Doc. TSG-676
14 December 2005

ATSC Technology Group Report: Caption Carriage in E-VSB and with New Codecs

Advanced Television Systems Committee
1750 K Street, N.W.
Suite 1200
Washington, D.C. 20006
www.atsc.org

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards for digital television. The ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries. Specifically, ATSC is working to coordinate television standards among different communications media focusing on digital television, interactive systems, and broadband multimedia communications. ATSC is also developing digital television implementation strategies and presenting educational seminars on the ATSC standards.

ATSC was formed in 1982 by the member organizations of the Joint Committee on InterSociety Coordination (JCIC): the Electronic Industries Association (EIA), the Institute of Electrical and Electronic Engineers (IEEE), the National Association of Broadcasters (NAB), the National Cable Television Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). Currently, there are approximately 140 members representing the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries.

ATSC Digital TV Standards include digital high definition television (HDTV), standard definition television (SDTV), data broadcasting, multichannel surround-sound audio, and satellite direct-to-home broadcasting.

Table of Contents

Technology Group Report: Caption Carriage in E-VSB and with New Codecs

1. INTRODUCTION

S13 was asked by TSG to address the carriage of closed captions in E-VSB and with the new codecs. The group completed a requirements study in January, which is included as Attachment A. Of note in the requirements is the decision to declare the cc_data() structure defined in A/53C (or subsequent revision) to be "captions." So, the technical task of this SG is to define the means to carry this structure for the various scenarios.

An analysis of the possible technical solutions was developed and documented, and is included as Attachment B[1] In addition to the techniques described in this analysis, an additional encapsulation was later proposed making use of "private data" in the transport layer adaptation field. The details of this proposal are included in Attachment C.

At its meeting on 29 March 2005, the group arrived at some interim decisions that were documented and communicated to TSG, S6, and S8 on 27 May 2005. This interim report was published internally as S13-295R1. This final report is consistent with, but expands on, those interim findings. The Specialist Group on Video and Audio has undertaken work to address related issues and published internally as S6-349r7.

The attachments were works in process of specialist group S13, and may be incomplete or inconsistent with the final recommendations in this report. They are provided as background material only.

Then recommendations in this report are grouped into several technical (not application) scenarios

- "Normal" video programming
- "Non-normal" video programming
- Audio-only programming
- Caption-only programming

2. CAPTIONS IN "NORMAL" VIDEO PROGRAMMING

"Normal video" is programming that has a video frame encoded for each display frame.

When normal video programming is present, captions should be carried in the video elementary stream analogous to the technique currently defined in A/53 for MPEG-2 – user data with each frame of video. While this technique has some disadvantages, the group felt that the similarity to existing architectural design for encoders and decoders outweighed these disadvantages. Encoder and decoder product research and development was a driving consideration.

3. CAPTIONS IN "NON-NORMAL" VIDEO PROGRAMMING

When video is present, but it lacks the 1:1 mapping of encoded frames to display frames, this is "non-normal" video. The main examples of this are MPEG-2 low_delay mode and still frame mode.

---

[1] This document is incomplete and dated with respect to decisions in this report, but still provides excellent background material on the technical issues.

Captions, to be fully functional, require the ability to output a caption packet with every frame. This is needed to define the 608 compatibility output for each frame and to preserve the 708 bandwidth and (implied) buffering model.

For the still frame mode, no recommendation was discovered that preserves all the properties and bandwidth of captions today. That is not to say that captions could not still be output on the infrequent video I frames, but that may have limited application, especially if the time between I frames is long. It would preclude any practical use of 608 compatibility, but it may be adequate for infrequently changing data such as song titles.

For the low_delay mode where the full functionality of captioning is desired, it is possible to pre-pad zero-delta pictures before any large picture that causes the decoder buffer underflow. This results in restoring the 1:1 mapping of encoded frames to display frames, and thus the video elementary stream is no longer low_delay mode. If this technique is not acceptable, then the limitations of still frame mode described above apply.

4. CAPTIONS IN AUDIO ONLY PROGRAMMING

For existing systems, as well as E-VSB applications, when there is no video elementary stream, captions can be carried in a "minimal" video encoding conforming to A/53, Annex A. This preserves the video format signaling and is friendly to existing caption encoder and decoder architectures, yet still provides a low bitrate stream desirable in E-VSB applications and other low-bandwidth applications.

The essence of the proposal, using MPEG-2 video as the example, is to send a legal MPEG-2 video elementary stream, but encode it to be a series of black frames at some minimal permitted video format. While today, this is constrained by A/53 to SD@24 fps, any ATSC-supported format can be used. For example if CIF@24 fps were to be added to A/53, that would provide an additional bandwidth savings. Such a minimally encoded PES packet may fit into a single transport packet.

More details on this technique can be found in Attachment D.

The application of this technique can be equally accomplished with other video codecs besides MPEG-2.

5. CAPTION ONLY PROGRAMMING

After consideration of scenarios where there is no video or audio ("caption-only"), the group felt that there was not sufficient business model support or well-defined requirements at this time to justify further investigation. Furthermore, when the requirements of such a service are better understood, it may be prudent to look beyond 708 to streaming text services such as 3GPP (as enabled in MPEG-4), and other similar designs.

No recommendation for caption only programming is made at this time.

Attachment A: Requirements Study (S13-270r3)

1. INTRODUCTION AND SCOPE

This document specifies use-case scenarios, requirements, and non-requirements for the purpose of developing an encapsulation of closed captioning in the ATSC robust mode transport (E-VSB) and when using the new codecs.

The term, *captions*, for the purpose of this document, is defined to be the cc_data() structure defined in A/53C, Annex A, Table A8.

2. USE-CASE SCENARIOS

Use-case scenarios are intended to provide representative, informative examples of situations where the extended design for the carriage of captioning can be used when MPEG-2 video is not present (and thus no way to carry captions as currently defined in A/53) or is in the E-VSB transport.

2.1 Support for E-VSB

When existing codecs (MPEG-2 video and AC-3 audio) are in use in E-VSB, then it needs to be possible for the transport to carry captions.

2.2 Support for New Codecs

When a new video elementary stream is present other than MPEG-2, then it needs to be possible for the transport to carry captions.

2.3 Support for No Video Programming

When no video programming is present and audio is present (e.g., *E-VSB Fallback Audio Scenario*), then it needs to be possible for the transport to carry captions.

3. REQUIREMENTS

This section describes the primary requirements to be fulfilled by the carriage of captions in the robust mode and with new codecs.

3.1 General Requirements

3.1.1 A/53 and A/65 Compatibility

The carriage of captions shall be compatible with the general announcement and signaling mechanisms defined by A/53 and A/65 and their work in process amendments. That is, nothing in the design shall be inconsistent with the core ATSC transport specifications.

3.1.2 SCTE Re-Use

Consideration shall be given to a design that enables the compatible carriage of captions in cable transports defined by SCTE.

3.1.3 International Re-use

Consideration shall be given to a design that enables the compatible carriage of captions in ATSC transports outside the US.

3.1.4 E-VSB Support

The design shall work in both the normal transport as well as the E-VSB portion of the transport, or both.

3.2 Operational Requirements

3.2.1 Buffer Model

The design shall include a buffer model.

3.2.2 Frame Rates

The design shall support all frame rates and video encodings in use in the ATSC transport, including low_delay mode (or equivalent in non-MPEG-2 streams) and still frames (or equivalent in non-MPEG-2 streams).

3.3 Functional Requirements

3.3.1 Signaling and Announcement Compatibility

The design should be compatible with the signaling and announcement defined in A/65, specifically the caption service descriptor. A design that requires new signaling and announcement should be avoided.

3.3.2 Payload Compatibility

The payload design shall be compatible with the cc_data() structure defined in A/53C, Annex A, Table A8.

3.3.3 Wrapper Compatibility

If encoded in the video elementary stream, any wrapper around the cc_data() structure should be compatible with the picture user data structure defined in A/53C, Annex A, Table A7, in order to facilitate encoding video metadata (e.g., Bar Data).

3.3.4 Frame Level Synchronization

The design shall enable presentation synchronization to specific frames of video at all supported video frame rates.

3.3.5 Elementary Stream Layer Abstraction

The captions shall be independent of the type of elementary stream in which it is carried.

3.3.6 Elementary Stream Independence

Consideration should be given to a design that could work without any other specific elementary stream present.

3.3.7 Backwards Compatibility

Consideration shall be given to minimizing the design changes going from existing equipment to equipment supporting the new encapsulation.

3.3.8 Extensibility

Consideration should be given to a design that easily supports future extensions to the ATSC transport design, new codecs, and new services with a mix of elementary streams.

3.4 Non-Requirements

3.4.1 Backwards Incompatibility

Although every effort should be made to reduce the design changes of existing equipment, the design need not be compatible with the MPEG-2 video solution currently in use.

Attachment B: Encapsulation Analysis (S13-272R2)

1. INTRODUCTION AND SCOPE

This document provides an analysis of the transport encapsulations for the purpose of leading to the design of one or more specific encapsulations for the carriage of closed captioning in the ATSC robust mode transport (E-VSB) and when using the new codecs.

The term *captions*, for the purpose of this document, is defined to be the cc_data() structure defined in A/53C, Annex A, Table A8.

The requirements for this work are defined in the external document, Requirements for Caption Carriage for Robust Mode and with New Codecs (T3-S13-270). Please refer to that document for background.

2. TRANSPORT ENCAPSULATIONS

2.1.1 Introduction

There are several transport encapsulations that could be employed to carry captions. These are:

1) MPEG-2 Sections
   a) Embedded in ATSC PSIP
   b) DSM-CC Synchronized Section
   c) New Private Section
2) "Data Piping"
3) Packetized Elementary Streams
   a) Embedded in other streams
   b) Separate ES

The existing caption carriage as defined in A/53-C, Annex A, uses encapsulation 1.a.i in the MPEG-2 Video ES, carried in the user data elementary stream startcode.

Follows is a discussion of the benefits and limitations of each of the encapsulations.

2.2 MPEG-2 Sections

2.2.1 Embedded in ATSC PSIP

Sections defined in A/65 have no inherent mechanism for synchronization with video and audio—that is, there is no PTS/DTS. Thus, they would fail to comply with any "normal" means for MPEG-2 transport synchronization to other elementary streams such as video or audio.

2.2.2 DSM-CC Synchronized Section

Synchronized DSM-CC sections are found in A/90, but there are several drawbacks:

1) There is modest overhead
2) There is no known implementation
3) Existing multiplexors, and other equipment that have to re-time-stamp streams, are not aware of them (re-time-stamping would be problematic)

2.2.3 New Private Section

Defining a new private section that was synchronized could probably reduce some of the overhead relative to DSM-CC, but would not overcome the other drawbacks listed above. These drawbacks are serious enough to not warrant further investigation.

2.3 Data piping

Data piping is a term used to describe the use of raw transport packets. This certainly has the least overhead, but has the following drawbacks:

1) A synchronization mechanism equivalent to PES and adaptation headers would need to be designed
2) There is, of course, no known implementation
3) Existing multiplexors and other equipment that have to re-time-stamp streams, are not aware of them (re-time-stamping would be problematic)

These drawbacks are serious enough to not warrant further investigation.

2.4 PES

2.4.1 Introduction

The most attractive encapsulation (relative to the alternatives) is PES. It is low overhead, has a built-in standard synchronization mechanism, is widely implemented, and is easily understood generically by multiplexor equipment.

There are several possibilities which will be considered in this section:

1) Embedded in other streams
   a) Audio ES
   b) MPEG-2 Video ES
2) Separate ES
   a) MPEG-2 Video black frames
   b) MPEG-2 Video no frames
   c) ETSI EN 301 775 and 300 472 (VBI)
   d) ETSI EN 300 743 (DVB Subtitling)
   e) A/90 and ETSI EN 301 192 (Data PES)

2.4.2 Embedded in Other Streams

2.4.2.1 Audio ES

One option is to embed the captions in the audio ES. This has the potential logical benefit of tying the captions to the audio to which they are usually a transcription. However, caption streams have been used for other purposes such as Descriptive Video Service (for the visually impaired). So, sometimes, captions are not related to the audio.

Embedding in the audio ES has most of the same pros and cons as embedding in the video ES (see the next section), plus the following additional drawbacks:

1) There is no legacy of using the audio ES for captions
2) Facilities today do not necessarily route captions along with audio
3) Audio encoders today do not reserve space for, or encode captions

These drawbacks are serious enough to not warrant further investigation.

2.4.2.2 MPEG-2 Video ES

Today, captions are carried in the MPEG-2 Video ES user data. There is no requirement or proposal to change this. However, with the introduction of new codecs, the question is how advisable is it to continue this practice with each new codec that is added to the ATSC transport.

The main advantage to continuing to include captions in the video elementary stream approach is that it is “safe”. That is, no matter what other pros and cons there are, it is well understood and reasonable to assume the behavior of the systems involved would not be altered

in any radically manner. The data paths remain the same and the role of the video encoder and caption server/encoder remains the same.

It also has the advantage of keeping the PID usage minimized, since the caption data is piggy-backed on the video PID.

It is potentially more efficient, but only if the video encoder is doing the caption insertion and does not pad out the caption structures in every frame.

2.4.3 Separate ES

2.4.3.1 PES Header Options

When carrying other than ISO defined payloads in PES, one must use a stream_id of private_stream_1 or private_stream_2. These are, among other things different by whether the stream is synchronized or not. That is, one has PTS/DTS and one doesn't. The details of exactly which PES header fields are included are which are not can be found in 13818-1. While the shorter header can save a few bits, it is also the case that the PTS and DTS can be omitted from the synchronized form, thus, the savings are minor if asynchronous operation is desired. Should a separate PES stream be defined for captions, then the PES stream_id should be private_stream_1.

2.4.3.2 MPEG-2 Video ES – Black Frames[2]

One of the most important is that we find a solution that minimizes new work on the part of either the encoder or decoder manufacturers. Furthermore, we shouldn't have to revisit this discussion every time someone wants to include a caption stream along with some new types of program elements. Also, one of the aspects I hadn't considered carefully involves the presentation timing, and the fact that in the current transport caption data is associated with particular video frames by virtue of being included in the video syntax at the picture level. We can look at the way current decoders decode video pictures and captions in consort.

So here's an idea. Let's specify that a standalone caption stream comes in the picture user data of an MPEG-2 video stream, just as today. But in this application, the MPEG-2 video is compressed black frames where all predicted frames use "skipped macroblocks" thus reducing the bit rate to a very low number. We can even use QCIF or CIF format (lower than the lowest Table A format) to further reduce the bits as the receiver does not have to display any upsampled video. It doesn't take many bits to encode black I-frames and very few bits to encode P or B frames with skipped macroblocks.

The advantages include:

- No change to existing caption decoder systems (hardware/software and VBI/overlay insertion into video)
- Caption overhead rate is still very low (small percentage of 9600 bps rate in A/53)
- For audio-only fallback with captions, an automated process in the encoder can create, in parallel with the real video, a black-video encoding retaining the captions.
- This caption encoding is usable for any carriage of captions where a real MPEG-2 video ES is not present in the program (if you have MPEG-2 video, you carry them the regular way). So we don't ever have to visit this again.
- The caption ES can carry PTS and use DTS in the regular way.
- Decoders maintain the same frame synchronization and output timing whether looking at the fallback caption stream or the regular one, because PTS/DTS is exactly aligned between them.

[2] This section is verbatim from Mark Eyer's email to S13-1 of 15 February 2005.

- Caption PTS is tied to the PCR of the program for synchronization with any other element in the program such as audio-only or stills.
- Caption is synchronized with frame level accuracy to another video ES or audio-frame.
- Acquisition and display of captions is tied to audio or video acquisition without any extra latency (compared to sending them in a PES packet where captions for several pictures have to be aggregated to reduce the transport overhead).
- Re-use of existing MPEG-2 video STD management tools in the encoder and receiver systems as opposed to definition of a new buffer management model for PES based carriage.

The disadvantages include[3]:

- MUX and receiver equipment don't usually work below 500 kbps. There are various failure modes, but certainly far from it working today. Without some better understanding of why—and even if—encoders and decoders can be made to work, then MUX's still may not. Low bitrate video is a known system problem—legal, and fixable, but doesn't work today. An upgrade to the decoders and possibly facility equipment would still be needed (as it will for any of the approaches being considered).
- This technique can't be used verbatim when video is present. The decoder would not know which video to decode and display if two VES were present in the PMT. So, a separate stream_id and stream_type is needed.
- If video is also present, the decoder would require two video decoder pipelines.
- Decoders today aren't very happy with video slaved to another PID containing the PCR (legal, just doesn't work today; try putting the PCR in the audio and pointing the video to it).
- In transport streams with lots of programs (such as those contemplated in cable distribution systems) bitrate is a factor, even if it is not in an ATSC transport today. This approach, no matter how small the black frame encoding is, it will result in significantly greater bitrate use than any other proposed solution. And, we have already heard a debate about the others being "too much." The threshold is subjective right now, but clearly this approach would be the worst use of bitrate.
- There is no benefit with regard to the PTS/DTS usage. A separate PES stream would have the same properties.
- In practice, actual caption data is "bunched up" and not evenly distributed over the video frames. So, I do not believe the aggregation latency argument above is relevant. And, certainly relative to the overhead of the black frame video encoding, any wasted packet bytes is overshadowed. That is, whatever extra overhead there would be in using a partial packet for just cc_data would be small relative to the black frame encoding.

2.4.3.3 MPEG-2 Video ES – No Pictures[4]

Along these same lines, have you considered a VES with no pictures at all—just user data? Some time ago, I studied 13818-2 and could not find any requirement that a legal bitstream had to actually contain pictures. This will suffer from the same (or probably more) practical issues I mention above, but if various equipment is going to fail to work right today anyway with your proposal for black pictures, we might as well reduce the bits to the minimum and leave out the pictures altogether.

---

[3] Summarized from Mike Dolan's email to S13-1 on 16 February 2005 and Mr. Eyer's reply.

[4] This section is verbatim from Mr. Dolan's email to S13-1 of 16 February 2005.

And, if just user data is legal, then we are a short step away from eliminating the user data start code, format_identifier and type_code, just putting cc_data() in the packet. But taking this step clearly makes it not be a VES any more, but a new PES stream.

2.4.3.4 ETSI EN 301 775[5] (VBI)

This ETSI specification from DVB defines a means to encode analog VBI data in MPEG-2. One form is the use of PES to allow synchronization with the video frames, and thus reconstruction of the analog signal, if desired.

This has the advantage of being already "baked". However, this encoding is not quite on the mark since it presumes the source of the data is analog VBI "characters". The data structures are designed with this in mind. It is technically possible to place the cc_data structure into this format, and obtain the necessary payload identifier from DVB. But it's encoding would have no meaning relative to analog VBI signals, and could be confusing.

ETSI 300 472 (ITU-R System B Teletext) is essentially the same as 301 775.

2.4.3.5 ETSI EN 300 743 [DVB Subtitling]

This is a PES carriage for subtitling based somewhat on ETSI 301 192. Like 775, it has a data_identifier field, but then it has a subtitling stream number field, then the subtitle payload.

This does not really fit well for cc_data since the caption services are multiplexed at a lower level in 708 so it would not be possible to make use of the stream number field. And the payload is fixed to be DVB subtitling segments so could not be extended to carry cc_data.

2.4.3.6 A/90 and ETSI 301 192

A/90, Section 9.2 is a slightly constrained version of the synchronized PES for data design of 301 192, Section 6. This is a general purpose data payload in PES and could be used with minor extra design.

There are no known implementations of A/90, Section 9.2. And, the DVB data PES designs (VBI, Teletext and Subtitling) do not follow 301 192 for some reason, even though it pre-dates them.

2.4.4 Summary of the Pros and Cons of Carrying Captions in a Separate PES[6]

2.4.4.1 Pros

ES can be parsed and processed by similar methods (hardware/firmware) to those we already use for captions inside MPEG-2 video. We scan for user data start codes and then parse the data structure to follow.

Captions can be more easily added to existing video, because they can be added without touching video syntax. The practice of inserting dummy cc_data() packets into video (to accommodate later caption insertion) need no longer be done.

This method works regardless of what other present or future video compression formats or ES components may be present in the program. It's a "once and for all" solution. We don't need to define a new and perhaps different method for carrying captions every time we include a different video codec in the ATSC system.

The complexity involved in having caption data inserted within video frames, themselves not always delivered in presentation order, is avoided. When sent in a separate stream, caption bytes can be sent in order of their output.

---

[5] From Pat Waddell's email to S13-1 of 18 February 2005 with added analysis.

[6] This section is verbatim from Mr. Eyer's email to S13-1 of 14 January 2005.

This method allows insertion equipment to avoid the latency involved in including them in the video compression data path. Caption streams can be inserted directly to the output mux. This is helpful in cases of real-time caption encoding.

In some cases, the program may include more than one video encoding (for example, a simulcast of multiple compression formats or different resolutions). With captions in a separate stream, they're included just once per program.

Latency can be traded off against transport overhead. The lowest latency will involve the highest overhead (see first bullet below), but if latency is not an issue, many bytes of caption data (corresponding to many output frames) can be collected into one packet.

2.4.4.2 Cons

Caption data may be very low rate stream (for example, 608 data is two bytes per field, or 960 bps). Some overhead will be needed to maintain a rate of two bytes per field, because the cc data occupies less than a full TS packet yet the packets need to be sent at regular intervals for latency. If caption bytes were sent every frame time, you would need to send one TS packet per frame (30 packets per second), which translates to using a total BW of 188*8*30 = 45 kbps.

Changes to the current decoder design will be needed anyway, because currently, captions for up to three fields are provided within one cc_data() structure, and the association to video is done by tying the caption data to a video picture that carried that user data. If captions are provided in a separate stream, association would be done via PTS.

3. DECODER CONSIDERATIONS

There are several aspects of the decoding and display process that need attention. When there is video present, the video format and frame rate are well defined. Video overlays take this video format information into consideration when displaying captions over the video. If there is no video, and thus no explicit specification of video format, then the decoder must choose some output format in which to display the captions.

ANNEX A. A/53 cc_data Structure (aka "Captions")

From A/53, Annex A, Table A8:

Table A8 Caption Data Syntax

Syntax	No. of Bits	Format
cc_data() {		
reserved	1	'1'
process_cc_data_flag	1	bslbf
additional_data_flag	1	bslbf
cc_count	5	uimsbf
reserved	8	'1111 1111'
for (i=0 ; i < cc_count ; i++) {		
marker_bits	5	'1111 1'
cc_valid	1	bslbf
cc_type	2	bslbf
cc_data_1	8	bslbf
cc_data_2	8	bslbf
}		
marker_bits	8	'1111 1111'
if (additional_data_flag) {		
while (nextbits() != '0000 0000 0000 0000 0000 0001') {		
additional_cc_data		
}		
}		
}		

ANNEX B. MPEG-2 Video ES Encapsulation

From A/53, Annex, Tables A6 and A7:

Table A6 Picture Extension and User Data Syntax

Value	No. of Bits	Mnemonic
extension_and_user_data(2) {		
while ((nextbits( ) == extension_start_code) || (nextbits() == user_data_start_code)) {		
if (nextbits()== extension_start_code)		
extension_data(2)		
if (nextbits() == user_data_start_code)		
user_data(2)		
}		
}		

Table A7 Picture User Data Syntax

Syntax	No. of Bits	Format
user_data( ) {		
user_data_start_code	32	bslbf
ATSC_identifier	32	bslbf
user_data_type_code	8	uimsbf
if (user_data_type_code == '0x03')		
cc_data()		
else if (user_data_type_code == '0x06')		
bar_data()		
else {		
while (nextbits() != '0000 0000 0000 0000 0000 0001') {		
ATSC_reserved_user_data	8	
}		
next_start_code()		
}		

user_data_start_code – This is set to 0x0000 01B2.

ATSC_identifier – This is a 32 bit code that indicates that the video user data conforms to this specification. The value ATSC_identifier shall be 0x4741 3934.

user_data_type_code – An 8-bit value that identifies the type of ATSC user data to follow. Value 0x03 indicates cc_data(), value 0x06 indicates bar_data(), and other values are either in use in other standards or are reserved for future use.

ANNEX C. H.264 Proposed Encapsulation

From T3S6-277R5-WD, Annex F, Table F5:

Table F1 Caption Data Syntax

Syntax	No. of Bits	Format
user_data_registered_itu_t_t35 ( ) {		
itu_t_t35_country_code	8	bslbf
itu_t_t35_provider_code	16	bslbf
ATSC_identifier	32	bslbf
user_data_type_code	8	uimsbf
if (user_data_type_code == '0x03')		
cc_data()[7]		
else {		
while **(**user_data_type_code **!=** '0x03') {		
ATSC_reserved_user_data	8	
}		
}		

[7] T3S6-277R5-WD defines cc_data differently than in A/53, Annex A.

itu_t_t35_country_code – A fixed 8-bit field registered by the ATSC. The value is [TBD] and shall be a country code as specified by ITU-T Recommendation T.35 Annex A.

itu_t_35_provider_code – A fixed 16-bit field registered by the ATSC. The value is [TBD].

ANNEX D. PES Data Encapsulation

From A/90, Section 9.2[8]:

Table 9.3 Syntax for PES Synchronized Data Packet Structure

Syntax	No. of Bits	Format
synchronized_data_packet () {		
data_identifier	8	uimsbf
sub_stream_id	8	uimsbf
PTS_extension_flag	1	bslbf
output_data_rate_flag	1	bslbf
reserved	2	'11'
synchronized_data_packet_header_length	4	uimsbf
if (PTS_extension_flag=='1') {		
reserved	7	'1111111'
PTS_extension	9	uimsbf
}		
for (i=0;i<N1;i++) {		
synchronized_data_private_data_byte	8	bslbf
}		
for (i=0;i<N2;i++) {		
synchronized_data_byte	8	bslbf
}		
}		

The semantics of the synchronized_data_packet are defined below:

data_identifier – This 8-bit field shall identify the type of data carried in the PES data packet. It shall only be set to 0x22[9].

sub_stream_id – This 8-bit field shall be user private.

PTS_extension_flag – This 1-bit field shall indicate the presence of a PTS extension field. A value of '1' indicates the presence of the PTS_extension field in the PES_data_packet. If the PTS_extension field is not present this flag shall be set to '0'.

output_data_rate_flag – This 1-bit field shall be set to '0'.

synchronized_data_packet_header_length – This 4-bit field shall specify the length of the optional fields in the packet header. This includes the fields that are included when PTS_extension_flag is equal to '1' and it also includes the synchronized_data_private_data_byte(s).

PTS_extension – This 9-bit field shall extend the PTS conveyed in the PES header of this PES packet. This field when present shall contain the 9-bit Program Clock Reference (PCR)

[8] Compatible with ETSI 301 192, Section 6.

[9] A/90 notes: This particular value is chosen to be consistent with Table 2 of ETSI 301 192. However, this code point may need assignment from DVB/ETSI.

extension as defined in [11]. This extends the time resolution of synchronized data PTSs from the MPEG-2 standard resolution of 11.1 microseconds (90 kHz) to 37 nanoseconds (27 MHz).

synchronized_data_private_data_byte — This 8-bit field shall represent a service specific data byte.

synchronized_data_byte — This 8-bit field shall represent a byte of the synchronized PES packet payload. If the protocol_encapsulation field of the Data Service Table (defined in Chapter 11 of this standard) signals synchronized datagrams, either IP datagrams without LLC/SNAP or multiprotocol datagrams with LLC/SNAP, the synchronized_data_byte field shall carry one and only one datagram. Thus, when LLC/SNAP is used, the 8-byte LLC/SNAP header defined in [9] and [10] shall appear in the first 8 synchronized_data_byte bytes of a PES packet and nowhere else.

Attachment C: Adaptation Field Design (S13-291R1)

1. PURPOSE

The purpose of this amendment is to: a) better enable the design of the carriage of cc_data() in other codecs; and b) enable the movement of cc_data() into the draft CEA 708-C as proposed by the joint ATSC/CEA/SMPTE caption scope effort.

2. CARRIAGE IN VIDEO USER DATA

For the case when video frames are being sent per Table 3 [A/53], carriage is as defined in A/53, Annex A.

3. CARRIAGE WHEN NO VIDEO OR SPECIALLY CONSTRAINED VIDEO

For the case when video frames are not being sent at one of the frame rates in Table 3 [A53]; a means to enable delivery of captioning is needed.

An optional method for carriage of cc_data() bytes shall be to place then in the adaptation field of MPEG-2 TS packets as constrained below. These packets may be identified with the same PID as the video with which they are associated for the cases still picture and low delay. They may be identified with the same PID as the packets that carry audio.

The general approach is to use accumulate cc_data() triplets and insert them in a packet when enough payload is present. The rate is to be controlled and constrained by a buffer model. The approach is independent from adaptation field control values used.

The private data flag in the TS Packet Header (Table 2-6 of MPEG2) shall be set to '1'.

The Transport Private data length field shall be set between 7 and 182 bytes.

There shall be an integer number of cc_data structures present. More than one cc_data construct may be present per adaptation field, but all contents of the data for the value of the cc_count loop shall be in one TS packet. In other words, the set of data in the cc_count loop shall not be split across packets. If an integer number of cc_data structures finish before the end of a TS Packet payload, and the adaptation field control value is set to '10' or '11,' the remainder of the packet shall have bytes with a value 0xFF.

3.1 Private Data Syntax

Table 1 describes the data syntax that shall be used for carriage of ATSC private data in the adaptation header for ATSC-defined program elements and for adaptation-headed only TS packets in ATSC Transport Streams. The [MPEG 2] semantic elements transport_private_data_length and private_data_byte are included to show placement of the data structure. Non-ATSC-standard use of the adaptation header shall be permitted with use of the MRD per A/53D Annex C, Section 5.2.4.

Table 1 Private CC Data Syntax

Syntax	No. of Bits	Format
private_data( ) {		
transport_private_data_length	8	bslbf
private_data_byte // cc_data()		bslbf
key_code	8	0x03
reserved	1	'1'
process_cc_data_flag	1	bslbf
zero_bit	1	'0'[10]
cc_count	5	uimsbf
reserved	8	'1111 1111'
for (i=0 ; i < cc_count ; i++) {		
marker_bits	5	'1111 1'
cc_valid	1	bslbf
cc_type	2	bslbf
cc_data_1	8	bslbf
cc_data_2	8	bslbf
}		

3.1.1 CC Data Semantics

key_code – This field shall be set to a value between 0x01 and 0xFE. When used for cc_data() it shall be set to 0x03. The other values are ATSC reserved.

cc_data() – A data structure defined in [CEA 708C].

ATSC_reserved_user_data – Reserved for use by ATSC or used by other standards.

process_cc_data_flag – This flag is set to indicate whether it is necessary to process the cc_data. If it is set to 1, the cc_data has to be parsed and its meaning has to be processed. When it is set to 0, the cc_data can be discarded.

cc_count: This 5-bit integer indicates the number of closed caption constructs following this field. It can have values 0 through 31. The value of cc_count shall be set according to the frame rate and coded picture structure (field or frame) such that a fixed bandwidth of 9600 bits per second is maintained for the closed caption payload data. Sixteen (16) bits of closed caption payload data are carried in each pair of the fields cc_data_1 and cc_data_2.

cc_valid – This flag is set to '1' to indicate that the two closed caption data bytes that follow are valid. If set to '0' the two data bytes are invalid, as defined in [708].

cc_type – Denotes the type of the two closed caption data bytes that follow, as defined in [708].

cc_data_1 – The first byte of a closed caption data pair as defined in [708].

cc_data_2 – The second byte of a closed caption data pair as defined in [708].

3.2 Buffer Model

We will need a "leak rate" and buffer model when captions are aggregated. That is, in addition to sprucing up the 708 buffer model to more formally characterize the "9600 baud over 1 second" model, when receiving a chunk of captions every 2 seconds, they need to be doled out to the

[10] For backwards compatibility, this bit must be zero, not one.

caption decoder at some rate that provides the same basic affect as when they were spread out in time on every frame. This will be key to making the overall bandwidth very low yet still make it work “right.”

3.3 Display Coordinate Signaling

When this carriage method is chosen there shall be a caption service descriptor which signals the caption information. The value of wide_aspect_ratio shall be set to reflect the coordinate aspect ration in the captions. The video output of the receiver otherwise doesn't matter and can be receiver-dependent.

Attachment D: MPEG Grey Pictures (S13-292R2)

1. MPEG-2 GRAY PICTURE SCHEME

The MPEG-2 'Gray' picture scheme is a proposal to support captioning for audio only, no audio and video, still picture and low-delay mode applications. For audio only applications, it is recommended that the MPEG-2 program for this service include an MPEG-2 video component in addition to the audio. For applications with no audio or video, it is recommended that the MPEG-2 program for this service include an MPEG-2 video component and any other program component.

MPEG-2 Gray video is coded as (IPPPPP…) or as (PPPPPPP…) without any B-pictures so that there is no re-ordering of captions between transmitted pictures and displayed pictures. In addition, each picture is coded as "sequence-header, sequence-extension, picture-header, picture-user-data with captions" so that random access is made possible at each picture.

For a picture size of 176 x 128 (listed in the latest draft of Annex F), each P-picture requires (12 bytes for sequence-header + 10 bytes for sequence-extension + 81 bytes for picture data + 12 bytes for caption data) = 115 bytes. 5 such pictures per PES packet fits into the payload of 3 transport packets (3 x 188 bytes) and one can send the compressed video data for 30 pictures/second using 18 transport packets or 27 Kbps. Video data for 24 pictures/second rate requires 22 Kbps.

MPEG-2 specifies use of a reference picture with Y= Cr = Cb = 128 when decoding starts at a P picture with no reference. For audio only and no audio and video applications, this proposal recommends the MPEG-2 video to be coded as (PPPPP….). This does not require an I picture and the background for captions will appear as Gray.

If a background other than 'Gray' is desired, then video should be coded as (IPPPPP…IPPPPPPP…IPPPPPP.) where the I-pictures carry the desired background and this will stay (as the P pictures repeat the I picture) till the next I picture with a different background is received. I pictures need to be repeated at regular intervals to assist in producing the desired background. Note that acquisition is made at each picture (as each picture is coded with a sequence-header) and if acquisition is made at a P picture the background will stay 'Gray' till an I picture is received. This coding scheme requires slightly larger than 22-27 Kpbs rate as the I pictures with the desired background require more than 115 bytes. However this provides the ability for broadcasters to customize a desired background. This scheme can also be used for still video where the I pictures represent the 'still video frames' with enough P pictures sent between the 2 still video I frames to keep the frame rate constant. Such P pictures can also be used to fill the missing pictures between coded pictures and frame rate in low-delay mode applications also.

This scheme enables re-use of most of end to end caption systems deployed currently in encoders (from feeding captions at an encode station) and decoders (reconstruction of captions into NTSC and DTV signals). This also gives the flexibility to transmit the caption data at all the frame rates allowed by the A/53 standard and in display order as no B pictures are used.

This proposal assumes QCIF video format in MPEG-2 video (176 x 128) with multiple pictures carried in a PES packet. Note that the current ATSC standard specifies a lowest resolution of 704 x 480 with one picture per PES packet.