SMPTE RP 2110-25:2023

SMPTE RECOMMENDED PRACTICE

Professional Media over Managed IP Networks: Measurement Practices

Table of Contents — Page

Approved April 13, 2023

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual. This SMPTE Engineering Document was prepared by Technology Committee TC-32NF Network Facilities and Architectures.

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any clause explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; tables shall be next; then formal languages; then figures; and then any other language forms.

Introduction

This clause is entirely informative and does not form an integral part of this Engineering Document.

In practical systems, there is a need to measure the compliance of streams against the models defined in the relevant documents. This recommended practice defines terms and methods which can be used in test equipment.

At the time of publication, no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

1 Scope

This Recommended Practice specifies recommended nomenclature for measurements on SMPTE 2110 systems, together with their associated formulae for consistency in implementation and reporting of measurements.

This Recommended Practice describes some possible methods for implementing the ST 2110-21 buffer measurements. For these methods their characteristics and differences are described along with ways to report the results so that users understand the differences.

2 Normative References

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

European Broadcasting Union (EBU) EBU Tech 3337: TS-DF Algorithm to measure network jitter on RTP streams

SMPTE ST 2059-1:2021 Generation and Alignment of Interface Signals to the SMPTE Epoch

SMPTE ST 2110-21:2022 Professional Media Over Managed IP Networks: Traffic Shaping and Delivery Timing for Video

3 Terms and Definitions

No terms and definitions are listed in this document.

4 Recommended measurement practices

4.1 General measurement requirements

Measuring equipment is seen as a receiver of streams. Times of events are measured based upon the receiver's clock which shall be synchronized to the common reference clock.

SMPTE ST 2110-10 defines a common reference clock. Measurement device and the devices under test shall use the same common reference clock and associated epoch.

To capture or measure a stream in an accurate fashion, the following prerequisites apply:

- The sender and receiver shall be locked to the common reference clock.
- If a sender is an SDI encapsulator, the input SDI signal shall be locked to the common reference clock and shall be aligned in accordance with SMPTE ST 2059-1.
- The measurement device should capture the stream as close to the sender as possible and be locked to the reference clock.

NOTE 1 When analyzing a stream capture using Wireshark, the timestamp representing the arrival time of the packet at the receiver, can be found on the frame level, labeled as epoch time.

NOTE 2 To measure a stream as accurately as possible, the timestamp of the ethernet frames needs to be of high accuracy.

NOTE 3 To retrieve a usable / sensible packet timestamp, the device capturing the stream needs to be a high accuracy capture device.

4.2 Measurement time window

Test & Measurement equipment can report results in many implementation-specific ways. Measurement devices should report a minimum, a maximum and an average value over a time window of one second. Other time windows may be provided. This can enable users to compare results between different T&M devices.

NOTE The time represented by the RTP Timestamp encapsulated in the RTP packet is limited by the resolution of the 90 kHz RTP Clock, which has a quantization of 11µs. Therefore, the latency measurement will have a quantization of 11µs.

4.3 Mathematical Functions

The following functions are defined for one or more numerical arguments.

round(arg)	returns the integer value that is nearest to arg, with halfway cases rounded away from zero.
int(arg)	returns the largest integer not greater than arg.
trunc(arg):	returns the nearest integer value of arg, rounded towards zero.
time()	retrieves the arrival time of a packet expressed in seconds. The time is represented as the number of seconds since the SMPTE Epoch.

4.4 Variables

TPA	time of arrival of a packet relative to the SMPTE Epoch.
TPA_j	time of arrival of packet j at the receiver relative to the SMPTE Epoch (j = 0 for the first packet of a frame). This value is measured by the Test Receiver.
TPR_j	time relative to the SMPTE Epoch when a packet j will be removed from the Virtual Receiver Buffer. (Time-Packet-Read-j). (As defined in SMPTE ST 2110-21)
N	index of the frame since Epoch defined in SMPTE ST 2059-1. (As defined in SMPTE ST 2110-21)
$N_{PACKETS}$	number of packets per frame of video (depends on mapping details). (As defined in SMPTE ST 2110-21)
T_{FRAME}	time period between consecutive frames of video at the prevailing frame rate. (As defined in SMPTE ST 2110-21)
$RTP_{Timestamp}$	attribute provided in the RTP header.
$RTP_{BitDepth}$	number of bits provided in the RTP header to represent the $RTP_{Timestamp}$ is 32. $RTP_{Bitdepth} = 32$
$RTP_{Clockrate}$	rate at which the counter sampled to determine the timestamps included in RTP packets is advanced by the Media Clock specified for the media type.

$RTP_{Wraparound}$ number of wraps around of the RTP timestamp value for a given *time*.

The timestamp is a 32-bit unsigned integer that increases at a media-dependent rate and wraps around to zero when the maximum value is exceeded. With typical video codecs, a clock rate of 90kHz is used, corresponding to a wrap around of approximately 13 hours. Formula (1) illustrates how to calculate the number of wraps around of the RTP timestamp value for a given *time*.

$RTP_{Timestamp\_encoded}$ time relative to the SMPTE Epoch encoded in the RTP timestamp of a packet. Formula (2) illustrates how to calculate this value.

T_{CF} rounded arrival time of current frame, equal to N × T_{FRAME}, where N = round (TPA_0 / T_{FRAME}).

4.5 Formulae

$$\boldsymbol{RTP_{Wraparound}} = \boldsymbol{int}\left(\boldsymbol{time}() * \frac{\boldsymbol{RTP_{Clockrate}}}{\boldsymbol{2^{RTP_{Bitdepth}}}}\right) \quad (1)$$

$$\boldsymbol{RTP_{Timestamp\_encoded}} = \frac{\boldsymbol{RTP_{Wraparound}} * \boldsymbol{2^{RTP_{Bitdepth}}} + \boldsymbol{RTP_{Timestamp}}}{\boldsymbol{RTP_{Clockrate}}} \quad (2)$$

NOTE When RTP packets are encoded, they contain a timestamp that represents the moment when the packet was created. If an RTP packet is encoded before the wraparound time, but is received after the wraparound time, then the timestamp can be misinterpreted. The wraparound time is the point at which the timestamp value rolls over and starts counting from zero again.

4.6 Illustration of timing relationship

Figure 1 is a graphical aid to understand the measurements defined in this document.

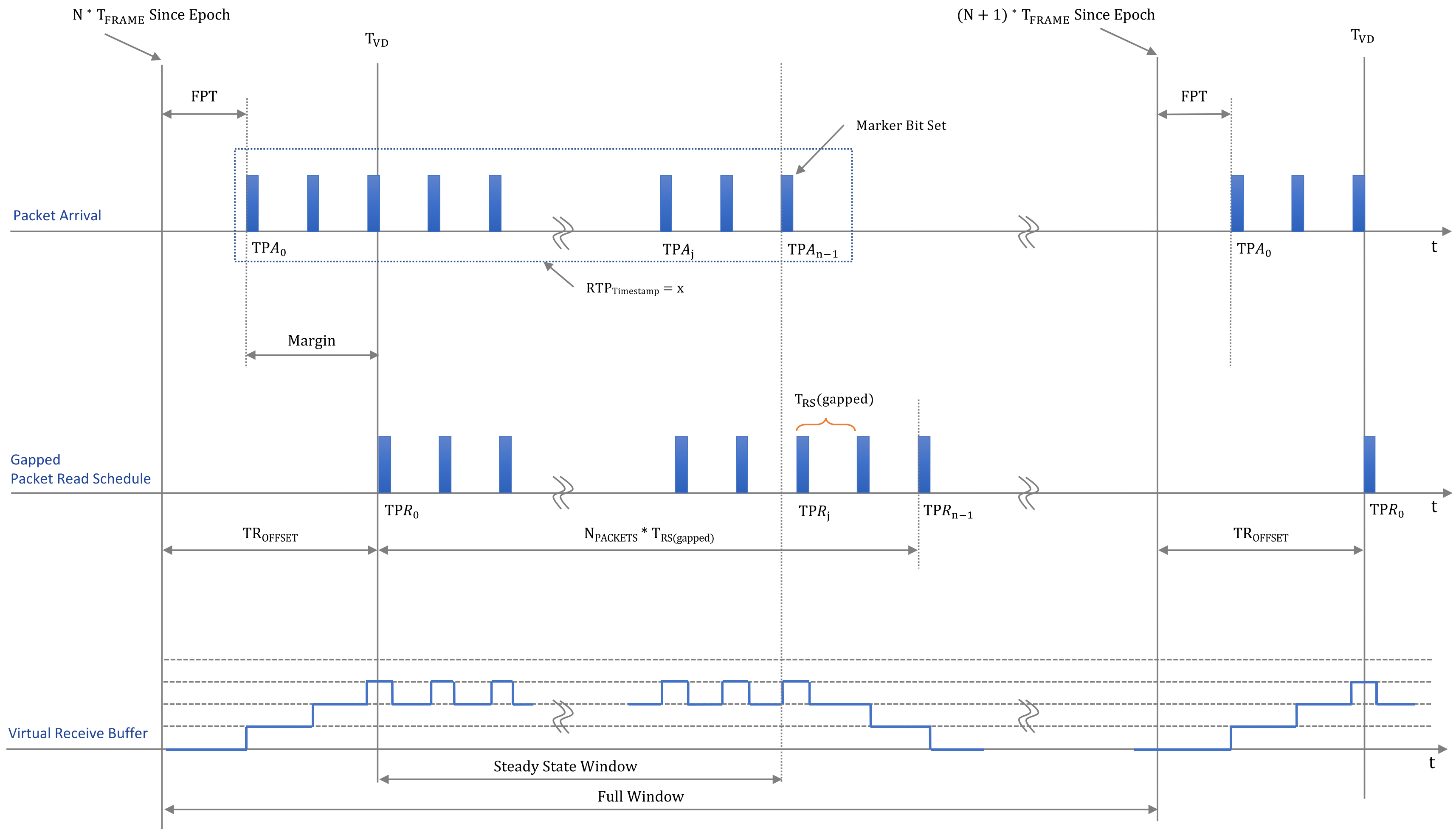


Figure 1 — Diagram illustrating different measurements

4.7 Measurements for SMPTE ST 2110-10

No specific measurements for SMPTE ST 2110-10 are proposed.

4.8 Measurements for SMPTE ST 2110-20

4.8.1 Measuring the Network Compatibility Model (C_{INST})

Measurement of the Network Compatibility Model (C_{INST}) is described in Clause 4.9.1

4.8.2 Measuring the Virtual Receive Buffer Model (VRX)

Measurement of the Virtual Receive Buffer Model (VRX) is described in Clause 4.9.2

4.8.3 Measurement – First Packet Time (FPT)

FPT is the equivalent to video timing for analog or SDI. Since the first packet of a frame generally comes from the start of the active region, First Packet Time is delayed by the vertical blanking period compared to traditional V = 0 timing.

FPT measures the arrival time of the first packet of a frame, relative to the Alignment Time (as defined in SMPTE ST 2059-1) of the corresponding frame in an SDI signal. This measurement is taken using the receiver's time reference for the actual arrival time. The range of the measurement will be plus or minus one-half of a frame period, relative to the current frame time T_{CF} and shall be expressed in microseconds, as specified in Formula (3).

$$\boldsymbol{FPT = TPA_0 - T_{CF}} \tag{3}$$

4.8.4 Measurement – RTP_{OFFSET}

This RTP_{OFFSET} is the time difference between the time encoded in the RTP timestamp and the epoch-aligned start time of the frame period, as specified in Formula (4).

$$\boldsymbol{RTP_{OFFSET} = RTP_{Timestamp\_encoded} - T_{CF}} \tag{4}$$

4.8.5 Measurement – Video Latency (VL)

The video latency measure is the difference between the arrival time of the first packet of the frame and the time encoded in the RTP timestamp of that packet ($RTP_{Timestamp\_encoded}$) which is encapsulated in the RTP header of the respective IP packet.

By comparing the latency of the video streams, a user can determine how much the streams would need to be shifted to realign them according to the timestamps, as specified in Formula (5).

$$\boldsymbol{Video\ Latency\ = TPA_0 - RTP_{Timestamp\ \_encoded}} \tag{5}$$

4.8.6 Measurement – Margin (M)

Margin is the time difference between the arrival time of the first packet and the time of the first VRX Offset buffer read as set by TR_{OFFSET} and shall be expressed in microseconds. In effect, this is the time for the buffer model to pre-fill before it starts reading, as specified in Formula (6).

$$\boldsymbol{Margin = TR_{OFFSET} - FPT} \tag{6}$$

Measurement Result Range: The Margin Measurement shall be displayed in units of time.

NOTE Since both TR_{OFFSET} and FPT are limited, the range of the Margin measurement is inherently limited to a time of -0.5 to +1.5 frame.

4.8.7 Measurement – GAP

GAP is the time difference between the arrival of the first packet of the current field or frame (*CF*) and the last packet of the previous field or frame (*CF – 1*) and shall be expressed in microseconds, as specified in Formula (7).

$$\boldsymbol{GAP} = \left(\boldsymbol{TPA}_{\mathbf{0}\ (\boldsymbol{CF})} - \boldsymbol{TPA}_{\boldsymbol{N_{PACKETS}}-\mathbf{1}\ (\boldsymbol{CF}-\mathbf{1})}\right) \qquad (7)$$

Measurement Result Range: The GAP measurement will be represented as time, typically between 0 to 4 percent of a frame time. A negative number indicates an out-of-order packet.

4.9 Measurements for SMPTE ST 2110-21

4.9.1 Measuring the Network Compatibility Model (C_{INST})

SMPTE ST 2110-21 uses a leaky bucket with a maximum level (C_{MAX}) to model network switch buffers and to prevent packet loss in the network. Therefore, it is recommended to measure the instantaneous number of packets in this virtual buffer and keep track of the peak number of packets (C_{PEAK}), to check that it does not exceed C_{MAX}. While displaying C_{PEAK} it is advised to report the applicable C_{MAX} value, as this value might be different for different type of streams.

NOTE An example algorithm to calculate C_{PEAK} is described in B.2.

4.9.2 Measuring the Virtual Receive Buffer Model (VRX)

The virtual receive buffer model (VRX) describes a maximum buffer size (VRX_{FULL}) that cannot be exceeded, nor be underrun at any time. The level of the VRX buffer is influenced by the arrival of packets and the draining of the packets at the Packet Read Schedule (PRS) as illustrated in Figure 1. While displaying VRX_{PEAK} it is advised to report the applicable VRX_{FULL} value together with $N_{PACKETS}$, as this value might be different for different type of streams.

The following parameters are suggested for measuring:

- VRX_{PEAK}
  - VRX_{PEAK} keeps track of the maximum value of VRX_{INST}. It is recommended to measure the instantaneous value (VRX_{INST}) of the virtual buffer to keep track of the peak value VRX_{PEAK}, as it is not allowed to exceed VRX_{FULL}.
- $VRX_{OVERFLOW}$
  - A $VRX_{OVERFLOW}$ has occurred whenever the value of VRX_{PEAK} exceeds VRX_{FULL}.
- $VRX_{UNDERFLOW}$
  - A $VRX_{UNDERFLOW}$ has occurred whenever, at reading time, there are no packets available in the buffer and the buffer is empty.
- $VRX_{PACKET\ MISSING}$
  - $VRX_{PACKET\ MISSING}$ has occurred whenever the expected packet is not available at the expected read time.
- $VRX_{MIN\text{-}SS}$
  - $VRX_{MIN\text{-}SS}$ is the minimum value during the Steady State window.
  - The Steady State window, as illustrated in Figure 1, is a time window between TPR_0, the scheduled read time for the first packet of the frame, and TPA_{n-1}, the arrival time of the last packet of the frame.
- $VRX_{MIN\text{-}GAP}$
  - $VRX_{MIN\text{-}GAP}$ is the minimum value during the time interval between the arrival of the last packet of a given frame and the arrival of the first packet of the subsequent frame.
  - Annex A describes why this value can be of interest.

- VRX_{AVG}
  - VRX_{AVG} tracks the mean fullness of the virtual receive buffer. The sample of fullness of the buffer is taken before reading each packet at the time indicated by the read schedule. The default averaging and reporting window of this measurement shall be 1 second.
- $VRX_{AVG\text{-}SS}$
  - $VRX_{AVG\text{-}SS}$ tracks the mean fullness of the virtual receive buffer during the Steady State window.
  - The Steady State window, as illustrated in Figure 1, is a time window between TPR_0, the scheduled read time for the first packet of the frame, and TPA_{n-1}, the arrival time of the last packet of the frame.

The VRX calculation shall use the TR_{OFFSET} value as signaled in the SDP (or the default if not signaled).

NOTE 1 As described in Annex A, multiple approaches do exist, and they result in slightly different values.

NOTE 2 An example algorithm to calculate VRX_{PEAK} is described in B.3.

4.10 Measurements for SMPTE ST 2110-22

4.10.1 Measuring the Network Compatibility Model (C_{INST})

Measurement of the Network Compatibility Model (C_{INST}) is described in Clause 4.9.1.

4.10.2 Measuring the Virtual Receive Buffer Model (VRX)

The Virtual Receive Buffer model does not apply to SMPTE ST 2110-22 streams.

4.11 Measurements for SMPTE ST 2110-30

4.11.1 Measuring the Audio Delay Variance (ADV)

Jitter introduced by an audio sender could lead to increased latency between sender and receiver. The minimum delay of the audio path is dictated by the packet time. ST 2110-30 describes two packet times: Level A specifies 1 ms packet time; Levels B and C specify a 125 µs packet time.

AES67 defines a jitter output level for senders:

- AES67 requires jitter to be lower than 17 x packet time (17 PT) or 17 ms, whichever is smaller.
- AES67 proposes jitter to be lower than 1 x packet time (1 PT).

AES67 defines a tolerance level for receiver:

- AES67 requires the receiver to be able to accept at least 3 x packet time (3 PT) of input jitter.
- AES67 proposes the receiver to be able to accept at least 20 x packet time (20 PT) of input jitter.

The measurement methodology for Audio Delay Variance shall be the Time-Stamped Delay Factor measurement (TS-DF) as defined in EBU Tech 3337. Audio Delay Variance can be regarded as equivalent to “jitter” in the AES67 specification. The measurement is based on the Relative Transit Time defined in RFC 3550 (RTP: A Transport Protocol for Real-Time Applications). This is defined as the difference between a packet's RTP timestamp (held in the RTP header) and the receiver's clock at the time of arrival, measured in the same units.

The TS-DF measurement period is 1 second. In this algorithm, the first packet at the start of the measurement period is considered to have no jitter and is used as a reference packet.

For each subsequent packet that arrives within the measurement period, the Relative Transit Time between this packet and the reference packet is calculated as specified in Formula (8):

$$\boldsymbol{D(i,0) = \big(R(i) - R(0)\big) - \big(S(i) - S(0)\big)} \tag{8}$$

where R is the packet arrival time of the packet and S is the RTP timestamp, i is the measured packet and 0 is the reference packet. The maximum and minimum values are extracted, and the Time-stamped Delay Factor is calculated as specified in Formula (9):

$$\boldsymbol{Timestamped\ Delay\ Factor = D(Max) - D(Min)} \tag{9}$$

The TS-DF function will yield the size of de-jitter receive buffer needed for every second.

4.11.2 Measurement – Packet Interval Time (PIT)

The Packet Interval Time (PIT) measurement is the time difference between the arrival of consecutive packets. Analysis of the PIT distribution of packets will provide an indication of the jitter of the packet stream.

For a gapped flow, the results may optionally be displayed including or excluding the interval between two consecutive frames or fields.

4.11.3 Measurement – Audio Latency (AL)

The audio latency measurement is the time difference between the arrival time of the packet (TP*A*) and the time encoded in the RTP timestamp ($RTP_{Timestamp\_encoded}$) which is encapsulated in the RTP header of the respective IP packet, as specified in Formula (10).

$$\boldsymbol{Audio\ Latency\ = \mathrm{TP}A - RTP_{Timestamp\_encoded}} \tag{10}$$

The time resolution represented by the RTP Timestamp is limited by the resolution of the source RTP Clock.

NOTE 1 Audio samples are packed together to increase overall network efficiency (e.g., 1 ms or 125 µs profile). As described in RFC 3550, the RTP Timestamp is the sample time of the first sample. Since the packet needs to be constructed with the full complement of audio samples related to the packet time, in normal operation the result of this measurement is at minimum one packet time (1 PT).

NOTE 2 The result of the measurement is affected by the quantization of the RTP timestamp, which is up to 20.83 µs. For a 1 ms audio packet time, this is a 2% increment. For a 125 µs audio packet time this is a 16.64% increment.

4.11.4 Measurement – Audio Video Differential Latency (AVDL)

The Audio Video Differential Latency measurement is the difference between Video Latency (VL) and the Audio Latency (AL). By comparing the latency of the audio and video streams, a user can determine how much the streams would need to be shifted to realign them according to the timestamps, as specified in Formula (11).

$$\boldsymbol{Audio\ Video\ Differential\ Latency\ = AL - VL} \tag{11}$$

The time resolution represented by the RTP Timestamp is limited by the resolution of the source RTP Clock. This is the amount of correction needed to make the audio and video line up according to the timestamps.

NOTE 1 The results of the measurement will be negative when the video latency is higher than the audio latency.

NOTE 2 This recommendation follows a common practice already widely implemented in the industry despite the existence of BT.1359-1-1998, which suggests flipping the polarity of the formula.

4.12 Measurements for SMPTE ST 2110-40

4.12.1 Measurement - First Packet Time (FTP)

The First Packet Time (FPT) as defined in Clause 4.8.3 applies also to SMPTE ST 2110-40.

4.12.2 Measurement - RTP_{OFFSET}

RTP offset as defined in Clause 4.8.4 applies also to SMPTE ST 2110-40.

4.12.3 Measurement – ANC Latency (ANCL)

The ANC Latency measure (ANCL) is the time difference between the arrival time of the first packet of the frame and the time encoded in the RTP timestamp of that packet, averaged over 1 second.

By comparing the latency of the video and ANC streams, a user can determine how much the streams would need to be shifted to realign them according to the timestamps, as specified in Formula (12).

$$\boldsymbol{ANC\ Latency\ = TPA_0 - RTP_{Timestamp\_encoded}} \qquad (12)$$

NOTE The time represented by the RTP Timestamp encapsulated in the RTP packet is limited by the resolution of the 90 kHz RTP Clock, which has a quantization of 11 µs. Therefore the latency measurement will have a quantization of 11 µs.

4.12.4 Measurement – ANC Video Differential Latency (ANC VDL)

The ANC Video Differential Latency measurement is the difference between Video Latency (VL) and the ANC Latency (ANCL). By comparing the latency of the ANC and video streams, a user can determine how much the streams would need to be shifted to realign them according to the timestamps, as specified in Formula (13).

$$\boldsymbol{ANC\ Video\ Differential\ Latency\ = ANCL - VL} \qquad (13)$$

The time resolution represented by the RTP Timestamp is limited by the resolution of the source RTP Clock.

This is the amount of correction needed to make the ANC and video line up according to the timestamps.

NOTE The results of the measurement will be negative when the Video Latency is higher than the ANC Latency.

4.12.5 Measurement – Relative RTP_{OFFSET} (RRTPO)

This measurement is the time difference between the timestamps in the video packets and those of the associated ancillary data packets. The measurement is an indication of synchronization at the sender without considering any network latencies. The $RRTP_{OFFSET}$ is a representation of the synchronization of metadata and video at the source, as specified in Formula (14).

$$\boldsymbol{RRTP_{OFFSET} = RTP_{Timestamp\_encoded}(video) - RTP_{Timestamp\_encoded}(anc)} \qquad (14)$$

Ideally, the $RRTP_{OFFSET}$ will be close to 0 as video and metadata will likely have been time stamped at the same moment in the senders. A large RTP offset is an indication something is wrong and is most likely going to lead to bad transport timing compensation.

The measurement can be performed on every frame (instantaneous value) with a maximum and minimum. The measure is expected to be static. A dynamically changing value is an indication the metadata frame rate is not the same as the video being compared to or video and metadata media rates are not the same.

NOTE 1 One can use the timestamps of any pair of frames and then apply a modulo operation to constrain the range to +/- 0.5 a frame.

NOTE 2 Another method is to choose the closest two timestamps and use a simple subtraction. This method does not work in some corner cases.

4.12.6 Measurement – Metadata Margin (MM)

This measure is an indicator of how much time margin is available for metadata reinsertion in SDI given the relative arrival time of video and metadata as well as the location of metadata as indicated in the payload header.

The metadata margin measurement is the time available from the arrival of metadata packets up to $T_{TRANSMIT}$ given by the transmission model in ST 2110-40. $T_{TRANSMIT}$ is the latest time at which one can transmit those packets and have them available for reinsertion into the SDI signal. Figure 2 illustrates the Metadata Margin measurement if the RTP timestamp is coincident with the time of the SDI alignment point:

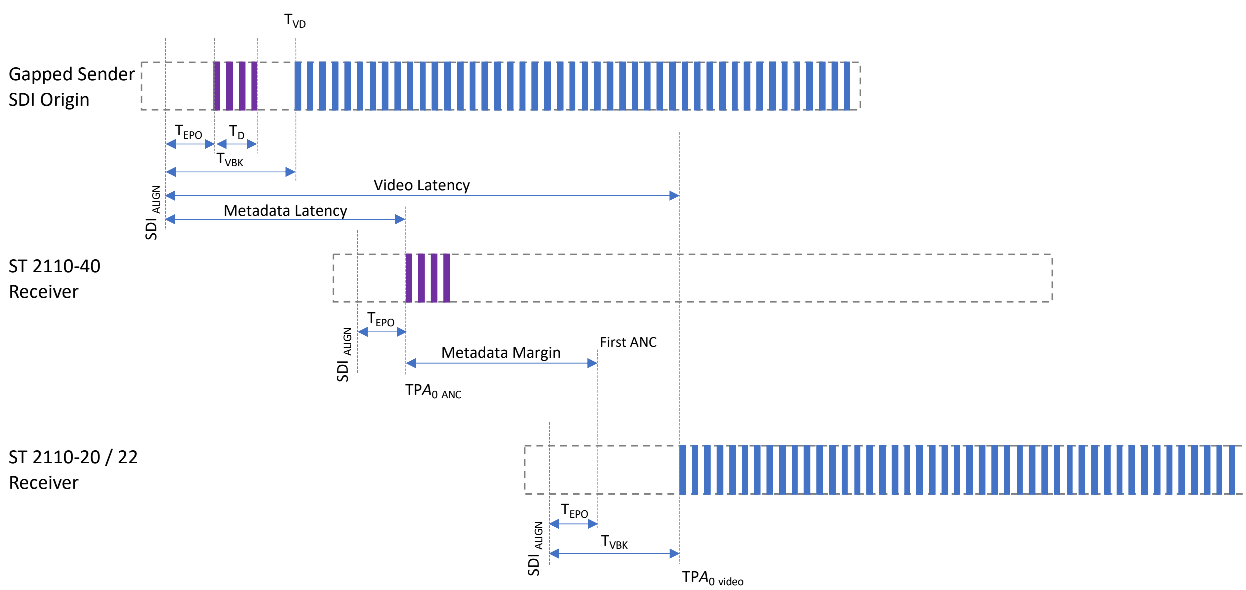


Figure 2 — Metadata Margin measurement illustration

The measurement is defined as specified in Formula (15):

$$\boldsymbol{Metadata\ Margin} = (\boldsymbol{Video\ Latency} - \boldsymbol{ANC\ Latency}) - \boldsymbol{T_{VBK}} + \boldsymbol{T_{EPO}} \tag{15}$$

T_{VBK} is format dependent and defined by the underlying SDI raster. It defines the time between the SDI alignment point and the start of active video. Note that this is not necessarily equal to TR_{OFFSET} as it is relative to the SDI alignment point which is not necessarily aligned with reference. T_{EPO} is defined as the Earliest Packet Time Offset in SMPTE ST 2110-40. A negative metadata margin means the video needs to be delayed by at least that much for ANC to be reinserted at the right location.

Annex A (Informative) Methods to measure VRX

A.1 Multiple methods

This annex describes two different methods to implement the ST 2110-21 VRX test, other methods can be used:

- Event History method
- Residence Time method

A.2 Event History method

A.2.1 General

The Event History method is analogous to the operation of a ‘physical or real’ buffer and gives a reading of the buffer level on a packet-by-packet basis. It looks at the arrival time of a packet and increments the buffer level and looks at the read time to decrement the buffer level. In the presence of a missing or out-of-order packet being written to the buffer, the ‘j^{th}’ packet is not necessarily read out of the buffer at the ‘j^{th}’ time.

Some Event History method implementations stop reading at the Marker bit (M), others read the entire buffer. There are two possible measurement windows that can be applied to the Event History method:

- ‘Full’ Measurement window
  - This measures from the arrival of the first packet (TPA_0) up to the reading of the last packet out of the receiving buffer ($TPR_{Npacket-1}$).
  - This includes the buffer ramp-up and ramp-down as the buffer fills and empties.
  - This will give a minimum reading of the buffer once the ramp down has been completed. In a Gapped schedule this is the ‘gapped’ minimum value.
- ‘Steady State’ Measurement window
  - This measures from TPR_0 to TPA_{n-1}, this is from time TR_{OFFSET} to the ‘M’ bit, which is the time period from the first Read to the last Write.
  - This will give a minimum reading within the Steady State window and will ignore the gap minimum of a gapped schedule.
  - It is also possible to post process or window the Full Measurement window data set to generate the Steady State Measurement data set.

A.3 Residence Time method

A.3.1 Introduction

The residence time method calculates the difference between the received timestamp of each packet arrival and the scheduled read time for that packet. In this methodology the j^{th} packet is always read at the j^{th} time even in the presence of missing or out-of-order packets. Dividing by the read interval gives the buffer fullness.

A.4 Discussion of the methods

A.4.1 Advantages and Disadvantages

Both methods have their limitations and advantages

The Event History method treats all packets in the same way. This results in sensitivity to packet loss and a single late packet will not show an underflow. It will also show a ramp up and ramp down at the start and end of a gapped schedule

The Residence Time method might report an overflow for a few early packets, especially if they occur early in the Video Frame in a gapped schedule.

Both methods evaluate the buffer fullness at each packet arrival and read, according to the VRX buffer model.

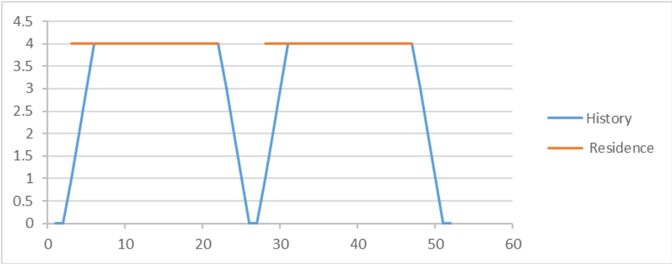
The methods differ in the minimum value that is reported. The Full or non-windowed event history method will report a 0 for a gapped schedule, whereas the instantaneous method provides an 'Intra Frame' minimum value.

A windowed or steady state event history method will produce the same min and max values as the Residence method with the advantage that it will not report an overflow for a few early packets but is susceptible to missing or out-of-order packets.

A.4.2 Representative Examples

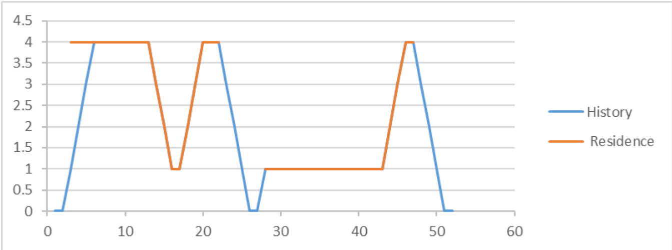
A.4.2.1 Packet arrival according to Gapped Packet Read Schedule:

	'Full-Event History'	'Residence'
Max	4	4
Min	0	4



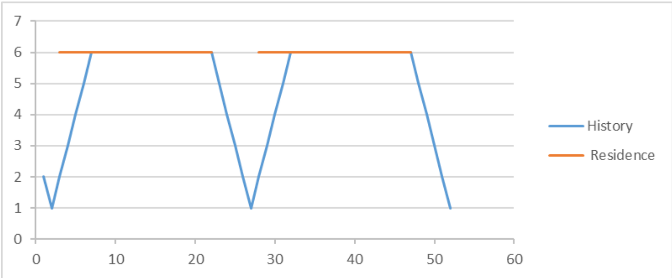
A.4.2.2 Late Packets:

	'Full-Event History'	'Residence'
Max	4	4
Min	0	1



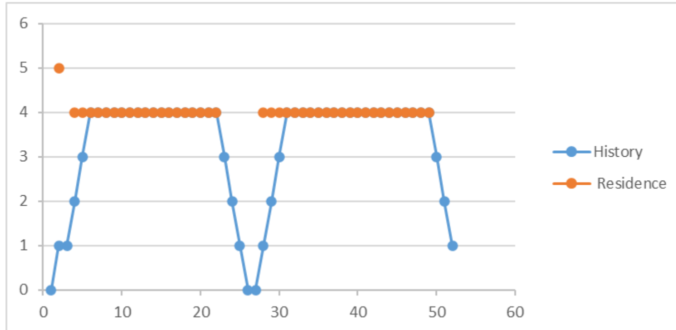
A.4.2.3 Large TR_{OFFSET}:

	'Full-Event History'	'Residence'
Max	4	4
Min	1	4



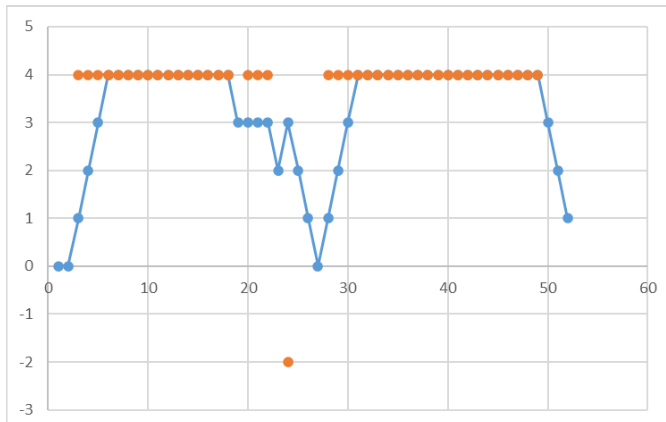
A.4.2.4 Early Packet:

	'Full-Event History'	'Residence'
Max	4	5
Min	0	4



A.4.2.5 Very Late Packet:

	'Full-Event History'	'Residence'
Max	4	4
Min	0	-2



Annex B (Informative) Example code

B.1 Introduction to the example code

The following example code in B.2 and B.3 for illustration purposes only, written in Python, demonstrates how C_{PEAK} and VRX_{PEAK} can be calculated. Both function definitions are provided for clarification of the measurements. Other methods are possible.

The parameters provided to the functions are:

capture: Provides a network capture of the video stream. The information which will be used is the timestamp of the ethernet frames and the RTP header information providing the marker bit status. The example code uses the PyShark library.

`trs:` T_{RS} as Time-Read-Spacing.

`tframe:` T_{FRAME} is the time period between consecutive frames of video at the prevailing frame rate.

`npackets:` $N_{PACKETS}$ is the number of packets per frame of video (depends on mapping details).

`B:` β is a scaling factor of 1.1

B.2 Example code to calculate C_{PEAK}

```
def cfull_analysis(capture, tframe, npackets, B):
    cpeak = 0     # Initialize the C peak value
    cinst = 0     # Initialize the C inst value
    tcurr = None  # Tcurr as Time of the current packet
    tprev = None  # Tprev as Time of the previous packet

    # Calculate Tdrain as defined in SMPTE ST 2110-21
    tdrain = tframe / npackets / Decimal(B)

    for pkt in capture:
        tcurr = Decimal(pkt.sniff_timestamp)
        if tprev:
            cinst += 1
            cinst -= (tcurr - tprev)/tdrain
            if cinst < 0:
                cinst = 0
        tprev = tcurr
        if cinst > cpeak:
            cpeak = math.ceil(cinst)

    return cpeak
```


B.3 Example code to calculate VRX_{PEAK}

```
def vrx(capture, trs, tframe, npackets, troffset):
    evt_scale = 0.0000001  # Event Time Scale should be max half of Trs.
    evt_timer = 0.0.       # Initialize the Event Time
    rd_schedule = None     # Packet Read Schedule Time
    frame_idx = 0          # frame index
    prev = None            # previous packet

    sequence_number_current  = None
    sequence_number_expected = 0

    vrx_buff = []
    vrx_peak = 0
    vrx_packet_missing = 0

    n_frames = None
    j = 0
    pkt_time = None


    for pkt in capture:
        # Capture the timestamp of the ethernet frame
        pkt_time = Decimal(pkt.sniff_timestamp)
        sequence_number_current = int(pkt.rtp.seq)

        if prev and prev.rtp.marker == '1':  # new frame
            #continue draining
            #clear buffer
            vrx_buff = []

            j = 0
            sequence_number_expected = sequence_number_current

            n_frames = int(pkt_time/tframe)
            rd_schedule = Decimal(n_frames * tframe) + troffset + Decimal(j*trs)

            # If it is the first video frame of the PCAP file; intialize the event timer
            if frame_idx == 0:  # first frame
                frame_idx = 1
                evt_timer = rd_schedule

        # Calculate the VRX buffer, starting only from the first full frame of video
        if frame_idx>0 :

            vrx_buff.append(sequence_number_current)
```



```
        while evt_timer <= pkt_time:
            #evt_timer = Decimal(evt_timer) + Decimal(evt_scale)

            if evt_timer >= rd_schedule:
                try:
                    vrx_buff.remove(sequence_number_expected)
                except:
                    print ("Packet missing")
                    if len(vrx_buff) == 0:
                        print ("VRX Underrun")
                    pass

                if j < npackets:
                    j += 1
                    sequence_number_expected=(sequence_number_expected + 1) % PKT_SEQUENCE_BIT_DEPTH
                    rd_schedule = Decimal(n_frames * tframe) + troffset + Decimal(j * trs)

            evt_timer = Decimal(evt_timer) + Decimal(evt_scale)

        if len(vrx_buff) > vrx_peak:
            vrx_peak = len(vrx_buff)

    prev = pkt
return vrx_peak
```

Bibliography (Informative)

Audio Engineering Society (AES) AES67:2018, AES standard for audio applications of networks - High-performance streaming audio-over-IP interoperability

Internet Engineering Task Force (IETF) RFC 3550: RTP: A Transport Protocol for Real-Time Applications

Rec. ITU-R BT.1359 RELATIVE TIMING OF SOUND AND VISION FOR BROADCASTING

SMPTE ST 2110-10:2022 Professional Media Over Managed IP Networks: System Timing and Definitions

SMPTE ST 2110-20:2022 Professional Media Over Managed IP Networks: Uncompressed Active Video

SMPTE ST 2110-22:2022 Professional Media Over Managed IP Networks: Constant Bit-Rate Compressed Video

SMPTE ST 2110-30:2017 Professional Media Over Managed IP Networks: PCM Digital Audio

SMPTE ST 2110-40:2018 Professional Media Over Managed IP Networks: SMPTE ST 291-1 Ancillary Data

