

GY

# 中华人民共和国广播电影电视行业标准

GY/T 201—2004

---

## 数字电视系统中的数据广播规范

Specification for data broadcasting in digital television system

2004-03-22 发布

2004-05-01 实施

---

国家广播电影电视总局 发布

# 目 次

前言 .....	II
1 范围 .....	1
2 规范性引用文件 .....	1
3 缩略语 .....	1
4 数据封装格式 .....	3
4.1 数据管道 .....	3
4.2 数据流 .....	3
4.3 多协议封装 .....	6
4.4 数据轮播 .....	10
4.5 对象轮播 .....	19
4.6 解码器模型 .....	23
5 数据广播相关业务的规定 .....	23
5.1 EPG 信息的发送方式 .....	23
5.2 基于中间件业务的发送方式 .....	24
6 数据广播中 CA 系统的规定 .....	24
7 回传信道以及双向业务的规定 .....	24
8 实施方法 .....	24
附录 A (规范性附录) 实施指南 .....	25
附录 B (规范性附录) 数据轮播的 DSM-CC 消息 .....	63
附录 C (规范性附录) 将 DSM-CC 消息封装到 GB/T 17975.1 段中 .....	68
附录 D (规范性附录) 目录中的对象命名 .....	70
附录 E (资料性附录) 文本字符编码 .....	72
附录 F (规范性附录) 私有数据广播系统注册 .....	76

## 前 言

本标准主要参照了EN 301 192《DVB specification for data broadcasting》和ETS 300 802《Network-independent protocols for DVB interactive services》，并结合了国际和国内有关数字电视数据广播的生产、开发、试验和应用的情况，从我国国情的实际出发编制的。

本标准的附录A、附录B、附录C、附录D和附录F为规范性附录，附录E为资料性附录。

本标准由全国广播电视标准化技术委员会归口。

本标准起草单位：国家广播电影电视总局数字（高清晰度）电视标准工作组、国家广播电影电视总局广播科学研究院、国家广播电影电视总局网络中心、中央电视台、北京歌华有线电视网络股份有限公司、北京海特荣德科技发展有限公司、北京蓝拓扑电子有限公司、中国华大集成电路设计中心、北京中视联数字系统有限公司、北京算通科技发展股份有限公司、北京首创系统集成技术有限公司、北京清华永新信息工程有限公司、华为技术有限公司、深圳傲龙宽频科技有限公司、深圳同洲电子股份有限公司、深圳经天通信股份有限公司、四川视达集团公司、浙江天屹网络科技股份有限公司。

本标准主要起草人：杨杰、崔竞飞、李熠星、曾庆军、梅剑平、符洪涛、朱佩江、胡毓铭、杨威、陈勇、秦葵龙、于松亮、田风武、马缚龙、姚建军、梅晓舟、谢朝阳、余志勇、许永红、李立、李飞、孙功宪、张基、傅雪鹏、李光伟、朱北川。

# 数字电视系统中的数据广播规范

## 1 范围

本标准规定了地面、有线、卫星等数字电视广播系统中基于传输流的数据广播的数据结构、数据交换、数据传输协议等。

本标准适用于地面、有线、卫星等数字电视广播系统中的数据广播系统。

## 2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版本均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

GB 2312 信息交换用汉字编码字符集基本集

GB/T 8565.1 信息技术 文本通信用编码字符集 拉丁字母

GB 13000.1 信息技术 通用多八位编码字符集(UCS) 第1部分：体系结构与基本多文种平面

GB/T 15273.1 信息处理 八位单字节编码图形字符集

GB/T 15629.1 信息技术 系统间远程通讯和信息交换局域网和城域网 特定要求 第1部分：局域网标准综述

GB/T 15629.2 信息技术 系统间远程通讯和信息交换局域网和城域网 特定要求 第2部分：逻辑链路控制

GB/T 17975.1 信息技术 运动图像及其伴音信息的通用编码 第1部分：系统

GB/T 17975.2 信息技术 运动图像及其伴音信号的通用编码 第2部分：视频

GB/T 17975.3 信息技术 运动图像及其伴音信号的通用编码 第3部分：音频

GY/Z 174—2001 数字电视广播业务信息规范

GY/Z 175—2001 数字电视广播条件接收系统规范

ISO/IEC 13818-6 信息技术 运动图像及其伴音信息的通用编码 第6部分：数字存储媒体命令和控制的扩展

ETS 300 802 Network-independent protocols for DVB interactive services

IETF RFC 1112 Host extensions for IP multicasting Common Object Request Broker Architecture and Specification OMG, Revision 2.0

IETF RFC 1521 MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies

IETF RFC 1590 Media Type Registration Procedure, (Updates RFC 1521), J. Postel.

IETF RFC 1950 ZLIB Compressed Data Format Specification version 3.3

IETF RFC 1951 DEFLATE Compressed Data Format Specification version 1.3

IETF RFC 2045 Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies

IETF RFC 2046 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types

## 3 缩略语

下列缩略语适用于本标准。

AFI	Authority and Format Identifier	授权和格式标识
API	Application Portability Interface	应用移植接口
BIOP	Broadcast Inter ORB Protocol	广播ORB间协议
bslbf	bit string, left bit first	比特串, 左位在先
CA	Conditional Access	条件接收
CDR	Common Data Representation	公用数据表示
CORBA	Common Object Request Broker Architecture	公共对象请求代理结构
CRC	Cyclic Redundancy Check	循环冗余校验
DC	Data Carousel	数据轮播
DDB	DownloadDataBlock message of DSM-CC	DSM-CC下载数据块消息
DII	DownloadInfoIndication message of DSM-CC	DSM-CC下载指示消息
DSI	DownloadServerInitiate message of DSM-CC	DSM-CC下载服务器初始化消息
DSM-CC	Digital Storage Media - Command & Control	数字存储媒体-命令与控制
DSM-CC U-N	DSM-CC User to Network	DSM-CC用户-网络模式
DSM-CC U-U	DSM-CC User to User	DSM-CC用户-用户模式
EIT	Event Information Table	事件信息表
EPG	Electronic Program Guide	电子节目指南
ES	Elementary Stream	基本流
GIF	Graphics Interchange Format	图形交换格式
HTML	HyperText Mark-up Language	超文本描述语言
IDL	Interface Definition Language	接口定义语言
IETF	Internet Engineering Task Force	互联网工程任务组
IIOP	Internet Inter ORB Protocol	互联网ORB间协议
IOR	Interoperable Object Reference	互操作对象基准
IP	Internet Protocol	互联网协议
JPEG	Joint Photographic Experts Group	联合图像专家组
LLC	Logical Link Control	逻辑链路控制
MAC	Media Access Control	媒体接入控制
MTU	Maximum Transport Unit	最大传输单元
NPT	Normal Play Time	正常播放时间
NSAP	Network Service Access Point	网络业务接入点
OC	Object Carousel	对象轮播
OMG	Object Management Group	对象管理组
OMT	Object Modelling Technique	对象建模技术
ORB	Object Request Broker	对象请求代理
OUI	Organizational Unique Identifier	组织机构统一标识
PAT	Program Association Table	节目关联表
PCR	Program Clock Reference	节目时钟参考
PES	Packetized Elementary Stream	打包的基本流
PID	Packet Identifier	包标识
PLL	Phase Locked Loop	锁相环
PMT	Program Map Table	节目映射表
PPP	Point to Point Protocol	点到点协议

PSI Program Specific Information 节目特定信息  
 PTS Presentation Time Stamp 展现时间戳  
 RFC Request For Comments 评议征求  
 rpchof remainder polynomial coefficient, higher order first 余数多项式系数, 高阶在前  
 SDT Service Description Table 业务描述表  
 SI Service Information 业务信息  
 SNAP SubNetwork Attachment Point 子网附属点  
 simsbf signed integer, most significant bit first 有符号整数, 高位在先  
 TCP Transfer Control Protocol 传输控制协议  
 TS Transport Stream 传输流  
 uimsbf unsigned integer, most significant bit first 无符号整数, 高位在先

## 4 数据封装格式

### 4.1 数据管道

数据管道规范支持数字电视系统中简单异步端到端的数据广播业务, 数据直接在TS包的载荷中传输, 见GB/T 17975. 1。

#### 4.1.1 数据传输规范

数据广播业务直接将要广播的数据插入TS包中。

数据广播业务可以业务特有的方式来使用TS包中的payload\_unit\_start\_indicator域和transport\_priority域。adaptation\_field的使用应符合GB/T 17975. 1。

按照数据管道进行数据传输时, 其数据格式是与业务相关的, 在本标准中不做规定。

#### 4.1.2 PSI 和 SI 规范

数据广播业务通过包含在SI中的一个或多个data\_broadcast\_descriptor来标明使用了数据管道。每个data\_broadcast\_descriptor通过component\_tag与一个具体的数据管道相关联。特别指出, component\_tag域的值应该与可能存在于用作数据管道的码流的PMT中stream\_identifier\_descriptor的component\_tag域的值相同。

##### 4.1.2.1 data\_broadcast\_descriptor

data\_broadcast\_descriptor按以下方式使用:

- data\_broadcast\_id: 此域置为0x0001, 标明一个数据管道, 见附录F;
- component\_tag: 此域的值应该与用作数据管道的码流的stream\_identifier\_descriptor的component\_tag域的值相同;
- selector\_length: 此域置为0;
- selector\_byte: 此域不使用。

##### 4.1.2.2 流类型

PMT中的stream\_type不在本标准中定义。

### 4.2 数据流

数据流规范支持数字电视系统中面向数据流, 端到端的, 异步、同步或从同步方式传输的数据广播业务, 数据在打包的基本流(PES)的包中传输, PES定义见GB/T 17975. 1。

异步数据流定义为只有数据、没有任何时序要求的流。

同步数据流定义为带时序要求的流。这种情况下, 接收端能够从同步数据流中再生数据和时钟。

从同步数据流定义为带时序要求的流。这种情况下, 流中的数据能够同步地其它的数据流(例如音频, 视频)一起回放。

#### 4.2.1 异步数据流

4.2.1.1 数据传输规范

数据广播业务的数据放在GB/T 17975.1定义 PES包中，该PES包的长度不能为零。GB/T 17975.1中规定了如何把PES包映射到TS包中。

异步数据流的规范使用标准的PES包的句法和语义，并有以下约束：

- stream\_id：此域置为0xBF(private\_stream\_2)；
- PES\_packet\_length：16比特域，数值非零。

4.2.1.2 PSI 和 SI 规范

数据广播业务通过包含在SI 中的一个或多个data\_broadcast\_descriptor标明异步数据流的使用。每个data\_broadcast\_descriptor通过component\_tag来与特定的流相关联。特别指出，如果用作数据流的码流的PMT中出现了stream\_identifier\_descriptor的component\_tag，则这两个component\_tag的值应该是相同的。

4.2.1.2.1 data\_broadcast\_descriptor

data\_broadcast\_descriptor按以下方式使用：

- data\_broadcast\_id：此域置为0x0002，标明异步数据流，见附录F；
- component\_tag：此域和流的stream\_identifier\_descriptor中的component\_tag值相同，数据在该流中传输；
- selector\_length：此域置为0；
- selector\_byte：此域不使用。

4.2.1.2.2 流类型

通过设定流类型为0x06或用户定义的数值，可以在一个业务的PMT中标明异步数据流。

4.2.2 同步或从同步数据流

4.2.2.1 数据传输规范

数据广播业务将要广播的数据放在GB/T 17975.1中定义 PES包内。该PES包应是非零长度的。把PES包映射到TS包的定义，见GB/T 17975.1。

同步和从同步数据流使用标准的PES包的句法和语义，并有以下约束：

- stream\_id：对于同步或从同步的数据流，此域置为0xBD(private\_stream\_1)；
- PES\_packet\_length：16比特的域，数值非零。

数据使用PES\_data\_packet结构插入PES包中。PES\_data\_packet结构的句法见表1。

表1 PES\_data\_packet 结构

句法	比特数	类型
PES_data_packet () {		
data_identifier	8	ui msbf
sub_stream_id	8	ui msbf
PTS_extension_flag	1	bsl bf
output_data_rate_flag	1	bsl bf
reserved	2	bsl bf
PES_data_packet_header_length	4	ui msbf
if (PTS_extension_flag== “ 1 ”) {		
reserved	7	bsl bf
PTS_extension	9	bsl bf
}		
if (output_data_rate_flag== “ 1 ”) {		
reserved	4	bsl bf

表1 PES\_data\_packet结构 (续)

句法	比特数	类型
output_data_rate	28	ui msbf
for (i=0; i<N1; i++) { PES_data_private_data_byte	8	bsl bf
for (i=0; i<N2; i++) { PES_data_byte	8	bsl bf
}		

PES\_data\_packet的语义如下：

- data\_identi fier：8比特域，定义了PES包中的数据类型。同一数据流中用于传送数据的PES包，data\_identi fier的值相同，编码见表2；

表2 data\_identi fier 域的编码

data_identi fier	意 义
0x00 to 0x0F	为将来使用保留
0x10 to 0x1F	保留
0x20	数字电视系统字幕
0x21	数字电视系统同步数据流
0x22	数字电视系统从同步数据流
0x23 to 0x7F	为将来使用保留
0x80 to 0xFF	用户定义

- sub\_stream\_id：8比特域，用户定义；
- PTS\_extension\_flag：1比特域。对于同步数据流，此域置为“1”。对于从同步数据流，此域置为“1”时表明在PES\_data\_packet中有PTS\_extension，此域置为“0”时表明没有PTS\_extension；
- output\_data\_rate\_flag：1比特域。对于从同步数据流，此域置为“0”。对于同步数据流，此域置为“1”时表明在PES\_data\_packet中有output\_data\_rate，此域置为“0”时表明没有output\_data\_rate；
- PES\_data\_packet\_header\_length：4比特域。标明包含 PES\_data\_private\_data\_byte的包头中的可选择域的长度；
- PTS\_extension：9比特域。此域扩展了PES包头中的PTS；此域含有9比特PCR值，将数据的PTS 时间分辨率从11.1us(90kHz)扩展到37ns(27MHz)；
- output\_data\_rate：28比特域，标明同步数据流再生信号的码率。输出码率被编码成28比特的正整数；
- PES\_data\_private\_data\_byte：此域与具体业务相关联。如果此域出现，标准的数字电视接收机将会跳过这些数据；
- PES\_data\_byte：此域传送要广播的数据。

#### 4.2.2.2 PSI 和 SI 规范

数据广播业务通过包含在SI中的一个或多个data\_broadcast\_descriptor来标明同步数据流或从同步数据流的使用。每个data\_broadcast\_descriptor通过component\_tag和一个特定的流相关。特别指出，



component\_tag的值应该与可能出现的，用作数据流的码流的PMT中的stream\_identifier\_descriptor的component\_tag的值相同。

4.2.2.2.1 data\_broadcast\_descriptor

data\_broadcast\_descriptor按以下的方式使用：

- data\_broadcast\_id：此域置为0x0003，标明同步数据流；置为0x0004，标明从同步数据流；见附录F；
- component\_tag：此域与传输数据的码流中stream\_identifier\_descriptor的component\_tag域值相同；
- selector\_length：此域置为零；
- selector\_byte：此域不使用。

4.2.2.2.2 流类型

通过在业务的PMT中将stream\_type置为0x06或用户定义的值来标明业务中存在同步数据流或从同步的数据流。

4.3 多协议封装

多协议封装规范支持数字电视系统中需要用通信协议中的数据报来传输的数据广播业务。

遵循多协议封装规范的数据报传输把数据报封装在DSM-CC段中，见ISO/IEC 13818-6，这和GB/T 17975.1的私有段格式是兼容的。

4.3.1 数据传输规范

数据报封装在与私有数据DSM-CC段兼容的datagram\_section中，见ISO/IEC 13818-6。GB/T 17975-1定义了如何将该段映射到TS流中。

datagram\_section的句法见表3。

表3 datagram\_section 的句法

句法	比特数	类型
datagram_section() {		
table_id	8	ui msbf
section_syntax_indicator	1	bsl bf
private_indicator	1	bsl bf
reserved	2	bsl bf
section_length	12	ui msbf
MAC_address_6	8	ui msbf
MAC_address_5	8	ui msbf
reserved	2	bsl bf
payload_scrambling_control	2	bsl bf
address_scrambling_control	2	bsl bf
LLC_SNAP_flag	1	bsl bf
current_next_indicator	1	bsl bf
section_number	8	ui msbf
last_section_number	8	ui msbf
MAC_address_4	8	ui msbf
MAC_address_3	8	ui msbf
MAC_address_2	8	ui msbf

表3 datagram\_section的句法（续）

句法	比特数	类型
MAC_address_1	8	ui msbf
if (LLC_SNAP_flag == “1”) { LLC_SNAP() }		
else { for (j=0; j<N1; j++) { IP_datagram_data_byte	8	bsl bf
}		
}		
if (section_number == last_section_number){ for (j=0; j<N2; j++) { stuffing_byte	8	bsl bf
}		
}		
if (section_syntax_indicator == “0”) { checksum	32	ui msbf
}		
else { CRC_32	32	rpchof
}		
}		

datagram\_section的语义如下：

- table\_id：8比特域，置为0x3E(私有数据的DSM-CC段，见ISO/IEC 13818-6)；
- section\_syntax\_index\_indicator：此域按照ISO/IEC 13818-6中定义的设置；
- private\_indicator：此域按照ISO/IEC 13818-6中的定义设置；
- reserved：2比特域，置为“11”；
- section\_length：此域按照ISO/IEC 13818-6中的定义设置；
- MAC\_address\_[1~6]：48比特域，内容为目标的MAC地址。MAC地址被分成6个8比特的域，从MAC\_address\_1到MAC\_address\_6分别标识。MAC\_address\_1包含MAC地址的最高字节，MAC\_address\_6域包含MAC地址的最低字节。图1显示了MAC地址在该段中各个域的映射。MAC\_address中包含了一个在address\_scrambling\_control域中指定的未加扰或加扰的MAC地址；

注：各个字节中的比特次序并没有反转，每个字节的最高位仍然最先传送。

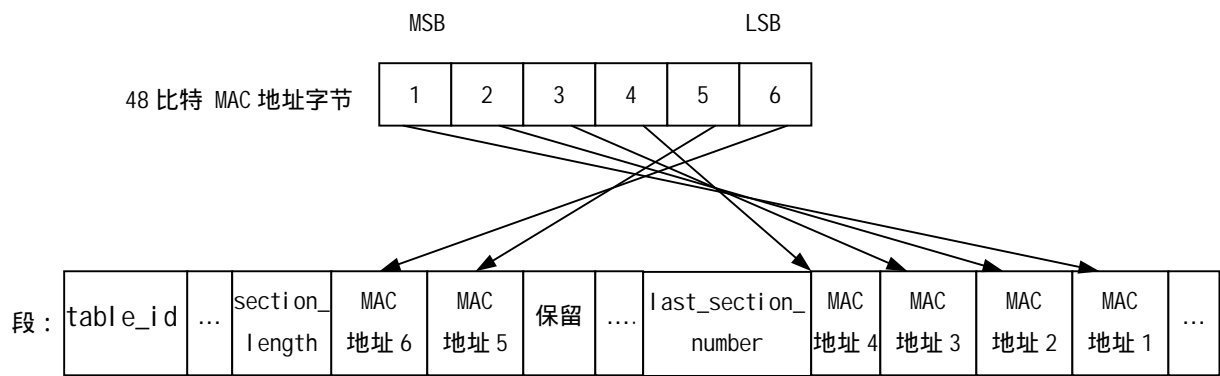


图1 MAC 地址字节到 section 的映射

- payload\_scrambling\_control :2比特域,标明该段载荷的加扰模式。包括了跟在MAC\_address\_1后面的载荷,但不包括校验和或CRC32域在内,见表4。所采用的加扰方法由用户定义;

表4 payload\_scrambling\_control 域的编码

值	Payload_scrambling_control
00	未加扰
01	业务定义
10	业务定义
11	业务定义

- address\_scrambling\_control :2比特域,标明MAC地址的加扰模式,见表5。该域使MAC地址可以动态变化。所采用的加扰方法由用户定义;

表5 address\_scrambling\_control 域的编码

值	Address_scrambling_control
00	未加扰
01	业务定义
10	业务定义
11	业务定义

- LLC\_SNAP\_flag : 1比特标志位。该标志位置为“1”,标明在载荷的MAC\_address\_1之后携带LLC/SNAP封装的数据报。LLC/SNAP结构应标明传送的数据报的类型。该标志位置为“0”,标明在载荷的MAC\_address\_1之后将包含一个没有用LLC/SNAP封装的IP数据报;
- current\_next\_indicator :1比特域,置为“1”;
- section\_number : 8比特域。如果数据报需要多个段传输,此域标明在分割过程中段的位置。否则置为“0”;
- last\_section\_number :8比特域,标明携带数据报的最后一段的序号,即分割过程的最后段的序号;
- LLC\_SNAP : 此结构包含LLC遵循GB/T 15629.2规范和SNAP遵循GB/T 15629.1规范的数据报。如果该段中的载荷被加扰(见payload\_scrambling\_mode),这些字节也将被加扰;

- IP\_datagram\_data\_byte：这些字节包含数据报的数据。如果该段中的载荷被加扰(见 payload\_scrambling\_mode), 这些字节也将被加扰；
- stuffing\_byte：可选的8比特域，值不指定。如果该段中的载荷被加扰(见 payload\_scrambling\_mode), 这些字节也将被加扰，以便协助宽总线环境中的分组加密和数据处理。使用的Stuffing\_byte数目应符合定义在data\_broadcast\_descriptor中的数据对齐需求；
- checksum：此域按ISO/IEC13818-6定义设置。它对整个datagram\_section进行运算；
- CRC\_32：此域按ISO/IEC13818-6定义设置。它对整个datagram\_section进行运算。

#### 4.3.2 PSI 和 SI 规范

数据广播业务通过包含在SI中的一个或多个data\_broadcast\_descriptor来标明数据报的传输。每个data\_broadcast\_identifier通过component\_tag和特定的流相关。特别指出，component\_tag值应与可能出现的、用作数据报传输的流的PMT的stream\_identifier\_descriptor中component\_tag的值相同。

##### 4.3.2.1 data\_broadcast\_identifier\_descriptor

data\_broadcast\_identifier\_descriptor按以下的方式使用：

- data\_broadcast\_id：此域置为0x0005，标明多协议封装，见附录F；
- component\_tag：此域与可能出现在用作数据广播的流的PMT的stream\_identifier\_descriptor中component\_tag的值相同；
- selector\_length：此域置为0x02；
- selector\_byte：此域用于传输表6定义的multiprotocol\_encapsulation\_info结构；

表6 multiprotocol\_encapsulation\_info 结构的句法

句法	比特数	类型
<pre> multiprotocol_encapsulation_info () {     MAC_address_range     MAC_IP_mapping_flag     alignment_indicator     reserved     max_sections_per_datagram } </pre>		
MAC_address_range	3	ui msbf
MAC_IP_mapping_flag	1	bsl bf
alignment_indicator	1	bsl bf
reserved	3	bsl bf
max_sections_per_datagram	8	ui msbf

multiprotocol\_encapsulation\_info结构的语义如下：

- MAC\_address\_range：3比特域，标明MAC地址字节数目，该数目依照表7来说明区分接收端时哪几位的MAC\_address字节是有效的；

表7 MAC\_address\_range 域的编码

MAC_address_range	有效的 MAC_address 字节
0x00	保留
0x01	6
0x02	6, 5
0x03	6, 5, 4
0x04	6, 5, 4, 3
0x05	6, 5, 4, 3, 2
0x06	6, 5, 4, 3, 2, 1
0x07	保留

- MAC\_IP\_mapping\_flag：1比特标志位。如果使用IETF RFC 1112中描述的从IP到MAC的映射，此标志位置为“1”。如果此标志位被置为“0”，从IP到MAC的映射在本标准范围之外；
- alignment\_indicator：1比特域。标明datagram\_section和TS流字节对齐，见表8；

表8 alignment\_indicator 域的编码

值	对齐的比特数
0	8（缺省值）
1	32

- reserved：3比特域，置为“111”；
- max\_section\_per\_datagram：8比特域，标明传输一个数据报单元的段的最大数目。

4.3.2.2 流类型

当一个业务中出现多协议数据流时，则把该流的stream\_type值在该业务的PMT中设置为0x0D或一个用户定义值。

4.4 数据轮播

数据轮播规范支持数字电视系统中周期性数据模块传输的数据广播业务。这些模块是已知大小的，并且可在数据轮播中及时更新、添加或移除。如果业务需要，这些模块还可以组成模块群，模块群也可以组成模块超群。

遵循数据轮播规范的数据广播在DSM-CC中定义的DSM-CC数据轮播中传输，而本标准定义了数字电视系统中使用的附加结构和描述符。因为用于准备离线内容的PID和时序参数没有明确基准，所以按照以上方式定义。

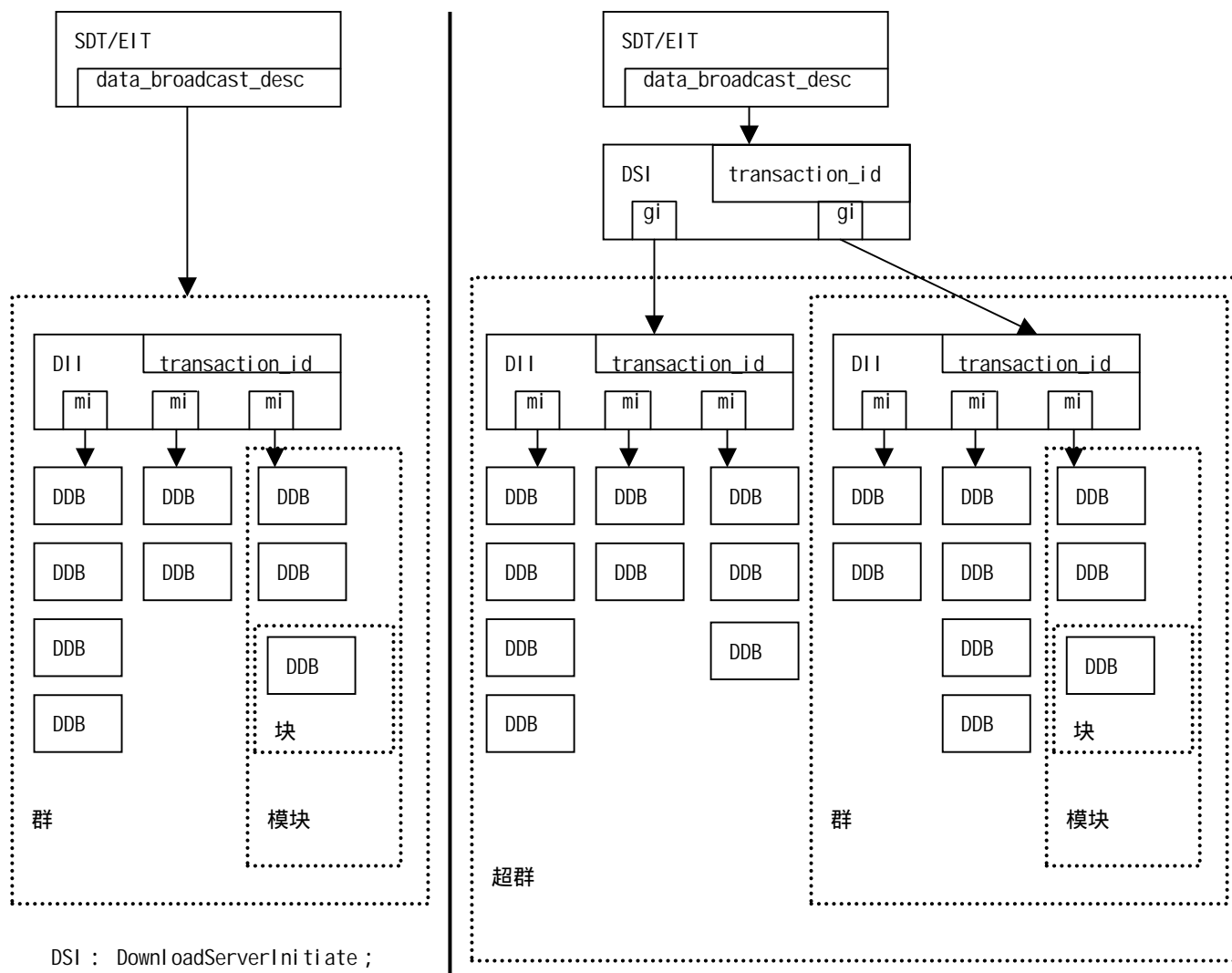
4.4.1 数据传输规范

数据轮播规范是基于DSM-CC数据轮播规范的，见ISO/IEC 13818-6。DSM-CC数据轮播规范实现了到接收端的数据循环传输。在数据轮播中传输的数据按照“模块”传输，模块由“块”组成。数据轮播中，除了每个模块中的最后一个块可能较小外，模块中的所有其它块都是相同大小的。模块是数据轮播中逻辑无关的数据组的描述。当业务需要时，模块可以组合成一个模块群。同样，群又可以组合成超群。

数据轮播规范用到了DSM-CC的下载规范中的四个消息。数据由DownloadDataBlock(DDB)消息携带，对模块的控制由DownloadInfoIndication、DownloadServerInitiate和DownloadCancel消息提供。DownloadServerInitiate描述了超群中的群，DownloadInfoIndication描述了一个群中的模块。基于这些控制消息，接收端可以从网络中获得模块的子集。这些消息的句法和语义见ISO/IEC 13818-6。本标准规定了这些消息在数据轮播中的使用。

## 4.4.1.1 数据轮播的结构

一层的数据轮播



DSI : DownloadServerInitiate ;

gi : GroupInfoBytes ;

DII : DownloadInfoIndication ;

mi : ModuleInfoBytes ;

DDB : DownloadDataBlock ;

—▶ : 位置参考 (transactionId, optional componentTag)。

图2 数据轮播的结构

数据轮播可以有一层或两层控制信息，如图2所示。数据轮播最简单的模式是一个数据轮播拥有一个控制层，该控制层描述了单个群。在这种情况下，SDT/EIT表中包含了指向DownloadInfoIndication消息的data\_broadcast\_descriptor。这个消息描述了数据轮播使用的ModuleInfoByte域中的模块。这个域包含了一个描述符的循环，描述符中可能包含了各种不同的信息，如指向DownloadDataBlock消息位置的指针。

当需要两层控制信息时，DownloadServerInitiate消息用于描述超群中的不同的群。DownloadInfoIndication消息的用途和单层数据轮播中的用途一样。

DownloadServerInitiate消息描述了拥有GroupInfoByte域的群，同时支持平台多样性。GroupInfoByte域同样由一个描述符的循环组成，描述符中可能包含了各种不同的信息。

解码器应能在两种轮播方式下都可以工作。业务提供商可以任意选择使用其中一种轮播方式。

群和模块可以通过专用PID与/或共享PID传送。如果没有提供明确的位置相关信息，位置由控制消息继承得到。图2中每个箭头表示了获得所指向的消息所需要的访问信息。

在数据轮播中这些信息由以下部分组成：

- component\_tag，指向业务中特定码流的指针；
- transaction/module identifier，控制消息或模块的独有的指示符。

接收端可以利用这些值从码流中有效地滤出消息。而且基于DSM-CC，也对说明模块和块大小的DownloadServerInitiate和DownloadInfoIndication消息参数进行了规定。

在本标准中，DSM-CC规定的compatibilityDescriptor()用法被限制为一个从DownloadServerInitiate到DownloadInfoIndication的前向查询机制。

DSI消息的compatibilityDescriptor位于GroupInfoIndication域中，称作GroupCompatibility()。

在一个超群（在两层数据轮播情况下）或一个群（在单层数据轮播情况下）中的所有DownloadDataBlock和DownloadInfoIndication消息拥有相同的downloadId。这意味着群可以共享模块，因为所有的moduleId在downloadId的范围内是唯一的。

每个控制消息都拥有一个transaction\_id作为该消息唯一的指示符。基于以下的语义，Transport\_id和module\_id可以用于有效地滤出数据轮播中的数据。

对于两层轮播：

- 对于DownloadServerInitiate消息，TransactionId的最低2个有效字节应该在0x0000-0x0001范围中；
- 对于DownloadInfoIndication消息，TransactionId的最低2个有效字节应该在0x0002-0xFFFF范围中。

对于单层轮播：

对于DownloadInfoIndication消息，TransactionId的最低2个有效字节应该在0x0000-0x0001范围中。

4.4.1.2 DownloadServerInitiate 消息

DownloadServerInitiate消息用于建立一个超群。语义如下：

- serverId：此域置为20个字节的0xFF；
- compatibilityDescriptor()：此域置为0x0000，此结构应该只包含DSM-CC定义的compatibilityDescriptor()中compatibilityDescriptorLength域；
- privateDataLength：此域定义了后面的GroupInfoIndication结构的字节长度；
- privateDataByte：此域用于传输表9中定义的GroupInfoIndication结构。

表9 GroupInfoIndication 结构

句法	字节数
GroupInfoIndication() {	
NumberOfGroups	2
for(i=0; i< numberOfGroups; i++) {	
GroupId	4
GroupSize	4
GroupCompatibility()	
GroupInfoLength	2
for(i=0; i<N; i++) {	

表9 GroupInfoIndication结构 (续)

句法	字节数
<pre> groupInfoByte     }     } privateDataLength for(i=0; i&lt; privateDataLength; i++) {     privateDataByte }     }     }</pre>	<p>1</p> <p>2</p> <p>1</p>

GroupInfoIndication结构的语义如下：

- NumberOfGroup：16比特域，标明了此域后面循环中群的个数；
- GroupId：32比特域，此域值等于描述该群的DownloadInfoIndication消息中的transactionId的值；
- GroupSize：32比特域，标明了群中所有模块的字节总长度；
- GroupCompatibility：此结构和DSM-CC中CompatibilityDescriptor结构相同；
- GroupInfoLength：16比特域，标明了后面的描述符循环的字节长度；
- GroupInfoByte：此域传输了描述符列表，每个描述符定义了一个或多个属性。在循环中的描述符说明了群的特性；
- PrivateDataLength：此域定义了后面的privateDataByte域的字节长度；
- PrivateDataByte：此域由用户定义。

#### 4.4.1.3 DownloadInfoIndication 消息

DownloadInfoIndication消息包含了群内模块的描述以及数据轮播的一些普通参数(如downloadId和blockSize)。每个模块由一些属性来描述，其中moduleId，moduleSize和moduleVersion被DSM-CC定义为DownloadInfoIndication消息中的域。模块的其它属性由后面定义的描述符传输。moduleId在0xFFFF0-0xFFFF范围内的值在本标准中不使用。

下面是数据轮播中DownloadInfoIndication消息的语义如下：

- compatibilityDescriptor()：此域置为0x0000，此结构只包含DSM-CC定义的compatibilityDescriptor()中的CompatibilityDescriptorLength域；
- moduleInfoLength：此域定义了被描述模块的moduleInfo域的字节长度；
- moduleInfoByte：此域传输描述符列表，当moduleId的值不在0xFFFF0-0xFFFF范围内时，每个描述符定义了一个或多个模块的属性。在这种情况下，moduleInfoByte结构包含了带有用作描述符循环结构的privateDataByte域的ModuleInfo结构；
- privateDataLength：此域定义了privateDataByte域的字节长度；
- privateDataByte：此域由用户定义。

#### 4.4.1.4 DownloadDataBlock 消息

DownloadDataBlock消息包含了被分割的模块中的块，按照DSM-CC规范通过TS包的载荷传输。

#### 4.4.1.5 DownloadCancel

DownloadCancel消息向接收端指示终止数据轮播模块的周期传输。DownloadCancel消息在群一级、或在超群一级，按照DSM-CC规范通过TS包的载荷传输：

- privateDataLength：此域定义了privateDataByte域的字节长度；
- privateDataByte：此域由用户定义。



4.4.2 描述符

4.4.2.1 描述符的标识和位置

表10包含数据轮播规范所定义的描述符。这些描述符有自己专用的descriptor\_tag ,这些描述符不能在本标准的数据轮播规范之外使用。

表10 数据轮播定义的描述符、值和允许的位置

描述符	Tag 值	DII-moduleInfo	DSI-groupInfo	简短描述
reserved	0x00			
type	0x01	+	+	数据的类型描述符
name	0x02	+	+	数据的名称描述符
info	0x03	+	+	文本描述
module_link	0x04	+		连接的数据模块
CRC32	0x05	+		CRC
location	0x06	+	+	数据位置
estimated_download_time	0x07	+	+	预计下载时间
group_link	0x08		+	连接描述群的DII消息
compressed_module	0x09	+		指示了压缩结构
注：“+”表示描述符允许出现的位置。				

数据轮播描述符的tag值的分配见表11。

表11 数据轮播描述符的 tag 值的分配

数据轮播中的私有描述符的tag	值
0x00 - 0x09	表10中目前分配的值
0x0A - 0x7F	为将来使用保留
0x80 - 0xFF	私有描述符

4.4.2.2 type\_descriptor

type\_descriptor包含作为一系列字符的模块或群的类型。表12给出了类型描述符的句法。

表12 type\_descriptor 的句法

句法	字节数	值
type_descriptor(){ descriptor_tag descriptor_length for (i=0; i<N; i++) { text_char } }	1 1 1	0x01  字符串

type\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于type\_descriptor此域置为0x01；
- descriptor\_length：8比特域，标明紧跟在此域后面的描述符内容的字节长度；
- text\_char：8比特域，一个字符串域说明了符合媒体类型规范IETF RFC 2045和IETF RFC 2046的模块和群的类型。

4.4.2.3 name\_descriptor

name\_descriptor包含了模块或群的名称。表13给出了name\_descriptor的句法。

表13 name\_descriptor 的句法

句法	字节数	值
name_descriptor(){ descriptor_tag descriptor_length for (i=0; i<N; i++) { text_char } }	1 1 1	0x02  模块的名称, 如 “index”

name\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于name\_descriptor此域设置为0x02；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- text\_char：8比特域，一个字符串域说明了模块和群的名称。文本信息的编码方式参见本标准的附录E。

#### 4.4.2.4 info\_descriptor

info\_descriptor包含了一个无格式文本描述。表14给出了info\_descriptor的句法。

表14 info\_descriptor 的句法

句法	字节数	值
info_descriptor(){ descriptor_tag descriptor_length ISO_639_language_code for (i=0; i<N; i++) { text_char } }	1 1 3 1	0x03  模块或群的描述

info\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于info\_descriptor此域设置为0x03；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- ISO\_639\_language\_code：此3字节域标识了后面的文本域使用的语言。ISO\_639\_language\_code包含了ISO 639-2描述的3字符编码。每个字符按GB/T15273.1的规定编码成8比特，并依次放入3字节域中。对于中文此域置为“chi”；
- text\_char：8比特域，字符串域说明了模块和群的名称。文本信息的编码方式参见本标准的附录E。

#### 4.4.2.5 module\_link\_descriptor

module\_link\_descriptor包含了将哪些模块从数据轮播中取出并连接成完整数据片的信息，同时告知解码器模块连接的顺序。表15给出了module\_link\_descriptor的句法。

表15 module\_link\_descriptor 的句法

句法	字节数	值
module_link_descriptor(){ descriptor_tag descriptor_length position Module_id }	1 1 1 2	0x04

module\_link\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于module\_link\_descriptor此域设置为0x04；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- position：8比特域，标明了模块在数据链中的位置。值为0x00表示模块是列表的第一个模块，值为0x01表示模块是列表中间的模块，值为0x02表示模块是列表的最后一个模块；
- Module\_id：16比特域，标明了列表中的下一个模块。对于列表中最后一个值，此域忽略。

4.4.2.6 CRC32\_descriptor

CRC32\_descriptor标明了整个模块的CRC32的计算结果。表16给出了CRC32\_descriptor的句法。

表16 CRC32\_descriptor 的句法

句法	字节数	值
CRC32_descriptor(){ descriptor_tag descriptor_length CRC_32 }	1 1 4	0x05

CRC32\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于CRC32\_descriptor此域设置为0x05；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- CRC\_32：32比特域，包含此模块计算出的CRC，计算方法见GB/T 17975.1的附录A。

4.4.2.7 location\_descriptor

location\_descriptor 包含了块、模块或群在轮播中相关数据所在的PID。表17给出了location\_descriptor的句法。

表17 location\_descriptor 的句法

句法	字节数	值
location_descriptor(){ descriptor_tag descriptor_length Location_tag }	1 1 1	0x06

location\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于location\_descriptor此域设置为0x06；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；

- location\_tag：8比特域，此域值与stream\_identifier\_descriptor(SI的描述符)中的component\_tag值相同。

#### 4.4.2.8 estimated\_download\_time\_descriptor

estimated\_download\_time\_descriptor包含了一个模块或群预计下载时间的整数值。表18给出了estimated\_download\_time\_descriptor的句法。

表18 estimated\_download\_time\_descriptor 的句法

句法	字节数	值
estimated_download_time_descriptor(){ descriptor_tag descriptor_length estimated_download_time }	1 1 4	0x07

estimated\_download\_time\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于estimated\_download\_time\_descriptor此域设置为0x07；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- est\_download\_time：32比特域，给出了以秒为单位的预计下载时间。

#### 4.4.2.9 group\_link\_descriptor

group\_link\_descriptor包含了连接起来的群描述，以描述一个更大的群的信息。当一个群的大小超过了单个DownloadInfoIndication消息的最大长度而必须展开放入多个这样的消息中时，此描述符是非常重要的，它同时告知解码器关于所连接的群的描述的顺序。当连接顺序并不重要时，此描述符不是严格需要的，它只是提供了一种方法来识别所有需要被连接的群描述。表19给出了group\_link\_descriptor的句法。

表19 group\_link\_descriptor 的句法

句法	字节数	值
group_link_descriptor(){ descriptor_tag descriptor_length position group_id }	1 1 1 4	0x08

group\_link\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于group\_link\_descriptor此域设置为0x08；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- position：8比特域，标明了群描述在数据链中的位置。值为0x00表示群描述是列表的一个群描述，值为0x01表示群描述是列表中间的群描述，值为0x02表示群描述是列表的最后一个群描述；
- group\_id：32比特域，表明了列表中的下一个群描述。对于列表中最后一个值，此域忽略。

#### 4.4.2.10 private\_descriptor

表20给出了private\_descriptor的句法。

表20 private\_descriptor 的句法

句法	字节数	值
private_descriptor () { descriptor_tag descriptor_length for (i=0; i<N; i++) { descriptor_byte } }	1 1 1	见表11

private\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于private\_descriptor此域如表11中所定义，设置范围为0x80～0xFF；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- descriptor\_byte：private\_descriptor的数据。

4.4.2.11 compressed\_module\_descriptor

compressed\_module\_descriptor的出现指示了模块中的数据使用IETF RFC 1950中定义的“zlib”结构。表21给出了compressed\_module\_descriptor的句法。

表21 compressed\_module\_descriptor 的句法

句法	字节数	值
compressed_module_descriptor(){ descriptor_tag descriptor_length compression_method original_size }	1 1 1 4	0x09

compressed\_module\_descriptor的语义如下：

- descriptor\_tag：8比特域，标识了描述符。对于compressed\_module\_descriptor此域设置为0x09；
- descriptor\_length：8比特域，标明了紧跟在此域后面的描述符内容的字节长度；
- compressed\_method：8比特域，标明了所用的压缩方法。见IETF RFC 1950中zlib的结构定义；
- original\_size：32比特域，标明了模块压缩前的字节长度。

4.4.3 PSI 和 SI 规范

数据广播业务通过包含在SI中的一个或多个data\_broadcast\_descriptor来标明使用了数据轮播。每个描述符应该通过component\_tag和特定的码流关联起来。component\_tag的值应该与可能出现的、用作数据轮播的码流的PMT中stream\_identifier\_descriptor中component\_tag的值相同。

4.4.3.1 data\_broadcast\_descriptor

data\_broadcast\_descriptor按以下方式使用：

- data\_broadcast\_id：此域置为0x0006，标明了数据轮播，见附录F；
- component\_tag：此域值和用于数据轮播的码流中的PMT中stream\_identifier\_descriptor中component\_tag的值相同；
- selector\_length：此域置为0x10；
- selector\_byte：用于传输表22中定义的数据\_carousel\_info结构。

表22 data\_carousel\_info 结构的句法

句法	比特数	类型
data_carousel_info () {		
carousel_type_id	2	bsl bf
reserved	6	bsl bf
transaction_id	32	ui msbf
time_out_value_DSI	32	ui msbf
time_out_value_DII	32	ui msbf
reserved	2	bsl bf
leak_rate	22	bsl bf
}		

data\_carousel\_info结构的语义如下：

- carousel\_type\_id：2比特域，标明了数据轮播的类型。此域采用表23所示的编码方式；

表23 carousel\_type\_id 的值

carousel_type_id的值	
00	保留
01	单层轮播
10	两层轮播
11	保留

- reserved：6比特域，置为“111111”；
- transaction\_id：32比特域，此域和最顶层的DownloadServerInitiate消息或DownloadInfoIndication消息中的transactionId的值相同。值0xFFFFFFFF用于向接收端表明相关码流上的任何收到的DownloadServerInitiate消息（两层轮播的情况）或DownloadInfoIndication消息（一层轮播的情况）都是有效的；
- time\_out\_value\_DSI：32比特域，标明了以ms为单位的推荐超时时限，接收端将它用于中断超时的DownloadServerInitiate的获取。值0xFFFFFFFF用于向接收端指出没有推荐的超时时限；
- time\_out\_value\_DII：32比特域，标明了以ms为单位的推荐超时时限，接收端将它用于中断超时的DownloadInfoIndication的获取。值0xFFFFFFFF用于向接收端指出没有推荐的超时时限；
- reserved：2比特域，置为“11”；
- leak\_rate：22比特域，标明了业务所用的解码器模型中的移出速率 $R_{X_n}$ 。移出速率编码成一个22比特正整数。leak\_rate的单位为50字节/秒。

#### 4.4.3.2 流类型

包含数据轮播的码流的PMT中该业务的stream\_type按GB/T 17975.1的规定设置为0x0B或用户定义的值，表明了业务中存在数据轮播。

#### 4.5 对象轮播

对象轮播规范用来支持数字电视系统中需要对DSM-CC U-U对象进行周期性广播的数据广播业务。依照对象轮播规范的数据广播参照DSM-CC中定义的DSM-CC对象轮播和DSM-CC数据轮播规范进行传输。

##### 4.5.1 对象轮播的范围

为了支持通过数字电视系统进行DSM-CC U-U对象的周期性广播的数据广播业务，加入了对象轮播的定义。

符合对象轮播规范的数据广播要遵循DSM-CC对象轮播和DSM-CC数据轮播规范进行传输。

##### 4.5.2 数据传输规范

对象轮播规范是基于DSM-CC对象轮播规范制定的。对象轮播代表了由网络中的DSM-CC U-U对象集合组成的一个特定业务域，该业务域有一个业务网关，为接收机提供业务图表和对象名。

广播网络中的业务网关的唯一识别是通过在DSM-CC中定义的NSAP地址来实现的。该地址包含了网络相关部分，使得在当前网络环境中地址唯一。轮播NSAP地址用来从其它业务域查询对象轮播，其句法和语义在下面定义。

4.5.2.1 轮播 NSAP 地址

轮播NSAP地址格式定义如图3所示。

AFI 1字节	Type 1字节	carousel Id 4字节	specifier 4字节	privateData 10字节
------------	-------------	--------------------	------------------	---------------------

图3 轮播 NSAP 地址格式

AFI、Type、carousel Id 和specifier的语义在DSM-CC中定义，补充如下：

- AFI：8比特域，置为0x00。标明NSAP格式用于用户定义；
- Type：8比特域，置为0x00。标明对象轮播使用NSAP地址；
- carousel Id：32比特域，设置为对象轮播的标识符，也就是carousel Id域；
- specifier：32比特域，按照DSM-CC规范传输specifierType域(值置为0x01)和OUI码。OUI码的值应遵循IEEE802注册机构为数字电视系统分配的值；
- privateData：此域传送表24定义的service\_location结构。

表24 service\_location 结构的句法

句法	比特数	类型
service_location() { transport_stream_id org_network_id service_id reserved }	16 16 16 32	ui msbf ui msbf ui msbf bsl bf

service\_location结构的语义如下：

- transport\_stream\_id：16比特域，标明轮播所在的TS流；
- org\_network\_id：16比特域，标明了生成轮播的传输系统的network\_id；
- service\_id：16比特域，标明了包含该对象轮播的业务标识符。此域与PMT中相关的program\_number相同；
- reserved：32比特域，保留。

4.5.3 PSI 和 SI 规范

数据广播业务通过包含在SI中的一个或多个data\_broadcast\_descriptor来标明使用了对象轮播。每个描述符指向一个对象轮播并且通过component\_tag来和特定的流相关联。特别指出，component\_tag的值应该和可能出现在用作数对象轮播的码流的PMT中stream\_identifier\_descriptor的component\_tag的值相同。通过使用对象名表，每个data\_broadcast\_descriptor可以启动一个基于语言判据的更高层协议。

对象轮播可以通过多个数据广播业务来实现。数据广播业务通过使用PMT第一个描述符循环中由DSM-CC定义的carousel\_identifier\_descriptor来标识它们是特定对象轮播的一部分。

此外，对象轮播使用Tap(见DSM-CC)来标识广播对象的码流。Tap和数据业务流之间的关联要么通过DSM-CC中定义的association\_tag来实现，要么通过SI中定义的stream\_identifier\_descriptor来实现。在后面这种情况中，若stream\_identifier\_descriptor的component\_tag值是作为基准的association\_tag最低有效字节的值，该association\_tag最高有效字节置为0x00。

最后，U-U对象轮播中的流对象可以与数据广播的ES流绑定，也可以与其它数字电视业务的ES流绑定，或者与完整的数字电视业务绑定。如果数据对象要与其它数字电视业务的ES流或完整的数字电视业务绑定，数据广播业务的PMT要在第一个描述符循环中包含deferred\_association\_tags\_descriptor。

deferred\_tags\_descriptor将在4.5.3.2中描述。

#### 4.5.3.1 data\_broadcast\_descriptor

data\_broadcast\_descriptor按以下的方式使用：

- data\_broadcast\_id：此域置为0x0007，标明对象轮播，见附录F；
- component\_tag：此域值等于用作对象轮播的流中可能出现的PMT中stream\_identifier\_descriptor的component\_tag的值；
- selector\_length：此域值为后面的selector域的字节长度；
- selector\_byte：用于传送表25定义的object\_carousel\_info结构。

表25 object\_carousel\_info 结构的句法

句法	比特数	类型
object_carousel_info () {		
carousel_type_id	2	bslbf
reserved	6	bslbf
transaction_id	32	ui msbf
time_out_value_DSI	32	ui msbf
time_out_value_DII	32	ui msbf
reserved	2	bslbf
leak_rate	22	ui msbf
for (l=0; l<N1; l++) {		
ISO_639_language_code	24	bslbf
object_name_length	8	ui msbf
for (j=0; j<N2; j++) {		
object_name_char	8	ui msbf
}		
}		
}		

object\_carousel\_info结构的语义如下：

- carousel\_type\_id：2比特域，置为“10”。标明两层的轮播；
- reserved：6比特域，置为“111111”；
- transaction\_id：32比特域，与携带业务网关对象基准的DownloadServerInitiate消息的transactionId值相同。置为“0xFFFFFFFF”向接收端指出从相关码流上接收到的任何DownloadServerInitiate消息都是有效的；
- time\_out\_value\_DSI：32比特域，标明了以ms为单位的推荐超时时限，接收端将它用于中断超时的DownloadServerInitiate的获取。值0xFFFFFFFF用于向接收端指出没有推荐的超时时限；
- time\_out\_value\_DII：32比特域，标明了以ms为单位的推荐超时时限，接收端将它用于中断超时的DownloadInfoIndication的获取。值0xFFFFFFFF用于向接收端指出没有推荐的超时时限；
- reserved：2比特域，置为“11”；
- leak\_rate：22比特域，标明了业务所用的解码器模型中的移出速率 $R_{Xn}$ 。移出速率编码成一个22比特正整数。leak\_rate的单位为50字节/秒；



- ISO\_639\_language\_code：24比特域，包含ISO\_639-2的3字符语言编码，用来选择启动更高层协议所需的对象；
- object\_name\_length：8比特域，标明了跟在object\_name\_length之后用来描述对象名称字符的字节长度；
- object\_name\_char：8比特域。object\_name\_char字符串指定了用来启动更高层协议的对象名。文本信息使用本标准的附录E描述的字符集来编码。

4.5.3.2 Deferred\_association\_tags\_descriptor

数字电视系统中deferred\_association\_tags\_descriptor()的句法见表26。

表26 deferred\_association\_tags\_descriptor 的句法

句法	比特数	类型
deferred_association_tags_descriptor(){ descriptor_tag descriptor_length association_tags_loop_length for(i=0; i<N1; i++){ association_tag } transport_stream_id program_number for(i=0; i<N2; i++){ private_data_byte } }	8 8 8 16 16 16 8	uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf

deferred\_association\_tags\_descriptor的语义如下：

- descriptor\_tag：8比特域，置为0x15；
- descriptor\_length：8比特域，标明描述符的字节长度；
- association\_tags\_loop\_length：8比特域，标明了此域后面的association\_tag循环的字节长度；
- association\_tag：16比特域，包含的association\_tag与非数据广播业务流关联，或者与另外的数字电视业务相关联；
- transport\_stream\_id：16比特域，标明了列出的association\_tag所关联的业务所在的TS流；
- program\_number：16比特域，应置为列出的association\_tag所关联的业务的service\_id；
- private\_data\_byte：此域包含了表27定义中的deferred\_service\_location结构。

表27 deferred\_service\_location 结构的句法

句法	比特数	类型
deferred_service_location(){ org_network_id for(i=0; i<N; i++){ private_data_byte } }	16 8	uimsbf uimsbf

deferred\_service\_location结构的语义如下：

- org\_network\_id：16比特域，标明了生成业务的传输系统的network\_id；
- private\_data\_byte：8比特域，在本标准中不做定义。

#### 4.5.3.3 流类型

把包含轮播的码流中PMT的stream\_type的值置为0x0B或用户定义的值。

#### 4.6 解码器模型

解码器模型对于数据流、多协议封装、数据轮播和对象轮播规范是通用的。

数据业务解码器模型是解码数据流的理论模型。解码器模型用来按时序规范比特流传输。解码器模型不规定实际解码器的实现方式，不排除不遵循本模型结构和时序的实现方式。

图4显示了单个数据流n的数据业务解码器的结构，它和在GB/T 17975.1中定义的传输系统目标解码器(T-STD)很相似。

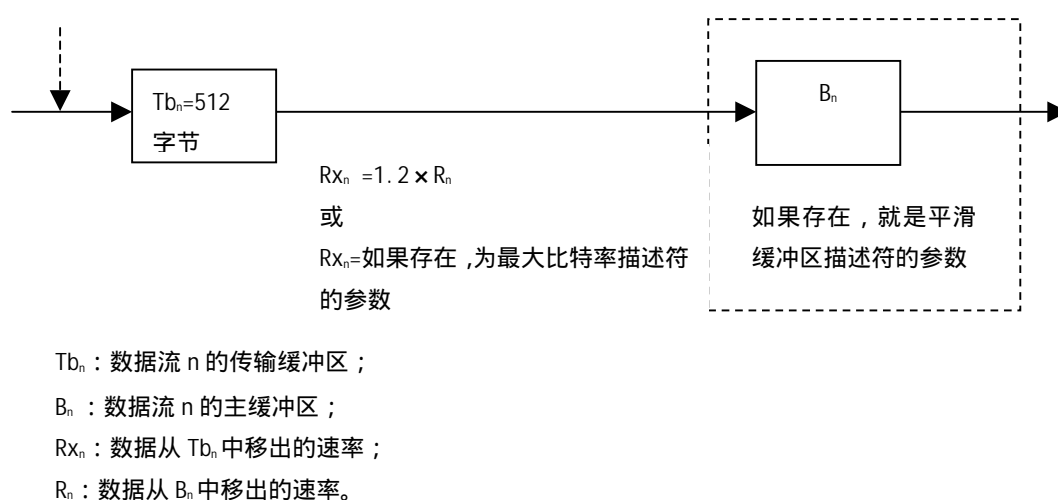


图4 数据业务解码器模型

将包含数据流n的数据的完整TS包送入流n的传输缓冲区 $Tb_n$ 中。所有的进入 $Tb_n$ 的字节以下面规定的速率 $RX_n$ 移出。将构成PES包、段或者其中内容的字节传送到主缓冲区 $B_n$ 中。其它字节不送入 $B_n$ ，可能用于控制系统。复制的TS包不传送到 $B_n$ 中。所有的进入 $B_n$ 中的字节以下面规定的速率 $R_n$ 移出。

传输缓冲区 $Tb_n$ 为512字节。

对一个特定的业务，传输缓冲区的移出速率 $RX_n$ 、缓冲区的大小 $B_n$ 和移出速率 $R_n$ 是确定的。该业务可以通过maximum\_bit\_rate\_descriptor和smoothing\_buffer\_descriptor来标明 $RX_n$ 、 $B_n$ 和 $R_n$ 的值，见GB/T 17975.1。如果使用这些描述符，它们应包含在业务的SDT或EIT以及PMT之中。

maximum\_bit\_rate\_descriptor的maximum\_bit\_rate域标明 $RX_n$ 的值。

smoothing\_buffer\_descriptor的sb\_size域包含 $B_n$ 的值。sb\_leak\_rate域包含 $R_n$ 的值。

若SI和PSI中不包含maximum\_bit\_rate\_descriptor而包含smoothing\_buffer\_descriptor，那么 $RX_n = 1.2R_n$ 。

若SI和PSI中不包含smoothing\_buffer\_descriptor，而包含maximum\_bit\_rate\_descriptor，那么双缓冲区模型就变成了移出速率为 $RX_n$ 、容量为 $Tb_n$ 的单缓冲区模型。

若SI和PSI中不包含任何一个描述符，那么缓冲区模型不适用。在这种情况下，比特流的传送是业务特定的。

## 5 数据广播相关业务的规定

### 5.1 EPG 信息的发送方式

对于需要使用数据广播通道来传输EPG业务内容的情况，使用本标准规定的轮播规范。

对于EPG系统的具体数据传输方式及数据交换协议，本标准不做规定。

## 5.2 基于中间件的业务的发送方式

对于基于中间件的业务，其业务有关的传输协议、格式等，本标准不做规定。

## 6 数据广播中 CA 系统的规定

关于CA系统的构成机制与工作方式，见GY/Z 175 - 2001。

在数字电视系统中实施数据广播时，需要对所传输的数据进行加密管理时，采用通用加扰算法对TS流中的载荷进行加扰。

## 7 回传信道以及双向业务的规定

本标准现阶段不做规定。

## 8 实施方法

本标准的具体实施方法见附录A。

附 录 A  
( 规范性附录 )  
实施指南

A.1 操作规则

本章为本标准应用建议。本附录主要参照了TR 101 202《DVB Implementation Guidelines for Data Broadcasting》。

在实际操作中，建议根据表A. 1的对应关系来选择数据广播应用所采用的数据封装。

表 A.1 各种应用数据对应的数据封装

应用数据	数据管道	数据流			多协议封装	数据轮播	对象轮播
		异步	同步	从同步			
实时动态数据流							
时序相关数据							
IP 数据报							
EPG							
静态数据文件							

表中的每行代表一类应用数据，每列代表一种封装类型。在行列的交叉点如果标有“ ”，说明本标准建议对该应用数据类型使用相应的数据封装格式；如果为空白，则表示本标准不建议对该应用数据类型使用相应的数据封装格式。

A.2 简介

图A. 1显示了数据广播标准的概况。

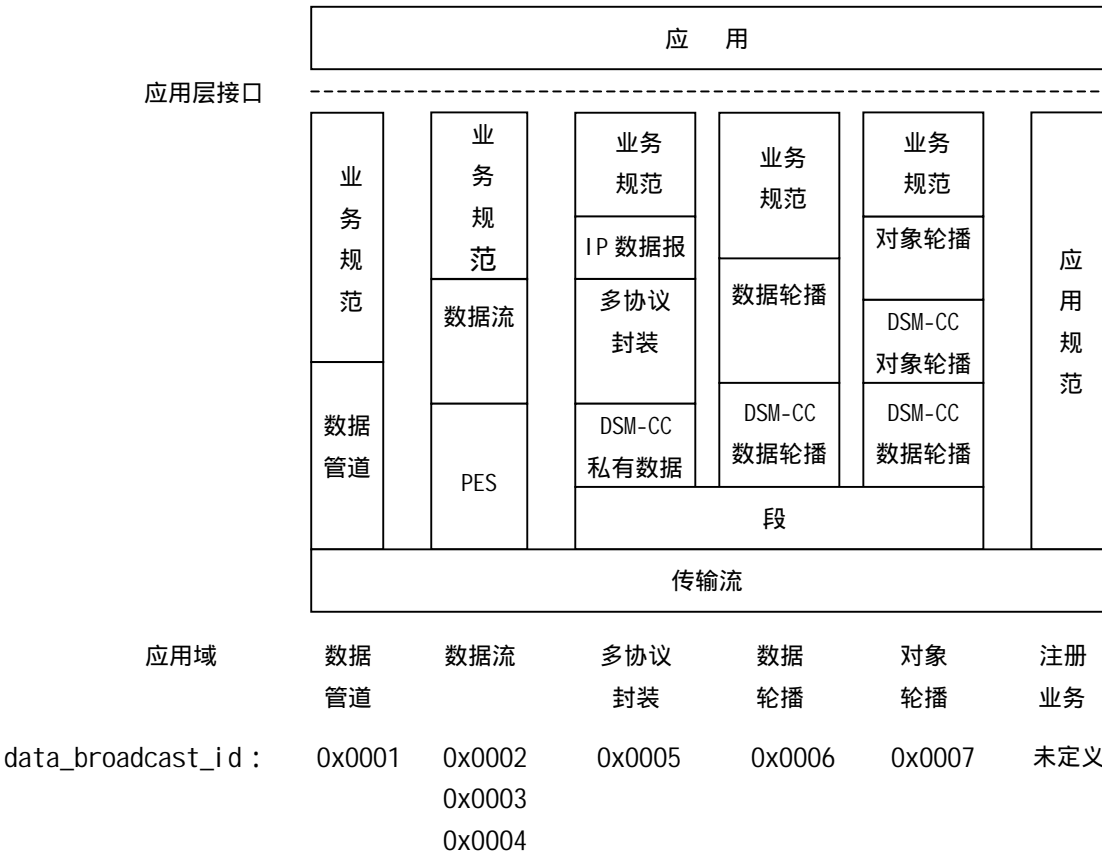


图 A. 1 与其它标准关系的图例

TS流是本标准的基础，依据应用域，数据信息在TS流中传送，这些应用域有：

- 数据管道；
- 数据流；
- 多协议封装；
- 数据轮播；
- 对象轮播。

图A. 1的右边所示的是注册业务，数字电视广播允许自定义方法实现数据广播业务。

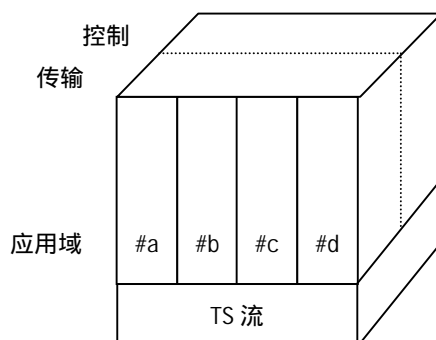
图A. 1显示各部分标准的内容，TS流的标准见GB/T 17975.1，DSM-CC 结构框架见ISO/IEC 13818-6，互连网络协议(IP)见IETF RFC 791，本标准定义了数字电视广播中的数据管道、数据流、多协议封装、数据轮播和对象轮播，图A. 1封装的IP协议只是一个例子，其它协议也可以被封装进来。

如图A. 1所示，本标准规定了所有应用域的不同业务层。数据管道规范对如何从TS流中提取数据没有给出太多信息，而只是规定怎样将数据打成TS流的包。与其它应用域相比，许多业务的运行需要专用的软件和硬件。

数据流规范具有更多功能性规范，尤其是在时序方面。数据广播可以是异步的、同步的或从同步的。本数据流规范是基于GB/T 17975.1定义的PES包。

多协议封装、数据轮播和对象轮播的应用规范都是建立在DSM-CC框架结构之上的，基于GB/T 17975.1中定义的私有段。本标准增加了使框架结构在数字电视广播环境中工作的特定信息，尤其是与GY/Z 174—2001相关联的部分。

在本标准中，每个应用分两部分定义，见图A. 2。



控制：SI 和 PSI；

传输：数据广播传输规范。

图 A.2 传输和控制规范部分

控制规范是GY/Z 174—2001的一部分，传输规范是本标准的一部分。以下子条款给出在不同应用域的实施建议。

### A.3 选择合适的类

如图A.1所示，可以有几种不同方式通过数字电视广播TS流来传输数据。其机制根据不同应用有不同的特征参数，如过滤器、开销、大小等。要根据目标应用的特定要求来选择合适的机制。

对于不同的类，规范的详细程度不同。例如，多协议封装和数据轮播规定得比较详细，几乎不需要应用相关说明，而其它类则自由得多。

另外，应用可以采用推荐以外的类。例如，类似数据轮播的应用也可以基于数据管道实现，IP广播也可以基于数据流实现。当然这种安排是应用选择的一部分。

#### 数据报的分割

任何协议的数据通常都是以包(数据报)的形式传输的，这些数据报的长度可能不同。如果数据没有被打包，或者打包的方法与数字电视广播传输链路无关或隐藏在其中，那么最好的传输方式是数据管道。

在GB/T 17975.1传输层，数据以固定长度188字节(184字节的载荷)的包长来传输的。因此，更上层的数据报在发送端被分割，在接收端重组。数据报的分割有三种可能方式，见图A.1：

- 基于数据管道的自定义机制；
- 打包的基本流(PES)；
- 段。

PES流提供一种传输机制，可传输最大64K字节的可变长度数据报，另外还提供不同数据流之间的准确同步(例如用于视音频同步)。因此数字电视广播用其来传输同步、从同步和异步的数据流。

段可用于传输最大为4K字节的可变长度数据报，这种传输是异步的。段可以通过解复用器利用硬件滤出，从而降低对接收机软件处理能力的要求。这就是选择段来传输封装的协议和数据轮播的主要原因。

对于数字电视广播框架中的数据广播业务，data\_broadcast\_id\_descriptor可能出现在PMT表中，即这一描述符的使用是可选的，见GY/Z 174—2001。

### A.4 数据管道

数据管道是一种异步传输机制，它将数据直接插入到TS包的载荷中。

数据报的分割与重组机制没有定义，如果需要由应用定义。例如，可以用payload\_unit\_start\_indicator来表示数据报的开始，transport\_priority表示数据报的结束。

continuity\_counter的使用方法见GB/T 17975.1的2.4.3。

### 适应域的使用

适应域的主要作用是填充，也可以有其它用途，如：传输PCR。

## A.5 异步/从同步/同步数据流

### A.5.1 适应域的使用

根据GB/T 17975.1的2.4.3，每个PES包总是开始于TS包载荷中的第一个字节。这就意味着与TS包没有对齐的PES包需要填充。对于段GB/T 17975.1允许将填充字节填充在包的尾部，而对于PES填充只能使用适应域来完成。由于大多数解复用器都能够自动将适应域分离出来，对解码器的性能也没有限制，也不需要额外的处理能力。

当把所有适应域标志(discontinuity\_indicator、random\_access\_indicator、elementary\_stream\_priority\_indicator、PCR\_flag、OPCR\_flag、splicing\_point\_flag、transport\_private\_data\_flag、adaptation\_field\_extension\_flag)设置为0，并且插入所需数量的填充字节时，适应域只用来填充。

Elementary\_stream\_priority\_indicator和adaptation\_field\_extension\_flag应该设为0，因其相关特征对数据流都没意义。

### A.5.2 异步数据流

在需要异步传输数据报的应用中，PES机制具有优势的情况下，就利用异步数据流。既然这种传输不需要同步，就选择stream\_id“private\_stream\_2”，它表明不使用PES包头，见GB/T 17975.1。这样，数据报紧跟在PES\_packet\_length之后。

数据报的定义是自定义实现的部分，因此本附录不作规定。

### A.5.3 同步和从同步数据流

为了满足同步和从同步数据流的需要，本附录的应用类中定义了一个附加头。

stream\_id应该设为“private\_stream\_1”，允许使用PES包头，特别是PTS。使用时间标志需要访问单元的定义。这些都是与应用相关的，因此在数据广播标准中没有定义。

这个头中的第一个字节包含data\_identifier，指明数据流类型(同步/从同步)。

在同一个PES包中不允许既有同步的数据流，又有从同步的数据流。但是，两种流数据如有不同的PID它们就可以存在于同一节目中。

sub\_stream\_id可自定义。

PTS\_extension\_flag和output\_data\_rate\_flag两个标志指示输出数据速率域和PTS扩展域的存在。下面列出了这些域的用法：

- PES\_data\_packet\_header\_length指示头长度，并允许在头部加入私有字节；
- PES头中的DTS域没有使用，而PTS的编码与GB/T 17975.1定义的方法相同。

### A.5.4 同步数据流

在接收端要求输出数据速率非常准确时，就需要使用同步数据流。接收端时钟由PCR同步，9比特的PTS\_extension要在较大数据速率范围内(从kbits/s到Mbps/s)非常准确地定位数据访问单元。根据定义，访问单元可以是一个比特、一个或多个字节。这9比特使PTS时钟的精度从11 μs扩展到37ns(27MHz时钟的精度)。如果要求接收同样数据业务的多个数据解码器同时以对齐的方式输出数据，或者要求输出流在短暂信号丢失后保持同步，那么就需要准确的数据定位。

为了规定同步数据流的输出数据率，使用output\_data\_rate。用28比特的精度(取代PES头中22比特ES\_rate的400bits/s分辨率)，可实现与高达27MHz( $\pm 30 \times 10^{-6}$ )的数据输出率成比例的PLL(利用时钟下转换)，覆盖很宽的数据率。output\_data\_rate传送同步数据流再生信号的比特率。将数据流的比特率编码成output\_data\_rate与应用有关。需要比特率是1bit/s倍数的应用，以bit/s为output\_data\_rate的单位，直接将数据流的比特率编码成output\_data\_rate。在GB/T 17975.1对27MHz

system\_clock\_frequency规定的 $30 \times 10^{-6}$ 误差范围内,如果应用需要连续变化的比特率,要将数据流的比特率编码成output\_data\_rate的方式,见式(1):

$$\text{output\_data\_rate} = \text{比特率} \times M / \text{system\_clock\_frequency} \dots \dots \dots (1)$$

式(1)中,M是一个足够大的数,应能够表示应用所需比特率精度要求的比特率范围。具有27MHz system\_clock\_frequency的同步数据流的实际比特率范围是从1bit/s到27Mbit/s。

注:如果使用output\_data\_rate,不必使用本标准中描述的解码器模型。

对于400bit/s ES\_rate精确度足够的应用,在PES data\_packet中没有output\_data\_rate的情况下,可以使用PES头中的ES\_rate。如果在一个编码流中同时出现ES\_rate和output\_data\_rate,那么解码器可以使用其中的任何一个比特率。

从同步数据流的建议缓冲区容量为4800字节。对于4ms典型最大多路复用抖动和高达9Mbit/s的比特率,这提供了足够的容量。

#### A.5.5 从同步数据流

当数据流要与另一个PES流同步时,使用从同步数据流。

### A.6 多协议封装

#### A.6.1 概述

多协议封装为在数字电视广播网络中的TS流之上传送数据网络协议提供了一个机制。多协议封装对传送IP协议已进行优化,但使用LLC/SNAP封装可以传送其它任意的网络协议,包括单播(目标是单个接收端的数据报)、多播(目标是多个接收端的数据报)和广播(目标是全部接收端的数据报)。48比特的MAC地址用于对接收端寻址,但不规定如何分配给接收端MAC地址。

由于数字电视广播网络的广播特性,数据安全是非常重要的。通过支持包加密和动态改变MAC地址,封装保证了数据的安全传输。

#### A.6.2 数据传输

在datagram\_sections中传送数据报,与私有数据的DSMCC\_section格式兼容。段格式提供了数据报到TS流包映射的一种有效格式,支持使用已有硬件或软件解复用器进行基于MAC地址的数据报过滤。

段格式允许将数据报分成多个段。如果数据报小于或等于4080字节(包括可能的LLC/SNAP报头),数据报放在一个段里。在IP和LLC/SNAP报头忽略的情况下,MTU应置为4080字节或更少,以便数据报不被分开。在IP和LLC/SNAP报头存在的情况下,MTU应置为4074字节或更少。

MAC地址为6个字节分成两组。其原因是字节5和6在DSMCC\_section的table\_id\_extension中,而字节1、2、3、4在DSMCC\_section的载荷中,见图A.3。

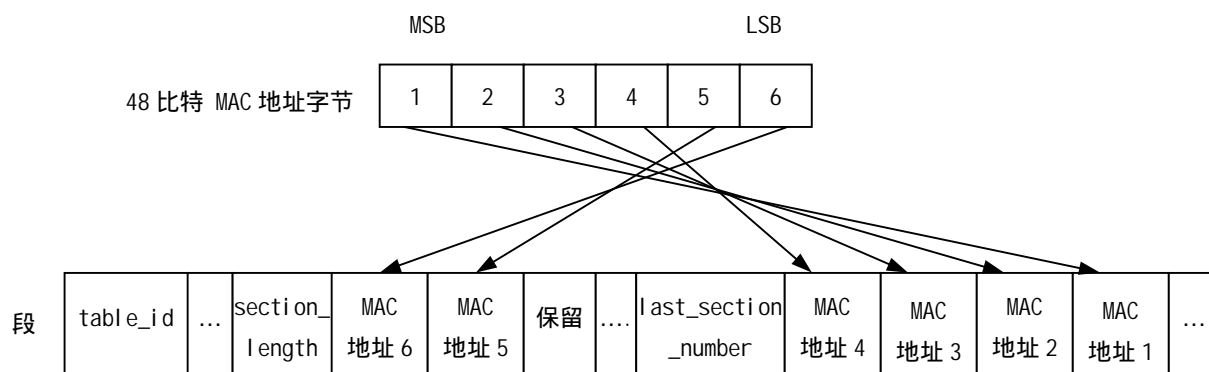


图 A.3 MAC 地址字节到段的映射

一些解复用器能用硬件过滤字节5和6,而用软件过滤字节1、2、3、4。建议将MAC地址的最有可能区分接收端的两个字节放在字节5、6。这和通常情况的IEEE MAC地址一致,建议所有的MAC地址都应以此方式构造。



载荷的加扰是由2比特payload\_scrambling\_control控制。如果它的值是“00”，载荷不加扰。如果载荷加扰，加扰算法和在数字电视广播中加扰方法如何传送给接收机的机制由国家CA标准GY/Z 175—2001定义。

通过动态变化MAC地址，MAC地址加扰提供更高的安全性。通过周期性地改变用于加扰MAC地址的控制字，特定的数据报的目的地不能通过观察MAC地址而确定，能够防止流被监视。由于很难得到送到单一接收端的数据报，因此加强了安全性。数字电视广播不规定加扰MAC地址控制字的分发机制。

在段头中地址加扰是由2比特域address\_scrambling\_control控制。如果它的值是“00”，则不对MAC地址加扰。应注意由于作为数据报一部分的协议地址在透明传输中是可见的，因此没有载荷加扰的MAC地址加扰是没用的。

LLC/SNAP封装提供了能够用于传送任何网络协议的多封装，包括IP协议。对传送IP进行了优化，允许传输没有LLC/SNAP报头的IP数据报，由段头中的LLC\_SNAP\_flag控制。如果它的值是“0”，则载荷包含原始IP数据报。如果它的值是“1”，则载荷包含一个LLC/SNAP IP封装，该封装由LLC/SNAP的结构LLC\_SNAP()组成，随后是数据报字节。传送IP的优化方法能用于IPv4和IPv6。当传送IP协议时，section\_number和last\_section\_number都应是“0”。

段可以包含数据报之后的填充。例如，当使用分组加密编码时，填充字节可以使段的载荷是分组长度的倍数。对这些字节的值不做规定，在载荷加密的情况，不要分配给固定值，固定值将有助于破密。

根据section\_syntax\_indicator的值，datagram\_section的结尾处有和校验或CRC\_32。建议使用CRC\_32，因为它提供更好一点的比特误码保护，大多硬件解复用器能够对CRC\_32进行硬件校验，而和校验通常应用软件进行校验。

#### A.6.3 在SI中的信息

对于使用多协议封装的业务，在SDT或EIT中有data\_broadcast\_descriptor。在selector\_byte中传送multiprotocol\_encapsulation\_info结构。

MAC\_address\_range用于通知接收端MAC\_address的哪个字节对过滤是有效的。

MAC地址的有效字节在MAC地址低位端。

如果依照IETF RFC 1112将多播IP地址映射为MAC地址，则发出MAC\_IP\_mapping\_flag。

值得注意的是，数字电视广播并未规定使用IEEE定义的MAC地址，因此允许可能更优化的映射。

对齐指示符指示datagram\_section与TS流包是8比特对齐排列还是32比特对齐。8比特对齐基本意味着不对齐。对齐在输入TS流包的实现中是有用的，并且依靠段的开始，在该段上实现32比特范围内的有效过滤比较操作。

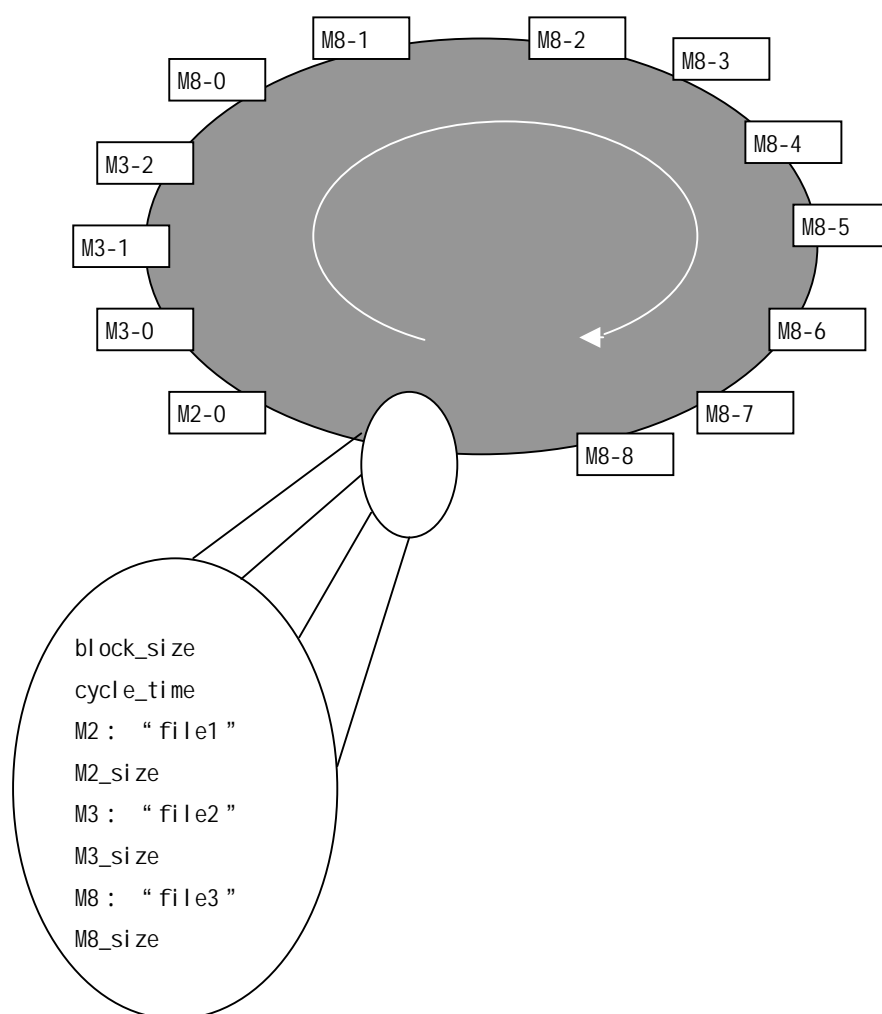
max\_sections\_per\_datagram定义用于传送单个数据报的最大段数(对IP来说，限定为1)。这定义了数据报的最大长度。典型的接收端在数据报传出去之前应将分割的数据报组合起来，因此该域定义了缓冲区的容量。缓冲区的容量是接收端为组合最大长度数据报所必需的。

### A.7 数据轮播

#### A.7.1 概述

数据轮播是一种传输机制，它允许服务器(应用的广播组件)通过一次或多次循环重复轮播的内容，为解码器(接收端运行的程序)提供一套清楚的数据模块。如果应用解码器要从数据轮播接入一个特定的模块，它可以只是等待所要求模块数据的下一次广播时间。

一个被广泛理解的数据轮播概念的好例子是图文电视系统。在这个系统中，一整套的图文页以在场逆程的模拟视频信号的行中循环广播。当用户想看一页信息时，他常常需要等到该页的下一次广播时间。用户所需要等待的最大时间取决于进行一次完整轮播循环所占用的时间，该时间可由轮播的大小和数据广播的速率来推导。



下载数据消息 (MX-Y) :

DownloadDataBlock ()

X = module\_id

Y = block\_number

下载控制消息 :

DownloadServerInitiate () 或

DownloadInfoIndication ()

图 A. 4 数据轮播中信息的循环发送

在数据轮播中,数据由模块构成,如在图A. 4中描述的M2、 M3和M8。这可能就是一些文件内容,比如这个例子中的“文件1”、“文件2”和“文件3”。每个模块都被分割,用来构成用DSM-CC DownloadDataBlock句法定义的一个或多个下载数据消息的载荷。下载控制消息提供描述每个模块和逻辑分组的信息,在适合的情况下定义使用DSM-CC的DownloadServerInitiate或DownloadInfoIndication。

在这个例子中每个下载消息都仅插入一次,并且同一模块的DownloadDataBlock依序相邻插入。然而,对一个特定消息插入到轮播的一个循环的频度以及消息插入的次序和位置没有作约束。这允许数据轮播可以使用任何最适合一个特定应用的方式来创建。此外,消息插入数据轮播的频度和次序不需固定并且根据需要可以动态改变。

#### A. 7. 2 数据轮播群和超群

一组数据轮播中的逻辑上一致的模块可以组合成群。DownloadInfoIndication消息提供群中模块的描述。这里没有对模块如何关联形成群作限定，特别的，一个模块可以属于多个群。

群可以组合成超群。DownloadServerInitiate消息提供超群中群的描述。

注：超群可以包含任意数目的群，甚至是只有一个群。

数据轮播的结构(在群或超群中)没有必要反映内容的结构。

为了清楚阐述，具体的DSM-CC消息在本标准的附录B中描述。本标准的附录C给出了关于在段中包含DSM-CC消息的信息。

#### A. 7. 3 使用一层数据轮播

如果数据轮播仅由单个群组成并且群的完整描述可以包含在单个的DownloadInfoIndication消息中(也就是控制信息的一层)，那么可以使用一层数据轮播。在其它所有情况中，应使用两层数据轮播。

DownloadInfoIndication消息是数据轮播中唯一的下载控制消息并且作为最高级别的控制消息来使用的。

注：尽管在数据轮播中只定义了一个下载控制消息，但是在该数据轮播的单一循环中，相同消息可能多次插入。

一个单层轮播适用的例子是一个小的(10~20模块)基于HTML应用的传输，该应用仅支持HTML V1.0版本。

#### A. 7. 4 使用两层数据轮播

两层数据轮播有一个或多个DownloadInfoIndication消息并且只有单个DownloadServerInitiate消息(也就是控制信息的两层)。DownloadServerInitiate消息是被作为最高级别的控制消息来使用的。

两层数据轮播在下面描述的情况下使用。这是用于具体情况的指南，并且有必要时可以混合使用。

##### A. 7. 4. 1 数据轮播由单个群组成，且该群的描述对单个的DownloadInfoIndication消息来说太大

在这种情况下，如果需要，就要有足够多的DownloadInfoIndication消息被用来描述在这个大的群中所有的模块。这样有效地将这个大的群分割成许多小群，这些小群每个都是由它自己的DownloadInfoIndication消息来定义的。既然只能拥有一个最高级别控制消息，这就强制地规定了两层数据轮播的使用。为了能够重建原始的大的群，这些新的较小的群应该连接在一起。这是通过在DownloadServerInitiate消息中的每个较小的群的描述中包含group\_link\_descriptor()来完成的。

例如基于只支持HTML V1.0版本的大的HTML应用(500个模块以上)的传输。

##### A. 7. 4. 2 数据轮播只传输单个版本的应用，但支持许多不同的接收端类型

在这种情况下数据轮播由通过在多个群中共享通用模块的方式来支持不同的接收端类型的群构成。这样的例子就是一个基于支持HTML V1.0、V2.0和V3.0 的小的HTML应用的传输。该数据轮播将由一个包含三个群的超群构成。多数的模块对所有的三个群都是通用的，例如GIF和JPEG，不过一些模块对于某个群来说是特定的，例如一个页面的特定HTML版本。

每个群关联的groupCompatibility结构用来决定一个给定的接收端类型使用哪个群。

##### A. 7. 4. 3 单一的接收端类型数据轮播同时传输一个应用的多个版本

在这种情况下数据轮播应该由要被传输的应用的每一个版本的群构成。由于这里没有版本更新机制，DownloadServerInitiate消息仅用来为描述最新版本的群提供关联关系。这意味着一个通过最高级别控制消息接入数据轮播的接收端将自动使用这个版本。

当仍然在传输一个应用已存在的版本时，如果要加入一个新版本，则将产生一个新的群，那么DownloadServerInitiate消息将被更新用来消除上一个“最新”群的所有关联关系，并且与新的“最新”群关联起来。这就强制限定了如果接收端想要继续接入仍然在被广播的旧版本，接收端应本地存储相关的最高级别的控制消息(DownloadServerInitiate)。

注：在data\_broadcast\_descriptor中的transactionId能被用来直接指向DownloadInfoIndication消息，该消息描述了仍然在数据轮播中的一个较旧版本的群。

例如使用接收端作为一个大容量存储器件的软件下载接口，即使相同文件的一个较新的版本正在被广播，它还希望继续完成广播一个大文件。

### A.7.5 transactionId值的分配和使用

在数据轮播中transactionId的使用方式继承于它在ISO/IEC 13818-6中定义的使用方式,这样可能显得有些复杂。transactionId有双重作用,它提供下载控制消息(DownloadInfoIndication和DownloadServerInitiate消息)的确认和版本更新机制。在数据轮播中transactionId唯一标明一个下载控制消息,然而当该消息的任何一个域被修改时,它应当“被增量”。

注:术语“被增量”在ISO/IEC 13818-6中使用,指的是控制消息发生改变时,transactionId的16-29位的值加1。

在数据轮播的应用范围内它解释为“被改变”。

transactionId已被切分成许多定义在表A.2中的子域。这反映了transactionId的作用(上文给出了大致的框架)并且数字电视制定了一些约束来减少接收端要求的过滤的最小级别。然而,为了增加互操作性,transactionId的分配被设计成与在目标接收端预期的过滤无关。

表 A.2 transactionId 子段

位	值	子段	描述
0	用户定义	更新标志	控制信息每次更新都将触发该位。
1-15	用户定义	识别	对于顶层控制信息,所有位为0; 非顶层控制信息,至少一位不等0。
16-29	用户定义	版本	每次控制信息更新,版本号加1。
30-31	位30-0 位31-非0	出处	在ISO/IEC 13818-6中定义,如果网络分配了transactionId,其值为0x02。

由于transactionId的功能,作为版本切换机制,数据轮播中任意消息的任何改变都将导致最高级别的控制消息的transactionId被增量。这些变化通过下面的数据轮播结构传输。对模块的任何变化将强制改变它的moduleVersion。该变化将通过DownloadInfoIndication消息中的模块描述相应的域反映出来,该消息描述任何包含它的群。在DownloadInfoIndication消息中的域被改变后,它的transactionId也应该改变,以标明一个消息的新版本。这些变化将再次通过描述这个超群的DownloadServerInitiate消息中对群进行描述的域反映出来。由于DownloadServerInitiate消息中的域已经被改变,它的transactionId也应该被改变。这非常有用,因为只要通过观察最高级别控制消息中的transactionId,就能检测到数据轮播中任何消息的改变。

如果任意控制消息中的transactionId都参考SI中data\_broadcast\_descriptor中相应的域,那么它要被更新以反映各种变化。这样的结果就是如果数据轮播内容发生变化,就应再去取那些被改变的SI表。由于SI的重复间隔可能到2s,这里可能会有我们不希望的、会降低动态数据反应速度的副作用。为了避免这样的结果,可以把data\_broadcast\_descriptor中的transactionId置为0xFFFFFFFF,以便指出任何最高级别控制消息都是有效的。

在TS流中下载控制消息的封装在ISO/IEC 13818-6中定义。它规定了将transactionId中2个最低有效字节拷贝到DSMCC\_section头的table\_id\_extension中。这意味着如果知道正在被广播的数据轮播的PID,通过将段过滤器设为table\_id=0X3B(下载控制消息)和table\_id\_extension=0X0000或0X0001,不用知道轮播数据的transactionId,就可以找到最高级别控制消息。

表A.3反映了不同消息类型的段头的编码。

表 A.3 DSM-CC 段头中的各域编码

消息	Table_id	Table_id_extension	Version_number	Section_number	Last_section_number
DSI	0x3B	DSI中transaction_id 的 两个最低有效字节	0x00	0x00	0x00
DII	0x3B	DII中transaction_id 的 两个最低有效字节	0x00	0x00	0x00
DDB	0x3C	Module_id	Module Version % 32	BlockNumber % 256	Section_number 的最大值

A.7.6 具体的数据轮播描述符的使用

这一节描述的所有描述符都是可选的。

A.7.6.1 type\_descriptor

数据轮播的群或模块的类型使用type\_descriptor来传输。它的使用是业务供应商专有的。按照媒体类型规范IETF RFC 1521和IETF RFC 1590, 由一个字符串来指定该模块的类型。

A.7.6.2 name\_descriptor

数据轮播的群或模块的名称使用name\_descriptor传输。它的使用是业务供应商专有的。

A.7.6.3 info\_descriptor

数据轮播的群或模块的相关信息使用info\_descriptor来传输。它的使用是业务供应商专有的。

A.7.6.4 module\_link\_descriptor

module\_link\_descriptor提供了关于一个群中的哪些模块应该被连接在一起的信息, 这样可以从该数据轮播中取出完整的数据。在这个描述符的两个域position和module\_id一起用来标明多个模块连接时, 模块连接列表中的第一个模块 (position=0x00, module\_id= 下一个模块号), 下一个模块 (position=0x01, module\_id=下一个模块号) 以及最后一个模块 (position=0x02)。

A.7.6.5 CRC32\_descriptor

使用CRC32\_descriptor, 给出一个完整模块上的CRC-32的计算值。这些CRC-32数据是描述符的一部分。

A.7.6.6 location\_descriptor

在下载ServerInitiate消息中的location\_descriptor标明能找到该数据轮播群的PID。DownloadInfoIndication消息应在该PID上找到。相同的机制还可用在下载InfoIndication消息上, 用来找到不同PID上的所有模块。

这对于不同的用户来说是一种处理群和模块的非常有用的方法。

A.7.6.7 estimated\_download\_time\_descriptor

estimated\_download\_time\_descriptor为接收端提供下载模块或群的时间。

需要注意它的使用方式。很显然, 下载时间对于传输数据轮播的可用比特率是敏感的。数据轮播生成位置和播放位置不同可能会出现问題, 例如, 同一数据轮播如果在第一天以一个比特率(例如: 1 Mbits/s)而在第二天以另一个比特率(例如: 2Mbits/s)播出, 那么可能两天中有一天的下载时间是不准确的(甚至两天都不准)。

注: 基于最小播放比特率可以计算出预计下载时间的值, 很明显, 这种方法基于接收到的消息, 向接收端指出收到多少数据更为实用。

A.7.6.8 group\_link\_descriptor

群中模块的描述由DownloadInfoIndication消息提供。被描述模块的数目由该消息的最大尺寸和每个模块描述的尺寸来决定。GB/T 17975.1段的封装限制了下载控制消息的最大尺寸低于4K字节。一个简单模块描述的大小(例如30个字节的基本信息和名字描述符)大概是40个字节。这就意味着

DownloadInfoIndication消息可以描述大约100个模块，这在绝大多数情况下已经足够了，但并不能满足所有的情况。

在后一种情况中，要描述一个大群中的所有模块，就应有足够多的DownloadInfoIndication消息。这就有效地把一个大群分割成许多较小的群，每个较小的群都由各自的DownloadInfoIndication消息来定义。为了能够重建原始的大群，这些较小的群应连接在一起。这应通过包含DownloadServerInitiate消息中的，每一个新的较小群的描述中的group\_link\_descriptor()来实现。

#### A.7.6.9 private\_descriptor

如果一个业务供应商有private\_descriptor的需要，将使用private\_descriptor的句法。

#### A.7.6.10 compressed\_module\_descriptor

compressed\_module\_descriptor的出现标明模块中的数据具有定义在IETF RFC 1951中的“zlib”结构。ZLIB结构的定义见表A.4。

表 A.4 zlib 结构

句法	字节数
zlib structure	
{	
compression_method	1
flags_check	1
compressed_data	n
check_value	4
}	

#### A.7.7 SI和PSI中的信息

数据轮播的访问可以通过SI或PSI中的描述符来完成。这就为定义业务提供了灵活性。

##### A.7.7.1 SI中的描述符

对于使用数据轮播的业务，data\_broadcast\_descriptor应出现在SDT或EIT中。也就是说，这种描述符的使用是强制性的。

data\_broadcast\_id域应置为0X0006来标明数字电视数据轮播的使用。

component\_tag通过与PMT中具体流的stream\_identifier\_descriptor中的component\_tag相关联的方式来识别传输数据轮播所使用的PID。

data\_carousel\_info结构在selector\_byte中传输。

carousel\_type\_id标明了使用了哪种数据轮播，见表23。

transaction\_id的使用在A.7.4中描述。

time\_out\_value\_DSI和time\_out\_value\_DII向接收端给出标识，告知接收端在判定一个错误情况之前，应等待多少时间。

为了优化接收设备，使用了leak\_rate。通过给出leak\_rate，解码器计算是否能够解码一个业务。移出速率也可以由smoothing\_buffer\_descriptor或maximum\_bitrate\_descriptor给出，在这两个描述符都出现的情况下，两个描述符给出的值应一致。但是，不推荐使用maximum\_bitrate\_descriptor。

使用基于SI代替PSI访问轮播的优点在于：

- 使用transactionId可以明确地标明数据轮播中的最高级别控制消息；
- 通过在data\_broadcast\_descriptor包含transactionId域，数据轮播的更新（这将导致transactionId的变化）可以只通过过滤SI就能被检测到；

注：这种方法可以通过使用特殊的transactionId值(0xFFFFFFFF, 在A.7.4中描述)来避免。

- 描述符不占用PSI表(可以说是一种稀缺资源)的空间。

使用基于SI代替PSI接入轮播的缺点在于，SI的重复周期可能长达2s，这可能会对业务首次访问带来延迟。

A.7.7.2 PSI中的描述符

对于使用数据轮播的业务，data\_broadcast\_id\_descriptor可能会出现在PMT中，也就是说该描述符的使用是可选的。

data\_broadcast\_id域应置为0X0006来标明数字电视数据轮播的使用。

使用这种机制的优点在于，PSI的最大重复周期仅为0.1s，这就能实现对业务的首次快速访问。

这种机制的缺点在于：

- 没有transactionId清楚地识别最高级别控制消息。这样只有来自单个数据轮播的下载控制消息可以在被标明的ES流中传输；
- 该描述符没有提供任何关于下载控制消息超时周期的信息。这些信息仍然要从SI的描述符中获得；
- 该描述符消耗PSI表的部分空间(即使很小)；
- 该描述符在SI中仍然需要。

A.8 对象轮播

A.8.1 概述

DSM-CC对象轮播使得利用目录对象、文件对象和流对象的，从广播服务器到广播接收端(客户端)的对象的结构化的群的传输变得容易。实际的目录和内容(对象实现)都位于该服务器上。该服务器不断地将相关的对象插入到使用对象轮播协议的，数字电视相兼容的TS流当中去。如图A.5所示，对象轮播是数字电视业务的一部分。传输的目录和文件对象包含对象的内容，而传输的流对象在广播中是其它流的参考基准。该流对象可能包含在一个特定流中广播的关于DSM-CC事件的信息。DSM-CC事件可以通过一般流数据来广播，并且可以用来触发DSM-CC应用。

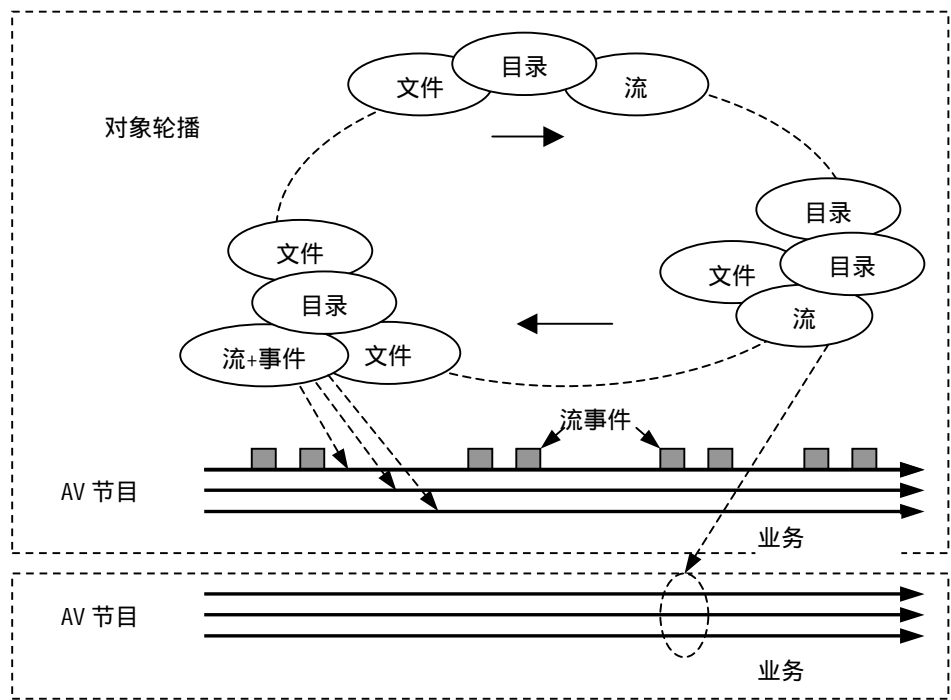


图 A.5 含有对象轮播的业务示例

通过读取反复传输的轮播数据，多个客户端可以恢复对象实施，因此在本地对象实施中可以模仿服务器的对象。轮播中的对象为客户提供了访问应用和应用内容的方式，或多或少地，好象与服务器之间存在交互连接一样。

下面内容提供了在数字电视广播网络和遵循ETS 300 802的交互系统中实施和使用DSM-CC U-U 对象轮播的应用指南。其主要集中在以下几方面：

- 平台无关性；
- 应用在U-U 对象轮播中的BIOP控制结构的编码；
- 应用在U-U 对象轮播中的BIOP数据消息的编码；
- 下载数据轮播消息的编码；
- DSM-CC 段的编码；
- 对象轮播PSI描述符的应用；
- 对象轮播SI描述符的应用。

图A. 6表明了为广播和交互网络定义的协议栈，图中粗线所包围的部分说明了其涉及的范围。

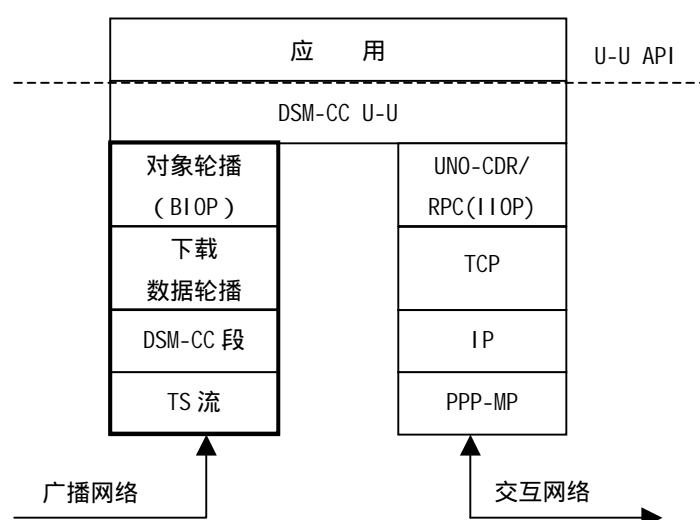


图 A. 6 对象轮播协议在交互业务结构框架中的位置

## A. 8. 2 平台无关性

### A. 8. 2. 1 概述

对象轮播规范是平台无关的，与 ISO/IEC 13818-6 规范中 U-U 的规定以及 CORBA 定义的 ORB 结构兼容。在 DSM-CC U-U 系统环境中，对象的结构化组被称作业务域。业务域有一个业务网关，该网关可以看作是对象结构化组最高一级的目录，而且这样一个业务网关可以为公布给客户端的业务名称（也就是对象名称）的图表提供相互关系。业务域可以在交互网络的服务器上定位，也可以在广播网络的服务器上定位，在后面一种情况下，业务域中的对象通过对象轮播的方式广播。

注意：对象命名应遵照本标准的附录 D。

对象轮播中一个对象的数据和特性是在单独一个消息中传输的。消息的格式由对象轮播规范指定，称做 BIOP 消息格式。BIOP 消息通过 DSM-CC 数据轮播的一个模块来广播，但是一个模块可包含一个或多个 BIOP 信息。依照 DSM-CC 数据轮播规范，一个模块可以被分成一个或多个块，块在 DownloadDataBlock 消息中传输。每个 DownloadDataBlock 消息大小相同（除了模块的最后一个数据块可能小些），并且按照 DSM-CC 的规定在段中依次传输。DownloadDataBlock 消息在段中封装的规则是通过解复用器中常见的硬件过滤器，块能够直接从 TS 流中得到。

业务域中的对象通过使用对象基准来标识。DSM-CC U-U 使用 CORBA 定义的 IOR 结构。对象基准包含了在网络中一个或多个服务器上检索对象所需的全部信息。IOR 中包含有被存储对象的实例的位置信息的结构体，称作类型体 (Profile Body)。一个 IOR 可以包含多个类型体以指示对象的多个（网络）位置。

对象轮播规范使用两个类型体，即 BIOP 类型体和 LiteOptions 类型体。BIOP 类型体用于在同一个对象轮播里检索对象，LiteOptions 类型体用于在另一个对象轮播里检索对象。



BIOP类型体只用来检索同一对象轮播(即业务域)中的对象。它通过使用对象轮播的标识符、广播对象的模块标识符和模块中标识对象的唯一的键值来实现对象的唯一标识。对象轮播的标识符通过节目中PMT的描述符链接到数字电视业务上。

Lite Options类型体用于检索另一业务域(要么是交互的,要么是广播的)中的对象。它通过使用全局唯一的NSAP地址格式来帮助应用链接到其它业务域。对于数字电视网络中的业务域, NSAP地址连接到一个特定的数字电视业务。

A.8.2.2 所支持的 U-U 对象

对象轮播规范是用来支持DSM-CC U-U的API所定义的大量接口的。本条提供了对象轮播中所支持的对象和接口的指南, 接口定义见ISO/IEC 13818-6。READER接口所支持的对象见表A. 5。

表 A. 5 具有 READER 支持接口的对象

对象	READER接口所支持的对象
DSM::Directory	Access, Directory
DSM::File	Base, Access, File
DSM::Stream	Base, Access, Stream
DSM::ServiceGateway	Access, ServiceGateway
BIOP::StreamEvent	Base, Access, Stream, Event

应该注意的是广播网络的API句法与交互网络的API句法稍有差别, 原因在于网络自身的广播特性。一个典型的例子是对于通过广播网络传输的流, 其流接口的暂停API指令会让屏幕上的画面静止, 但并不中止(广播)流的传输。

本附录不提供任何关于在广播网络中精确操作DSM-CC U-U 接口的指导。

DSM-CC接口访问会返回缺省属性(即目标特性, 如只读属性和访问时间), 因为这些特性的广播在BIOP (ISO/IEC 13818-6, GB/T 17975.1)中没有定义。

本附录不提供任何关于对象轮播中访问特性的广播规范。

图A. 7 用OMT注释法标明了U-U 对象之间的关系。

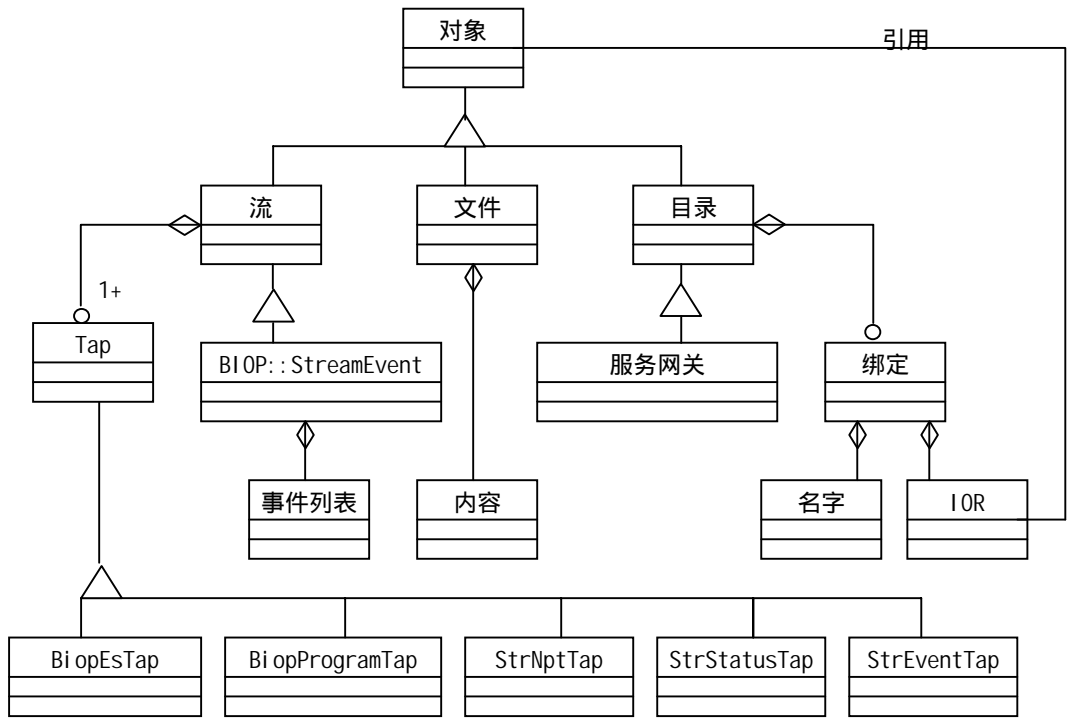


图 A. 7 对象轮播所支持的对象

在对象轮播中，可为每个对象传送以下信息：

- 目录对象数据 (Directory object data)：绑定列表，其中每个绑定都把一个名字与一个对象基准(IOR)连接起来。另外，每一个绑定还可能包含绑定对象的一些附加特性，以支持通过目录快速浏览。在当前的对象轮播规范中，这仅用于文件对象的contentSize属性；
- 文件对象数据 (File object data)：文件内容数据和contentSize属性；
- 流对象数据 (Stream object data)：标识符(称为 Tap)列表，指向广播网络中一个或多个流。每个Tap要么指向基本流(Bi opEsTap)，要么指向完整的节目(Bi opProgramTap)。另外，其它的标识符可能会出现，以指向包含流控制信息的广播通道(例如：指向NPT、状态/模式、事件的StreamDescriptor的Tap)。流对象数据还包括StreamInfo属性；
- 业务网关对象数据 (ServiceGateway object data)：与目录对象相同，因为业务网关是从目录继承的。对于业务网关的特别之处是，它包含有业务域的根目录；
- 流事件对象数据 (StreamEvent object data)：类似于流对象数据，但是扩展了EventList属性和 eventID列表。这些属性包括了DSM-CC事件名列表以及这些事件名称与eventID的映射关系。

#### A.8.2.3 对象传输

在对象轮播中一个U-U对象的数据和特性在一个消息中传输。消息格式由BIOP规定，称做BIOP Generic Object Message format(BIOP通用对象消息格式,或简称BIOP消息格式)。一个 BIOP消息由一个MessageHeader、一个MessageSubHeader和一个MessageBody构成。MessageHeader提供了BIOP协议版本和BIOP消息长度。MessageSubHeader包含了关于传输对象的信息，如objectType(文件、流和目录)和objectKey(模块中的唯一标识符)。MessageBody依赖于objectType，包含实际的U-U对象数据。一个BIOP消息的大小是可变的。

BIOP消息在数据轮播的模块中广播，见ISO/IEC 13818-6。一个模块由一个或多个链接的BIOP消息组成，见图A.8，因此为长度可变的。在模块中每个对象由objectKey标识，对象可以很容易地通过解析BIOP消息的objectKey域和BIOP消息的长度找到。

依照 DSM-CC数据轮播规范，每个模块被分成一个或多个通过DownloadDataBlock消息传输的数据块。每个DownloadDataBlock消息大小相同(只有模块的最后一个块可能小一些)，在私有段中依次传输，见ISO/IEC 13818-6。DownloadDataBlock在私有段中的封装规则使得通过解复用器中常见的硬件过滤器，能够直接从TS流得到块。

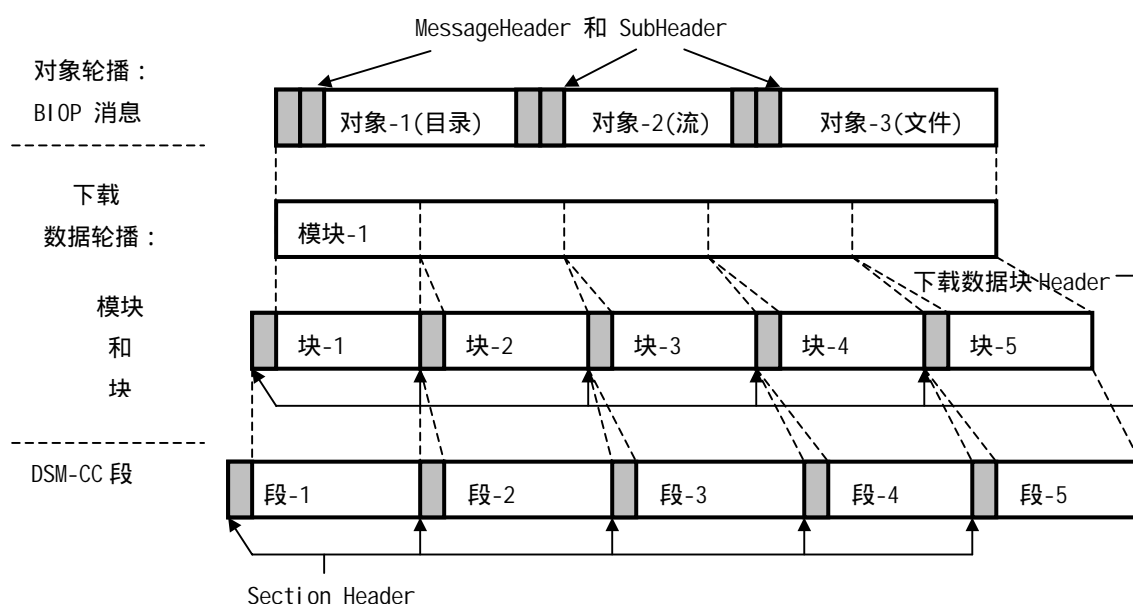


图 A.8 BIOP 消息在模块、数据块、GB/T 17975.1 段中的封装和切分

从广播网络中获取对象，应完整获取包含该对象的模块，这就要求了解模块的传输参数，如模块版本、模块大小、块大小、时序和广播通道。这些传输参数在DownloadInfoIndication消息中传输，该消息应在获取DSM-CC的模块之前从网络中得到。一个DownloadInfoIndication消息能描述多个模块的传输参数。因而从广播网络中检索一个对象需要两个步骤。

在BIOP中，一个业务域的业务网关的对象基准通过DownloadServerInitiate消息传输，见ISO/IEC 13818-6。通过PSI 或者PSI 与SI 一起，可以找到这个消息。

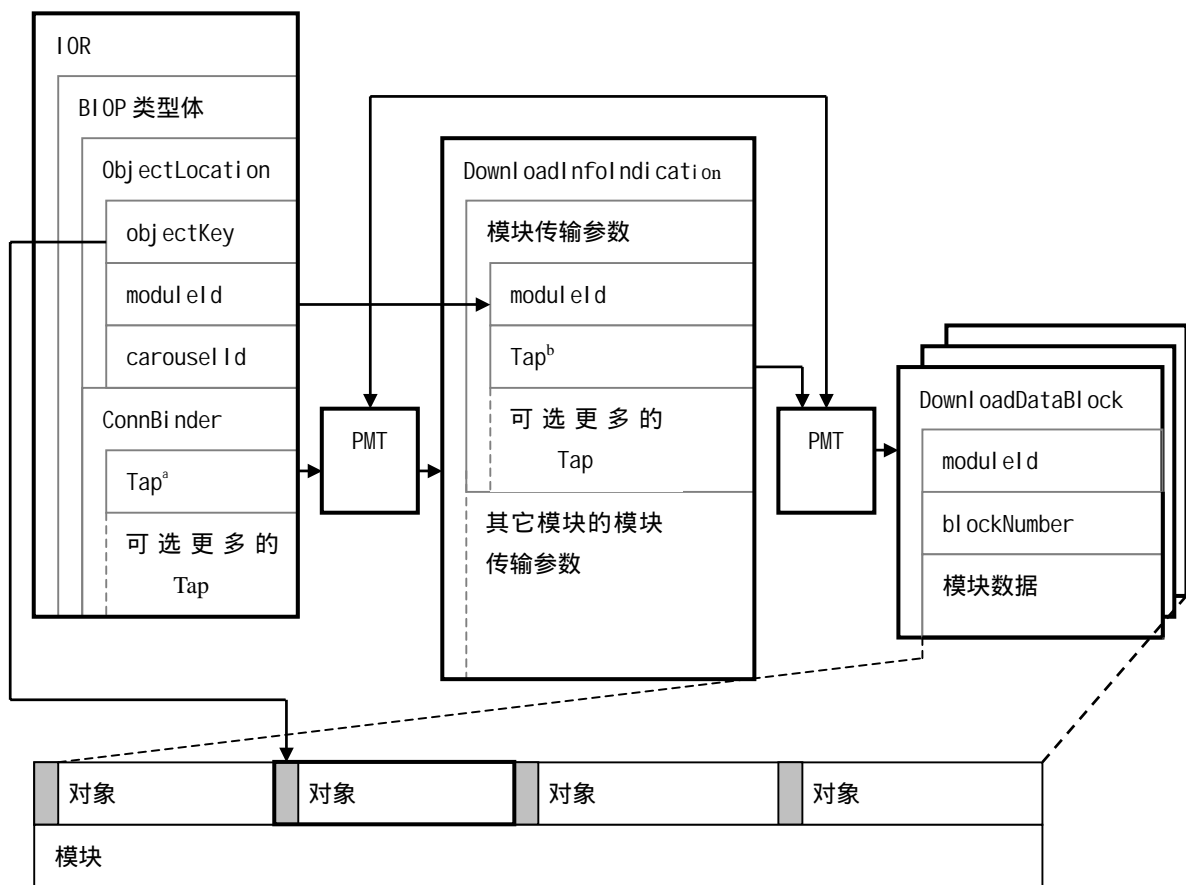
A.8.2.4 对象基准

BIOP 使用了 CORBA 的 IOR-Interoperable Object Reference(可互操作的对象基准)，见 ISO/IEC13818-6。对象基准为每个网络位置都保留了一个类型体。类型体的类型依赖于从服务器获取对象所必需的协议。

因为IOR指向同一个广播业务域中(在同一个对象轮播中)的对象，BIOP的类型体就标明了传递对象数据和属性的BIOP信息的位置，因此BIOP由ObjectLocation和ConnBinder组成，见图A. 9。

图A. 9说明了带有BIOP类型体的IOR如何被解析成它所指向的对象。ObjectLocation通过carouselId、moduleId、objectKey标识了U-U对象轮播中的对象。

ConnBinder由Tap的序列组成。Tap通过PMT来标识广播DownloadInfoIndication消息、含有对象模块传输参数的码流。



a TapUse=BIOP\_DELIVERY\_PARA\_USE

b TapUse=BIOP\_OBJECT\_USE

图 A.9 带有 BIOP 类型体的 IOR 如何被解析成对象

ConnBinder至少要包含一个通过PMT指向DownloadInfoIndication消息的Tap。IOR中的moduleId用于判定DownloadInfoIndication消息中合适的传输参数。传输参数要依次包含至少一个指向(也通过PMT)传送模块的DownloadDataBlock消息的Tap。最后,来自IOR的objectKey用来标识模块中的对象消息。

注:ConnBinder和模块传输参数都可能包含多于一个的Tap,额外的Tap可能标识了传输同一模块(可能使用其它传输参数)的其它可选码流。

对于指向另外一个业务域的IOR,就要使用LiteOptions类型体。LiteOptions类型体使用全局唯一的NSAP地址来标识交互的或者广播性质的业务域。对于数字电视兼容广播网络中的业务域,NSAP地址标识了一个本标准规定的具体数字电视业务,见图A.10。

图A.10说明了含有LiteOptions类型体对象基准(IOR)如何被解析为广播业务域中的业务网关。该类型体包含了ServiceLocation成员,其中依次包含有NSAP地址。通过广播对象轮播的数字电视业务的transport\_stream\_id, service\_id, 和 original\_network\_id, NSAP地址标识了广播业务域。通过使用业务的PAT和PMT,在DownloadServerInitiate消息中可以找到业务网关的IOR。该IOR依次包含有指向广播业务域的业务网关对象的BIOP类型体。BIOP类型体的解析操作与图A.9相同。

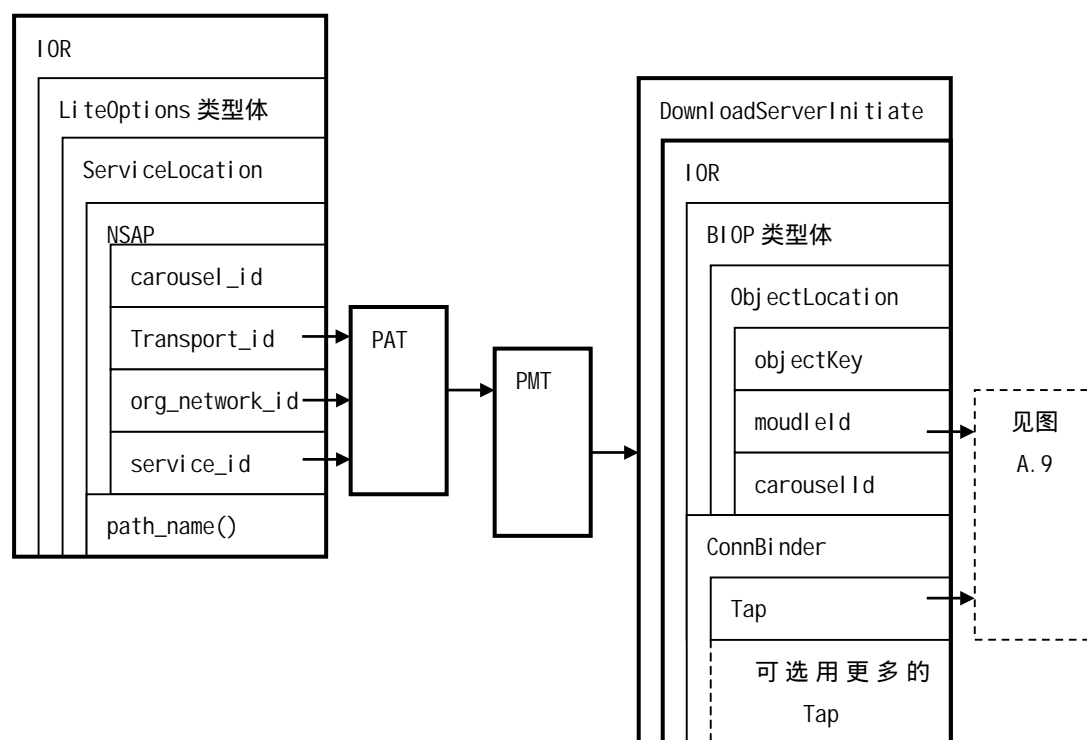


图 A.10 带有 LiteOptions 类型体的 IOR 如何被解析成业务网关

#### A.8.2.5 Tap和关联

IOR并不通过PID直接指向码流,因为再复用器可以改变PID。因而DSM-CC定义了Tap,见ISO/IEC 13818-6,按照类似于SI中component tag的方式使用。

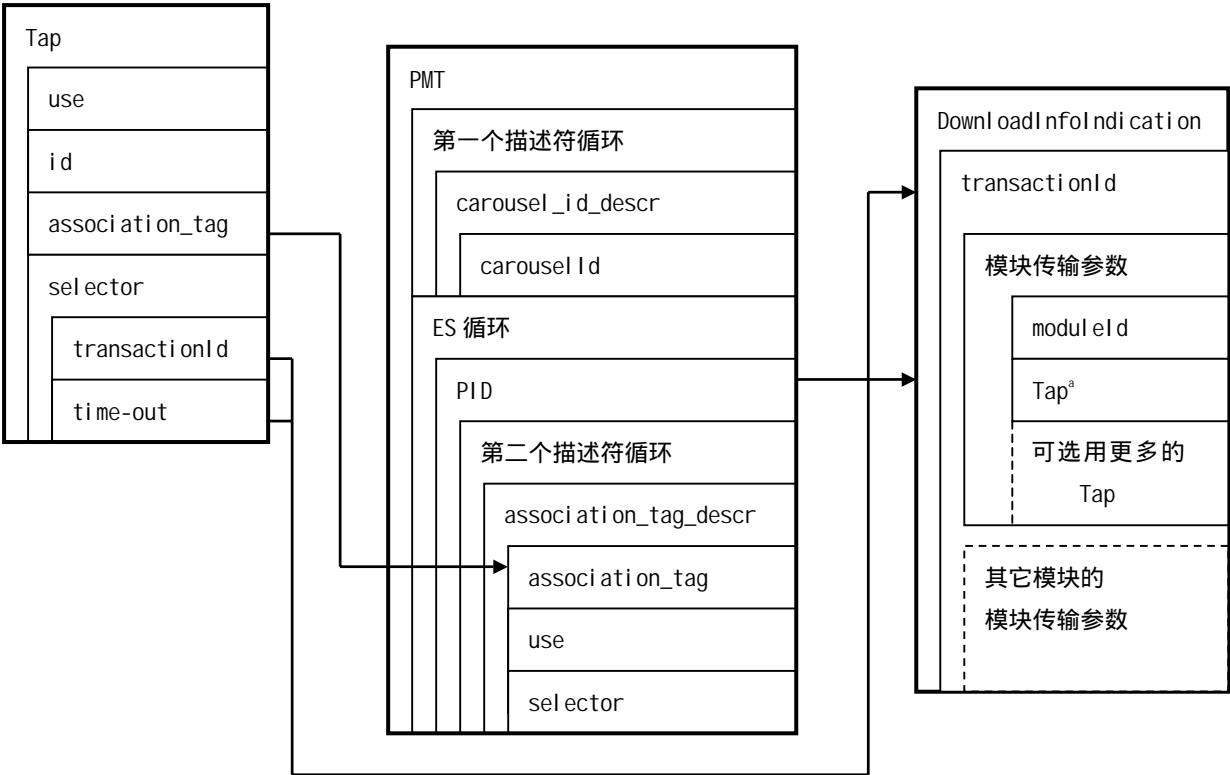
一个tap由以下几部分组成:

- id: 用户私有(不用时置0);
- use: 该域标明 Tap 的用途;
- association\_tag: 该域用于把 Tap 和一个特定的 ES 流关联起来;
- selector: 可选的选择器,从相关 ES 流中选择相关数据。选择器的存在依赖于 use 域。

下面use域的赋值在对象轮播中的使用:

- BIOP\_DELIVERY\_PARA\_USE: BIOP 类型体的 ConnBinder 成员应包含 Tap 来标识广播 DownloadInfoIndication()消息的联结,而在 DownloadInfoIndication()消息中描述了传输

- 对象的模块的传输参数，见图 A. 11。该 Tap 的 selector 域包含一个 transactionId 域和一个 timeout 域。transactionId 的值要设置成包含模块传输参数的 DownloadInfoIndication() 消息的 transactionId 的值。timeout 要设定为用以判定获取 DownloadInfoIndication 消息是否超时、以  $\mu s$  为单位的超时时限；
- BIOP\_OBJECT\_USE：在传送模块的模块传输参数的 DownloadInfoIndication() 消息中使用，表明模块在哪个 ES 流中传输。selector 域为空；
  - BIOP\_ES\_USE、BIOP\_PROGRAM\_USE：码流对象包含这些 Tap 以标明码流与对象相关。不过 BIOP\_ES\_USE 指向一个基本流、BIOP\_PROGRAM\_USE 指向一个完整的 GB/T 17975.1 节目(数字电视业务)。两类 Tap 的 selector 域都为空；
  - STR\_STATUS\_AND\_EVENT\_USE、STR\_EVENT\_USE、STR\_STATUS\_USE、STR\_NPT\_USE：流对象和流事件对象可能包含这些 Tap 以标明与对象相关的子流。所有这些 Tap 的 selector 域均为空。



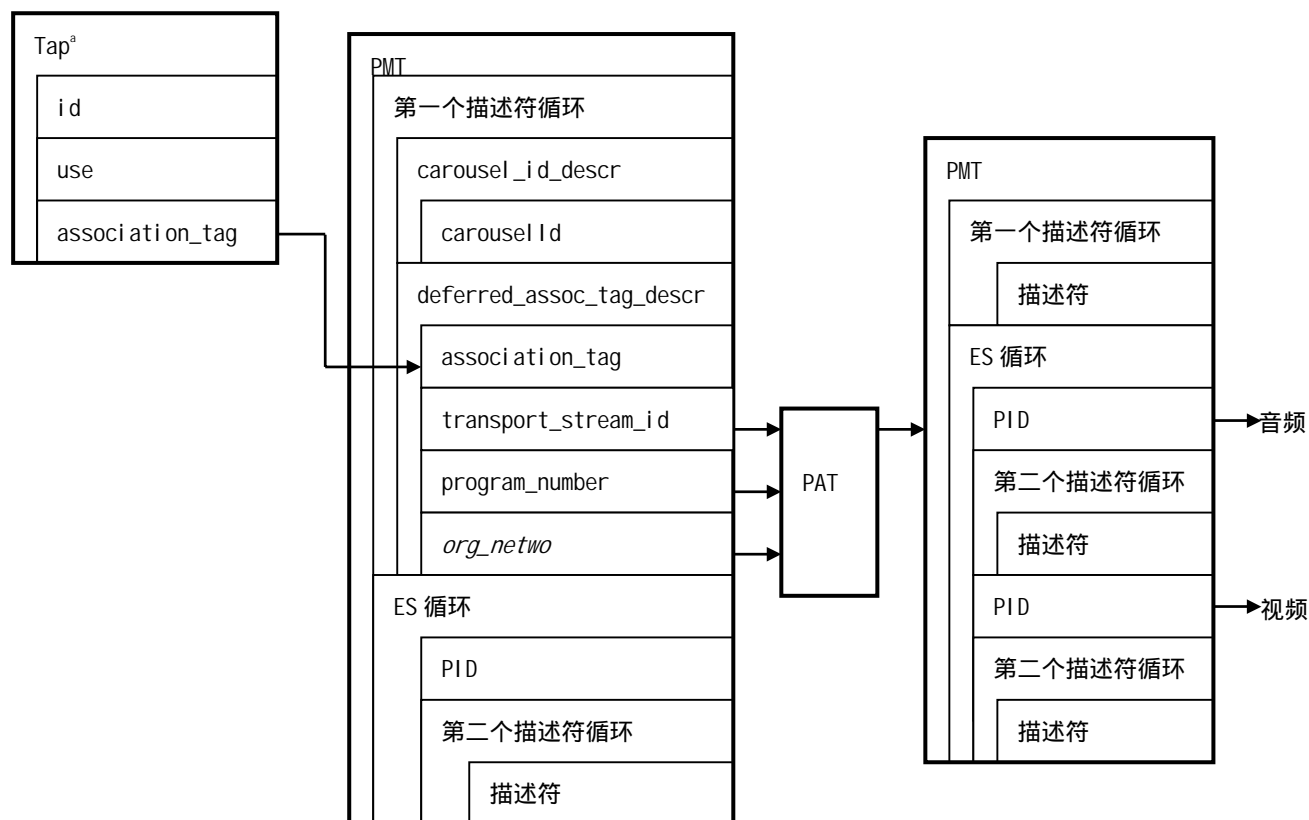
a TapUse = BIOP\_DELIVERY\_PARA\_USE

图 A. 11 标识基本流的 association\_tag 描述符的用法

在解析对象的过程中，客户端应把 Tap 与广播网络的链路关联起来。因此，客户端需要关联表来完成 Tap 和广播网络链路的关联。为了支持在基于 GB/T 17975.1 TS 流的广播网络中实现 U-U 对象轮播，ISO/IEC 13818-6 定义了三个能插入 GB/T 17975.1 PMT 的描述符：

- carousel\_id\_descriptor 用一个 carousel\_id 标注 PMT，说明在 PMT 中出现的所有 association\_tag 都属于 U-U 对象轮播(为 association\_tag 提供一个范围)，见图 A. 11；
- association\_tag\_descriptor 用 association\_tag 标注一个 ES 流，把所有包含此 Tap 的 Tap 与这个 ES 流关联起来，见图 A. 11。与 Tap 类似，association\_tag\_descriptor 也包含一个 use 域和一个可选择的 selector 域。把该 use 域置为 0x0000 就标志着在此 ES 流中传输了 DownloadServerInitiate 消息(DSI)。这个 DSI 包含了业务网关的 IOR；

- The deferred\_association\_tags\_descriptor 包含一系列的，与其它节目的 ES 流(PMT)相关联的，或者指向其它节目的 association\_tag。图 A.12 说明了 deferred\_association\_tags\_descriptor 如何指向其它节目。



a TapUse = PROGRAM\_USE

图 A.12 使用 deferred\_association\_tags\_descriptor 标识节目

### A.8.3 BIOP控制结构

BIOP控制和数据结构在ISO/IEC 13818-6中定义，使用了定义在CORBA中的平台无关规范语言OMG IDL。“bits-on-the-wire”编码由转化IDL句法到bits on the wire的CDR编码规则来定义。BIOP使用的是CDR Lite编码规则，见ISO/IEC 13818-6，该规则使用了序列的最大长度和字节对齐。因此，相比CDR来说，CDR Lite在数据的打包上更加紧凑。

注：这就意味着所有的串都是由一个空字符结束的，而且这个字符属于字符串长度的一部分，例如表A.12中的objectKind\_length和objectKind\_data域。

在该段当中BIOP控制结构使用了GB/T 17975.1的句法并且提供了和该域编码相关的指导准则。那些受本附录影响的域在表中用阴影表示。在段0中BIOP消息使用了GB/T 17975.1句法。在ISO/IEC 13818-6中定义的IDL结构和在下面段定义的结构之间有差异的情况下，将以ISO/IEC 13818-6中的定义为准。

#### A.8.3.1 IOR

DSM-CC使用由OMG定义的，用在客户机-服务器可交互性接口的IOR格式。表A.6是IOP::IOR的句法，见ISO/IEC 13818-6。

表 A.6 IOP::IOR 的句法

句法	比特数	类型	值	注释
IOP::IOR { type_id_length for (i=0; i<N1; i++){	32	ui msbf	N1	
type_id_byte	8	ui msbf		见表A. 7
}				
if (N1 % 4 != 0) { for (i=0; i<(4-(N1%4)); i++){				CDR alignment rule
alignment_gap	8	ui msbf	0xFF	
}				
}				
taggedProfiles_count for (n=0; n<N2; n++){	32	ui msbf	N2	Profile bodies
IOP::taggedProfile(){				
profile_id_tag	32	ui msbf		例如 TAG_BIOP 例如 TAG_LITE_OPTIONS
profile_data_length for (i=0; i<N3; i++){	32	ui msbf	N3	
profile_data_byte	8	ui msbf		例如 BIOPProfileBody 例如 LiteOptionsProfileBody
}				
}				
}				
}				

IOP的type\_id\_byte域形成一个代表对象类型的串。对于在CORBA机制中的对象标识，串标识以“<模块>::<接口>”的形式使用。为了减小IOP的大小，DSM-CC定义了3个字符的别名。用在对象轮播中的对象type\_id列在表A. 7中。

表 A.7 U-U 对象的 type\_id

type_id全称	type_id别名
“DSM::Directory”	“dir”
“DSM::File”	“fil”
“DSM::Stream”	“str”
“DSM::ServiceGateway”	“srg”
“BIOP::StreamEvent”	“ste”

只有type\_id这个别名可用在数字电视系统中。这意味着当使用type\_id时，服务器没有必要插入对齐填充字节。

指向相同U-U对象轮播中传输对象的IOP包含了taggedProfileList中的BIOP的类型体。ISO/IEC 13818-6 允许一个IOP包含多个类型体。

数字电视接收端应该至少能够处理这些类型体中的第一个，其它的类型体可以被忽略。

IOP中至少应该包含一个用tag标注的类型。对于对象轮播广播中的对象来说，第一个用tag标注的类型要么是一个TAG\_BIOP类型体，要么是一个TAG\_LITE\_OPTIONS。

## A.8.3.2 BIOP类型体

BIOP类型体有一个跟随在MultipleComponentProfile结构后的LiteComponetProfile结构。表A.8说明了一个包含强制性的ObjectLocation组件和ConnBinder组件的BIOP类型体。

表 A.8 BIOP 类型体的句法

句法	比特数	类型	值	注释
BIOPProfileBody {				
profileId_tag	32	uimsbf	0x49534F06	TAG_BIOP(BIOP Profile Body)
profile_data_length	32	uimsbf		
profile_data_byte_order	8	uimsbf	0x00	big endian 字节顺序
liteComponents_count	8	uimsbf	N1	
BIOP::ObjectLocation {				
componentId_tag	32	uimsbf	0x49534F50	TAG_ObjectLocation
component_data_length	8	uimsbf		
carouselId	32	uimsbf		
moduleId	16	uimsbf		
version.major	8	uimsbf	0x01	BIOP 协议主版本 1
version.minor	8	uimsbf	0x00	BIOP 协议副版本 0
objectKey_length	8	uimsbf	N2	
for (k=0; k<N2; k++){				
objectKey_data_byte	8	uimsbf		
}				
}				
DSM::ConnBinder {				
componentId_tag	32	uimsbf	0x49534F40	TAG_ConnBinder
component_data_length	8	uimsbf		
taps_count	8	uimsbf	N3	
BIOP::Tap {				
id	16	uimsbf	0x0000	用户私有
use	16	uimsbf	0x0016	BIOP_DELIVERY_PARA_USE
association_tag	16	uimsbf		
selector_length	8	uimsbf	0x0A	
selector_type	16	uimsbf	0x01	
transactionId	32	uimsbf		
timeout	32	uimsbf		
}				
for (m=0; m<N3-1; m++){				
BIOP::Tap {				
id	16	uimsbf	0x0000	用户私有
use	16	uimsbf	0x0016	BIOP_DELIVERY_PARA_USE
association_tag	16	uimsbf		
selector_length	8	uimsbf	N4	
for (i=0; i<N4; i++){				
selector_data_byte	8	uimsbf		



表A.8 BIOP类型体的句法（续）

句法	比特数	类型	值	注释
<pre>    }   } } for (n=0; n&lt;N5; n++){   BIOP::LiteComponent {     componentId_tag     component_data_length     for (i=0; i&lt;N6; i++){       component_data_byte     }   } }</pre>	32 8 8	ui msbf ui msbf ui msbf	N6	N5 = N1 - 2

当byte\_order域值为0x00时表明下面的数据采用big-endian字节排序编码。

CarouselId域为moduleId域指明了前后关系。carouselId在广播网络中唯一标明该轮播并且允许该轮播的分散实现。

BIOP类型体只用来指向同一轮播中的对象。也就是carouselId的值等于传输了IOR的对象轮播的carouselId值。为了指向其它轮播中的对象, 要使用Lite Options类型体。

LiteOptionComponents 列表 应该 按照 次序 包含 一个 BiopObjectLocation 和 一个 DsmConnectionBinder作为头两个组件。

该moduleId标明了该轮播中传输该对象的模块。

ObjectKey标明了广播ObjectKey的模块的对象。该域是由服务器提供的一系列字节并且只对该服务器有意义。

ObjectKey长度的值应该小于等于0x04。

多个Tap可能会共享相同的association\_tag，使一个ES流可以用作多个目的。表A. 9说明了定义了Tap的用途。

表 A. 9 BIOP 类型体中 Tap 允许使用的 TapUse 定义

TapUse 域	值	PID 中的广播内容
BIOP_DELIVERY_PARA_USE	0x16	模块传输参数
BIOP_OBJECT_USE	0x17	模块中的 BIOP 对象

如果BIOP\_DELIVERY\_PARA\_USE出现在ConnBinder组件中，那么它将是ConnBinder中的第一个Tap。

数字电视兼容接收端将跳过IOR中的BIOP类型体中的BIOP\_OBJECT\_USE。

如果不使用，id域的值应被置为0。

TapUse的值为BIOP\_DELIVERY\_PARA\_USE的Tap域的语义如下：

- use域标明使用了该Tap，该域的值应该为BIOP\_DELIVERY\_PARA\_USE；
- association\_tag标明广播DownloadInfoIndication的信道(即ES流)；
- 该selector域应包含值为MESSAGE(=0x0001)的selectorType、TransacoiId和timeout域。TransactionId域的值应置为包含该模块传输参数的DownloadInfoIndication() 消息的

TransactionId。timeout域应该以 $\mu s$ 为单位标明超时时限，用来表示DownloadInfoIndication()消息的接收超时。

TapUse值为BIOP\_OBJECT\_USE的Tap域的语义如下：

- use域标明使用了该Tap，该域的值应该为BIOP\_OBJECT\_USE；
- association\_tag标明了广播该模块的信道(即ES流)；
- 该selector域的值应为0长度。

注：TapUse值为BIOP\_OBJECT\_USE的Tap，在数字电视系统中仅用于DownloadInfoIndication()而不用在IOR中。

#### A.8.3.3 Lite Options类型体

为了指向其它业务域中的对象，IOR的Lite Options类型体中要包含ServiceLocation。当DSM-CC U-U API使用者试图解析一个名字时(Directory::resolve，见ISO/IEC 13818-6)，那将导致遇到这样一个IOR，将产生一个SERVICE\_XFR异常(exception)。SERVICE\_XFR异常中携带有在IOR的Lite Options类型体中的ServiceLocation结构。API使用者可以通过ServiceLocation结构中的serviceDomain来依次连接新的业务网关。PathName是可选的，它包含了可以让业务网关找到对象的路径。

Lite Options类型体有一个跟在MultipleComponentProfile结构之后的LitecomponentProfile结构。表A.10给出了一个含有ServiceLocation组件的Options类型体的句法。

表 A.10 包含有 ServiceLocation 组件的 Options 类型体的句法

句法	比特数	类型	值	注释
LiteOptionsProfileBody {				
profileId_tag	32	ui msbf	0x49534F05	TAG_LITE_OPTIONS (Lite Options 类型体)
profile_data_length	32	ui msbf		
profile_data_byte_order	8	ui msbf	0x00	big endian 字节顺序
component_count	8	ui msbf	N1	
DSM::ServiceLocation {				
componentId_tag	32	ui msbf	0x49534F46	TAG_ServiceLocation
component_data_length	32	ui msbf		
serviceDomain_length	8	ui msbf	0x14	轮播 NSAP 地址的长度
serviceDomain_data()	160	ui msbf		轮播 NSAP 地址, 见表 A.11
CosNaming::Name() {				路径名称
nameComponents_count	32	ui msbf	N2	
for (i=0; i<N2; i++){				
id_length	32	ui msbf	N3	NameComponent id
for (j=0; j<N3; j++){				
Id_data_byte	8	ui msbf		
}				
kind_length	32	ui msbf	N4	NameComponent kind
for (j=0; j<N4; j++){				
kind_data_byte	8	ui msbf		同 type_id, 见表 A.7
}				
}				
initialContext_length	32	ui msbf	N5	
for (n=0; n<N5; n++){				
InitialContext_data_byte	8	ui msbf		
}				
}				

表A. 10 包含有ServiceLocation组件的Options类型体的句法（续）

句法	比特数	类型	值	注释
<pre>    }</pre>				
<pre>    for (n=0; n&lt;N6; n++) {         BIOP::LiteOptionComponent{             componentId_tag             component_data_length             for (i=0; i&lt;N7; i++){                 component_data_byte             }         }     }</pre>	32 8 8	ui msbf ui msbf ui msbf	N7	N6 = N1 - 1
<pre>    }</pre>				

ServiceLocation组件应为该类型体的第一个组件。

A. 8. 3. 4 轮播NSAP地址

每个U-U对象轮播的实例都代表一个业务域。每个业务域都有一个全局唯一的，用来标明具体轮播的实例，叫做轮播NSAP地址(Network Service Access Point，网络业务接入点)。其句法见表A. 11。

表 A. 11 Carousel NSAP address 的句法

句法	比特数	类型	值	注释
<pre>carouselNSAPaddress(){     AFI     Type     carouselId     specifierType     specifierData{IEEE_OUI}     service_location(){         transport_stream_id         original_network_id         service_id         reserved     } }</pre>	8 8 32 8 24 16 16 16 32	ui msbf ui msbf ui msbf ui msbf ui msbf ui msbf ui msbf bsl bf	0x00 0x00 0x01 0x<> 0xFFFFFFFF	为私有使用定义的 NSAP 对象轮播 NSAP 地址 IEEE OUI 数字电视的 OUI 常量 (=program_number)

AFI、type、carouselId、specifierData、transport\_stream\_id、original\_network\_id、service\_id和各域定义见本标准。

A. 8. 4 BIOP消息

A. 8. 4. 1 目录

BIOP::DirectoryMessageBody 结构由一个绑定关系的循环组成。绑定把对象名（也就是bindingName）关联到IOR上去并且提供关于该对象的附加信息。当关联的对象属于该轮播时，该IOR应包含BIOP类型体。

串必须以“0x0”为结束。

BIOP目录消息是通用对象消息格式的实例。BIOP::DirectoryMessageBody 结构的句法见表A. 12。

表 A.12 BIOP::DirectoryMessage 的句法

句法	比特数	类型	值	注释
BIOP::DirectoryMessage(){				
magic	4x8	ui msbf	0x42494F50	“BIOP”
biop_version.major	8	ui msbf	0x01	BIOP 主版本 1
biop_version.minor	8	ui msbf	0x00	BIOP 副版本 0
byte_order	8	ui msbf	0x00	big_endian 字节排序
message_type	8	ui msbf	0x00	
message_size	32	ui msbf		
objectKey_length	8	ui msbf	N1	
for(i=0; i<N1; i++){				
objectKey_data_byte	8	ui msbf		
}				
objectKind_length	32	ui msbf	0x00000004	
objectKind_data	4x8	ui msbf	0x64697200	“dir” type_id 别名
objectInfo_length	16	ui msbf	N2	objectInfo
for(i=0; i<N2; i++){				
objectInfo_data_byte	8	ui msbf		
}				
serviceContextList_count	8	ui msbf	N3	serviceContextList
for(i=0; i<N3; i++){				
serviceContextList_byte	8	ui msbf		
}				
message_Body_length	32	ui msbf		
bindings_count	16	ui msbf	N4	
for(i=0; i<N4; i++){				
BIOP::Name(){				
nameComponents_count	8	ui msbf	N5	
for(i=0; i<N5; i++){				
id_length	8	ui msbf	N6	NameComponent id
for(j=0; j<N6; j++){				
id_data_byte	8	ui msbf		
}				
kind_length	8	ui msbf	N7	NameComponent kind
for(j=0; j<N7; j++){				
kind_data_byte	8	ui msbf		同 type_id, 见表 A.7
}				
}				
}				
bindingType	8	ui msbf		0x01 用于 noobject 0x02 用于 ncontext
IOP::IOR()				ObjectRef, 见表 A.6
objectInfo_length	16	ui msbf	N8	

表 A. 12 BIOP::DirectoryMessage 的句法（续）

句法	比特数	类型	值	注释
<pre>for(j=0;j&lt;N8;j++){     objectInfo_data_byte } }</pre>	8	ui msbf ui msbf		

BIOP::DirectoryMessageBody 的语义如下：

- byte\_order 域标明了消息要素(包括消息长度)使用的字节排序。值FALSE(0)表示big-endian 字节排序，值True(1)表示little-endian字节排序；
- byte\_order的值为0x00时表示后面的数据使用big-endian 字节排序来编码；
- objectKey域标识了消息中传输的对象,objectKey与该对象IOR的BIOP::objectLocation 组件中出现的objectKey相同。objectKey的值仅对广播服务器有意义，客户端不解释它。然而，客户端将把它与IOR中的objectKey进行逐字节地比较；
- objectKey长度的值应小于等于0x04；
- objectKind域标明了该消息中对象的种类。它和对象的IOR中的type\_id串是一致的，见表A. 7。objectKind的值定义了objectInfo和messageBody域的句法和语义。目录消息的objectKind应该是“dir”；
- objectInfo域含有对象一部分或全部的特性。这个域的句法和语义是依赖于objectKind域的；
- serviceContextList用来传递用户私有相关数据到对象接口上去,提供与对象接口相联系的私有数据(它的使用未在本附录中定义)。数字电视兼容接收端可以跳过ServiceContextList域；
- BindingName域(也就是id和kind)包含了对象路径详细信息(由CosNaming定义)；
- BIOP::Name，该Name仅含有一个NameComponent，因此nameComponents\_count被置为1；
- BindingType域标明了对象绑定类型。当这个名字被绑定到一个目录或业务网关对象时，绑定的类型为“ncontext”，当这个名字被绑定到非目录或业务网关对象时，绑定的类型为“nobject”。U-U对象轮播不支持BindingType为“composite”；
- ObjectRef域包含有对象的IOR；
- ObjectInfo域含有绑定对象的一些特性和关于该对象的用户私有信息。如果绑定对象的特性在这个域中，它们应是被封装在该域的第一个结构中；
- 数字电视接收端可以跳过objectInfo域中的信息。

A. 8. 4. 2 文件

FileMessageBody含有8进制流的文件数据，其句法见表A. 13。

表 A. 13 BIOP::FileMessage 的句法

句法	比特数	类型	值	注释
BIOP::FileMessage(){				
Magic	4x8	ui msbf	0x42494F50	“BIOP”
biop_version.major	8	ui msbf	0x01	BIOP 主版本 1
biop_version.minor	8	ui msbf	0x00	BIOP 副版本 0
byte_order	8	ui msbf	0x00	big-endian 字节排序
message_type	8	ui msbf	0x00	
message_size	32	ui msbf		
objectKey_length	8	ui msbf	N1	

表A.13 BIOP::FileMessage的句法(续)

句法	比特数	类型	值	注释
for(i=0; i<N1; i++){ objectKey_data_byte }	8	ui msbf		
objectKind_length	32	ui msbf	0x00000004	
objectKind_data	4x8	ui msbf	0x66696C00	“fil” type_id 别名
objectInfo_length	16	ui msbf	N2	objectInfo
DSM::File::ContentSize for(i=0; i<N2-8; i++){ objectInfo_data_byte }	64	ui msbf		
serviceContextList_count	8	ui msbf	N3	serviceContextList
for(i=0; i<N3; i++){ serviceContextList_data_byte }	8	ui msbf		
MessageBody_length	32	ui msbf	N4	实际文件内容
Content_length	32	ui msbf		
for(i=0; i<N4; i++){ Content_data_byte }	8	ui msbf		
}				

BIOP::File消息的语义和BIOP::Directory消息的语义是相同的，除了以下原则：

- objectKind域标明了在该消息中的对象种类。它和该对象的IOR中的type\_id串是相同的，见表A.7。objectKind的值定义了objectInfo域和messageBody域的句法和语义；
- 一个文件消息的objectKind的值为“fil”。

#### A.8.4.3 流

流消息的objectKind值为“str”。

BIOP::StreamMessageBody 由一系列与流对象相关的Tap构成，其句法见表A.14。

表A.14 BIOP::StreamMessage 的句法

句法	比特数	类型	值	注释
BIOP::StreamMessage(){ Magic	4x8	ui msbf	0x42494F50	“BIOP”
biop_version.major	8	ui msbf	0x01	BIOP 主版本 1
biop_version.minor	8	ui msbf	0x00	BIOP 副版本 0
byte_order	8	ui msbf	0x00	big_endian 字节排序
message_type	8	ui msbf	0x00	
message_size	32	ui msbf		
objectKey_length	8	ui msbf	N1	
for(i=0; i<N1; i++){ objectKey_data_byte	8	ui msbf		
}				

表A. 14 BIOP::StreamMessage的句法（续）

句法	比特数	类型	值	注释
objectKind_length	32	ui msbf	0x00000004	“ str ” type_id 别名
objectKind_data	32	ui msbf	0x73747200	
objectInfo_length	16	ui msbf	N6	objectInfo aDescription  AppNPT 秒 AppNPT 微秒
DSM::Stream::Info_T{ aDescription_length	8	ui msbf	N2	
for(i=0; i<N2; i++){ aDescription_bytes	8	ui msbf		
}				
duration.aSeconds	32	si msbf		
duration.aMicroSeconds	32	ui msbf		
Audio	8	ui msbf		
Video	8	ui msbf		
Data	8	ui msbf		
}				
for(i=0; i<N6-(N2+10); i++){ objectInfo_byte	8	ui msbf		serviceContextList
}				
serviceContextList_count	8	ui msbf	N3	
for(i=0; i<N3; i++){ serviceContextList_data_byte	8	ui msbf		
}				
messageBody_length	32	ui msbf		
taps_count	8	ui msbf	N4	
for(i=0; i<N4; i++){ Id	16		0x0000	
Use	16			
association_tag	16			
selector_length	8		0x00	无 selector
}				
}				

该流包含一个或多个与流对象相关联的Tap。当一个或多个Tap使用BIOP\_ES\_USE的TapUse的值时，该流由许多ES流构成，每个ES流由一个BIOP\_ES\_USE Tap标明；当仅有一个Tap使用BIOP\_PROGRAM\_USE的TapUse的值时，该流由一个节目构成，该节目由BIOP\_PROGRAM\_USE Tap标明。

指向ES流的Tap域的语义如下：

- use域标明了该Tap的使用，此域的值BIOP\_ES\_USE；
- association\_tag标明了ES流；
- selector域为空。

指向节目的tap域的语义如下：

- use域标明了Tap的使用，该域的值BIOP\_PROGRAM\_USE；

- association\_tag标明了描述该广播节目的PMT。association\_tag的值应和指向该PMT的，deferred\_association\_tags\_descriptor 中的 association\_tag 的值相同，见 A.8.7.4 的 deferred\_association\_tags\_descriptor；
- selector域为空。

注：流中的Tap可能也指向NPT、状态、事件ES流。

在BIOP::StreamMessage中允许使用的TapUse定义见表A. 15。

表 A. 15 在 BIOP::StreamMessage 中允许使用的 TapUse 定义

TapUse 域	值	所在 PID 的广播内容
STR_NPT_USE	0x000B	码流 NPT 描述符
STR_STATUS_AND_EVENT_USE	0x000C	同时有流模式和流事件描述符
STR_EVENT_USE	0x000D	流事件描述符
STR_STATUS_USE	0x000E	流模式描述符
BIOP_ES_USE	0x0018	ES 流(音/视频)
BIOP_PROGRAM_USE	0x0019	节目基准

#### A.8.4.4 业务网关

业务网关消息的句法和语义与BIOP::DirectoryMessage的句法和语义相同，除了业务网关消息的objectKind 值为“srg”的情况。

#### A.8.4.5 StreamEvent

BIOP::StreamEventMessage的句法见表A. 16。

表 A. 16 BIOP::StreamEventMessage 的句法

句法	比特数	类型	值	注释
BIOP::StreamEventMessage() {				
magic	4x8	ui msbf	0x42494F50	“BIOP”
version.major	8	ui msbf	0x01	BIOP 主版本 1
version.minor	8	ui msbf	0x00	BIOP 副版本 0
byte_order	8	ui msbf	0x00	Big_endian 字节顺序
message_type	8	ui msbf		
message_size	32	ui msbf		
objectKey_length	8	ui msbf	N1	
for (i = 0; i < N1; i++) {				
objectKey_data_byte	8	ui msbf		
}				
objectKind_length	32	ui msbf	0x00000004	
objectKind_data	4x8	ui msbf	0x73746500	“ste” type_id 别名
objectInfo_length	16	ui msbf	N6	
DSM::Stream::Info_T {				
aDescription_length	8	ui msbf	N2	aDescription
for (i = 0; i < N2; i++) {				
aDescription_bytes	8	ui msbf		见 BIOP::StreamMessage()
}				
duration.aSeconds	32	si msbf		见 BIOP::StreamMessage()



表A. 16 BIOP::StreamEventMessage的句法（续）

句法	比特数	类型	值	注释
duration.amicroSeconds	16	ui msbf		见 BIOP::StreamMessage()
audio	8	ui msbf		见 BIOP::StreamMessage()
video	8	ui msbf		见 BIOP::StreamMessage()
data	8	ui msbf		见 BIOP::StreamMessage()
}				
DSM::Event::EventList_T {				
eventNames_count	16	ui msbf	N3	
for (i = 0; i < N3; i++) {				
eventName_length	8	ui msbf	N4	
for (j = 0; j < N4; j++)				
{				
eventName_data_byte	8	ui msbf		包括零终止符
}				
}				
}				
for (i = 0; i = N6 - (N2 + 10) - (2				
+ N3 + sum(N4)); i++) {				
objectInfo_byte	8	ui msbf		
}				
serviceContextList_count	8	ui msbf	0x00	空的 serviceContextList
for (i = 0; i < N3; i++) {				
serviceContextList_data_byte	8	ui msbf		
}				
messageBody_length	32	ui msbf		
taps_count	8	ui msbf	N5	
for (i = 0; i < N5; i++) {				
id	16	ui msbf	0x0000	未定义
use	16	ui msbf		见表 A. 15
association_tag	16	ui msbf		
selector_length	8	ui msbf	0x00	没有 selector
}				
eventIds_count	8	ui msbf	N3	= eventNames_count
for (i = 0; i < N3; i++) {				
eventId	16	ui msbf		
}				
}				

StreamEvent消息的objectKind 值为 “ste”。

EnventIdList包含了与EventList\_t属性中公布的EventName相关联的eventId。EventId的序列计数应该与EventName的序列计数一致。

注：DSM-CC中的event与SI中的event不是一个概念。

A. 8. 5 下载数据轮播消息

## A.8.5.1 DownloadInfoIndication

广播网络中模块的传输参数由DownloadInfoIndication()消息来传送,见ISO/IEC 13818-6。一条DownloadInfoIndication()消息可以传送同一个U-U对象轮播的多个模块的传输参数。下面的语义规则适用于DownloadInfoIndication()消息的各个域:

- transactionId域应该与封装在对象的IOR的BIOP\_DELIVERY\_PARA\_USE Tap的selector中的transactionId取相同的值,这些对象在本消息所描述的模块中传送;
- 如果DownloadInfoIndication消息的任何域发生了变化,transactionId域将加上一个正整数成为一个新的唯一的值;
- downloadId域与DownloadDataBlock()消息的downloadId域取相同的值,这些DownloadDataBlock()消息传送DownloadInfoIndication()消息所描述的模块的块。因此,该域的值也应等于U-U对象轮播中的carouselId;
- BlockSize域包含所有DownloadDataBlock()消息的块的大小,这些DownloadDataBlock()消息传送DownloadInfoIndication()消息所描述模块的块;
- windowSize、ackPeriod、tCDownloadWindow、和 tCDownloadScenario不使用,置为零;
- compatibilityDescriptor()域不使用,长度为零;
- moduleId、moduleSize和moduleVersion的语义同ISO/IEC 13818-6 的7.3.2条;
- moduleInfoLength域为所描述的模块定义了moduleInfo域的字节长度。

ModuleInfoBytes域包含BIOP::ModuleInfo结构。BIOP::ModuleInfo结构提供了附加的传送参数和用于在网络中广播这些模块的Tap。其句法见表A.17。

表 A.17 BIOP::ModuleInfoMessage 的句法

句法	比特数	类型	值	注释
BIOP::ModuleInfo() { ModuleTimeout BlockTimeout MinBlockTime taps_count for (j = 0; j < N1; j++) {	32 32 32 8	ui msbf ui msbf ui msbf ui msbf	N1	
Id	16	ui msbf	0x0000	用户私有
Use	16	ui msbf	0x0017	BIOP_OBJECT_USE
association_tag	16	ui msbf		
selector_length	8	ui msbf	0x00	
}				
UserInfoLength	8	ui msbf	N2	
for (j = 0; j < N2; j++){				
userInfo_data_byte	8	ui msbf		(包括零终止符)
}				
}				

BIOP::ModuleInfo的语义如下:

- ModuleTimeout域给出了以 $\mu s$ 为单位的超时时限,可以用来判断获取一个模块的全部块是否超时;
- BlockTimeout域给出了以 $\mu s$ 为单位的超时时限,可以用来判断在已经获得一个块的情况下,获取下一个块是否超时;

- MinBlockTime域标明传送所描述的模块两个相邻块的最小时间间隔,客户端可利用该值来调整接收程序,以达到优化的目的。

BIOP::ModuleInfo的Tap域至少包含一个TapUse的值为BIOP\_OBJECT\_USE的Tap。该Tap指向广播这些模块的网络连接。其各域的语义在A.8.2.5.中描述。

将BIOP::ModuleInfo的userInfo域组织成一个描述符的循环,以便能够使用数字电视数据轮播中定义的模块描述符。

接收端尤其要支持compressed\_module\_descriptor (标签为0x09),该描述符表明模块以压缩的形式传输。

privateDataLength和privateDataByte域的使用在本附录中不定义。

数字电视接收端应该能够跳过私有数据域。

A.8.5.2 DownloadServerInitiate

业务网关的IOR通过DownloadServerInitiate()消息进行广播。

下面的语义规则适用于消息DownloadServerInitiate()的各个域:

- serverId域是用0xFF填充的20个字节。Carousel Specifier在下面定义;
- compatibilityDescriptor()域不使用,长度为零;
- privateDataLength域定义后面的privateDataByte域的字节长度;
- DownloadServerInitiate()消息的privateDataByte域中的数据要包含BIOP::ServiceGatewayInfo结构。该结构的句法见表A.18。

表 A.18 ServiceGatewayInfo()的句法

句法	比特数	类型	值	注释
ServiceGatewayInfo () { IOP::IOR() downloadTaps_count for (i = 0; i < N1; i++) { Tap() } serviceContextList_count for (i = 0; i < N2; i++) { serviceContextList_data_byte } userInfoLength for (i = 0; i < N3; i++) { userInfo_data_byte } }	8	ui msbf	N1	见表 A.6 软件下载Tap
	8	ui msbf		
	8	ui msbf	N2	ServiceContextList
	8	ui msbf		
	16	ui msbf	N3	用户信息
	8	ui msbf		

BIOP::ServiceGatewayInfo的语义如下:

- objectRef域包含ServiceGateway的IOR;
- Tap域与serviceContextList的语义在本附录中不定义;
- 将userInfo域组织成一个描述符的循环。循环中的描述符要么是数字电视数据广播规范定义的描述符,要么是私有描述符。

A.8.5.3 DownloadDataBlock

DownloadData消息在ISO/IEC 13818-6中定义。各域的使用在ISO/IEC 13818-6中定义。

A.8.6 段

DSM-CC用private\_section将TS包重组成DSM-CC信息,DSM-CC为private\_section定义了附加的语义以支持DSM-CC的额外需要。由此得到的section称为DSMCC\_section,其结构与private\_section的句法兼容,以便GB/T 17975.1系统解码器可用。DSM-CC\_section的句法在ISO/IEC 13818-6中定义。

table\_id\_extension、version\_number、section\_number和last\_section\_number的编码在表A.19中定义。

表 A.19 DSM-CC\_section 域编码

消息	table_id	table_id_extension	version_number	section_number	last_section_number
Download-ServerInitiate (DSI)	0x3B	DSI 中 transaction_id 的两个 LSB 字节	0x00	0x00	0x00
Download-InfoIndication (DII)	0x3B	DII 中 transaction_id 的两个 LSB 字节	0x00	0x00	0x00
Download-DataBlock (DDB)	0x3C	moduleId	Module Version % 32	BlockNumber % 256	Max(section_number)

DownloadServerInitiate消息中transaction\_id的最低两个有效字节取值范围为0x0000 ~ 0x0001。

DownloadInfoIndication消息中transaction\_id的最低两个有效字节取值范围为0x0002 ~ 0xFFFF。

数字电视对关于transaction\_id域的DSM-CC规范给出了一些限制,以方便客户接收端的过滤操作。特别的,DSI消息的最低两个有效字节为0x0000或0x0001。这使得接收端通过设置table\_id = 0x3B (DownloadControlMessages)与table\_id\_extension = 0x0000或0x0001的section过滤器,来自动引导轮播。一旦DSI消息被获取,接收端可以设置section过滤器来监测transaction\_id的最低两个有效字节的其它值。这样轮播内容一旦改变,接收端能立即被触发。

#### A.8.7 PSI描述符的使用

ISO/IEC 13818-6的对象轮播规范与网络无关,因此可以应用到各种广播网络中。网络独立性通过Tap概念的使用来获得。通过使用关联标签,一个Tap可以方便地与某个特定的网络连接关联起来。在对象解析的过程中,客户应将Tap与网络连接关联起来。因此,客户需要一个关联表,该表确定了各Tap与网络连接的对应关系。

基于TS流的广播网络中实现U-U对象轮播,PSI机制有利于:

- 将节目(即利用PMT表)和对象轮播关联起来;
- 将Tap与某个PID或节目关联起来;
- 用于广播业务网关的IOR的PID的定位;
- 一个对象轮播在多个节目上的分布式实现。

这一节对以下GB/T 17975.1中的描述符的使用做了说明,它们提供了以上功能,见ISO/IEC 13818-6。

##### A.8.7.1 CarouselIdentifier\_Descriptor

轮播标识符描述符建立了节目与对象轮播的关联关系。CarouselIdentifierDescriptor的句法见表A.20,也可参见ISO/IEC 13818-6。

通过这种可选的机制,无须事先加载DownloadServerInitiate和DownloadIndicationInformation消息就可以获取ServiceDomain的ServiceGateway。

表 A.20 carousel\_identifier\_descriptor 的句法

句法	比特数	类型	值	注释
carousel_i d e n t i f i e r _ d e s c r i p t o r () {				
d e s c r i p t o r _ t a g	8	u i m s b f	0x13	
d e s c r i p t o r _ l e n g t h	8	u i m s b f		
c a r o u s e l _ i d	32	u i m s b f		
F o r m a t I d	8	u i m s b f		FormatSpeci fier 的注册的标识符
F o r m a t S p e c i f i e r () {				
F o r m a t S p e c i f i e r _ b y t e	8	u i m s b f		见表 A. 21 N2 字节
}				
f o r ( i = 0 ; i < N1 ; i ++ ) {				
p r i v a t e _ d a t a _ b y t e	8	u i m s b f		
}				
}				

carousel\_identifier\_descriptor()描述符应该插入到PMT表第二个描述符循环(ES\_info)中,对应传送对象轮播的DSI的ES流。这样每个节目可以包含多个对象轮播,每个轮播启动的PID也相应确定。

为了支持carousel NSAddress的使用,描述符carousel\_identifier\_descriptor()同样是必需的,例如在LiteOptions类型体关联的确定过程中。

FormatId用于标识描述符私有数据域的FormatSpecifier的格式。其句法见表A. 21。

表 A.21 描述符 carousel\_identifier\_descriptor 中的 FormatSpecifier 的句法

FormatId值	Format Specifier 定义	长度 (比特)	注释
0x00	没有FormatSpecifier		值0x00标明FormatSpecifier缺少。因此，ServiceGateway的位置只可能通过“标准”的方式解释DSI和DII消息。
0x01	FormatSpecifier{ ModuleVersion ModuleId BlockSize ModuleSize CompressionMethod OriginalSize Timeout ObjectKeyLength for (i = 0; i < N1; i++){ ObjectKeyData } }		此 FormatSpecifier 是 定位 ServiceGateway所必需的域的集合，在 DSI和DII消息中也可以找到。
		8	
		16	
		16	
		32	所有域的类型都是“ui msbf”。
		8	
		32	
		8	以秒为单位的超时时限
		8	
		8	
0x02...0x7F	为将来使用保留		formatId的值从0x02到0x7F为数字电视系统将来使用保留

表 A.21 描述符 carousel\_identifier\_descriptor 中的 FormatSpecifier 的句法 (续)

FormatId值	Format Specifier 定义	长度 (比特)	注释
0x80...0xFF	为私有使用保留		formatId的值从0x80到0xFF为用户私有使用保留

FormatId为0x01表明FormatSpecifier中包含了可以用来定位对象轮播的ServiceGateway的信息(在消息DSI和DII中同样可找到)。支持FormatID可能会影响到广播服务器,因为这些信息应与ServiceGateway对象和传递它的模块的变化保持一致。

FormatId等于0x01的FormatSpecifier表明DSI消息与包含ServiceGateway的模块是用同一个PID传送的。

#### A.8.7.2 association\_tag\_descriptor

association\_tag\_descriptor建立了association\_tag与PID间的关联关系,见ISO/IEC 13818-6,因此与SI的stream\_identifier\_descriptor相似。但association\_tag\_descriptor使用16位的association\_tag(而在描述符stream\_identifier\_descriptor中使用的是8比特的component\_tag),这有利于广播ServiceGateway的PID的识别。接收端因此能够从PMT表大量的PID中有效地过滤出对象轮播。为了给PID标注一个特定的association\_tag值,服务器应该向该PID的描述符循环中插入association\_tag。

association\_tag\_descriptor的句法见表A.22。

表 A.22 association\_tag\_descriptor 的句法

句法	比特数	类型	值	注释
association_tag_descriptor () { descriptor_tag descriptor_length association_tag	8 8 16	ui msbf ui msbf ui msbf	0x14	
use	16	ui msbf	0x0000 0x0100-0x1FFF 0x2000-0xFFFF	具有 SGW 中 IOR 的 DSI 数字电视系统保留 用户私有
if (use == 0x0000) { selector_length transaction_id timeout }	8 32 32	ui msbf ui msbf ui msbf	0x08	DSI 的 transaction_id 为 DSI 定义的超时时限
else if (use == 0x0001) selector_length }	8	ui msbf	0x00	
else { selector_length for (i = 0; i < N1; i++) { selector_byte } } for (i = 0; i < N2; i++) {	8 8	ui msbf ui msbf	N1	

表 A. 22 association\_tag\_descriptor 的句法 (续)

句法	比特数	类型	值	注释
private_data_byte } }	8	ui msbf		私有数据

use域能指出PID的用途，同时规范了selector域的句法和语义。如果use域的值等于0x0000，则传送业务网关的IOR的DownloadServerInitiate消息是用这个PID广播的。在这种情况下，selector\_byte域的数据将包含transaction\_id和timeout的值。

transaction\_id和timeout的语义如下：

- transaction\_id的值应该对应于DownloadServerInitiate()消息中的transaction\_id,该消息传送U-U对象轮播的业务网关的IOR。例外的是当association\_tag\_descriptor中的transaction\_id取值为0xFFFFFFFF时，这表明DownloadServerInitiate()消息的transaction\_id目前未知，但是所有已标识的PID广播的DownloadServerInitiate()消息都是有效的。当DownloadServerInitiate()消息的内容允许改变(因此消息中的transaction\_id改变)，而没有必要更新包含association\_tag\_descriptor的PMT表时，transaction\_id可以取值0xFFFFFFFF；
- Timeout域给出以μs为单位的超时时限，用来判断DownloadServerInitiate()消息的获取是否超时。一个特殊的超时值(0xFFFFFFFF)意味着目前尚无已知的超时的值。同上面一样，允许“静态的”PMT表；
- use域的缺省值应为0x0100。这表明相关联的PID能或不能用来广播一条DSI消息；
- 数字电视保留use域中从0x0101到0x01FF的取值，以备将来使用。

A. 8. 7. 3 stream\_identifier\_descriptor

stream\_identifier\_descriptor有效建立了component\_tag与PID的关联，可以替代(或结合)association\_tag\_descriptor来使用。但是由于stream\_identifier\_descriptor的component\_tag域只有8位，因此在component\_tags和association\_tags之间进行映射是必要的。

某个PID描述符循环中的stream\_identifier\_descriptor应该与该PID的association\_tag\_descriptor相同,其association\_tag最低有效字节LSB = <component\_tag>，use域的值为0x0100。

注：这种匹配机制为一个对象轮播在多个ES上的分布实现提供了灵活性，并且在不同PMT表中仍然用同一个component\_tag值来指向这个特定的数据广播业务。

A. 8. 7. 4 deferred\_association\_tags\_descriptor

一个对象轮播可以使用多个PID、多个业务、甚至多个TS流来广播其对象和相应的控制信息。为便于客户端确定该对象轮播在不同节目里使用的所有association\_tag，定义了一个描述符，它可以插入到实现这个对象轮播的所有节目的PMT表的第一个描述符循环中。deferred\_association\_tags\_descriptor包含了对象轮播中使用的所有与描述符本身所在PMT的PID不相关联的association\_tag。因此，deferred\_association\_tags\_descriptor包含了对节目的前向引用，该节目包含有association\_tag所链接的PID。如果需要，多个deferred\_association\_tags\_descriptor可以插入到一个PMT表中。

而且由于BIOP\_PROGRAM\_USE Tap的存在，deferred\_association\_tag\_descriptor还可以用来指向另一个数字电视业务(节目)。

只要一个对象轮播使用多个业务，就应使用deferred\_association\_tag\_descriptor。对每个只传送一部分对象轮播内容的业务，association\_tag的列表应是完全的，以避免association\_tag映射失败或出错。

deferred\_association\_tags\_descriptor()的句法和语义见表A.23：

表 A.23 deferred\_association\_tags\_descriptor 的句法

句法	比特数	类型	值	注释
deferred_association_tags_descriptor () {				
descriptor_tag	8	ui msbf	0x15	
descriptor_length	8	ui msbf		
association_tags_loop_length	8	ui msbf	2 × N1	以字节为单位的长度
for (n = 0; n < N1 ; n++) {				
association_tag	16	ui msbf		
}				
transport_stream_id	16	ui msbf		
program_number	16	ui msbf		
org_network_id	16	ui msbf		
for (n = 0; n < N ; n++) {				
Private_data_byte	8	ui msbf		
}				
}				

#### A.8.8 SI和PSI中的信息

为表明使用数字电视对象轮播，data\_broadcast\_id应该置为0x0007。

注：如果对象轮播(use域)是已经注册了data\_broadcast\_id的规范的组成部分，这个id值(selector域句法正确)可以替代0x0007使用。

##### A.8.8.1 SI描述符

具有上述值的SI的data\_broadcast\_descriptor可以用来表明一个业务中使用数字电视对象轮播。

这种情况下，data\_broadcast\_descriptor的selector域包含了对象名称的循环，使对象轮播中的应用能够自动引导。循环还包含了ISO\_639\_language\_code域，可以用来基于首选的语言启动应用。

data\_broadcast\_descriptor使用的对象名称应存在于对象轮播中。

##### A.8.8.2 PSI的描述符

data\_broadcast\_descriptor的使用方式与数据轮播类似，见A.7.7.1。

#### A.8.9 transactionId值的分配与使用

对象轮播中transactionId的使用方法是从小SM-CC规范的定义继承下来的，因此显得有点复杂。TransactionId具有双重功能，它提供了控制消息（即DownloadInfoIndication和DownloadServerInitiate）的识别和版本机制。TransactionId应该唯一标识数据轮播中的下载控制消息，但是消息的任何字节一旦改变，它都将被“增量”。

对象轮播在数据轮播之上传送，而且可以由多个数据轮播分发。对象轮播之下的数据轮播，在本附录中是指一系列用一个PID传送的DownloadInfoIndication消息以及传送该消息描述的模块的DownloadDataBlock消息。与DownloadInfoIndication不同的是，DownloadDataBlock消息可以分布到其它ES中。在对象轮播中，消息DownloadServerInitiate被看作是对象轮播的最上层的一部分，与任何数据轮播不发生关联。

当一个模块改变，其版本号需要更新。这意味着引用该模块的DownloadInfoIndication消息也需要更新。由于DownloadInfoIndication消息更新了，transactionId也需要改变。然而，DownloadInfoIndication消息的transactionId还同时在别的消息中使用，但特别需要避免改变其它消息，由模块更新带来的影响应该限制在该模块以及引用它的DownloadInfoIndication消息中。因此，下面说明了使用transactionId的附加规则。



TransactionId被分解成许多表A. 24中定义的子域。这反映了transactionId的双重功能(上文已列出)以及为减小模块更新的影响所强加的限制。但是,为增强互操作性,transactionId的分配设计成与目标接收端的预期过滤无关。

表 A. 24 transactionId 的子域

位数	值	子域	描述
0	用户定义	更新标志	每次控制消息更新都会触发这一比特。
1 ~ 15	用户定义	标识	对于 DownloadServerInitiate 消息,这几比特只能都为零。 所有其他的控制消息应该有一位或更多的位的非零比特。
16 ~ 29	用户定义	版本	这几个比特每当控制消息更新的时候都会增量/改变。
30 ~ 31	第 30 比特为零 第 31 比特非零	生成者	当transactionId为网络分配时,这两比特在ISO/IEC 13818-6中被定义为0x02 – 在广播的操作中是隐含的。

由于transactionId具有版本机制的功能,控制消息的任何变化都将导致其transactionId增量。模块发生任何变化都应增量其moduleVersion域的值。这一变化将反映在描述该模块的DownloadInfoIndication消息的对应域上。既然DownloadInfoIndication消息的域改变了,其transactionId应该增量,以表示一个新版本的消息。

同样,DownloadServerInitiate消息的任何改变也意味着transactionId的增量。然而,当transactionId被分解成上述子域以后,消息的更新只需改变transactionId的版本部分,而其识别部分保持不变。

由于transactionId也用于在其它结构中引用消息时识别消息,自然很希望这些引用的消息不要在控制消息每次更新时都随之更新。所以按照关联关系定位消息时应该使用这样的规则,即当根据引用消息的transactionId定位一条消息时,只有识别部分(比特1...15)需要匹配。

运用这一规则,更新一个模块的影响可以限制在模块本身以及描述该模块的DownloadInfoIndication消息中。同时,这意味着如果接收端只需要过滤出描述该模块的DownloadInfoIndication消息并检查其是否改变,就可以判定它接收过的模块当前是否发生了变化。

附 录 B  
( 规范性附录 )  
数据轮播的 DSM-CC 消息

本附录包含了DSM-CC下载消息的句法。每个域的语义描述标明了按照本附录实现一个数据轮播时可能使用这些域的地方。

### B.1 dsmccMessageHeader

dsmccMessageHeader的句法见表B. 1。

表 B. 1 dsmccMessageHeader 的句法

句法	字节数
dsmccMessageHeader(){	
protocolDiscriminator	1
dsmccType	1
messageId	2
transactionId	4
reserved	1
adaptationLength	1
messageLength	2
if ( AdaptationLength > 0 ) {	
dsmccAdaptationHeader()	
}	
}	

dsmccMessageHeader的语义如下：

- protocolDiscriminator域标明本消息是DSM-CC消息，其值为0x11；
- dsmccType域标明DSM-CC消息的类型，取值0x03标明该消息是U-N下载消息；
- messageId域标明正在传送的消息的类型，其值在dsmccType范围内定义；
- transactionId用于对话完整性和出错处理，在一段时间内应该保持唯一性，以确保命令序列不发生冲突。transactionId只在本地有效，也就是说其值应该由广播服务器选定；
- reserved域由ISO/IEC 13818-6保留，置为0xFF；
- adaptationLength标明adaptation header的字节总长度；
- messageLength标明接下来的消息的字节总长度。这个长度包括adaptationLength给出的任何adaptation header和MessageId给出的消息载荷。

### B.2 dsmccDownloadDataHeader

dsmccDownloadDataHeader的句法见表B. 2。

表 B.2 dsmccDownloadDataHeader 的句法

句法	字节数
dsmccDownloadDataHeader(){ protocolDiscriminator dsmccType messageId downloaded reserved adaptationLength messageLength if ( AdaptationLength > 0 ){ dsmccAdaptationHeader() } }	 1 1 2 4 1 1 2

dsmccDownloadDataHeader的语义如下：

- protocolDiscriminator域标明本消息是DSM-CC消息，其值为0x11；
- dsmccType域标明DSM-CC消息的类型，取值0x03表明该消息是一条U-N下载消息；
- messageId域给出正在传送的消息的类型，其值在dsmccType范围内定义；
- downloadId域把下载数据消息与下载场景的一个实例的下载控制消息关联起来；
- reserved域由ISO/IEC 13818-6保留，置为0xFF；
- adaptationLength标明adaptation header的字节总长度；
- messageLength标明接下来的消息的字节总长度。这个长度包括adaptationLength给出的任何adaptation header和MessageId给出的消息载荷。

B.3 DownloadServerInitiate

DownloadServerInitiate消息的句法见表B.3。

表 B.3 DownloadServerInitiate 消息的句法

句法	字节数
DownloadServerInitiate(){ dsmccMessageHeader() serverId compatibilityDescriptor() privateDataLength for (i=0; i<privateDataLength; i++){ privateDataByte } }	 20  2 1

DownloadServerInitiate消息的语义如下：

- serverId域置为20个字节的0xFF(也就是不使用该域)；
- compatibilityDescriptor() 结构只包含DSM-CC定义的compatibilityDescriptor的compatibilityDescriptorLength域，置为0x0000(也就是不使用该域)；

- privateDataLength定义了后面的私有字节的长度；
- privateDataByte域传送本标准定义的GroupInfoIndication结构。

#### B.4 DownloadInfoIndication

DownloadInfoIndication消息的句法见表B.4。

表 B.4 DownloadInfoIndication 消息的句法

句法	字节数
DownloadInfoIndication(){	
dsMccMessageHeader()	
downloadId	4
blockSize	2
windowSize	1
ackPeriod	1
tCDownloadWindow	4
tCDownloadScenario	4
compatibilityDescriptor()	
numberOfModules	2
for (i=0; i<numberOfModules; i++){	
moduleId	2
moduleSize	4
moduleVersion	1
moduleInfoLength	1
for (i=0; i<moduleInfoLength; i++){	
moduleInfoByte	1
}	
}	
privateDataLength	2
for (i=0; i<privateDataLength; i++){	
privateDataByte	1
}	
}	

DownloadInfoIndication消息的语义如下：

- downloadId是进程中下载操作的标识符。对于数据轮播的操作，downloadId应该在网络中唯一定义；对于流控制或无流控制的操作，downloadId应该在连接中唯一定义。这一标识符将在进程中的下载操作所使用的DownloadDataBlock、DownloadDataRequest和DownloadCancel消息中使用；
- blockSize是DownloadDataBlock消息中传输的每个块的字节长度，而每个模块的最后一个块的长度可以小于blockSize；
- windowSize在数据轮播中不使用，置为0；
- ackPeriod在数据轮播中不使用，置为0；
- tCDownloadWindow在数据轮播中不使用，置为0；
- tCDownloadScenario标明进程中整个下载操作以 $\mu s$ 为单位的超时时限；

- compatibilityDescriptor() 结构只包含 DSM-CC 定义的 compatibilityDescriptor 的 compatibilityDescriptorLength域。置为0x0000(也就是不使用该域)；
  - NumberOfModules是此域后面的循环中描述的模块的个数。对于流控制和无流控制的下载操作，该循环描述了客户端要下载的所有模块。对于数据轮播的操作，虽然循环可以描述与数据轮播相关的所有的模块，但它描述的只是所有模块的一个子集；
  - moduleId是模块的标识，模块由域moduleSize、moduleVersion和moduleInfoByte描述。moduleId在每个downloadId范围内是唯一的；
  - moduleSize是所描述模块的字节长度；
  - moduleVersion是所描述模块的版本；
  - moduleInfoLength定义了所描述模块moduleInfo域的字节长度；
  - moduleInfoByte域传送描述符的列表。每个描述符定义相应模块的一个或多个属性；
- 注：在本标准中，moduleId不使用从0xFFFF0到0xFFFFF范围的值。
- privateDataLength定义后面的私有数据的字节长度；
  - privateDataByte域由用户定义。

B.5 DownloadDataBlock

DownloadDataBlock的句法见表B.5：

表 B.5 DownloadDataBlock 的句法

句法	字节数
DownloadDataBlock(){ dsmccDownloadDataHeader() moduleId moduleVersion reserved blockNumber for (i=0; i<N; i++){ BlockDataByte } }	  2 1 1 2  1

DownloadDataBlock的语义如下：

- moduleId给出该块所属的模块；
- moduleVersion给出块所属模块的版本号；
- reserved域由ISO/IEC 13818-6保留，置为0xFF；
- blockNumber给出块在模块中的位置。块0为模块的第一个块；
- blockDataByte传送该块的数据。

B.6 DownloadCancel

DownloadCancel 消息的句法见表B.6。

表 B.6 DownloadCancel 消息的句法

句法	字节数
DownloadCancel () {	
dsmccMessageHeader ()	
downloadId	4
moduleId	2
blockNumber	2
downloadCancel Reason	1
reserved	1
privateDataLength	2
for (i=0; i<privateDataLength; i++){	
privateDataByte	1
}	
}	

DownloadCancel 消息的语义如下：

- downloadId是某个正在进行的特定下载实例的标识。它将这个DownloadCancel 消息与一个正在进行的特定下载或数据轮播关联起来；
- moduleId和blockNumber标明取消下载时已经处理过的最后一条DownloadDataBlock消息。如果没有块被处理，这些域应该被置为0；
- downloadCancel Reason包含了取消下载的原因代码；
- reserved域由ISO/IEC 13818-6保留，设置为0xFF；
- privateDataLength定义后面的私有数据的字节长度；
- privateDataByte域的使用在数据轮播中不作规定，可以用来传送私有信息。

附 录 C  
( 规范性附录 )

将 DSM-CC 消息封装到 GB/T 17975.1 段中

本附录说明了DSM-CC消息是怎样被封装到GB/T 17975.1段中的。请参考DSM-CC规范中的准确语义。

当DSM-CC下载消息封装到TS流时，使用DSMCC\_section句法结构。该结构继承了GB/T 17975.1中定义的private\_section的所有句法。特定语义用于DSMCC\_section头中的特定域的编码。DSMCC\_section到TS流包的映射及DSMCC\_section的最大长度都符合GB/T 17975.1中定义的private\_section的语义。

在某些实现中，希望使用private\_sections中的CRC\_32。由于某些系统计算CRC\_32可能有困难，DSMCC\_section句法定义了一种CRC\_32的替代方式。为了与GB/T 17975.1保持一致，如果section\_syntax\_indicator置为“1”，CRC\_32应该存在并且正确。section\_syntax\_indicator置“0”时，该段的句法与section\_syntax\_indicator置“1”时是一样的，只是CRC\_32域被校验和域取代。这样句法仍然与GB/T 17975.1兼容，因为跟在section\_length域之后的载荷会被当成私有数据看待。

由于 section\_syntax\_indicator 本身可能出错，private\_indicator 的值应该为 section\_syntax\_indicator 的补数。如果section\_syntax\_indicator为“0”，private\_indicator被置为“1”，如果不是这样，表明这个段出错了。类似的，如果section\_syntax\_indicator为“1”，则 private\_indicator应为“0”。

当section\_syntax\_indicator为“0”(不使用CRC)且和校验域也设置为0时，将在不同层上提供另一种形式的出错检测。这种是强制性要求，以确保本文档对DSMCC\_section传输协议的最小需求。

在TS流中传送private\_section(也即DSMCC\_section)的有关句法和语义，请参考GB/T 17975.1 中 2.4.4的PSI。其中包括对payload\_unit\_start\_indicator的设置，TS包载荷中pointer\_field的存在，以及包填充字节的使用。

除非特别限制，如果table\_id解析完成的话，DSM-CC表(也就是一个或多个带有相同table\_id的DSMCC\_section)可以与其它private\_section格式的表使用相同的PID封装到TS包中。

当在TS流中传送DownloadDataBlock消息时，只有downloadId相同的DownloadDataBlock消息才能封装到PID相同的TS包。这意味着，每个PID只能传送一个数据轮播的下载数据消息。但对下载控制消息不存在这种限制，允许多个数据轮播的下载控制消息在同一个ES流中传送。在这些情况下，要获得预期的效果，应使用SI的data\_broadcast\_descriptor来明确标明一个特定的最高等级控制消息的transactionId。

DSM-CC段格式的句法见表C.1。

表 C.1 DSM-CC 段格式的句法

句法	比特数	类型
DSMCC_section(){		
table_id	8	ui msbf
section_syntax_indicator	1	bsl bf
private_indicator	1	bsl bf
reserved	2	bsl bf
dsmcc_section_length	12	ui msbf
table_id_extension	16	ui msbf
reserved	2	bsl bf
version_number	5	ui msbf

表C.1 DSM-CC段格式的句法(续)

句法	比特数	类型
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
if(table_id==0x3A){ LLCSNAP() }		
else if(table_id==0x3B){ userNetworkMessage() }		
else if(table_id==0x3C){ downloadDataMessage() }		
else if(table_id==0x3D){ DSMCC_descriptor_list() }		
else if(table_id==0x3E){ for(i=0; i<DSMCC_section_length-9; i++){ private_data_byte } }		
if(section_syntax_indicator=="0"){ checksum }	32	uimsbf
else{ CRC_32 }	32	rpchof
}		
注1:DownloadServerInitiate、DownloadInfoIndication、DownloadCancel 在 userNetworkMessage 类中。		
注2:DownloadDataBlock 在 downloadMessage 类中。		



## 附录 D

### (规范性附录)

#### 目录中的对象命名

在DSM-CC中,目录对象提供了树状等级目录结构。每个目录对象都可以包含对其它目录(例如子目录)和对象的关联关系。当对象被绑定到目录时,将有一个名称字符串分配给它,用于在该目录内唯一标识该对象。业务网关对象是目录等级图的根目录。由根目录业务网关名称、经过的所有子目录名称和最后到达的对象的名称组成的一条路径,唯一标识了一个对象轮播中的某个对象。一个对象可能被绑定到多个目录,因此可以有多个路径指向同一个对象。

下面给出了分隔目录名称与对象名称的约定:

- 用反斜杠“/”来分隔目录名称和对象名称;
- 反斜杠“/”不能作为名称的一部分。

注:不需要使用相对路径,因此也不必要有其它约定。

由于各种原因,DSM-CC与对象轮播使用了大量略有差别的数据结构在不同的环境中存储路径。因此,在不同的环境中,尽管存储目录可能指向同一个路径,也就是在不同数据结构中的名称字符串相同,但是它们的数据结构可能不同。

#### D.1 DSM-CC的U-U API 中名称使用的数据结构

U-U API 为不同环境中的目录提供了两种不同的数据结构。原因是DSM-CC的目录对象是从CORBA的CosNaming::NamingContext对象继承的。然而由于DSM-CC为它增添了一些功能,为了这些额外的功能,应定义独立的数据结构来传递路径信息。

CosNaming::NameComponent是所有名称数据结构的基础。它代表了对象完整路径名称的一部分,比如在子目录中的名称。NameComponent结构包含两个域:id和kind。id域包含确切的名称字符串;kind域包含对象的类型。

CosNaming::Name是NameComponent的序列,代表整个路径。但是,这个结构通常只表示从其所在目录开始的相对路径。当它在根目录业务网关中使用时,自然表示完整的绝对路径。

在某些用途中,Name包含在叫做CosNaming::Binding的结构中。除了Name,该结构还包含一个BindingType域。BindingType的作用是为Name所指对象分类。

DSM-CC为路径定义了另外一个数据结构DSM::PathSpec。PathSpec由DSM::Step结构序列构成。Step包含了CosNaming::Name中使用的相同的NameComponent,还有一个额外的处理标记,用于在某些功能中标明该操作是否应用到这部分路径。在DSM-CC中使用PathSpec时,通常还有另外一个PathType参数。PathType给出解释PathSpec的方式。它区分了使用PathSpec的两种不同方式。当PathType为DEPTH,PathSpec的意义与Name等同,例如它表示从当前目录沿树状图往下的相对路径。然而,当PathType为BREADTH,PathSpec中的NameComponents用来标识同一目录下的多个不同的对象。

#### D.2 对象轮播中名称使用的数据结构

为优化传输,对象轮播使用与U-U API 略有不同的数据结构。但是,这些数据结构被设计成与API中使用的结构等同。

BIOP::NameComponent与CosNaming::NameComponent等同,但增添了字符串的最大长度以优化编码。

BIOP::Name与CosNaming::Name等同,但定义了其中NameComponent个数的上限以优化编码。

对象轮播的DirectoryMessage提供了实现目录对象的必要信息。目录信息包含BIOP::Binding,BIOP::Binding中包括了从本目录到对象的路径的名称;目录信息还包含了IOR(Interoperable Object

Reference互操作对象基准)，其中给出了定位一个确定对象的必要信息。BIOP::Binding不同于CosNaming::Binding，前者包含了对对象的引用，而后者没有。这是因为，在对象轮播中可以用它来传送对象的位置，而在API中对象的定位属于目录对象的内部事务，在应用程序中是不可见的。

### D.3 对象轮播中的CORBA串

在许多地方对象轮播消息中包括文本串。这些串都遵循CORBA V2.0的12.3.2给出的格式。例如每个串前面都加上了一个声明串长度的32比特整数，结尾加上了一个空字符。通常，这可以从下面的句法表中明显看出来。但是，为清楚起见，表D.1使用的是CORBA格式字符串。

表 D.1 CORBA 格式字符串的定位

字符串	位置
objectKind_data	BIOP::FileMessage 句法
objectKind_data, id_data, kind_data	BIOP::DirectoryMessage 句法
objectKind_data	BIOP::StreamMessage 句法
objectKind_data, eventName_data	BIOP::StreamEventMessage 句法
type_id_byte	BIOP::IOR 句法
id_data, kind_data	带有 ServiceLocation 的 Options 类型体的句法

附 录 E  
( 资料性附录 )  
文本字符编码

文本条款可能包含某些选择信息 , 用来选择较大范围的字符编码表 ( 如下文 ) 。如果文本条款中没有字符编码选择信息 , 则认为使用缺省的字符编码集。GB 13000.1 的基本多文种页结构见图 E. 1。

E.1 控制码

范围在 0x80 到 0x9F 之间单字节控制码的分配见表 E. 1。

表 E. 1 单字节控制码

控制码	描述
0x80 至 0x85	预留使用
0x86	字符强调开
0x87	字符强调关
0x88 至 0x89	预留使用
0x8A	CR/LF
0x8B 至 0x9F	用户定义

对于双字节的字符表 , 本附录在 GB 13000.1 专用区 0xE080 到 0xE09F 范围内定义的控制码的分配见表 E. 2。

表 E. 2 双字节控制码

控制码	描述
0xE080 至 0xE085	预留使用
0xE086	字符强调开
0xE087	字符强调关
0xE088 至 0xE089	预留使用
0xE08A	CR/LF
0xE08B 至 0xE09F	预留使用

E.2 字符表的选择

文本域可以用非空格、不可显示的数据开始 , 该数据指定文本条目的剩余部分使用另外一个字符表。字符表的选择如下所述 :

- 如果文本域第一个字节的数值在 0x20 到 0xFF 之间 , 那么文本条目的这个字节和后续的所有字节都使用默认的字符编码表 ( 表 00—拉丁字符 ) , 见图 E. 1 ;

- 如果文本域第一个字节的数值在0x01到0x05之间，那么文本条目的剩余部分的编码见GB/T 15273.1；
- 如果文本域第一个字节的数值是0x10，那么接下来的两个字节所携带的16位数值N(ui msbf)表示了文本条目剩余部分数据使用GB/T 15273.1定义的字符编码表进行编码；
- 如果文本域第一个字节的数值是0x11，那么文本条目剩余部分的编码根据GB 13000.1 定义的基本多语言平面进行双编码；
- 如果文本域第一个字节的数值是0x12，表示文本条目中的剩余字节按照韩国字符集KSC 5601编码；
- 如果文本域第一个字节的数值是0x13，表示文本条目中的剩余字节按照GB 2312及其扩展集进行编码；
- 对于文本条目的第一个字节，值0x00、0x06至0x0F、0x14至0x1F预留使用。

Row-octet

00		Basic Latin 基本拉丁文		Latin-1 Supplement 拉丁文-1 补充
01	Latin Extended-A 拉丁文扩充-A			Latin Extended-B 拉丁文扩充-B
02	Latin Extended-B 拉丁文扩充-B	IPA Extensions 国际音标扩充	Spacing Modifier Letters 进格的修饰字符	
03	Combining Diacritical Marks 组合用发音		Basic Greek 基本希腊文	Greek Symbols and Coptic 希腊符号及哥普特文
04	Cyrillic 西里尔文			
05		Armenian 亚美尼亚文	Hebrew (Basic and Extended) 希伯来文(基本和扩充)	
06	Basic Arabic 基本阿拉伯文		Arabic Extended 阿拉伯文扩充	
09	Devanagari 天成文书(梵文)		Bengali 孟加拉文	
0A	Gurmukhi 锡克教文		Gujarati 古吉拉特文	
0B	Oriya 奥利雅文		Tamil 泰米尔文	
0C	Telugu 泰卢固文		Kannada 卡纳达文	
0D	Malayalam 德拉维族文			
0E	Thai 泰文		Lao 老挝文	
10			Georgian 格鲁吉亚文	
11	Hangul Jamo 朝鲜文字母			
1E	Latin Extended Additional 拉丁文扩充增补			
1F	Greek Extended 希腊文扩充			
20	General Punctuation 广义标点	Super-/Subscripts 上/下标	Currency Symbols 货币符号	Comb. Diacrit Marks for Symbols <sup>a</sup>
21	Letterlike Symbols 类似字母的符号		Number Forms 数字形式	Arrows 箭头
22	Mathematical Operators 数学运算符			
23	Miscellaneous Technical 零杂技术用符号			
24	Control Pictures 控制图符	O. C. R 光学字符识别	Enclosed Alphanumerics 带括号的字母数字	
25	Box Drawing 制表符		Block Elements 方块元素	Geometric Shapes 几何图形符
26	Miscellaneous Symbols 零杂符号			
27	Dingbats 丁贝符(示意符号)			

30	CJK Symbols and Punctuation CJK 符号和标点		Hiragana 平假名	katakana 片假名
31	Bopomofo 注音	Hangul Compatibility Jamo 朝鲜文兼容字母	CJK Miscellaneous CJK 零杂字符	
32	Enclosed CJK Letters and Months 带括号的 CJK 字母及月份			
33	CJK Compatibility CJK 兼容字符			
34	Hangul 朝鲜文			
3D				
3E	Hangul Supplementary-A 朝鲜文补充-A			
44	Hangul Supplementary-B 朝鲜补充文-B			
45				
4D				
4E	CJK Unified Ideographs CJK 统一文字			
9F				
A0				
DF				
E0	Private Use Area 专用区			
F8				
F9	CJK Compatibility Ideographs CJK 兼容汉字			
FA				
FB	Alphabetic Presentation Forms 拼音文字变形显示形式			
FC	Arabic Presentation Forms-A 阿拉伯文变形显示形式-A			
FD				
FE	Comb. Half Marks 半形组合用标志	CJK Compat Foms CJK 兼容形式	Small Form Variants 小写变体	Arabic Presentation Forms-B <sup>b</sup>
FF	Halfwidth and Fullwidth Forms 半形及全形字母			Specials 特殊字符

■ 表示留作将来标准化用

■ 表示非图形字符

a 用于符号的组合用区分标志

b 阿拉伯文变形显示形式-B

图 E.1 GB 13000.1 的基本多文种页结构图

附 录 F  
( 规范性附录 )  
私有数据广播系统注册

SI 扩充包含data\_broadcast\_id的值的分配。在一个复用流中，每一个数据流的data\_broadcast\_id标明了该数据广播的类型或使用的私有系统。

data\_broadcast\_id为本标准定义的7种类型保留了7个值，见表F.1。还留下很宽的范围(0x100-0xFFFF)用作私有系统的注册。

由于允许最小互操作性，因此建议数据广播解决方案进行注册，以帮助解码器识别所能支持的数据流并且防止它们试图获得不兼容的数据流。

表 F.1 data\_broadcast\_id 值的分配

数据广播规范	Data_broadcast_id
为将来使用保留	0x0000
数据管道	0x0001
异步数据流	0x0002
同步数据流	0x0003
从同步数据流	0x0004
多协议封装	0x0005
数据轮播	0x0006
对象轮播	0x0007
数字电视系统 ATM 流	0x0008
本标准为将来使用保留	0x0009-0x00FF
为注册保留	0x0100-0xFFFFE
为将来使用保留	0xFFFF
注：表中阴影部分为本标准定义的 7 种类型。	

## 数字电视系统中的数据广播规范 编制说明

### 一 任务来源

本标准是根据国家广播电影电视总局科技司下发的广技监字[2001]442号《关于下达 2001年广播影视行业标准制、修订计划项目》的通知,由国家广播电影电视总局数字(高清晰度)电视标准工作组负责编制的。

参加本标准编制工作的单位还有:国家广播电影电视总局广播科学研究院、国家广播电影电视总局网络中心、中央电视台、北京歌华有线电视网络股份有限公司、北京海特荣德科技发展有限公司、北京蓝拓扑电子有限公司、中国华大集成电路设计中心、北京中视联数字系统有限公司、北京算通科技发展有限公司、北京首创系统集成技术有限公司、北京清华永新信息工程有限公司、华为技术有限公司、深圳傲龙宽频科技有限公司、深圳市同洲电子股份有限公司、深圳经天通信股份有限公司、四川视达集团公司、浙江天屹网络科技股份有限公司、泰克电子(中国)有限公司、飞利浦(中国)投资有限公司上海研发中心。

### 二 编制目的及意义

数字电视系统中的数据广播是数字电视领域的重要业务之一,它使运营商能够在提供音视频业务的基础上,向用户提供各种数据服务,诸如股票信息、互联网数据等等,而许多其他业务,如中间件、电子节目指南等等,也需要数据广播的支持。

当前在我国,随着数字电视系统的建立和推广,各种新业务的实现已经提上了日程,各级电视台大都正在准备开展数据广播业务,有些电视台已经开始了数据广播的试验播出。但是,由于缺乏一个统一的技术标准来规范数字电视系统中的数据广播业务,从而导致了数据广播系统设备通用性差、运营成本高、系统维护困难,以及无法进行统一的技术测试等问题。本标准的制定,将使上述问题得到有效解决,为我国数字电视系统中数据广播业务的发展提供技术依据、技术支持和技术保障。

### 三 编制原则

本标准在编制过程中遵循了以下原则:

1. 一致性:要求本标准与数字电视系列标准中的其它相关标准保持一致,并对未来相关标准的制定提供支持。
2. 开放性:要求本标准对于现有和可能出现的技术内容给予最大程度地支持,避免成为封闭系统。
3. 合理性:要求本标准简练、精确、层次分明,充分体现广播电视行业的特点。
4. 前瞻性:要求本标准在一定时间范围内,保持先进性和可用性。
5. 可达性:要求本标准具有可实现性、方便性和灵活性,在广播电视数字化进程中能够起到促进作用。

### 四 编制过程

2002年3月8日,起草小组召开了第一次工作会议。会议明确了本标准的任务和目的,确定了编制原则、编写方法、编写格式和具体编写任务。会后起草小组认真查阅了国际的相关标准,开展了调查研究工作。

2002年4月,起草小组确定了本标准的基本结构框架,分配编制任务。

2002年5月,起草小组完成了相关国际标准的翻译、研究工作和本标准的范围、规范性引用文件、术语和定义、缩略语等部分的编制工作,并对数据封装格式、数据广播相关业务的补充规定和接收端规定等部分进行了研讨。

2002年7月,完成了本标准的数据封装格式、数据广播相关业务的补充规定、用户接收端规定、确认对于通用解扰算法的支持,以及数据广播中PSI/SI/EPG/CA系统的规定等部分的编制工作。同时完成了在IPE上对各种封装方式的系统延时和封装效率的比较测试。参加测试的单位有:广科院有线所、北



京蓝拓扑电子有限公司、北京海特容德信息技术有限公司、北京中视联数字系统有限公司、北京算通科技发展有限公司。

测试结果分析：

1．数据管道

广科院有线所与海特荣德公司共同开发了基于数据管道封装方式的数据广播系统,实现了股票数据广播业务,并证实了该方式的有效性和可行性。采用数据管道封装方式,程序空转时(不运行股票分析软件),在10Mbit/s时丢包率是0.1%左右。

2．多协议封装(MPE)

北京蓝拓扑信息技术有限公司对多协议封装进行了效率测试。

“多协议封装”可以使用解复用器的硬件来同步和过滤数据包,可大大减少接收端软件的工作量,提高系统对数据的处理能力,使接收端能够处理更高速率的数据码流,见下表。

MPE 封装效率测试结果

<div>封装TS包的码率 Mbps</div> <div>Wsend发送IP 包的码率 Mbps</div> <div>Wsend发送IP 包的包长(字节)</div>	1	2	3	4	5	6
128	1.596	3.184	4.779	6.375	7.968	9.168
256	1.594	3.187	4.771	6.373	7.965	9.555
512	1.594	3.185	4.772	6.375	7.968	9.560
1024	1.196	2.390	3.581	4.781	5.975	7.170
2048	1.295	2.589	3.880	5.179	6.473	7.776
4096	1.245	2.489	3.734	4.980	6.221	7.468

Wsend软件以UDP的方式发送多播IP包,在发送的IP包长为1024字节时,效率最高,为1.196;在IP包长较小时,效率较低,为1.596。与IP封装成TS时需要增加固定长度的SECTION头的分析结果一致。数据中更长包长的效率反而有所降低的原因是,在封装时使用了填充字节,即每个SECTION结束后,下一个SECTION在新的TS包中开始。

2002年10月,完成了在机顶盒上对各种封装方式的系统资源消耗的比较测试。起草小组在经过了先后五次标准起草工作会议讨论后,初步形成了本标准的讨论稿。

2003年3月,起草小组将本标准讨论稿提交全国广电标委会秘书处审阅,几经修改后,形成了本标准的征求意见稿,并送交各位专家征求意见。

2003年8月,起草小组根据各位专家对征求意见稿提出的意见和建议,对征求意见稿进行了修改和完善,形成了本标准的送审稿。

2004年元月,全国广电标委会组织召开了本标准的审查会。审查委员会同意本标准通过审查,并形成了会议纪要。会后,起草小组根据会议纪要,对相应条款进行了认真修改,形成了本标准的报批稿。

五 有关技术问题的说明

1. 数据广播系统的双向业务和回传信道的规定是非常重要的,但是在本标准中暂未作出规定。其主要原因是我国现有传输网络的双向改造差异较大,同时国家还未确定相关的标准。为了不断适应运营的需要,本标准将会适时增补相关的部分。

2. 关于加密的问题。在国内现有运营系统中已部分采用了IP加密机制,但是鉴于本标准是数字电视系统基带数据的传输标准,而IP加密机制是在应用级别上的处理,为了保持本标准与数字电视系列标准

的一致性和实际操作的可行性，本标准规定了对于数据的处理要遵循已有的CA规范。用户是否选择IP加密，可以自行决定。