SCTE | STANDARDS

Digital Video Subcommittee

SCTE STANDARD

SCTE 42 2019 (R2024)

IP MULTICAST FOR DIGITAL MPEG NETWORKS

NOTICE

Document Tags

☒ Specification	☐ Checklist	☐ Facility
☐ Test or Measurement	☐ Metric	☒ Access Network
☐ Architecture or Framework	☒ Cloud	☐ Customer Premises
☐ Procedure, Process or Method		

Document Release History

Release	Date
SCTE 42 2002	*2002*
SCTE 42 2008	*2008*
SCTE 42 2019	*2019*
SCTE 42 2019 (R2024)	*4/29/2024*

Note: Standards that are released multiple times in the same year use: a, b, c, etc. to indicate normative balloted updates and/or r1, r2, r3, etc. to indicate editorial changes to a released document after the year.

This document is identical to SCTE 42 2019 except for informative components which may have been updated such as the title page, NOTICE text, headers and footers. No normative changes have been made to this document.

TABLE OF CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

1.1 SCOPE

This document is identical to SCTE 42 2019 except for informative components which may have been updated such as the title page, NOTICE text, headers and footers. No normative changes have been made to this document.

The document describes two methods to transmit multicast IP datagrams over MPEG 2 digital transport streams. It describes the use of Digital Video Broadcasting (DVB) Multi-Protocol Encapsulation (MPE) Datagram Sections and the Advanced Television Systems Committee's (ATSC) Addressable Sections, to encapsulate IP datagrams for subsequent segmentation into fixed length MPEG transport packets. It also describes how the encapsulated data will be included within an MPEG Program in a manner that allows a digital MPEG Decoder to efficiently locate the data PID streams carrying multicast IP content.

This document does not cover how the multicast IP data is used in conjunction with the audio / visual content of an MPEG program.

2 REFERENCES

2.1 NORMATIVE REFERENCES

[1] Information Technology - Generic coding of moving pictures and associated audio information, Part 6: Extensions for DSM-CC, ISO/IEC 13818-6: 1998/Amd. 1:2000(E)

[2] ISO/IEC 8802-1 and 2: "Information technology; Telecommunications and information exchange between systems; Local and metropolitan area networks; Specific requirements; Part 1: Overview of Local Area Network Standards, Part 2: Logical Link Control"

[3] Host Extensions for IP Multicasting, IETF RFC 1112, S. Deering, August 1989

[4] Assigned Numbers, IETF RFC 1700, J. Reynolds and J. Postel, October 1994

[5] ISO/IEC 13818-1 | ITU-T Rec. H.222.0:2007, Information Technology — Generic coding of moving pictures and associated audio — Part 1: systems.

[6] Digital Video Broadcasting (DVB); DVB Specification for data broadcasting, ETSI EN 301 192 v1.4.1 (2004-11)

[7] ATSC Data Broadcast Specification, ATSC A/90, December 2000 with Amendment 1 (14 May 2002) and Corrigendum 1 and 2 (1 April 2002)

2.2 INFORMATIVE REFERENCES

[8] SMPTE Standard 357M Declarative Data Essence, IP Multicast Encapsulation.

[9] Advanced Television Enhancement Forum Specification (ATVEF), Draft Version 1.1r26, updated 02/02/99

2.3 ACRONYMS AND ABBREVIATIONS

The following acronyms and abbreviations are used within this specification:

ATSC	Advanced Television Systems Committee
ATVEF	Advanced Television Enhancement Forum
bslbf	bit serial, leftmost bit first
CRC	Cyclic Redundancy Check
DSM-CC	Digital Storage Media Command and Control
DVB	Digital Video Broadcasting
FRAGnkj	IP fragmentation buffer for fragment identifier j, multicast address k, in elementary stream n.
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPM	IP Multicast
IPGRMBnk	IP datagram buffer for k-th IP multicast address in the n-th elementary stream
MAC	Media Access Control
MPE	Multi-Protocol Encapsulation
MPEG	Moving Picture Experts Group
MTU	Maximum Transmission Unit
PAT	Program Association Table
PDU	Protocol Data Unit
PID	Packet Identifier
PMT	Program Map Table
RFC	Request for comments
SBn	Smoothing Buffer for the n-th elementary stream
TBn	Transport Buffer for the n-th elementary stream
TS	Transport Stream
uimsbf	Unsigned Integer, Most Significant Bit First
UDP	User Datagram Protocol
UDPnkji	UDP buffer for port i, fragment identifier j, IP multicast address k, in elementary stream

2.4 GLOBAL TERMS

The following terms are used throughout this document:

bit rate: The rate at which the compressed bit stream is delivered from the channel to the input of a decoder.

bps: Bits per second.

byte-aligned: A bit in a coded bit stream is byte-aligned if its position is a multiple of 8-bits from the first bit in the stream.

CRC: The cyclic redundancy check used to verify the correctness of the data.

datagram: A datagram is the fundamental protocol data unit in a packet-oriented data delivery protocol. Typically, a datagram is divided into header and data areas, where the header contains full addressing information (source and destination IP addresses) with each data unit. Datagrams are most often associated with connectionless network and transport layer services.

elementary stream (ES): A generic term for one of the coded video, coded audio or other coded bit streams. One elementary stream is carried in a sequence of PES packets with one and only one stream_id.

IP Multicast data stream: A collection of IP packets with the same IP Multicast address and port number.

kbps: 1,000 bits per second.

Maximum Transmission Unit: The largest amount of data that can be transferred in a single unit across a specific physical connection. When using the Internet Protocol, this translates to the largest IP datagram size allowed.

Mbps: 1,000,000 bits per second.

MPEG-2: Refers to the collection of ISO/IEC standards 13818-1 through 13818-6.

Multiprotocol encapsulation: The encapsulation or splitting of datagrams in DSM-CC private sections.

packet: A packet is a set of contiguous bytes consisting of a header followed by its payload.

packet identifier (PID): A unique integer value used to associate elementary streams of a program in a single or multi-program transport stream.

payload: Payload refers to the bytes following the header byte in a packet.

program: A collection of Program Elements. Program Elements may be elementary streams. Program Elements need not have any defined time base; those that do have a common time base and are intended for synchronized presentation.

program element: A generic term for one of the elementary streams or other data streams that may be included in an ISO/IEC 13818-1 (MPEG-2) Program. The MPEG-2 Transport Stream packets conveying a Program Element are referenced by a unique PID value in the MPEG-2 Program.

Transport Stream: Refers to the MPEG-2 Transport Stream syntax for the packetization and multiplexing of video, audio, and data signals for digital broadcast systems.

3 MULTICAST IP DATA ENCAPSULATION

For IP datagram carriage, this standard defines two encapsulation protocols based on the DVB Multi-protocol encapsulation (MPE) and the ATSC/DSM-CC addressable sections. A given program shall carry IP data using either one of the encapsulation protocols, but not both. Changes to the protocol encapsulation may occur at any time within a program

The minor variations between MPE techniques are presented below. The overall transport of IP data over MPEG streams is specified in Section 4, in conjunction with the support of both MPE techniques.

3.1 DVB MPE DATAGRAM

This DVB MPE format is compliant with the DSM_CC Sections format for private data (see reference [1]). The table_id field shall be set to 0x3E, which indicates that this is a DSM_CC Section containing private data. However the DVB specification for data broadcasting (see reference [6]) utilizes table_id 0x3E to indicate MPE Datagram Sections.

The DVB MPE format listed in Table 1 is included for the reader's convenience only, and should not be considered a normative part of this specification. Consult reference [6] for the normative definition of the DVB MPE format.

Table 1. DVB MPE Datagram Section Format (Informative)

Syntax	No. of bits	Mnemonic
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	uimsbf
}		
}		

3.1.1 DVB DATAGRAM SECTION FIELD CONSTRAINTS (NORMATIVE)

The semantics of the following fields in the datagram_section are constrained as follows:

LLC_SNAP_flag: this is a 1-bit flag. This flag shall be set to "0" to indicate that the section contains an IP datagram without LLC/SNAP encapsulation.

3.2 ATSC MPE DATAGRAM

The ATSC MPE format is also compliant with the DSM_CC Sections format for private data. The table_id field shall be set to 0x3F, which is identical to a DSM_CC Section containing private data.

The ATSC MPE format listed in Table 2 is included for the reader's convenience only, and should not be considered a normative part of this specification Refer to reference [7] for the normative definition of ATSC MPE.

Table 2. ATSC MPE Datagram Section Format (Informative)

Syntax	No. of bits	Mnemonic
DSMCC_addressable_section() {		
table_id	8	0x3F
section_syntax_indicator	1	'0'
error_detection_type	1	bslbf
Reserved	2	'11'
section_length	12	uimsbf
deviceId[7…0]	8	uimsbf
deviceId[15…8]	8	uimsbf
reserved	2	'11'
payload_scrambling_control	2	bslbf
address_scrambling_control	2	bslbf
LLCSNAP_flag	1	'0'
current_next_indicator	1	'1'
section_number	8	uimsbf
last_section_number	8	uimsbf
deviceId[23…16]	8	uimsbf
deviceId[31…24]	8	uimsbf
deviceId[39…32]	8	uimsbf
deviceId[47…40]	8	uimsbf
for(j =0; j < N1; j++) {		
datagram_data_byte	8	bslbf
}		
if(section_number == last_section_number) {		
for(j=0;j<N2;j++) {		
stuffing_byte	8	bslbf
}		
}		
If(error_detection_type == 1) {		

checksum	32	uimsbf
} else {		
CRC	32	uimsbf
}		
}		

3.3 ATSC AND DVB COMPARISON (INFORMATIVE)

The ATSC and DVB multi-protocol encapsulation techniques vary in the following ways:

Table 3. MPE comparison

Field	ATSC	DVB	Note
table_id	0x3F	0x3E	can be used to differentiate types
section_syntax_indicator	0	1	values are distinctively different
CRC computation	error_detection_type =0	private indicator = 0	Different semantics but identical value
Destination	DeviceId [..]	MAC_address_n	different name but identical semantics
Data_byte	IP_datagram_data_byte	datagram_data_byte	different name but identical semantics

3.4 FORMULATION OF THE MAC_ADDRESS / DEVICEID (NORMATIVE)

RFC 1112 (see reference [3]) specifies the mapping from Multicast IP addresses to Ethernet MAC addresses. The MAC_address and deviceId fields shall be encoded in accordance with this mapping. The mapping is straight-forward and is as follows: the IP multicast address will be mapped into the corresponding hardware multicast address by placing the low-order 23 bits of the IP multicast address into the lower order 23 bits of the address 01:00:5E:00:00:00 (base 16). It is also noted that bit 23 (zero relative) of the MAC address will always be set to zero, per RFC 1700 (see reference [4]).

4 TRANSPORTING IP DATA OVER MPEG

Figure 1 illustrates how IP datagrams are encapsulated using either DVB or ATSC multi-procotol encapsulation and are then segmented into MPEG transport packets. IP datagrams shall be fragmented at the IP layer such that they do not exceed the specified Maximum Transfer Unit (MTU) for the payload portion of the MPE Datagram Section.

Figure 1 illustrates how an IP packet is fragmented for carriage in a datagram section that is no longer than 4080 bytes. The proposed datagram encapsulation allows datagrams as large as 4080 bytes.

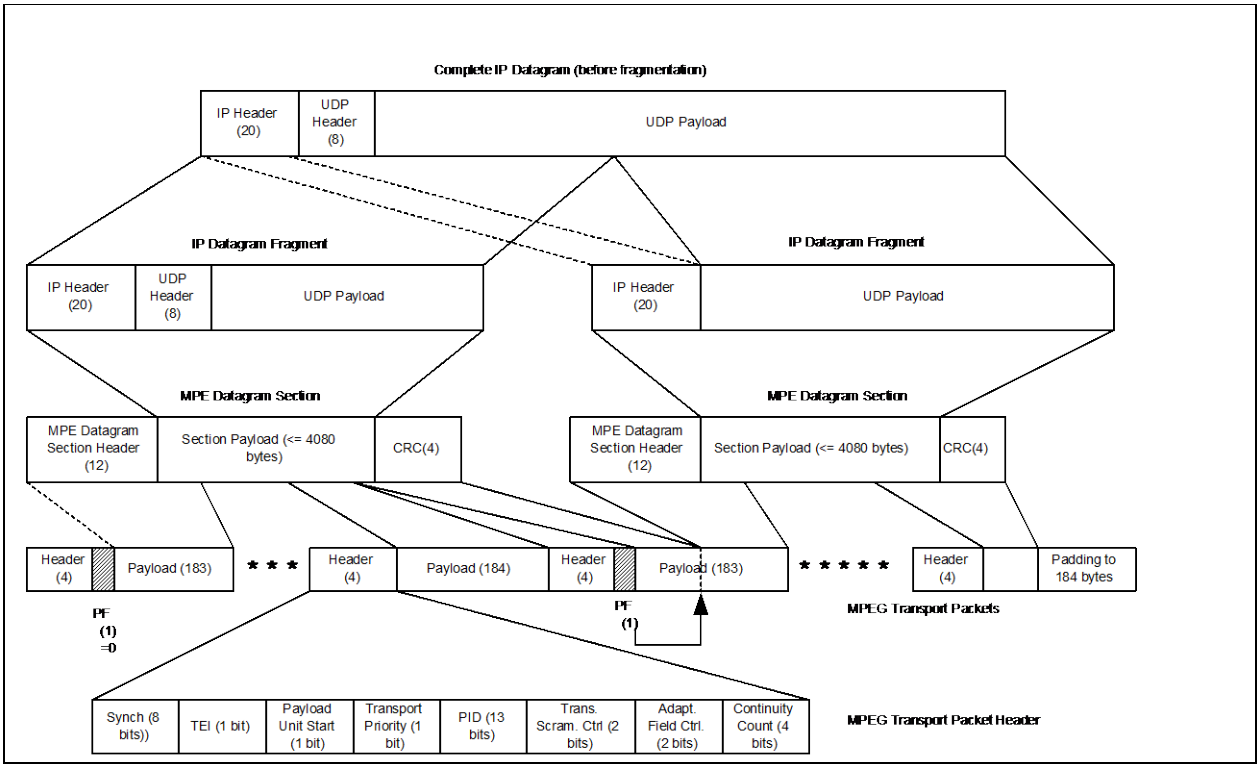


Figure 1. Carrying Multicast IP Datagrams over MPEG

A single Datagram Section may span many MPEG packets of the same PID. Also, messages may be placed back-to-back in the MPEG packet payload. This requires the use of the pointer_field (PF) byte to point to the location of the beginning of the next message as illustrated in Figure 1. The Payload Unit Start bit shall be 'set' in this case, meaning that the MPEG transport packet contains a pointer_field byte which points to the beginning of the new datagram section.

The MPEG packets associated with a particular PID value constitute a PID stream that will be multiplexed with audio, video, and possibly other data PID streams that form an MPEG Program. A Program Map Table (PMT), which identifies the component PIDs (audio, video, and data) that make up the MPEG program, shall be added. These elementary stream component(s), carrying IP data, shall be identified as containing DSM_CC Sections by assigning the appropriate stream_type value within the PMT. A new PMT descriptor shall be included that identifies the data, by multicast MAC group addresses, being carried by each data elementary stream component.

4.1 DSM-CC STREAM TYPE

Each elementary PID stream component that carries IP data shall have a **stream_type** of 0x0D associated with it within the PMT. This indicates that this PID stream carries DSM_CC Sections.

4.2 MAC_ADDRESS_LIST_DESCRIPTOR()

An MPEG Program that carries IP data of stream_type 0x0D (DSM-CC Sections), shall include a special descriptor within its PMT for each IP data component PID. The MAC_Address_List descriptor shall appear in the inner loop (i.e. ES info) of the PMT. This descriptor shall be processed by the Decoder to determine if a component PID stream carries IP data that is of interest to the receiver. The number of MAC addresses that can be carried in the descriptor is limited by the maximum length of a descriptor (255 bytes). In order to allow more IP addresses to be associated within an elementary (PID) stream, a form of the descriptor that specifies ranges of MAC addresses is allowed. This descriptor will allow the Decoder to quickly and efficiently identify the data PID stream carrying the requested IP data, without having to extract and re-assemble MPE messages in order to filter on MAC address.

The MAC Address list descriptor shall be present, even if the MAC addresses in a particular data stream are not known. Values of 0x000000000000 for the lowest_mac_address and 0xFFFFFFFFFFFF for the highest_mac_address shall be used. If more than one IP data stream is present in a program, a descriptor containing this data shall be included for each IP data stream. Although this provides no information that would assist hardware in the use of a PID filter extension for filtering on a required MAC address, some information, for example the IP encapsulation method, might be used to differentiate between IP data streams.

The MAC_Address_List descriptor is shown in Table 4:

Table 4. MAC_Address_List_descriptor()

Syntax	No. of bits	Mnemonic
MAC_Address_List_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf (L)
mac_addr_list	1	uimsbf
mac_addr_range	1	uimsbf
pdu_size	2	uimsbf {1024 bytes, reserved1, reserved2, 4096 bytes}
encapsulation_type	2	uimsbf {DVB, reserved1, reserved2, ATSC }
reserved	2	uimsbf
if (mac_addr_list == 1) {		
num_in_mac_list	8	uimsbf (m)
M = m*sizeof(mac_address)		
L = L – M		
for (i=0; i < m; i++) {		
mac_address	48	uimsbf
}		
}		
if (mac_addr_range == 1) {		
num_of_mac_ranges	8	uimsbf (n)
N = (n*sizeof(mac_address)*2)		
L = L – N		
for (i=0; i< n; i++) {		
highest_mac_address	48	uimsbf
lowest_mac_address	48	uimsbf
}		
}		
for (i=0; i< L – 1; i++) {		
private_data_byte	8	uimsbf
}		
}		

4.2.1 MAC_ADDRESS_LIST_DESCRIPTOR() FIELD DEFINITIONS

The semantics of this descriptor is as follows:

descriptor_tag: An 8-bit unsigned integer that defines the particular descriptor data structure. Value for the MAC_Address_List_descriptor() is 0xAC.

descriptor_length: An 8-bit unsigned integer number that defines the length of the descriptor immediately following the field. The length includes everything in the descriptor structure except the descriptor_tag and the length byte itself.

mac_addr_list: set if descriptor includes a set of destination MAC addresses.

mac_addr_range: set if descriptor includes a range of destination MAC addresses where the range is specified by the **highest_mac_address** and **lowest_mac_address** fields.

pdu_size: A 2 bit unsigned integer that indicates the maximum length of IP data fragments encapsulated in the associated DSM-CC stream. Table 5 describes the encoding of this field.

Table 5. Coding of the pdu_size field

Value	Pdu_size
00	Maximum 1024 bytes sections
01	Reserved
10	Reserved
11	Maximum 4096 byte sections

encapsulation_type: A 2 bit unsigned integer that describes the type of Multi-protocol encapsulation performed on sections. Table 6. Coding of encapsulation_type field 6 describes the encoding of this field.

Table 6. Coding of encapsulation_type field

Value	Encapsulation_type
00	DVB MPE
01	reserved
10	reserved
11	ATSC MPE

num_in_mac_list: An 8 bit unsigned integer that indicates the number of MAC addresses contained within this descriptor.

num_of_mac_ranges: An 8 bit unsigned integer that indicates the number of MAC address pairs with a high / low range contained within this descriptor.

mac_address: A 48 bit MAC group address.

highest_mac_address: A 48 bit group MAC address that is the largest carried by descriptor.

lowest_mac_address: A 48 bit group MAC address that is the smallest carried by descriptor.

private_data_byte: Additional information that may be used to identify the stream.

4.3 BUFFER MODEL

An application buffer shall collect the IP multicast data bytes out of the smoothing buffer SBn. The application buffer shall be used to re-assemble the UDP datagrams before they are made available to the application. Only IP multicast datagrams shall enter the application buffer. Other bytes are discarded and may be used to control the system. Bytes shall enter the application buffer at the rate specified by the sb_leak_rate (see references [6] and [7]) field in the smoothing buffer descriptor associated with the IP multicast program element. If no smoothing buffer descriptor is present, a default sb_leak_rate of 19.2 kbps shall be used.

The size of the IP Multicast application buffer shall be equal to 262144 bytes. The IP multicast application buffer shall not overflow. (If the application buffer is composed of a collection of buffers, then by implication, each individual buffer in the collection shall not overflow.)

The size of the smoothing buffer SBn shall be equal to 10,000 bytes (see reference [7]). Note that buffer SBn corresponds to smoothing buffer B in reference [6].

This corresponds to the acquisition of 65536 byte UDP datagrams on two distinct ports, with the assumption that one UDP datagram is being removed from the buffer while another one is being aggregated in the buffer. Bytes of a UDP datagram payload shall be removed out of the application buffer once it has been reconstructed in the application buffer. The removal rate is specified by the sb_leak_rate field in the smoothing buffer descriptor associated with IP multicast program element. If no smoothing buffer descriptor is present, a default sb_leak_rate of 19.2 kbps shall be used.

The application buffer corresponds to the collection of buffers IPGRMBnk, FRAGBnkj and UDPnkji defined in Annex C.

4.4 DEFINITION OF IP NETWORK

Internet hosts expect IP packets received on a single interface to be coordinated so that multicast IP address collisions do not occur. For the purposes of this specification, an IP network shall be defined as a single MPEG program. More precisely, the collection of MPEG elementary (PID) streams associated with an MPEG program shall be considered as a single interface, or IP network.

Annex A: Coordination of IP Address Ranges (Informative)

Internet hosts expect IP packets received on a single interface to be coordinated so that multicast IP addresses are not used for different purposes by different transmitting servers. The collection of MPEG elementary (PID) streams associated with an MPEG program should be considered as a single interface, or “network”, and it is therefore important that multiple server hosts contributing data to a single program should coordinate their multicast IP addresses so that collisions in usage do not occur. Figure 2 llustrates a typical multi-server configuration that is used in cable.

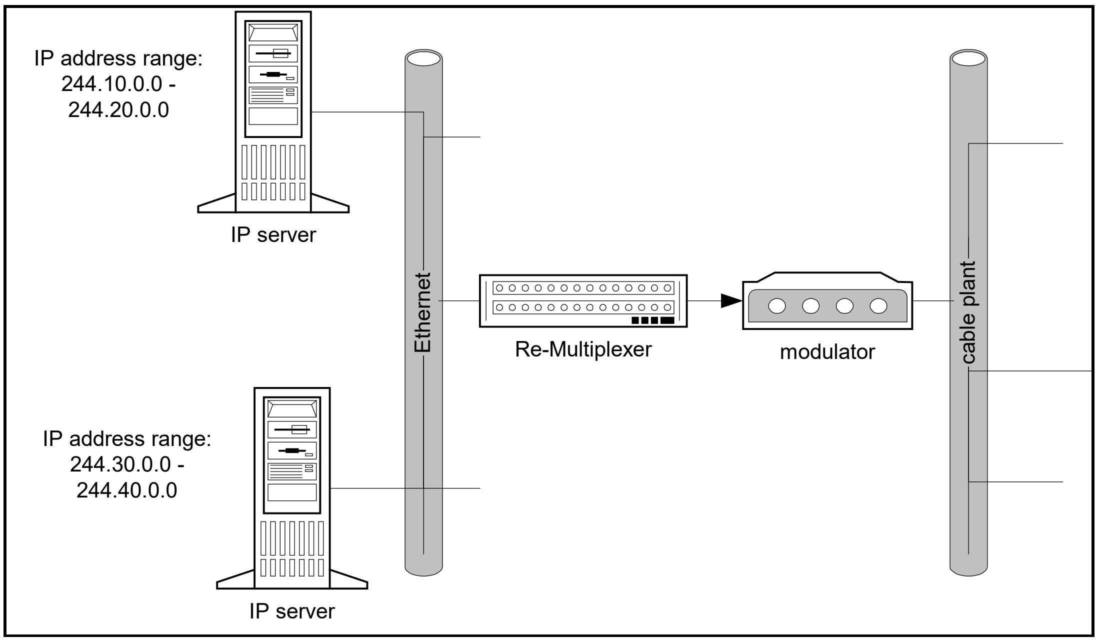


Figure 2. IP Multicast Servers

Since a single MPEG-2 program supports only one “network”, the importance of defining network boundaries comes from the fact that when multiple IP sources are injected into one network, the operator must ensure that there are no IP address conflicts. Note that no mechanism exists to automatically coordinate the use of IP multicast addresses and that this coordination must therefore come through private arrangements between the servers.

Annex B: Including ATVEF data within an MPEG Program (Informative)

SMPTE (reference [8]) and ATVEF (reference [9]) Transport Type B Broadcast data using IP Multicast binding are delivered to the application in three different components; announcements, triggers and resources. Announcements are delivered to the application on a well known Multicast IP address and UDP port. Each Announcement points to triggers and resources that are available to the application on specified Multicast IP addresses and UDP ports. See reference [9] for further details.

In most cases, the Announcements, Triggers and Resources will be included as part of the same data stream component as shown in Figure 3. This elementary data stream component will be associated with a particular program by identifying the component within the program's PMT. Data stream components of stream_type = '0x0D' (i.e. DSM_CC_Sections), may contain ATVEF IP data. The PMT will also carry a special descriptor (see Section 4.2) that will be used to assess if a particular data stream component carries multicast IP data. In addition, the descriptor will include the associated multicast MAC addresses of the MPE messages that encapsulate multicast IP data within the stream. The Decoder (or Digital Set-top Box) can use this information to quickly determine if the stream carries data destined for the currently tuned program, without having to continuously extract the IP data and pass it to the higher layers for filtering.

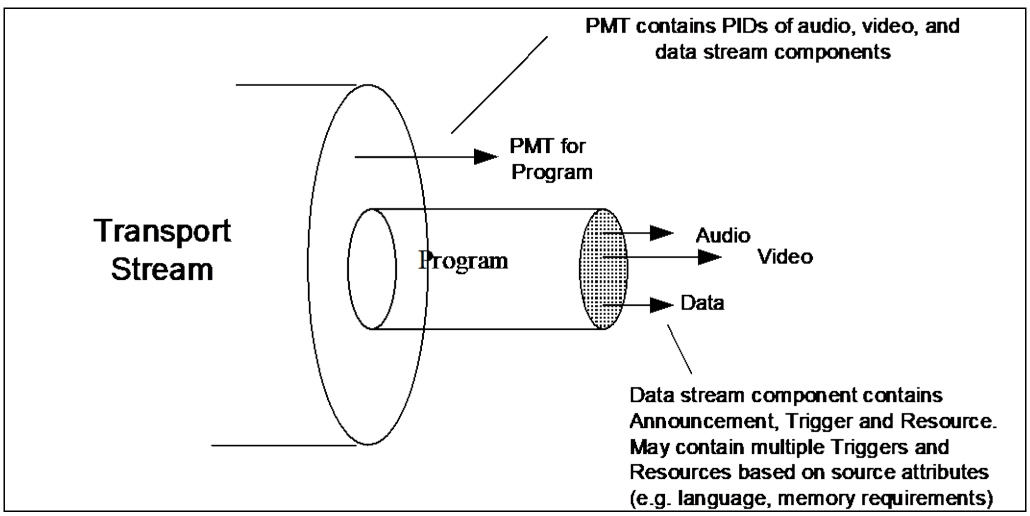


Figure 3. Announcement, Trigger and Resource Data Carried on a Single PID Stream

In other cases, it may be desirable to place the Announcements on their own component PID streams, whereas the Triggers and Resources are carried together or separately on another component PID stream(s) (See Figure 4). The advantage of this approach is that the decoder will not have to process IP datagrams, associated with triggers and resources, if the ATVEF application is not enabled to receive them. However, Announcements will always be processed, but are small in comparison.

Triggers and Resources, associated with a particular program, may also be delivered on separate PID streams based on their characteristics (e.g. language type, target audience/demographics content, size of resource, maximum bit rate requirements of content data). For example, an older version Decoder may not have enough memory to accommodate a particular resource or it may not be capable of receiving data above a certain maximum bit rate. In this case, the Decoder would connect to the stream carrying the smaller resource or one with lower bit rate requirements. The application may have to repeatedly try connecting to various resources of differing size or bandwidth requirements, until one can be acquired or no more are available.

It is also assumed that the ATVEF application will request a connection to trigger and resource streams based on selected user preferences (e.g. language preference). The ATVEF application may allow the user to enter the information directly or it could be automatically determined by making a call to the appropriate API Server. The application will use this information to choose the appropriate triggers/resources identified in an announcement. The application will then acquire the appropriate IP data streams.

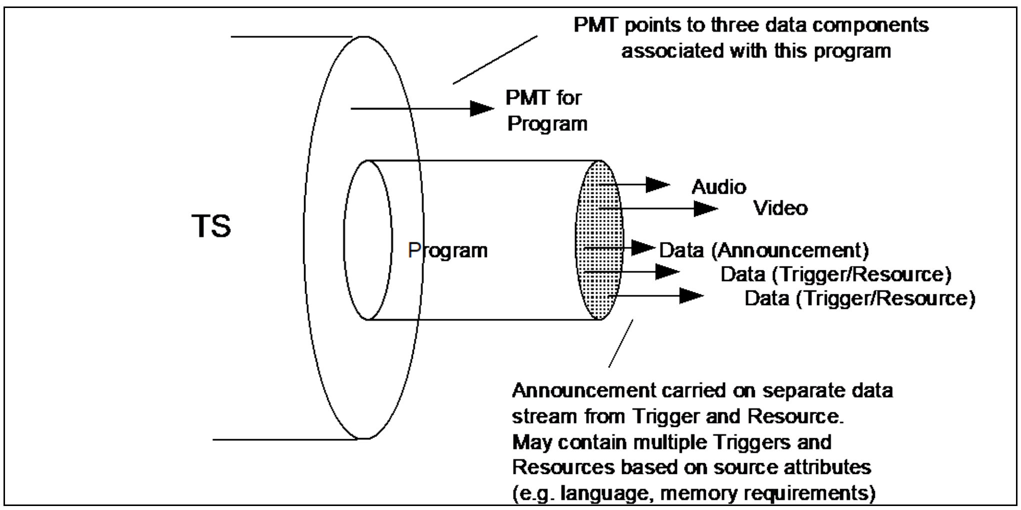


Figure 4. Announcement Carried Separately from Triggers and Resources

An extension of the case shown in Figure 4 is illustrated in Figure 5. Here, multiple MPEG programs may include elementary data PID streams that are shared with other programs. (The reader should note that IP address collison may occur if elementary data PID streams are shared across programs.) Each program's PMT would contain a PID reference that pointed to the stream carrying an ATVEF announcement (or the Announcement could be shared). Each PMT would also reference a common set of PID streams that were to be shared by each program. For example, two MPEG programs may wish to provide data from the same weather service. The set of common PID streams could carry Triggers and Resources of varying attributes that would be processed based on a particular Decoder's settings (e.g. language preference, memory resources, etc.).

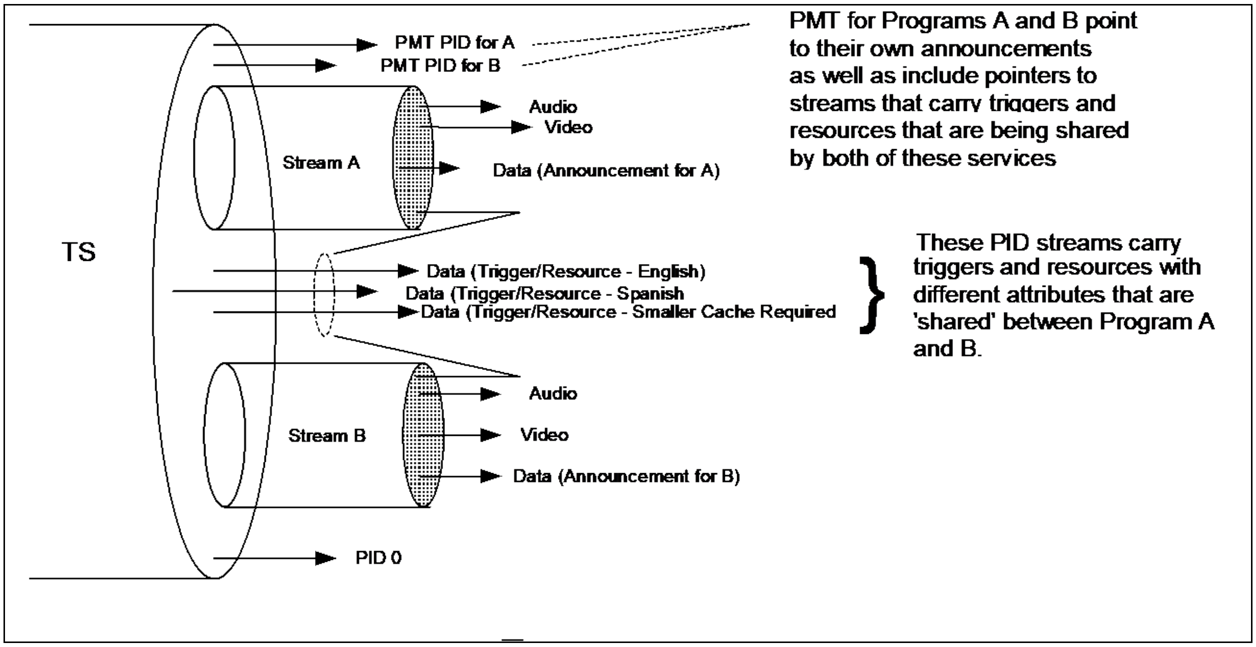
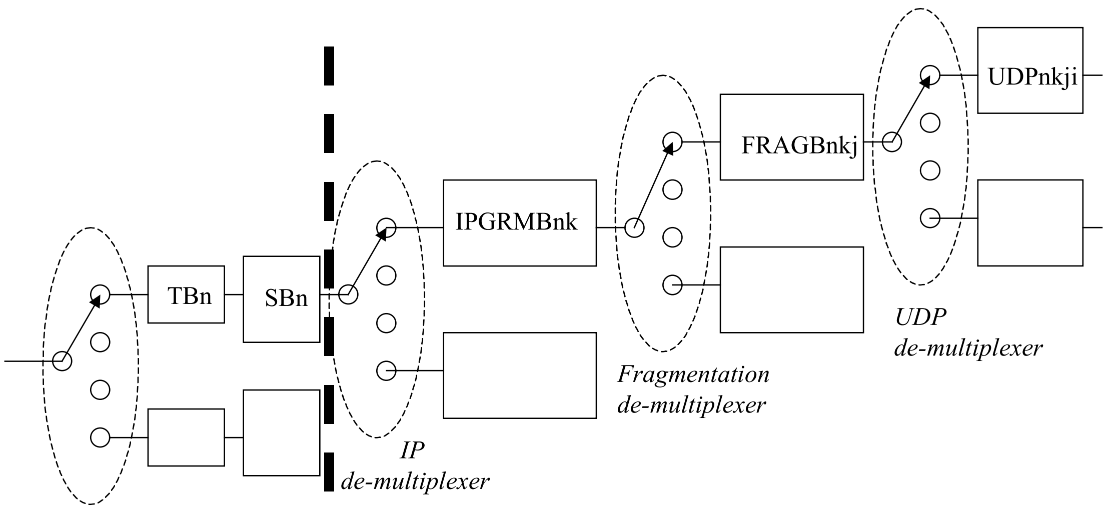


Figure 5. Sharing Triggers and Resources

Annex C: IP Multicast Buffer Model (Informative)

The application buffer model for acquisition of the IP Multicast datagrams is illustrated in Figure 6. The model assumes that datagrams are carried in MPE sections using either the DVB or ATSC encapsulation method.



TBn: is the Transport Buffer for data elementary stream n
SBn: is the smoothing buffer for data elementary stream n
IPGRMBnk : Is the IP datagram buffer for IP datagrams of multicast address k in data elementary stream n
FRAGBnkj: Is the IP fragmentation buffer for fragmentation identifier j for IP datagrams of multicast address k in data elementary stream n
UDPnkji: Is the UDP buffer for destination Port i for fragmentation identifier j for IP datagrams of IP multicast address k in data elementary stream n

Figure 6. Application Buffer Model for the acquisition of IP Multicast data

Complete Transport Stream packets containing data from Program Element n are passed to the transport buffer for data elementary stream n, TBn. The size of TBn is fixed at 512 bytes. All bytes that enter TBn are removed from TBn at a rate Rxn = 1.2 Rmax, where Rmax is equal to 26.97 Mbps. When there is no data in buffer TBn, rate RXn is equal to zero. Duplicate Transport Stream packets are not delivered to SBn. The Transport Stream buffer is not allowed to overflow.

Only the header of MPE section, payload and checksum/CRC_32 data bytes are delivered to buffer SBn; all other bytes do not enter SBn and may be used to control the system. The size of SBn is specified at 10,000 bytes. If there is data in buffer SBn, the data is taken out of SBn at a rate specified by the leak value specified by the smoothing buffer descriptor associated with the IPM Program Element n. If the smoothing buffer descriptor is not present, a default value of 19.2 kbps shall be used. The buffer SBn is not allowed overflow.

The IP Multicast datagram bytes in buffer SBn are all delivered to their associated IP datagrams buffer at the rate specified by the smoothing buffer descriptor associated with the IPM Program Element n, or the default rate if no smoothing buffer descriptor is present. Only IP datagram payload data bytes for IP Multicast address k enter buffer IPGRMBnk. Bytes from the MPE section header bytes carrying bytes of a datagram of IP multicast k in data elementary stream n are discarded and may be used to control the system. Bytes from the MPE section checksum/CRC_32 fields that immediately follow the last IP datagram byte are discarded and may be used to verify the integrity of the data. When there is no MPE section data present in SBn, no data is removed from SBn. All data that enters SBn leaves it. All MPE payload bytes of data elementary stream n enter the IP de-multiplexer instantaneously upon leaving SBn. MPE section header bytes and checksum/CRC32 bytes are removed instantaneously upon leaving SBn. The buffer IPGRMBnk is not allowed to overflow.

The payload bytes of the IP multicast datagrams in buffer IPGRMBnk are all delivered to their associated IP Fragmentation buffer at the rate specified by the smoothing buffer descriptor associated with the IPM Program Element n or the default rate if no smoothing buffer descriptor is present. Only IP datagram data bytes of fragment pertaining to identification j of IP datagram with IP multicast address k in data elementary stream n enter buffer FRAGnkj. Bytes from the IP datagram header are discarded and may be used to control the system and verify the integrity of the IP header. When there is no IP datagrams data present in IPGRMBnk, no data is removed from IPGRMBnk. All data that enters IPGRMBnk leaves it. All IP datagram payload bytes of IP multicast address k in data elementary stream n enter the Fragmentation de-multiplexer instantaneously upon leaving IPGRMBnk. IP datagram header bytes are removed instantaneously upon leaving IPGRMBnk. The buffer FRAGbnkj is not allowed to overflow.

The IP datagram fragment data bytes in buffer FRAGBnkj are used to reconstruct the original, non-fragmented, IP datagram payload. Last fragment pertaining to a re-constructed datagram may be identified by means of the control fragmentation flag bit in the IP datagram header. The payload bytes of the original, non-fragmented, IP datagram are all delivered to the UDP buffer at a rate specified by the smoothing buffer descriptor associated with the IPM Program Element n or default rate if no smoothing buffer descriptor is specified. Only UDP PDU payload data bytes of destination port i of the IP datagram re-constructed under fragmentation identifier j of the IP datagram with IP multicast address k in data elementary stream n enter buffer UDPnkji. Bytes from the UDP PDU header are discarded and may be used to control the system and may be used to verify the integrity of the IP pseudo-header. When there is no IP payload data bytes in the FRAGBnkj buffer, no data is removed from FRAGBnkj. All data that enters FLAGBnkj leaves it. All IP datagram payload bytes belonging to fragmentation of identification j of IP multicast address k in data elementary stream n enter the UDP de-multiplexer instantaneously upon leaving FRAGBnkj. UDP PDU header bytes are removed instantaneously upon leaving FRAGBnkj. The buffer UDPnkji is not allowed to overflow.

Once a UDP datagram has been fully reconstructed, the payload data bytes of the UDP PDU in buffer UDPnkji are all delivered at the rate specified by the smoothing buffer descriptor associated with the IPM Program Element n or default rate is no smoothing buffer descriptor is present. Only UDP PDU payload data bytes are delivered to the output at a rate specified by default or by the smoothing buffer descriptor associated with the IPM Program Element n. Other bytes are discarded and may be used to control the system. When there is no UDP PDU data present in UDPnkji, no data is removed from

UDPnkji. All data that enter UDPnkji leaves it. UDP PDU header bytes are removed instantaneously upon leaving UDPnkji. The buffer UDPnkji is not allowed to overflow.

The size of the IPGRMBnk, FRAGBnkj and UDPBnkji is IPGRMBSnk, FRAGBSnkj and UDPBSnkji, respectively.

