



中华人民共和国国家标准

GB/T 28430—2012

数字电视系统数据广播技术规范

Technical specification of data broadcasting in digital television system

2012-06-29 发布

2012-10-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会

发布

目 次

前言	I
引言	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	4
4.3 多协议封装	6
4.4 数据轮播	10
4.5 对象轮播	21
4.6 解码器模型	25
5 数据广播相关业务的规定	26
5.1 EPG 信息的发送方式	26
5.2 中间件业务的发送方式	26
6 数据广播中 CA 系统的规定	26
7 实施方法	26
8 数据广播前端技术要求和测量方法	26
附录 A (规范性附录) 实施方法	27
附录 B (规范性附录) 私有数据广播系统注册	68
附录 C (资料性附录) 文本字符编码	69
附录 D (规范性附录) 数据广播前端的技术要求	72
附录 E (规范性附录) 前端测量方法	73
附录 F (规范性附录) 数据轮播的 DSM-CC 消息	75
附录 G (规范性附录) 将 DSM-CC 消息封装到 GB/T 17975.1—2010 段中	80
附录 H (规范性附录) 目录中的对象命名	82

前 言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准由国家广播电影电视总局提出。

本标准由全国广播电影电视标准化技术委员会(SAC/TC 239)归口。

本标准起草单位: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广播电视规划院。

本标准主要起草人:杨杰、崔竞飞、朱佩江、胡毓铭、杨威、李培琳、牛妍华、万倩、李博、崔俊生。

引 言

本标准从我国数字电视系统数据广播的技术和业务特点出发,在 GY/T 201—2004《数字电视系统中的数据广播规范》的基础上进行了补充和修改,对数据广播前端系统通用技术要求和测量方法进行了补充规范。在本标准的编制过程中,还参照了 GB/T 28161—2011《数字电视广播业务信息规范》和 GY/T 221—2006《有线数字电视系统技术要求和测量方法》的相关内容。

数字电视系统数据广播技术规范

1 范围

本标准规定了数字电视广播系统中基于传送流的数据广播的数据结构、数据交换、数据传输协议等。

本标准适用于地面数字电视、有线数字电视和卫星数字电视广播系统中的数据广播系统。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB 2312—1980 信息交换用汉字编码字符集 基本集

GB 13000—2010 信息技术 通用多八位编码字符集(UCS)

GB/T 15273.1—1994 信息处理 八位单字节编码图形字符集 第一部分:拉丁字母一

GB/Z 15629.1—2000 信息技术 系统间远程通信和信息交换 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—2008 信息技术 系统间远程通信和信息交换 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—2010 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 28161—2011 数字电视广播业务信息规范

GY/T 221—2006 有线数字电视系统技术要求和测量方法

ISO/IEC 13818-6:1998 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 (Information technology—Generic coding of moving pictures and associated audio information—Part 6: Extensions for DSM-CC)

IETF RFC 1112 对于IP多点传送的主机扩展名(Host extensions for IP multicasting)

IETF RFC 1950 ZLIB 压缩数据格式规范 3.3 版(ZLIB Compressed Data Format Specification version 3.3)

IETF RFC 1951 简化压缩数据格式规范 1.3 版(DEFLATE Compressed Data Format Specification version 1.3)

IETF RFC 2045 多目标互联网邮件扩展 第1部分:INTERNET 消息主体格式 (Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies)

IETF RFC 2046 多目标互联网邮件扩展 第2部分:媒体类型 (Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types)

ETSI EN 300 802 数字视频广播(DVB) DVB 交互业务的网络独立协议 (Digital Video Broadcasting (DVB); Network-independent protocols for DVB interactive services)

3 缩略语

下列缩略语适用于本文件。

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

PTS 展现时间戳(Presentation Time Stamp)
 RFC 评议征求(Request For Comments)
 SDT 业务描述表(Service Description Table)
 SI 业务信息(Service Information)
 SNAP 子网附属点(SubNetwork Attachment Point)
 TCP 传输控制协议(Transfer Control Protocol)
 TS 传送流(Transport Stream)
 bsbf 比特串,左位在先(bit string, left bit first)
 ppm 百万分之一(parts per million)
 rpchf 余数多项式系数,高阶在前(remainder polynomial coefficient, higher order first)
 uimbf 无符号整数,高位在先(unsigned integer, most significant bit first)

4 数据封装格式

4.1 数据管道

4.1.1 基本概念

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—2010。数据管道的具体实施方法见附录 A。

4.1.2 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。

数据广播业务可通过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—2010。

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。

4.1.3 PSI 和 SI 规范

4.1.3.1 PSI 和 SI 的使用方法

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。

4.1.3.2 `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 B;
- `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。

4.1.3.3 流类型

PMT 中的 `stream_type` 不在本标准中定义。

4.2 数据流

4.2.1 基本概念

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—2010。

异步数据流定义为只有数据、没有任何时序要求的流，例如 RS-232。

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。

从同步的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与其他数据流(例如音频，视频)一起回放。

数据流方式的具体实施方法见附录 A。

4.2.2 异步数据流

4.2.2.1 数据传输规范

数据广播业务的数据放在 GB/T 17975.1—2010 定义的 PES 包中，该 PES 包的长度不能为零。GB/T 17975.1—2010 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。

4.2.2.2 PSI 和 SI 规范

4.2.2.2.1 PSI 和 SI 的使用方法

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，如果作为数据流的码流 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。

4.2.2.2.2 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 B；
- 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。

4.2.2.2.3 流类型

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

4.2.3 同步或从同步数据流

4.2.3.1 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—2010 中定义的 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—2010。

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 sub_stream_id PTS_extension_flag output_data_rate_flag reserved PES_data_packet_header_length if (PTS_extension_flag==“1”) { reserved PTS_extension } if (output_data_rate_flag==“1”) { reserved output_data_rate } for (i=0;i<N1;i++) { PES_data_private_data_byte } for (i=0;i<N2;i++) { PES_data_byte } }	 8 8 1 1 2 4 7 9 4 28 8 8	 uimsbf uimsbf bslbf bslbf bslbf uimsbf bslbf bslbf bslbf uimsbf bslbf bslbf

PES_data_packet 的语义如下:

- data_identifier: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_identifier 的值相同,编码见表 2;

表 2 data_identifier 域的编码

data_identifier	意 义
0x00 ~ 0x0F	为将来使用保留
0x10 ~ 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
0x23~0x7F	为将来使用保留
0x80~0xFF	用户定义

- sub_stream_id: 8 比特域, 用户自定义;
- 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 11.1 us(90 kHz)扩展到 37 ns(27 MHz);
- 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。

4.2.3.2 PSI 和 SI 规范

4.2.3.2.1 PSI 和 SI 的使用方法

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 的值应与可能出现的作为数据流的码流 PMT 的 stream_identifier_descriptor 的 component_tag 的值相同。

4.2.3.2.2 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 B。
- 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。

4.2.3.2.3 流类型

通过在业务的 PMT 中将 stream_type 置为 0x06 或用户自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

4.3.1 基本概念

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 13818-6:1998, 这和 GB/T 17975.1—2010 的私有段格式是兼容的。

多协议封装的具体实施方法见附录 A。

4.3.2 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 13818-6:1998。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	助记符
datagram_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
MAC_address_6	8	uimsbf
MAC_address_5	8	uimsbf
reserved	2	bslbf
payload_scrambling_control	2	bslbf
address_scrambling_control	2	bslbf
LLC_SNAP_flag	1	bslbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
MAC_address_4	8	uimsbf
MAC_address_3	8	uimsbf
MAC_address_2	8	uimsbf
MAC_address_1	8	uimsbf
if (LLC_SNAP_flag == “1”) {		
LLC_SNAP()		
}		
else {		
for (j=0;j<N1;j++) {		
IP_datagram_data_byte	8	bslbf
}		
}		
if (section_number == last_section_number){		
for (j=0;j<N2;j++) {		
stuffing_byte	8	bslbf
}		
}		
if (section_syntax_indicator == “0”) {		
checksum	32	uimsbf
}		
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 13818-6:1998);
- section_syntax_index_indicator: 此域按照 ISO/IEC 13818-6:1998 中定义的设置;
- private_indicator: 此域按照 ISO/IEC 13818-6:1998 中的定义设置;
- reserved: 2 比特域, 置为“11”;
- section_length: 此域按照 ISO/IEC 13818-6:1998 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。

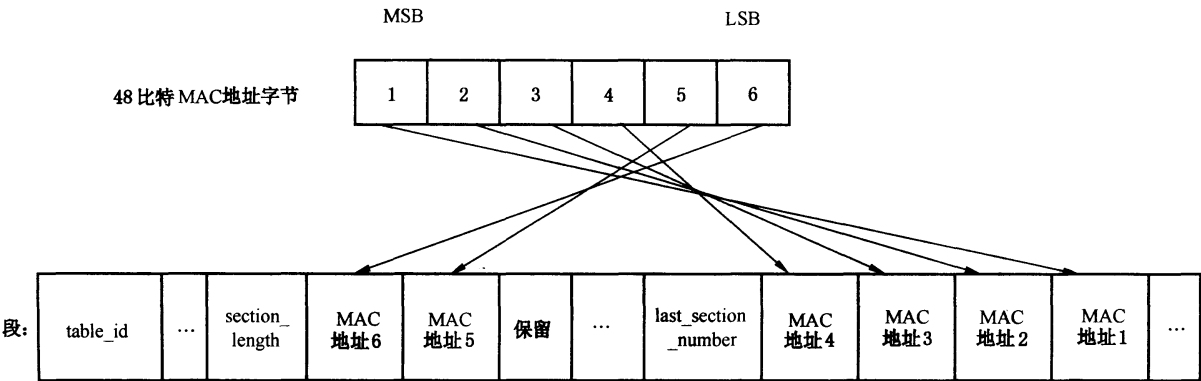


图 1 MAC 地址字节到 section 的映射

- 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自定义;

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 地址可动态变化。所采用的加扰方法由用户自定义;

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”,表明在载荷的 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”,表明在载荷的 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—2008 规范和 SNAP 遵循 GB/Z 15629.1—2000 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/IEC 13818-6:1998 定义设置。它对整个 datagram_section 进行运算;
- CRC_32: 此域按 ISO/IEC 13818-6:1998 定义设置。它对整个 datagram_section 进行运算。

4.3.3 PSI 和 SI 规范

4.3.3.1 PSI 和 SI 的使用方法

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。

4.3.3.2 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 B;
- 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
multiprotocol_encapsulation_info () {		
MAC_address_range	3	uimsbf
MAC_IP_mapping_flag	1	bslbf
alignment_indicator	1	bslbf
reserved	3	bslbf
max_sections_per_datagram	8	uimsbf
}		

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。

4.3.3.3 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

4.4.1 基本概念

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、添加或删除。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。

数据轮播的具体实施方法见附录 A。

4.4.2 数据传输规范

4.4.2.1 描述

数据轮播规范是基于 DSM-CC 数据轮播规范的，见 ISO/IEC 13818-6:1998。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，模块中的所有其他块都是相同大小的。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 13818-6:1998。本标准将规定这些消息在数据轮播中的使用。

4.4.2.2 数据轮播的结构

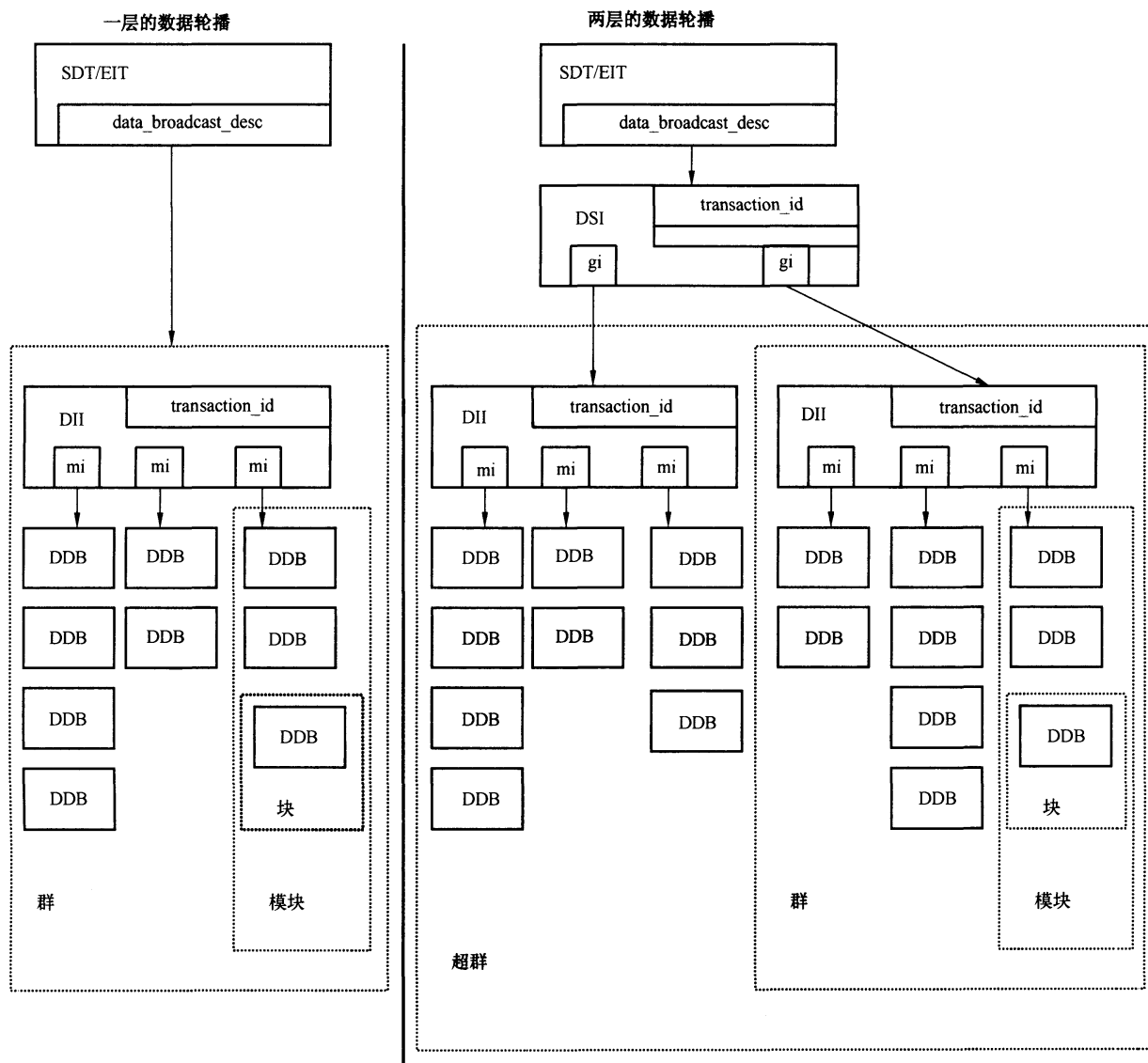
数据轮播可有一层或两层控制信息，结构见如图 2。数据轮播最简单的模式是一个数据轮播拥有一个控制层，该控制层描述了单个群。在这种情况下，SDT/EIT 表中包含了指向 DownloadInfoIndication 消息的 data_broadcast_descriptor。这个消息描述了数据轮播使用的 ModuleInfoByte 域中的模块。这个域包含了一个描述符的循环，描述符中可能包含了各种不同的信息，如指向 DownloadDataBlock 消息位置的指针。

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。



说明：

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

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。

4.4.2.3 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 for(i=0;i< numberOfGroups;i++) { GroupId GroupSize GroupCompatibility() GroupInfoLength for(i=0;i<N;i++) { groupInfoByte } } privateDataLength for(i=0;i< privateDataLength;i++) { privateDataByte } }	 2 4 4 2 1 2 1

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。

4.4.2.4 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。模块的其他属性由后面定义的描述符传输。moduleId 在 0xFFF0-0xFFFF 范围内的值在本标准中不使用。

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 0xFFF0-0xFFFF 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。

4.4.2.5 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。

4.4.2.6 DownloadCancel 消息

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。

4.4.3 描述符

4.4.3.1 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	+		级联的数据模块
CRC32	0x05	+		CRC
location	0x06	+	+	数据位置
est_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

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

4.4.3.2 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。

4.4.3.3 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 C。

4.4.3.4 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/T 15273.1—1994 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 C。

4.4.3.5 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。

4.4.3.6 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—2010 的附录 A。

4.4.3.7 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_de-

scriptor 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 (PSI 的描述符) 中的 component_tag 值相同。

4.4.3.8 est_download_time_descriptor

est_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 est_download_time_descriptor 的句法。

表 18 est_download_time_descriptor 的句法

句 法	字节数	值
est_download_time_descriptor(){ descriptor_tag descriptor_length est_download_time }	1 1 4	0x07

est_download_time_descriptor 的语义如下：

- descriptor_tag: 8 比特域, 标识了描述符。对于 est_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。

4.4.3.9 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。

4.4.3.10 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。

4.4.3.11 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。

4.4.4 PSI 和 SI 规范

4.4.4.1 PSI 和 SI 的使用方法

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。

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 B;
- component_tag: 此域值和用于数据轮播的码流的 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 reserved transaction_id time_out_value_DSI time_out_value_DII reserved leak_rate }	2 6 32 32 32 2 22	bslbf bslbf uimsbf uimsbf uimsbf bslbf bslbf

data_carousel_info 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。

4.4.4.2 流类型

包含数据轮播的码流的 PMT 中该业务的 steam_type 按 GB/T 17975.1—2010 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

4.5.1 基本概念

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按照 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。

对象轮播的具体实施方法见附录 A。

4.5.2 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。

4.5.3 数据传输规范

4.5.3.1 描述

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 地址用来从其他业务域查询对象轮播,其句法和语义在下面定义。

4.5.3.2 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	carouselId 4字节	specifier 4字节	privateData 10字节
------------	-------------	-------------------	------------------	---------------------

图 3 轮播 NSAP 地址格式

AFI、Type、carouselId 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 格式用于用户自定义;
- Type:8 比特域,置为 0x00。标明对象轮播使用 NSAP 地址;
- carouselId:32 比特域,设置为对象轮播的标识符,也就是 carouselId 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	uimbsf uimbsf uimbsf bslbf

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。

4.5.4 PSI 和 SI 规范

4.5.4.1 PSI 和 SI 的使用方法

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,也可以与其他数字电视业务的 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他数字电视业务的 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 4.5.4.3 条中描述。

4.5.4.2 data_broadcast_descriptor

data_broadcast_descriptor 按以下方式使用:

- data_broadcast_id:此域置为 0x0007,标明对象轮播,见附录 B;
- 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

句 法	比特数	类型
object_carousel_info () { carousel_type_id reserved transaction_id time_out_value_DSI time_out_value_DII reserved leak_rate for (I=0;i<N1;i++) { ISO_639_language_code object_name_length for (j=0;j<N2;j++) { object_name_char } } }	2 6 32 32 32 2 22 24 8 8	bslbf bslbf uimsbf uimsbf uimsbf bslbf uimsbf bslbf uimsbf uimsbf

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_{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；
- 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 C 描述的字符集来编码。

4.5.4.3 Deferred_association_tags_descriptor

数字电视系统中 deferred_association_tags_descriptor() 的句法和语义见表 26。

表 26 deferred_association_tags_descriptor

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。

4.5.4.4 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。解码器模块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—2010 中定义的传输系统目标解码器(T-STD)很相似。

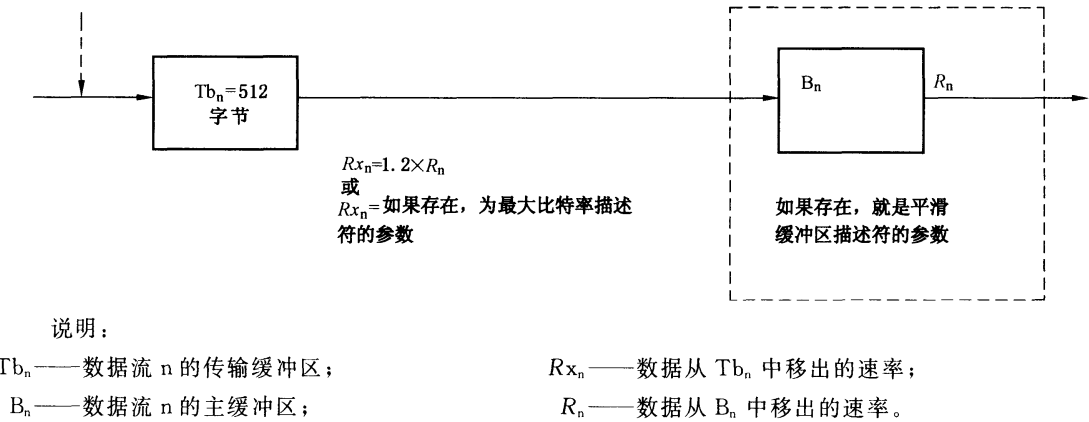


图 4 数据业务解码器模型

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 R_{x_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。其他字节不送入 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。

对于一个特定的业务，传输缓冲区的移出速率 R_{x_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 R_{x_n} 、 B_n 和 R_n 的值，见 GB/T 17975.1—2010。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 maxmum_bit_rate 域标明 R_{x_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 $R_{x_n} = 1.2R_n$ 。

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 R_{x_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，使用本标准规定的数据轮播或对象轮播规范。

如果存在与音视频业务具有时间相关性/同步性的 EPG 业务内容，使用 Data-Streaming 中的同步或从同步规范传输。

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 中间件业务的发送方式

对于需要使用数据广播通道传输特定中间件应用、应用数据的情况，使用本标准规定的数据轮播或对象轮播规范，并遵循相应的数据交换协议。

对于中间件系统的具体数据传输方式、数据交换协议以及内容组织机制，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 实施方法

本标准的具体实施方法详见附录 A。

8 数据广播前端技术要求和测量方法

数据广播前端的技术要求测量方法见附录 D 和附录 E。

附录 A
(规范性附录)
实施方法

A.1 操作规则

这一条款为本标准应用实施方法。本实施方法主要参照了 TR 101 202《Digital Video Broadcasting (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			※				
双向业务数据	※				※		
EPG			※			※	※
中间件						※	※
(准)静态数据文件	※					※	※

表 A.1 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。

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—2010,DSM-CC 结构框架见 ISO/IEC 13818-6:1998,互联网络协议(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,其他协议也可以被封装进来。本附录中的表中灰色部分为本附录进行了扩展定义的内容。

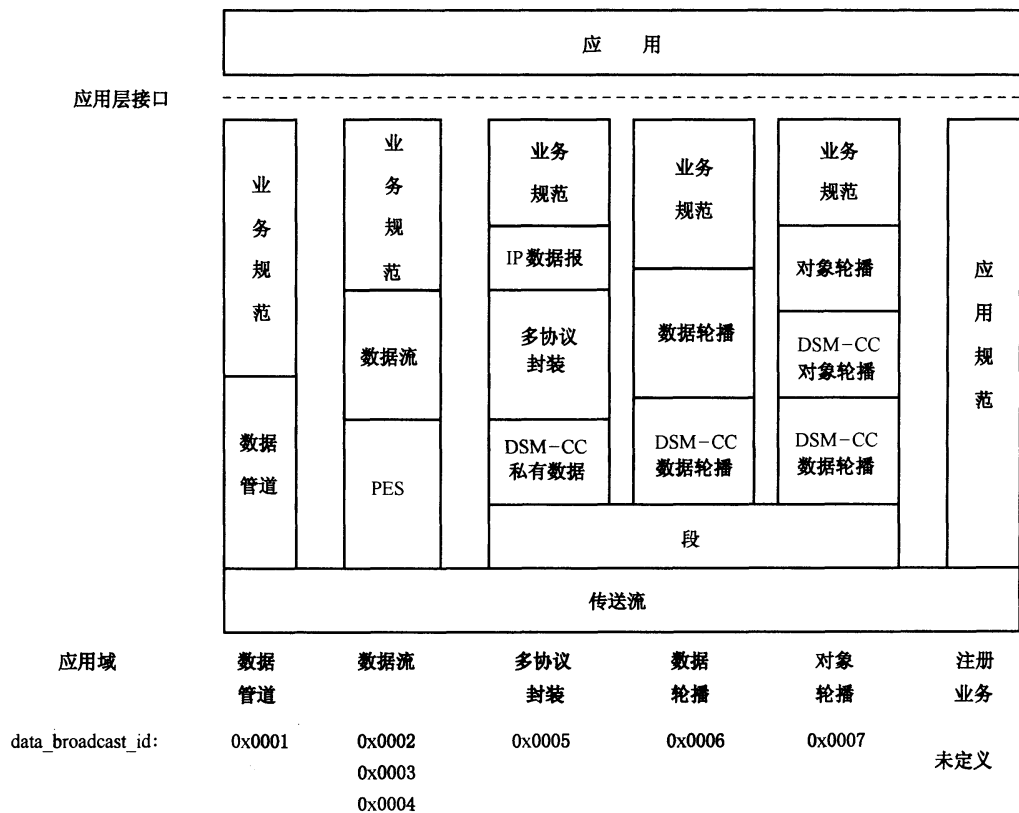


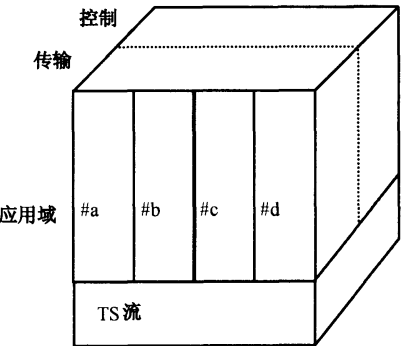
图 A.1 与其他标准关系的图例

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。与其他应用域相比,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—2010 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—2010 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/T 230—2008 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。



说明:

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/T 230—2008 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,而其他类则自由得多。

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。

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—2010 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/T 230—2008。

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—2010 的 2.4.3 条。

适应域的主要作用是填充,也可以有其他用途,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—2010 的 2.4.3 条,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—2010 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—2010。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—2010 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 kb/s~Mb/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 μ s 扩展到 37 ns (27 MHz 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 400 b/s 分辨率), 可实现与高达 27 MHz(± 30 ppm)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 1 b/s 倍数的应用, 以 b/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—2010 对 27 MHz system_clock_frequency 规定的 30 ppm 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(A.1):

$$\text{output_data_rate} = \text{比特率} \times M / \text{system_clock_frequency} \cdots \cdots \cdots (\text{A.1})$$

式(A.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 27 MHz system_clock_frequency 的同步数据流的实际比特率范围是从 1 b/s 到 27 Mb/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 400 b/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 4 800 字节。对于 4 ms 典型最大多路复用抖动和高达 9 Mb/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 封装可以传送其他任意的网络协议,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 4 080 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 4 080 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 4 074 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。

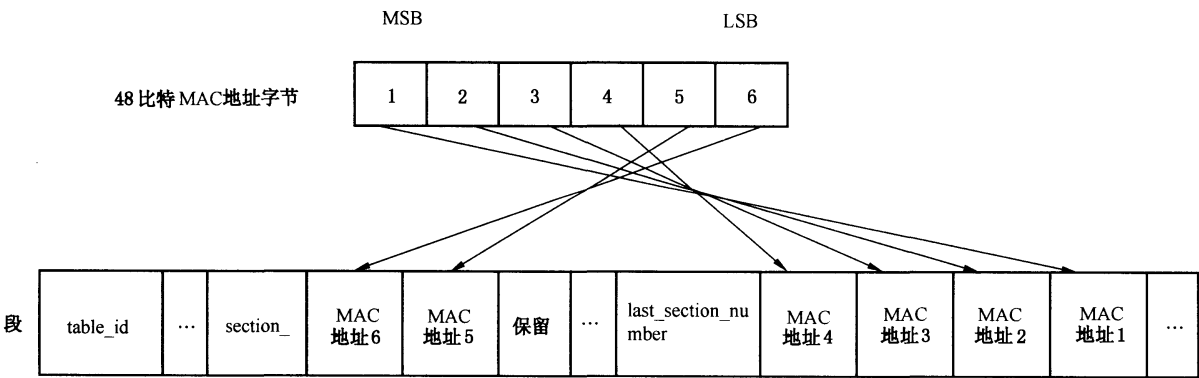


图 A. 3 MAC 地址字节到段的映射

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。

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“file1”、“file2”和“file3”。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 F 中描述。本标准的附录 G 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。在其他所有情况中,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

A.7.4.1 基本概念

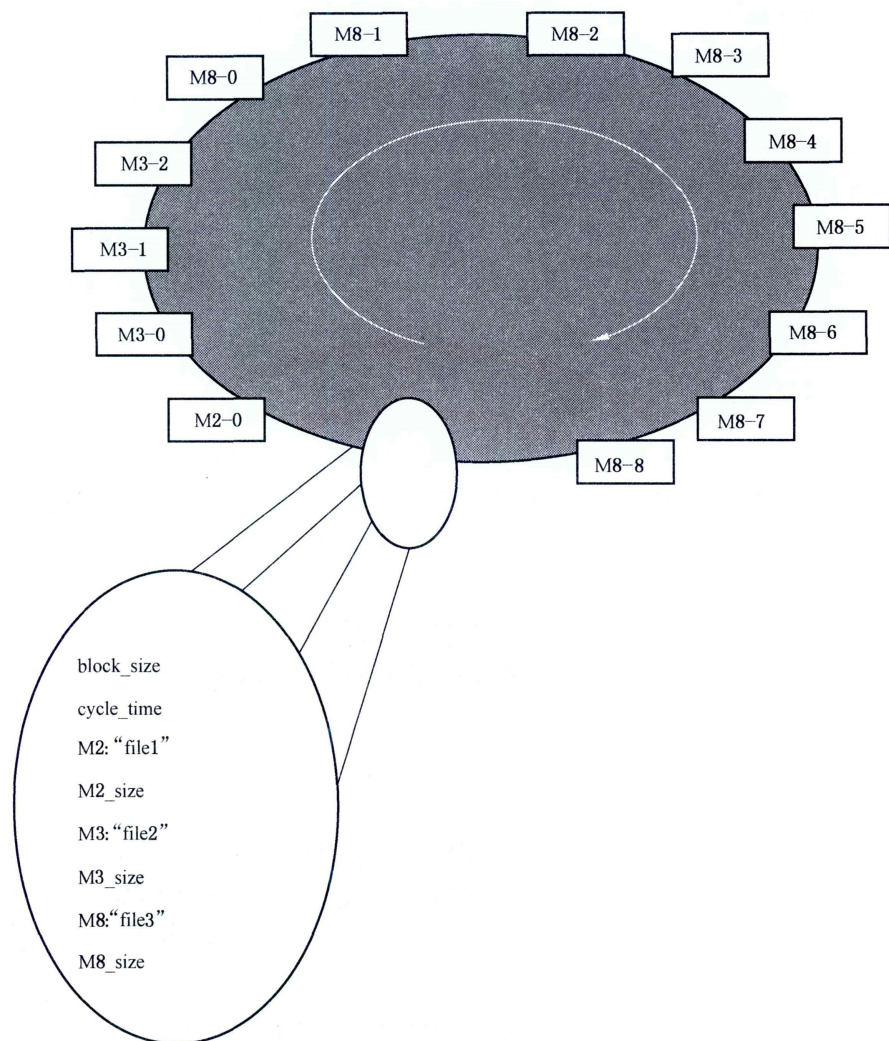
两层数据轮播有一个或多个 DownloadInfoIndication 消息并且只有单个 DownloadServerInitiate 消息(也就是控制信息的两层)。DownloadServerInitiate 消息是被作为最高级别的控制消息来使用的。

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.7.4.2 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。



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

A.7.4.3 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 groupCompatability 结构用来决定一个给定的接收端类型使用哪个群。

A. 7. 4. 4 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 DownloadInfoInitiate 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 13818-6:1998 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 13818-6:1998 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 13818-6:1998 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 moudleVersion。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 2 s,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 13818-6:1998 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	moduleId	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

A.7.6.1 说明

这一条描述的所有描述符都是可选的。

A.7.6.2 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 2045,由一个字符串来指定该模块的类型。

A.7.6.3 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.7.6.4 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.7.6.5 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,这样就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.7.6.6 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.7.6.7 location_descriptor

在 DownloadServerInitiate 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 DownloadInfoIndication 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.7.6.8 est_download_time_descriptor

est_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 Mb/s)而在第二天以另一个比特率(例如 2 Mb/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.7.6.9 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—2010 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.7.6.10 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.7.6.11 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

A.7.7.1 说明

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.7.7.2 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, 见本标准的表 23。

transacion_id 的使用在 A.7.4 条中描述。

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 transacionId 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 条中描述)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 2 s, 这可能会对业务首次访问带来延迟。

A.7.7.3 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 0.1 s, 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他流的参考基准。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。

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,因此在本地图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 ETSI EN 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。

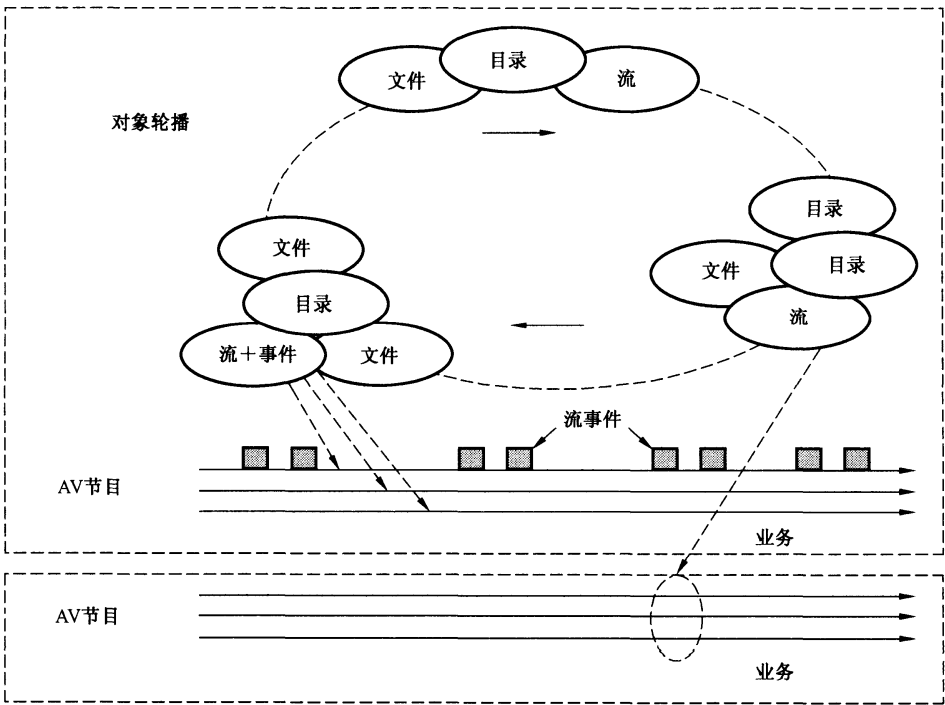


图 A.5 含有对象轮播的业务示例

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。

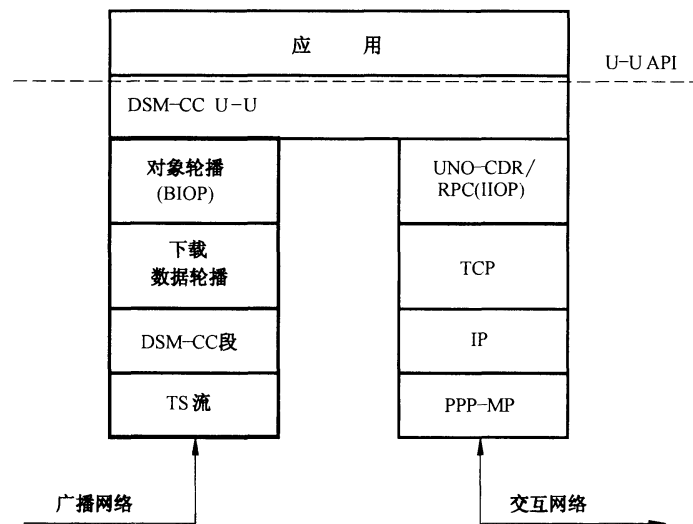


图 A.6 对象轮播协议在交互业务结构框架中的位置

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 13818-6:1998 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 H。

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他业务域。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 13818-6:1998。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 13818-6:1998,GB/T 17975.1—2010)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。

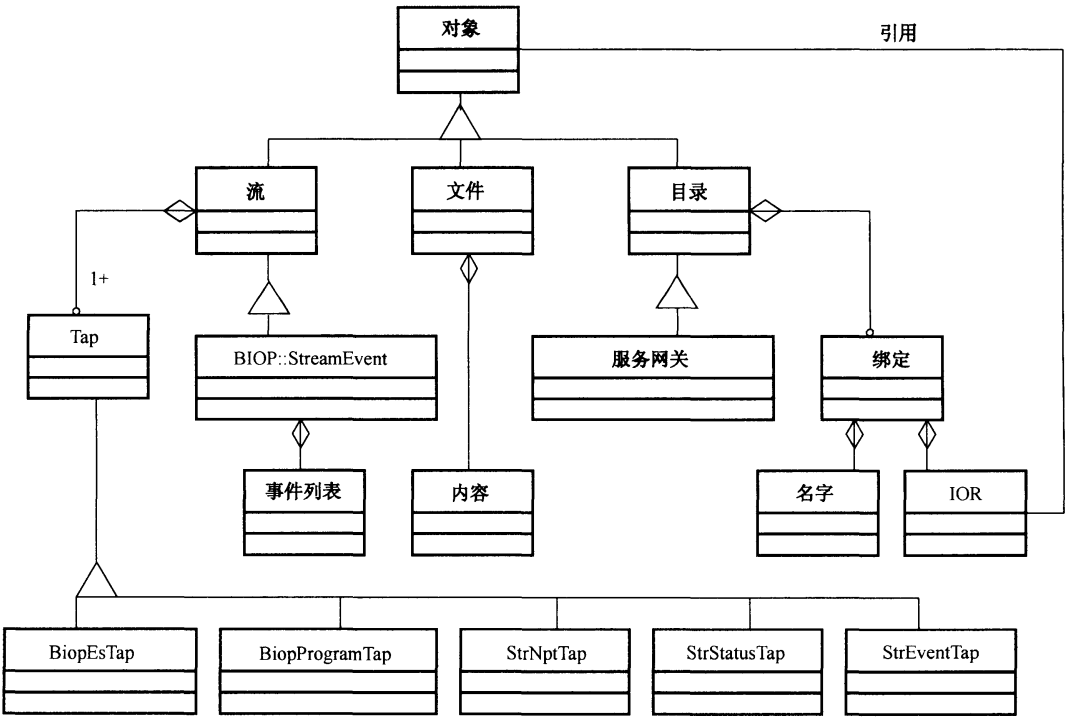


图 A.7 对象轮播所支持的对象

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(BiopEsTap),要么指向完整的节目(BiopProgramTap)。另外,其他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 13818-6:1998。一个模块由一个或更多个链接的 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 13818-6:1998。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。

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 13818-6:1998。通过 PSI 或者 PSI 与 SI 一起,可以找到这个消息。

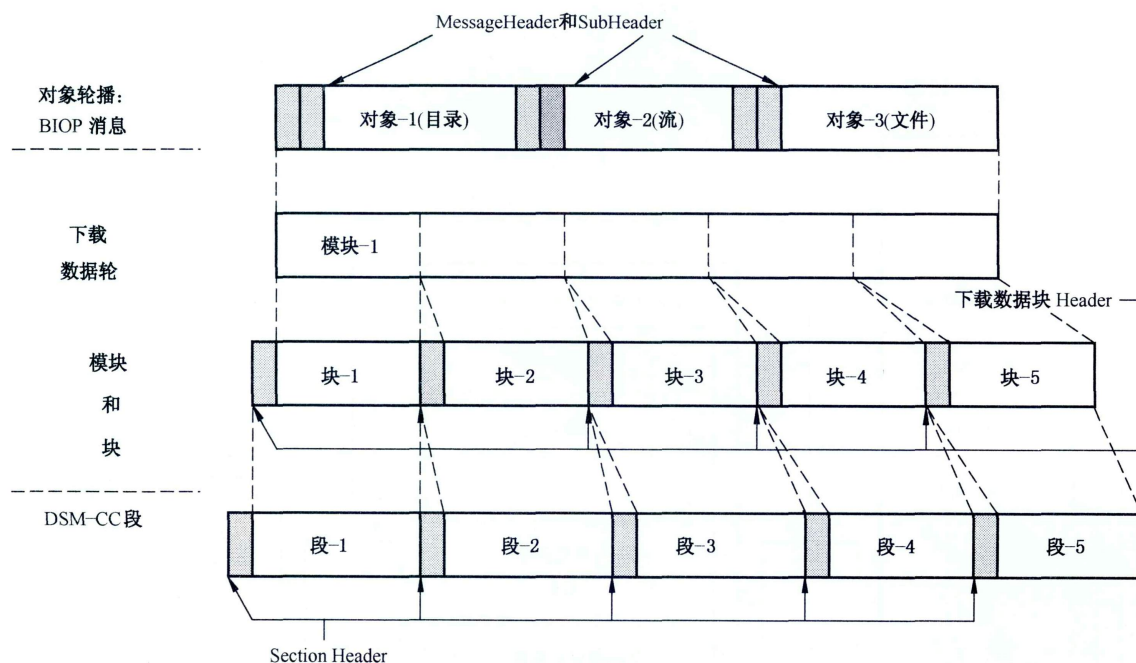


图 A.8 BIOP 消息在模块、数据块、GB/T 17975.1—2010 段中的封装和切分

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/IEC 13818-6:1998。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须的协议。

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。

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(可能使用其他传输参数)的其他可选码流。

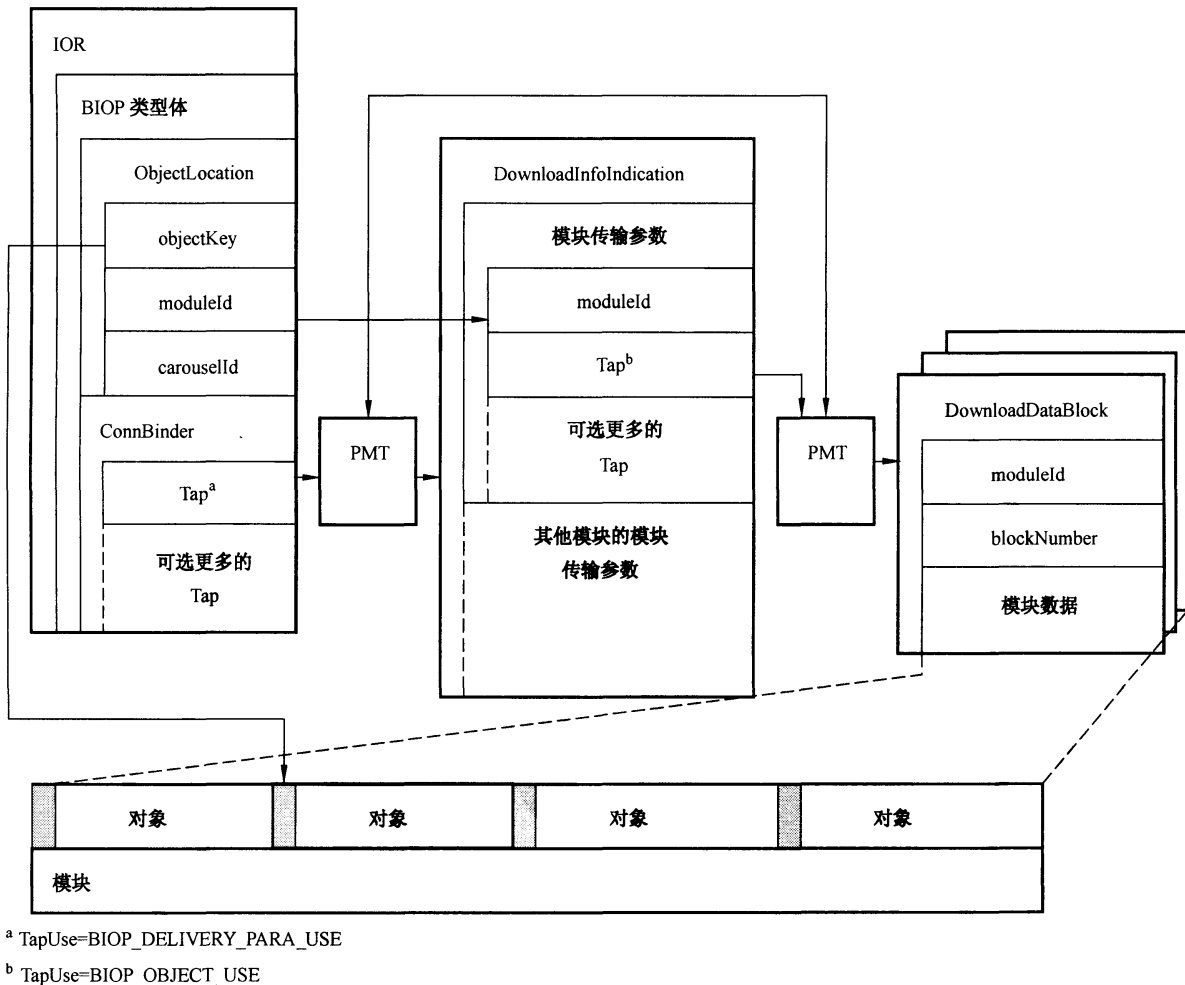


图 A.9 带有 BIOP 类型体的 IOR 如何被解析成对象

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 Service Location 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 orignal_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。

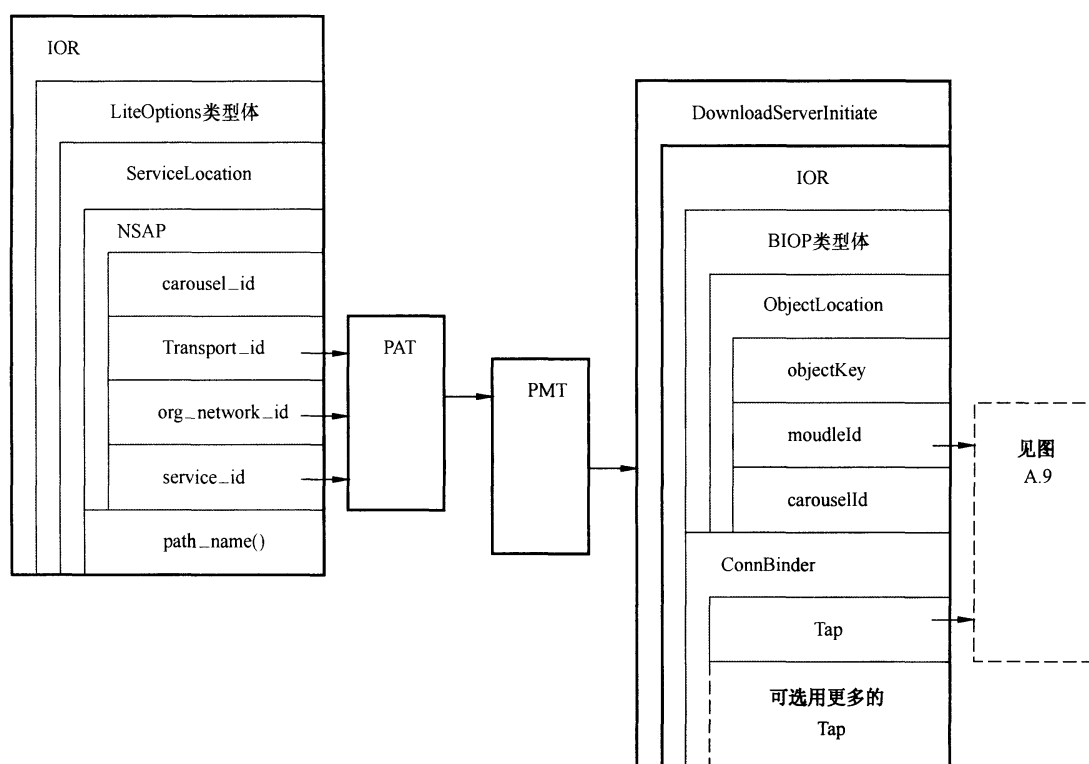


图 A.10 带有 LiteOptions 类型体的 IOR 如何被解析成业务网关

A.8.2.5 Tap 和关联

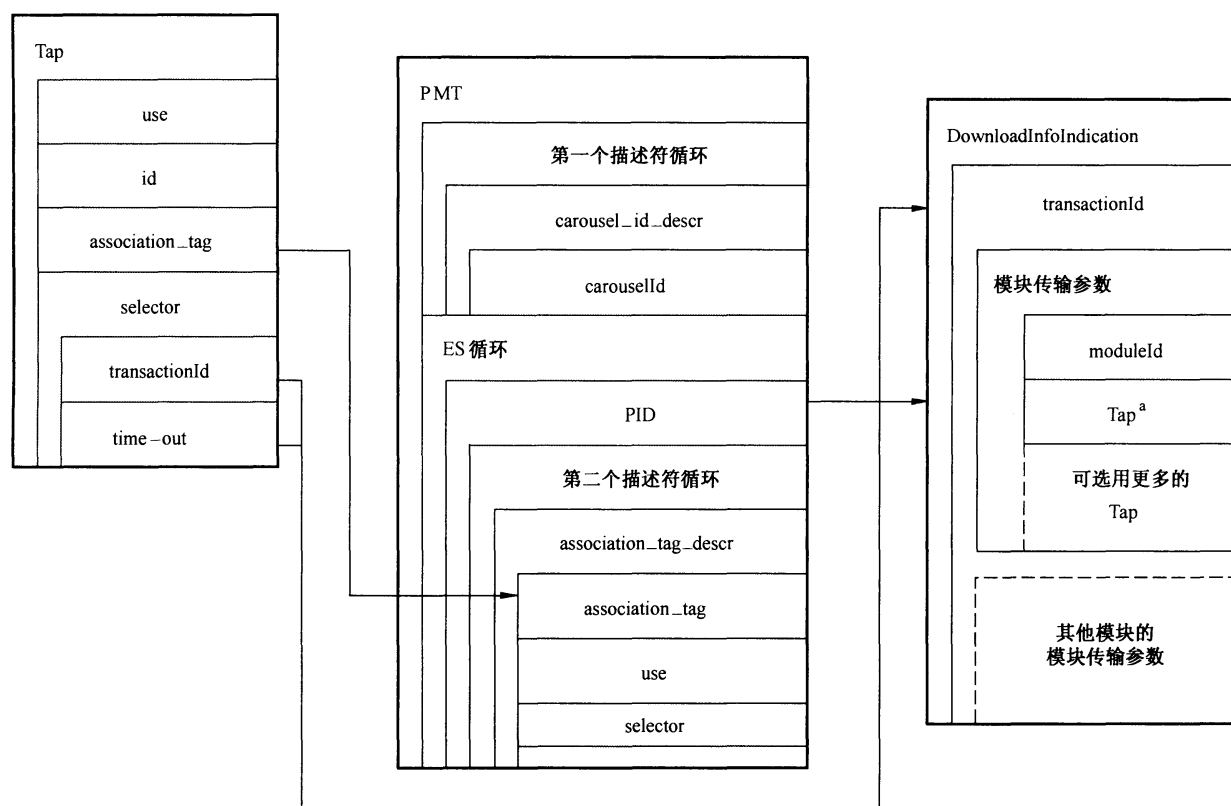
IOR 并不通过 PID 直接指向码流,因为再复用器可以改变 PID。因而 DSM-CC 定义了 Tap,见 ISO/IEC 13818-6:1998,按照类似于 SI 中 component tag 的方式使用。

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 μs 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—2010 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—2010 TS 流的广播网络中实现 U-U 对象轮播,ISO/IEC 13818-6:1998 定义了三个能插入 GB/T 17975.1—2010 PMT 的描述符:

- carousel_identifier_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 Download-ServerInitiate 消息(DSI)。这个 DSI 包含了业务网关的 IOR;
- The deferred_association_tags_descriptor 包含有一系列的,与其他节目的 ES 流(PMT)相关联的,或者指向其他节目的 association_tag。图 A.12 说明了 deferred_association_tags_descriptor 如何指向其他节目。

A.8.3 BIOP 控制结构

A.8.3.1 基本概念

BIOP 控制和数据结构在 ISO/IEC 13818-6:1998 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 13818-6:1998,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。

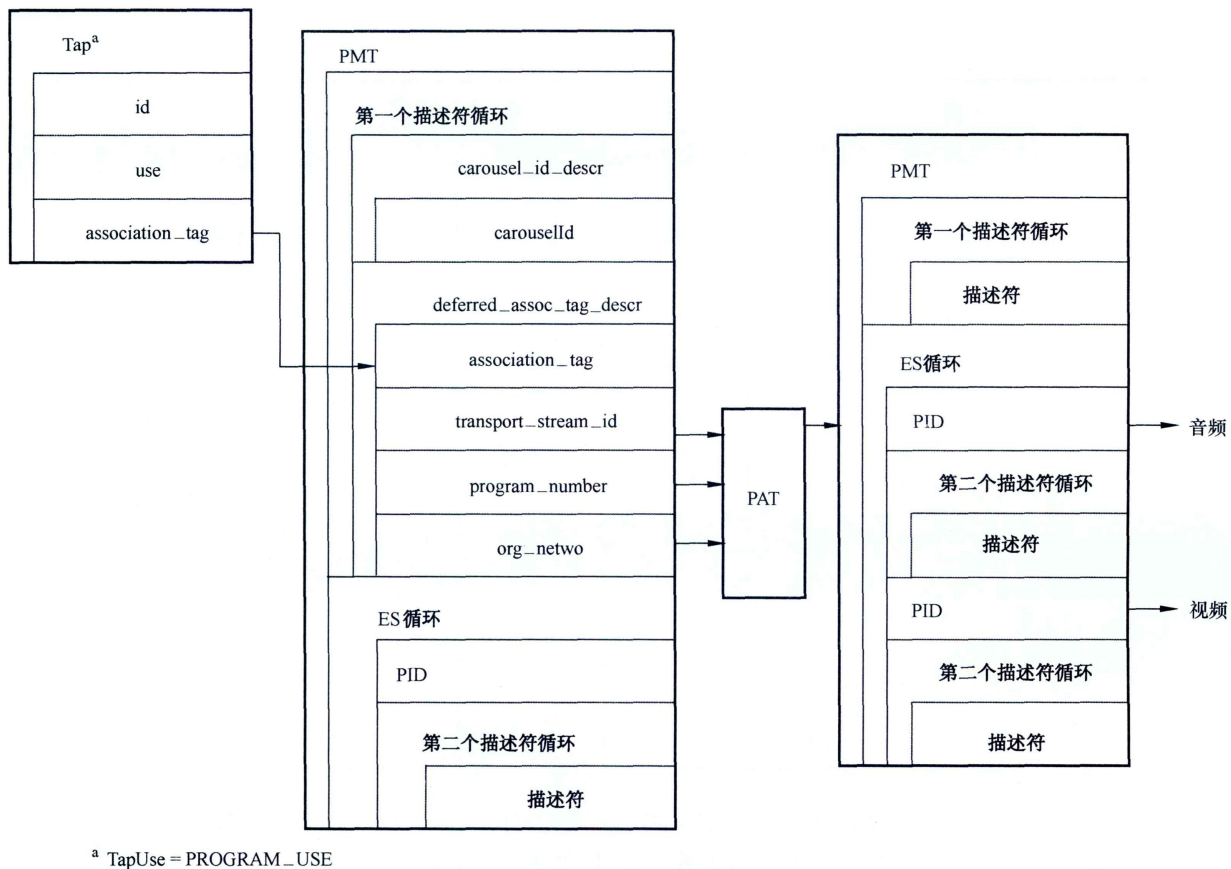


图 A.12 使用 deferred_association_tags_descriptor 标识节目

在该段当中 BIOP 控制结构使用了 GB/T 17975.1—2010 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—2010 句法。在 ISO/IEC 13818-6:1998 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 13818-6:1998 中的定义为准。

A.8.3.2 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 13818-6:1998。

表 A.6 IOP::IOR 句法

句 法	比特数	类型	值	注 释
IOP::IOR { type_id_length for(i=0;i<N1;i++){ type_id_byte } if(N1 % 4≠0){ for(i=0;i<(4-(N1%4));i++){ alignment_gap } }	32	uimbsf	N1	
	8	uimbsf		见表 A.7
	8	uimbsf	0xFF	CDR alignment rule

表 A.6 (续)

句 法	比特数	类型	值	注 释
}				
} taggedProfiles_count for(n=0;n<N2;n++){ IOP::taggedProfile(){ profileId_tag	32	uimbsf	N2	Profile bodies
profile_data_length	32	uimbsf		例如 TAG_BIOP 例如 TAG_LITE_OPTIONS
for(i=0;i<N3;i++){ profile_data_byte	8	uimbsf	N3	例如 BIOPProfileBody 例如 LiteOptionsProfileBody
} } } }				

IOR 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 IOR 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

全称 type_id	别名 type_id
“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 IOR 包含了 taggedProfileList 中的 BIOP 的类型体。ISO/IEC 13818-6:1998 允许一个 IOR 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,其他的类型体可被忽略。

IOR 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.8.3.3 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 (续)

句 法	比特数	类型	值	注 释
<pre> } } } for(n=0;n<N5;n++){ BIOP::LiteComponent { componentId_tag 32 uimsbf component_data_length 8 uimsbf for(i=0;i<N6;i++){ component_data_byte 8 uimsbf } } }</pre>			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。为了指向其他轮播中的对象,要使用 LiteOptions 类型体。

LiteOptionComponents 列表应该按照次序包含一个 BiopObjectLocation 和一个 DsmConnection-Binder 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 流可用作多个目的。表 A.9 说明了定义了的 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、transacionId 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 μs 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:

- 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.8.3.4 Lite Options 类型体

为了指向其他业务域中的对象,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 13818-6:1998),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 使用者可通过 ServiceLocation 结构中的 service Domain 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 类型体句法

句 法	比特数	类型	值	注 释
LiteOptionsProfileBody {				
profileId_tag	32	uimsbf	0x49534F05	TAG_LITE_OPTIONS(Lite Options 类型体)
profile_data_length	32	uimsbf		
profile_data_byte_order	8	uimsbf	0x00	big endian 字节顺序
component_count	8	uimsbf	N1	
DSM::ServiceLocation {				
componentId_tag	32	uimsbf	0x49534F46	TAG_ServiceLocation
component_data_length	32	uimsbf		
serviceDomain_length	8	uimsbf	0x14	轮播 NSAP 地址的长度
serviceDomain_data()	160	uimsbf		轮播 NSAP 地址,见表 A.11
CosNaming::Name(){				路径名称
nameComponents_count	32	uimsbf	N2	
for(i=0;I<N2;i++){				
id_length	32	uimsbf	N3	NameComponent id
for(j=0;j<N3;j++){				
Id_data_byte	8	uimsbf		
}				
}				

表 A. 10 (续)

句 法	比特数	类型	值	注 释
kind_length	32	uimsbf	N4	NameComponent kind
for(j=0;j<N4;j++){ kind_data_byte	8	uimsbf		同 type_id,见表 A. 7
}				
}				
initialContext_length	32	uimsbf	N5	
for(n=0;n<N5;n++){ InitialContext_data_byte	8	uimsbf		
}				
}				
}				
for(n=0;n<N6;n++){ BIOP::LiteOptionComponent{ componentId_tag	32	uimsbf		N6=N1 - 1
component_data_length	8	uimsbf	N7	
for(i=0;i<N7;i++){ component_data_byte	8	uimsbf		
}				
}				
}				

ServiceLocation 组件应为该类型体的第一个组件。

A. 8. 3. 5 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 句法

句 法	比特数	类型	值	注 释
carouseINSAPaddress(){ AFI	8	uimsbf	0x00	为私有使用定义的 NSAP
Type	8	uimsbf	0x00	对象轮播 NSAP 地址
carouselId	32	uimsbf		
specifierType	8	uimsbf	0x01	IEEE OUI
specifierData{IEEE OUI}	24	uimsbf	0x<>	数字电视的 OUI 常量
service_location(){ transport_stream_id	16	uimsbf		
original_network_id	16	uimsbf		
service_id	16	uimsbf		(=program_number)
reserved	32	bslbf	0xFFFFFFFF	
}				
}				

AFI、type、carouselId、specifierData、transport_stream_id、original_network_id、service_id 和各域定义在本标准正文中。

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 binding-Name)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。

串应以“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 句法

句 法	比特数	类型	值	注 释
BIOP::DirectoryMessage(){				
magic	4x8	uimsbf	0x42494F50	“BIOP”
biop_version.major	8	uimsbf	0x01	BIOP 主版本 1
biop_version.minor	8	uimsbf	0x00	BIOP 副版本 0
byte_order	8	uimsbf	0x00	big_endian 字节排序
message_type	8	uimsbf	0x00	
message_size	32	uimsbf		
objectKey_length	8	uimsbf	N1	
for(i=0;i<N1;i++){				
objectKey_data_byte	8	uimsbf		
}				
objectKind_length	32	uimsbf	0x00000004	
objectKind_data	4x8	uimsbf	0x64697200	“dir”type_id 别名
objectInfo_length	16	uimsbf	N2	objectInfo
for(i=0;i<N2;i++){				
objectInfo_data_byte	8	uimsbf		
}				
serviceContextList_count	8	uimsbf	N3	serviceContextList
for(i=0;i<N3;i++){				
serviceContextList_byte	8	uimsbf		
}				
message_Body_length	32	uimsbf		
bindings_count	16	uimsbf	N4	
for(i=0;i<N4;i++){				
BIOP::Name(){				
nameComponents_count	8	uimsbf	N5	绑定

表 A. 12 (续)

句 法	比特数	类型	值	注 释
<pre> for(i=0;i<N5;i++){ id_length for(j=0;j<N6;j++){ id_data_byte } kind_lenght for(j=0;j<N7;j++){ kind_data_byte } } } bindingType IOP::IOR() objectInfo_length for(j=0;j<N8;j++){ objectInfo_data_byte } } </pre>	8	uimsbf	N6	NameComponent id
	8	uimsbf		
	8	uimsbf	N7	NameComponent kind
	8	uimsbf		同 type_id,见表 A. 7
	8	uimsbf		0x01 用于 nobject 0x02 用于 ncontext ObjectRef,见表 A. 6
	16	uimsbf	N8	
	8	uimsbf		

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 句法

句 法	比特数	类型	值	注 释
BIOP::FileMessage(){ Magic biop_version.major biop_version.minor byte_order message_type message_size objectKey_length for(i=0;i<N1;i++){ objectKey_data_byte } objectKind_length objectKind_data objectInfo_length DSM::File::ContentSize for(i=0;i<N2*8;i++){ objectInfo_data_byte } serviceContextList_count for(i=0;i<N3;i++){ serviceContextList_data_byte } MessageBody_length Content_length for(i=0;i<N4;i++){ Content_data_byte } }	4x8 8 8 8 8 32 8 8 32 4x8 16 64 8 8 8 32 32 8	uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf uimsbf	0x42494F50 0x01 0x00 0x00 0x00 N1 0x00000004 0x66696C00 N2 N3 N4	“BIOP” BIOP 主版本 1 BIOP 副版本 0 big_endian 字节排序 “fil”type_id 别名 objectInfo serviceContextList 实际文件内容

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 句法

句 法	比特数	类型	值	注 释
BIOP::StreamMessage(){				
Magic	4x8	uimsbf	0x42494F50	“BIOP”
biop_version.major	8	uimsbf	0x01	BIOP 主版本 1
biop_version.minor	8	uimsbf	0x00	BIOP 副版本 0
byte_order	8	uimsbf	0x00	big_endian 字节排序
message_type	8	uimsbf	0x00	
message_size	32	uimsbf		
objectKey_length	8	uimsbf	N1	
for(i=0;i<N1;i++){				
objectKey_data_byte	8	uimsbf		
}				
objectKind_length	32	uimsbf	0x00000004	
objectKind_data	32	uimsbf	0x73747200	“str”type_id 别名
objectInfo_length	16	uimsbf	N6	
DSM::Stream::Info_T{				objectInfo
aDescription_length	8	uimsbf	N2	aDescription
for(i=0;i<N2;i++){				
aDescription_bytes	8	uimsbf		
}				
duration.aSeconds	32	simsbf		AppNPT 秒
duration.aMicroSeconds	32	uimsbf		AppNPT 微秒
audio	8	uimsbf		
video	8	uimsbf		
data	8	uimsbf		
}				
for(i=0;i<N6-(N2+10);i++){				
objectInfo_byte	8	uimsbf		
}				
serviceContextList_count	8	uimsbf	N3	serviceContextList
for(i=0;i<N3;i++){				
serviceContextList_data_byte	8	uimsbf		
}				
messageBody_length	32	uimsbf		
taps_count	8	uimsbf	N4	
for(i=0;i<N4;i++){				
id	16		0x0000	未定义
use	16			见表 A.15
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 为 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 为 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 条的 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 句法

句 法	比特数	类型	值	注 释
BIOP::StreamEventMessage(){ magic version, major version, minor byte_order	4x8 8 8 8	uimsbf uimsbf uimsbf uimsbf	0x42494F50 0x01 0x00 0x00	“BIOP” BIOP 主版本 1 Big_endian 字节顺序

表 A. 16 (续)

句 法	比特数	类型	值	注 释
message_type	8	uimsbf		
message_size	32	uimsbf		
objectKey_length	8	uimsbf	N1	
for(i=0; i < N1; i++){ objectKey_data_byte	8	uimsbf		
}				
objectKind_length	32	uimsbf	0x00000004	
objectKind_data	4x8	uimsbf	0x73746500	“ste” type_id 别名
objectInfo_length	16	uimsbf	N6	
DSM::Stream::Info_T { aDescription_length	8	uimsbf	N2	aDescription
for(i=0; i < N2; i++){ aDescription_bytes	8	uimsbf		见 BIOP::StreamMessage()
}				
duration_aSeconds	32	simsbf		见 BIOP::StreamMessage()
duration_aMicroSeconds	16	uimsbf		见 BIOP::StreamMessage()
audio	8	uimsbf		见 BIOP::StreamMessage()
video	8	uimsbf		见 BIOP::StreamMessage()
data	8	uimsbf		见 BIOP::StreamMessage()
}				
DSM::Event::EventList_T { eventNames_count	16	uimsbf	N3	
for(i=0; i < N3; i++){ eventName_length	8	uimsbf	N4	
for(j=0; j < N4; j++){ eventName_data_byte	8	uimsbf		包括零终止符
}				
}				
}				
for(i=0; i=N6-(N2+10)-(2+N3 + sum(N4)); i++){ objectInfo_byte	8	uimsbf		
}				
serviceContextList_count	8	uimsbf	0x00	空的 serviceContextList
for(i=0; i < N3; i++){ serviceContextList_data_byte	8	uimsbf		
}				
messageBody_length	32	uimsbf		
taps_count	8	uimsbf	N5	
for(i=0; i < N5; i++){ id	16	uimsbf	0x0000	未定义
use	16	uimsbf		见表 A. 15
association_tag	16	uimsbf		

表 A. 16 (续)

句 法	比特数	类型	值	注 释
selector_length	8	uimsbf	0x00	没有 selector
eventIds_count	8	uimsbf	N3	=eventNames_count
for(i=0; i < N3; i++){ eventId	16	uimsbf		
}				

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 事件不与 SI 事件对应。应该是原来的意思。

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 13818-6:1998。一条 DownloadInfoIndication()消息可传送同一个 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,transaction_id 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 13818-6:1998 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 句法

句 法	比特数	类型	值	注 释
BIOP::ModuleInfo(){ ModuleTimeOut	32	uimsbf		
BlockTimeOut	32	uimsbf		

表 A. 17 (续)

句 法	比特数	类型	值	注 释
MinBlockTime	32	uimsbf		
taps_count	8	uimsbf	N1	
for(j=0; j < N1; j++){				
Id	16	uimsbf	0x0000	用户私有
Use	16	uimsbf	0x0017	BIOP_OBJECT_USE
association_tag	16	uimsbf		
selector_length	8	uimsbf	0x00	
}				
UserInfoLength	8	uimsbf	N2	
for(j=0; j < N2; j++){				
userInfo_data_byte	8	uimsbf		(包括零终止符)
}				
}				

BIOP::ModuleInfo 的语义如下：

- ModuleTimeOut 域给出了以 μs 为单位的超时时限,可用来判断获取一个模块的全部块是否超时；
- BlockTimeOut 域给出了以 μs 为单位的超时时限,可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 条中描述。

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::ServiceGateway-Info 结构。该结构的句法见表 A. 18。

表 A. 18 ServiceGatewayInfo() 句法

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

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 user info 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 13818-6:1998 中定义。各域的使用在 ISO/IEC 13818-6:1998 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—2010 系统解码器可用。DSM-CC_section 的句法在 ISO/IEC 13818-6:1998 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 DSMCC_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

表 A. 19 (续)

消息	table_id	table_id_extension	version_number	section_number	last_section_number
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他值。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

A. 8. 7. 1 PSI 的使用机制

ISO/IEC 13818-6:1998 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。

这一条对以下 GB/T 17975.1—2010 中的描述符的使用做了说明,它们提供了以上功能,见 ISO/IEC 13818-6:1998。

A. 8. 7. 2 Carousel_Identifier_Descriptor

轮播标识符描述符建立了节目与对象轮播的关联关系。Carousel_identifier_descriptor 的句法见表 A. 20,也可见 ISO/IEC 13818-6:1998。

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 句法

句 法	比特数	类型	值	注 释
carousel_identifier_descriptor(){ descriptor_tag	8	uimsbf	0x13	
descriptor_length	8	uimsbf		
carousel_id	32	uimsbf		

表 A. 20 (续)

句 法	比特数	类型	值	注 释
FormatId	8	uimsbf		FormatSpecifier 的注册的标识符
FormatSpecifier(){ FormatSpecifier_byte	8	uimsbf		见表 A. 21 N2 字节
} for(i=0;i < N1;i++){ private_data_byte	8	uimsbf		
} }				

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 carouselNSAPaddress 的使用,描述符 carousel_identifier_descriptor()同样是必须的,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 描述器 carousel_identifier_descriptor 中的 FormatSpecifier 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 所必须的域的集合,在 DSI 和 DII 消息中也可以找到。
		8	
		16	
		16	
		32	
		8	
		32	所有域的类型都是“uimsbf”。
		8	
		8	以秒为单位的超时时限
		8	
		8	业务网关对象的 Object key
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
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. 3 association_tag_descriptor

association_tag_descriptor 建立了 association_tag 与 PID 间的关联关系,见 ISO/IEC 13818-6:1998,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

句 法	比特数	类型	值	注 释
association_tag_descriptor(){ descriptor_tag descriptor_length association_tag	8 8 16	uimsbf uimsbf uimsbf	0x14	
use	16	uimsbf	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	uimsbf uimsbf uimsbf	0x08	DSI 的 transaction_id 为 DSI 定义的超时时限
else if(use == 0x0001) selector_length }	8	uimsbf	0x00	
else{ selector_length for(i=0; i < N1; i++){ selector_byte } } for(i=0; i < N2; i++){ private_data_byte } }	8 8 8	uimsbf uimsbf uimsbf	N1	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. 4 stream_identifier_descriptor

stream_identifier_descriptor 有效建立了 component_tag 与 PID 的关联,可替代(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 $LSB = \langle component_tag \rangle$, use 域的值 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. 5 deferred_association_tags_descriptor

一个对象轮播可使用多个 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

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

A. 8. 8 SI 和 PSI 中的信息

A. 8. 8. 1 数据广播标识

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)可替代 0x0007 使用。

A. 8. 8. 2 SI 描述符

具有上述值的 SI 的 data_broadcast_descriptor 可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. 3 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 条。

A. 8. 9 transactionId 值的分配与使用

对象轮播中 transactionId 的使用方法是从小数点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 消息可分布到其他 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他消息,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 13818-6:1998 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 也用于在其他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)
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 B.1。还留下很宽的范围(0x0100-0xFFFF)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 B.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	0x0100-0xFFFF
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。	

附 录 C
(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—2010 的基本多文种页结构见图 C.1。

C.1 控制码

范围在 0x80 到 0x9F 之间单字节控制码的分配见表 C.1。

表 C.1 单字节控制码

控 制 码	描 述
0x80~0x85	预留使用
0x86	字符强调开
0x87	字符强调关
0x88~0x89	预留使用
0x8A	CR/LF
0x8B~0x9F	用户定义

对于双字节的字符表,本附录在 GB 13000—2010 专用区 0xE080~0xE09F 范围内定义的控制码的分配见表 C.2。

表 C.2 双字节控制码

控 制 码	描 述
0xE080~0xE085	预留使用
0xE086	字符强调开
0xE087	字符强调关
0xE088~0xE089	预留使用
0xE08A	CR/LF
0xE08B~0xE09F	预留使用

C.2 字符表的选择

文本域可用非空格、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 0x20~0xFF 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—拉丁文),见图 C.1;
- 如果文本域第一个字节的数值在 0x01~0x05 之间,那么文本条目的剩余部分的编码见 GB/T 15273.1—1994;
- 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 $N(uimsbf)$ 表示了文本条目剩余部分数据使用 GB/T 15273.1—1994 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—2010 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—1980 及其扩展集进行编码;
- 对于文本条目的第一个字节,值 0x00、0x06~0x0F、0x14~0x1F 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
03	Combining Diacritical Marks 组合用发音符	Basic Greek 基本希腊文
04	Cyrillic 西里尔文	Greek Symbols and Coptic 希腊符号及哥普特文
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 天城文书(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
21	Letterlike Symbols 类似字母的符号	Number Forms 数字形式
22	Mathematical Operators 数学运算符	Arrows 箭头
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
25	Box Drawing 制表符	Enclosed Alphanumerics 带括号的字母数字
26	Miscellaneous Symbols 零杂符号	Block Elements 方块元素
27	Dingbats 丁贝符(示意符等)	Geometric Shapes 几何图形符
30	CJK Symbols and Punctuation CJK 符号和标点	Hiragana 平假名
31	Bopomofo 注音	katakana 片假名
32	Hangul Compatibility Jamo 朝鲜文兼容字母	CJK Miscellaneous CJK 零杂字符
33	Enclosed CJK Letters and Months 带括号的 CJK 字母及月份	
34	CJK Compatibility CJK 兼容字符	
3D	Hangul 朝鲜文	
3E	Hangul Supplementary-A 朝鲜文补充-A	
44	Hangul Supplementary-B 朝鲜文补充-B	
4D		
4E	CJK Unified Ideographs CJK 统一汉字	
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 拼音文字变形显现形式	
FC		
FD	Arabic Presentation Forms-A 阿拉伯文变形显现形式-A	
FE	Comb. Half Marks 半形组合用标志	CJK Compat Foms CJK 兼容形式
FF	Halfwidth and Fullwidth Forms 半形及全形字符	Small Form Variants 小写变体
		Arabic Presentation Forms-BP
		Specials 特殊字符

1) 用于符号的组合用区分标志。

■—留作将来标准化用

2) 阿拉伯文变形显现形式-B。

□—非图形字符

图 C.1 GB 13000—2010 的基本多文种平面总貌

附 录 D
(规范性附录)
数据广播前端的技术要求

D.1 前端功能要求

数据广播前端功能面至少满足下述要求：

- 系统输出码流数据封装方式应使用本规范定义的 5 种封装格式中的一种或多种封装格式，并符合第 4 章中的规定；
- 能够按照 GB/T 17975.1—2010 和本规范的规定设置封装码流所使用的 PID 值，并按照设置正确封装；
- 能够对 PSI 和 SI 相关可编辑内容进行设置，并能够根据设置和第 4 章中的要求在输出码流中正确插入 PSI 和 SI，主要可设置内容见表 D.1。

表 D.1 PSI 和 SI 中可设置内容

PSI 或 SI 表	可设置字段	备 注
PMT	PMT_PID	设置 PMT_PID 的同时应在 PAT 中正确体现
	program_number	取值应与对应业务的 service_id 一致
SDT	service_id	位于 service_descriptor 中
	service_name	
	service_type	
	service_provider	
	bouquet_name	位于 bouquet_name_descriptor 中

- 能够对播出总带宽进行设置，并根据设置控制输出带宽；
- 能够对输出码流中单个 PID 标识的数据码流带宽进行设置，并控制该 PID 所标识的数据码流的输出带宽。

D.2 前端数据接口要求

数据广播前端数据输出接口为 ASI 接口，应符合本规范的相关规定。

接口性能具体指标要求见表 D.2。

表 D.2 ASI 接口指标要求

序号	项 目	单位	指标要求
1	输出幅度	mV	[720,880]
2	上升时间(20%~80%)	ps	≤1200
3	下降时间(20%~80%)	ps	≤1200
4	确定性抖动	%	≤10

附 录 E
(规范性附录)
前端测量方法

E.1 前端测量条件

温度:15℃~35℃;
相对湿度:45%~75%RH;
大气压:86 kPa~106 kPa;
电源电压:198 V~242 V 交流,50 Hz±1 Hz。

E.2 前端测量框图

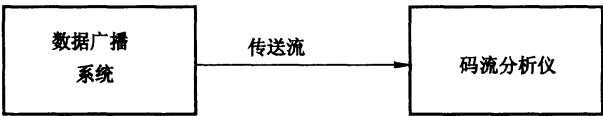


图 E.1 数字电视系统中数据广播功能和性能测量框图



图 E.2 数字电视系统中数据广播接口测量框图

E.3 前端测量设备

推荐使用表 E.1 给出的测量设备。

表 E.1 测量设备及要求

设备名称	要 求	数量
码流分析仪	能够对符合 GB/T 17975.1-2010、GB/T 28161—2011 及 GY/T 221—2006 第 4 章的 TS 流进行实时分析; 能够对符合 GB/T 17975.1—2010、GB/T 28161—2011 及 GY/T 221—2006 第 4 章的 TS 流文件进行离线分析	1
数字示波器	—	1

E.4 前端测量步骤

E.4.1 前端功能测量步骤

E.4.1.1 封装功能

- a) 按照图 E.1 连接测量系统;

- b) 用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测输出码流是否符合第4章中数据封装技术要求。

E.4.1.2 PID 设置功能

- a) 按照图 E.1 连接测量系统;
- b) 检查被测前端系统是否具有 PID 设置功能;
- c) 用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测输出 PID 是否与被测前端系统设置 PID 一致;
- d) 更改 PID,用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测更改后输出 PID 是否与被测前端系统设置 PID 一致。

E.4.1.3 PSI 和 SI 相关编辑设置功能

- a) 按照图 E.1 连接测量系统;
- b) 检查被测前端系统是否具有附录 D 中表 D.1 中所列项目设置功能;
- c) 用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测输出码流与附录 D 中表 D.1 中内容是否与被测前端系统设置内容一致;
- d) 更改被测前端系统附录 D 中表 D.1 中各项内容,用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测更改后各项内容一致是否与被测前端系统设置内容一致。

E.4.1.4 输出总带宽设置功能

- a) 按照图 E.1 连接测量系统;
- b) 检查被测前端系统是否具输出码流总码率设置功能;
- c) 用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测输出码流总码率是否与被测前端系统设置码率一致;
- d) 更改被测前端系统输出总码率,用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测更改后输出总码率是否与被测前端系统设置总码率一致。

E.4.1.5 单个 PID 对应数据码流带宽设置功能

- a) 按照图 E.1 连接测量系统;
- b) 检查被测前端系统是否具输出单个 PID 对应数据码流带宽设置功能;
- c) 用码流分析仪的实时码流分析功能分析被测前端系统的输出码流,检测输出码流单个 PID 对应数据码流带宽是否与被测前端系统设置带宽一致;
- d) 更改被测前端系统单个 PID 对应数据码流带宽,用码流分析仪的实时码流分析功能分析被测系统的输出码流,检测更改后单个 PID 对应数据码流带宽是否与被测前端系统设置单个 PID 对应数据码流带宽一致。

E.4.2 前端系统接口测量步骤

- a) 按照图 E.2 连接测量系统;
- b) 用数字示波器分析被测前端系统的输出信号,检测输出信号是否符合附录 D 中 D.2 的要求。

附 录 F
(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。

F.1 dsmccMessageHeader

dsmccMessageHeader 的句法见表 F.1。

表 F.1 dsmccMessageHeader 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 13818-6:1998 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。

F.2 dsmccDownloadDataHeader

dsmccDownloadDataHeader 的句法见表 F.2。

表 F.2 dsmccDownloadDataHeader 句法

句 法	字 节 数
<pre>dsmccDownloadDataHeader(){ protocolDiscriminator dsmccType messageId downloadId reserved adaptationLength messageLength if(AdaptationLength>0){ dsmccAdaptationHeader() } }</pre>	<div>1</div> <div>1</div> <div>2</div> <div>4</div> <div>1</div> <div>1</div> <div>2</div>

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 13818-6:1998 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。

F.3 DownloadServerInitiate

DownloadServerInitiate 消息的句法见表 F.3。

表 F.3 DownloadServerInitiate 消息的句法

句 法	字 节 数
<pre>DownloadServerInitiate(){ dsmccMessageHeader() serverId compatibilityDescriptor() privateDataLength for(i=0;i<privateDataLength;i++){ privateDataByte } }</pre>	<div>20</div> <div>2</div> <div>1</div>

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。

F.4 DownloadInfoIndication

DownloadInfoIndication 消息的句法见表 F.4。

表 F.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 μs 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 0xFFF0 到 0xFFFF 范围的值。

- privateDataLength 定义后面的私有数据的字节长度;
- privateDataByte 域由用户自定义。

F.5 DownloadDataBlock

DownloadDataBlock 的句法见表 F.5。

表 F.5 DownloadDataBlock 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 13818-6:1998 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;

附 录 G
(规范性附录)

将 DSM-CC 消息封装到 GB/T 17975.1—2010 段中

本附录说明了 DSM-CC 消息是怎样被封装到 GB/T 17975.1—2010 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—2010 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—2010 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 17975.1-2010 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 17975.1-2010 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—2010 中 2.4.4 条的 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)可以与其他 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 G.1。

表 G.1 DSM-CC 段格式的句法

句 法	比特数	类型
DSMCC_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
dsmcc_section_length	12	uimsbf
table_id_extension	16	uimsbf

表 G.1 (续)

句 法	比特数	类型
reserved	2	bslbf
version_number	5	uimsbf
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 H

(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。每个目录对象都可包含对其他目录(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,因此也不必要有其他约定。

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。

H.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,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。

H.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 Cos.Naming::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 Cos.Naming::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。

H.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 条给出的格式。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 H.1 使用的是 CORBA 格式字符串。

表 H.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

中 华 人 民 共 和 国
国 家 标 准
数字电视系统数据广播技术规范
GB/T 28430—2012

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100013)
北京市西城区三里河北街16号(100045)

网址 www.spc.net.cn

总编室:(010)64275323 发行中心:(010)51780235

读者服务部:(010)68523946

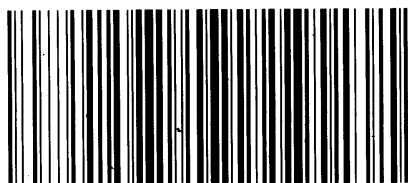
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

*

开本 880×1230 1/16 印张 5.5 字数 164 千字
2012年10月第一版 2012年10月第一次印刷

*

书号: 155066 • 1-45671 定价 72.00 元



GB/T 28430-2012