SMPTE 325M-1999

SMPTE STANDARD

for Digital Television — Opportunistic Data Broadcast Flow Control

1 Scope

This standard defines the flow control protocol to be used between an emission multiplexer and data server for opportunistic data broadcast. Opportunistic data broadcast inserts data packets into the output multiplex to fill any available free bandwidth. The emission multiplexer maintains a buffer from which it draws data to be inserted. The multiplexer will request additional MPEG-2 transport packets from the data server as its buffer becomes depleted. The number of packets requested depends upon the implementation, with the most stringent requirement being requesting a single MPEG-2 transport packet where the request and delivery can occur in less than the emission time of an MPEG-2 transport packet from the multiplexer. This protocol is designed to be extensible and provide a basis for low-latency, real-time backchannel communications from the emission multiplexer.

Encapsulated in MPEG-2 transport packets, the messages of the flow control protocol are transmitted via MPEG-2 DSM-CC sections, following the message format defined in ISO/IEC 13818-6, chapter 2. Such sections provide the capability to support error correction or error detection (or to ignore either).

2 Normative references

The following standards contain provisions, which through reference in this text constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ISO/IEC 13818-1:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems

ISO/IEC 13818-6:1998, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 6: Extensions for DSM-CC [digital storage media command and control]

3 Mnemonics

bslbf: Bit string, left bit first, where left is the order in which bit strings are written in this standard. Bit strings are written as a string of 1s and 0s within single quote marks; e.g., '1000 0001'. Blanks within a bit string are for ease of reading and have no sigiificance.

rpchof: Remainder polynomial coefficients, highest order first.

uimsbf: Unsigned integer, most significant bit first.

4 Flow control protocol

The first message defined in the flow control protocol is the FCPacketRequest() message. The FCPacketRequest() message allows the multiplexer to request the data server to send a specified number of packets associated with an opportunistic data service in order to maintain buffer fullness. The FCPacketRequest() messages shall be packaged in 188-byte MPEG-2 transport packets. The space remaining in the packet after the message shall be filled with 0xFF.

4.1 Transport packet

The MPEG-2 transport packet as defined in ISO/IEC 13818-1, clauses 2.4.3.2 and 2.4.3.3, shall be used with the following constraints as defined below (see table 1):

Approved
August 19, 1999

Table 1 – Transport packet

Syntax	No. of bits	Mnemonic
Transport_packet () {		
sync_byte	8	0x47
transport_error_indicator	1	'0'
payload_unit_start_indicator	1	'1'
transport_priority	1	'0'
PID	13	uimsbf
transport_scrambling_control	2	'00'
adaptation_field_control	2	'01'
continuity_counter	4	uimsbf
pointer_field	8	0x00
for (i = 0; i < N; i++) {		
data_byte	8	bslbf
}		
}		

4.1.1 Semantic definition of fields in transport packet

sync_byte: This field shall be set to 0x47.

transport_error_indicator: This field shall be set to ' 0'.

payload_unit_start_indicator: The payload_unit_start_indicator shall be set to a value of '1' to indicate the start of a private section within the current transport packet.

transport_priority: This field shall be set to ' 0.'

PID: The PID is a 13-bit field conveying the session number. The session number (PID) is mapped to the output service (a PID or aggregate of PIDs) during the provisioning process. The following PID ranges shall not be used: 0x0000-0x000F and 0x1FFB.

transport_scrambling_control: This field shall be set to '00.'

adaptation_field_control: This field shall be set to '01,' indicating no adaptation field, payload only.

continuity_counter: The continuity_counter is a 4-bit field that shall increment with each transport stream packet with the same PID. The continuity_counter wraps around to 0 after its maximum value.

pointer_field: This is an 8-bit field whose value shall be the number of bytes, immediately following the pointer_field until the first byte of the first section that is present in the payload of the transport stream packet (so a value of 0x00 in the pointer_field indicates that the section starts immediately after the pointer_field). In this usage, sections may not overlap within packets — so the pointer_field always has a value of 0x00.

data_byte: Data_bytes shall be contiguous bytes of data from the DSM-CC section defined below. Any data_bytes remaining in the transport packet following the checksum or the CRC_32 field of the DSM-CC section structure shall be set to 0xFF. The total number of data_bytes, N, within the transport packet is 183.

4.2 DSM-CC section for encapsulation of flow control protocol in transport packets

Flow control messages shall be encapsulated in the DSM-CC section structure defined in clause 9.2.2 of ISO/IEC 13818-6. The stream_type value associated with this elementary stream shall be 0x0D (which per ISO/IEC 13818-6, clause 9.2.3, indicates that the elementary stream consists of DSM-CC sections). A table_id value of 0xD7 shall be used to

Table 2 – Flow control message section

Syntax	No. of bits	Mnemonic
DSMCC_section () {		
table_id	8	0xD7
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	'11'
dsmcc_section_length	12	uimsbf
table_id_extension	16	0xFFFF
reserved	2	'11'
version_number	5	0x01
current_next_indicator	1	'1'
section_number	8	0x00
last_section_number	8	0x00
if (table_id) == 0xD7) {		
FlowControlMessage()		
}		
if (section_syntax_indicator) == '0') {		
checksum	32	uimsbf
} else {		
CRC_32	32	rpchof
}		
}		

signal the presence of flow control messages in DSM-CC sections (see table 2).

4.2.1 Semantic definition of fields in flow control message section

table-id: This is an 8-bit field which, in the case of flow control messages, shall be set to 0xD7.

section_syntax_indicator: This is a 1-bit indicator that when set to '1' shall indicate the use of a valid CRC_32 in the CRC_32/checksum field. When set to '0,' the bit indicates the use of a valid checksum in the CRC_32/checksum field. This field shall be set as defined by ISO/IEC 13818-6, clause 9.2.2.1.

private_indicator: This is a 1-bit flag that shall be set to the complement value of the section_syntax_indicator. This field shall be set as defined by ISO/IEC 13818-6, clause 9.2.2.1.

dsmcc_section_length: This field shall be set as defined by ISO/IEC 13818-6, clause 9.2.2.1.

table_id_extension: This 16-bit field value shall be set to 0xFFFF.

version_number: This field shall convey the version number of the protocol. The current standard defines version 1.

current_next_indicator: This is a 1-bit field that shall be set to '1.'

section_number: This field shall be set to 0x00.

last_section_number: This field shall be set to 0x00.

CRC_32: This field shall be used and set as specified in ISO/IEC 13818-6, clause 9.2.2.1.

checksum: This field shall be used and set as specified in ISO/IEC 13818-6, clause 9.2.2.1. In particular, if the checksum is set to 0, then no checksum is calculated.

4.2.2 Flow control messages

Flow control messages use the DSM-CC message format defined in chapter 2 of ISO/IEC 13818-6. The dsmccType in the message header shall be set to 0x80 for flow control messages. Table 3 defines the flow control message format. This format is called the FlowControlMessage(). The only flow control messages defined in this standard are the FCPacketRequest() messages.

4.2.2.1 Semantic definition of fields in flow control messages

The dsmccMessageHeader() is defined in 4.2.2.2.

The MessagePayload() is constructed from data fields and differs in structure depending on the function of the particular message as defined by the messageId.

4.2.2.2 DSM-CC message header

All flow control messages shall begin with the DSM-CC MessageHeader. This header contains information about the type of message being passed as well as any adaptation data which is needed by the transport mechamism. The DSM-CC message header is specified in table 4.

4.2.2.2.1 Semantic definition of fields in DSM-CC message header

protocolDiscriminator: This field is used to indicate that the message is an MPEG-2 DSM-CC message. The value of this field shall be 0x11.

dsmccType: This field is used to indicate the type of MPEG-2 DSM-CC message. Table 5, extending table 2 of ISO/IEC 13818-6, defines the possible dsmccType field values. The value 0x80 shall be used for flow control messages.

messageId: This field indicates the type of message which is being passed. The values of the messageId are defined within the scope of the dsmccType. Only the values in table 6 shall be used as values for the messageId field.

Table 3 – General format of flow control message

Syntax
FlowControlMessage () {
dsmccMessageHeader()
MessagePayload()
}

Table 4 – MPEG-2 DSM-CC message header format

Syntax	No. of bits	Mnemonic
dsmccMessageHeader() {		
protocolDiscriminator	8	0x11
dsmccType	8	0x80
messageId	16	uimsbf
transactionId	32	uimsbf
reserved	8	uimsbf
adaptationLength	8	uimsbf
messageLength	16	uimsbf
if (adaptationLength > 0) { dsmccAdaptationHeader() }		
}		

transactionId: This field shall be set to 0x40000000 for the version 1 protocol.

reserved: This field is ISO/IEC 13818-6 reserved. This field shall be set to 0xFF.

adaptationLength: This field shall be set to zero for the version 1 protocol. This means that there will not be an adaptation field used.

messageLength: This field shall be used to indicate the total length in bytes of the message following this field. This length shall include any adaptation headers indicated in the adaptationLength and the message payload indicated by the messageId field.

4.2.2.2.2. Packet request message

The FCPacketRequest message is defined in table 7. The messageID value associated with this message is 0x0001.

4.2.2.2.2.1 Semantic definition of fields in FCPacketRequest message

numberOfPackets: This 32-bit field shall specify the number of MPEG-2 transport packets requested by the mux from the server.

Table 5 – MPEG-2 DSM-CC dsmccType values

dsmccType	Description
0x00	ISO/IEC 13818-6 reserved
0x01	Identifies the message as an ISO/IEC 13818-6 IS user-to-network configuration message
0x02	Identifies the message as an ISO/IEC 13818-6 IS user-to-network session message
0x03	Identifies the message as an ISO/IEC 13818-6 IS download message
0x04	Identifies the message as an ISO/IEC 13818-6 IS SDB channel change protocol message
0x05	Identifies the message as an ISO/IEC 13818-6 IS user-to-network pass-through message
0x06 – 0x7F	ISO/IEC 13818-6 reserved
0x80	Identifies the message as an SMPTE 325M flow control message
0x81 – 0xFF	User defined message type

Table 6 – messageId values

messageId	Description
0x0000	Reserved
0x0001	Packet request
0x0002–0x00FF	Reserved for flow control
0x0100–0xFFFE	User defined message type
0xFFFF	Reserved

Table 7 – FCPacketRequest message

Syntax	No. of bits	Mnemonic
FCPacketRequest() {		
dsmccMessageHeader()		
numberOfPackets	32	uimsbf
}		

Annex A (informative)
Additional data

The environment for this standard is illustrated in figure A.1. Within an emission station, one or more data servers are providing broadcast data (contained within MPEG-2 transport packets, with appropriate protocol encapsulations) to an emission multiplexer. A real-time control path is available from the emission multiplexer to the data server for flow control messages. Opportunistic data broadcast will attempt to fill any bandwidth available in the emission multiplex with broadcast data on a nearly instantaneous basis. The emission multiplexer is in control of the opportunistic broadcast, since it is aware of the instantaneous gaps in the multiplex.

The operational model is that the emission multiplexer would maintain an internal buffer from which it could draw MPEG-2 data packets to insert into the emission multiplex as the opportunity arose. As the buffer emptied, the mux would request a number of packets from the data server to maintain buffer fullness over the control path. These data packets would be delivered over the data path. To avoid buffer overflow problems (should the data server be delayed in servicing the packet request), the following conventions are recommended: 1) the data server should not queue requests for a given service (that is, a new request will displace one that has not been acted upon), and 2) the emission multiplexer should request no more than half of its buffer size at a time.

While near-term implementations are likely to utilize at least a moderate sized buffer and request a significant number of packets each time, future-proofing should require the capability to handle the most stringent case, which is requests of single MPEG-2 transport packets, where the request can be made and satisfied within the time required to emit one transport packet from the mux.

Support for multiple opportunistic streams (multiple data servers, multiple opportunistic broadcasts from a single server) is provided by utilizing the MPEG-2 transport header PID as a session identifier. The mapping from PID number to physical resource shall be done during the provisioning process.

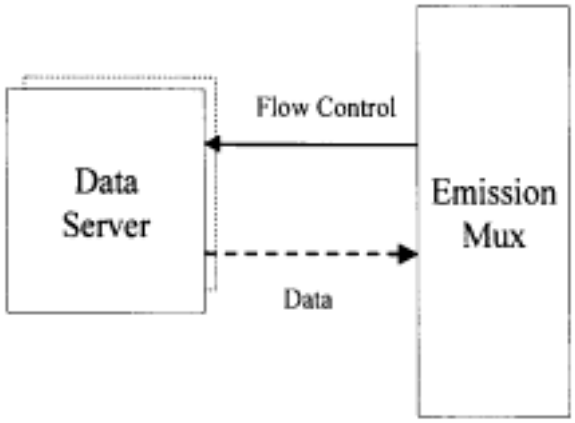

Figure A.1 – Opportunistic flow control environment