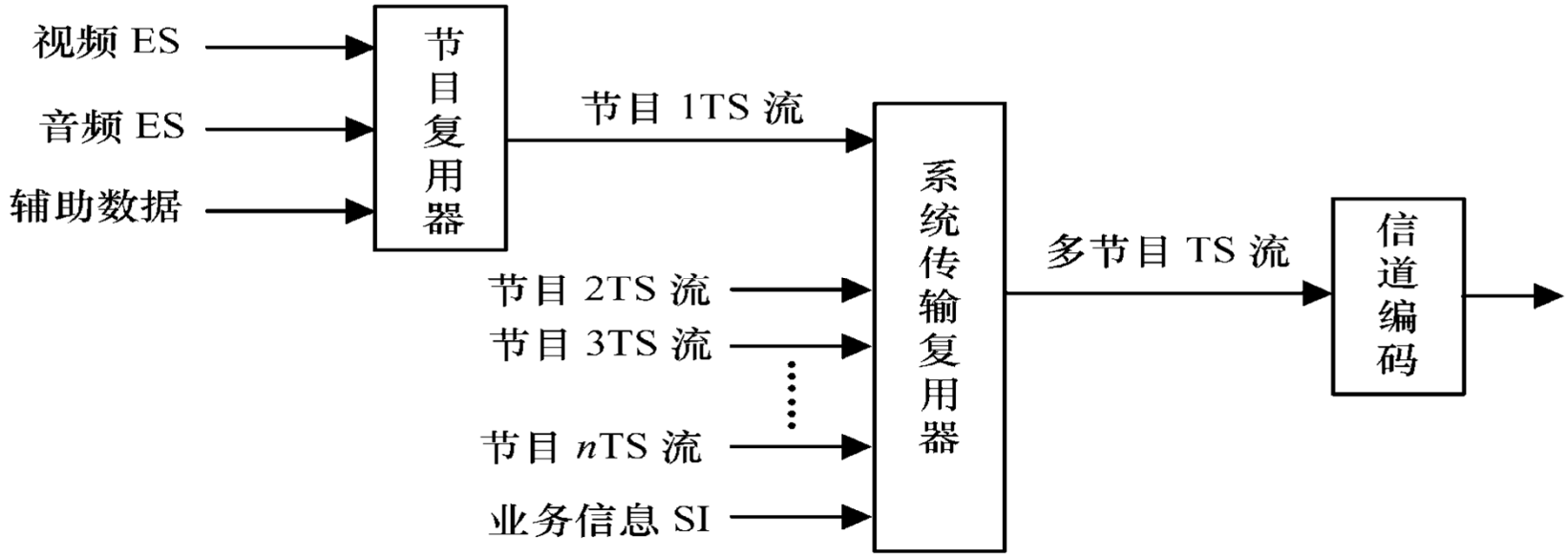
实验一：TS码流分析

一、实验目的

- 加深对**MPEG-2**系统复用的理解；
- 理解**TS**包包头中各参数的含义；
- 掌握使用**PSI**从码流中提取所需视音频节目**PID**号的方法。

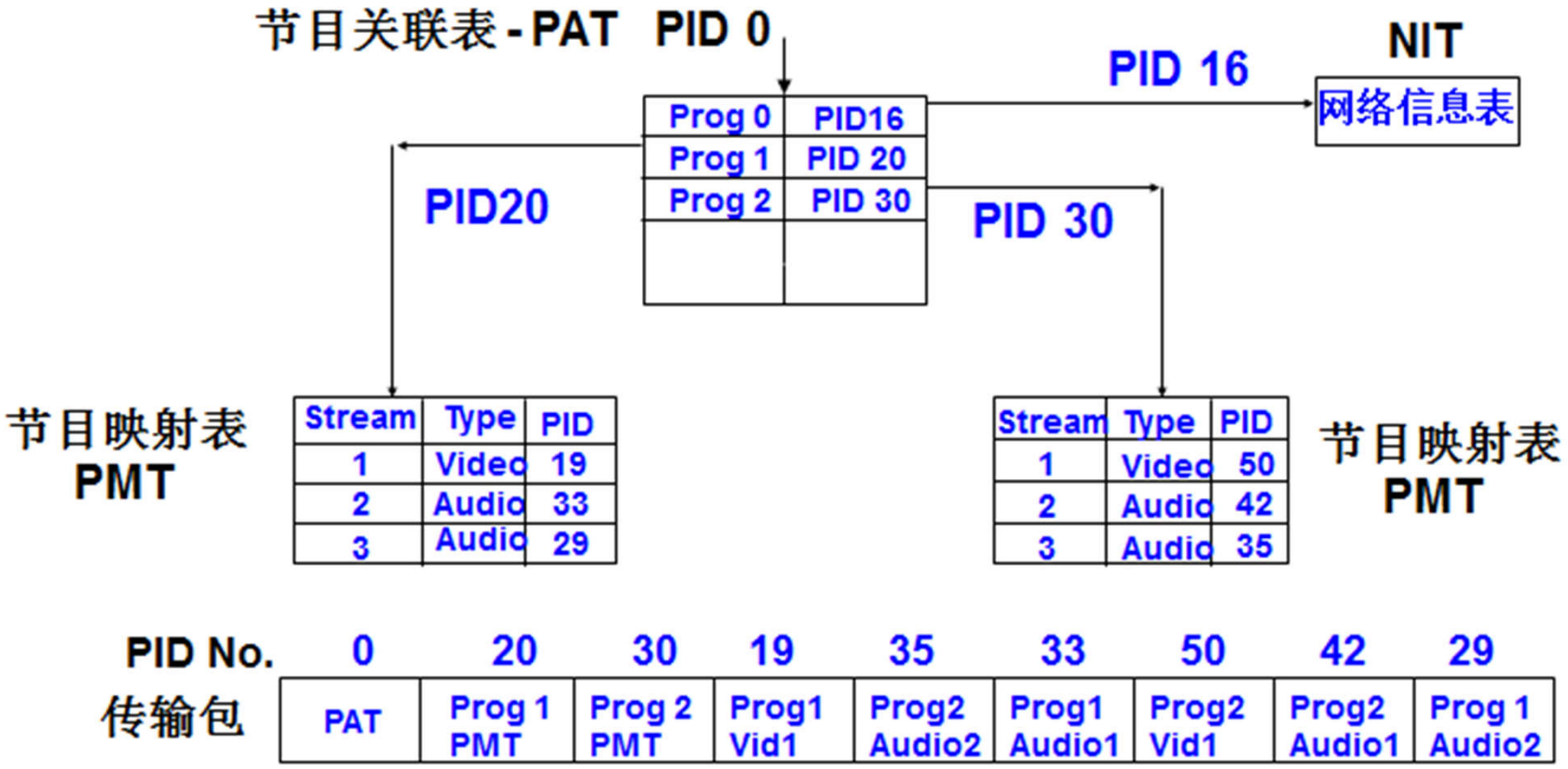
二、实验原理


视频 ES
音频 ES
辅助数据
节目复用器
节目 1TS 流
节目 2TS 流
节目 3TS 流
节目 nTS 流
业务信息 SI
系统传输复用器
多节目 TS 流
信道编码

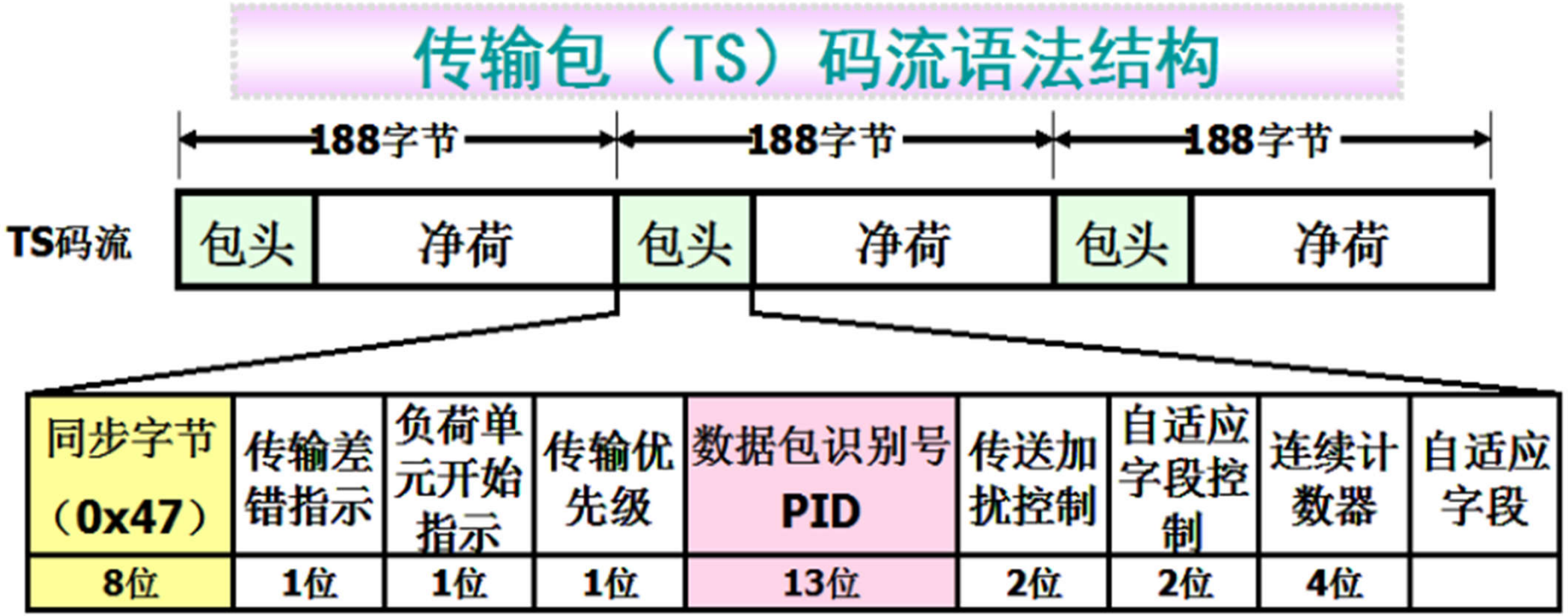

V1
A1
V1
V2
A1
A2
V2
V1
V1
A1
A2
V1
V2
V1

传输流

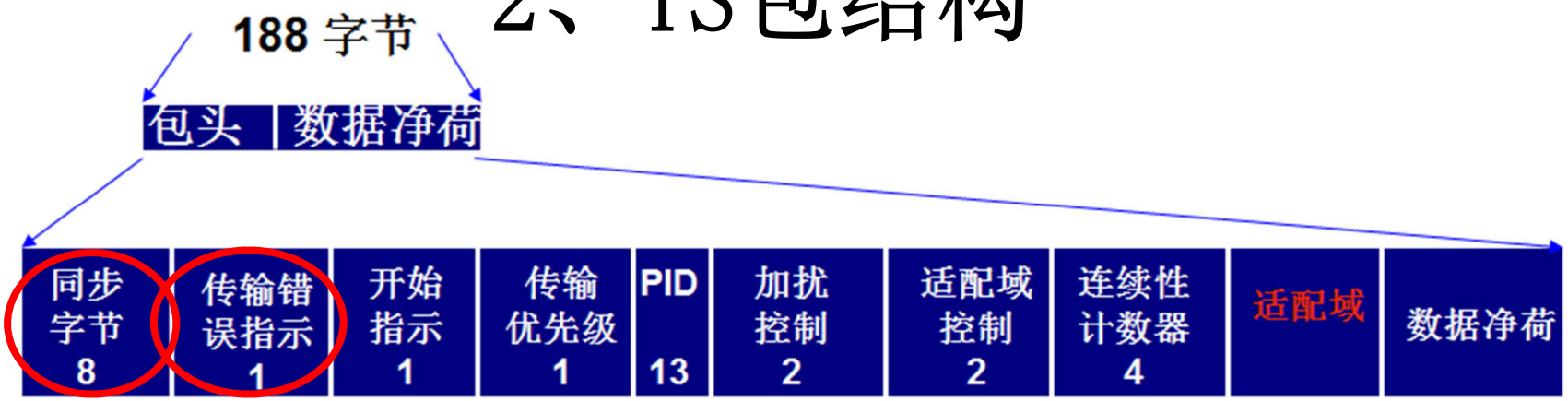
1、MPEG TS 流逻辑结构



2、TS包结构


传输包（TS）码流语法结构
188字节
188字节
188字节
TS码流
包头
净荷
包头
净荷
包头
净荷
同步字节（0x47）
传输差错指示
负荷单元开始指示
传输优先级
数据包识别号PID
传送加扰控制
自适应字段控制
连续计数器
自适应字段
8位
1位
1位
1位
13位
2位
2位
4位

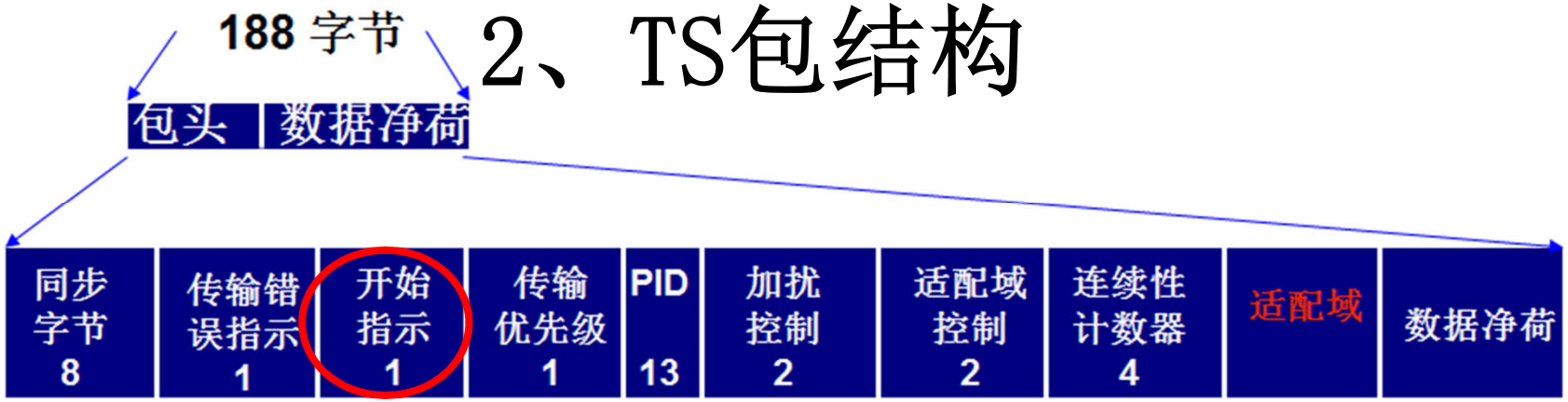
2、TS包结构



- **sync_byte**：固定为**0100 0111 (0x47)**的 **8** 位字段；

- **transport_error_indicator:1** 比特标志位。当置为 **1** 时表明在相关的传送分组中至少有一个不可纠正的错误位。当被置**1**后，在错误被纠正之前不能重置为**0**。

2、TS包结构

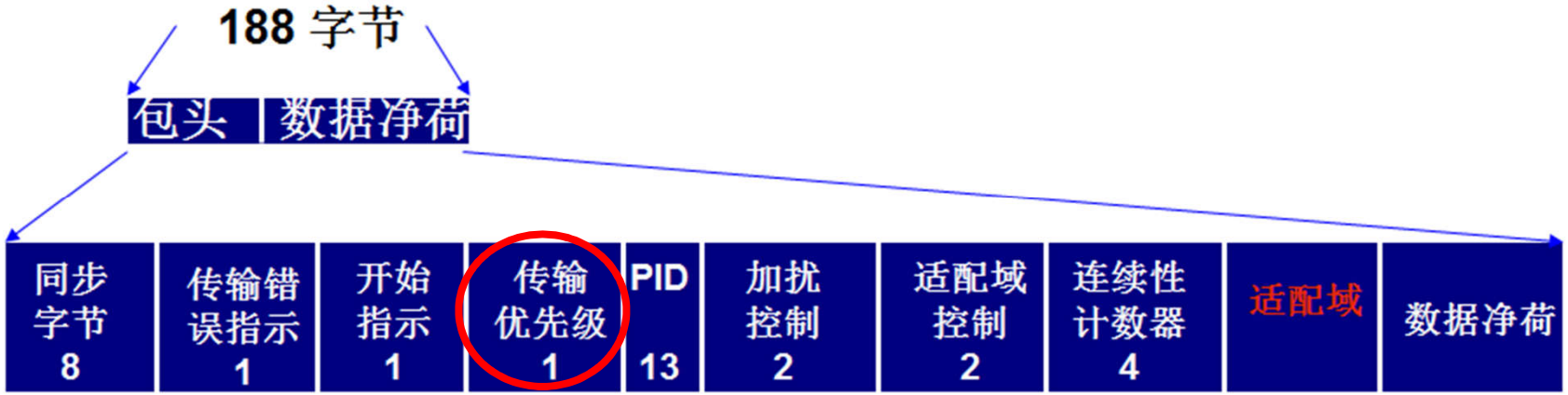


- **payload_unit_start_indicator:**1 比特标志位，用来指示传输流分组带有 **PES** 分组或**PSI**数据时的情况。

- 当传输流分组的有效负载带有 **PES** 分组数据时：置‘**1**’表明传输流分组的有效负载将以 **PES** 分组的第一个字节开始；

- 当传输流分组带有一个 **PSI**部分的第一个字节：置‘**1**’，表明传输流分组的第一个字节带有 **pointer_field**。置‘**0**’，表明传输流分组不带有一个 **PSI**部分的第一个字节，在有效负载中没有**pointer_field**。

2、TS包结构

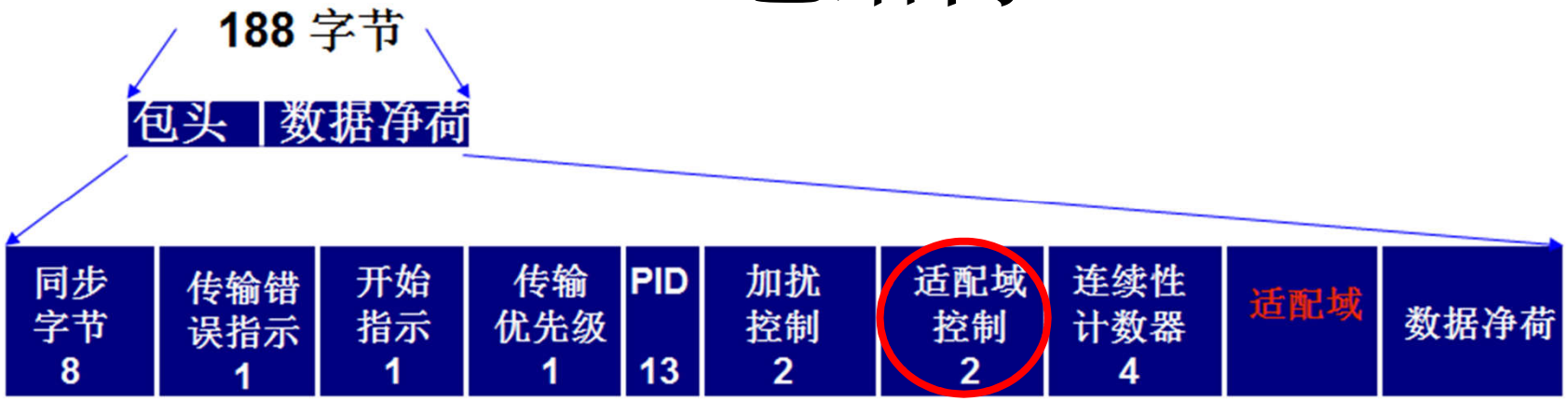


- **transport_priority**：1 位指示器。当被置为‘1’时表明相关的分组比其他具有相同PID 但此位没有被置‘1’的分组有更高优先级。

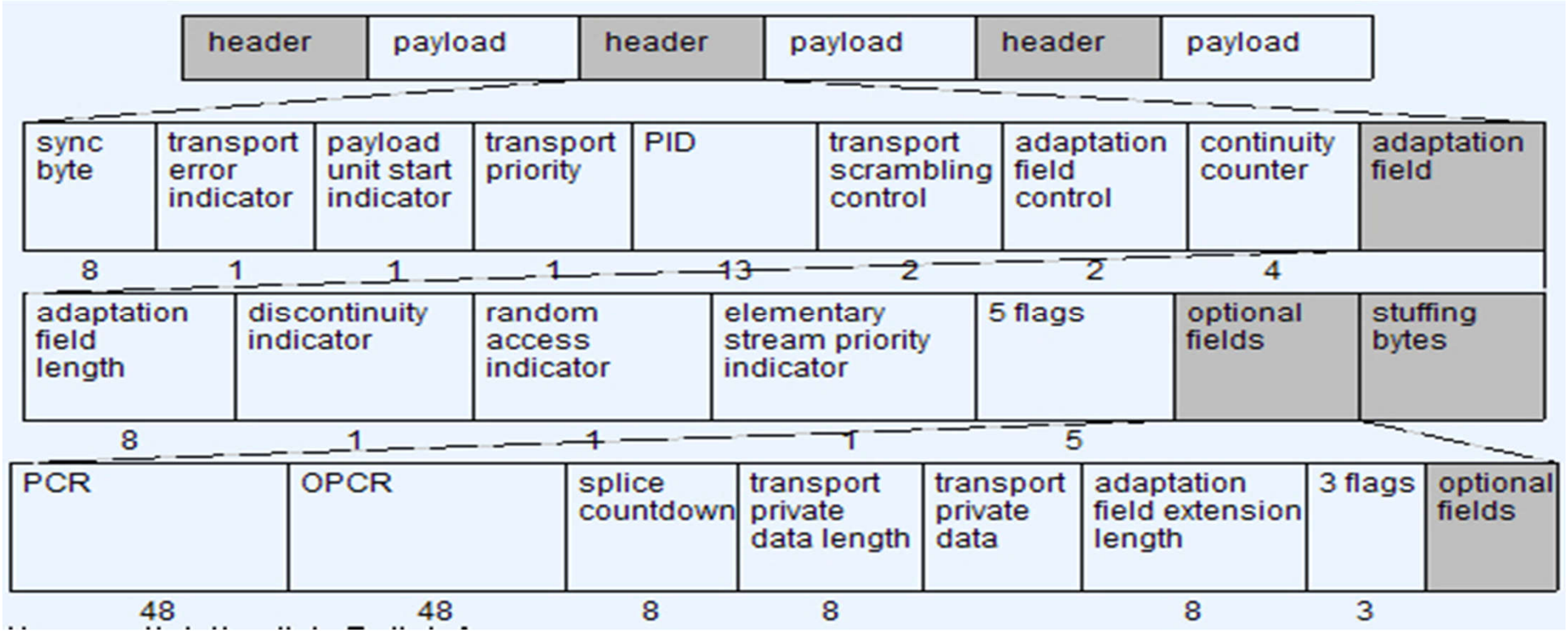
2、TS包结构

- **PID:13** 位字段，唯一识别携带某一类型数据的传输包。
- **transport_scrambling_control：2** 位字段，用来指示传输流分组有效负载的加密模式。

2、TS包结构

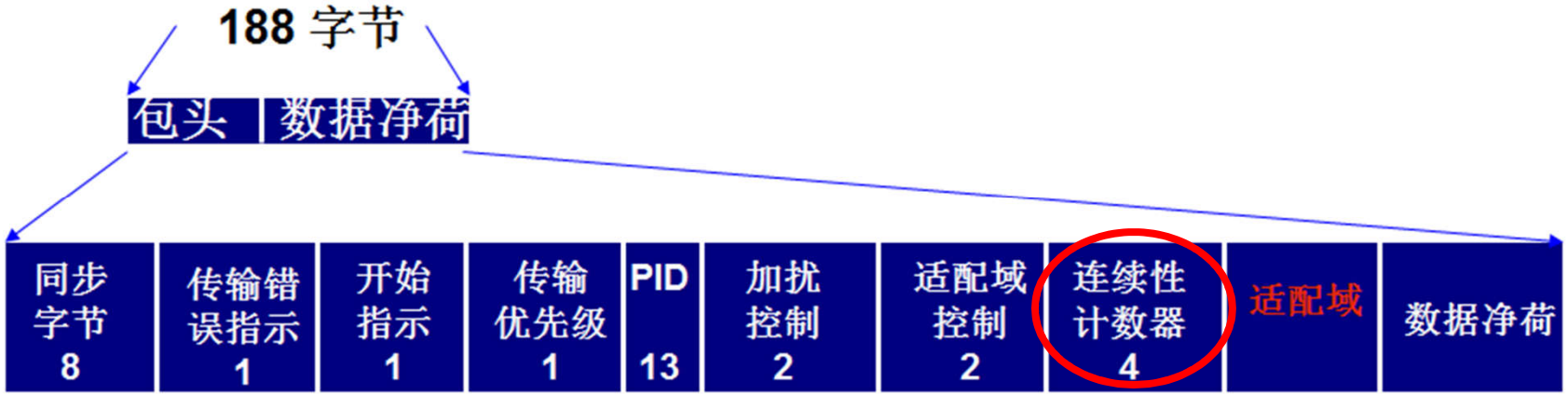


- **adaptation_field_control：2** 位字段。用于指示本传输流分组首部是否跟随有调整字段和 /或有效负载。 ”**00**”未来使用保留；“**01**”仅含有效载荷；“**10**”无有效载荷；“11”调整字段后为有效载荷。


header
payload
header
payload
header
payload
sync byte
transport error indicator
payload unit start indicator
transport priority
PID
transport scrambling control
adaptation field control
continuity counter
adaptation field
8
1
1
1
13
2
2
4
adaptation field length
discontinuity indicator
random access indicator
elementary stream priority indicator
5 flags
optional fields
stuffing bytes
8
1
1
1
5
PCR
OPCR
splice countdown
transport private data length
transport private data
adaptation field extension length
3 flags
optional fields
48
48
8
8
8
3

TS包结构

2、TS包结构



- **continuity_counter：** 4 位字段。随着每一个具有相同 PID 的传输流分组而增加，当它达到最大值后又回复到0。

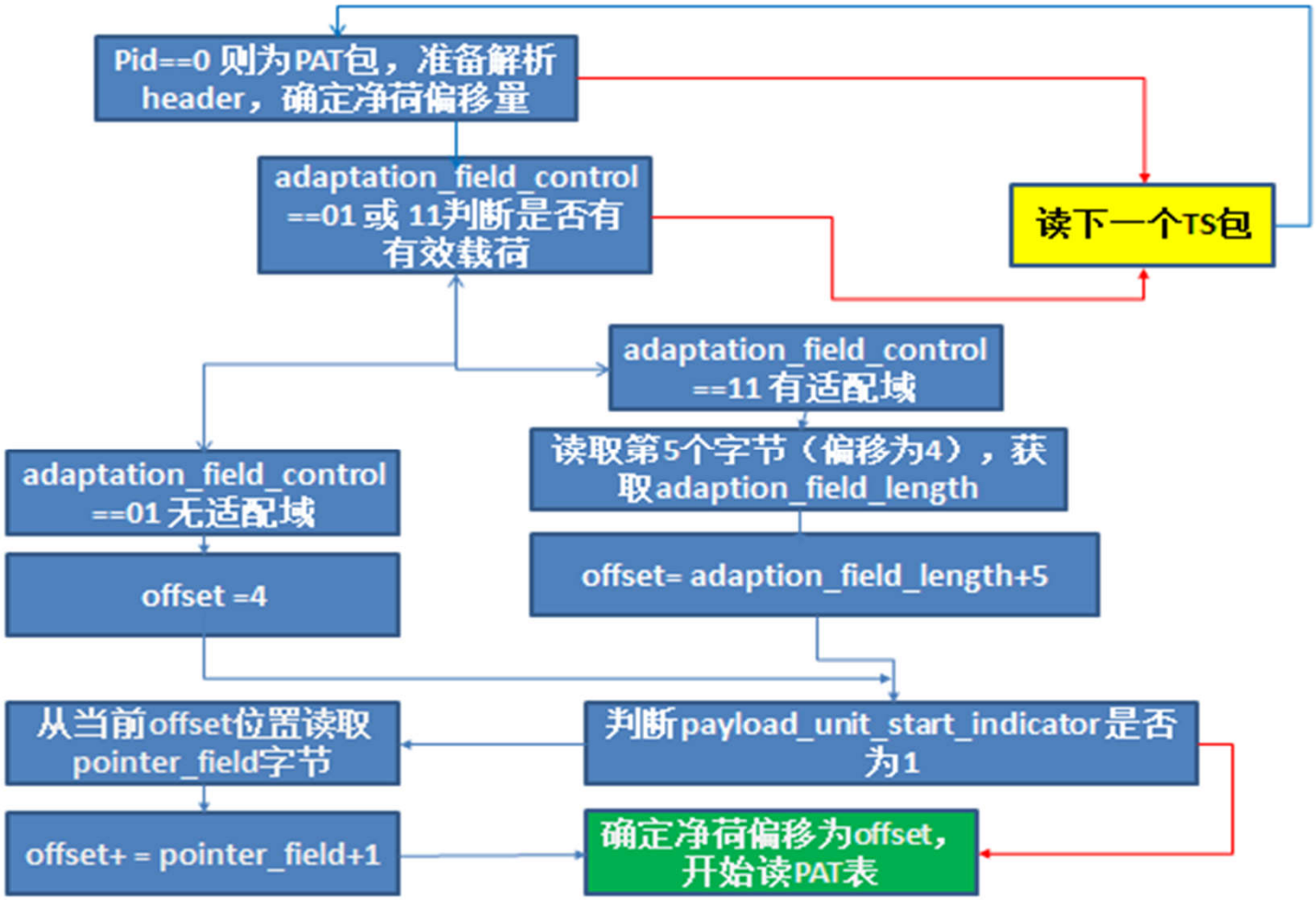

Pid==0 则为PAT包，准备解析header，确定净荷偏移量
adaptation_field_control ==01 或 11判断是否有有效载荷
读下一个TS包
adaptation_field_control ==11 有适配域
读取第5个字节（偏移为4），获取adaption_field_length
offset= adaption_field_length+5
adaptation_field_control ==01 无适配域
offset =4
从当前offset位置读取pointer_field字节
判断payload_unit_start_indicator是否为1
offset+ = pointer_field+1
确定净荷偏移为offset，开始读PAT表

Table 2-3 -- ITU-T Rec. H.222.0 | ISO/IEC 13818 transport packet

Syntax	No. of bits	Mnemonic
transport_packet(){		
TS包必有成分 **sync_byte**	**8**	**bslbf**
transport_error_indicator	**1**	**bslbf**
payload_unit_start_indicator	**1**	**bslbf**
transport_priority	**1**	**bslbf**
PID	**13**	**uimsbf**
transport_scrambling_control	**2**	**bslbf**
adaptation_field_control	**2**	**bslbf**
continuity_counter	**4**	**uimsbf**
if(adaptation_field_control=='10' || adaptation_field_control=='11'){		
adaptation_field() 自适应区：包含PCR等信息		
}		
if(adaptation_field_control=='01' || adaptation_field_control=='11') {		
for (i=0;i<N;i++){		
data_byte 有效载荷区：	**8**	**bslbf**
}		
}		
}		

3、PAT

- PAT表由PID为0x0000的TS包传送，作用是为复用的每一路传送流提供出所包含的节目和节目编号，以及对应节目的节目映射表（PMT）的位置，即PMT的TS包的包标识符（PID）的值，同时还提供网络信息表（NIT）的位置，即NIT的TS包的包标识符（PID）的值。

PAT (PID=0)

节目	PMT的PID
节目0	16
节目1	22
节目3	33
……	……

NIT (PID=16)

网络私有数据

4、PAT结构

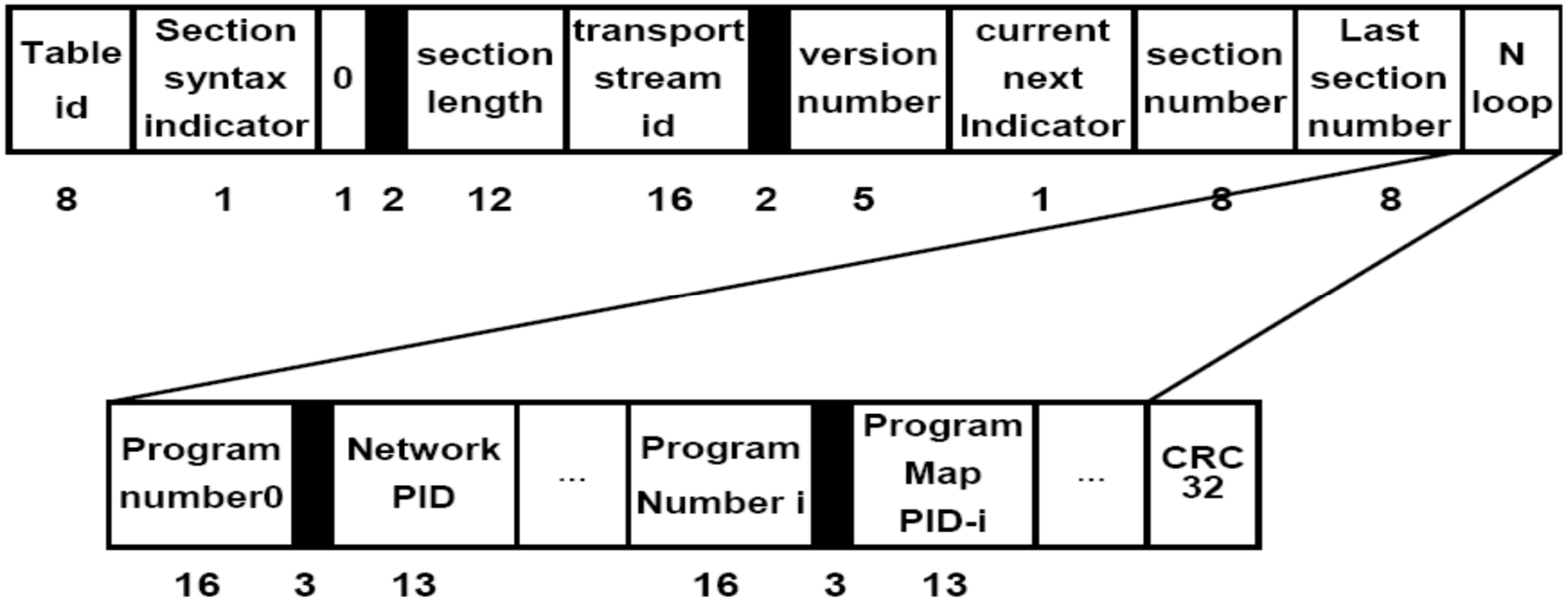
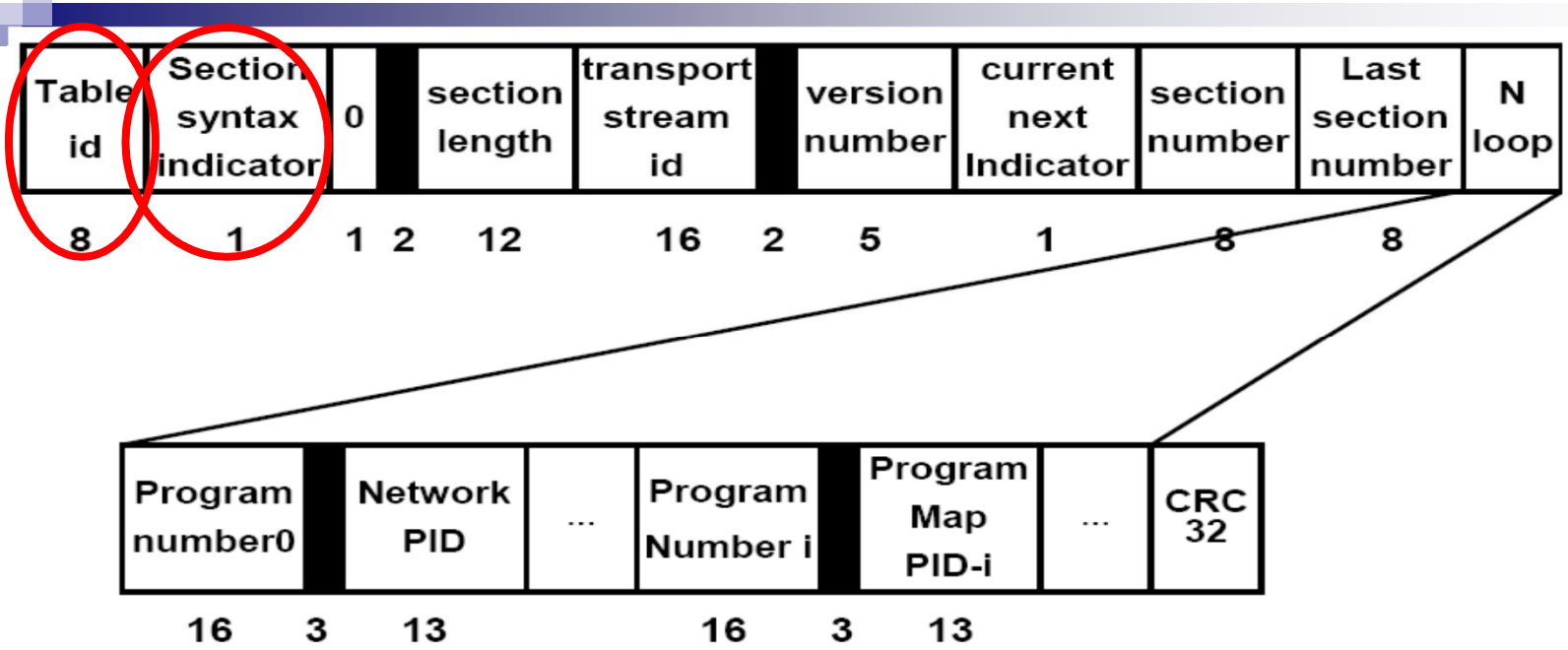
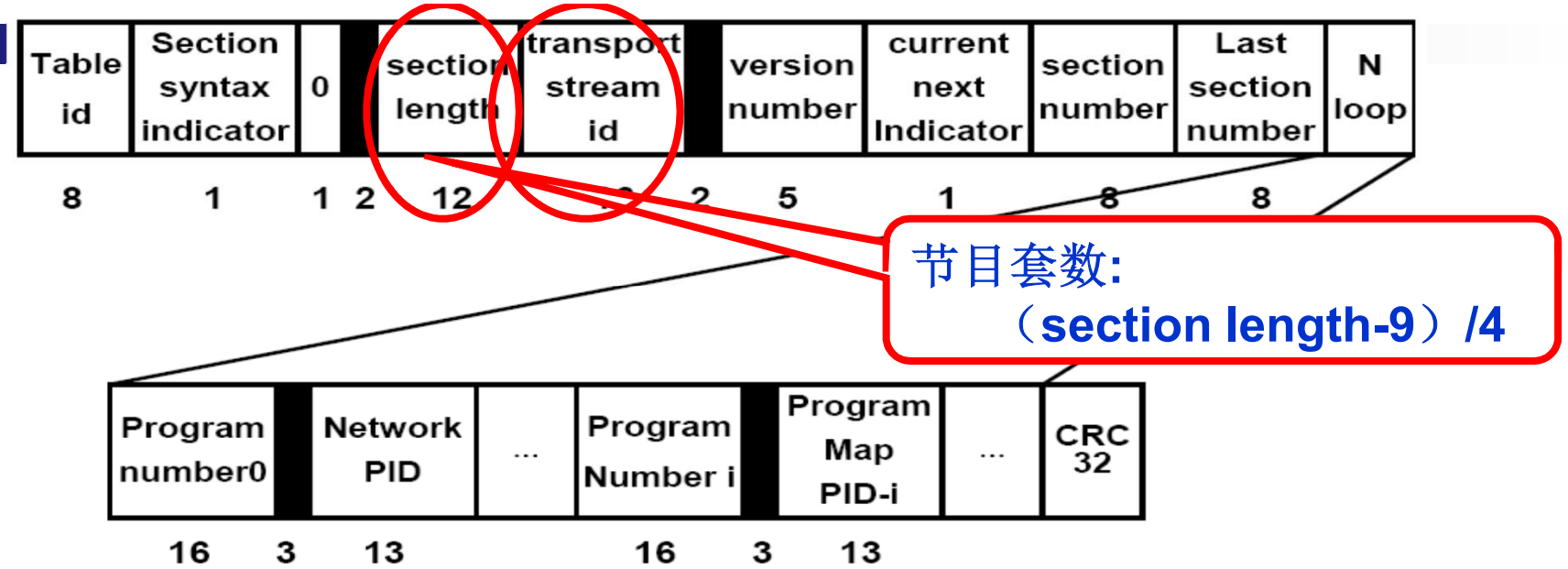


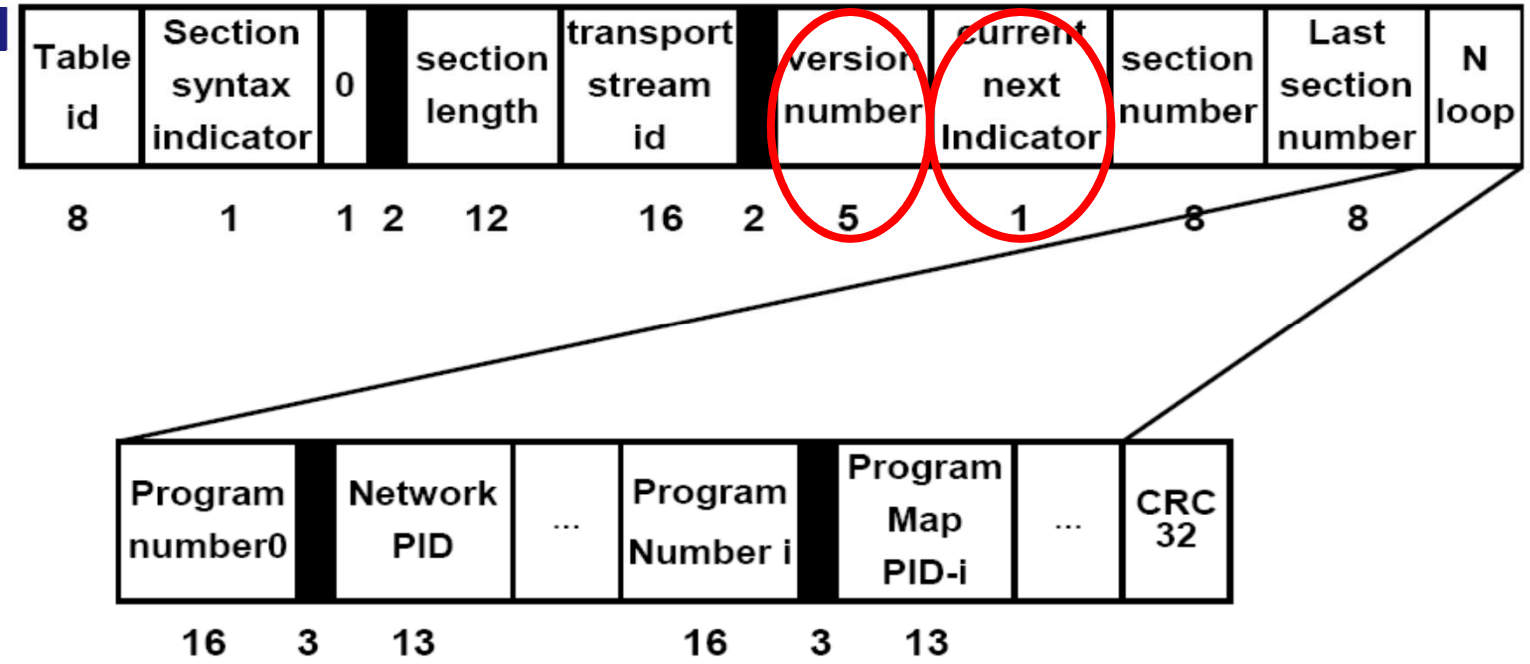
Table id
Section syntax indicator
0
section length
transport stream id
version number
current next Indicator
section number
Last section number
N loop
8
1
1
2
12
16
2
5
1
8
8
Program number0
Network PID
...
Program Number i
Program Map PID-i
...
CRC 32
16
3
13
16
3
13



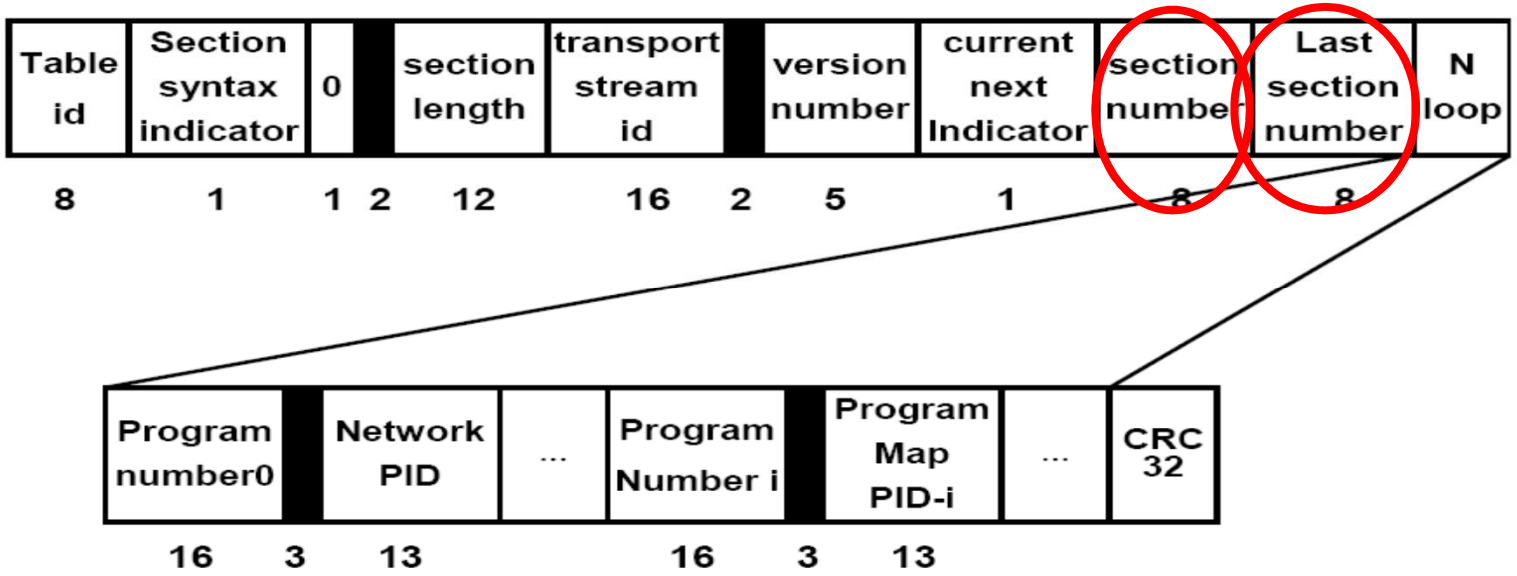
- **table_id：8位字段。固定为0x00，标志该表是PAT表。**
- **section_syntax_indicator：1位字段，段语法标志位，固定为1。**



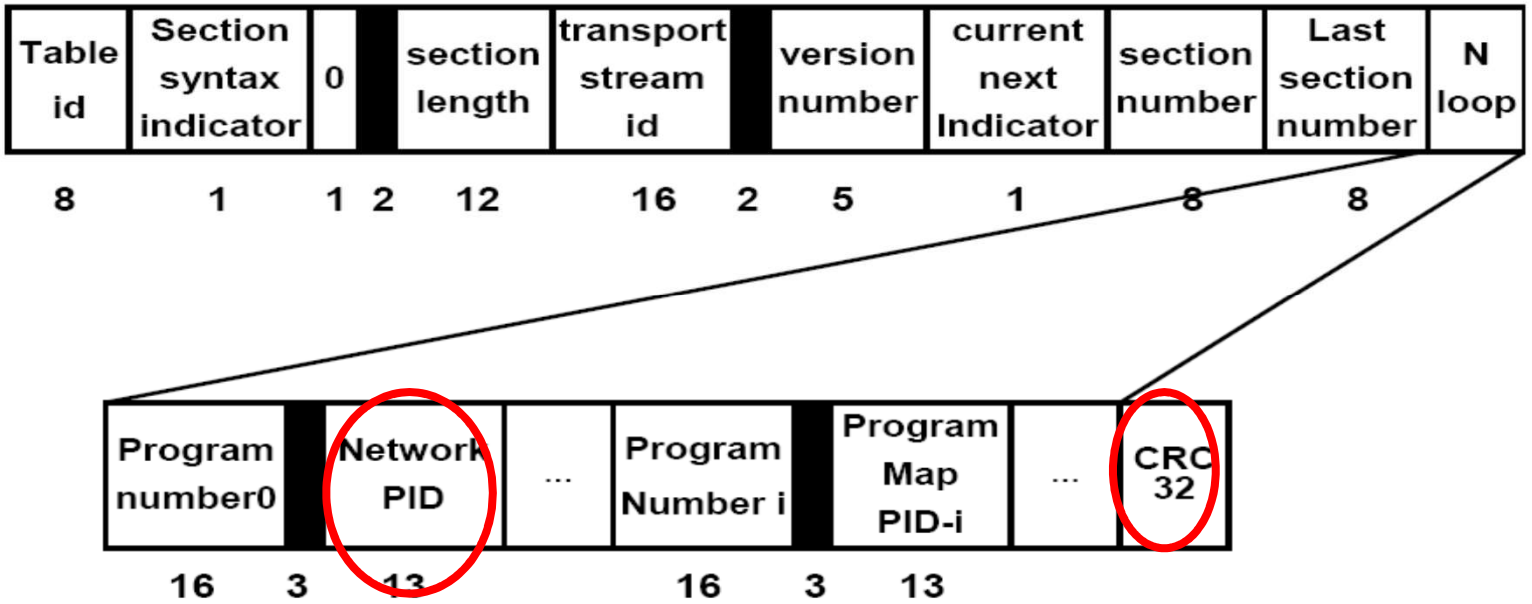
- **section_length：12位字段，表示这个字节后面有用的字节数，包括CRC32。**
- **transport_stream_id：16位字段，表示该传输流的ID，区别于同一个网络中其它多路复用流。**



- **version_number**：5位字段，表示**PAT**的版本号。
- **current_next_indicator**：1位字段，表示发送的**PAT**表是当前有效还是下一个**PAT**有效。



- **section_number**：8位字段，表示分段的号码。**PAT**可能分为多段传输，第一段为**0**，以后每个分段加**1**，最多可能有**256**个分段。
- **last_section_number**：8位字段，表示**PAT**最后一个分段的号码。



- **network_PID：13位字段，表示网络信息表（NIT）的PID，节目号为0时对应ID为network_PID。**
- **CRC_32：32位字段，CRC32校验码Cyclic Redundancy Check。**

PAT表的段结构：

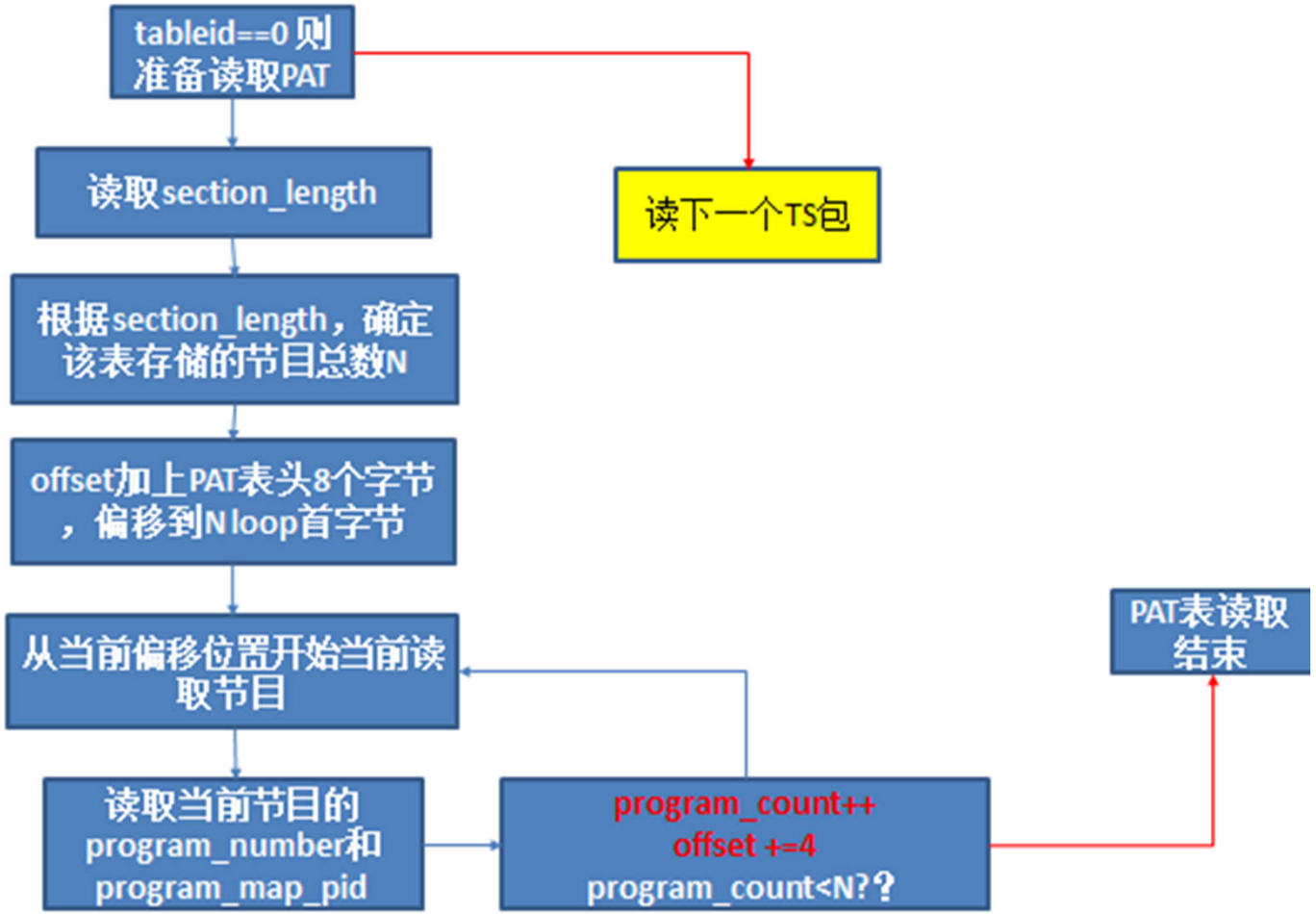
Syntax	No. of bits	Mnemonic
program_association_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i=0; i<N;i++) {		
program_number	16	uimsbf
reserved	3	bslbf
if(program_number == '0') {		
network_PID	13	uimsbf
}		
else {		
program_map_PID	13	uimsbf
}		
}		
CRC_32	32	rpchof
}		

某TS流中用于传输PAT表格信息的段：

table_id = 0x00	表明这个段是属于PAT表的
section_syntax_indicator=1	
section_length = 0x55	该段的长度
…	
version_number = 1	
current_next_indicator = 1	
section_number = 0x00	表明这个段是该PAT表第一个段
last_section_number = 0x02	该PAT表总共分成了多少个段
…	

table_id = 0x00	表明这个段是属于PAT表的
…	
section_number = 0x01	表明这个段是该PAT表第二个段
last_section_number = 0x02	该PAT表总共分成了多少个段
…	

table_id = 0x00	表明这个段是属于PAT表的
…	
section_number = 0x02	表明这个段是该PAT表第三个段
last_section_number = 0x02	该PAT表总共分成了多少个段
…	


tableid==0 则
准备读取PAT
读取section_length
读下一个TS包
根据section_length，确定
该表存储的节目总数N
offset加上PAT表头8个字节
，偏移到N loop首字节
从当前偏移位置开始当前读
取节目
读取当前节目的
program_number和
program_map_pid
program_count++
offset +=4
program_count<N?
PAT表读取
结束

三、实验准备

- 文件操作
- 位运算
- 参考文档

1、文件操作

- 文件打开：

virtual BOOL Open(LPCTSTR *lpszFileName*, **UINT** *nOpenFlags*, **CFileException*** *pError* **= NULL);**

  - ***lpszFileName*** *需要打开文件的路径字符串，这个路径可以是相对路径也可以是绝对路径，或者是网络名字（**UNC**)*
  - ***nOpenFlags*** *一个**UINT**定义文件的存取共享模式。它指定文件打开时可以采取的操作。你可以使用”|“号来组合多个选项。文件的一个存取权限和一个共享选项是必须要指定的。*
  - 返回值：成功为非**0**，否则为**0**，仅当返回值为**0**时**pError**参数才有意义。

1、文件操作

- 例：

```
CFile cfile;
CString FileName；
if(!cfile.Open(FileName,CFile::modeRead,NULL))
 {
      cout<<"Open File Error!!"<<endl;
      return;
 }
```

1、文件操作

- 文件读取：

virtual UINT Read(void* *lpBuf*, **UINT** *nCount* **);**

- 例：

```
CFile cfile;
char pbuf[100];
cfile.Read( pbuf, 100 );
```

1、文件操作

- 获得文件长度：

  virtual ULONGLONG GetLength( ) const;

- 例：

  CFile cfile;

  long length=cfile.GetLength();

1、文件操作

- 指针指向位置：

  virtual ULONGLONG Seek(LONGLONG *lOff*, **UINT** *nFrom* **);**

  - **CFile::begin**
  - **CFile::current**
  - **CFile::end**

- 例：

  CFile cfile;

  cfile.Seek(len,CFile::begin);

2、位运算

- 47 44 00 3D
- 01000111 01000100 00000000 00111101

- Sync=pBuff[0];
- transport_error_indicator=pBuff[1]/128;
- payload_unit_start_indicator=(pBuff[1]<<1)/128;
- PID=(pBuffer[1]%32)*256+pBuffer[2];

3、MPEG标准中文.pdf

- P25　TS语法结构
- P47　PSI
- P49　PAT (PID=0)
- P52　PMT
- P66　stream_type
- P120　语法图

四、实验结果

- 完善程序内容；
- 输出以下参数：
  - **TS**包分组长度；
  - 节目套数及对应各节目**PMT PID**；
- 提交程序。