

<http://blog.2lic.com/user1/4482/archives/2008/45574.html>

## MPEG-2 学习笔记(一)

烧火棍 发表于 2008-1-30 14:39:00

最近有点时间，看了一部分 MPEG-2 的规范，看后想总结点东西，算是做了点作业，另外希望能和大家讨论讨论，请大家指点。

中文版很多概念翻译得很模糊，不易理解，但总体来说还算是不错，适合像我这种入门级别的看，不过建议和英文版对照看，对一些概念能比较准确的理解。整个规范包括三部分：系统，视频编码，音频编码。对应的标准号分别为 ISO/IEC 13818-1，ISO/IEC 13818-2，ISO/IEC 13818-3，在规范中经常可以看到这几个字符串。

第一部分“系统”和我们现在的工作关系较紧密，我也主要学习了第一部分。后面两部分主要是讲解编码过程，编码部分看了实在让人犯晕，先偷一下懒吧，把第一部分搞清楚了再看去啃难啃的骨头吧。

下面进入正题了。

### 一、概念

规范中讲述的概念很多，容易让人糊涂，所以先把一些概念理清，弄清楚它们之间的关系，再看后面的就可提高很多的效率。

(1) ES- Elementary Streams (原始流)，对视频、音频信号及其他数据进行编码压缩后的数据流称为原始流。原始流包括访问单元，比如视频原始流的访问单元就是一副图像的编码数据。

(2) PES- Packetized Elementary Streams (分组的原始流)，原始流形成的分组称为 PES 分组，是用来传递原始流的一种数据结构

(3) 节目是节目元素的集合。节目元素可能是原始流，这些原始流有共同的时间基点，用来做同步显示。

(4) 传输流和节目流

TS-Transport Stream 翻译为“传输流”

PS-Program Stream 翻译为“节目流”

PS 用来传输和保存一道节目的编码数据或其他数据。PS 的组成单位是 PES 分组。

TS 用来传输和保存多道节目的编码数据或其他数据，TS 的组成单位是节目。

PS 适用于不容易发生错误的环境，以及涉及到软件处理的应用，典型应用如 DVD 光盘的文件存储

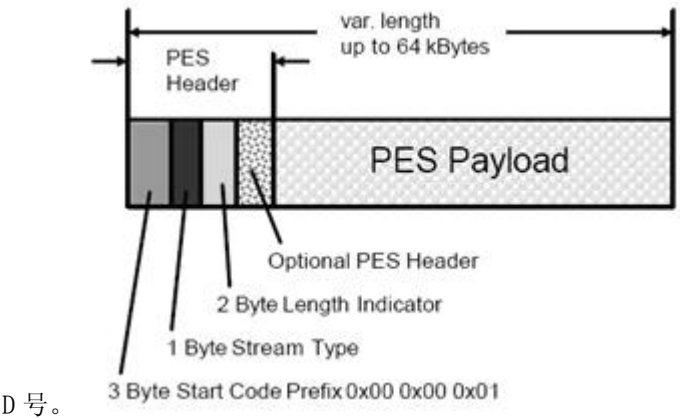
TS 适用于容易发生错误的环境，典型应用就是数字电视信号的传输。

TS 和 PS 是可以互相转换的，比如从 TS 中抽取一道节目的内容并产生有效的 PS 是可能。

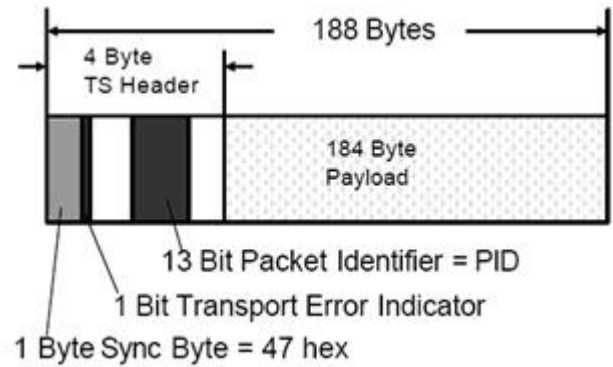
(5) 传输流分组和 PES 分组

原始流分成很多 PES 分组，保持串行顺序，一个 PES 分组只包含一个原始流的编码数据。PES 分组长度很大，最大可为 64K 字节。

PES 分组分为“分组首部(header)”和“有效负载(payload)”。“有效负载”指跟随在首部字节之后的字节。首部的前 4 个字节构成分组的起始码，标识了该分组所属原始流的类型和 I

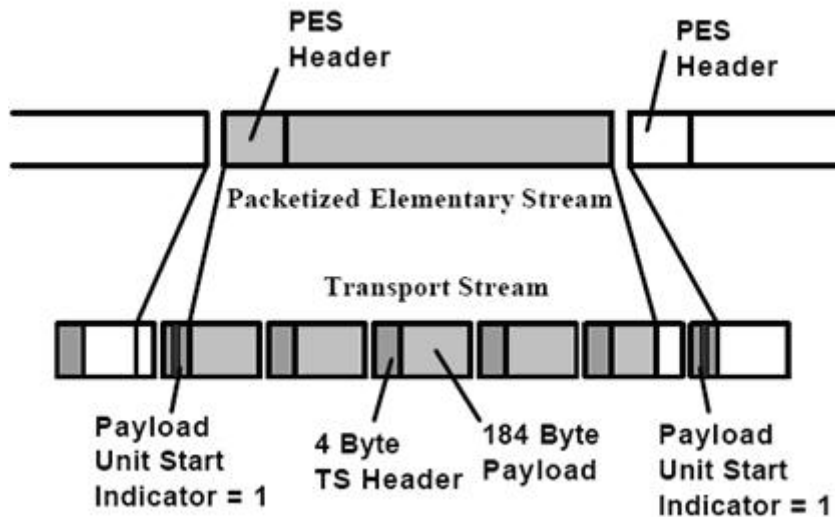


TS 分组也就是传输流数据形成的数据包。每个 TS 分组长度为 188 字节，包括“分组首部”和“有效负载，前 4 个字节是分组首部，包含了这个分组的一些信息。有些情况下需要更多的信息时，需在后面添加“调整字段(adaption field)”。



两者之间的关系：

PES 分组是插入到 TS 分组中的，每个 PES 分组首部的第一字节就是 TS 分组有效负载的第一字节。一个 PID 值的 TS 分组只带有来自一个原始流的数据。



#### (5)PSI

全称 Program Specific Information，意为节目专用信息。传输流中是多路节目复用的，那么，怎么知道这些节目在传输流中的位置，区分属于不同节目呢？所以就还需要一些附加信息，这就是 PSI。PSI 也是插入到 TS 分组中的，它们的 PID 是特定值。

MPEG-2 中规定了 4 个 PSI，包括 PAT(节目关联表)，CAT(条件访问表)，PMT(节目映射表)，NIT(网络信息表)，这些 PSI 包含了进行多路解调和显示节目的必要的和足够的信息。

具体的应用中可能包括更多的信息，比如 DVB-T 中定义了 SDT(服务描述表),EIT(环境信息表),BAT(节目组相关表),TDT(时间日期表)等，统称为 DVB-SI(服务信息)。

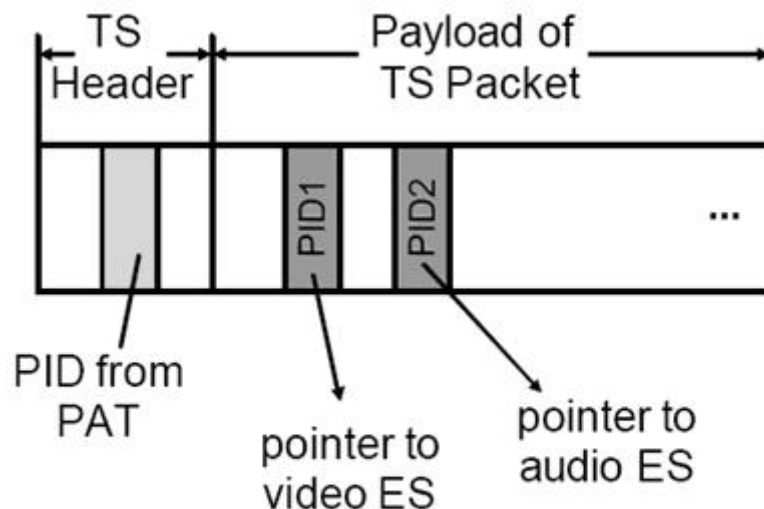
- PSI 的 PID 是特定的，含 PSI 的数据包必须周期性的出现在传输流中。

PMT (Program Map Table ) 节目映射表

PMT 所在分组的 PID 由 PAT 指定，所以要先解出 PAT，再解 PMT

PMT 中包含了属于同一节目的视频、音频和数据原始流的 PID。

找到了 PMT，解多路复用器就可找到一道节目对应的每个原始流的 PID，再根据原始流 PID，去获取原始流。如下图：PID1 和 PID2 分别对应某道节目的视频原始流和音频原始流的 PID。

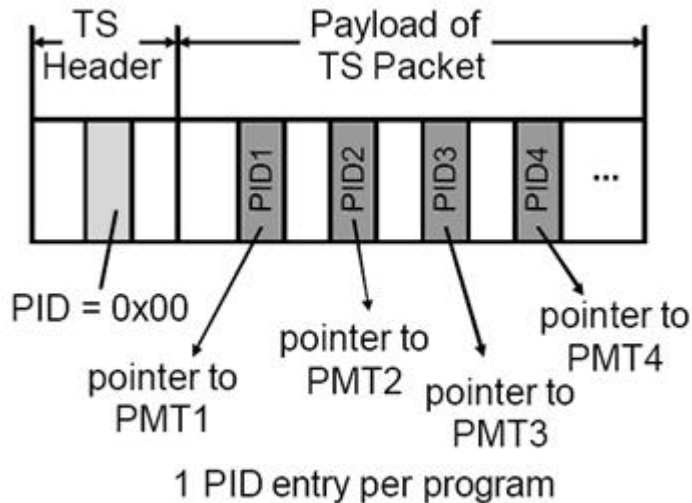


1 PID entry per elementary stream

- PAT (Program Association Table ) 节目关联表
- PAT 所在分组的 PID=0

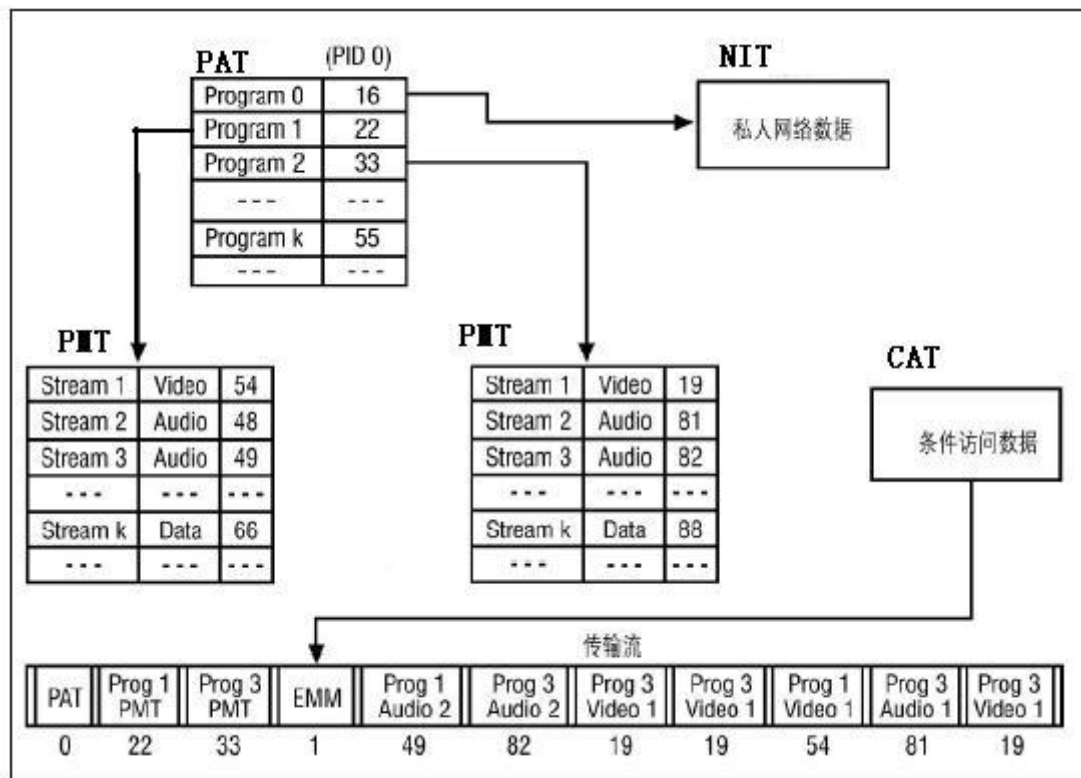
PAT 中列出了传输流中存在的节目流

- 
- PAT 指定了传输流中每个节目对应 PMT 所在分组的 PID
- PAT 的第一条数据指定了 NIT 所在分组的 PID，其他数据指定了 PMT 所在分组的 PID，如下图所示：



- CAT (Conditional Access Table )条件访问表
- CAT 所在分组的 PID=1
- CAT 中列出了条件控制信息(ECM)和条件管理信息(EMM)所在分组的 PID。
- CAT 用于节目的加密和解密
- NIT( Network Information Table)网络信息表
- NIT 所在分组的 PID 由 PAT 指定
- NIT 提供一组传输流的相关信息，以及于网络自身特性相关的信息，比如网络名称，传输参数(如频率, 调制方式等)。
- NIT 一般是解码器内部使用的数据，当然也可以做为 EPG 的一个显示数据提供给用户做为参考。

几种 PSI 之间的关系，如下图所示：首先 PAT 中指定了传输流中所存在的节目，及每个节目对应的 PMT 的 PID 号。比如 Program 1 对应的 PMT 的 PID=22,然后找到 PID=22 的 TS 分组，解出 PMT，得到这个节目中包含的原始流的 PID，再根据原始流的 PID 去找相应的 TS 分组，获取原始流的数据，然后就可以送入解码器解码了。



## 二、数据结构

### (1) TS 分组

前面提到，TS 分组由 188 个字节构成，其结构如下：

```
transport_packet()
{
    sync_byte                // 8
    transport_error_indicator //1
    payload_unit_start_indicator //1
    transport_priority        // 1
    PID                      //13
    transport_scrambling_control // 2
    adaptation_field_control  //2
    continuity_counter        //4
    if(adaptation_field_control=='10' || adaptation_field_control=='11'){
        adaptation_field()
    }
    if(adaptation_field_control=='01' || adaptation_field_control=='11') {
        for (i=0;i<N;i++){
            data_byte                //8
        }
    }
}
```

前面 32bit 的数据即 TS 分组首部，它指出了这个分组的属性。

**sync\_byte** 同步字节，固定为 0x47，表示后面的是一个 TS 分组，当然，后面包中的数据是不会出现 0x47 的

**transport\_error\_indicator** 传输错误标志位，一般传输错误的话就不会处理这个包了

**payload\_unit\_start\_indicator** 这个位功能有点复杂，字面意思是有效负载的开始标志，根据后面有效负载的内容不同功能也不同，后面用到的时候再说。

**transport\_priority** 传输优先级位，1 表示高优先级，传输机制可能用到，解码好像用不着。

**PID** 这个比较重要，指出了这个包的有效负载数据的类型，告诉我们这个包传输的是什么内容。前面已经叙述过。

**transport\_scrambling\_control** 加密标志位，表示 TS 分组有效负载的加密模式。TS 分组首部(也就是前面这 32bit)是不应被加密的，00 表示未加密。

**adaption\_field\_control** 翻译为“调整字段控制”，表示 TS 分组首部后面是否跟随有调整字段和有效负载。01 仅含有效负载，10 仅含调整字段，11 含有调整字段和有效负载。为 00 的话解码器不进行处理。空分组没有调整字段。

**continuity\_counter** 一个 4bit 的计数器，范围 0-15，具有相同的 PID 的 TS 分组传输时每次加 1，到 15 后清 0。不过，有些情况下是不计数的。如下：(1)TS 分组无有效负载(2)复制的 TS 分组和原分组这个值一样(3)后面讲到的一个标志 **discontinuity\_indicator** 为 1 时

**adaptation\_field()** 调整字段的处理

**data\_byte** 有效负载的剩余部分，可能为 PES 分组，PSI，或一些自定义的数据。



## (2)PAT

PAT 数据结构如下:

```

program_association_section() {
    table_id // 8
    section_syntax_indicator //1
    '0' //1
    reserved // 2
    section_length //12
    transport_stream_id // 16
    reserved // 2
    version_number // 5
    current_next_indicator //1
    section_number //8
    last_section_number // 8
    for (i=0; i<N;i++) {
        program_number // 16
        reserved // 3
        if(program_number == '0') {
            network_PID // 13
        }
        else {
            program_map_PID // 13
        }
    }
    CRC_32 // 32
}

```

**table\_id** 固定为 0x00，标志是该表是 PAT

**section\_syntax\_indicator** 段语法标志位，固定为 1

**section\_length** 表示这个字节后面有用的字节数，包括 CRC32。假如后面的字节加上前面的字节数少于 188，后面会用 0xFF 填充。假如这个数值比较大，则 PAT 会分成几部分来传输。

**transport\_stream\_id** 该传输流的 ID，区别于一个网络中其它多路复用的流。

**version\_number** 范围 0-31，表示 PAT 的版本号，标注当前节目的版本。这是个非常有用的参数，当检测到这个字段改变时，说明 TS 流中的节目已经变化了，程序必须重新搜索节目。

**current\_next\_indicator** 表示发送的 PAT 是当前有效还是下一个 PAT 有效。

**section\_number** 分段的号码。PAT 可能分为多段传输，第一段为 00，以后每个分段

加 1，最多可能有 256 个分段

**last\_section\_number** 最后一个分段的号码

**program\_number** 节目号

**network\_PID** 网络信息表（NIT）的 PID，网络信息表提供了该物理网络的一些信息，和电视台相关的。节目号为 0 时对应的 PID 为 **network\_PID**

**program\_map\_PID** 节目映射表的 PID，节目号大于 0 时对应的 PID，每个节目对应一个

**CRC\_32** CRC32 校验码

上面 **program\_number**，**network\_PID**，**program\_map\_PID** 是循环出现的。**program\_number** 等于 0 时对应 **network\_PID**，**program\_number** 等于其它值时对应 **program\_map\_PID**。

举个例子，下述流为带 PAT 的 TS 分组：

**47 40 00 1c 00 00 b0 15 13 f6 e7 00 00 00 00 e0 10 00 01 e0 20 00 02 e0 21 1a 34 b4 77 ff.....ff**

其中红色的四个字节是 TS 分组头部，用数据结构解出首部，得到 **PID=0x00**，表示为该分组的有效负载是 PAT。蓝色的 00 称为“指针域”----**Pointer field**，表示了一个偏移量，即从后面第几个字节开始是 PAT 部分。为 00 表示后面紧接着的就是 PAT：**00 b0 15 13 f6 e7 00 00 00 00 e0 10 00 01 e0 20 00 02 e0 21 1a 34 b4 77**

再利用 PAT 的数据结构解出 PAT，得到如下信息：

-----PAT Information-----

table\_id: 00

section\_syntax\_indicator: 01

section\_length: 0015

transport\_stream\_id: 13f6

version\_number: 13

current\_next\_indicator: 01

section\_number: 00

last\_section\_number: 00

program\_number: 0000

network\_PID: 0010

program\_number: 0001

program\_map\_PID: 0020

program\_number: 0002

program\_map\_PID: 0021

CRC\_32: 1a34b477

可以看出，此 PAT 只有一段，包含了三个节目，节目号 0000 对应于 **network\_PID=0010**，节目号 0001 对应于 **program\_map\_PID =0020**，节目号 0002 对应于 **program\_map\_PID =0021**，从实际的角度，我们应该把这三个节目号理解为三个频道，第一个频道中的内容是网络信息，第二、三个频道包含了节目信息。在数字电视中，一个频道即对应于一个频点，如 498MHz，一个频道上可以有多个节目，后面的 PMT 即是告诉了我们某个频道中所有节目对应的 PID。

于是现在就搜寻 **PID=0x0020** 的 TS 分组，即是频道 2 对应的 PMT 信息。

## MPEG-2 学习笔记(五)

烧火棍 发表于 2008-1-30 16:26:00

### (3)PMT

PMT 数据结构如下:

```
TS_program_map_section() {
    table_id // 8
    section_syntax_indicator //1
    '0' // 1
    reserved // 2
    section_length // 12
    program_number //16
    reserved // 2
    version_number //5
    current_next_indicator //1
    section_number // 8
    last_section_number //8
    reserved //3
    PCR_PID //13
    reserved 4
    program_info_length //12
    for (i=0; i<N; i++) {
        descriptor()
    }
    for (i=0;i<N1;i++) {
        stream_type //8
        reserved //3
        elementary_PID //13
        reserved //4
        ES_info_length //12
        for (i=0; i<N2; i++) {
            descriptor()
        }
    }
    CRC_32 //32
}
```

**table\_id** 固定为 0x02 , 标志是该表是 PMT

**section\_syntax\_indicator**

**section\_length**

**version\_number**

**current\_next\_indicator** 以上四个字段意思和 PAT 相同, 可参考上面解释

**section\_number**

**last\_section\_number** 以上两个字段意思和 PAT 相同，不过值都固定为 0x00，我觉得这样的原因可能是因为 PMT 不需要有先后顺序，因为先定义哪个节目都是无所谓。

**program\_number** 节目号，表示该 PMT 对应的节目

**PCR\_PID** PCR（节目时钟参考）所在 TS 分组的 PID，根据 PID 可以去搜索相应的 TS 分组，解出 PCR 信息。

**program\_info\_length** 该节目的信息长度，在此字段之后可能会有一些字节描述该节目的信息

**stream\_type** 指示了 PID 为 elementary\_PID 的 PES 分组中原始流的类型，比如视频流，音频流等，见后面的表

**elementary\_PID** 该节目中包括的视频流，音频流等对应的 TS 分组的 PID

**ES\_info\_length** 该节目相关原始流的描述符的信息长度。

stream\_type 对应的类型：

Value↕	Description↕
0x00↕	ITU-T   ISO/IEC Reserved↕
0x01↕	ISO/IEC 11172 Video↕
0x02↕	ITU-T Rec. H.262   ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream↕
0x03↕	ISO/IEC 11172 Audio↕
0x04↕	ISO/IEC 13818-3 Audio↕
0x05↕	ITU-T Rec. H.222.0   ISO/IEC 13818-1 private_sections↕
0x06↕	ITU-T Rec. H.222.0   ISO/IEC 13818-1 PES packets containing private data↕
0x07↕	ISO/IEC 13522 MHEG↕
0x08↕	ITU-T Rec. H.222.0   ISO/IEC 13818-1 Annex A DSM CC↕
0x09↕	ITU-T Rec. H.222.1↕
0x0A↕	ISO/IEC 13818-6 type A↕
0x0B↕	ISO/IEC 13818-6 type B↕
0x0C↕	ISO/IEC 13818-6 type C↕
0x0D↕	ISO/IEC 13818-6 type D↕
0x0E↕	ISO/IEC 13818-1 auxiliary↕
0x0F-0x7F↕	ITU-T Rec. H.222.0   ISO/IEC 13818-1 Reserved↕
0x80-0xFF↕	User Private↕

还是举个例子，下述是一个包含 PMT 的 TS 分组，

**47 40 20 1c 00 02 b0 1f 00 01 e7 00 00 e1 00 f0 00 02 e1 00 f0 05 02 03 b2 44 5f 04 e1 10 f0 03 03 01 67 c9 ab c8 d2**

红色的四个字节是 TS 分组头部，蓝色的 00 是“指针域”，意义同 PAT 中的指针域。所以下面的数据就是 PMT 的内容：**02 b0 1f 00 01 e7 00 00 e1 00 f0 00 02 e1 00 f0 05 02 03 b2 44 5f 04 e1 10 f0 03 03 01 67 c9 ab c8 d2**

再解出 PMT，得到下列信息：

table\_id: 02  
section\_syntax\_indicator: 01  
section\_length: 01f  
program\_number: 0001  
version\_number: 13  
current\_next\_indicator: 01  
section\_number: 00  
last\_section\_number: 00  
PCR\_PID: 0100  
program\_info\_length: 000  
descriptor:  
stream\_type: 00  
elementary\_PID: 0001  
ES\_info\_length: 000  
descriptor:  
stream\_type: 02  
elementary\_PID: 0001  
ES\_info\_length: 005  
descriptor: 02 03 b2 44 5f  
stream\_type: 04  
elementary\_PID: 0011  
ES\_info\_length: 003  
descriptor: 03 01 67  
CRC\_32: c9abc8d2

可以看出，该节目号 0001 包含了三个流的信息，流类型分别为 00，02，04，00 的流为保留值，可以不考虑，02 表示原始流为视频流，其 elementary\_PID 为 0001，04 表示原始流为音频流，其 elementary\_PID 为 0011，两个流分别还带有 descriptor（描述符），说明了该原始流的一些信息。

得到了这个 elementary\_PID，再从后面的传输流中找到 PID 为这个值的 TS 分组，其有效负载即为这个原始流的数据，获取数据送到解码器，即可还原这个视频或音频了。

上面的都是一些零散的知识,跟我们实际应用有什么关系呢?下面就是一个简易的应用过程---搜台。搜台过程大致如下:

上述过程主要涉及到 PAT 和 PMT 的一些解码和解复用知识，这也是目前我学习到的，当然，数字电视涉及到的知识远远不止这些，解码方面就还包括调整字段的处理，SI(业务信息)应用，时钟的处理，CA 加密解 MI 系统等，还需要继续的学习和实践。

TSParse

C:\Documents and Settings\Administrator\桌面\new.ts

Open(O)

File Length: 752000 Total Packet: 4000

47 1f ff 10 ff ff ff ff ff ff ff ff ff ff ff ff ff ff  
 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  
 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  
 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  
 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  
 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

PACKET NO. 1

GO! (G) <-- -->

TS Clear (C) Parse (P)

Head Information

Packet NO.: 1  
 sync\_byte: 47  
 transport\_error\_indicator: 00  
 payload\_unit\_start\_indicator: 00  
 transport\_priority: 00  
 PID: 1fff(NULL Packet)  
 transport\_scrambling\_control: 00  
 adaption\_field\_control: 01  
 continuity\_counter: 00

Payload

ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  
 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

Write Packet Info(W)

Write TS Info(T)

TSParse.rar