SMPTE STANDARD

for Television — SDTI Content Package Format (SDTI-CP)

1 Scope

This standard specifies the format for the transport of content packages (CP) on the serial digital transport interface (SDTI). This format is abbreviated to the term SDTI-CP. The format is a packaging structure for the assembly of system, picture, audio, and auxiliary data items as follows:

– A system item assembled as a group of timing and control elements, plus any metadata associated with the picture, audio, and auxiliary data items;

– A picture item assembled as a group of up to 255 picture stream elements;

– An audio item assembled as a group of up to 255 audio stream elements;

– An auxiliary item assembled as a group of up to 255 auxiliary data elements such as ancillary data lines, teletext, and other data.

This standard defines the structure of the content package mapped onto the SDTI transport. All element and metadata formats are defined by SMPTE 331M.

An SDTI-CP compliant receiver shall be capable of receiving and parsing the structure of the SDTI-CP format.

An SDTI-CP compliant decoder is defined by the ability to both receive and decode a defined set of elements and metadata according to an associated decoder template document.

The baseline operation of this standard is defined by the transport of content packages locked to the SDTI transport frame rate. This standard additionally defines format extension capabilities as follows:

– Allow content package transfers at higher and lower than the specified rate through isochronous and asynchronous transfer modes;

– Provision of a timing mode to reduce delay and provision for two content packages in each SDTI transport frame;

– Carriage of content packages in a low-latency mode; and

– Multiplexing of content packages from different sources onto one SDTI transport.

This standard is limited to SDTI operating at a bit rate of 270 Mb/s and 360 Mb/s as defined by SMPTE 305M.

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.

ANSI/SMPTE 298M-1997, Television — Universal Labels for Unique Identification of Digital Data

SMPTE 12M-1999, Television, Audio and Film — Time and Control Code

Approved
January 31, 2000

SMPTE 305M-1998, Television — Serial Data Transport Interface

SMPTE 331M-2000, Television — Element and Metadata Definitions for the SDTI-CP

SMPTE RP 168-1993, Definition of Vertical Interval Switching Point for Synchronous Video Switching

SMPTE RP 204-2000, SDTI-CP MPEG Decoder Templates

3 General specification

Figure 1 shows the basic layered structure of a content package. It shall be constructed of up to four items where each item is constructed of one or more elements.

The system item carries content package metadata and may contain a control element. The system item also carries metadata which is related to elements in the other items.

The picture item can consist of up to 255 picture stream elements.

The audio item can consist of up to 255 audio stream elements.

The auxiliary item can consist of up to 255 auxiliary data elements.

A content package contains the associated contents of one content package frame period starting with a system item and optionally containing picture, audio, and auxiliary items.

Element and metadata formats are defined in SMPTE 331M. New element and metadata types may be added as new requirements are defined.

An SDTI-CP decoder shall be specified by its ability to receive and decode an element or set of elements together with associated metadata defined by an associated decoder template document. The MPEG decoder template is SMPTE RP 204. Other decoder template recommended practices may be defined as required for other applications of the SDTI-CP. An SDTI-CP encoder which creates a content package with greater capabilities than those defined by a decoder template may risk undocumented decoder effects.

The term baseline operation is used throughout this standard and has a specific meaning defined as the combination of synchronous transfer mode operating in normal timing mode. Transfer and timing modes are defined in clauses 8 and 9.

This standard first describes baseline operation. Later parts of this standard define the methods by which the SDTI-CP format may:

– allow content package transfers at higher and lower than the specified rate through isochronous and asynchronous transfer modes;

– provide a timing mode to reduce delay and provide for two content packages in each SDTI transport frame;

– carry content packages in a low-latency mode; and

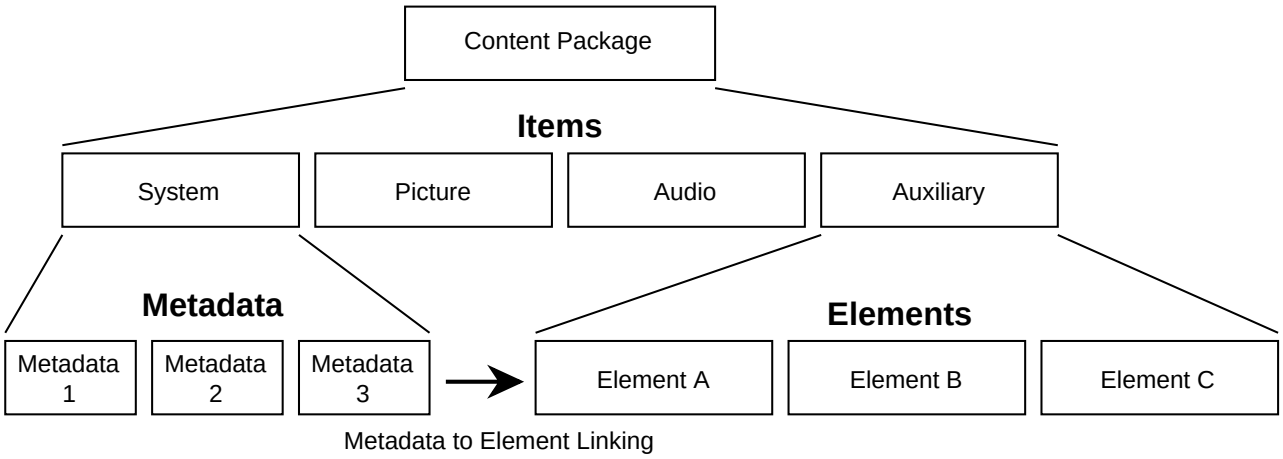

Figure 1 – Basic content package structure

– multiplex content packages from different sources onto one SDTI transport.

In general, special decoders are required to support these modes and the level of support provided shall be indicated in an associated decoder template document.

4 SDTI block structure

The system, picture, audio, and auxiliary items are each formatted as SDTI variable blocks beginning with the separator word and terminating with the end code word. Each variable block shall start immediately following the SAV sequence. Thus, the starting point of a content package can be found through the detection of a system item type code.

The format of each variable length block is shown in figure 2. The input format is 8-bit data entered into bits b0 to b7 of the 10-bit word.

Bits b8 and b9 are both set to 1 for the separator and end code words. For the item type, word count, and data block words, bit 8 is set to be the even parity of bits b0 to b7 and bit 9 is set to be the odd parity of bits b0 to b7. These specifications follow those of SMPTE 305M.

The SDTI data type word values are as follows:

– System item: 04_h
– Picture item: 05_h
– Audio item: 06_h
– Auxiliary item: 07_h

These data type values shall be registered in SMPTE 305M.

4.1 SDTI line and address numbers

Since the data in each SDTI variable block continue through as many lines as necessary until the block end, it is necessary that the SDTI header line numbers are contiguous. It is also necessary that the SDTI header source and destination address values are constant throughout the transmission of all lines associated with any one content package.

4.2 SDTI switching

The arbitrary switching of SDTI data streams, although at the picture frame boundary, may affect the ability to successfully decode picture, audio, and auxiliary data without the use of special processing equipment to mitigate the switching effects. The lines affected by a picture switch are defined in SMPTE RP 168. A continuity count is provided in the system item which can be used to indicate content packages affected by a switch.

5 Content package structure

Each content package shall consist of a minimum of the system item together with any, all, or none of the picture, audio, and auxiliary items.

The system item shall appear first in any sequence of items within a content package. The order of the picture, audio, and auxiliary items may be restricted in an associated decoder template document. There shall be only one item of any type in any one content package.

A system item shall be present in the content package. The presence of the other items in the content package is optional depending upon the transmission requirements.

In baseline operation, the content package start reference is defined by the switching point of SMPTE RP 168. The system data marks the first item of the content package and, for baseline operation, starts on the following reserved line for current television systems:

525/60: Line 13 625/50: Line 9

Figure 2 – Format of the SDTI variable block

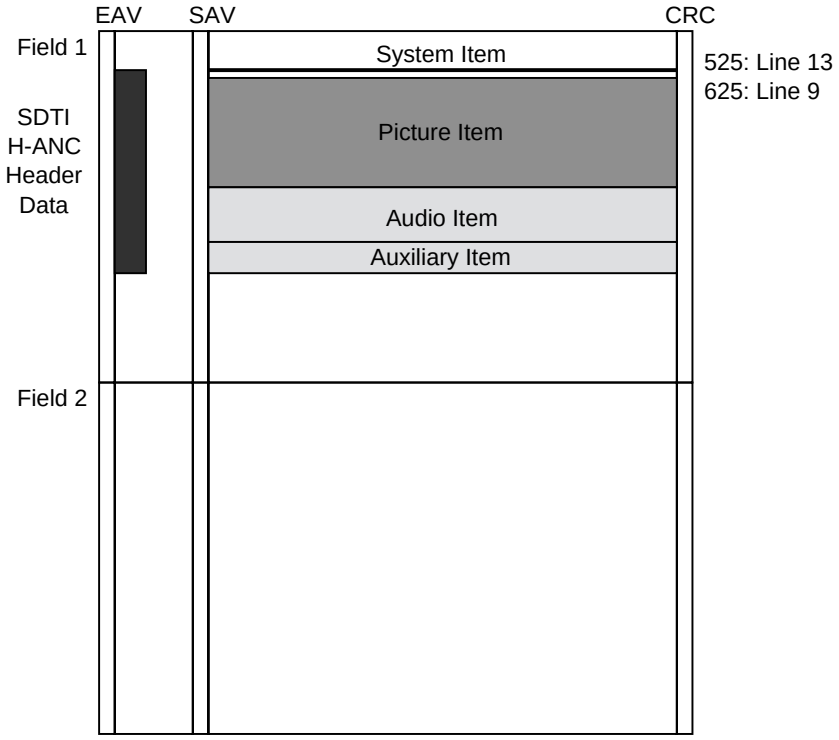

Figure 3 – Arrangement of system, picture, audio and auxiliary items in the content package

The general arrangement of system, picture, audio, and auxiliary data is given in figure 3.

The system item is provided to give content package metadata and control information together with metadata of the associated picture, audio, and auxiliary items.

A picture item is an assembly of up to 255 picture stream elements.

An audio stream item is an assembly of up to 255 audio stream elements.

An auxiliary data item is an assembly of up to 255 auxiliary data elements.

Assignment of an element to an item type is defined in SMPTE 331M. An element shall be placed in its defined item type.

The content package formed by the system, picture, audio, and auxiliary items shall represent the associated contents of one content package frame period defined by the content package rate variable in the system item.

The content package shall not exceed the frame period of the SDTI except where specifically provided by the special transfer modes described later in this standard.

The content package distribution for an example 12-frame MPEG-2 GOP is shown in figure 4.

In the case of special transfer modes, a content package may overlap the interface frame period. In some optional modes, there may be more than one content package per interface frame period. Details of these modes are available in later clauses of this standard.

5.1 Timing considerations

Each content package shall be regarded as a potentially editable unit, so time alignment of the items and elements within a content package is recommended. There may be occasions where time alignment is impractical, but it should be noted that the specification of timing misalignment between items, or elements within an item, is not automatically supported.

The transmission order for all elements in a content package is the order of transmission at the output of each respective element encoder. This may be different for each element (notably in the case of MPEG-2

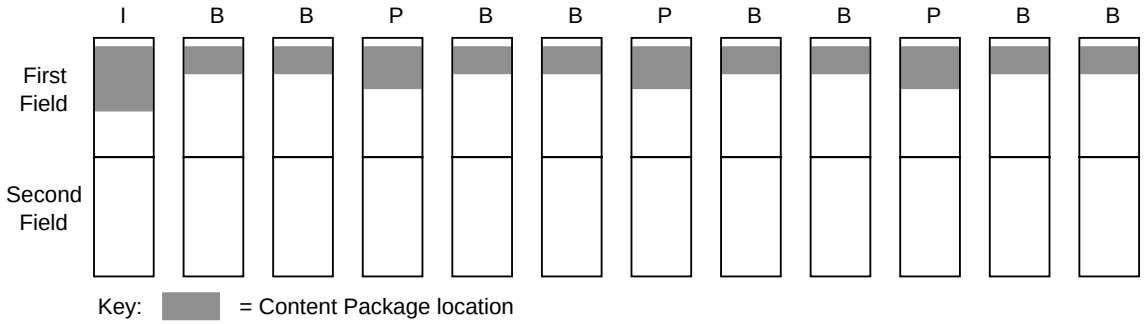

Figure 4 – Content package distribution over an example MPEG-2 GOP

encoding using B frames). Users of this standard should be aware that the elements of a content package may not be time aligned on a frame-by-frame basis and should take appropriate action to avoid timing errors.

Metadata for any element within a content package shall maintain accurate frame association with the element data.

6 Picture, audio, and auxiliary item structure

The picture, audio, or auxiliary item type value is followed by a 4-byte word count and a 1-byte item header word which defines the number of elements in the item.

The item word count value for each item may be set to the length of the data block as defined by SMPTE 305M. However, when not known in advance, the word count value may be set to zero to indicate an undefined block length as defined by SMPTE 305M.

The value of the item header word is the element count and has the range 1 to 255 as shown in figure 5. An element count value of 0 is not a valid value and shall not be used. The maximum number of elements will be typically restricted by an associated decoder template document.

The element data block structure is shown in figure 6.

Each element data block starts with a 1-byte element type value, followed by a 4-byte word count of the element data block size, a 1-byte element number, and the element data.

The element word count has the same format as specified in SMPTE 305M and its value shall be the length of the element number and element data words. However, where the element word count value is not known at the point of transmission, it may be set to zero to indicate an undefined data block length.

The value of the element number shall lie in the range 0 to 255 and shall be unique among the elements within any item. It is recommended that the element number increment by one for each element in sequence in the item.

The element type and element data formats shall be as defined in SMPTE 331M.

7 System item structure

The system item contains content package metadata and control data together with metadata for each element of the picture, audio, and auxiliary items.

The system item structure is defined in figure 7. It shall start with a 7-byte system item header and may be followed by an SMPTE universal label, timing, package, picture, audio, and auxiliary metadata sets, and control as indicated by the system item bitmap word.

The seven bytes of the system item header define:

– a system item bitmap word;

– a content package rate word;

– a content package type word, including stream status flags;

– a channel handle word;

– a continuity count word.

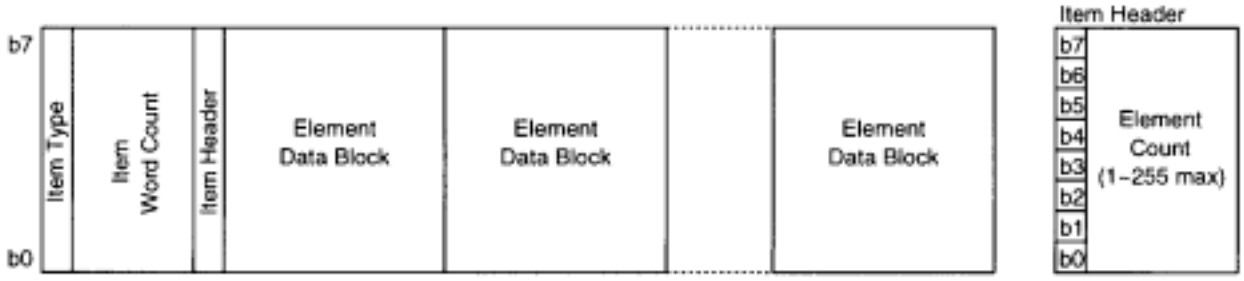

Figure 5 – Structure of picture, audio and auxiliary items

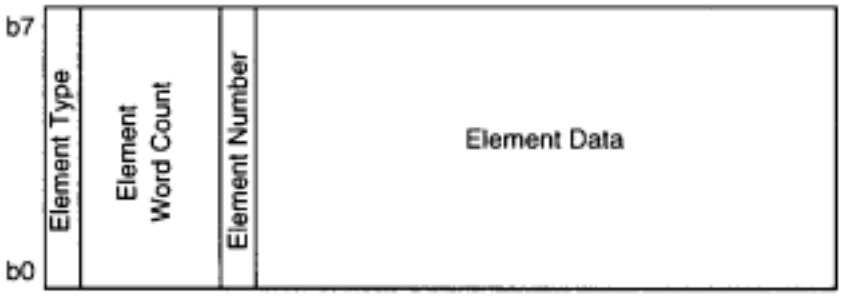

Figure 6 – Structure of element data blocks

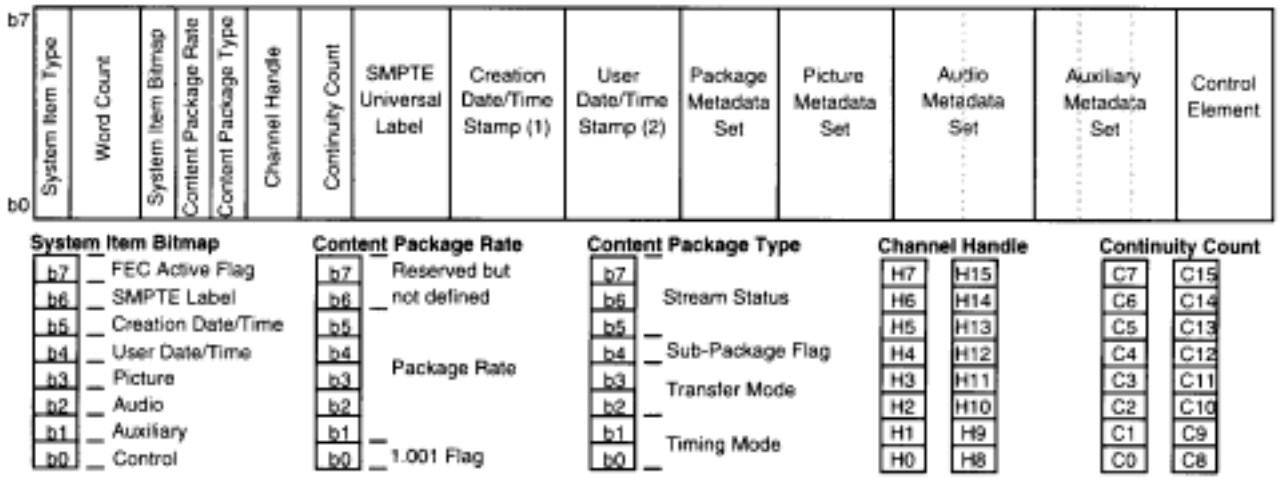

Figure 7 – System item structure

7.1 System item bitmap

The system item bitmap word is 1 byte in which bits b0-b7 have a default value of 0 indicating no data presence for the respective component. When a bit is set to 1, the corresponding part is present in the system item.

If bit b7 = 1, then forward error correction (FEC) shall be present and active;

If bit b6 = 1, then an SMPTE universal label shall be present;

If bit b5 = 1, then a creation date/time stamp shall be present;

If bit b4 = 1, then a user date/time stamp shall be present;

If bit b3 = 1, then a picture item shall be present and a picture metadata set may be present;

If bit b2 = 1, then an audio item shall be present and an audio metadata set may be present;

If bit b1 = 1, then an auxiliary item shall be present and an auxiliary metadata set may be present;

If bit b0 = 1, then a control element shall be present.

The data space associated with bits b6, b5, and b4 shall always be allocated. If bits b6, b5, or b4 are set to 0, then the value of the associated data space shall be deemed unusable.

The data space assigned to the remaining components shall be allocated if the associated bit is 1; otherwise no data space shall be allocated.

It is recommended that bit b5 be set to 1, and that a creation date/time stamp be entered correctly.

In the case of bits b3, b2, and b1, setting to 0 means that neither the associated item nor its metadata are present. When set to 1, the associated item is present and a metadata set is present. An item with no metadata will have a metadata set which indicates no metadata content.

7.2 Content package rate

The content package rate word is 1 byte which shall be used to identify the rate of content packages in baseline operation. It shall not be used to identify the frame rate of the SDTI transport structure.

Bits b7 and b6 are not defined, but are reserved for future use.

Bits b5 to b1 shall define the content package rate per second in baseline operation. These 4 bits identify 16 states defined as follows:

0 = undefined value

1 = 24	2 = 25	3 = 30
4 = 48	5 = 50	6 = 60
7 = 72	8 = 75	9 = 90
10 = 96	11 = 100	12 = 120

13-31 = reserved, but not defined.

Bit b0 identifies whether the content package rate is an exact value or offset by a factor of 1.001 and has the following values:

– 0 if the package rate is exact;

– 1 if the package rate is reduced by a factor of 1.001.

7.3 Content package type

The content package type word is 1 byte which shall be used to identify key aspects of the type and status of content package transfer.

Bits b7 to b5 shall define the position of the current content package in a stream of content packages. These 3 bits identify eight stream states as follows:

0 = the content package position in a stream is undefined;

1 = the content package is a stream head package which is any package which precedes the stream start package (e.g., preroll packages);

2 = the content package is a stream start package which is the first package of a stream;

3 = the content package is a midstream package which is any package between the stream start and stream end packages;

4 = the content package is a stream end package which is the last package of a stream;

5 = the content package is a stream tail package which is any package which follows the stream end package (e.g., postroll packages);

6 = the content package is both a stream start package and a stream end package signifying a stream of length 1;

7 = reserved but undefined.

Bit b4 shall be default 0 for all modes except low-latency transfer mode. The operation of this bit is described in 9.3.

Bits b3 and b2 identify the content package transfer mode. These 2 bits identify four states defined as follows:

0 = synchronous transfer mode: The package rate is temporally locked to the transport rate and each package is spatially locked in the defined position set by the content package timing mode;

1 = isochronous transfer mode: The package rate is not temporally locked to the transport rate, but each package is spatially locked in the defined position set by the content package timing mode;

2 = asynchronous transfer mode: The package rate is not temporally locked to the transport rate and packages are not spatially locked in defined positions;

3 = low-latency mode: Content packages are divided into subpackages and distributed throughout the transport frame period.

Bits b1 and b0 identify the content package timing mode. The timing modes are only valid for synchronous and isochronous transfer modes. These 2 bits identify four states defined as follows:

0 = normal timing mode: One content package starting on the defined first field start position;

1 = advanced timing mode: One content package starting on the defined second field start position;

2 = dual timing mode: Two content packages starting on the defined first and second field start positions;

3 = reserved but not defined.

Table 1 identifies the valid combinations of timing and transfer modes together with the clauses where they are described.

NOTE – Although all receivers shall be capable of receiving the structure of content packages in baseline operation, not all timing or transfer modes may be supported by all receivers. Even though a receiver may be capable of receiving special transfer modes, the decoder may not be able to manage correctly the decoded data to a satisfactory level as it may require special processing operations. The range of content package timing and transfer modes supported by a receiver/decoder is defined in an associated decoder template document.

Table 1 – Application of content package timing and transfer modes

	Transfer mode			
Timing mode	Synchronous	Isochronous	Asynchronous	Low-latency
Normal timing	Baseline operation	9.1.1	9.2	9.3
Advance timing	8.1	9.1.2		
Dual timing	8.2	9.1.3		
Reserved	N/A	N/A		

7.4 Channel handle number

The channel handle word consists of 2 bytes allowing a number in the range 0 to 65535 (bits H15 to H0 in figure 7). This number shall be used to distinguish different content packages having the same source and destination addresses and sent over the same SDTI link. The default value is 0000_h for single-channel transfers.

A channel handle value other than zero shall not be used until a standard or recommended practice is approved for its consistent application.

7.5 Continuity count

The continuity count word consists of 2 bytes allowing a number to be created by a modulo 65536 counter (bits C15 to C0 in figure 7). The continuity count shall increment by 1 for each newly transmitted content package with the same SDTI source and destination addresses. The continuity count may be used to detect whether the content package sequence has been broken by an operation such as a routing switch.

7.6 SMPTE universal label

The SMPTE universal label is a unique code which identifies the bitstream as a content package together with an associated decoder template.

The format comprises 16 bytes of the universal label as defined in ANSI/SMPTE 298M. If an SMPTE universal label is not present, then all 16 bytes shall be filled with 00_h and bit b6 of the system itembitmap word (7.1) shall be set to zero.

The SDTI content package has a base label value. Any associated decoder template will identify three further words of the label which shall be defined as:

– the decoder template class value;
– the decoder template type value; and
– the decoder template extension value.

The full SMPTE universal label string of 16 bytes shall be used to identify this as an SDTI content package format specified to a decoder template having the value indicated by the string values given in table 2.

Table 2 – Specification of the content package label

Byte No.	Description	Value (hex)
1	Object identifier	06_h
2	Label size	$0E_h$
3	ISO organization	$2B_h$
4	Designation: SMPTE	34_h
5	Registry: Labels	04_h
6	Labels category: Interchange	01_h
7	Labels registry	01_h
8	Registry version	01_h
9	Wrapper labels	01_h
10	Simple wrapper labels	01_h
11	Standard: CP	01_h
12	CP version	01_h
13	Template class	XX_h
14	Template type	XX_h
15	Template extension	XX_h
16	Zero fill	00_h

This label will allow a receiver/decoder to identify the content package template.

The template class value refers to an associated decoder template document and the template type value refers to the clause within an associated decoder template document. The template extension value is 00_h where the encoded content package lies within the limits defined by the template type. A template extension value of 01_h defines that the encoded content package can be received by the specified decoder, but that there are additional backwards compatible extensions. This facility is provided where decoders have capabilities beyond the defined template type, but do not meet the capabilities defined by any other template type.

7.7 Creation and user date/time stamps

The date/time metadata components are a creation date/time stamp and a user date/time stamp where each applies to the content package timing rather than the timing of individual items or elements.

Each date/time stamp metadata has a fixed 17-byte allocation which shall always be assigned. The date/time stamp metadata type is identified by the first byte and the remaining 16 bytes shall contain the date/time stamp data. The date/time stamp metadata type and data format are defined in SMPTE 331M. If bit b5 or b4 of the system item bitmap word (7.1) is set to zero, then all 17 bytes of the associated date/time stamp shall be set to 00_h. This creates a special case where a metadata type has a value of 00_h.

It is recommended that the creation date/time stamp records the date and time of content package creation. Once set, the creation date/time stamp shall not be modified in subsequent operations.

The user date/time stamp shall be used as defined by the application requirements. Space shall be allocated for the user date/time stamp although insertion of the stamp value is optional.

Note that it is possible (and likely) that some or all of the elements of the content package may have their own date/time stamp data.

7.8 Package, picture, audio and auxiliary metadata sets

The first metadata set shall be the package metadata set. The package metadata set shall contain metadata pertaining to the content package as a whole, such as, for example, a program title.

The package metadata set shall be followed by picture, audio, and auxiliary metadata sets which shall follow the order of picture, audio, and auxiliary items in the content package. Picture, audio, and auxiliary metadata sets shall be present only if the associated picture, audio, and auxiliary item is present in the content package as indicated by bits b3, b2, and b1 of the system item bitmap word. It is recommended that metadata blocks associated with any element appear in the same order as the element within the item. There may be more than one metadata type associated with each element type.

Each metadata set starts with a 1-byte metadata count which defines the number of metadata blocks in the set. A metadata count value of 00_h is a valid value and indicates that a metadata set has no metadata blocks. In this case, the metadata set is only one word long. Any metadata set shall consist of at least one word defining the metadata count value.

For coding efficiency, there is no word count value to define the length of a metadata set. Figure 8 illustrates the structure of package, picture, audio, and auxiliary metadata sets.

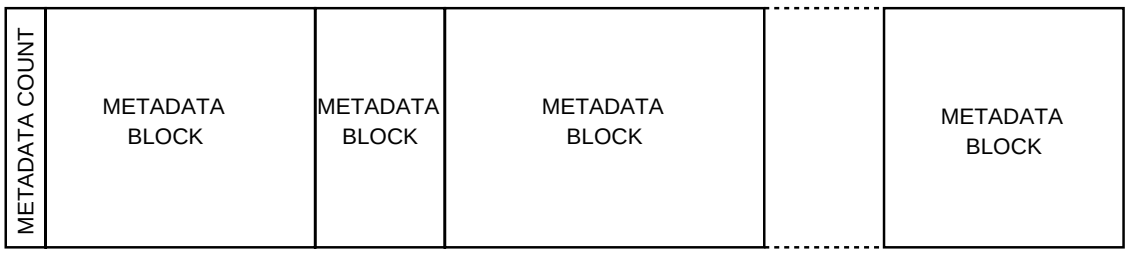

Figure 8 – Structure for metadata sets

7.8.1 Structures for package, picture, audio, and auxiliary metadata blocks

All metadata blocks shall consist of a 1-byte metadata type word, followed by a 2-byte word count word and completed by the metadata.

A metadata type value of 00_h is not permitted. Nonzero values of metadata type are defined in SMPTE 331M.

The word count value identifies the length of the metadata in bytes and shall be entered correctly.

All metadata blocks associated with any one picture, audio, and auxiliary element are grouped in sequence and each sequence is immediately preceded by a metadata link item to indicate the element to which the sequence refers. The metadata link item shall consist of a 1-byte metadata type word followed immediately by the element type and element number to which the sequence of metadata blocks refers. The element type and element number provide an unambiguous link between the metadata block sequence and the associated element.

The left side of figure 9 illustrates the structure of the metadata link item. The common structure for package, picture, audio, and auxiliary metadata blocks is illustrated on the right side.

7.9 Control element

The control element shall comprise a 1-byte element type value followed by a 4-byte word count and the control element data. The 4-byte word count value shall be entered correctly. The control element type and data format shall be as defined in SMPTE 331M.

The presence of a control element shall be as specified in an associated decoder template document.

8 Special timing modes

The modes in this clause apply only to the synchronous transfer mode.

8.1 Advanced timing mode

The advanced timing mode defines the content package timing to be advanced by 1 field. This mode can be used in certain operations to reduce system timing. Figure 10 illustrates the timing of this mode.

The content package start lines for 525/60 and 625/50 SDTI transports are defined as follows:

	525/60	625/50
Content package start:	Line 276	Line 322

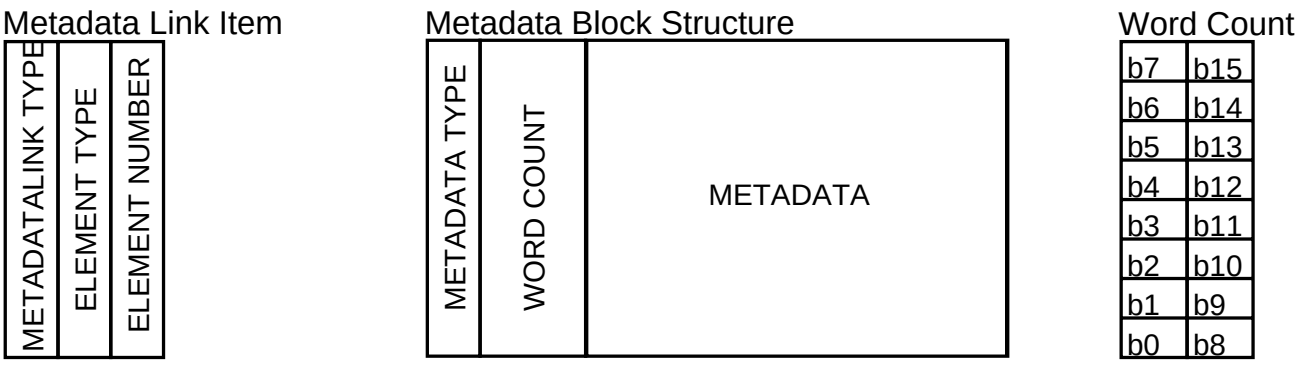

Figure 9 – Structures for the metadata link item and metadata blocks

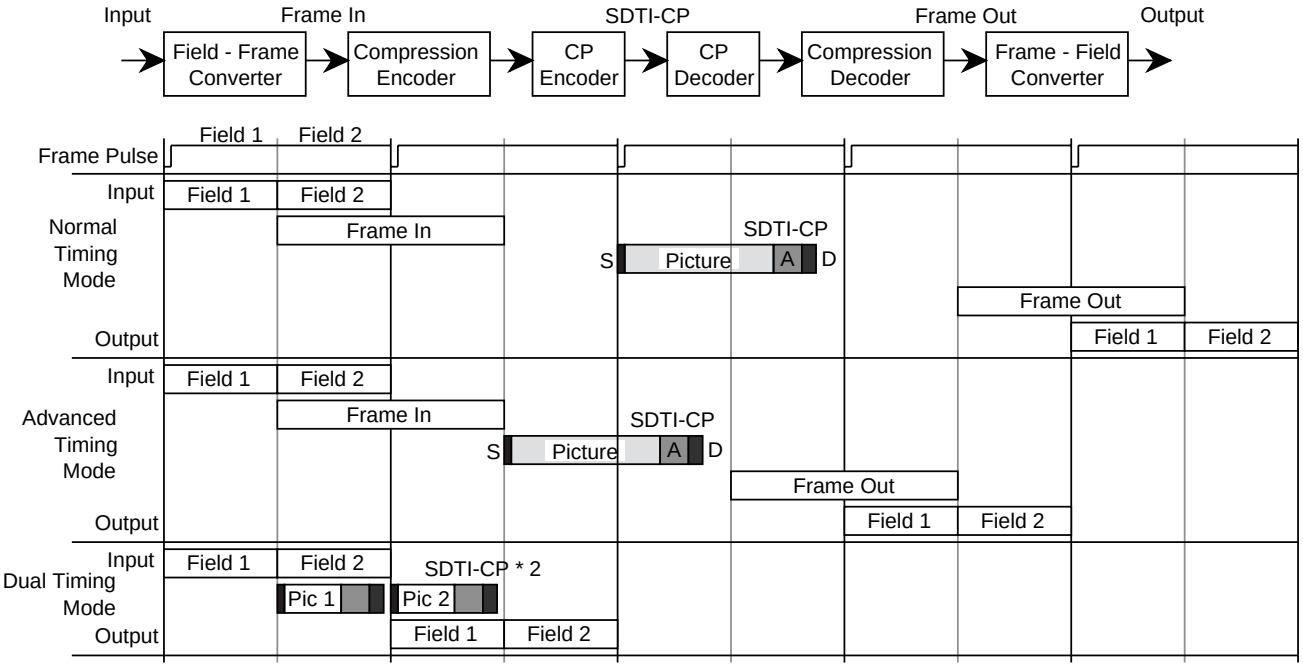

Figure 10 – Example timing diagrams for normal, advanced and dual timing modes

An associated decoder template document defines the capability to support the advanced timing mode.

8.2 Dual timing mode

The content package optionally may represent the contents of either a field of interlaced scanned picture data, or a frame of progressively scanned picture data (at 50-Hz or 60-Hz frame rate). This representation is called the dual timing mode. The general arrangement of system, picture, audio, and auxiliary items in dual timing mode is shown in figure 11.

For the dual timing mode, there are two content packages per frame with each occupying a different field and starting on the following lines:

	525/60	625/50
First field content package start:	Line 13	Line 9
Second field content package start:	Line 276	Line 322

A timing diagram of this mode is shown in figure 10.

All receivers shall be able to receive the content package on the first field of the SDTI. However, reception of the content package on the second field may not be supported by all receivers. An associated decoder template document defines the capability to fully support the dual timing mode.

8.3 Timing diagram for timing modes

The normal, advanced, and dual timing modes do not require decoder buffer delay metadata as they are defined by the synchronous nature of the SDTI transport. The timing diagram of figure 10 gives examples of the timing of the uncompressed source, content package bitstream, and decoded uncompressed outputs for all three modes, thus providing designers with a means of defining the required decoder buffer size.

The timing alignments show the minimum possible delays in each timing mode. Larger delays will be typically encountered where the input may be an integer number of frames earlier and the output an integer number of frames later than shown in figure 10.

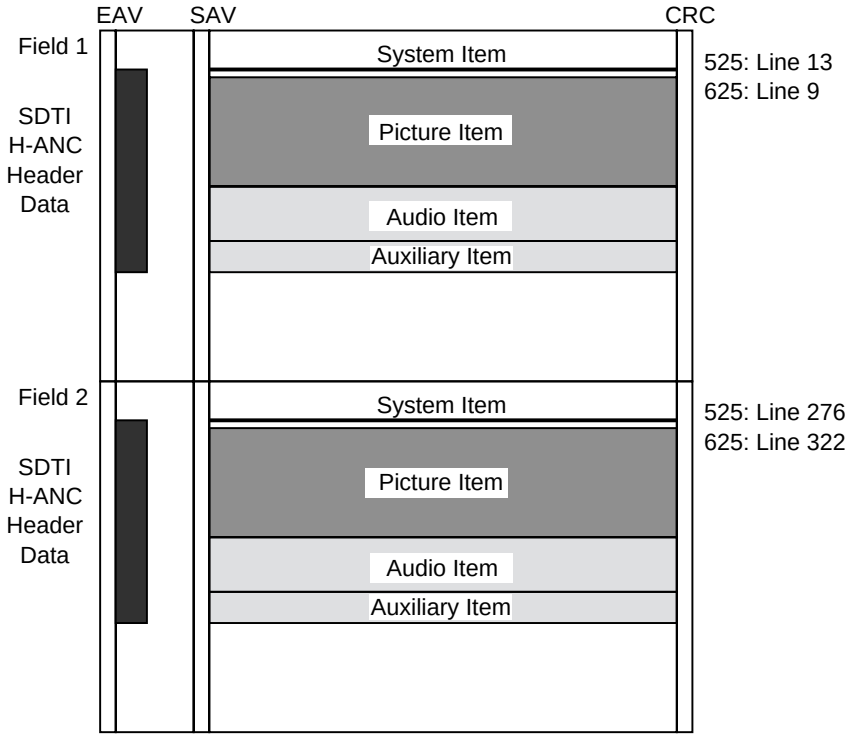

Figure 11 – Arrangement of items for dual timing mode content packages

9 Special transfer modes

When content packages are transmitted synchronous to the SDTI frame rate, the start line is defined by the timing mode. This standard allows content packages to be transmitted slower and faster than the SDTI frame rate. Furthermore, there is the capability for content packages containing different program material from the same or different source equipment to be multiplexed onto the same SDTI link. This standard provides the following points to aid correct operation of such modes:

– With the exception of the low-latency mode, each content package shall be transmitted as a continuous sequence of SDTI lines. No content packages shall suffer breakup as a result of multiplexing or any other operation except in the case of the low-latency mode.

– Demultiplexing of multichannel sources can be achieved by using the SDTI source and destination addresses. Different content package sequences multiplexed onto the same SDTI link and having identical source and destination addresses shall have different channel handle numbers.

In these special transfer modes, decoders must provide sufficient buffering for the intended application area and this buffering will typically be greater than that required for synchronous transfers. It should also be noted that these special transfer modes may require special decoding processes to ensure adequate signal recovery to be able to present viewable pictures and acceptable sound quality.

9.1 Isochronous transfer mode

Isochronous transfers are defined as the transfer of content packages at nonnormal transfer rates, such as slower or faster than real time, while retaining their lock to the spatial alignment position; i.e., the system item always appears on the defined start line. It should be noted that in the case of transfer rates which are changing (e.g., during a tape speed transition), transient conditions may occur which do not meet the specifications set in the remainder of this clause. Decoders preferably shall have sufficient buffering to be able to manage transient states without undue perturbations in the buffer output. In the descriptions following, the number N represents the ratio of the transfer speed to real time transfer in baseline operation.

Thus, a value of N=2 means that the transfer speed is twice the real time transfer speed.

9.1.1 Normal timing mode

In slower-than-real-time transfer, the start of each content package shall be at every 1/Nth frame on the defined line for normal timing mode. Where N is not an integer value, it is preferred that the distribution of content packages over time be as even as possible with the number of frames between content packages differing by less than one for any given transfer speed up to unity.

In faster-than-real-time transfer, the start of every Nth content package shall be on the defined line for normal timing mode followed immediately by the next N-1 content package. Where N is not an integer value, it is preferred that the number of N content packages in a frame period be as evenly distributed as possible with the number differing by less than one for any given transfer speed above unity.

9.1.2 Advanced timing mode

In slower-than-real-time transfer, the start of each content package shall be at every 1/Nth frame on the defined line for advanced timing mode. Where N is not an integer value, it is preferred that the distribution of content packages over time be as even as possible with the number of frames between content packages differing by less than one for any given transfer speed up to unity.

In faster-than-real-time transfer, the start of every Nth content package shall be on the defined line for advanced timing mode followed immediately by the next N-1 content package. Where N is not an integer value, it is preferred that the number of N content packages in a frame period be as evenly distributed as possible with the number differing by less than one for any given transfer speed above unity.

The timing modes described in this clause are illustrated in outline form in figure 12.

9.1.3 Dual timing mode

In slower-than-real-time transfer, the start of each content package shall be at every 1/Nth field on the defined line for dual timing mode. Where N is not an integer value, it is preferred that the distribution of content packages over time be as even as possible with the number of fields between content packages differing by less than one for any given transfer speed up to unity.

In faster-than-real-time transfer, the start of every Nth content package shall be on the defined line for dual timing mode followed immediately by the next N-1 content package. Where N is not an integer value, it is preferred that the number of N content packages in a field period be as evenly distributed as possible with the number differing by less than one for any given transfer speed above unity.

The timing modes described in this clause are illustrated in outline form in figure 12.

9.2 Asynchronous transfer mode

Asynchronous transfers are used to transfer content packages in the correct sequence, but with an arbitrary timing not locked to frame or field boundaries. Such transfers require a specialist buffer or a storage medium capable of receiving packages with an undefined separating space between packages. By the nature of its definition, an asychronous transfer overrides the relevance of the timing modes. Figure 12 illustrates the use of the asychronous transfer mode.

The ability of a receiver to decode the structure of content packages in the asychronous transfer mode is defined in an associated decoder template document.

Asynchronous transfer mode also permits the application of nonreal time transfers of content packages whose size exceeds an SDTI frame length. Any application requiring the transfer of content packages whose size exceeds the length of an SDTI frame shall be defined by a specific decoder template recommended practice for that application.

9.3 Low-latency transfer mode

To reduce system delay, content packages may be divided into a sequence of subpackages. Each subpackage shall contain a system item together with one or more subitems. Subitems are defined as per an item, but have a fraction of the frame based data content associated with baseline operation.

The system item of the first subpackage shall be as defined for baseline operation and shall contain any metadata as appropriate for the content package as a whole.

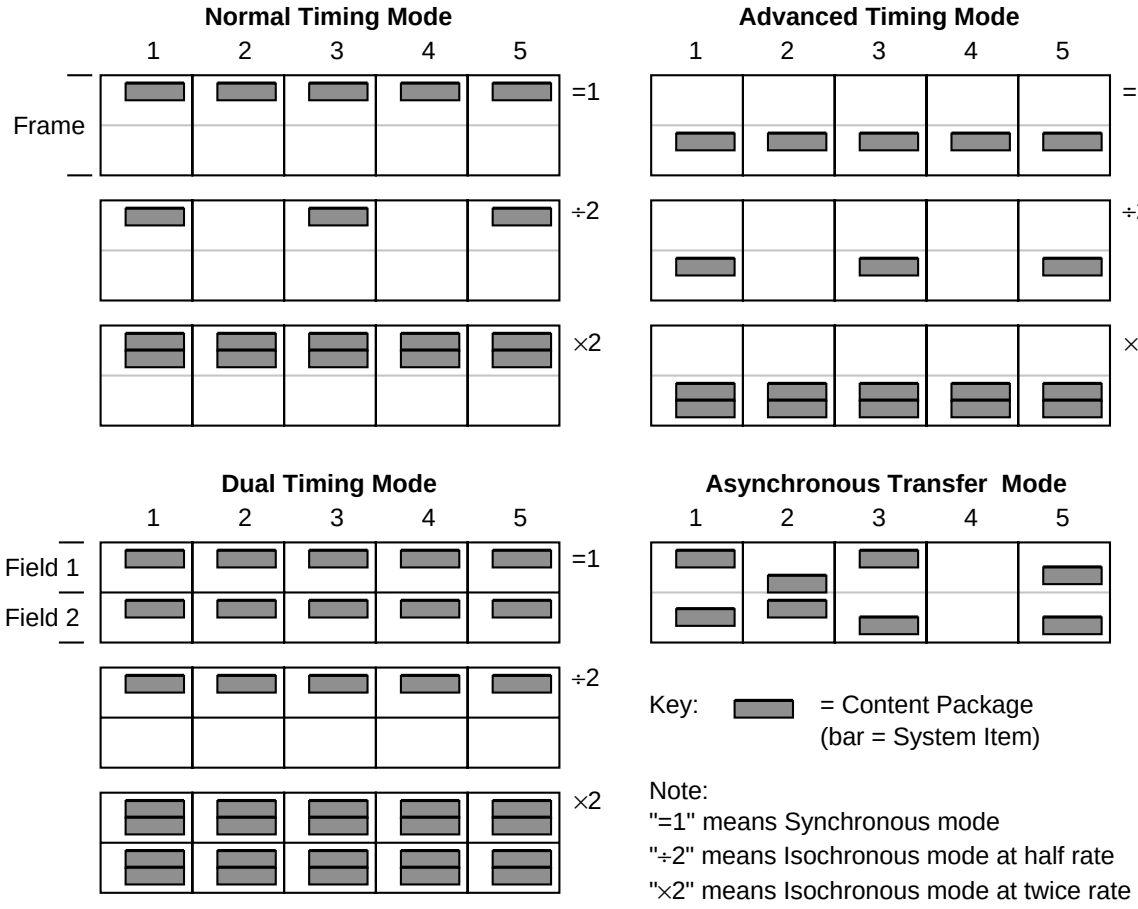

Figure 12 – Illustration of timing and transfer modes

Bit b4 of the content package type word in the system item shall be set to 0 of the first subpackage.

The system item of subsequent subpackages shall contain at least the first 7 header bytes from the system item bitmap to the continuity count inclusive. The remaining parts of the system item are optional, but shall be ignored by a decoder if present.

Bit b4 of the content package type word shall be set to 1 for the system item of these subsequent subpackages. All other data in the 6 header bytes following the system item bitmap word shall be identical to the corresponding words in the system item of the first subpackage.

9.3.1 Buffer delay

To support low-latency transfer mode, a decoder delay metadata item may explicitly define the buffer delay of the receiver. This metadata item is specified in SMPTE 331M.

The buffer delay operation is initiated by the reception of the first subitem sample written into the item buffer. The buffer read operation is set to start reading the buffer data when the decoder delay time has been equaled or exceeded. The decoder delay time is specified in units of a 90-kHz clock.

An example of low-latency transfer mode subpackages is illustrated in figure 13.

The ability of a receiver to receive and parse content packages in the low-latency transfer mode is defined in an associated decoder template document. If an associated decoder template document allows low-latency transfer mode, it may also restrict the number and size of subitems and the order of subitems in a subpackage.

10 Error correction

Forward error correction (FEC) may optionally be added to the payload area where extra security is required.

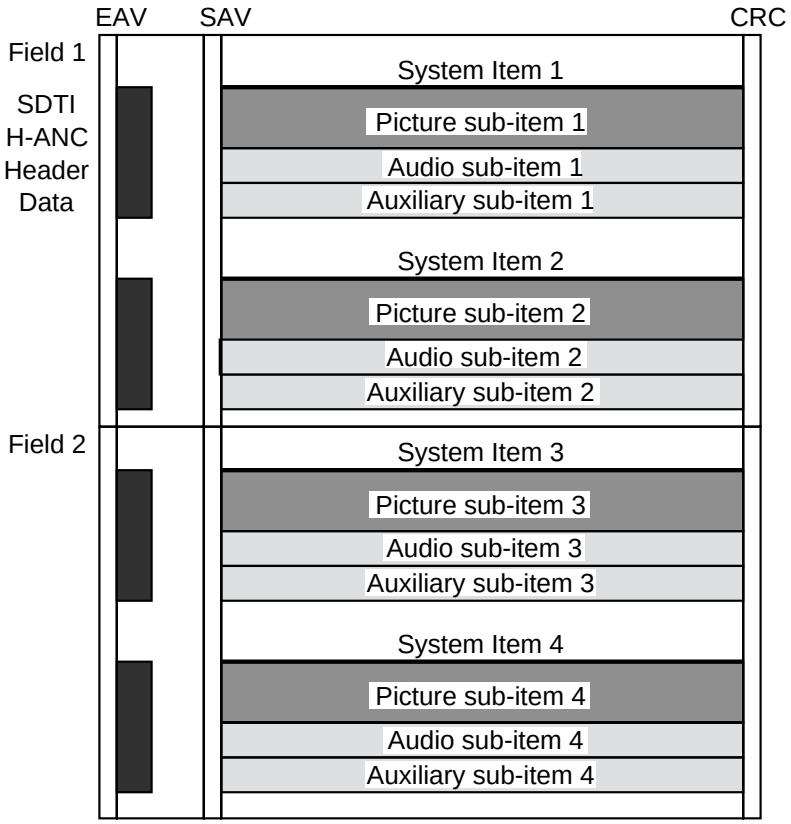

Figure 13 – Illustration of the low-latency transfer mode

A case where FEC might be added is where the payload data have no in-built redundancy; e.g., long GOP MPEG-2 coded pictures and vital IT data files.

A case where FEC need not be added is where the payload data have some residual redundancy; e.g., uncompressed or lightly compressed picture or audio signals.

10.1 Error correction format

FEC shall only be applied to the least significant 8 bits of the 10-bit interface which have previously been identified as bits b0 to b7. The payload data area shall be grouped into 240 word FEC blocks as illustrated in figure 14. The first 234 words of each FEC block shall be used for content package data and the last 6 words are used for the Reed-Solomon FEC. For all 240 words of the FEC block, bits b8 and b9 shall be as defined by SMPTE 305M.

FEC blocks will apply across the whole of any line associated with the content package data. Thus, for an active line length of 1440 words, the available data space for content package data would be reduced to 1404 words. Any line associated with the content package will have FEC blocks for the whole line length regardless of whether the content package data occupies all the available data space.

The error correction is defined as a Reed-Solomon R-S (240, 234, T=3) shortened code from the original R-S (255, 249, T=3) code.

The R-S code generator polynomial shall be:

$$\text{R-S}(x) = (x \oplus a^0).(x \oplus a^1).(x \oplus a^2).(x \oplus a^3).(x \oplus a^4).(x \oplus a^5)$$

where a is defined by the Galois field GF(256) generator polynomial:

$$GF(x) = x^8 \oplus x^4 \oplus x^3 \oplus x^2 \oplus 1.$$

Each word of a 240-word FEC block shall have bits b8 and b9 set according to SMPTE 305M.

Figure 14 illustrates the application of FEC to the payload area.

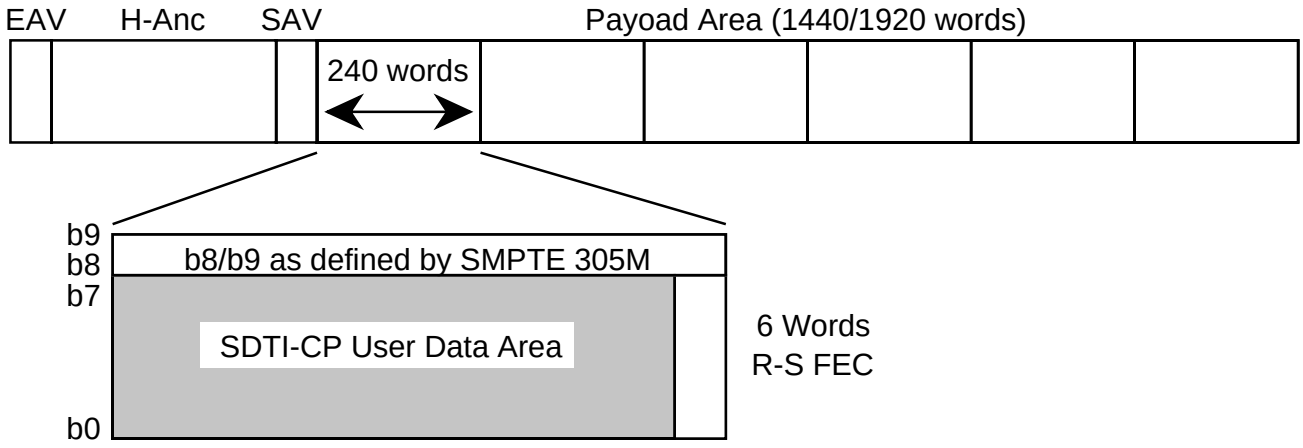

Figure 14 – Specification of the optional SDTI-CP FEC

10.2 FEC Identification

Bit b7 of the system item bitmap word in the system item header is used to flag the application of FEC in the payload area of all lines used in the associated content package. In the case of low-latency mode, this also applies to all associated subpackages. Where a line has FEC applied, the FEC shall apply to all the FEC blocks in that line.

If the SDTI CRC and the SDTI-CP FEC are both active, the last two words of the last FEC block in any line may interfere with the CRC words. To pevent this condition, the following rules shall be applied to any line carrying content package data:

– If the FEC is active, then the SDTI CRC shall not be used and the last 2 bytes of the last FEC block on the line will overwrite the CRC location. The payload CRC flag of SMPTE 305M shall be set to 00_h.

– If the FEC is not active, then the line may apply the SMPTE 305M CRC and set the payload CRC flag accordingly.

10.3 Word count values

The word count value of the system, picture, audio, and auxiliary items are defined by the variable block structure of SMPTE 305M. Where FEC is applied, the word count values of items shall include all FEC words which are contained by a variable block in order to comply with SMPTE 305M.

In the case of word counts defined solely by this standard, the values shall represent the block sizes without including any FEC words which may be present.

Annex A (normative)
Syntax of the content package structure as pseudo-code

The following pseudo-code illustrates the structure of the SDTI-CP fomat in baseline operation only. The values in square brackets indicate the length of the variable in bytes; [var] indicates a variable length packet.

```
For (content_package=start; content_package<=end;
  content-package++)
{
  System item()
  {
    Separator [1]
    System item type [1]
    Word count [4]

    System item bitmap [1]
      FEC_flag = 80h and system item bitmap
      SMPTE_flag = 40h and system item bitmap
      Creation_TC_flag = 20h and system item bitmap
      User_TC_flag = 10h and system item bitmap
      Picture_flag = 08h and system item bitmap
      Audio_flag = 04h and system item bitmap
      Aux_flag = 02h and system item bitmap
      Control_flag = 01h and system item bitmap

    Content package rate [1]
    Content package type [1]
    Channel handle [2]
    Continuity count [2]

    If (SMPTE_flag)
      SMPTE universal label [16]
    Else
      Null data [16]

    If (Creation_TC_flag)
      Type [1]
      Creation date/time code [16]
    Else
      Null data [17]

    If (user_TC_flag)
      Type [1]
      User date/time code [16]
    Else
      Null data [17]

    Package metadata ()
    {
      Metadata_count [1]

      for (metadata=0; metadata<metadata_count; metadata++)
      {
        Metadata type [1]
        Metadata word count [2]

        Package metadata [var]
      }
    }

    If (picture_flag)
    {
      Picture metadata ()
      {
        Metadata_count [1]

      for (metadata=0; metadata<metadata_count; metadata++)
      {

        If (metadata_sequence_start)
        {
          Metadata link type [1]
          Element type [1]
          Element number [1]
        }

        Metadata type [1]
        Metadata word count [2]

        Picture metadata [var]
      }
    }
  }

  If (audio_flag)
  {
    Audio metadata ()
    {
      Metadata_count [1]

      for (metadata=0; metadata<metadata_count; metadata++)
      {
        If (metadata_sequence_start)
        {
          Metadata link type [1]
          Element type [1]
          Element number [1]
        }

        Metadata type [1]
        Metadata word count [2]

        Audio metadata [var]
      }
    }
  }

  If (aux_flag)
  {
    Aux metadata ()
    {
      Metadata_count [1]

      for (metadata=0; metadata<metadata_count; metadata++)
      {
        If (metadata_sequence_start)
        {
          Metadata link type [1]
          Element type [1]
          Element number [1]
        }

        Metadata type [1]
        Metadata word count [2]

        Aux metadata [var]
      }
```

```
    }
  }

  If   (control_flag)
  {
    Control  element  ()
    {
      Element  type  [1]
      Element  word  count  [4]
      Element  packet  [var]
      }
    }

    End  code  [1]
  }

  If   (picture_flag)
  {
    Picture   item
    {
      Separator   [1]
      Picture  item  type  [1]
      Item  word  count  [4]
      element_count   [1]

      for  (element=0;  element<element_count; element++)
      {
        Element type [1]
        Element word count [4]
        Element number [1]

        Picture element data [var]
      }

      End code [1]
    }
  }

  If (audio_flag)
  {
    Audio item
    {
      Separator [1]
      Audio item type [1]
      Item word count [4]
      element_count [1]

      for (element=0; element<element_count; element++)
      {
        Element type [1]
        Element word count [4]
        Element number [1]

        Audio element data [var]
      }

      End code [1]
    }
  }

  If (aux_flag)
  {
    Aux item
    {
      Separator [1]
      Aux item type [1]
      Item word count [4]
      Element_count [1]

      for (element=0; element<element_count; element++)
      {
        Element type [1]
        Element word count [4]
        Element number [1]

        Aux element data [var]
      }

      End code [1]
    }
  }
}
```

Annex B (informative)
Bibliography

ANSI/SMPTE 259M-1997, Television — 10-Bit 4:2:2 Component and 4f_{sc} Composite Digital Signals — Serial Digital Interface

SMPTE 291M-1998, Television — Ancillary Data Packet and Space Formatting

ISO/IEC 13818-2:1996, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video

ISO/IEC 13818-2:1996, Amendment 2:1997 (MPEG-2, 4:2:2P@ML)

EBU/SMPTE Task Force for Harmonized Standards for the Exchange of Program Material as Bitstreams, Final Report: Analyses and Results, July 1998, SMPTE J. 107(9):603-815; 1998 September