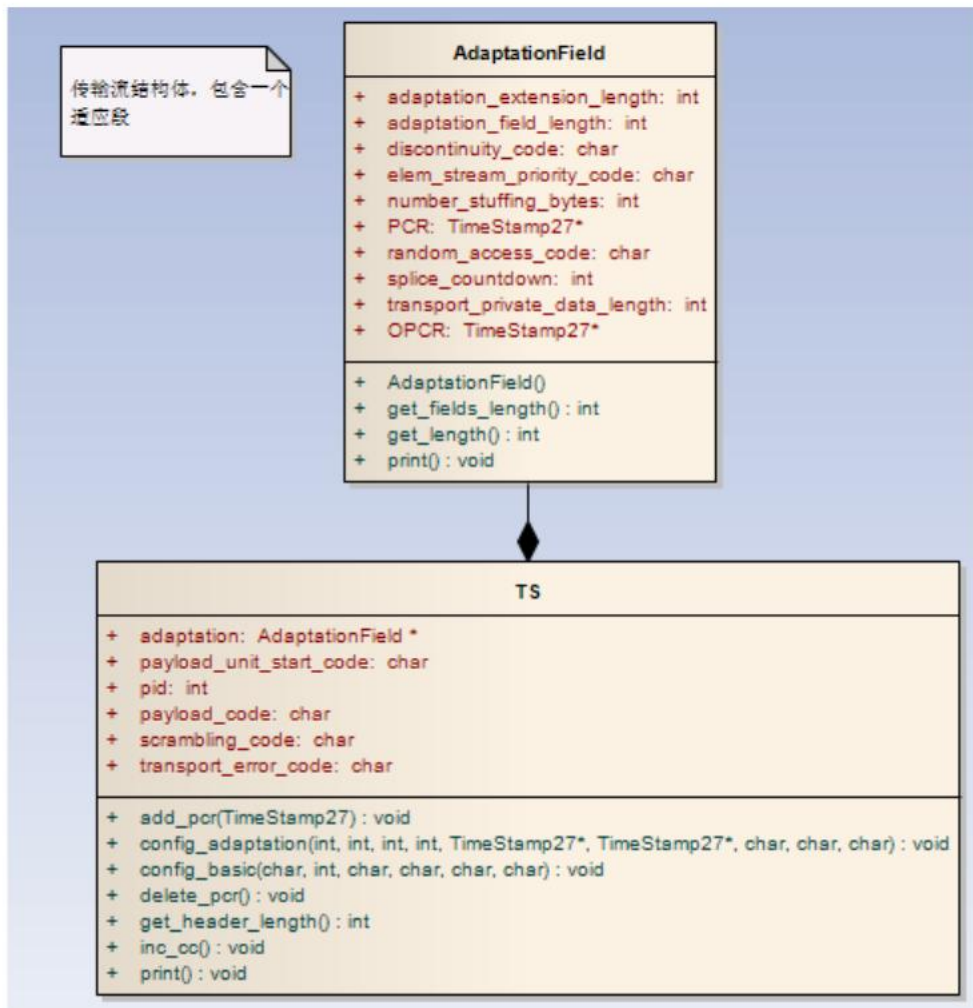
解析 PMT 中携带的节目信息到 program，再把 program 插入到 directory。类似于机顶盒的搜台系统。

Directory: 函数 add\_program、remove\_program 表示添加和删除节目。get\_program 表示获取节目，get\_num\_programs 表示获取节目数，

Program: 函数 active、deactivate 表示添加激活和停止节目，add\_estream、remove\_estream 表示给节目添加和删除 ES 流，set\_pcr\_estream 表示设置该流含有 PCR，get\_estream 表示获取 ES 流，get\_num\_streams 表示获取节目中含流的数目。



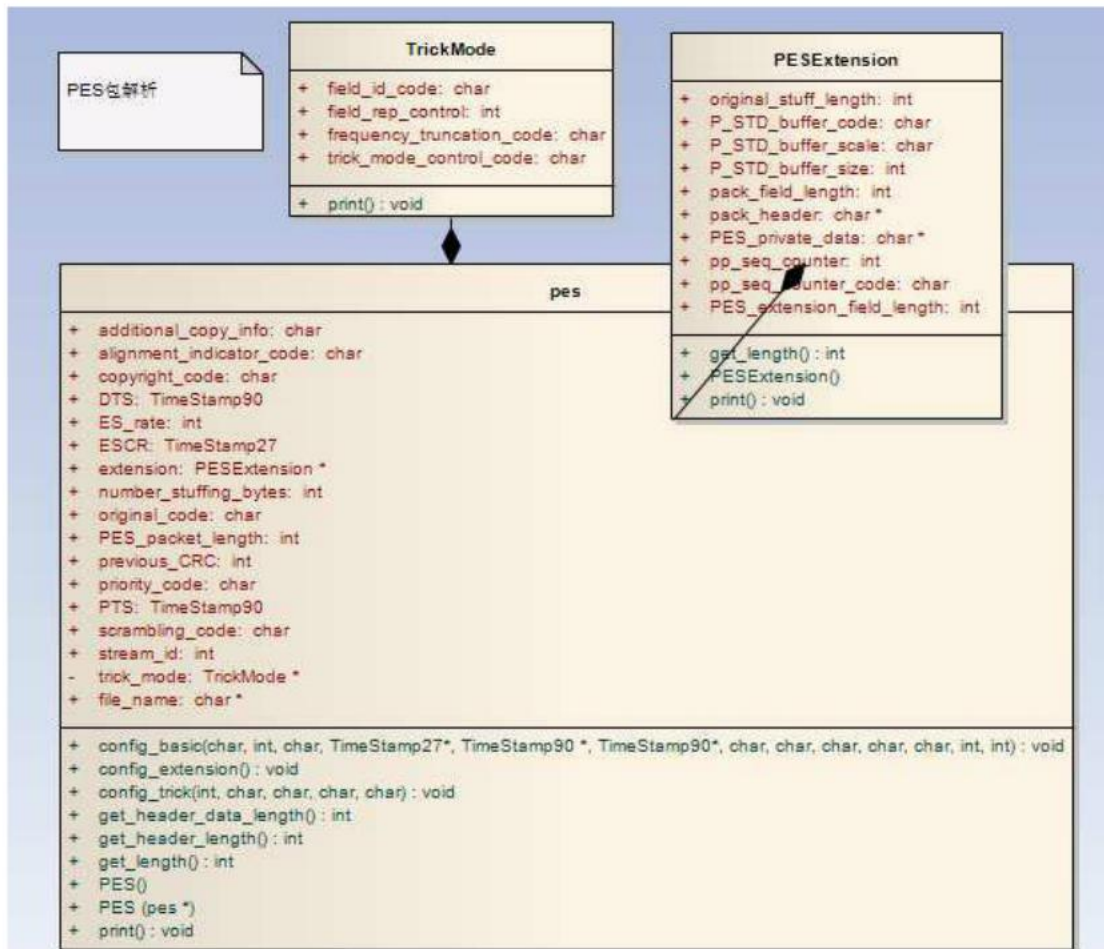
### 3.2.6 解析 TS 包



TS 包结构，函数 `get_header_length` 表示 TS 包头的长度，`config_basic` 表示 TS 包基本标志参数设置，`config_adaptation` 表示适应字段设置，`inc_cc` 表示包数量增加，`add_pcr`，`delete_pcr` 添加和删除 PCR。

`get_fields_length` 表示调整字段长，不包括标志位，`get_length` 表示 total length of the adaptation field。

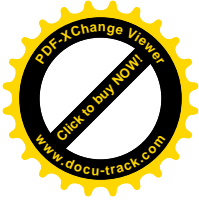
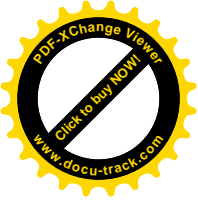
### 3.2.7 解析 PES 包



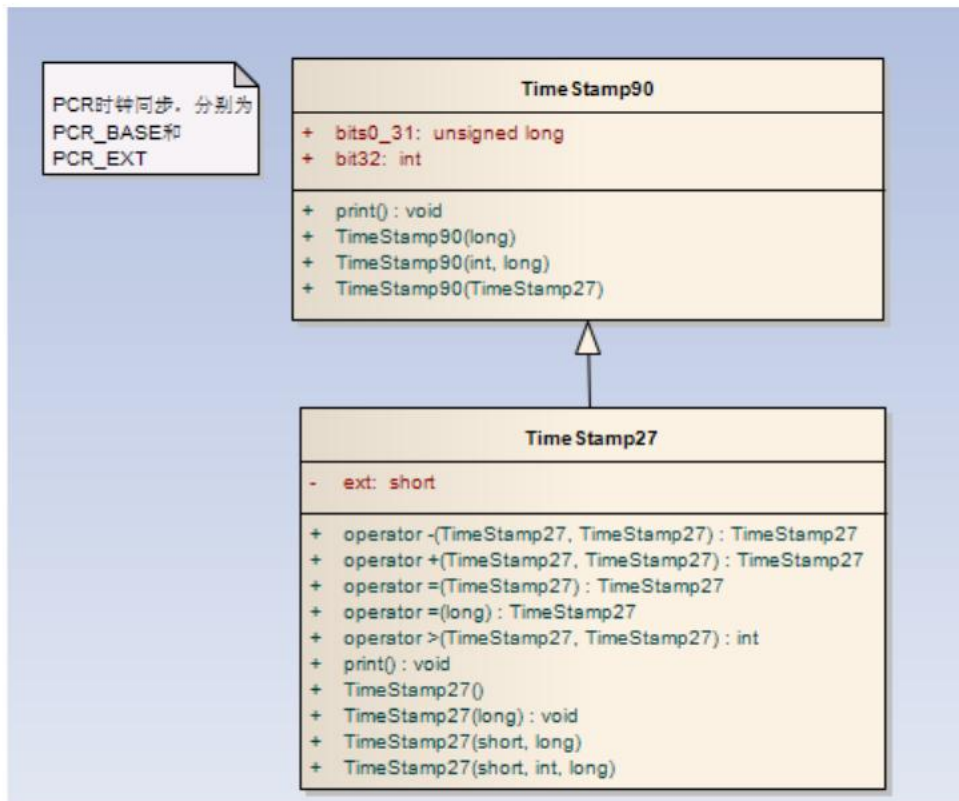
对 PES 包进行解析。解析出头部数据，解析后的数据分别存贮在”stream68”和”stream69”2 个文件中。

函数 get\_length 表示包总长度。get\_header\_data\_length 表示 PES 头长度，不包括填充字段。get\_header\_length 表示 PES 头长度，包括填充字段。config\_basic 表示 PES 参数设置。





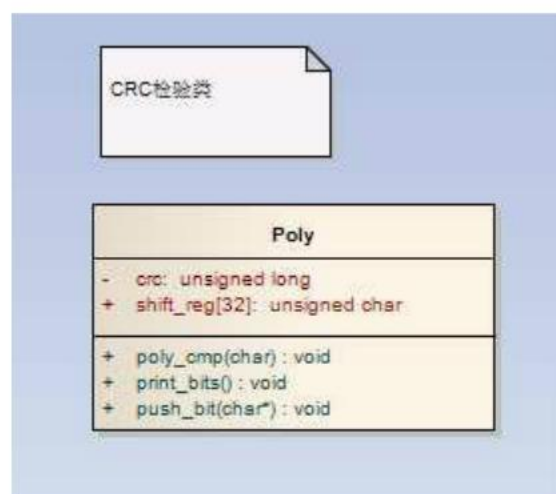
### 3.2.8 PCR 时钟类



系统时钟的相关类，对系统时钟进行比较。其中对 27MHZ 的时钟算术和逻辑运算符进行了重载。

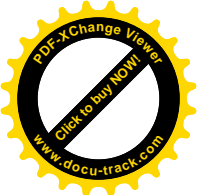
注：PCR 的校正一直是学术界研究的热点。

### 3.2.9 CRC 校验类



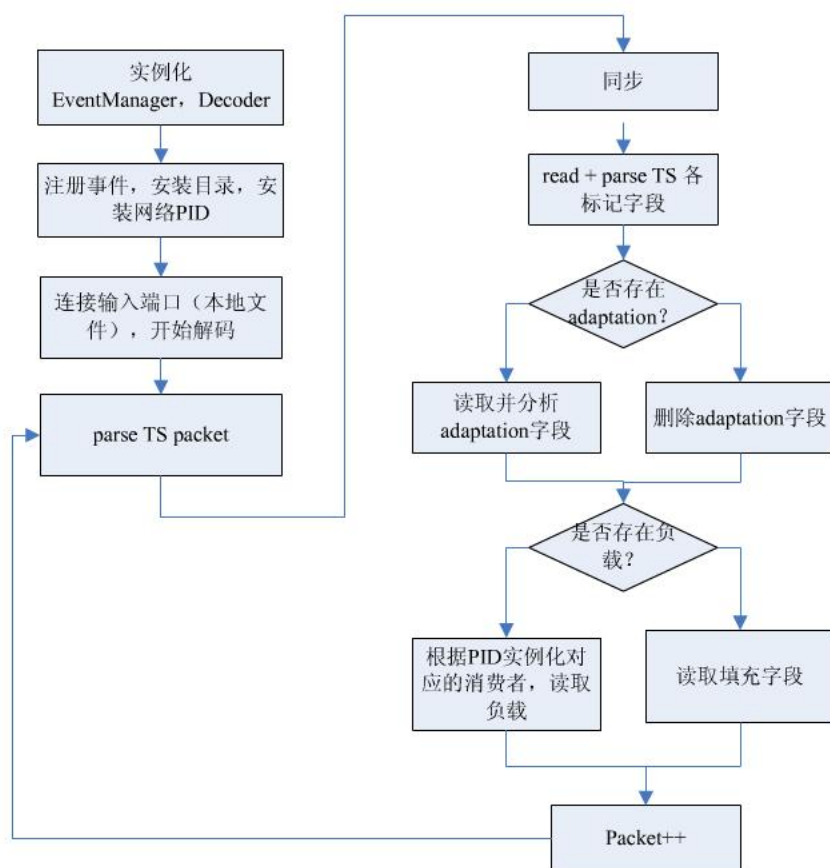
每位进行异或运算后存储在一个 4BYTE 的数组中。

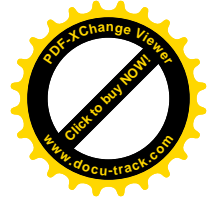
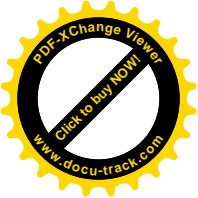
函数 `poly_cmp` 作 CRC 验证。函数 `push_bit` 作 CRC 的生成。



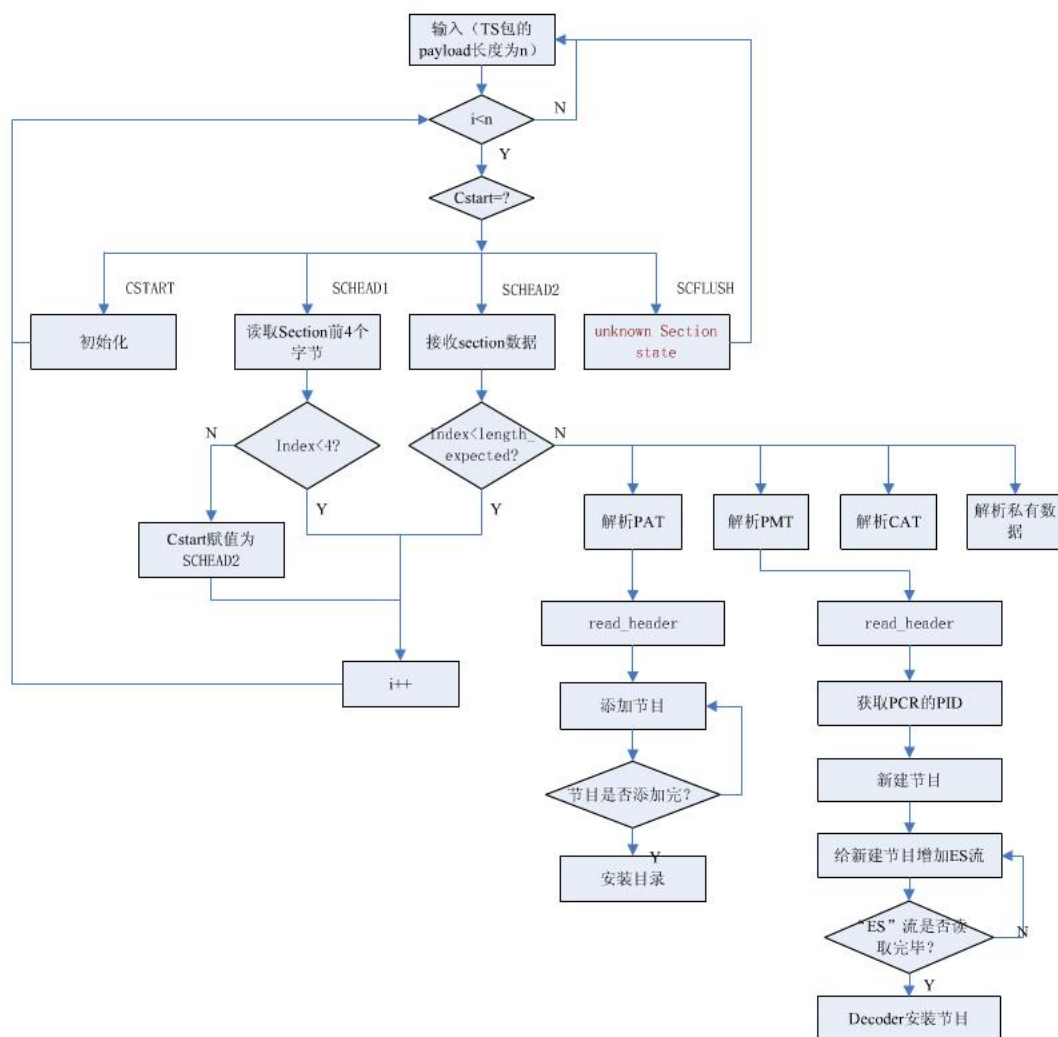
## 3.3 解码流程图

### 3.3.1 TS 包解析流程图

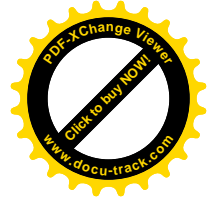
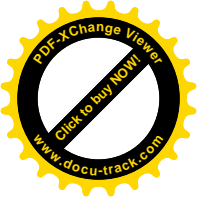




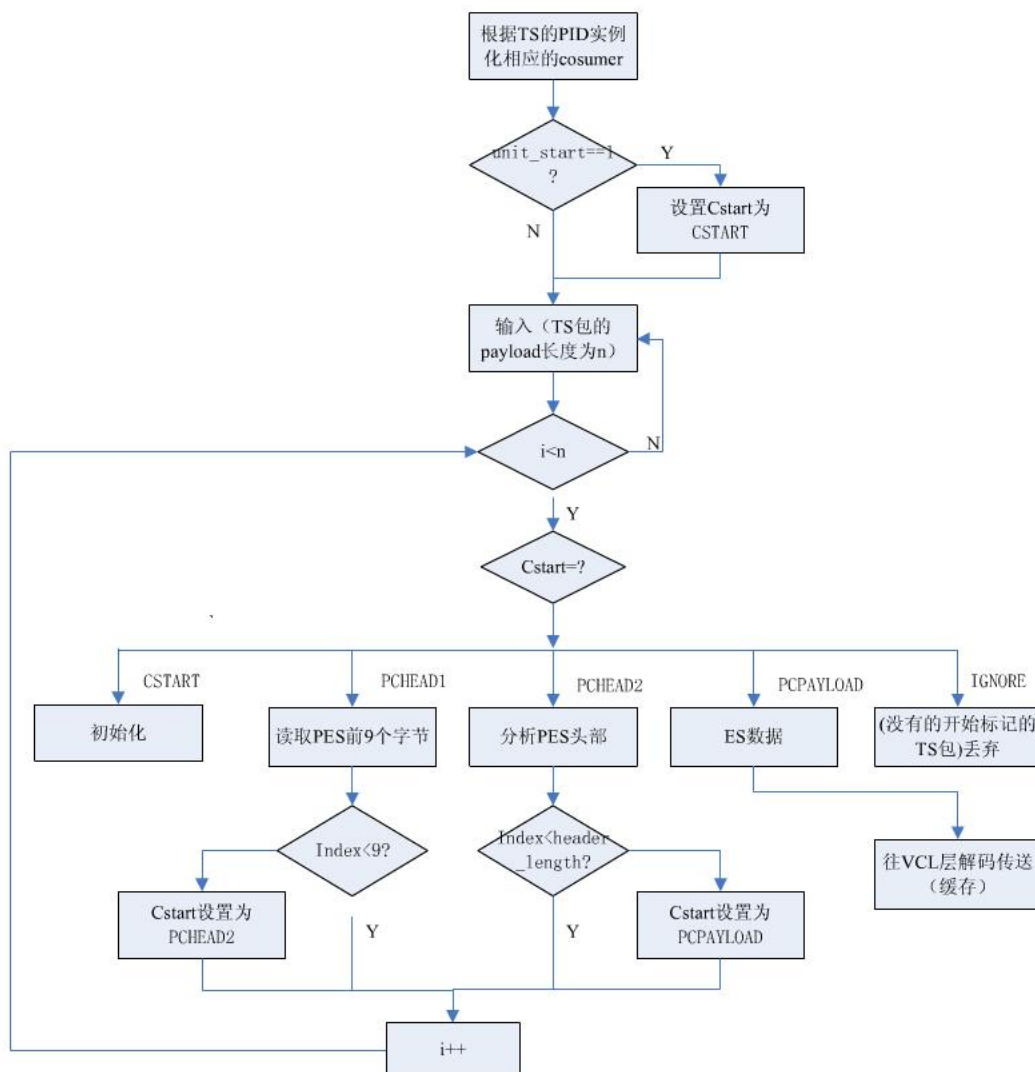
### 3.3.2 Section 解码流程图

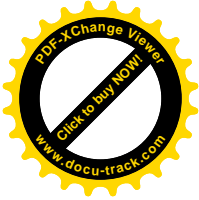
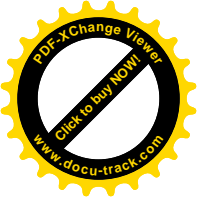






### 3.3.3 PES 解码流程图

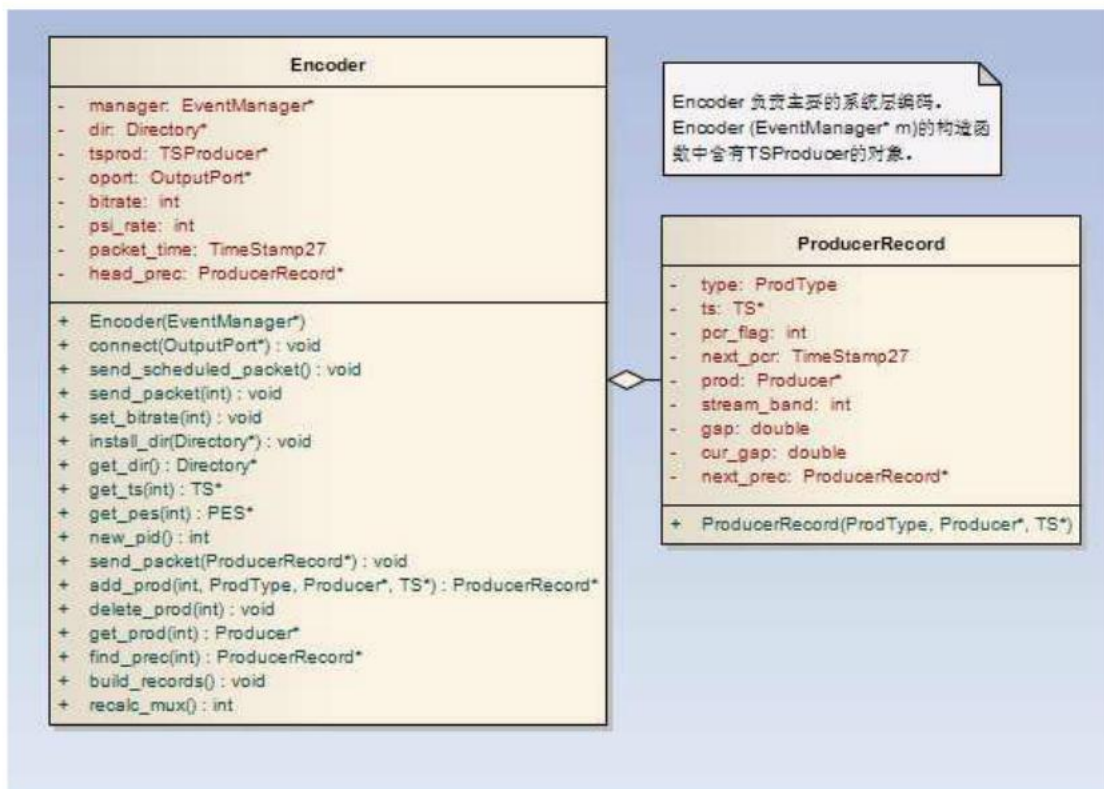




## 4 TS 流合成（编码）

### 4.1 各编码类的 UML 图

#### 4.1.1 编码主类



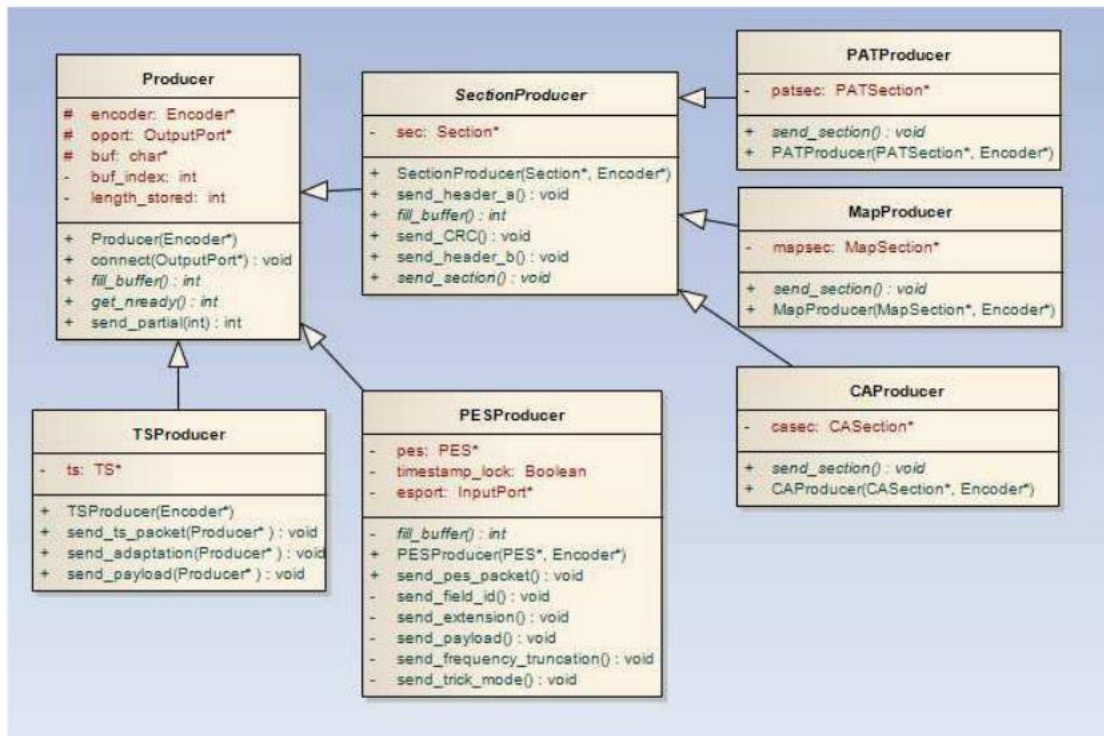
函数 install\_dir 首先安装 PAT

函数 send\_packet 表示按 PID 发送 TS 包

函数 send\_scheduled\_packet 表示按一定的规律轮流发送 TS 包。

函数 recalc\_mux 表示节目复用

## 4.1.2 合成各类之间的关系



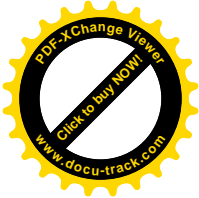
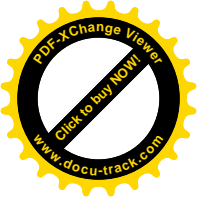
函数 `get_nready` 获取已经填充的字节长度.

函数 `fill_buffer` 填充 buffer.

函数 `send_payload` 发送负载.

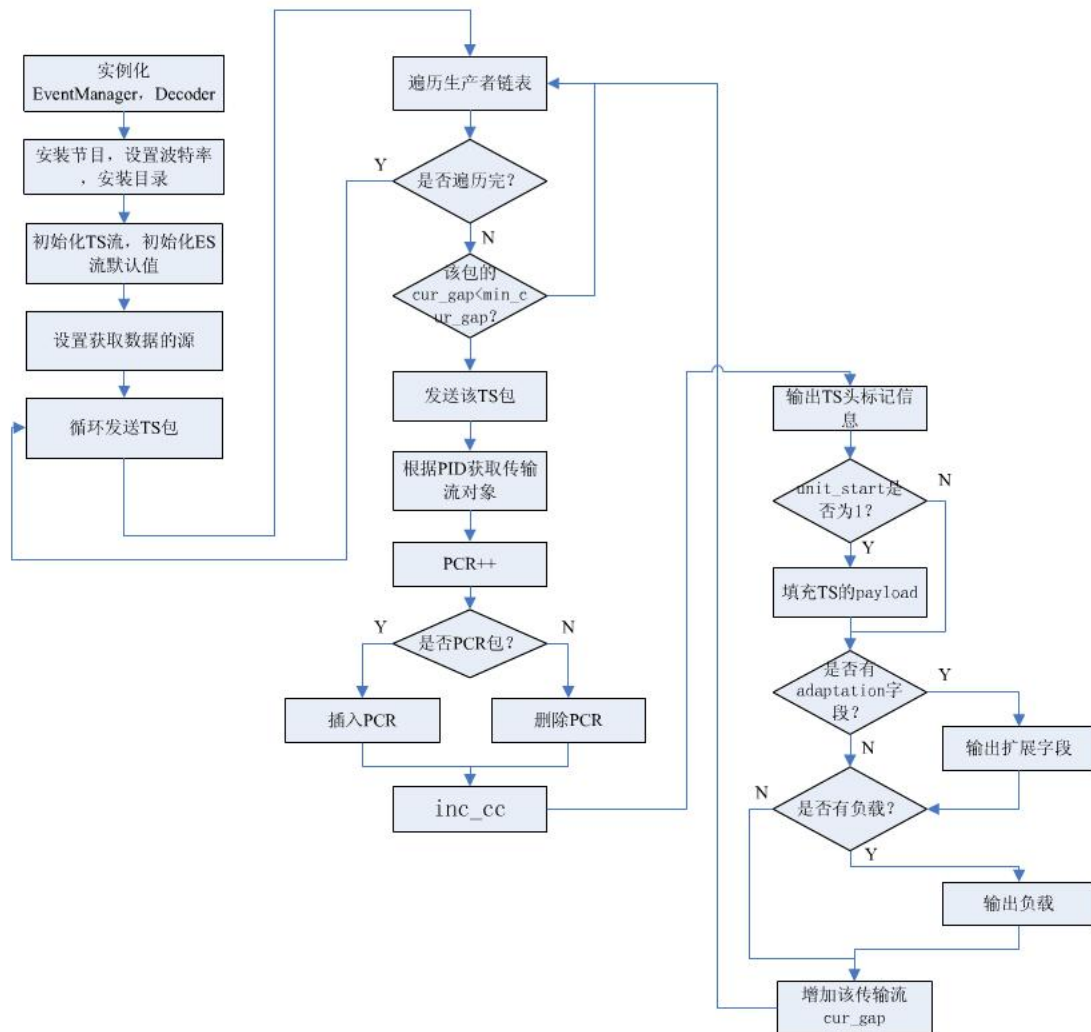
函数 `send_section` 发送 section packet.

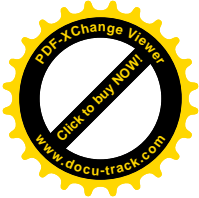
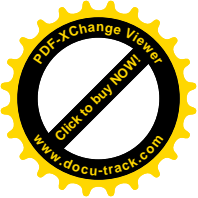




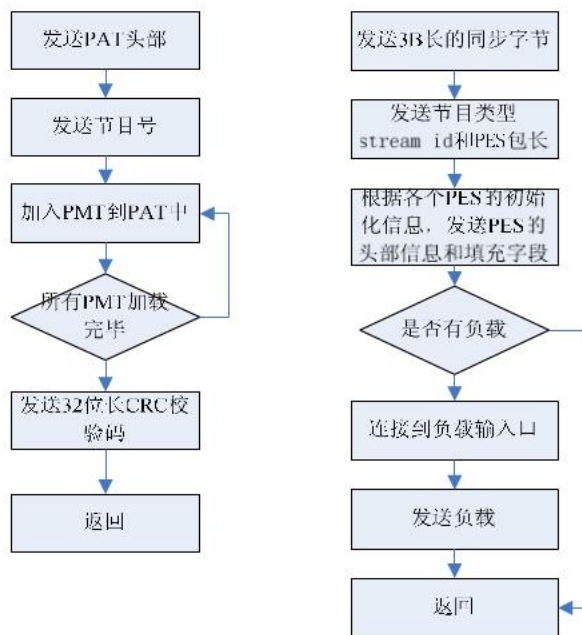
## 4.2 编码主要流程

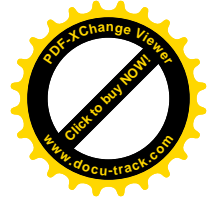
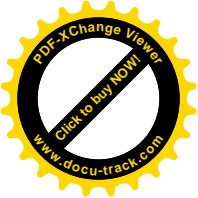
### 4.2.1 TS 编码流程图





## 4.2.2 负载（pes、section）打包过程

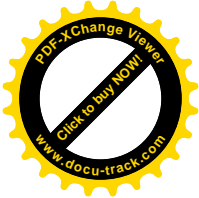
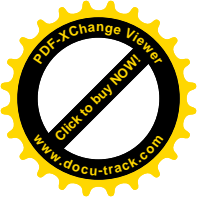




附表一：

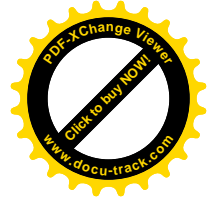
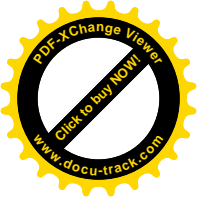
● PES 分组字段

packet_start_code_prefix(24)	开始码字为 0X00 00 01
stream_id(8)	原始流的类型和数目,取值从 1011 1100 到 1111 1111 之间。各值含义具体见 13818-1。
PES_packet_length(16)	表示从此字节之后 PES 包长(单位字节)。0 表示 PES 包长不限制,且只能用于视频 PES。
10	填充字节。
PES_scrambling_control(2)	PES 有效负载的加密模式。00 表示不加密,其余表示用户自定义。
PES_priority(1)	PES 数据包的优先级。类似于 TS 的此字段。
data_alignment_indicator(1)	为 1 时,表明此分组头部之后紧跟着 数据流描述子中定义的访问单元类型。
copyright(1)	版权,1 表示有版权,具体见版权描述子 13818-1 1-2-6-24。0 表示没有。
original_or_copy(1)	1 表示原始数据,0 表示备份
PTS_DTS_flag(2)	10 表示含有 PTS 字段,11 表示含有 PTS 和 DTS 字段,00 表示不含有 PTS 和 DTS,01 无定义。
ESCR_flag(1)	1 表示 ESCR 在 PES 首部出现,0 表示不出现
ES_rate_flag(1)	1 表示 PES 分组含有 ES_rate 字段。0 表示不含有。
DSM_trick_mode_flag(1)	1 表示有 8 位的 trick_mode_flag 字段,0 表示不出现此字段。只对 DSM 有效。在广播中不用。
additional_copy_info_flag(1)	1 表示有 copy_info_flag 字段,0 表示不出现此字段。
PES_CRC_flag(1)	1 表示 PES 分组中有 CRC 字段,0 表示不出现此字段。
PES_extention_flag(1)	1 表示扩展字段在 PES 包头存在,0 表示扩展字段不存在
PES_header_data_length(8)	表示可选字段和填充字段所占的字节数。
0010	0010 填充字段,表示只含有 PTS,不含有 DTS 当含有 DTS 时,这个填充字段为 0011  PTS 字段。
PTS[32...30](3)	
marker_bit(1)	
PTS[29...15](15)	
marker_bit(1)	
PTS[14...0](15)	
marker_bit(1)	0001 填充字段,表示接下来为 DTS。
0001	
DTS[32...30](3)	
marker_bit(1)	



DTS[29...15](15)	DTS 数据解码时间
marker_bit(1)	
DTS[14...0](15)	
marker_bit(1)	
reserved(2)	填充字段  基本系统参考时间，和扩展系统参考时间。
ESCR_base[32...30](3)	
marker_bit(1)	
ESCR_base[29...15](15)	
marker_bit(1)	
ESCR_base[14...0](15)	
marker_bit(1)	
ESCR_extention(9)	
marker_bit(1)	
marker_bit(1)	
ES_rate(22)	
marker_bit(1)	
trick_mode_control(3)	特技模式，比如快进，快退。具体见 13818-1
field_id(2)	表示在特技模式中，哪些场将被显示。00 表示仅显示顶场，01 表示仅显示底场，10 表示显示全帧。11 保留。 1 表示编码间隙中可能丢失宏块，0 表示不会丢失宏块。丢失宏块用前面已解码的图像宏块代替。 编码的受限系数集，表示 DCT 后系统的非零限制，具体见 13818-1
intra_slice_refresh(1)	
frequency_trunction(2)	
frequency_trunction(2)	
field_rep_cntrl(5)	隔行扫描中首场和底场显示顺序。
reserved(3)	填充
marker_bit(1)	填充
additional_copy_info(7)	包含和版权有关的私用数据
previous_PES_packet_CRC(16)	CRC 校验 PES 包数据（不包括包头）
PES_private_data_flag(1)	1 表示 PES 包首部中含有私用数据，0 表示没有。
pack_head_field_flag(1)	1 表示 PES 首部含有一个 ISO/IEC 11172 的组首部。0 表示 PES 首部没有含有组首部。
program_packet_sequence_counter_flag(1)	1 表示 PES 分组含 program packet sequence counter 和 <b>original_stuff_length</b> 字段，0 表示不含有。
P-STD_buffer_flag(1)	
reserved(3)	填充
PES_extention_flag_2(1)	1 表示出现 PES_extention_flied 及相关字段。0 表示不出现。
PES_private_data(128)	私用数据





pack_field_length(8)	指示 pack_header_field() 字节长度。
marker_bit(1)	
program_packet_sequence_counter(7)	计数器，到 0 后重新计数，计算 PES 分组的数量。
marker_bit(1)	
MPEG1_MPEG2_identifier(1)	1 表示 PES 分组带有 ISO/IEC 11172-1 系统信息，0 代表带有 PS 流信息。
original_stuff_length(6)	
01	
P-STD_buffer_scale(1)	联合使用控制 BSn 缓冲区的大小，具体定义见 13818-1。仅用于 PS 流中。
P-STD_buffer_size(13)	
marker_bit(1)	
PES_extention_field_length(7)	规定此字段之后的扩展(填充)字段长度(单位字节)
reserved(8)	保留位 8 位
stuffing_byte(8)	填充字段。1111 1111
PES_packet_data_byte(8)	PES 包原始流数据长度，等于 PES_packet_length 减去从 PES_packet_length 到此字段之前的长度。
padding_byte(8)	填充字段。1111 1111

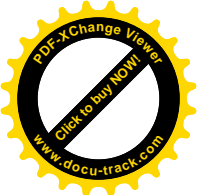
附表二：

● 传送流结构

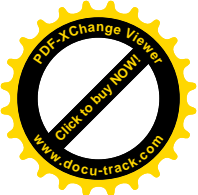
sync_byte (8)	01000111
transport_error_indicator(1)	传送层外被置 1，表示至少一个不可纠正的错误位，改正前不被置 0
payload_unit_start_indicator(1)	该字段置 1 表示携带的是 PSI 或 PES 第一个包。该字段置 0 表示携带的不是 PSI 或 PES 第一个包。
transport_priority(1)	置 1 表示比其他同 PID 组有更高优先级
PID(13)	0x0000-0x000F 保留，0x1FFF 空分组
transport_scrambling_control(2)	00 表示未加密，其他“自定义”，空分组为 00
adaption_field_control(2)	00 保留，01 无调整字段，10 无负载，11 调整字段后为有效负载
continuity_counter(4)	依据前一成员，只在 10 和 11 时生效。计数同 PID 的分组数
data_byte	PES 分组、PSI 部分，以及其他结构中私有数据的连续字节；若为空分组，可以为任意值。 即有效负载的数据。

● 传送流调整字段

adaptation_field_length(8)	调整字段长度。(不包括他自身的 1byte，即他之后的长度)
discontinuity_indicator(1)	间断指示符。
Random_access_indicator(1)	随机存储指示符，为 1 时，表示 TS 为一个 PES



	分组原始流的开始。当 PID 和 PSI 中的 PCR_PID（该 PID 是 时钟 PID）相等，且 Random_access_indicator 为 1 时，调整字段中必须含有 PCR。
elementary_stream_priority_indicator(1)	基本码流优先级指示符。当码流中含有有效负载时才有效，1 表示比其他传送流优先级高。0 表示优先级等同。
PCR_flag(1)	PCR 标记，1 表示包含 PCR 字段，0 表示不包含。
OPCR_flag(1)	原始 OPCR 标记，1 表示包含 OPCR 字段，0 表示不包含。
splicing_point_flag(1)	节点标记，为 1 时表示 splice_countdown 出现在调整字段中，否则不出现。
transport_private_data_flag(1)	表示调整字段中是否含有私有字段，0 表示没有，1 表示有。
adaptation_field_extention_flag(1)	为 1 时，表示存在调整字段扩展，为 0 是，表示不存在调整字段扩展。
program_clock_reference_base(33)	PCR 时间参考基准
reserved(6)	6 位保留字段
program_clock_reference_extention(9)	PCR 扩展
original_program_clock_reference_base(33)	原始 PCR 时间参考基准
reserved(6)	保留 6 位
original_program_clock_reference_extention(33)	原始 PCR 时间扩展
splice_countdown(8)	可正、可负。正值表示此 PES 分组还剩下多少 TS 分组数目，0 表示此 TS 分组已经是此 PES 分组的最后一个字节。为负数的时候表示，TS 是拼接点之后的第 N 个分组。
transport_private_data_length(8)	表示其后面私有数据长度（字节为单位），私有数据不能延伸到调整字段之外。
private_data_byte	私有数据，以字节为单位
adaptation_field_extention_length(8)	表示其后面扩展字段数据长度（字节为单位）
ltw_flag(1)	合法时间窗口标记，为 1 时，表示 ltw_offset 字段存在。
piecewise_rate_flag(1)	为 1 时，表示 piecewise_rate 字段存在
seamless_splice_flag(1)	为 1 时，缓存控制。 为 0 表示此字段无意义。
reserved(5)	保留位 5 位
ltw_valid_flag(1)	为 1 时，表示 ltw_offset 字段有效，为 0 时表示此字段无意义。
ltw_offset(15)	控制是否下溢。
reserved(2)	保留位



piecewise_rate(22)	正整数，传输分组的比特率，具有延续作用。
splice_type(4)	规定具有此分组的相同 PID 的传送比特率。
DTS_next_au[32...30](3)	
marker_bit(1)	
DTS_next_au[29...15](3)	33 位表示系统时钟 DTS
marker_bit(1)	
DTS_next_au[14...0](3)	
marker_bit(1)	
reserved	填充字段，以字节为单位。
stuffing_byte(8)	11111111 填充字节。