SMPTE 332M-2000

SMPTE STANDARD

for Television —
Encapsulation of Data Packet Streams over SDTI (SDTI-PF)

1 Scope

This standard specifies an open framework for encapsulating data packet streams and associated control metadata over the SDTI transport (SMPTE 305M). Encapsulating data packet streams on SDTI allows them to be routed through conventional SDI (SMPTE 259M) equipment.

This standard specifies a range of packet types which may be carried over SDTI. The range of packet types carried may be expanded as requirements develop. The standard does not attempt to specify the payload contained in any packet. It offers options to add specific information to each individual packet including localized user data space, forward error correction (FEC), and a mechanism for accurate packet retiming at the decoder.

The standard also offers a limited capability for metadata to be added providing packet control information to aid the successful transfer of packets. The specification of the metadata follows the K-L-V approach of the SMPTE metadata dictionary and provides extensibility for future requirements.

For easy reference, this standard is abbreviated to the term SDTI-PF.

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 259M-1997, Television — 10-Bit 4:2:2 Component and $4f_{SC}$ Composite Digital Signals — Serial Digital Interface

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

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

3 General specification

Most packet streams do not have critical timing requirements and a decoder can output packets in the order in which they were encoded, but with increased packet jitter resulting from the buffering of packets onto SDTI lines. The result of the SDTI-PF packet encapsulation process is to introduce both delay and jitter to the packet stream. However, MPEG-2 transport stream (MPEG-2 TS) packets are one case where a relatively small packet jitter specification is required to ensure minimal impact on MPEG-2 transport stream clock recovery and buffering circuits. This standard contains provisions to allow the packet jitter to be reduced to insignificant levels while the delay is an issue addressed by the method of packet buffering at the encoder. As a benchmark, the specification is defined so that a low packet jitter source, such as DVB-ASI, can be carried through the SDTI-PF and be decoded to create an output with negligible packet jitter.

Although MPEG-2 TS packets are the most critical packet type for decoder timing accuracy, this standard

Approved
January 31, 2000

also allows for other kinds of packets to be carried over the SDTI, with or without buffering, to reduce packet jitter. Such packets may be ATM cells and packets based on the unidirectional internet protocol (Uni-IP).

This standard provides several features for the carriage of packet streams such as:

– A high packing density;

– A flexible arrangement of data packet types and associated packet metadata;

– Carriage of multiple channels of multiple data packet types;

– Provisions for a powerful forward error correction (FEC) specification;

– The ability to reproduce accurate data packet output timing;

– The ability to add embedded user information.

The control information associated with data packets can be used to:

– Provide a continuity counter to detect SDI switching;

– Provide a channel handle for transfers of multiple channels of data packets between the same source and destination addresses;

– Define the position of a data packet in a stream.

4 SDTI parameters

SMPTE 305M specifies the SDTI for bit rates of 270 Mb/s and 360 Mb/s. Equipment may operate on only 270 Mb/s and still be compliant with this standard.

The SDTI-PF uses the variable block mode of the SDTI as shown in figure 1. The separator and end code words are special 10-bit values defined in SMPTE 305M. All data in the space between the separator and end codes shall be 10-bit words where the data comprises 8 bits entered into bits b0 to b7 of the 10-bit word, b8 shall be set to even parity of bits b0 and b7, and b9 shall be set to odd parity.

The block type word of the SDTI header shall be set to variable block mode according to SMPTE 305M.

The SDTI data type word shall be set to the value 11_h indicating the data packet format (PF) variable block payload. This value shall be registered in the data types table of SMPTE 305M as a 10-bit word of value 211_h.

This standard shall not use the data extension facility of SMPTE 305M.

4.1 SDTI line and address numbers

Since the data in each SDTI variable block may 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 SDTI variable block.

4.2 SDTI switching

The arbitrary switching of SDTI data streams, although at the picture frame boundary, may affect the ability to decode successfully the contents of the data packets 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. It is recommended to avoid the use of the lines defined by SMPTE RP 168 together with the lines immediately prior to and following the switch line where transient conditions may occur. A continuity count can be provided to indicate variable blocks affected by a switch.

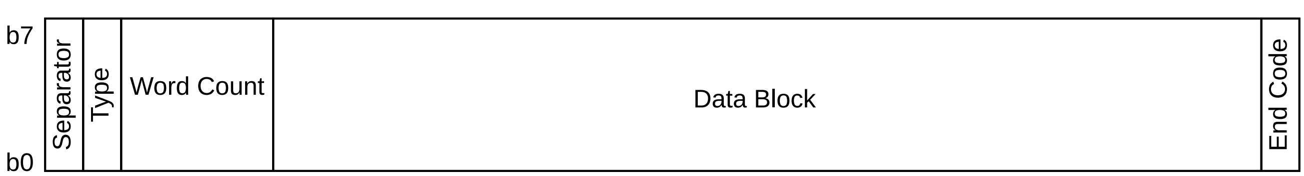

Figure 1 – Format of the SDTI variable block

5 TLD block structure

The data block area of each variable block shall be filled with one or more blocks defined by a type, length, and data (TLD) construct as shown in figure 2.

The three components of the TLD construct are:

– Type: The type of data contained in the value area as a local 1-byte label;

– Length: The length of the value; and

– Data: The data as defined by the type.

Type is a single byte which identifies the type of data carried in TLD blocks. The type value may identify either one kind of TLD packet (such as an MPEG-2 TS packet) or one kind of TLD metadata.

When a TLD metadata block is received, the metadata contained in the value area shall apply to all subsequent TLD packet blocks until either the end of the SDTI variable block or until a new TLD metadata construct of the same type is received. The data from a TLD metadata block do not carry any significance across SDTI variable blocks, so each new variable block shall carry a new set of TLD metadata as needed (see figure 3).

The data block area of each SDTI variable block shall be filled wholly with TLD blocks. There shall be no padding either at the start of the SDTI data block, between TLD blocks, or between the end of the last TLD block and the SDTI end code.

5.1 TLD type

The TLD type is a 1-byte word used to identify the type of data in the TLD block. The following rules shall be applied to the assignment of type values:

– A type value of 00_h shall not be used;

– Type values in the range 01_h to $0F_h$ shall be used to identify TLD metadata blocks;

– Type values in the range from 10_h to FF_h shall use only the most significant 4 bits of the type word to identify TLD packet types. Thus, only 15 of these TLD types can be identified.

The least significant 4 bits of type words with a value greater than $0F_h$ shall be defined as follows:

Bit b3 shall be used to identify the presence of an FEC at the end of the TLD data area.

– If b3 = 1, then the FEC shall be present, else the FEC is not present and no space is allocated.

Bit b2 shall be used to identify whether the data contain an embedded counter.

– If b2 = 1, then the embedded counter is present, else the embedded counter is not present and no space is allocated.

Bits b1 and b0 form a binary value in the range 0 to 3.

– If the value = 3, then 6 user bytes are contained at the head of the TLD data area;

– If the value = 2, then 4 user bytes are contained at the head of the TLD data area;

– If the value = 1, then 2 user bytes are contained at the head of the TLD data area;

– If the value = 0, then no user bytes are contained at the head of the TLD data area.

5.2 TLD length

The length of a TLD block specifies the length of the TLD data. Thus the length value can be used to skip to the next TLD block if needed.

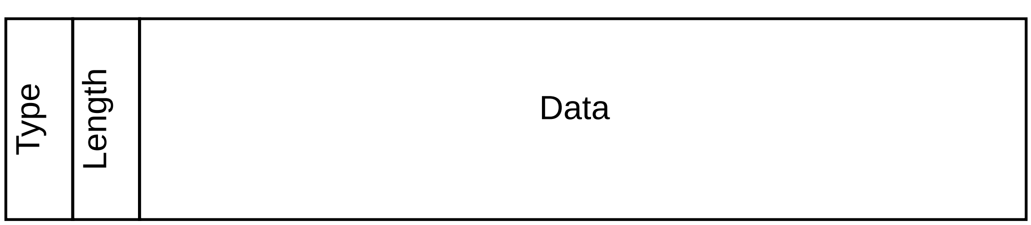

Figure 2 – TLD blocks

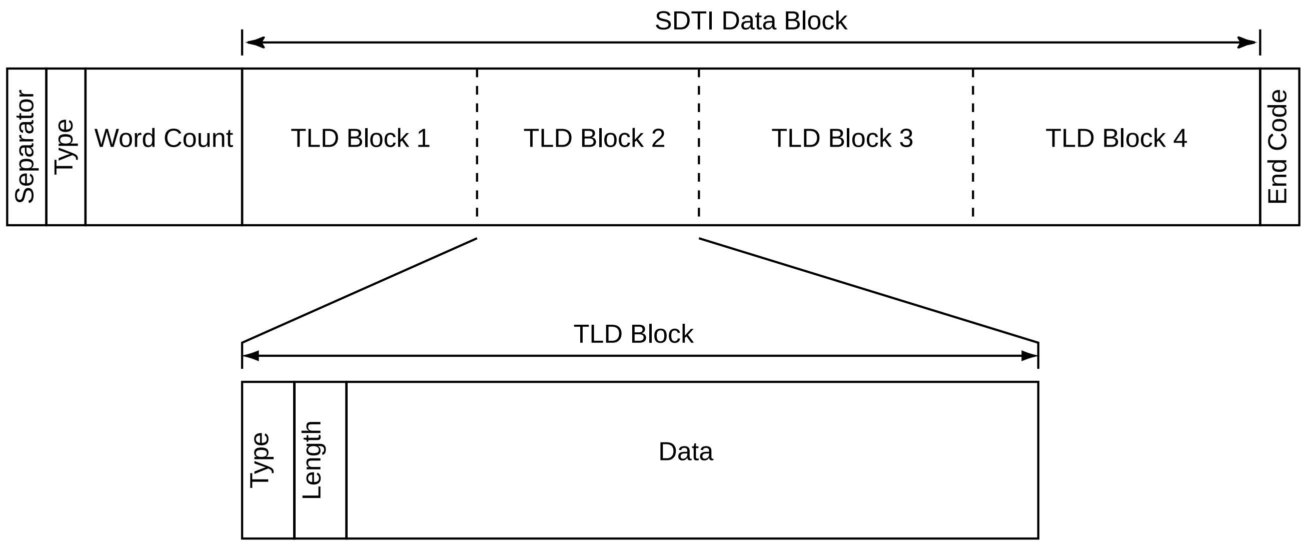

Figure 3 – Hierarchy of SDTI variable blocks and TLD blocks

The TLD length shall be a variable length word defined as follows:

– If the value of the first byte lies in the range 00_h to FE_h, then the length is given by the value of the first byte;

– If the value of the first byte is equal to FF_h, then the length value is contained in the 2 bytes which immediately follow.

Thus, TLD constructs with data blocks having a length greater than 254 bytes result in a length of 3 bytes of which the first byte is equal to the value of FF_h and the next two bytes contain a length whose value may range from 0000_h to $FFFE_h$. The 3-byte length value of $FFFFFF_h$ is reserved for future possible expansion.

5.3 TLD block

TLD blocks defined by a type value up to $0F_h$ contain metadata of a format defined in SMPTE 331M.

TLD blocks defined by a type value above $0F_h$ contain data packets as defined in the packet definitions.

A data packet may be formed from the following components:

– User data of length 0, 2, 4, or 6 bytes. The length of the user data shall be defined by bits b0 and b1 of the type word;
– A minor count value in combination with a major count value giving 3 bytes which can be used to retime the data packets at the decoder output with negligible jitter;

– A Reed-Solomon forward error correction (RS-FEC) of 6 bytes length.

The TLD packet is shown in figure 4.

At its simplest level, a data packet with a TLD type identifier which has the four LSBs set to 0 becomes a simple data packet containing only the packet data and no extra components.

5.3.1 User data area

User data space is defined in increments of 2 bytes allowing 0, 2, 4, or 6 bytes of user data area. The contents of the user data area are private data not defined in this standard.

5.3.2 Packet timing counter

The combination of major and minor count values forms the packet timing counter with integer and

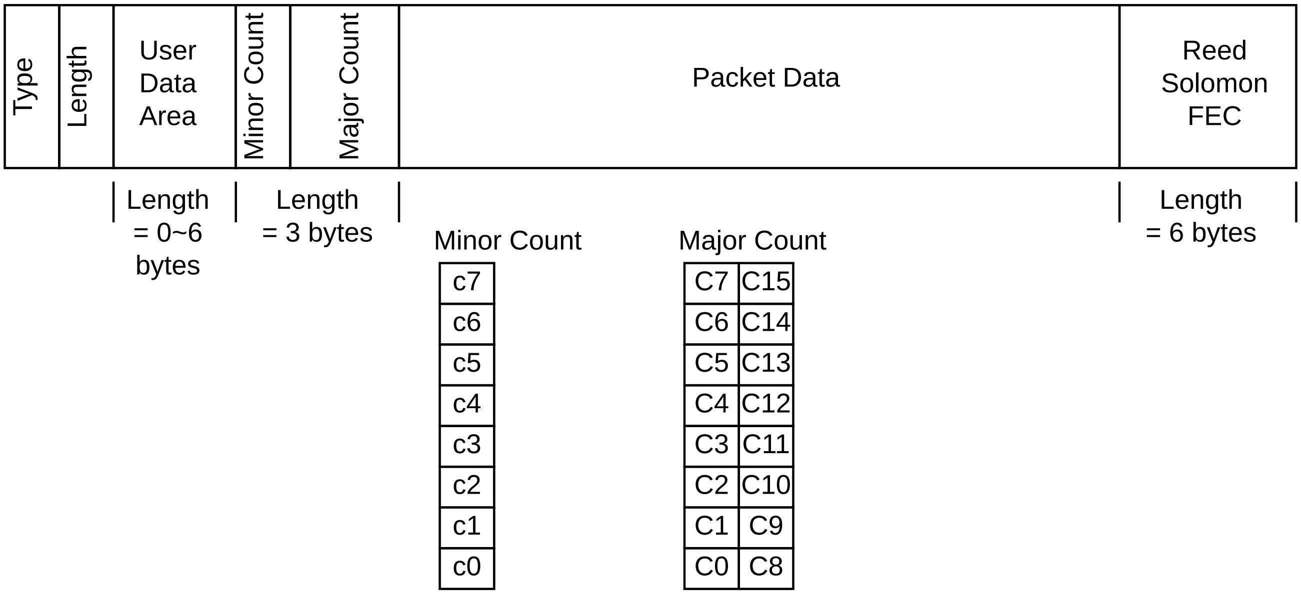

Figure 4 – TLD compound packet with all components

fractional parts defined by the major and minor count words respectively. The packet timing counter defines the timing of the start of each packet for a decoder to be able to output the packet at the correct time relative to the first packet in any stream. The packet timing counter can provide output packet start position timings which are as accurate as the SDTI clock period.

The packet timing counter has a cycle period of approximately 30 ms. The maximum period between packets shall be less than the packet timing counter cycle period to guarantee correct operation.

Annex A describes how the packet timing counter can be used for accurate output timing control.

5.3.2.1 Minor count

The 8-bit minor count word shall be a value sampled from a counter clocked by the SDTI word clock.

Bits c0 to c7 of the minor count word shall be used for the minor count value. Bit c0 shall be the least significant bit of the minor count value.

– In the case of 270 Mb/s SDTI, the minor count value is loaded from a counter clocked by a 27-MHz clock (which is at the same rate as the MPEG-2 TS PCR) and counting over the range 0~11 to create a 2.25-MHz clock period;

– In the case of 360 Mb/s SDTI, the minor count value is loaded from a counter clocked by a 36-MHz clock (which is at a higher rate than the MPEG-2 TS PCR) and counts over the range 0~15 to create a 2.25-MHz clock period;

– The number of bits available in the minor count word allows SDTI word clock rates up to 576 MHz.

5.3.2.2 Major count

The 16-bit major count word is a value sampled from a modulo 65536 counter clocked by the 2.25-MHz clock. The major count value is formed from bits C0 to C15 as shown in figure 4 representing, respectively, the LSB and MSB of the count value.

5.3.3 R-S FEC

When provided, the R-S FEC shall provide correction for the whole TLD structure content from the TLD type word to the last R-S FEC word. The R-S FEC shall not be interleaved over multiple TLD packets.

The error correction is defined as a Reed-Solomon R-S (L, L–6, T=3). Where L is less than 255, then the RS-FEC is a shortened code from the original R-S (255, 249, T=3) code.

The R-S code generator polynomial shall be:

$$R\text{-}S(x) = (x \oplus a^0) \,.\, (x \oplus a^1) \ldots (x \oplus a^5)$$

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

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

Note that although an encoder provides the capability of three error corrections, a decoder may choose to correct only one or two errors to preserve residual error detection performance.

Note also that the RS-FEC can be applied only to TLD structures of 255 bytes or less. A TLD block length value of greater than 253 is supported in this standard, but will exceed the RS-FEC limit and thus, where the TLD block length of a packet exceeds 253 bytes, the RS-FEC shall not be applied.

6 TLD block definitions

The following list of definitions is defined by this standard. Users of this standard should be aware that this list may be extended in future revisions to this standard.

Each TLD metadata definition has both a local identifier defined by the type value and a global identifier which defines the place of the metadata item in the SMPTE metadata dictionary. Both identifiers are referencing the same metadata specification. The reason for the shortened type value is for ease of parsing the data at the high speeds used by the SDTI transport. There is also a gain in packing density and hence simplified storage requirements on high-speed silicon. But it should be noted that any metadata item specified in this standard may be expanded to define the full K-L-V construct on which the metadata dictionary is based. This fully expanded K-L-V construct may then be used as a basis for the common interchange of metadata items between different applications.

The SMPTE metadata dictionary is currently under construction and the global key values given are tentative and may be subject to revision. Readers of this standard are encouraged to check with the published version of the metadata dictionary to confirm the accuracy of the global key values. For this reason, the global key values are shown for information only and shall not be regarded as definitive until this standard has been updated.

Note that the first 8 words of each global key value are the same and set at the value:

06_h, $0C_h$, $2B_h$, 34_h, 01_h, 01_h, 01_h, 01_h

To simplify the global key values, only the last 8 words are indicated in this standard. Thus, the first word of each global key value identifies the metadata dictionary class type.

6.1 Metadata definitions

6.1.1 Continuity count

Local type value: 01_h.

Global key value: 06_h, 01_h, 01_h, 01_h, 02_h, 00_h, 00_h, 00_h.

(Relational, numerical sequences, block numbers, continuity counts, 2-byte CC.)

Length: 2 bytes.

The continuity count is a 16-bit, modulo 65536 value which shall increment by one for every SDTI-PF variable block having the same SDTI source and destination addresses. The purpose of this count is to provide detection of signal path switching.

The bit significance is LSB first. Thus, the LSB lies at bit b0 of the first byte and the MSB lies at bit b7 of the second byte.

6.1.2 Channel handle

Local type value: 02_h.

Global key value: 06_h, 01_h, 04_h, 01_h, 02_h, 00_h, 00_h, 00_h.

(Relational, numerical sequences, channel numbers, channel handles, 2-byte CH.)

Length: 2 bytes.

The slot handle is a 2-byte word which shall be used to distinguish between TLD blocks within a SDTI-PF variable block having the same SDTI source and destination header addresses, but representing different TLD block channels. A slot handle shall be set to zero where all TLD blocks associated with the same SDTI source and destination addresses are from a single channel. Where two or more channels of TLD block streams are multiplexed onto SDTI-PF variable blocks, the slot handle values for each TLD block shall be nonzero. The number of multiplexed channels is limited to 65535.

If the channel handle metadata are not present in the SDTI-PF variable block, then the TLD blocks shall be considered to be from one channel.

6.1.3 Sequence position indicator

Local type value: 03_h.

Global key value: 06_h, 01_h, 02_h, 01_h, 01_h, 00_h, 00_h, 00_h.

(Relational, numerical sequences, packet numbers, sequence positions, 1-byte bitmap.)

Length: 1 byte.

The TLD block sequence position indicator is used to identify the position of the following TLD block in a sequence of TLD blocks.

Bits b2 to b0 shall define the position of the immediately following TLD block in a sequence of TLD blocks. These 3 bits identify eight sequence states as follows:

– 0: the TLD block position in a sequence is undefined;

– 1: the TLD block contains a sequence head packet which is any packet which precedes the sequence start packet (e.g., preroll packets);

– 2: the TLD block contains a sequence start packet which is the first packet of a sequence;

– 3: the TLD block contains a midsequence packet which is any packet between the sequence start and sequence end packets;

– 4: the TLD block contains a sequence end packet which is the last packet of a sequence;

– 5: the TLD block contains a sequence tail packet which is any packet which follows the sequence end packet (e.g., postroll packets);

– 6: the TLD block contains both a sequence start packet and sequence end packet signifying a sequence of length 1;

– 7: reserved but undefined.

Bits b7 to b3 are reserved but undefined.

6.2 TLD packet definitions

NOTE – At the time of publication, there is no global dictionary for data packet types so the following packet types are defined only by the local TLD type word. Global data packet labels may be added in a future version of this standard if a global data packet dictionary is published.

6.2.1 MPEG-2 transport streams

Type value: $8x_h$.

Length: variable depending on the presence and type of embedded FEC together with the presence of user, count, and FEC components.

MPEG-2 transport stream (MPEG-2 TS) packets are 188 bytes in length. If forward error correction (FEC) has been added, then the packet length is increased to 204 bytes for DVB emission and 208 bytes for ATSC emission. The MPEG-2 TS FEC may be interleaved across packets and is designed for high levels of data loss through transmission systems which may introduce error bursts. Although in extreme cases SDTI can create occasional errors, the MPEG-2 TS FEC is unsuitable for correcting SDTI errors due to the interleave length, and this FEC will not exist if the packets carried are only 188 bytes in length. Consequently, the SDTI-PF format includes an optional noninterleaved FEC to correct for any errors which may occur through the SDTI link.

Figures 5a, 5b, and 5c illustrate MPEG-TS packets with different FEC capabilities.

It is permissible for SDTI-PF blocks to use all three services — user data, retiming, and RS-FEC — as needed.

Figure 5a – MPEG-2 transport stream packet

Figure 5b – MPEG-2 Transport stream packet with null FEC space

Figure 5c – MPEG-2 Transport stream packet with active 16/20 byte FEC

6.2.2 SDTI-CP transmission packets

Type value: $9x_h$.

Length: variable depending on the presence of user, count, and FEC components.

SDTI-CP transmission packets adopt a packet structure based on 188-byte MPEG-2 transport stream packets and are managed in an identical manner.

6.2.3 Unidirectional internet protocol packets (Uni-IP)

Type value: Ax_h.

Length: variable up to 65535 bytes.

Because Uni-IP packets can have a length in excess of that capable of being handled by the RS-FEC, Uni-IP packets would not normally be able to use FEC. However, if the Uni-IP packets are constrained by the source device to be less than the limit set for correct FEC operation, then it may be used with success on only those packets which are below 254 bytes in length. Users should also be cautioned that some RS decoders rely on a fixed packet length in order to operate correctly in a pipelined operation. So Uni-IP packets with fluctuating packet lengths may cause decoder problems if the RS-FEC is active.

6.3 ATM cells

Type value: Bx_h.

Length: 53 bytes.

ATM cells have a fixed length of 53 bytes to which the user, count, and FEC may be added as needed.

Annex A (informative)
Using the SDTI-PF TLD block counter

The combination of the major and minor count values defines a TLD block count resolution at least as high as 27 MHz and with a repeat period of approximately 30 ms. On reception of the first packet in a sequence, the decoder may output the packet whenever the output buffer is ready and, at the same time, sets an internal counter running at the SDTI word clock rate (27 or 36 MHz) based on the TLD block counter values. This counter operates in the same manner as defined for the major and minor counters. Thereafter, for all remaining packets, the decoder outputs a packet when the decoder counter equals the counter stored in the TLD block counter thus guaranteeing the integrity of the output packet timing. It should be noted that this is not the only method by which timing accuracy can be maintained through application of the major and minor count values.

In the case of 270-Mb/s SDTI, the count value clocks at the same rate as the MPEG-2 TS packet PCR and should result in no timing jitter. In the case of 360-Mb/s SDTI, the count increments by 4 for every 3 increments of the MPEG-2 TS packet PCR; therefore, resulting in a $\pm$ 1 PCR clock jitter. This value is well within the MPEG-2 defined limits. Figure A.1 illustrates the SDTI-PF coding process from the MPEG-2 TS packet stream input to output through the SDTI-PF format.

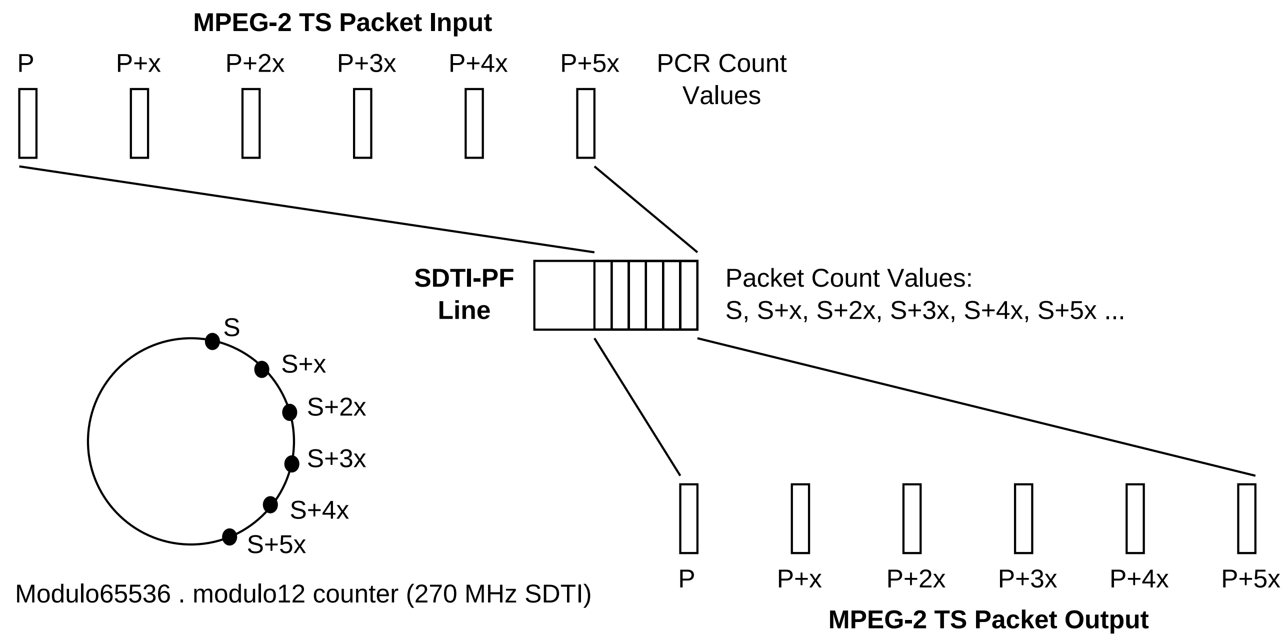

Figure A.1 – SDTI-PF timing reconstruction

Annex B (informative)
System delay considerations

The SDTI line is generally packed with as many packets as the SDTI can manage. This ensures that SDTI lines are not wasted and thus the maximum number of lines is available for the transfer of other data.

The following examples are based on 270-Mb/s SDTI-PF carrying MPEG-2 TS packets:

At a bit rate of 4 Mb/s, approximately 15 lines per frame (525/60) are used for the carriage of MPEG-2 TS packets and the SDTI-PF codec delay is just over 2 ms.

At a bit rate of 50 Mb/s, approximately 185 lines per frame (525/60) are used for the carriage of MPEG-2 TS packets and the SDTI-PF codec delay is thus reduced to around 200 µs.

To a first approximation, the SDTI-PF codec delay is inversely proportional to the bit rate. Thus at low bit rates, delay can only be reduced by reducing the number of TS packets per line and thus occupying proportionately more SDTI lines. If the 4-Mb/s example above occupied only 1 TS packet per line, then the delay would be reduced to approximately 400 µs.

Annex C (informative)
Bibliography

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

Forthcoming SMPTE RP 210, Metadata Dictionary

DVB, Interfaces for CATV/SMATV Headends and Similar Professional Equipment; Asynchronous Serial Interface (ASI)

IETF RFC 1112, Host Extensions for IP Multicasting, 1989

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

ITU-T I.361 (1993), B-ISDN ATM Layer Specification