**3GPP TSG- S4 Meeting #118e *S4-220470***

**, 6 - 14 April 2022**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **Psuedo CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.517** | **CR** |  | **rev** | **-** | **Current version:** | **1.0.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **x** | Radio Access Network |  | Core Network | **x** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | [5MBP3]: Stage 3 Proposal for Clause 7 (Packet Distribution Method) | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Ericsson LM | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | 5MBP3 | | | | |  | ***Date:*** | | | 31.3.2022 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | B |  | | | | | ***Release:*** | | |  |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | This pCR starts proposing Stage 3 text for the Packet Distribution Method, based on existing text in TS 26.346 | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  |  | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  |  | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  |  | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

\*\*\*\* First Change \*\*\*\*

# 7 Packet Distribution Method

Editor’s Note:

* Specify the stage 3 protocols for the MBS distribution methods (between MBSTF and MBS Client) based on existing MBS delivery methods.
  + Object distribution method, based on or reference to clause 7 of TS 26.346.
* Agreements per S4-220023
  + the relevant delivery aspects of transparent delivery method, group communication delivery method and streaming delivery method as defined in TS 26.346, clause 8B, 8A and 8 respectively.
  + For the packet delivery method, it is proposed to only support the Transparent Delivery Method as defined in clause 8B, both the proxy and the forward-only mode. This includes RTP-based delivery as a special case.
  + The following functions are expected to be included:
    - Packet sequencing.
    - FEC.
    - QoS, bit rates.
    - Multiple flows?
    - Specific protocol support such as RTP/AVP.

## 7.1 Session description

### 7.1.0 Introdunction

SDP is provided to the MBS Client via a discovery/announcement procedure to describe the MBS Packet Distribution session. The SDP describes one or more RTP sessions part of the MBS Packet Distribution session. The SDP shall be a correctly formed SDP according to [x].

### 7.1.1 SDP Parameters for MBS Packet Distribution session

The semantics of a Session Description of an MBS Packet Distribution session shall include the parameters:

- The sender IP address.

- The number of media in the session.

- The destination IP address and port number for each and all of the RTP or UDP(FFS) sessions in the MBS Packet Distribution session.

- The start time and end time of the session.

- The protocol ID (i.e. RTP/AVP).

- Media type(s) and fmt-list.

- Data rate using existing SDP bandwidth modifiers.

- MBS Service Type of MBS session

- FEC configuration and related parameters.

- Service-language(s) per media.

- QoE Metrics (FFS)

#### 7.1.1.1 Sender IP address

There shall be exactly one IP source address per media description within the SDP. The IP source address shall be defined according to the source-filter attribute ("a=source-filter:") [x] for both IPv4 and IPv6 sources, with the following exceptions:

1. Exactly one source address may be specified by this attribute such that exclusive-mode shall not be used and inclusive-mode shall use exactly one source address in the <src-list>.

2. There shall be exactly one source-filter attribute per complete MBS Packet Distribution session SDP description, and this shall be in the session part of the session description (i.e. not per media).

3. The \* value shall be used for the <dest-address> subfield.

#### 7.1.1.2 Destination IP address and port number for channels

Each RTP session part of an MBS Packet Distribution session is defined by two parameters:

- IP destination address.

- Destination port number(s).

The IP destination address shall be defined according to the "connection data" field ("c=") of [x]. The destination port number shall be defined according to the <port> sub-field of the media announcement field ("m=") of [x]. Multiple ports using "/" notation shall not be used. The RTCP port, if used, shall be RTP port +1.

#### 7.1.1.3 Media Description

The media description line shall be used as defined in [14] for RTP. The <media> part indicates the type of media, audio, video, or text. The usage of RTP and any applicable RTP profile shall be indicated by using the <proto> field of the '*m-*line'. The one or more payload types that are being used in this RTP session are enumerated in the <fmt> part. Each payload type for RTP session is declared using the "a=rtpmap" attribute according to [x] and use the "a=fmtp" line when required to describe the payload format parameters.

#### 7.1.1.4 Session Timing Parameters

A MBS Packet Distribution session start and end times shall be defined according to the SDP timing field ("t=") - [14].

#### 7.1.1.5 MBS service type of MBS session

The MBS service type declaration attribute shall be used for MBS Packet Distribution sessions, as defined in sub-clause 6.2.2.8.

#### 7.1.1.6 Service-language(s) per media

The existing SDP attribute "a=lang" is used to label the language of any language-specific media. The values are taken from [x] which in turn takes language and (optionally) country tags from ISO 639 [x] and ISO 3166 [x] (e.g. "a=lang:EN-US"). These are the same tags used in the User Service Description XML.

#### 7.1.1.7 Bandwidth specification

The bit-rate required by the MBS Packet Distribution session and its media components shall be specified using both the "AS" bandwidth modifier and the "TIAS" bandwidth modifier combined with "a=maxprate" [x] on media level in the SDP. On session level the "TIAS" bandwidth modifier combined with "a=maxprate" may be used, where the session level expresses the aggregated peak bit-rate, which may be lower than the sum of the individual media streams.

The bandwidth required for RTCP is specified by the "RR" and "RS" bandwidth modifiers (3GPP TS 26.244 [x]) on media level for each RTP session. The "RR" modifier shall be included and set to 0 to specify that RTCP receiver reports are not used. The bandwidth used for RTCP sender reports shall be specified using the "RS" bandwidth modifier.

#### 7.1.1.8 FEC Parameters

The FEC encoding ID and instance ID are provided using the "a=FEC-declaration" attribute defined in sub-clause 6.2.2.9. Any OTI information for that FEC encoding ID and instance ID is provided with below defined FEC OTI attribute.

The FEC OTI attribute must be immediately preceded by the "a=FEC-declaration" attribute (and so can be session-level and media-level). The fec-ref maps the oti-extension to the FEC-declaration OTI it extends. The purpose of the oti-extension is to define FEC code specific OTI required for RTP receiver FEC payload configuration; exact contents are FEC code specific and need to be specified by each FEC code using this attribute. The OTI for the MBS FEC Scheme is defined in sub-clause 7.2.2.2.10a.

The syntax for the attributes in ABNF [x] is:

- sdp-fec-oti-extension-line = "a=FEC-OTI-extension:" fec-ref SP oti-extension CRLF

- fec-ref = 1\*3DIGIT (the SDP-internal identifier for the associated FEC-declaration).

- oti-extension = base64

- base64 = \*base64-unit [base64-pad]

- base64-unit = 4base64-char

- base64-pad = 2base64-char "==" / 3base64-char "="

- base64-char = ALPHA / DIGIT / "+" / "/"

To provide the FEC repair packets with additional, non FEC specific parameters, a session and media level SDP attribute is defined.

- sdp-fec-parameter-line = "a=mbs-repair: 0\*1SP fec-ref SP parameter-list CRLF

- parameter-list = parameter-spec \*(1\*SP parameter-spec)

- parameter-spec = name "=" value;

- name = 1\*(ALPHA / DIGIT / "-")

- value = 1\*(safe) ; safe defined in [x]

Currently one FEC non code-specific parameter is defined:

**min-buffer-time**: This FEC buffering parameter specifies the minimum receiver buffer time (delay) needed to ensure that FEC repair has time to happen regardless of the FEC source block of the stream from which the reception starts. The value is in milliseconds and represents the wallclock time between the reception of the first FEC source or repair packet of a FEC source block, whichever is earlier in transmission order, and the wallclock time when media decoding can safely start.

The parameters name and value is defined in ABNF as follows:

Min-buffer-time-parameter-name = "min-buffer-time"

Min-buffer-time-parameter-value = 1\*8DIGIT ;Wallclock time in milliseconds.

The FEC declaration and FEC OTI information utilized in a specific source or repair packet is indicated using the FEC-ref number in the a=fec lines as described in sub-clauses 7.2.2.2.12 and 7.2.2.2.13.

#### 7.1.1.9 FEC Flow ID attribute

To indicate the mapping between destination IP address and UDP port number and FEC source block flow IDs, the "a=mbs-flowid" SDP attribute is defined. Each flowID that is used to construct a source block within the bundled sessions shall be included. It is a media level attribute that shall be present in any SDP media block using the "UDP/MBS-REPAIR" protocol identifier.

The syntax for the attributes in ABNF [x] is:

Sdp-mbs-flowid-attr = "a=mbs-flowid:" \*WSP flow-id-spec \*("," \*WSP flow-id-spec) CRLF

flow-id-spec = flowID "=" address-spec "/" port-spec

address-spec = IP4-multicast / IP6-multicast

IP4-multicast = m1 3\*( "." decimal-uchar )

m1 = ("22" ("4"/"5"/"6"/"7"/"8"/"9")) / ("23" DIGIT ))

IP6-multicast = hexpart

hexpart = hexseq / hexseq "::" [ hexseq ] /

"::" [ hexseq ]

hexseq = hex4 \*( ":" hex4)

hex4 = 1\*4HEXDIG

port-spec = 1\*5DIGIT

#### 7.1.1.10 Buffer Requirement Signaling

Due to the variable bitrate nature of some media streams (especially video streams), initial buffering at the receiver becomes necessary to smooth out those variations. The initial buffering delay SHOULD be signaled to the receiver in the SDP using the following media level attribute:

- "a=X-initpredecbufperiod:<initial pre-decoder buffering period>"

For H.264 video streams, the "X-initpredecbufperiod" [47] indicates the nominal removal time of the first access unit from the coded picture buffer (CPB).

For H.265 (HEVC) video streams, the "X-initpredecbufperiod" [47] indicates the nominal removal time of the first decoding unit from the coded picture buffer (CPB).

Note that X-initpredecbufperiod is expressed as clock ticks of a 90-kHz clock. Hence, conversion may be required if the RTP timestamp clock frequency is not 90 kHz.

#### 7.1.1.11 Interleaving Signaling

When interleaving is used in combination with FEC protection of an MBS service, the MBSF may indicate to receivers the order of transmission of the media units of a source block using the "X-3gpp-FEC-Interleaving" attribute. It also indicates whether intra-stream interleaving (described in section G.1) has been performed or not for each of the flows in the FEC source block.

The "X-3gpp-FEC-Interleaving" attribute is defined as follows:

- Interleaving="X-3gpp-FEC-Interleaving:" SP flow\_interleaving \*("," flow\_interleaving) CRLF

- flow\_interleaving=flowID "=" ["ordered" / "mixed" / "reverse"]

flowID is the indentification of the flow as described in section 7.1.1.9. The intra-stream interleaving modes may result in un-changed tranmission order ("Ordered"), a mixed transmission order ("Mixed"), or a reversed transmission order ("Reverse"). For a flow that is not listed in the X-3gpp-FEC-Interleaving attribute, the receiver should assume that no particular intra- or inter-stream interleaving has been performed. The transmission order does not preculde that some media units of a lower priority stream are interleaved with the media units of higher priority stream.

#### 7.1.1.12 Transport Framing Protocol

The "mbs-framing-header" or "mbs-framing-trailer" attributes, if present at the media level, indicate that MBS transport framing protocol is used and provides a version number and length field for the transport framing header or trailer, respectively.

The ABNF sytax for the "mbs-framing-header" and the "mbs-framing-trailer" attributes is as follows:

mbs-framing=("a=mbs-framing-header" / "a=mbs-framing-trailer") ":" SP version SP length SP parameters  
 version = 1\*2DIGIT  
 length = 1\*2DIGIT  
 parameters = \*(parameter [";"])  
 parameter = name "=" value  
 name = \*(ALPHA / DIGIT / "|" / "-" / "\_")  
 value = \*(ALPHA / DIGIT / "|" / "-"/ "\_")

### 7.1.2 SDP Example for a RTP streaming

Here is a full example of SDP description describing the media streams part of an MBS Packet Distribution session for RTP streaming:

v=0  
o=ghost 2890844526 2890842807 IN IP4 192.168.10.10  
s=3GPP MBS Packet Distribution SDP Example  
i=Example of MBS Packet Distribution SDP file  
u=http://www.infoserver.example.com/ae600  
e=ghost@mailserver.example.com  
*c=IN IP6 FF1E:03AD::7F2E:172A:1E24*  
t=3034423619 3042462419

b=AS:77

a=mbs-mode:broadcast 123869108302929 1

a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9

m=video 4002 RTP/AVP 96

b=TIAS:62000

b=RR:0

b=RS:600

a=maxprate:17

a=rtpmap:96 H264/90000  
a=fmtp:96 profile-level-id=42A01E; packetization-mode=1; sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==  
m=audio 4004 RTP/AVP 98

b=TIAS:15120

b=RR:0

b=RS:600

a=maxprate:10

a=rtpmap:98 AMR/8000

a=fmtp:98 octet-align=1

FEC is not used in that example. See clause 7.2.2.x for an example with FEC.

### 7.1.3 SDP Examples for a transparent streaming

The following is a full example of SDP description for transparent streaming with 2 MPEG-2 TS:

v=0  
o=ghost 2890844526 2890842807 IN IP4 192.168.10.10  
s=3GPP MBS Transport-only SDP Example  
i=Example of MBS transport-only SDP file  
u=http://www.infoserver.example.com/ae600  
e=ghost@mailserver.example.com  
*c=IN IP6 FF1E:03AD::7F2E:172A:1E24*  
t=3034423619 3042462419

b=AS:8000000

a=mbs-mode:broadcast 123869108302929

a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9

m=video 4002 UDP/RTP/AVP 96

b=TIAS:4000000

a=mms-framing-header:0 2

a=rtpmap:100 MP2T/90000

m=video 4002 RTP/AVP 98

b=TIAS:4000000

a=rtpmap:100 MP2T/90000

a=MBS-framing-trailer:0 2

## 7.2 Protocols

Include, RTP Streaming and transparent streaming

The purpose of the MBS Packet Distribution Method is to deliver continuous multimedia data (i.e. speech, audio and video) over an MBS session. The MBS Packet Distribution Method is particularly useful for multicast and broadcast of scheduled streaming content.

### 7.2.1 RTP Streaming

#### 7.2.1.0 RTP payload formats for media

RTP is the transport protocol for continuous media streaming using the MBS Packet Distribution Method. RTP provides means for sending real-time or streaming data over UDP and is already used for the transport of PSS in 3GPP. RTP provides RTCP for feedback about the transmission quality. The transmission of RTCP packets in the downlink (sender reports) is allowed. In this version of the specification, RTCP RR shall be turned off by SDP RR bandwidth modifiers. Note that in the context of MBS, detection of link aliveness is not necessary.

#### 7.2.1.1 RTP payload formats for media

The RTP payload formats and corresponding MIME types are closely aligned with those defined in PSS [x] . For RTP/UDP/IP transport of continuous media the following RTP payload formats shall be used:

- AMR narrow-band speech codec (see sub-clause x) RTP payload format according to RFC 4867 [x]. A MBS Client is not required to support multi-channel sessions.

- AMR wideband speech codec (see sub-clause x) RTP payload format according to RFC 4867 [x]. An MBS Client is not required to support multi-channel sessions.

- Extended AMR-WB codec (see sub-clause x) RTP payload format according to [x].

- Enhanced aacPlus codec (see sub-clause x) RTP payload format and MIME types according to RFC 3640 [x], namely the Low Bit-Rate AAC or the High Bit-Rate AAC modes.

- H.264 (AVC) video codec (see sub-clause x) RTP payload format according to [x]. An MBS Client supporting H.264 (AVC) is required to support all three packetization modes: single NAL unit mode, non-interleaved mode and interleaved mode. For the interleaved packetization mode, an MBS Client shall support streams for which the value of the "sprop-deint-buf-req" MIME parameter is less than or equal to MaxCPB \* 1000 / 8, inclusive, in which "MaxCPB" is the value for Video Coding Layer (VCL) parameters of the H.264 (AVC) profile and level in use, as specified in [x].

- H.265 (HEVC) [112] video codec (see clause x) RTP payload format according to [x].

- Timed Text (see sub-clause x) RTP payload format according to [x].

#### 7.2.1.2 FEC mechanism for RTP

##### 7.2.1.2.0 General

The "MBS FEC scheme" is the fully-specified FEC scheme defined in [x], section 6 with ID 1.

The source flows for the MBS FEC scheme are UDP flows including RTP, RTCP, SRTP and MIKEY packets. The payload of such UDP packets constitute an Application Data Unit (ADU) as defined in RFC 6363 [x]. The source data flow with which the ADUs are associated is the UDP flow identity of the corresponding UDP flow.

A UE that supports MBS User Services shall support a decoder for the "MBS FEC scheme". The use of MBS FEC by the sender is recommended, but it is permitted not to use it. In the case where the FEC is not used by the sender, the FEC Layer should not be used (i.e. RTP is mapped onto UDP directly).

The mechanism does not place any restrictions on the source data which can be protected together, except that the source data is carried over UDP. The data may be from several different UDP flows that are protected jointly.

A UE supporting the MBS Packet Distribution Method shall support the packet format for FEC packets..

If any FEC source packets have been lost, but sufficient FEC source and FEC repair packets have been received, FEC decoding can be performed to recover the FEC source block. The original packets UDP payload and UDP flow identity can then be extracted from the source block and provided to the upper layer. If not enough FEC source and repair packets were received, only the original packets that were received as FEC source packets will be available. The rest of the original packets are lost.

If a UE that supports MBS User Services receives a mathematically sufficient set of encoding symbols generated according to the encoder specification in RFC 5053 [x], section 5.3, for reconstruction of a source block, then the decoder shall recover the entire source block. Note that the example decoder described in [x] clause 5.5 fulfils this requirement.

Note that the receiver must be able to buffer all the original packets and allow time for the FEC repair packets to arrive and FEC decoding to be performed before media playout begins. The min-buffer-time parameter specified in sub-clause .7.1.1.8 helps the receiver to determine a sufficient duration for initial start-up delay.

The protocol architecture is illustrated in figure 11.



Figure 11: FEC mechanism for the RTP Streaming of MBS Packet Distribution Method interaction diagram

Figure 11 depicts how one or more out of several possible packet flows of different types (Audio, video, text RTP and RTCP flows, MIKEY flow) are sent to the FEC layer for protection. The source packets are modified to carry the FEC payload ID and a new flow with repair data is generated. The receiver takes the source and repair packets and buffers them to perform, if necessary, the FEC decoding. After appropriate buffering received and recovered source packets are forwarded to the higher layers. The arrows in the figure indicate distinct data flows.

##### 7.2.1.2.1 Sending Terminal Operation (Informative)

It is assumed that the sender has constructed or received original data packets for the session. These may be RTP, RTCP, MIKEY or other UDP packets. The following procedures are based on the UDP payload and the identity of the UDP flow. The UDP payload constitutes and ADU according to RFC 6363 [107] and the identity of the UDP flow is the integer identifier associated with the identifier of the ADU flow.

In order to FEC protect a sequence of original data packets, the sender constructs a source block as specified in RFC 6681 [106], section 5 to which the FEC algorithm is to be applied, and includes the original source packet data within FEC source packets. The following operations describe a possible way to generate compliant FEC source packet and FEC repair packet streams:

1. Each original packet is placed in the source block. In doing so, the Source FEC Payload ID information to be included in the FEC payload ID of the FEC source packet can be determined. In the source block the identity of the packet’s flow is marked using the Flow ID. See RFC 6681 [106], section 5 for details.

2. The FEC source packet is constructed according to sub-clause 7.2.2.2.4. The identity of the original flow is maintained by the source packet through the use of the destination UDP port number and destination IP address, which has been advertised (for example using SDP), as carrying FEC source packets generated from an original stream of a particular protocol (e.g. RTP, RTCP, SRTP, MIKEY etc.). See sub-clause 7.2.2.2.13.

3. The generated FEC source packet is sent using UDP.

When a source block is complete, the FEC encoder generates encoding symbols and places these symbols into FEC repair packets, to be conveyed to the receivers. These repair packets are sent using normal UDP procedures to a unique destination port to separate it from any of the source packet flows.

In particular cases it may be advantageous not to use FEC for some source blocks and to signal this to the receiver. In this case the sender may send one or more empty repair packets consisting exclusively of the Repair FEC Payload ID. This will be helpful in particular for selective FEC where some of the source blocks (e.g. consisting of reference video frames) are FEC protected while others (e.g. consisting exclusively of non-reference frames) will not be protected.

##### 7.2.1.2.2 Receiving Terminal Operation (Informative)

The following describes a possible receiver algorithm, when receiving an FEC source or repair packet:

1. If a FEC source packet is received (as indicated by the UDP port on which it was received):

a. The original source packet is reconstructed by removing the Source FEC Payload ID. The resulting packet is buffered to allow time for the FEC repair.

b. The resulting packet is placed into the source block according to the information in the Source FEC Payload ID and the source block format described in [106], section 5. The UDP port the packet was received on is used to determine the Flow ID written into the source block.

2. If an FEC repair packet is received (as indicated by the UDP port), the contained encoding symbols are placed into an FEC encoding block according to the Repair FEC Payload ID. In case the received FEC repair packet is empty, there are no repair symbols to be placed in the FEC encoding block.

3. If at least one source packet is missing, then FEC decoding may be desirable. The FEC decoder determines if the encoding block constructed in steps 1 and 2 contains enough symbols from the source and repair packets for decoding and, if so, performs the decoding operation. If only empty FEC repair packets are received, the receiver may start immediately some procedures to conceal the effect of missing media data.

4. Any missing source packets that were reconstructed during the decoding operation are then buffered as normal received packets (see step 1a above).

Note that the above procedure may result in that not all original packets are recovered, and they must simply be marked as being lost.

Obviously, buffering and packet re-ordering are required to insert any reconstructed packets in the appropriate place in the packet sequence if that is necessary according to the used higher layer protocol (RTP, RTCP or MIKEY). To allow receivers to determine the minimal start-up buffering requirement for FEC decoding, the min-buffer-time parameter indicates a minimum initial buffering time that is sufficient regardless of the position of the stream in which the reception starts.

##### 7.2.1.2.4 Packet format for FEC source packets

The packet format for FEC source packets as defined in RFC 6363 [x], section 5.3, shall be used to encapsulate an original UDP packet.

The destination IP address and UDP port shall be set as indicated in the session control signalling. This ensures that the receiver can determine which protocols and FEC Payload ID formats are used for this flow. The remaining fields in the IP and UDP headers shall be set according to their specifications.

The Source FEC Payload ID shall be constructed according to RFC 6681 [x], section 6.2.2.

The FEC Source packets over IP and UDP are indicated to be used for a flow by using one of the SDP protocol identifiers "UDP/MBS-FEC/RTP/AVP", "UDP/MBS-FEC/RTP/SAVP" depending on the upper layer protocol RTP/AVP or RTP/SAVP respectively. If MIKEY is FEC protected and encapsulated in source packets, then it is indicated in the security description using the *fecProtection* element and the destination IP address.

##### 7.2.1.2.5 Packet Format for Repair packets

The packet format for FEC repair packets as defined in RFC 6363 [x], section 5.4 shall be used for repair packets.

The UDP payload consists of the Repair FEC Payload ID, and zero, one or more repair symbols. The format of the Repair FEC payload ID is defined in clause [x], section 6.2.3.

The repair packet sent over IP and UDP is indicated in the SDP using the protocol identifier "UDP/MBS-REPAIR".

7.2.2.

##### 7.2.1.2.7 FEC block Construction algorithm and example (informative)

This section provides an example how to use the methods in RFC 6363 [x] and RFC 6681 [x] to generate a source block.

When the original UDP packet is placed into the source block, the value of the UDP flow identifier, F, followed by the value of the UDP payload length, L, are first written as a single byte and two-byte value in network byte order (i.e. with high order byte first) respectively into the first available bytes in the source block, followed by the UDP packet payload itself (i.e. not including the IP/UDP headers). Following this, if the next available byte is not the first byte of a new symbol, then padding bytes up to the next symbol boundary shall be included using the value 0 in each byte. As long as any source UDP packets remain to be placed, the procedure is repeated starting each UDP flow identifier at the start of the next encoding symbol.

An example of forming a source block is given in figure 14 below. In this example, three UDP packets of lengths 26, 52 and 103 bytes have been placed into a source block with symbol size T = 16 bytes. The first two packets are from UDP flow 0 and the third from UDP flow 1. Each entry in Figure 14 is a byte and the rows correspond to the source symbols and are numbered from 0 to 12. Bi,j denotes the (j+1)th byte of the (i+1)th UDP packet.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 26 | | *B*0,0 | *B*0,1 | *B*0,2 | *B*0,3 | *B*0,4 | *B*0,5 | *B*0,6 | *B*0,7 | *B*0,8 | *B*0,9 | *B*0,10 | *B*0,11 | *B*0,12 |
| *B*0,13 | *B*0,14 | *B*0,15 | *B*0,16 | *B*0,17 | *B*0,18 | *B*0,19 | *B*0,20 | *B*0,21 | *B*0,22 | *B*0,23 | *B*0,24 | *B*0,25 | 0 | 0 | 0 |
| 0 | 52 | | *B*1,0 | *B*1,1 | *B*1,2 | *B*1,3 | *B*1,4 | *B*1,5 | *B*1,6 | *B*1,7 | *B*1,8 | *B*1,9 | *B*1,10 | *B*1,11 | *B*1,12 |
| *B*1,13 | *B*1,14 | *B*1,15 | *B*1,16 | *B*1,17 | *B*1,18 | *B*1,19 | *B*1,20 | *B*1,21 | *B*1,22 | *B*1,23 | *B*1,24 | *B*1,25 | *B*1,26 | *B*1,27 | *B*1,28 |
| *B*1,29 | *B*1,30 | *B*1,31 | *B*1,32 | *B*1,33 | *B*1,34 | *B*1,35 | *B*1,36 | *B*1,37 | *B*1,38 | *B*1,39 | *B*1,40 | *B*1,41 | *B*1,42 | *B*1,43 | *B*1,44 |
| *B*1,45 | *B*1,46 | *B*1,47 | *B*1,48 | *B*1,49 | *B*1,50 | *B*1,51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 103 | | *B*2,0 | *B*2,1 | *B*2,2 | *B*2,3 | *B*2,4 | *B*2,5 | *B*2,6 | *B*2,7 | *B*2,8 | *B*2,9 | *B*2,10 | *B*2,11 | *B*2,12 |
| *B*2,13 | *B*2,14 | *B*2,15 | *B*2,16 | *B*2,17 | *B*2,18 | *B*2,19 | *B*2,20 | *B*2,21 | *B*2,22 | *B*2,23 | *B*2,24 | *B*2,25 | *B*2,26 | *B*2,27 | *B*2,28 |
| *B*2,29 | *B*2,30 | *B*2,31 | *B*2,32 | *B*2,33 | *B*2,34 | *B*2,35 | *B*2,36 | *B*2,37 | *B*2,38 | *B*2,39 | *B*2,40 | *B*2,41 | *B*2,42 | *B*2,43 | *B*2,44 |
| *B*2,45 | *B*2,46 | *B*2,47 | *B*2,48 | *B*2,49 | *B*2,50 | *B*2,51 | *B*2,52 | *B*2,53 | *B*2,54 | *B*2,55 | *B*2,56 | *B*2,57 | *B*2,58 | *B*2,59 | *B*2,60 |
| *B*2,61 | *B*2,62 | *B*2,63 | *B*2,64 | *B*2,65 | *B*2,66 | *B*2,67 | *B*2,68 | *B*2,69 | *B*2,70 | *B*2,71 | *B*2,72 | *B*2,73 | *B*2,74 | *B*2,75 | *B*2,76 |
| *B*2,77 | *B*2,78 | *B*2,79 | *B*2,80 | *B*2,81 | *B*2,82 | *B*2,83 | *B*2,84 | *B*2,85 | *B*2,86 | *B*2,87 | *B*2,88 | *B*2,89 | *B*2,90 | *B*2,91 | *B*2,92 |
| *B*2,93 | *B*2,94 | *B*2,95 | *B*2,96 | *B*2,97 | *B*2,98 | *B*2,99 | *B*2,100 | *B*2,101 | *B*2,102 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 14: Source block consisting of 3 source UDP packets of lengths 26, 52 and 103 bytes.

##### 7.2.1.2.9 Source FEC Payload ID

The Source FEC payload ID shall be the Source FEC Payload ID format A in section 6.2.2 of RFC 6681 [x].

##### 7.2.1.2.10 Repair FEC payload ID

The Repair FEC Payload ID shall be the Repair FEC Payload ID format A in section 6.2.3 of RFC 6681 [x].

##### 7.2.1.2.11 FEC Object Transmission information

The FEC Object Transmission information consists of:

- the maximum source block length, in symbols

- the symbol size, in bytes

The FEC Object Transmission information shall be the first four octets of the FEC Scheme Specific Information in section 6.2.1.2 of RFC 6681 [x].

NOTE: This corresponds to Payload ID Format A in RFC 6681 [106] as the last octet of FEC Scheme Specific Information is omitted.

The Source Block Length signalled within the Repair FEC Payload ID of any packet of a stream shall not exceed the Maximum Source Block Length signalled within the FEC Object Transmission Information for the stream.

The FEC Object Transmission Information shall be communicated as described in sub-clause 7.2.2.2.14. Note, the FEC Object Transmission Information is only communicated in SDP.

##### 7.2.1.2.12 Hypothetical FEC Decoder

This clause specifies the hypothetical FEC decoder and its use to check packet stream and MBS receiver conformance.

The hypothetical FEC decoder uses the packet stream, the transmission time of each packet, the initial buffering delay, and the SDP for the stream as inputs. The packet stream from the beginning of the FEC source block until the end of the stream shall comply with the hypothetical reference decoder as specified below when the initial buffer delay equals to the value of the min-buffer-time parameter.

The maximum hypothetical FEC decoding buffer size for MBS streaming is 1 Mbytes. The default hypothetical FEC decoding buffer size is equal to 1 Mbytes.

For the packet stream, the buffer occupancy level of the hypothetical FEC decoding buffer shall not exceed the value of the buf-size parameter, when it is present in the SDP, or the default FEC decoding buffer size, when the buf-size parameter is not present in the SDP. The output of the hypothetical FEC decoder shall comply with the RTP payload and decoding specifications of the media format.

The hypothetical FEC decoder operates as follows:

1) The hypothetical FEC decoding buffer is initially empty.

2) Each FEC source packet and FEC repair packet, starting from the first packet in transmission order, is inserted into a FEC source block at its transmission time. The FEC source block generation is done as specified in [106], section 6.2.3. The FEC source block resides in the hypothetical FEC decoding buffer.

3) When both the last FEC source packet and the last FEC repair packet of an FEC source block are transmitted, any elements of the FEC source block that are not original UDP packets (e.g. FEC repair packets and potential padding bytes) are removed from the hypothetical FEC decoding buffer.

4) Original UDP packets are not removed from the hypothetical FEC decoding buffer before the signalled initial buffering delay has expired. Then, the first original UDP packet in sequence number order is output and removed from the hypothetical FEC decoding buffer immediately. Each succeeding original UDP packet is output and removed when the following conditions are true:

i. The following time (in seconds) since the removal of the previous packet has elapsed:

8 × (size of the previous original UDP packet *including* UDP/IP header in bytes) / (1 000 × (value of "b=AS" SDP attribute for the stream))

ii. All the packets in the same FEC source block as the original UDP packet have been transmitted.

An MBS Client shall be capable of receiving a packet stream that complies with the hypothetical FEC decoder. Furthermore, in the case of RTP packets, when an MBS Client complies with the requirements for the media decoding of the packet stream, it shall be able to de-packetize and decode the packet stream and output decoded data at the correct rate specified by the RTP timestamps of the received packet stream.

##### 7.2.1.2.13 Signalling

The signalling for streaming FEC consists of several components:

- If several user services are bundled together they are indicated as a sequence of services in the User Service Bundle Description. See sub-clause 11.2.

- A separate SDP describing the FEC repair stream and all the flow IDs referenced from the User Service Bundle Description. See sub-clauses 11.2 and 7.2.2.2.14.

- SDP protocol identifiers and attributes to indicate the usage of the source packet format, how the FEC payload ID is configured and other FEC parameters such as minimal buffering delay, for the RTP/RTCP streams. See sub-clause 7.2.2.2.13a.

- Security description extensions to indicate usage of FEC source packet format, and the FEC parameters. See sub-clauses 11.3 and 7.2.2.2.13a.

The user service description contains either a single service or several bundled services. All of the streaming delivery methods and security descriptions that are present within the *bundleDescription* element must be considered when configuring the FEC operations. This includes RTP, RTCP and MIKEY flows. A receiver intending to perform FEC decoding to cover for packet losses shall receive all the flows that are indicated to be sent as FEC source packets, even if the flows are in a service currently not played out. A receiver intending to use FEC shall also receive the FEC repair stream as described by the FEC Repair Stream Description. The delivery method’s session description, and the security description both carry the FEC source packet configuration information: FEC encoding ID, FEC instance ID, and FEC OTI information. The FEC repair packet stream is configured using the similar methods as for the source packets, with the addition of the Flow ID information and buffer delay parameter.

##### 7.2.1.2.13a SDP for FEC source packet streams

To indicate the presence of the FEC layer between IP/UDP and, RTP or SRTP a SDP protocol identifier is used. Instead of the normal RTP/AVP and RTP/SAVP protocol identifiers, ‘UDP/MBS-FEC/RTP/AVP’ and ‘UDP/MBS-FEC/RTP/SAVP’ are defined respectively. Both these protocol identifiers shall use the FMT space rules that are used for RTP/AVP and RTP/SAVP respectively, i.e. payload types used in the RTP session is listed. The protocol identifiers are defined in Appendix C1.

The FEC parameters, FEC encoding ID, FEC instance ID and FEC-OTI-Extension information are signalled using the mechanism defined in sub-clause 8.3.1.8. The "a=FEC" SDP attribute shall be used to indicate the single definition that is used for each media component.

For MIKEY messages the Security Description is used to indicate when FEC source packet shall be used, see sub-clause 11.3. The FEC parameter used is also defined in the Security Description. As all MIKEY packets from all user services arrive on the same port, the receiver must use the destination address to separate FEC protected packets from not FEC protected packets. This requires that all MIKEY packets sent to a specific destination address are either FEC protected or not. Note that it is not possible to mix protected and non-protected packets within a single stream as there is no mechanism to determine whether they are protected or not.

##### 7.2.1.2.14 SDP for FEC repair packet streams

The repair packet stream is indicated in SDP using a media block with the protocol identifier "UDP/MBS-REPAIR". The media type shall be "application". The FEC parameters, FEC encoding ID, FEC instance ID, FEC-OTI-Extension information and repair parameters (min-buffer-time) are signalled using the mechanisms defined in sub-clause 8.3.1.9. Each media component shall reference only one FEC declaration.

The mapping of the FEC source block flow ID to the destination IP address and UDP port are done using the SDP attribute "a=mbs-flowid" defined in sub-clause 8.3.1.9.

Interleaving may be signaled using the "X-3gpp-FEC-Interleaving" attribute, which also gives the arrangement of the flows in the source block and by consequence their transmission order. The "X-3gpp-FEC-Interleaving" attribute is defined in sub-clause 8.3.1.11.

##### 7.2.1.2.15 SDP example for FEC

An example of how the SDP <http://www.example.com/3gpp/mbs/session1.sdp> could look for a session containing two media streams that are FEC protected. In this example we have assumed an audiovisual stream, using 56 kbps for video and 12 kbps for audio. In addition another 300 bits/second of RTCP packets from the source is used for the each of the sessions. Hence, the total media session bandwidth is 56+12+0.3+0.3 = 68.6 kbps.

v=0  
o=ghost 2890844526 2890842807 IN IP6 2001:210:1:2:240:96FF:FE25:8EC9  
s=3GPP MBS Packet Distribution SDP Example  
i=Example of RTP streaming of MBS Packet Distribution SDP file  
u=http://www.infoserver.example.com/ae600  
e=ghost@mailserver.example.com  
c=IN IP6 FF1E:03AD::7F2E:172A:1E24  
t=3034423619 3042462419

a=mbs-mode:broadcast 123869108302929

b=AS:62

b=TIAS: 60500

a=maxprate: 25

a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9

a=FEC-declaration:0 encoding-id=1

m=video 4002 UDP/MBS-FEC/RTP/AVP 96

b=TIAS:55000

b=RR:0

b=RS:300

a=rtpmap:96 H263-2000/90000  
a=fmtp:96 profile=3;level=10  
a=framesize:96 176-144

a=FEC:0

a=maxprate:15

m=audio 4004 UDP/MBS-FEC/RTP/AVP 98

b=TIAS: 11500

b=RR:0

b=RS:300

a=rtpmap:98 AMR/8000

a=fmtp:98 octet-align=1

a=FEC:0

a=maxprate:10

The SDP file from above is modified to use two different FEC flows.

v=0  
o=ghost 2890844526 2890842807 IN IP6 2001:210:1:2:240:96FF:FE25:8EC9  
s=3GPP RTP streaming of MBS Packet Distribution SDP Example  
i=Example of RTP streaming of MBS Packet Distribution SDP file  
u=http://www.infoserver.example.com/ae600  
e=ghost@mailserver.example.com  
c=IN IP6 FF1E:03AD::7F2E:172A:1E24  
t=3034423619 3042462419

a=mbs-mode:broadcast 123869108302929

b=AS:62

b=TIAS: 60500

a=maxprate: 25

a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9

m=video 4002 UDP/MBS-FEC/RTP/AVP 96

b=TIAS:55000

b=RR:0

b=RS:300

a=FEC-declaration:0 encoding-id=1

a=rtpmap:96 H263-2000/90000  
a=fmtp:96 profile=3;level=10  
a=framesize:96 176-144

a=FEC:0

a=maxprate:15

m=audio 4004 UDP/MBS-FEC/RTP/AVP 98

b=TIAS: 11500

b=RR:0

b=RS:300

a=FEC-declaration:1 encoding-id=1

a=rtpmap:98 AMR/8000

a=fmtp:98 octet-align=1

a=FEC:1

a=maxprate:10

The SDP file for the two FEC streams

v=0  
o=ghost 2890844526 2890842807 IN IP6 2001:210:1:2:240:96FF:FE25:8EC9  
s=3GPP RTP streaming of MBS Packet Distribution FEC SDP Example  
i=Example of RTP streaming of MBS Packet Distribution SDP file  
u=http://www.infoserver.example.com/ae600  
e=ghost@mailserver.example.com  
t=3034423619 3042462419

a=mbs-mode:broadcast 123869108302929

b=AS:15

a=source-filter: incl IN IP6 \* 2001:210:1:2:240:96FF:FE25:8EC9

m=application 4006 UDP/MBS-REPAIR \*

c=IN IP6 FF1E:03AD::7F2E:172A:1E24

b=AS:15

a=FEC-declaration:0 encoding-id=1

a=FEC-OTI-extension:0 ACAEAA==

a=MBS-repair: 0 min-buffer-time=2600

a=FEC:0

a=MBS-flowid: 1=FF1E:03AD::7F2E:172A:1E24/4002, 2=FF1E:03AD::7F2E:172A:1E24/4003

m=application 4008 UDP/MBS-REPAIR \*

c=IN IP6 FF1E:03AD::7F2E:172A:1E24

b=AS:15

a=FEC-declaration:1 encoding-id=1

a=FEC-OTI-extension:1 ACAEAA==

a=MBS-repair: 1 min-buffer-time=2600

a=FEC:1

a=MBS-flowid: 3=FF1E:03AD::7F2E:172A:1E24/4004, 4=FF1E:03AD::7F2E:172A:1E24/4005

#### 7.2.1.3 General RTP Header Extension Mechanism

##### 7.2.1.3.1 Introduction

The General RTP Header Extension Mechanism [x] is a general mechanism to use the header extension feature of RTP (the Real-Time Transport Protocol). The General RTP Header Extension Mechanism should be supported.

##### 7.2.1.3.2 Timestamp Offset

Timestamp offsets for RTP may be transmitted using the general RTP header extension mechanism.

The variable timestamp extension element is 32 bits long. The first byte is the extension element header, i.e. the ID and len fields, as defined in [x]. The remaining 3 bytes are the timestamp-offset measured in the same frequency as the RTP timestamp.

Timestamp-offset: A 24 bit unsigned integer signalling the offset of the received packets of the same media in the tune-in FEC block. The timestamp offset indicates at most the difference between the RTP timestamp of the current packet and the highest RTP timestamp of packets of the same media stream that are transmitted in the current FEC source block.

Timestamp offset shall not be used if FEC protection and Interleaving are not being used.

The following example is a general RTP header extension block containing a single variable timestamp extension element.

0 1 2 3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| 0xBEDE | length=1 |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

| ID | len=2 | timestamp-offset |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

The presence of variable timestamps is signaled in the SDP file using the header extension specification and the URI "http://www.3gpp.org/2008/TimestampOffset". The URI signals the possible presence of timestamp offsets with the given ID.

### [7.2.2 Group Communication streaming

The MBS group communication streaming delivers a UDP/IP packet flow to the UE. AF/AS could deliver the UDP/IP packets to the MB-UPF directly if there is no FEC requirements. If the FEC requirement is mandatory, it could reuse the transparent streaming proxy mode in subclause 7.2.3.]

### 7.2.3 Transparent streaming

#### 7.2.3.1 Introduction

The transparent streaming shall be used by the MBSTF to transmit downstream service content received over Nmb8 from the AF/AS when the *Session Type* property is Transport-Mode. The transparent streaming delivers application data units as part of UDP or IP flows over an MBS session to the UE. This transparent streaming is particularly useful for multicast and broadcast of IP-based services for which the media codecs and application protocols are defined outside of this specification.

The MBSTF receives Application Data Units (ADUs) from the AF/AS, typically provided as UDP/IP packets and forwards them to the destination multicast IP address and port number.

Transparent streaming may be used within MBS User Services, where the session description is delivered as a unit of a User Service Description, or they may be used independently, where the content provider will announce the session via external means.

An MBS transparent streaming may be operated in a forward-only or in a proxy mode. In the forward-only mode, the transport protocol on top of IP is opaque to the MBSF/MBSTF and the session announcement may be handled by the content provider itself. In the proxy mode, the UDP packet payload of the UDP streams is opaque to the MBS session and an MBS Client is expected to make the UDP Payloads available to an application, without further knowledge on the content.

In the proxy mode is used, the transport protocol and session description are described in clauses 7.2.3.2 and 7.2.3.3.

The content provider may also request the application of FEC. [FFS]

#### 7.2.3.2 UDP payload format

When the proxy mode is used, the transport protocol shall be UDP/IP.

The application layer protocol on top of UDP/IP is out of scope of this specification. However, examples for application layer protocols are RTP, packetized MPEG-2 TS or other UDP-based streams. Generally, a sequence of Application Data Units (ADUs), i.e. unit of source data provided as payload to the transport layer, can be delivered by this transport protocol. As an example, this framework can be applied to RTP flows as well.

ADUs may be encapsulated into frames by a transport framing protocol prior to transmission using the UDP protocol, in order to provide additional transport functionality.

A UE that does not understand the transport framing protocol shall discard the transport framing header or trailer, and recalculate the UDP checksum prior to forwarding ADU to the receiver. The usage of the transport framing protocol is signalled by the presence of the "mbs-framing" attribute in a media session of the SDP. If used, all datagrams of the UDP flow shall be framed using the same transport framing protocol.

If transport framing is used, the BM-SC shall encapsulate exactly one ADU in an IP/UDP multicast packet where the ADU carried as a UDP payload is appended or prepended by the transport framing trailer/header as shown in Figure 7.2.3.2-1.

Note: The content provider shall be aware of the path MTU and shall not generate ADUs that do not fit into a single IP/UDP datagram.

0 1 2 3   
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| |  
| IP Header |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| |  
| UDP Header |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| … |  
| Transport Framing Protocol Header |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| |  
| Transport Payload |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  
| |  
| Transport Framing Protocol Trailer |  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 7.2.3.2-1 Transport Framing Protocol Header/Trailer

The transport framing protocol trailer/header shall be of constant length for all packets of the same UDP flow. The length of the transport framing protocol trailer/header shall be signalled to the UE as part of the "mbs-framing" attribute of the session description of the transparent session.

Note: The current specification does not define a transport framing protocol. Instead, it defines the framework for this protocol to ensure backwards compatibility by future versions of the protocol.

If the transport framing protocol is used and the receiver does not recognize the version of the transport protocol, it shall discard the trailer/header. The sender shall ensure that simple discarding by a receiver that does not support the indicated version of the transport framing protocol does not impact the integrity and consistency of the payload.

The UDP checksum shall be recalculated after any transport framing is performed and also after that transport framing is terminated.

#### 7.2.3.3 SDP Parameters

The Session Description of an MBS transparent streaming includes the following parameters:

- The sender IP address

- Session timing information

- MBS service type of MBS session

- The bitrate of the session

- may include the alternative TMGI

- For each UDP flow:

- The destination IP address and port number for each media line

- An indication of the usage of a transport framing protocol or not

- The protocol ID for each media session

- Any other parameters of the transported flow for each media

\*\*\*\* Last Change \*\*\*\*