ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation

ITU-T G.8260 (11/2022)

SERIES G: Transmission systems and media, digital systems and networks

Packet over Transport aspects – Synchronization, quality and availability targets

Definitions and terminology for synchronization in packet networks

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
Ethernet over Transport aspects	G.8000–G.8099
MPLS over Transport aspects	G.8100–G.8199
Synchronization, quality and availability targets	**G.8200–G.8299**
Mobile network transport aspects	G.8300–G.8399
Service Management	G.8600–G.8699
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.8260

Definitions and terminology for synchronization in packet networks

Summary

Recommendation ITU-T G.8260 provides the definitions, terminology and abbreviations used in ITU-T Recommendations on timing and synchronization in packet networks.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T G.8260	2010-08-12	15	11.1002/1000/10907
2.0	ITU-T G.8260	2012-02-13	15	11.1002/1000/11521
2.1	ITU-T G.8260 (2012) Amd. 1	2013-08-29	15	11.1002/1000/12016
2.2	ITU-T G.8260 (2012) Amd. 2	2014-05-14	15	11.1002/1000/12189
3.0	ITU-T G.8260	2015-08-13	15	11.1002/1000/12545
3.1	ITU-T G.8260 (2015) Amd. 1	2016-04-13	15	11.1002/1000/12808
3.2	ITU-T G.8260 (2015) Amd. 2	2018-11-29	15	11.1002/1000/13765
4.0	ITU-T G.8260	2020-03-15	15	11.1002/1000/14206
5.0	ITU-T G.8260	2022-11-13	15	11.1002/1000/15127

Keywords

Frequency, phase and time, packet delay variation, synchronization definitions.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T G.8260

Definitions and terminology for synchronization in packet networks

1 Scope

This Recommendation provides the definitions, terminology and abbreviations used in Recommendations on frequency, phase and time synchronization in packet networks. It includes mathematical definitions for various synchronization stability and quality metrics for packet networks, and also provides background information on the nature of packet timing systems and the impairments created by packet networks.

Ethernet physical layer methods for synchronization are based on traditional time division multiplexing (TDM) physical layer synchronization and therefore most of the definitions related to these methods are covered by [ITU-T G.810]. Additional definitions are included in this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T G.810] Recommendation ITU-T G.810 (1996), *Definitions and terminology for synchronization networks*.

[ITU-T G.811] Recommendation ITU-T G.811 (1997), *Timing characteristics of primary reference clocks*.

[ITU-T G.8261] Recommendation ITU-T G.8261/Y.1361 (2019), *Timing and synchronization aspects in packet networks*.

[ITU-T Y.1413] Recommendation ITU-T Y.1413 (2004), *TDM-MPLS network interworking – User plane interworking*.

[IEEE 1588] IEEE Standard 1588-2019, *IEEE standard for a precision clock synchronization protocol for networked measurement and control systems*.

3 Definitions

3.1 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.1.1 adaptive clock recovery: Clock recovery technique that does not require the support of a network-wide synchronization signal to regenerate the timing. In this case, the timing recovery process is based on the (inter-)arrival time of the packets, e.g., timestamps or circuit emulation service (CES) packets. The information carried by the packets could be used to support this operation. Two-way or one-way protocols can be used.

3.1.2 arbitrary reference time clock (ARTC): A reference time generator that provides a reference time signal, or simply a reference phase signal, whose frequency has the accuracy of a primary reference clock (PRC) as specified in [ITU-T G.811], while the epoch does not necessarily have a relationship with an internationally recognized time standard.

3.1.3 **coherent time and frequency**: The condition where the timing signal-carrying frequency and the timing signal-carrying time-of-day or phase are traceable back to the same primary source.

3.1.4 **floor delay**: The notion of "floor delay" is derived from the notion of minimum possible transit delay of packets over a network. It may be useful to distinguish the notions of "*absolute floor delay*" and "*observed floor delay*":

- **absolute floor delay**: Absolute minimum possible transit delay of packets of a given size over a network. This may generally be described as the transit delay experienced by a packet that has experienced the minimum possible delay through each network element along a specified path. Depending on loading and other considerations, it is possible that in any given finite window of observation interval a packet with delay equal to this absolute minimum may not be observed. Full knowledge of the packet network, network elements, and routing path must be known in order to perform a theoretical analysis of the minimum transit delay.
- **observed floor delay (OFD)**: Minimum transit delay of packets of a given size over a network observed over a given observation interval [for instance, during a packet delay variation (PDV) measurement].

  NOTE – As mentioned above, the observed floor delay during a PDV measurement may differ from the absolute floor delay.

3.1.5 **floor delay step**: The difference between the observed floor delays of two consecutive, non-overlapping observation intervals, see Figure 1:

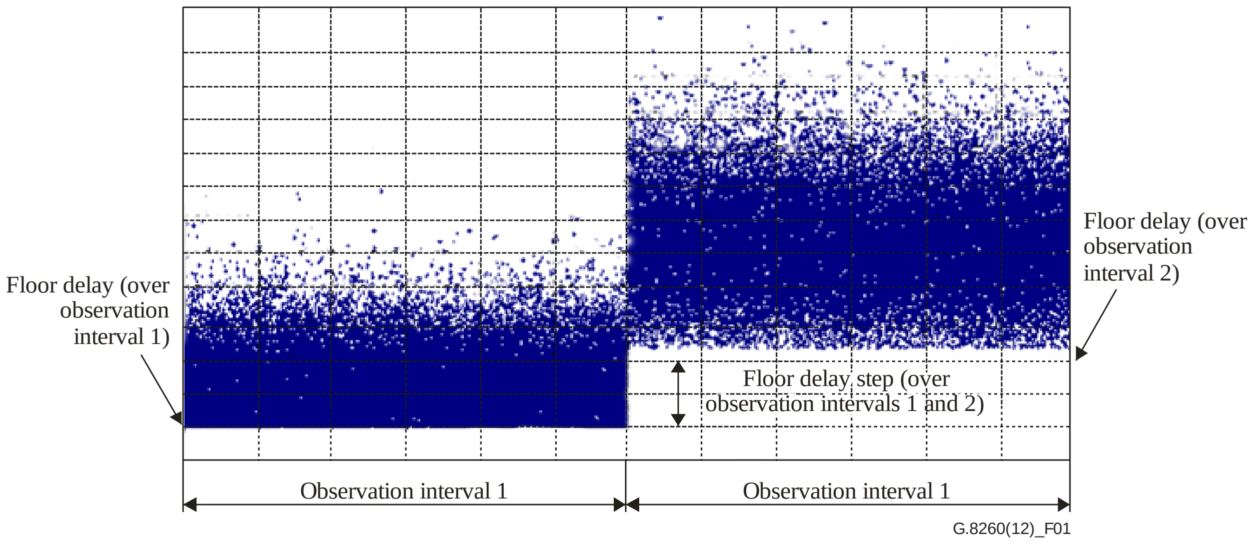

Figure 1 – Illustration of observed floor delays and floor delay step

3.1.6 **packet-based method**: Timing distribution method (for frequency or time or phase) where the timing information is associated with packets.

- The frequency can be recovered using two-way or one-way protocols.
- Time and phase information is recovered with a two-way protocol in order to compensate for the transfer delay from packet master clock to packet slave clock.

3.1.7 **packet-based method with full timing support to the protocol level from the network**: Packet-based method (frequency or time and/or phase synchronization) requiring that all the network nodes on the path of the synchronization flow implement one of the two following types of timing support:

- termination and regeneration of the timing [e.g., network time protocol (NTP) stratum clocks, precision time protocol (PTP) boundary clock];
- a mechanism to correct for the delay introduced by the network node or the connected links (e.g., PTP transparent clock).

3.1.8 packet-based method with partial timing support to the protocol level from the network: Packet-based method (frequency or time and/or phase synchronization) where not all of the network nodes on the path of the synchronization flow implement timing support.

3.1.9 packet-based method with physical frequency support from the network: Packet-based method for time and phase synchronization using frequency support from a traceable network reference clock carried by a physical layer timing trail.

NOTE – For instance, it can correspond to telecom boundary clocks syntonized by a frequency reference carried at the physical layer. This type of support is expected to provide "phase and/or time holdover" capacities, enabling phase and/or time local reference to be maintained during periods of failure of the phase and/or time distribution protocol.

3.1.10 packet-based method without timing support from the network: Packet-based method (frequency or time and/or phase synchronization) where the timing packets are transported over a timing transport agnostic network.

3.1.11 packet master clock: A clock that measures the precise times at which the significant instants of a packet timing signal pass the master's timing reference point (e.g., as they enter the network from the packet master clock or as they enter the packet master clock from the network). These measurements are done relative to the master clock's local time scale. They are forwarded to, and used to control, one or more packet slave clocks.

NOTE – In the case of a periodic packet timing signal (used for one-way frequency distribution), the event packets enter the network from the packet master clock at regular intervals, such that the master's timing information is implied from the nominal frequency of the packets.

3.1.12 packet network timing function (PNT-F): The set of functions within the inter-working function (IWF) that supports the synchronization network clock domain (see Figure B.2 of [ITU-T G.8261]). This includes the function to recover and distribute the timing carried by the synchronization network. The PNT-F clocks may be part of the IWF or may be part of any other network element in the packet network.

When the PNT-Fs are part of the IWF, they may support the CES IWF or change the layer over which timing is carried (i.e., from packet to physical layer and vice versa).

3.1.13 packet slave clock: A clock whose timing output is frequency locked or phase aligned or time aligned to one or more reference packet timing signals exchanged with a higher quality clock.

3.1.14 packet timing monitor: A device capable of analysing the packet flow [e.g., precision time protocol (PTP)] including precise measurement of the sending times and arrival times of timing event messages utilizing an accurate, stable clock. A tapped monitor does not substantively impact the transmission of packets between the communicating clocks; an in-line monitor introduces a fixed, symmetric, delay for packets in the two directions of transmission and thereby does not substantively impact the transfer of timing between the communicating clocks.

3.1.15 packet timing signal: A signal, consisting of a series of event packets or frames, that is used to convey timing information from a packet master clock to a packet slave clock.

Event packets in a packet timing signal may travel from a packet master clock to a packet slave clock or vice versa, but the flow of timing information is always in the direction from master to slave.

The significant instants of the packet timing signal are measured relative to the master's local time scale as they pass the master's timing reference point, and these measurements are communicated to the packet slave clock.

The significant instants of the packet timing signal are also measured relative to the slave's local time scale as they pass the slave's timing reference point.

NOTE 1 – The significant instants of the signal are the set of times that a defined location in each event packet or frame passes a given reference point in the network (e.g., the interface between the packet master clock and the network). Conventionally the defined location is the end of the start-of-frame delimiter, but it may be defined differently in any given packet timing protocol provided the definition is consistent.

NOTE 2 – In the case of a periodic packet timing signal, the master's timing information is implied from the nominal frequency of the packets.

3.1.16 phase synchronization: This term implies that all associated nodes have access to reference timing signals whose significant events occur at the same instant (within the relevant phase accuracy requirement). In other words, the term refers to the process of aligning clocks with respect to phase (phase alignment). This is shown in Figure 2.

NOTE 1 – Phase synchronization includes compensation for delay between the (common) source and the associated nodes.

NOTE 2 – This term might also include the notion of frame timing (i.e., the point in time when the timeslot of an outgoing frame is to be generated).

NOTE 3 – The concept of phase synchronization (phase alignment) should not be confused with the concept of phase locking, where a fixed phase offset is allowed to be arbitrary and unknown. Phase alignment implies that this phase offset is nominally zero. Two signals which are phase locked are implicitly frequency synchronized. Phase alignment and phase lock both imply that the time error between any pair of associated nodes is bounded.

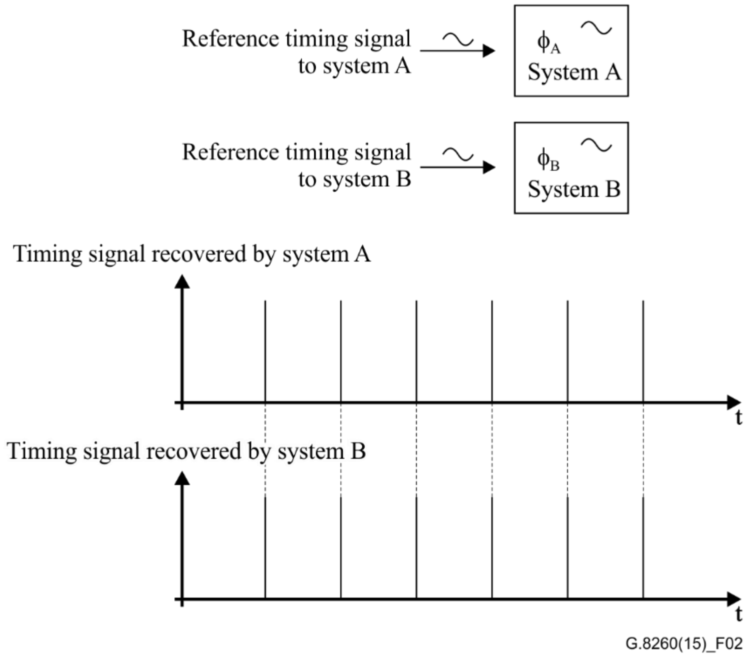

Figure 2 – Phase synchronization

3.1.17 primary reference time clock (PRTC): A reference time generator that provides a reference timing signal traceable to an internationally recognized time standard [e.g., Co-ordinated Universal Time (UTC)].

3.1.18 time clock: A device that provides the elapsed time from a reference epoch.

3.1.19 time synchronization: The distribution of a time reference to the real-time clocks of a telecommunication network. All the associated nodes have access to information about time (in other words, each period of the reference timing signal is marked and dated) and share a common time scale and related epoch (within the relevant time accuracy requirement), as shown in Figure 3.

Examples of time scales are:

- UTC
- International Atomic Time (TAI)
- UTC + offset (e.g., local time)
- Global Positioning System (GPS)
- PTP
- local arbitrary time

Note that distributing time synchronization is one way of achieving phase synchronization.

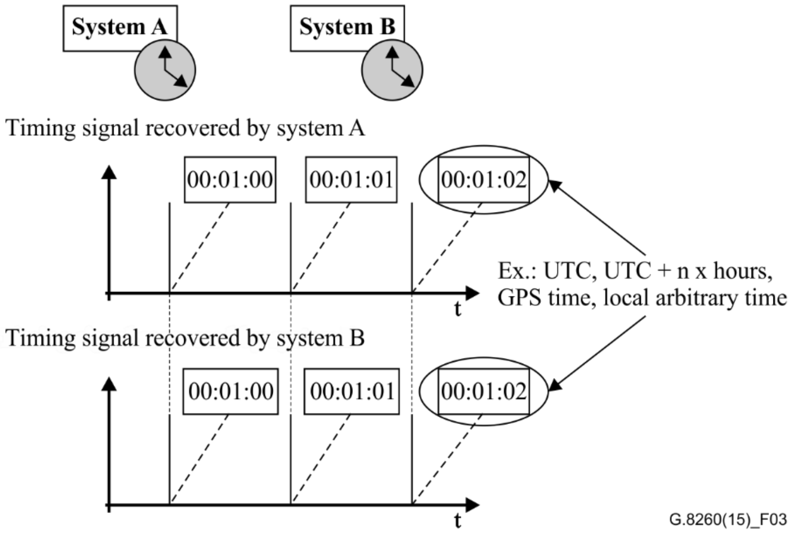

Figure 3 – Time synchronization

3.1.20 time error: The difference between the time of a clock and the time indicated by the time standard. A model for expressing the time error of a clock is described in clause I.3 of [ITU-T G.810].

- **constant time error**: With reference to the time error model provided in clause I.3 of [ITU-T G.810], the constant time error (cTE) of a synchronized clock is the term x_0.
- **constant time error estimate**: Given a time error sequence $\{x(n);\ n = 0, 1 \dots (N - 1)\}$ measured at constant temperature, an estimate of the constant time error is the average of the first M samples of the time error sequence. M is obtained from the observation interval providing the minimum value for TDEV as computed for the given time error sequence. If a frequency offset is present, then a linear regression method in accordance with Appendix II of [ITU-T G.823] can be applied. Considerations for measurement data containing transients is for further study.

  NOTE – In some cases, due to the frequency components of the noise of the signal being measured, it might be difficult to identify a stable, consistent observation interval. These cases must be addressed case by case.

- **dynamic time error**: With reference to the time error model provided in clause I.3 of [ITU-T G.810], the dynamic time error (dTE) of a synchronized clock is the random noise component, i.e.,

$$\frac{\varphi(t) - \varphi_{\text{ref}}(t)}{2\pi\nu_{\text{nom}}}.$$

  The shape of the dTE component may be expressed using the time interval error function TIE(t, τ), and characterized using the related metrics, maximum time interval error (MTIE) and time deviation (TDEV), although the offset from zero of the TIE function may vary depending on the time t when the measurement starts.

- **maximum absolute time error**: The (max|TE|): maximum absolute value of the time error function of a synchronized clock.

3.1.21 message timestamp point: See definition in [IEEE 1588], clause 3.1.33.

3.1.22 reference plane: The boundary between a port of a PTP clock and the network physical medium. Timestamp events occur as messages cross this interface.

3.1.23 timestamp measurement plane: The plane at which timestamps are captured. If the timestamp measurement plane is different from the reference plane, the timestamp is corrected for ingress latency and/or egress latency.

3.1.24 relative time error (TE_R): The difference between two timing signals carrying time. The timing signals may be from different interfaces of the same clock or from different clocks.

A model for expressing the time error is described in clause I.3 of [ITU-T G.810].

The relative time error between timing signal A and timing signal B can be expressed as:

- either a difference between the two timing signals:

$$TE_R(t)_{(A,B)} = T(t)_{(A)} - T(t)_{(B)}$$

- or the difference between the time errors of each timing signal with respect to a common reference timing signal, where each timing signal is separately but simultaneously measured:

$$TE_R(t)_{(A,B)} = TE_R(t)_{(A,\ Ref)} - TE_R(t)_{(B,\ Ref)}$$

As with time error, relative time error $TE_R(t)$ may be characterised by:

- the constant relative time error cTE_R
- the dynamic relative time error $dTE_R(t)$
- and the maximum absolute relative time error: $\max|TE_R|$

The definition above shows that the comparison of the two signals should be made at the same instant in time.

When dealing with sampled data, such as PTP time signals, it is possible that either their sample rates will be the same, but the sampling instants are not aligned, or that the sampling rates may be different. In both of these cases, the computed relative time error will be wrong, especially if there is any variation of time error at frequencies close to the Nyquist frequency.

As a principle, when measuring relative time error, the two time error sequences to be compared should be smoothed before comparison to remove high-frequency components of the time error, and then re-sampled such that the sampling rates and instants of the two signals are aligned. The smoothing process should be defined in the relevant recommendations.

3.1.25 synchronization network segment: Part of a synchronization network using the same synchronization distribution method (e.g., PTP clocks communicating within the same PTP domain).

3.1.26 synchronization interworking function (synchronization IWF): A node function, interworking between synchronization network segments using different synchronization methods.

NOTE – For example, if PTP is being used, there might be a different PTP profile in use in each segment. The two profiles are connected via the synchronization IWF node.

3.1.27 recognized time standard: See definition of recognized standard time source in [IEEE 1588], clause 3.1.62.

3.1.28 traceability: See definition in [IEEE 1588], clause 3.1.81.

3.1.29 traceable: See definition in [IEEE 1588], clause 3.1.82.

4 Abbreviations and acronyms

For the purposes of Recommendations on timing and synchronization in packet networks, the following abbreviations and acronyms apply:

ADEV	Allan Deviation
ARTC	Arbitrary Reference Time Clock
CES	Circuit Emulation Service
cTE	constant Time Error
dTE	dynamic Time Error
FFO	Fractional Frequency Offset
FFM	Flicker Frequency Modulation
FM	Frequency Modulation
FPC	Floor Packet Count
FPP	Floor Packet Per cent
FPR	Floor Packet Rate
GPS	Global Positioning System
IWF	Inter-Working Function
MAFE	Maximum Average Frequency Error
MATIE	Maximum Average Time Interval Error
MDEV	Modified Allan Deviation
MTIE	Maximum Time Interval Error
NTP	Network Time Protocol
OFD	Observed Floor Delay
PDV	Packet Delay Variation
PEC-M	Packet-based Equipment Clock Master
PEC-S	Packet-based Equipment Clock Slave
PM	Phase Modulation
PNT-F	Packet Network Timing Function
PRC	Primary Reference Clock
PRTC	Primary Reference Time Clock
PTP	Precision Time Protocol
RWFM	Random Walk Frequency Modulation
TAI	International Atomic Time
TDEV	Time DEViation
TDM	Time Division Multiplexing
TE	Time Error
TIE	Time Interval Error
UTC	Coordinated Universal Time

WFM White Frequency Modulation

5 Conventions

No conventions are used in this Recommendation.

6 Description of packet timing concepts

6.1 The nature of packet timing

A simplistic view of a generic slave clock is that it takes frequency information in, and puts frequency information out, as shown in Figure 4.

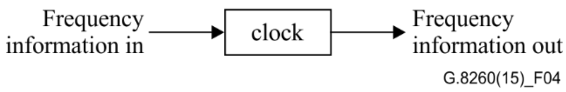

Figure 4 – Generic slave clock

Conventionally, this frequency information is encoded as a timing signal. This is typically implemented as a periodic digital signal, where the edges of the signal are reference points in time known as the "significant instants" of the signal. Timing jitter and wander causes these significant instants to vary slightly from their ideal position in time, i.e., they may not occur at precisely equally spaced points in time. A physical layer timing signal is shown in Figure 5.

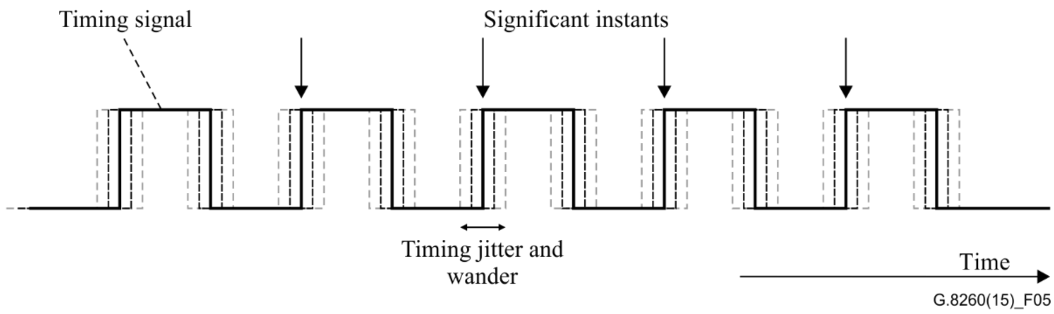

Figure 5 – Physical layer timing signal

A packet timing signal is similar in concept. The frequency is encoded as a series of time-critical packets in a network, known as event packets. While the transmission medium is different (packets on a network as opposed to signals on a wire), the packets still contain significant instants (normally the front edge of the packet), with a defined ideal position in time. The variation of the significant instants around their ideal position is termed packet delay variation (PDV). This is shown in Figure 6.

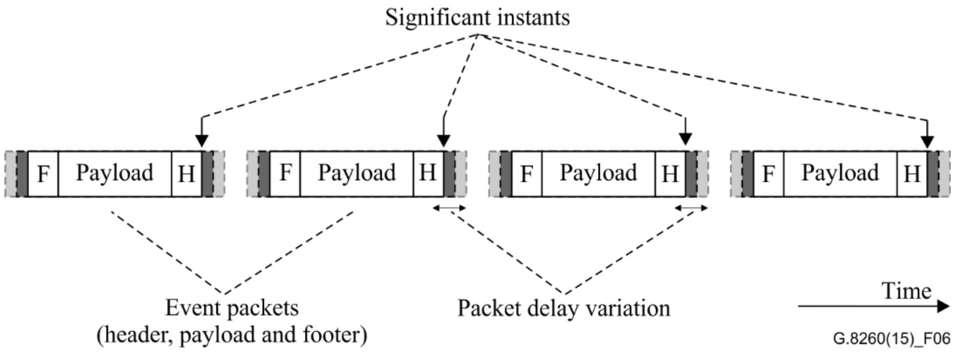

Figure 6 – Packet timing signal

Some of the causes and characteristics of PDV and other impairments that may be introduced by the packet network are discussed in clause 10 of [ITU-T G.8261].

Some packet timing signals may be periodic (e.g., circuit emulation packets containing constant bit rate data), and for these the ideal position in time is implicitly given by the packet rate. Other packet timing signals are not periodic (e.g., PTP or NTP), and for these the ideal position in time is given by a timestamp embedded in the packet data. It is important to note that both periodic and non-periodic packet timing signals are still time domain signals. It is the position in time of the packets that is significant, not the contents of the packets.

6.2 Differences between packet-based and physical layer timing systems

Packet-based timing systems are not fundamentally different from physical layer timing systems. Conceptually, both utilize timing signals that are sequences of periodic or timed events, termed significant instants, where there is a notion of the ideal position in time for each event. Similarly, after transmission of these timing signals through the network, there will be some phase noise component, corrupting this ideal position in time. The recovery of the original timing signal is achieved by filtering the incoming timing signal to remove the transport-related phase noise and generate a clean output.

However, there are some differences that lead to packet timing signals having different characteristics to physical layer timing signals:

- Rate of significant instants:

  The packet rate in a packet timing signal is much lower than the frequency of most physical layer timing signals. For example, in PTP (defined in [IEEE 1588]), the sync message rate will normally be in the range 1-128 Hz, while a conventional E1 timing signal has a frequency of 2.048 MHz.

  Secondly, the packets that form the significant instants need not be sent at precisely regular intervals. While the mean rate is specified, the intervals between packets may vary. Timestamps are used to identify the precise sending time, relative to a pre-determined epoch.

- Amplitude and nature of noise processes:

  The principal cause of noise in a packet timing system is PDV. The amplitude and distribution of PDV is much larger than jitter and wander in physical layer timing systems, and it may contain very low frequency components such as diurnal wander due to network loading variations.

  Unlike physical layer noise, the PDV depends not only on the physics of components but also on the architecture and implementation of network elements. Therefore, the noise is more complex and harder to model.

6.3 Classes of packet clocks

There can be several classes of packet-based clocks, depending on the combination of input and output timing signal classes. Table 1 shows the different classes, with real-world examples of each case.

Table 1 – Classes of packet-based clocks

Packet-based clock class	Input timing signal	Output timing signal	Examples
Packet master clock PEC-M	Physical layer timing signal	Packet timing signal	PTP master, NTP server Ingress CES IWF (Note 1)
Packet slave clock PEC-S	Packet timing signal	Physical layer timing signal	PTP slave, NTP client Egress CES IWF (Note 2)
Combined packet slave clock and packet master clock	Packet timing signal	Packet timing signal	PTP boundary clock NTP stratum *n* server ($n > 1$)
NOTE 1 – i.e., TDM to packet direction, see term "ingress IWF" in [ITU-T Y.1413]. NOTE 2 – i.e., packet to TDM direction, see term "egress IWF" in [ITU-T Y.1413].			

6.4 Two-way timing protocols

Packet timing signals may flow from packet master clock to packet slave clock or vice versa. However, in each case, the flow of timing and synchronization is always from master to slave.

In the case of a packet timing signal flowing from a packet master clock to a packet slave clock (e.g., the PTP *sync* messages), the time of exit of each event packet from the packet master clock is measured (to be precise, the time relative to the master's time scale at which the significant instant of each event packet passes the master's timing reference point). This information is sent to the packet slave clock either in a timestamp embedded in the event packet, or in a subsequent information packet (e.g., the PTP *follow_up* message).

On reception at the packet slave clock, the arrival time of the event packet is measured (to be precise, the time relative to the slave's local time scale at which the significant instant of each event packet passes the slave's timing reference point). The two times are compared, creating a series of time differences. These time differences are then filtered, and may be used to control the frequency of the output timing signal.

In the case of a packet timing signal flowing from a packet slave clock to a packet master clock (e.g., the PTP *delay_request* messages), the time of exit of each event packet from the packet slave clock is measured. On reception at the packet master clock, the arrival time of each event packet is measured, and this information is sent to the packet slave clock in a subsequent information packet (e.g., the PTP *delay_response* message).

The use of a two-way timing protocol (such as PTP or NTP) makes it possible to align the local time scale to the master time scale. The four times may be used to calculate the round-trip delay of the message exchange, and hence to calculate the time offset between the local and master time scales.

The timing message exchange is shown in Figure 7.

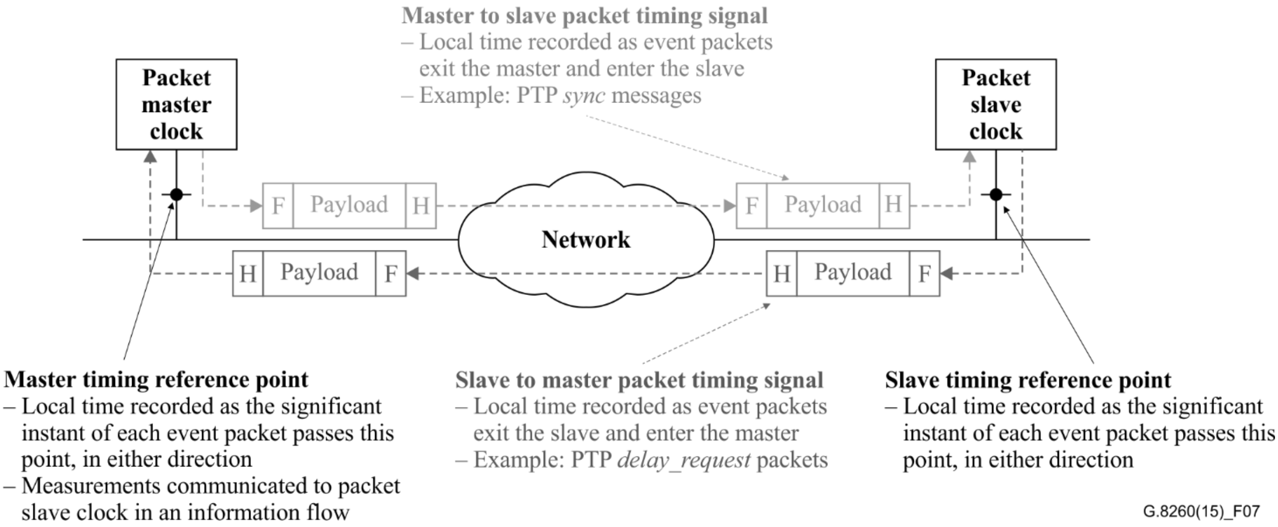

Figure 7 – Packet timing signal flow and timing reference points

6.5 Packet delay sequence measurement

In general, a packet delay sequence measurement involves comparing time instants on a sequence of packets, such as those of a packet timing signal, as they pass two points in the network. A configuration for performing such a measurement is shown in Figure 8. For each packet, a difference is computed between the time instant taken at the point of origin and the time instant taken at the point of destination.

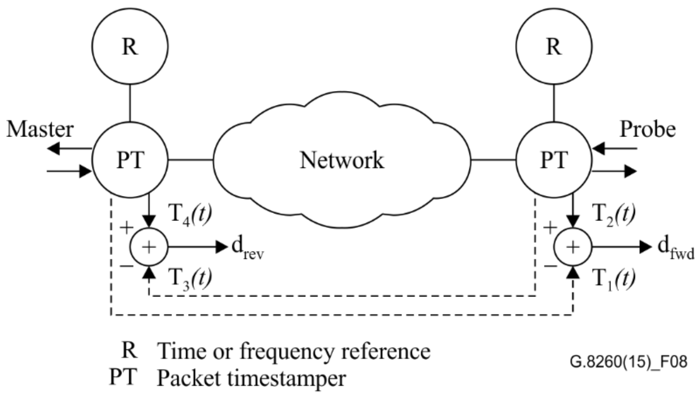

Figure 8 – Configuration for packet delay sequence measurement

The forward and reverse delay sequences are calculated from timestamp sequences.

The forward delays are:

$$d_{\mathrm{fwd}}(i) = T_2(i) - T_1(i) \tag{1}$$

Similarly, the reverse delays are:

$$d_{\mathrm{rev}}(j) = T_4(j) - T_3(j) \tag{2}$$

An ideal configuration for making this measurement places two references traceable to a common time standard at each of the two measurement points. Such a configuration assesses not only the variation of packet delay, but also the packet transit time.

In many circumstances, such as packet-based frequency synchronization, the focus is on variation of packet delay rather than absolute packet delay. In such a case, frequency references can be employed for the references R and a common time reference is not required. In this case there will be an arbitrary but constant time error in the resulting delay values, and the metrics used to qualify frequency synchronization cancel this constant time error out.

In many circumstances, such as packet-based frequency synchronization, the focus may be in either forward or reverse delay. However, in time synchronization, usually both directions are of equal importance.

The use of unstable or inaccurate references directly impacts the packet delay measurement, and may lead to limitations regarding the length of the packet delay measurement. If the references are frequency standards, packet delay can be studied with the same precision as for the case where the references are time standards. If practical, a common frequency standard should be used for both R references. In other cases, separate primary reference clocks could be used.

The probe function could be implemented as separate equipment or in the case where the first measurement point is at the source of the packet timing signal of interest, integrated into that equipment. In this case, the time instant could be delivered within the packet in the form of a timestamp. Similarly, in the case where the second measurement point is at the destination equipment, the probe function may be integrated into that equipment. Any inaccuracy of the timestamping function in the probes directly impacts the precision of the packet delay measurement.

In the case where packets are sent according to a schedule that is known in advance, such as packets spaced by a uniform interval of time (e.g., CES), the relative origin timestamps are implicit, and the PDV measurement can be performed with timestamping at the destination node.

If the network contains boundary clocks (BC), see Figure 9, the measurement device on the right communicates, instead of with the source device on the left, with an intermediate BC that has time error. The timestamps created by the BC are $T_1' = T_1 + Err_{\mathrm{BC}}$ and $T_4' = T_4 + Err_{\mathrm{BC}}$ where Err_{BC} is the time error of the BC. Due to the time error of the BC the differences between the times stamps:

$$d_{\mathrm{afwd}}(i) = T_2(i) - T_1'(i) \tag{3}$$

and

$$d_{\mathrm{arev}}(j) = T_4'(j) - T_3(j) \tag{4}$$

represent apparent delays d_{afwd} and d_{arev} rather than actual delays.

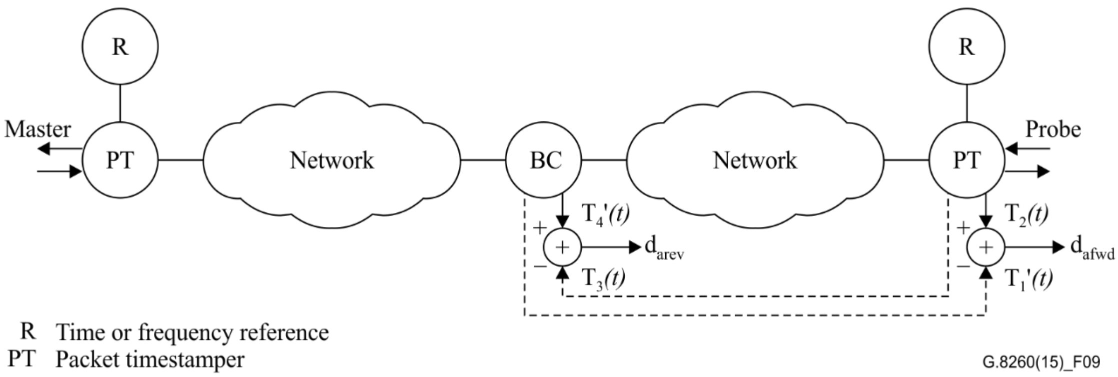

Figure 9 – Configuration for packet delay sequence measurement in a network including BCs

It can be shown that d_{afwd} and d_{arev} can be used interchangeably with d_{fwd} and d_{rev} in the formulas that derive time error from delay measurements.

6.6 Packet timing signal equipment interface characterization

The configuration described in clause 6.5 for measuring packet delay variation can be extended to measurement of the packet timing signal at an equipment interface. In this case, the packet timestamper with reference is connected directly to the packet timing signal interface with no intervening network.

A configuration for performing such a measurement is shown in Figure 10. The packet time-error signals resemble the time error function *x(t)* described in clause 4.5.13 of [ITU-TG.810]. For each packet, a difference is computed between the timestamp from the device and the timestamp taken on that same packet from the packet timestamper (PT) with reference (R). This is true for the streams in both directions with the details of the difference operations indicated in the figure.

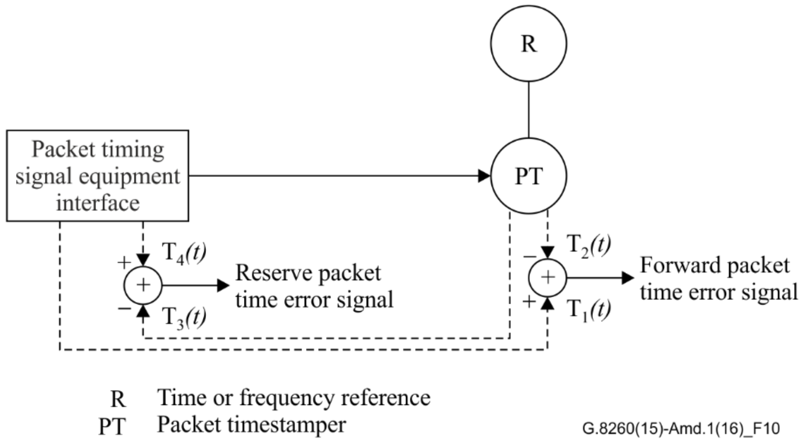

Figure 10 – Configuration for packet timing signal equipment interface measurement

Requirements on the accuracy of the reference (R) are driven by the characteristics of the packet timing signal, and in many cases might exceed those for studying network PDV. If the packet timing signal is derived directly from a primary reference, reference (R) would need to be a primary reference, ideally one with greater stability. Further, in cases where the device under study takes an external reference directly or is traceable to an external reference, the optimal configuration is for both the device and the packet timestamper (PT) in Figure 10 to share the same reference (R).

Appendix I

Definitions and properties of packet measurement metrics

(This appendix does not form an integral part of this Recommendation.)

NOTE – This appendix contains information related to ongoing studies on the definition of suitable PDV metrics. The text below is for information only, and may be revised in a future version of the Recommendation. It must not be used as normative text, nor as an implied specification of a packet slave clock.

I.1 Introduction

With the telecommunications industry evolving and rapidly adopting packet technology, much emphasis has been placed on addressing packet synchronization and timing, including the use of measurement data to assist in specifying the performance of packet-based clocks.

Physical layer timing signal stability quantities, including metrics such as maximum time interval error (MTIE) and time deviation (TDEV), have been used extensively and are central to synchronization measurement analysis. For a packet clock, the level of stability at the clock's packet network input has a direct bearing on the stability of the clock output.

In terms of the packet metrics, the goal of a first category of PDV metrics, introduced in clause I.4, is to formulate packet-based stability quantities (metrics) that will provide a means of estimating the physical-based stability quantities for the packet clock output. This is illustrated in Figure I.1.

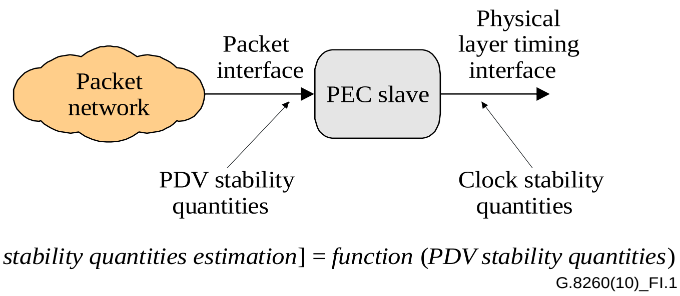

Figure I.1 – Packet equipment clock interfaces

A second category of PDV metrics is also introduced in clause I.5. The goal of this second category is not directly to provide an estimation of the physical-based stability quantities for the packet clock output, but simply to study the population of timing packets within a certain delay window range.

PDV measurement guidelines are provided in clause 6.5.

For packet measurement data analysis, packet selection is added as an important component to the analysis. Indeed, in order to reduce the input PDV noise, the packet slave clock implementations generally use only a subset of the received timing packets.

Therefore, a first simple approach to analyse the PDV as received by a packet slave clock can be to display the measured PDV in the form of a histogram. It generally provides useful information about the population of packets in different delay regions, and is in some cases sufficient to analyse the network conditions. Figure I.2 shows an example plot of the measured PDV and the corresponding histogram.

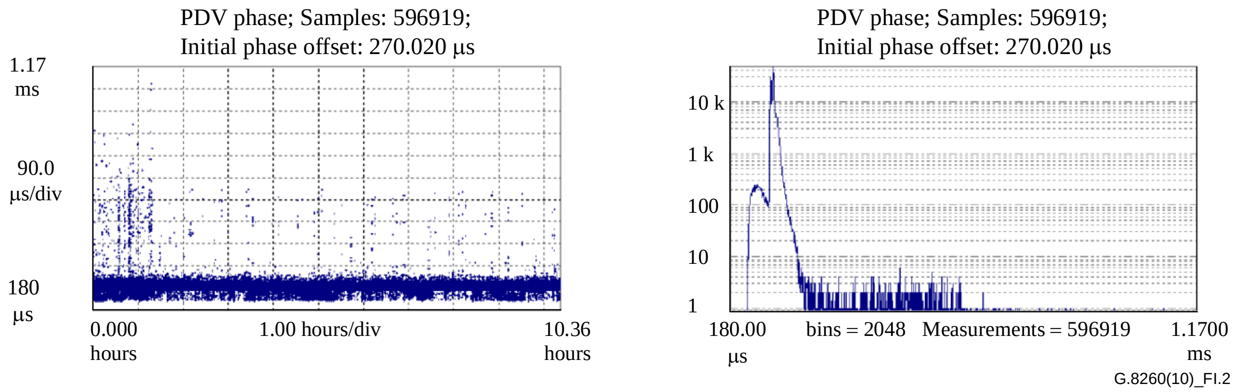

Figure I.2 – Measured packet delay and corresponding PDV histogram

In a second approach, mathematical tools (called "metrics" in this appendix) can be applied on a given PDV measurement to analyse it more in detail. Those metrics generally use only a subset of the packets. The packet selection can be either integrated into the calculation or performed as a pre-processing step. For example, the packet selection can focus on the minimum packet delay floor or more generally on some other region of packet delays.

With regard to the packet selection just discussed, it is important to point out the link between the methods of packet measurement data analysis described here and packet clock algorithms as they exist in actual equipment. For both, packet selection is important for optimization given the realities of packet delay variation.

However, it is important to mention that due to the proprietary nature of most of the packet slave clock implementations today, especially regarding the packet selection criteria, the packet selection used by a given PDV metric may not correspond to the criteria used in the packet slave clock of interest. Therefore, there can be some discrepancies between the information provided by a given PDV metric and the real performances achieved by a packet slave clock.

Methods for alignment of the results provided by the PDV metrics with the performance of the packet slave clock are still under study. Alignment may involve the specification of some minimum common behaviour in the packet selection criteria in the packet slave clock implementations.

Moreover, it is important to mention that PDV metrics compute an estimate of achievable performance through the use of PDV sample information only, and do not consider the effects of internal oscillator noise in a packet slave clock. Non-negligible differences between the estimate and the actual performance of a packet slave clock may sometimes be observed because of this effect. In order to take oscillator noise into account, the noise generation components of a packet slave clock are considered in [b-ITU-T G.8263].

While metrics can provide the basis for setting equipment requirements and network limits, their value as general analysis tools leading to insight into particular sets of measurement data should not be overlooked. For time division multiplexing (TDM) synchronization measurements, normative limits have been applied to the MTIE and TDEV calculations, but other metrics such as Allan deviation (ADEV) and modified Allan deviation (MDEV), while not associated with normative telecom limits, have great utility as analysis tools.

I.2 Definition of the time error sequence

For packet timing signals the packet time error sequence can be established in the following way. For specificity, consider the transfer of timing packets originating at the packet master clock and terminating at the packet slave clock. In the case of PTP (see [IEEE 1588]) the rate of packets is determined via negotiation between master and slave and can be as high as 128 packets/s. PTP packets may not be equally spaced, but will meet this nominal rate over the long term. The ideal position in time for these packets is given by a timestamp embedded in the packet data. The rate of packets in the reverse direction, from slave to master, can be different from the rate in the forward direction, from master to slave.

Packets leave the master with a long-term mean spacing of $\tau_F = 1/f_F$ where f_F is the average forward packet rate. From a signal processing perspective, the sampling rate is f_F and an arbitrary mathematical-time origin for describing the times of departure from the master can be chosen. With this choice of time origin, the ith packet departs the master at time $t = i\tau_F$. In practice the ith packet will depart at time $T_1(i)$ which is but approximately equal to $i\tau_F$. Note that in the case of circuit emulation, the times of departure are considered to be exactly spaced by τ_F. The ith packet then arrives at the slave at time $T_2(i)$ where:

$$d_{\text{fwd}}(i) = T_2(i) - T_1(i) \tag{I-1}$$

is the forward transit delay and $T_2(i)$ is expressed relative to the master clock. Assuming the time instants are determined by synchronized clocks, this forward transit delay is composed of a fixed, albeit unknown, delay and a random delay component that is the result of queuing delays in the intermediate network elements. If the application is frequency synchronization, the fixed (unknown) delay is not relevant and can be ignored, leaving the random delay component as the principal cause of time error.

The same principle applies for packets that traverse the network from the slave to the master. Denoting the packet rate in the reverse direction by $f_R = 1/\tau_R$, the jth packet will depart at time $T_3(j)$ which is approximately equal to $j\tau_R$.

$$d_{\text{rev}}(j) = T_4(j) - T_3(j) \tag{I-2}$$

is the reverse packet delay that is composed of a fixed, unknown, delay and a random delay component. $T_3(i)$ and $T_4(i)$ are expressed relative to the same clock.

If the forward delay were known *a priori*, then the master and slave clock could be time synchronized using Equation I-1. However, if the delay is not known *a priori*, and the slave takes the master time to be $T_1(i)$ at the instant it receives the forward packet, the (hypothetical) time-error at this instant will be $T_1(i) - T_2(i) = -d_{\text{fwd}}(i)$ (i.e., the slave's estimate of the master time minus the actual master time), because the actual time relative to the master clock that the forward packet is received at the slave is $T_2(i)$. Then, if the (hypothetical) forward packet time-error signal $x_F(t)$ is considered, the sample of the forward packet time-error signal taken at the sampling instant $T_1(i)$ is:

$$x_F(t) = -d_{\text{fwd}}(i) \tag{I-3}$$

Similarly, if the reverse packet time-error signal $x_R(t)$ is considered, then the sample of the reverse packet time-error signal taken at the sampling instant $T_3(j)$ is:

$$x_R(t) = d_{\text{rev}}(j) \tag{I-4}$$

The sign is reversed compared to the forward time error signal, because now the slave takes the master time to be $T_4(i)$ when it is actually $T_3(j)$; the time error is $T_4(j) - T_3(j) = d_{\text{rev}}(j)$.

This means that increasing forward delay corresponds to increasingly negative time error in the slave, whereas increasing reverse delay corresponds to increasingly positive time error. That is, the sequences $\{-d_{\text{fwd}}(i)\}$ and $\{d_{\text{rev}}(j)\}$ are equivalent to packet time error sequences, but on a non-uniform time grid. Note that delay measures the difference between two time instants, while time error measures the difference between two clocks at the same instant if the packet were used to set or estimate the time of the slave clock without any filtering. The normal packet time error sequences, $\{-d_{\text{fwd}}(i)\}$ and $\{d_{\text{rev}}(j)\}$ are actually sequences generated by sampling the packet time-error signal $x_F(t)$ and $x_R(t)$ on a uniform grid with sampling interval τ_F and τ_R in the forward and reverse directions creating time error sequences $x_F(i\tau_F)$ and $x_R(j\tau_R)$. In the following sections $x(t)$ corresponds to either $x_F(t)$ or $x_R(t)$ depending on which one-way sequence is of interest. Also, if the packet rates in the two directions are equal, then we set $\tau_F = \tau_R = \tau_0$.

I.3 Packet selection and filtering

Physical layer timing signals are stationary and Gaussian in nature. Therefore, the relevant applied stability quantities (i.e., MTIE and TDEV) will usually use every noise sample point (significant instant) in the stability quantification process in order to filter out as much noise as possible and achieve the best stability quantification possible.

Packet-based timing signals, on the other hand, are not always stationary or Gaussian in nature. Hence, methods of quantifying them (thus attaining a better estimation of their ability to carry timing information) would usually require the selection of only a subset of their entire population or in general performing some pre-filtering before applying the specific stability quantification analysis.

The following discussion focuses on the approach that involves a selection of packets.

I.3.1 Packet selection types

As mentioned in clause I.1, when applying some PDV metrics, packet selection and filtering can be incorporated into the calculation or into separate pre-processing steps.

- The packet selection and filtering techniques integrated into the calculation are useful in metrics that are intended to determine the characteristics of the packet network in terms of the PDV behavior. The main benefit of this approach is to provide a generic tool independent of the characteristics of a specific packet slave clock implementation (e.g., time interval used to select packets). The main purpose of this approach is therefore to support vendors with progressing packet timing recovery techniques (class B metrics).
- On the other hand, pre-processing selects packets from suitable, pre-defined time window lengths. Therefore, the selection process resembles that of a practical packet clock in steady-state operation. This approach is therefore more suitable for the definition of metrics used to specify network limits (i.e., class A metrics) as an assumption is made on a "minimum" implementation of a packet slave clock as specified in [b-ITU-T G.8263].

I.3.1.1 Pre-processed packet selection

With pre-processed packet selection, quantifying packet timing signals is carried in two steps:

1) Applying a specific packet selection procedure to select a specific subset of packet delay samples, having similar delay properties, among the entire population of packet delay samples.
2) Applying the required stability quantification algorithm (metric) over the selected group of samples to get an estimation of the achievable output clock quality estimation.

   NOTE – As mentioned earlier, there can be discrepancies between the information provided by a given PDV metric and the real performances achieved by a packet slave clock.

This is shown in Figure I.4:

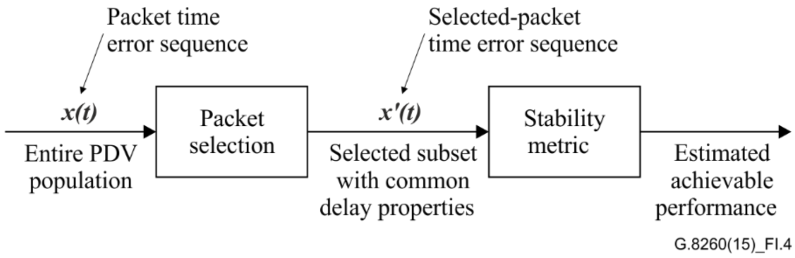

Figure I.4 – Pre-processed packet selection

In essence, an input packet time error sequence $x(t) = x(i\tau_0)$ is subject to packet selection, which produces a new packet time error sequence. The input packet time error sequence is divided into time windows of equal length $m\tau_0$ where *m* is the number of samples in the selection window. A fraction of packets is selected from each window in a similar manner and the information is combined so that each window produces a single value to the new packet time error sequence $x'(t) = x'(i\tau_s)$. The sample interval of the selected-packet time error sequence is $\tau_s = m\tau_0$. Unless otherwise specified, the time errors of the selected packets in each selection window are averaged to produce a single delay value.

When a metric is computed using pre-processed packet selection, the name of the metric is prepended by the term *pktselected*, e.g., *pktselectedTDEV* (see clause I.4.2.1).

In the case of pre-processed packet selection, the preliminary packet selection process is independent of the applied stability quantification. Thus, different combinations of the two might yield interesting properties that require investigation. Both need to be fully defined as each has significant influence on the resulting performance measurement.

I.3.1.2 Pre-processed packet filtering

As described above, an input packet time error sequence $x(t)$ that is subject to packet selection, produces a new selected-packet time error sequence $x'(t)$. Additionally, that new packet time error sequence $x'(t)$ may be subsequently filtered to create a filtered-packet time error sequence $y(t)$. This flow is shown below in Figure I.5.

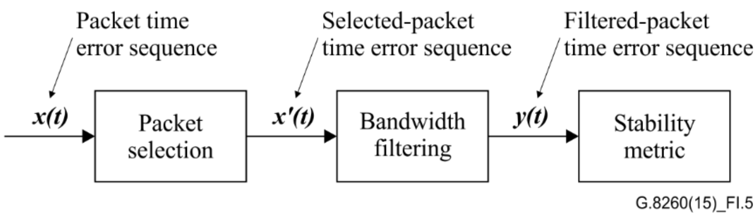

Figure I.5 – Packet selection and filtering flow

The selected-packet time error sequence $x'(t)$ may be filtered by applying an averaging function in line with the clock bandwidth, with averaging time related to the window length of the packet selection process. In particular, 1:10 is as an example of a suitable ratio of the window length of the packet selection block to the time constant of the bandwidth filtering block in Figure I.5.[1]

The parameters of the selection process must be chosen to ensure that sufficient packet information is available to allow the filtering process to operate. As an example, assuming a packet rate of 1 packet/s and a selection window of 100 s, the minimum possible selection percentage is 1%, resulting in the selection of 1 packet in every window.

In many cases a higher packet rate would be used for these measurements in order to get a higher number of samples.

The following applies a sliding window averaging function with length b (the number of windows, where each window has K samples):

$$y(n\tau_0) = \frac{1}{b}\sum_{i=n}^{n+b-1}\left(x'_i\right), i = 1,2,\ldots N - b + 1 \tag{I-7}$$

The filtered-packet time error sequence $y(t)$ may be used to compute TIE and applied to traditional synchronization metrics defined in [ITU-T G.810] such as MTIE and TDEV. When TIE or a metric are computed using $y(t)$, the term *pktfiltered* is prepended to TIE or the name of the metric, respectively. As illustrated in Figure I.5, the *pktfiltered* operation includes both packet selection and bandwidth filtering.

I.3.1.3 Integrated packet selection

With integrated packet selection, the packet selection and sometimes also the filtering steps are integrated into the metric calculation, as shown in Figure I.6. Generally, this involves replacing a full population averaging calculation with a selection process that may or may not itself include averaging.

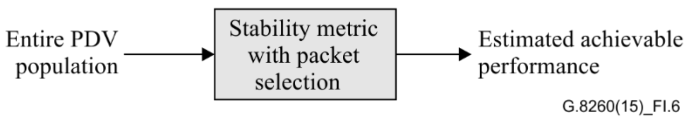

Figure I.6 – Integrated packet selection

NOTE – As mentioned earlier, there can be some discrepancies between the information provided by a given PDV metric and the real performances achieved by a packet slave clock.

I.3.1.4 Packet selection windows

For the packet selection described in the previous clauses, packet selection is carried out in a sequence of time windows applied to the entire set of data. There is a variety of possible approaches to this progression of windows. The windows could be non-overlapping but contiguous (also known as "jumping windows"), overlapping by some number of samples (known as "step-overlapping"), or overlapping by sliding the window sample by sample. Figure I.6*bis* illustrates these three approaches:

[1] The time constant of a PLL, also known as its characteristic response time, provides an indication of the duration of the effects on the output of the PLL due to a given input. This is why it is important that the selection window is properly chosen in order to get a significant number of samples during this period of time. Note that the time constant τ_c is related to the 3 dB bandwidth of the PLL $f_{3\,dB}$, by the following relationship: $\tau_c = 1/(2\pi \cdot f_{3\,dB})$.

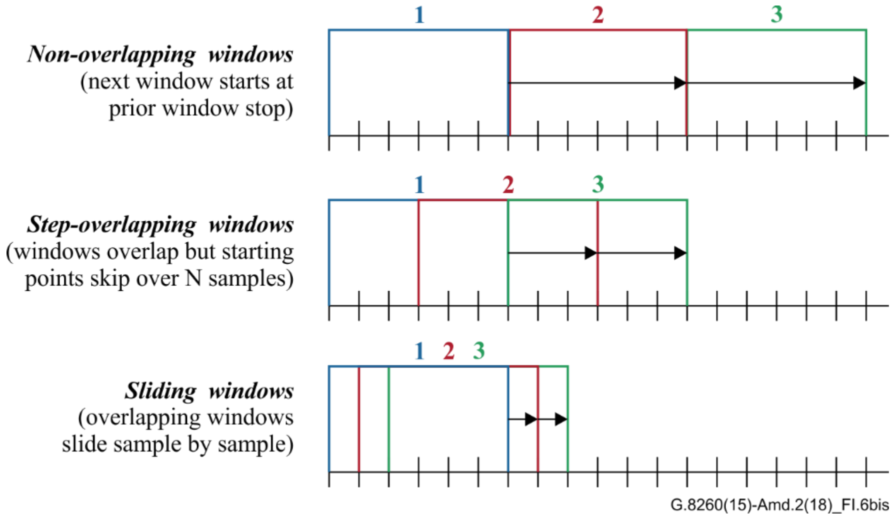

Figure I.6*bis* – Packet selection windowing

To illustrate applying windowing to a set of data, consider the following specific case of step-overlapping windows. If the duration of the packet selection window is 200 seconds, a suitable window step size is 10% of 200 seconds, or 20 seconds. The first window would span 0 to 200 seconds, the second window 20 to 220 seconds, the third window 40 to 240 seconds, and so on. If the duration of the entire set of data is 10 000 seconds, this process would continue until reaching the final window, which spans 9800 seconds to 10 000 seconds.

I.3.2 Packet selection methods

Four examples of packet selection methods are described in the clauses that follow. The first two, minimum packet selection and percentile average packet selection, focus on packet data at the floor delay. The second two, band average packet selection and cluster range packet selection, can be applied either at the floor delay or at some other region.

Equation I-4 defines that the reverse time errors are the same as the reverse delays. Thus, the minimum time error values correspond to the floor delays. However, Equation I-3 defines that the forward time errors are the inverse of forward delays. Thus, the maximum time error values correspond to floor delays.

I.3.2.1 Minimum packet selection method

The minimum packet selection method involves selecting the minimum delayed packet, i.e., the maximum or minimum value within a section of forward or reverse time error data, correspondingly. This can be represented as.

$$x_{\min}(i) = -\min\left[-x_F(j)\right] \text{for} \left(i \le j \le i+n-1\right) \qquad \text{(I-8)}$$

when using the forward time error sequence and.

$$x_{\min}(i) = \min\left[x_R(j)\right] \text{for} \left(i \le j \le i+n-1\right) \qquad \text{(I-8a)}$$

when using the reverse time error sequence.

I.3.2.2 Percentile average packet selection method

The percentile average packet selection method is related to the minimum packet selection method, except that instead of selecting the minimum, some number (or some percentage) of minimum values are chosen and averaged together. It is a special case of the band average packet selection method described in clause I.3.2.3 with the lower limit set to zero.

I.3.2.3 Band average packet selection method

The band average packet selection method can be used to select a section of packet data at the floor or from some other region such as the ceiling or somewhere else above the floor. The band is defined by two percentile values representing the upper and lower selection bounds. To perform the band average packet selection, it is first necessary to represent the sorted packet time-error sequence. Let x'' represent this sorted phase sequence from minimum to maximum for the reverse time error sequence, and maximum to minimum for the forward time error sequence, over the range $i \le j \le i + n - 1$. Next, it is necessary to represent the indices that are themselves set based on the selection of the two percentile values.

Let a and b represent indices for the two selected percentile values. The averaging is then applied to the x'' variable indexed by a and b. The number of averaged points m is related to a and b: $m = b - a + 1$.

$$x'_{band\_avg}(i) = \tfrac{1}{m}\sum_{j=a}^{b} x''_{j+i} \tag{I-9}$$

Each of the indices a and b is determined by rounding to find the closest index to the desired percentile value. The additional constraint is that both indices have a minimum value of the first index and a maximum value of the last index. Further, at least one point within the data set must be selected. Thus, for example, a set of ten points with the percentile values set to 0% and 2% (0.02), both a and b would be set to the minimum index so that at least a single point would be selected.

I.3.2.4 Cluster range packet selection method

The cluster range packet selection method uses a time- and/or phase-bounded range rather than indices based on percentiles (probabilities) to perform the packet selection. This selection method involves the selection of a group of one or more packets that are closely related with respect to their transit time. The location of the cluster may be made based on various criteria, for example, packets at the floor or from some other region observed in the window interval, or the location of the cluster may be based on other criteria or information outside the interval. The cluster of packets could then be processed in a variety of ways to generate a single value for that interval, such as the mean transit time of all packets within the cluster.

Figure I.7 shows an example of a packet delay sequence, zooming in on an example of a packet cluster for a single window interval.

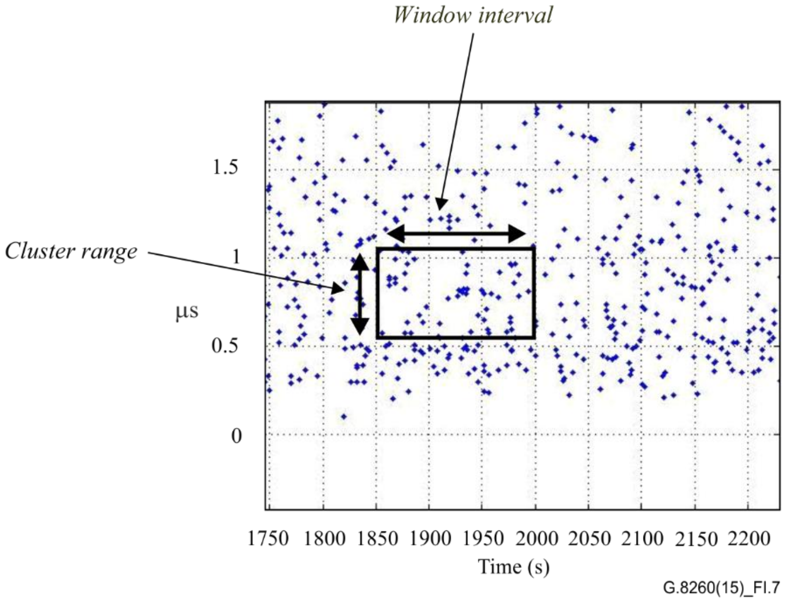

Figure I.7 – Example of concept of cluster range packet selection

The cluster selection method involves the following pre-determined choices:

- First, the range of the packet transit times accepted within a cluster is set and is related to the target performance. That is, the range can be chosen to best serve the application for which the clock is intended. This is the *cluster range*, δ, and is specified in units of time.
- Second, the selection window interval for the cluster is set.
- Third, the *cluster location* or *cluster anchor value*, $a(n)$, within the overall distribution of packet delays is variable and can be programmed to best characterize the type(s) of noise introduced by the packet network. That is, the optimal time error is obtained when the cluster selection method identifies the packets that represent the most stable transit delay. The specific cluster anchor type may be considered as the cluster rule, denoted as *clusterType = rule* or $clusterType_{rule}$.

Denote the time error sequence of the packet timing signal at the slave clock packet interface by $\{x(k\tau_P)\}$. That is, the underlying sampling interval (nominal packet interval) is τ_P. The cluster selection method considers K samples (packets) using a fixed-window processing architecture and generates a new time error sequence $\{x'(n\tau_0)\}$ where $\tau_0 = K\tau_P$. Note that the sample value $x(n\tau_0)$ is based on the K input samples $\{x(m\tau_P);\ m = nK, (nK+1), ..., ((n+1)K-1)\}$. This sample value can be expressed as:

$$x'(n\tau_0) = \frac{\sum_{i=0}^{(K-1)} x\big((nK+i)\tau_P\big)\cdot\phi(n,i)}{\sum_{i=0}^{(K-1)} \phi(n,i)} \qquad \text{(I-10)}$$

where in Equation I-10, ϕ( ) is the indicator function that expresses the selection mechanism in a mathematical manner and is given by:

$$\phi(n,i) = \begin{cases} 1 \quad \text{for} \quad \left|x\big((nK+i)\tau_P\big) - a(n)\right| \le \dfrac{\delta}{2} \\ \quad 0 \quad \text{otherwise} \end{cases} \qquad \text{(I-11)}$$

In Equation I-11, δ is the cluster range and $a(n)$ is the anchor value for the particular window of K samples. Note that Equation I-11 generates the new time error sequence by computing the average over those packets that satisfy the selection rule.

The anchor value can be interpreted as a nominal value for the window and is established according to a pre-determined cluster type. For example, the selection rule for this anchor value could be:

- Minimum transit delay over the K packets in the window, represented as *clusterType = min* or $clusterType_{min}$.
- Average (mean) transit delay over the K packets in the window, represented as *clusterType = mean* or $clusterType_{mean}$.
- An absolute minimum value that may be determined before, during or after the sample window, represented as *clusterType = min_absolute* or $clusterType_{min\_absolute}$.
- When using an absolute value, it is possible that no packets may be selected within the window (similar to a total packet loss situation). Note that the determination of an absolute minimum value after the sample window (as opposed to before or during) would only be used in post-analysis situations, as the information regarding the future packet delay transit times is not available to the client in a real-time system.

It is common to refer to the anchor value as the transit delay of a particular packet that is then called the anchor packet. This is generally true, except when the anchor value is not associated with a particular packet (e.g., mean value or absolute minimum value).

It may be helpful to use the representation *cluster* (δ, *clusterType*) where δ is the cluster range, and the *clusterType* is an indication of the rule used to generate the anchor value.

I.3.3 Consideration of non-stationary network conditions

As the packet selection can focus on a particular statistical region, it is important to consider the case where network packet delay statistics are not stationary, but rather change over time. For example, if a floor-based metric is applied to packet measurement data where the floor shifts, the application of the floor-based metric would perhaps be best applied to sections of the data separately (see Figures I.8 and I.9). In many cases, segregating the data into sections might not be so straightforward, such as the case of an increasing load ramp. Such a situation is for further study.

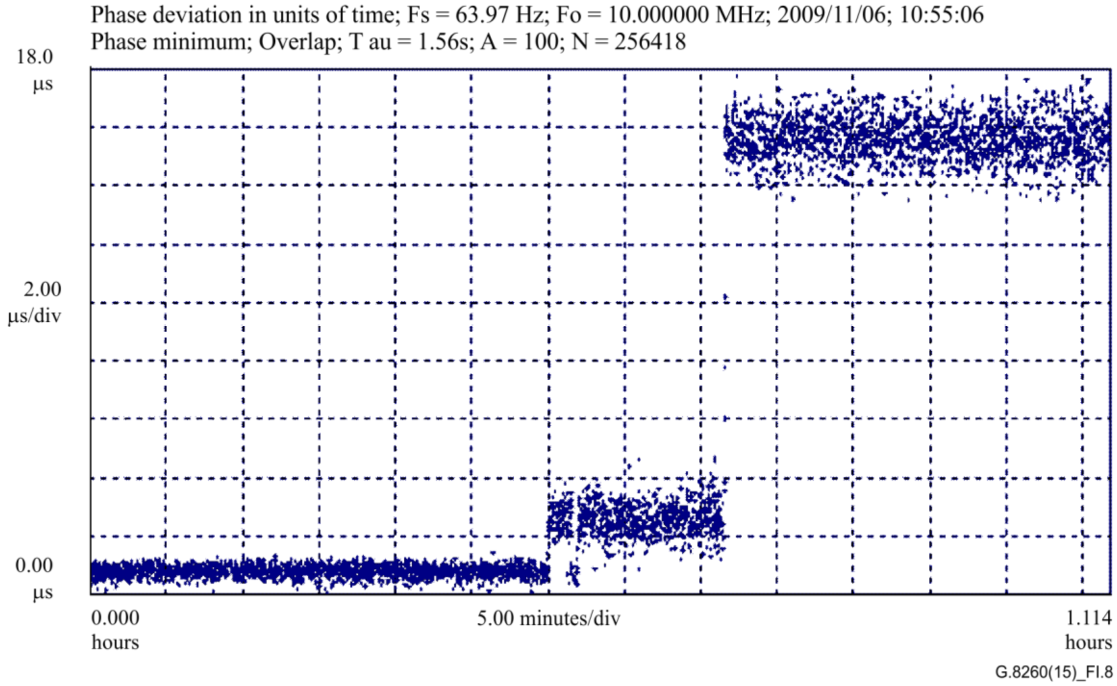

Figure I.8 – Minimum tracking statistic shows three distinct sections

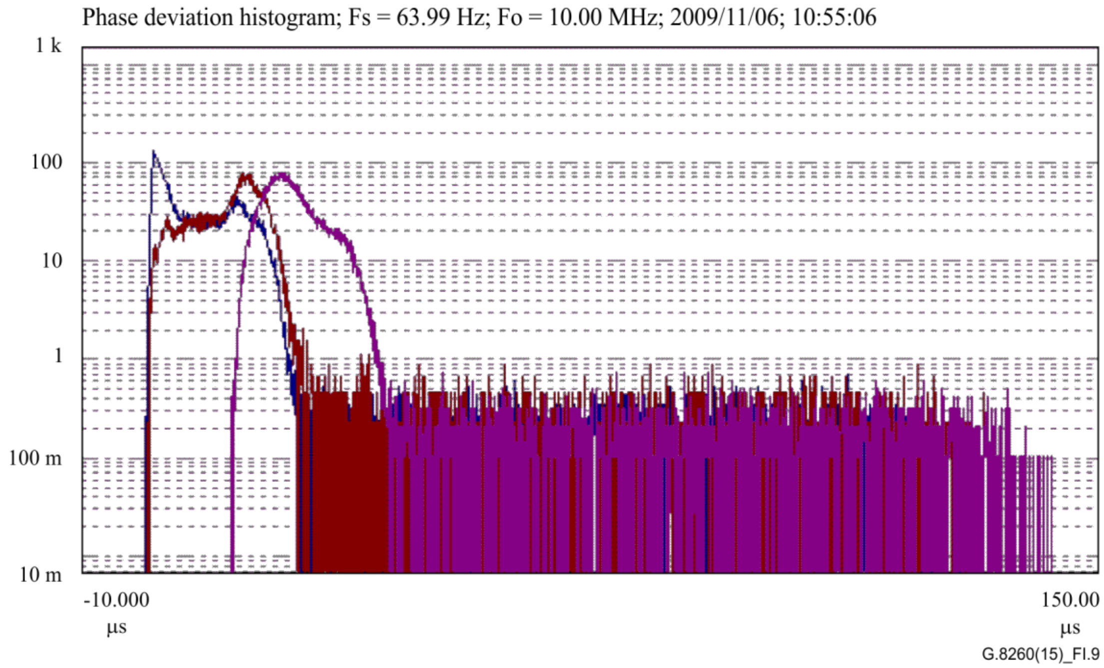

Figure I.9 – Histograms (PDFs) for the three sections

I.3.4 Two-way time error calculation

The two-way time error sequence x_C is calculated from the forward time error sequence x_F and reverse time error sequence x_R according to the following equation:

$$x_C(n\tau_0) = \frac{x_R(n\tau_0) + x_F(n\tau_0)}{2} \quad \text{(I-12a)}$$

where τ_0 is the mean packet spacing.

Figure I.9a shows the combination operation producing the two-way time error.

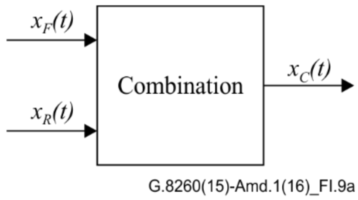

Figure I.9a – Two-way offset

When combined with packet selection, which is performed on the forward and reverse sequences independently, the packet-selected time-error sequence of the forward path, $x_F'(t)$, is combined with the packet-selected time-error sequence of the reverse path, $x_R'(t)$, to create the packet-selected two-way time error sequence $x_C'(t)$.

$$x_C'(n\tau_s) = \frac{x_R'(n\tau_s) + x_F'(n\tau_s)}{2} \qquad \text{(I-12b)}$$

where τ_S is the packet selection window width.

Figure I.10 shows that when the combination operation is preceded by packet selection, the packet-selected two-way time error sequence is produced. This can itself be optionally combined with bandwidth filtering and/or stability metrics. The sequence $x_C'(t)$ is referred to as packet-selected two-way time error (pktSelected2wayTE).

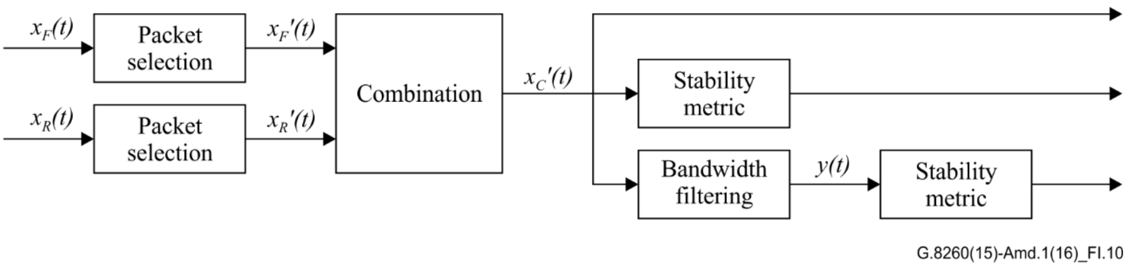

Figure I.10 – Two-way time error including packet selection and filtering

Note that the combination operation could be performed after bandwidth filtering as applied to each packet selection output separately, with the same results.

Subsequently, $x_C'(t)$ for two-way flows (or $x_C(t)$ if there is no packet selection) may be substituted into the various metrics in the same manner as $x'(t)$ for one-way flows. When used in this way, the prefix "2way" denotes the fact that the metric is computed on a two-way flow, e.g., "2wayTDEV", "pktSelected2wayMAFE", or "pktFiltered2wayMTIE".

I.4 PDV metrics estimating the performance of a packet slave clock

Clauses I.4.1 to I.4.4 describe the stability metrics and a few specific associations with packet selection that have been studied for quantifying packet network timing signals.

The metrics are divided into four main classifications:

1) **Metrics estimating phase wander**

   These include TIE and MTIE based metrics, similar to the use of TIE and MTIE to characterize the phase wander of conventional timing signals.

2) **Metrics estimating frequency stability**

These include TDEV based metrics, analysing the stability of the clock over different observation intervals.

3) **Metrics estimating frequency accuracy**

These include metrics based on maximum average frequency error (MAFE) and fractional frequency offset (FFO).

4) **Metrics estimating time error**

I.4.1 Metrics estimating phase wander

I.4.1.1 Maximum average time interval error

Maximum average time interval error (MATIE) and its related metric maximum average frequency error (MAFE) describe maximum phase or frequency deviations over an observation interval. MATIE and MAFE include a noise-averaging function similar to TDEV.

Definition

Two adjacent sliding observation windows are used to analyse the time error of a clock or selected packet delay data. The width of the observation windows (τ) is used as the independent variable (x-axis of the resulting curve) like in TDEV.

The average time error value is computed in the two adjacent windows. The averaging establishes the filtering capability that resembles the one used in TDEV. The unsigned difference between two consecutive windows is determined by subtracting the average of one window from the other and calculating the absolute value.

The sliding window averaging process described above is a low-pass filtering process approximately corresponding to the one applied by a PLL filter to a timing signal. The difference calculation compares the estimation of the phase of the clock output at two time instances, which are a distance of $n\tau_0$ apart, see the MATIE formula (Equation I-13).

The two adjacent sliding windows are swept over the whole time error data and the maximum value is taken to express the worst-case occurrence expected from the data.

For the MATIE analysis of packet data, the same calculation is done for different values of the window size (τ), similar to TDEV.

The function applied to discrete data samples is described in Equation I-14.

MATIE($n\tau_0$) is defined as a specified percentile, β, of the random variable:

$$X_n = \max_{1 \le k \le N-2n+1} \frac{1}{n} \left| \sum_{i=k}^{n+k-1} \left(x_{i+n} - x_i \right) \right| \tag{I-13}$$

for $n = 1, 2, ...,$ integer part ($N/2$)

where x_i is the packet time error sequence (and is a random sequence), $n\tau_0$ is the observation window length, n is the number of samples in the window, τ_0 is the sample interval, N is the number of samples in the data set, and k is incremented to slide the window. MATIE describes the maximum of average time changes between adjacent windows of length $n\tau_0$.

Estimator formula

MATIE($n\tau_0$) may be estimated by:

$$\text{MATIE}(n\tau_0) \cong \max_{1 \le k \le N-2n+1} \frac{1}{n} \left| \sum_{i=k}^{n+k-1} \left(x_{i+n} - x_i \right) \right| \tag{I-14}$$

for n = 1, 2, ..., integer part (N/2)

Equation I-14 gives a point estimate, obtained from measurements (i.e., samples x_i of the packet time error sequence or physical clock signal time error sequence, which represents the data values) over a single measurement period (see Figure II.1 of [ITU-T G.810]). Estimates of MATIE (for specified N, $\tau = n\tau_0$, and β), and their respective degrees of statistical confidence, may be obtained from measured data if measurements are made for multiple measurement periods (see clause II.5 of [ITU-T G.810]).

Usage

The recommended usage for determining the applicability of a network for packet synchronization is to apply the metric to pre-processed delay data, corresponding to the selected subset in Figure I.4. MATIE predicts the largest difference in averaged time interval error that occurs between adjacent averaging windows of width $n\tau_s$:

$$\mathrm{MATIE}(n\tau_s) \cong \max_{1\le k\le N-2n+1} \frac{1}{n}\left|\sum_{i=k}^{n+k-1}(x'_{i+n}-x'_i)\right| \tag{I-15}$$

for n = 1, 2, ..., integer part (N/2)

Figure I.11 shows some example MATIE data. The black curve represents MATIE applied to pre-processed delay data. The red curve depicts pktfilteredMTIE (see clause I.4.1.4) with a 960 s averaging function.

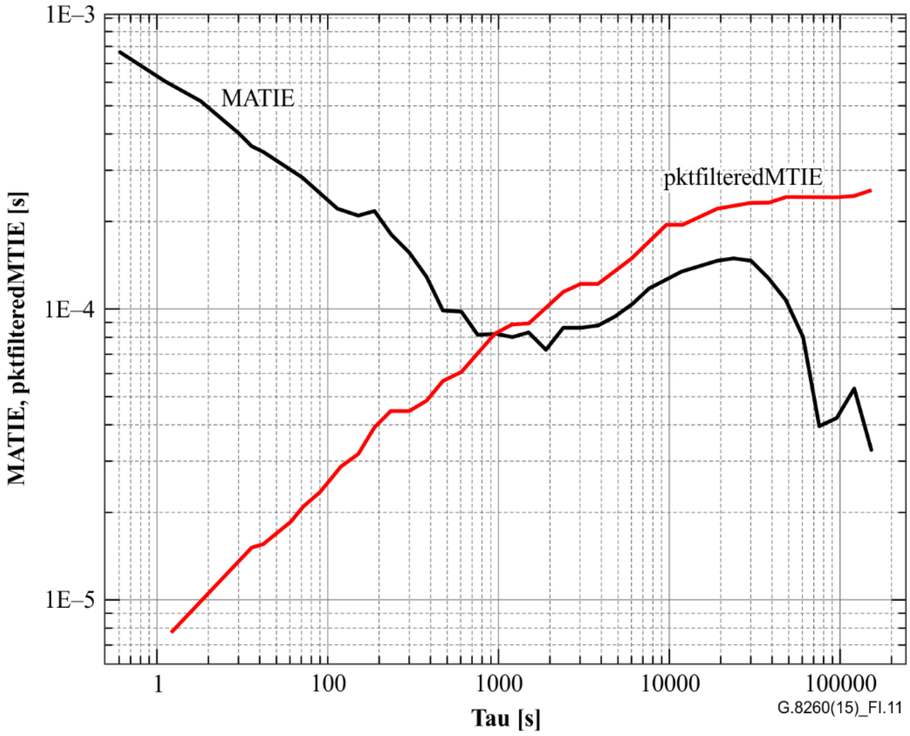

Figure I.11 – Example MATIE and pkfilteredMTIE data

I.4.1.2 minMATIE

The packet selection operation can also be integrated in the MATIE calculation. The definitions and estimator formulas for minMATIE are given as follows:

Definition

minMATIE($n\tau_0$) is defined as a specified percentile, β, of the random variable:

$$U_n = \max_{1\le k\le N-2n+1}\left|x_{\min}(k+n)-x_{\min}(k)\right| \tag{I-16}$$

where $x_{\min}(k)$ is as defined in equation (I-8) for the forward time error sequence, or (I-8a) for the reverse time error sequence, $n\tau_0$ is the observation window length, n is the number of samples in the window, τ_0 is the sample interval, N is the number of samples in the data set, and k is incremented for sliding the window.

Estimator formula

minMATIE($n\tau_0$) may be estimated by:

$$\text{minMATIE}(n\tau_0) \cong \max_{1 \le k \le N-2n+1} \left| x_{\min}(k+n) - x_{\min}(k) \right| \tag{I-17}$$

for n = 1, 2, ..., integer part (N/2)

The above is a point estimate, and is obtained for measurements (i.e., samples x_i of the packet time error sequence, which represent the data values) over a single measurement period (see Figure II.1 of [ITU-T G.810]). Estimates of minMATIE (for specified N, $\tau = n\tau_0$, and β), and their respective degrees of statistical confidence, may be obtained from measured data if measurements are made for multiple measurement periods (see clause II.5 of [ITU-T G.810]).

I.4.1.3 pktfilteredTIE

pktfilteredTIE is the TIE of the filtered-packet time error sequence, substituted into the formula defined in [ITU-T G.810].

$$\text{pktfilteredTIE}(t,\tau) = y(t+\tau) - y(t) \tag{I-18}$$

I.4.1.4 pktfilteredMTIE

pktfilteredMTIE is the MTIE of the filtered-packet time error sequence, obtained from the appropriate formula given in [ITU-T G.810] for the definition or estimator.

Definition

pktfilteredMTIE(τ) is defined as a specified percentile, β, of the random variable:

$$Y = \max_{0 \le t_0 \le T-\tau} \left(\max_{t_0 \le t \le t_0+\tau} [y(t)] - \min_{t_0 \le t \le t_0+\tau} [y(t)] \right), \tag{I-19}$$

where T is the measurement interval and τ is the observation interval.

Estimator formula

pktfilteredMTIE($n\tau_0$) may be estimated by:

$$\text{pktfilteredMTIE}(n\tau_0) \cong \max_{1 \le k \le N-n} \left[\max_{k \le i \le k+n} y_i - \min_{k \le i \le k+n} y_i \right], \quad n = 1,2,\ldots,N-1 \tag{I-20}$$

The above is a point estimate, and is obtained for measurements over a single measurement period (see Figure II.1 of [ITU-T G.810]). Estimates of pktfilteredMTIE (for specified T, $\tau = n\tau_0$ and β), and their respective degrees of statistical confidence, may be obtained from measured data, if measurements are made for multiple measurement periods (see clause II.5 of [ITU-T G.810]).

I.4.2 Metrics estimating frequency stability

I.4.2.1 TDEV-based metrics

TDEV has been specified in [ITU-T G.810] and used in other Recommendations to specify network wander limits for physical timing signals. TDEV is also applicable to packet timing data. In relation to packet timing data, TDEV can be applied to pre-processed PDV data or integrated into the calculation. The case where TDEV is applied to pre-processed PDV data, which can be referred to as pktselectedTDEV, is depicted by Figure I.4 with TDEV as the stability metric.

The integrated methods based on TDEV include minTDEV, percentileTDEV, and bandTDEV. The minTDEV and percentileTDEV metrics focus the packet selection on the minimum packet delay floor and the more general bandTDEV metric can select packet delays from any region, for example, the floor, just above the floor, in the middle, or the ceiling. The integrated methods can be applied for MATIE and MAFE metrics described further below but are described in depth only in the TDEV clause.

Like the TDEV metric, the TDEV-based packet measurement metrics study the noise processes in the packet measurement data – white phase modulation (PM), flicker PM, random walk PM, flicker frequency modulation (FM), and random walk FM. With the incorporation of packet selection it is often possible that one or more of these noise processes can be reduced as compared to analysis incorporating no selection.

I.4.2.1.1 MinTDEV

Definition

The minTDEV operator has been defined based on the TDEV metric. The TDEV metric is shown below in equation (I-21):

$$\sigma_x(\tau) = \mathrm{TDEV}(\tau) = \sqrt{\frac{1}{6n^2}\left\langle\left[\sum_{i=1}^{n}(x_{i+2n} - 2x_{i+n} + x_i)\right]^2\right\rangle} \qquad \text{(I-21)}$$

The TDEV operator is based on the mean of the sample window (Equation I-22):

$$x_{mean}(i) = \frac{1}{n}\sum_{j=0}^{n-1} x_{j+i} \qquad \text{(I-22)}$$

Compared with the TDEV operator, in the minTDEV operation the mean of the sample window is replaced by $x_{\min}(i)$ as defined in Equation I-8 for the forward time error sequence, or I-8a for the reverse time error sequence. Substituting $x_{\min}(i)$ back into original TDEV definition yields the definition of minTDEV(τ) (with $\tau = n\tau_0$):

$$\sigma_{x\_\min}(n\tau_0) = \mathrm{minTDEV}(n\tau_0) = \sqrt{\frac{1}{6}\left\langle\left[x_{\min}(i+2n) - 2x_{\min}(i+n) + x_{\min}(i)\right]^2\right\rangle}, \qquad \text{(I-24)}$$

$$\text{for } n = 1, 2, ..., \text{ integer part } \left(\frac{N}{3}\right)$$

where the angle brackets denote ensemble average.

Estimator formula

minTDEV($n\tau_0$) may be estimated by:

$$\mathrm{minTDEV}(n\tau_0) \cong \sqrt{\frac{1}{6(N-3n+1)}\sum_{i=1}^{N-3n+1}\left[x_{\min}(i+2n) - 2x_{\min}(i+n) + x_{\min}(i)\right]^2} \qquad \text{(I-25)}$$

$$\text{for } n = 1, 2, ..., \text{ integer part } \left(\frac{N}{3}\right)$$

Usage

The minTDEV operator has been indicated as a useful tool in combination with packet networks that exhibit a PDV behavior, where it is possible to identify a suitable set of packets with packet delay variation close to a minimum delay.

In fact, these packets are less impacted by the queuing delays, and therefore are more representative of the original timing. Because of its definition, the minTDEV may not fully address all network scenarios (e.g., those with two-sided PDV distributions for which minimum selection can show large variations and hence increased TDEV noise) and further study is needed.

Pros and cons

The minTDEV calculation gives information on network packet delay noise processes but is not suitable for frequency offset characterization.

Like TDEV, minTDEV is sensitive to systematic effects which could mask noise components.

Unlike TDEV, minTDEV is sensitive to a small number of outliers (low-lying in this case).

The definition of the precise aspects that create the potential sensitivities listed above and the subsequent method of handling these when applying this metric are for further study.

I.4.2.1.2 PercentileTDEV

Definition and estimator formula

The percentileTDEV calculation is a special case of bandTDEV where the lower index *a* is assigned to 0 (see the bandTDEV definition below). Therefore, the definition and estimator formula are given by the definition and estimator formula, respectively, for bandTDEV (see clause I.4.2.1.3) with $a = 0$. Like the minTDEV metric, percentileTDEV focuses on the minimum packet delay floor. Instead of selecting a single minimum point, a (typically) small set of points at the floor are averaged together.

Usage

The percentileTDEV metric is applied much like the minTDEV metric. See clause I.4.2.1.1 on minTDEV usage. The percentile TDEV metric has the advantage that in some circumstances, noise is reduced when a number of floor packet delay measurements are selected and averaged together as opposed to the selection of a single point (see Figure I.12).

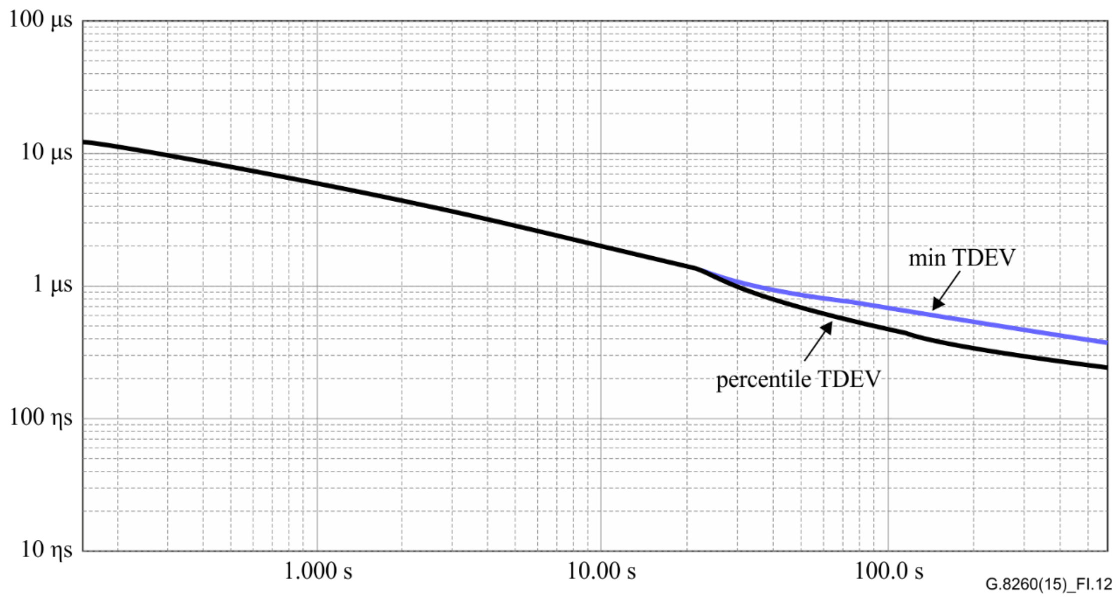

Figure I.12 – minTDEV vs. percentileTDEV (1%)

Pros and cons

Like minTDEV, percentileTDEV gives information on network packet delay noise processes but is not optimal for frequency offset characterization.

Like TDEV and minTDEV, percentileTDEV is sensitive to systematic effects which could mask noise components.

Unlike TDEV and like minTDEV, percentileTDEV is sensitive to a small number of low-lying outliers (though less sensitive than minTDEV).

An additional parameter, the percentile index, must be selected for percentileTDEV.

The definition of the precise aspects that create the potential sensitivities listed above and the subsequent method of handling these when applying this metric are for further study.

I.4.2.1.3 BandTDEV

Definition

BandTDEV represents the TDEV calculation where the band average selection operator (see clause I.3.2.3) is used to replace the mean of the sample window. The selected band is defined by two percentile values representing the upper and lower selection bounds, as shown in Equation I-9.

bandTDEV can then be defined as:

$$\sigma_{x\_band}(\tau) = \mathrm{bandTDEV}(\tau) = \sqrt{\tfrac{1}{6}\left\langle \left[x'_{band\_avg}(i+2n) - 2x'_{band\_avg}(i+n) + x'_{band\_avg}(i) \right]^2 \right\rangle} \qquad \text{(I-27)}$$

where the angle brackets denote ensemble average.

Estimator formula

bandTDEV($n\tau_0$) may be estimated by:

$$\mathrm{bandTDEV}(n\tau_0) \cong \sqrt{\frac{1}{6(N-3n+1)} \sum_{i=1}^{N-3n+1} \left[x'_{band\_avg}(i+2n) - 2x'_{band\_avg}(i+n) + x'_{band\_avg}(i) \right]^2} \qquad \text{(I-28)}$$

$$\text{for } n = 1, 2, \ldots, \text{integer part } \left(\frac{N}{3}\right)$$

Usage

The bandTDEV calculation has the flexibility, in comparison to minTDEV and percentileTDEV, of being able to select a region of packet delay values away from the floor. Thus, if the population of packet delay values at the floor is noisier than the population immediately above, bandTDEV indices could be selected to focus analysis on that region.

Some of the comments on minTDEV usage apply here, but bandTDEV can apply effectively to distributions other than one-sided distributions slanted towards the packet with the minimum delay. It is particularly effective for packet delay distributions with a well-populated mode somewhere in the packet delay distribution.

Pros and cons

Like minTDEV and percentileTDEV, bandTDEV gives information on network packet delay noise processes but is not optimal for frequency offset characterization.

Like TDEV, minTDEV, and percentileTDEV, bandTDEV is sensitive to systematic effects which could mask noise components.

The definition of the precise aspects that create the potential sensitivities listed above and the subsequent method of handling these when applying this metric are for further study.

I.4.2.1.4 ClusterTDEV

Definition

To define clusterTDEV, it is first necessary to represent the sorted phase data. Let x' represent this sorted phase sequence from minimum to maximum over the range $i \le j \le i + n - 1$. Next it is necessary to represent the cluster type that determines the cluster anchor $a(n)$ and the cluster range δ. Let a and b represent indices for the packets that fit within cluster range δ. The averaging is then applied to the x' variable for the cluster range δ. The number of averaged points is m, where $m = b - a + 1$ for that cluster.

$$x'_{\text{cluster}}(i) = \frac{1}{m}\sum_{j=a}^{b} x'_{j+i} \tag{I-29}$$

For clusterTDEV the average of the values in the cluster range as per Equation I-29 is substituted for the mean value in the defining equation for TDEV. ClusterTDEV($n\tau_0$) can then be defined as:

$$\sigma_{x\_\text{cluster}}(\tau) = \text{clusterTDEV}(\tau) = \sqrt{\frac{1}{6}\left\langle \left[x'_{\text{cluster}}(i+2n) - 2x'_{\text{cluster}}(i+n) + x'_{\text{cluster}}(i)\right]^2 \right\rangle} \tag{I-30}$$

where the angle brackets denote ensemble average.

Estimator formula

clusterTDEV($n\tau_0$) may be estimated by:

$$\text{clusterTDEV}(n\tau_0) \cong \sqrt{\frac{1}{6(N-3n+1)}\sum_{j=1}^{N-3n+1}[x'_{\text{cluster}}(i+2n) - 2x'_{\text{cluster}}(i+n) + x'_{\text{cluster}}(i)]^2}, \tag{I-31}$$

$$\text{for } n = 1, 2, ..., \text{ integer part } \left(\frac{N}{3}\right)$$

Usage

The clusterTDEV calculation has the flexibility of being able to select a region of packet delay values away from the floor. Thus, if the population of packet delay values at the floor is noisier than the population immediately above, clusterTDEV indices could be selected to focus analysis on that region. Generally speaking, clusterTDEV provides a quantitative measure of stability of transit delays that are in a pre-determined band based on a phase and/or time range centred at a value that is determined by a chosen selection rule.

Some of the comments on minTDEV usage apply to clusterTDEV as well, but clusterTDEV can apply effectively to distributions other than one-sided distributions slanted towards the minimum delayed packet. It is particularly effective for packet delay distributions with a well-populated mode somewhere in the packet delay distribution.

Pros and cons

Like minTDEV, clusterTDEV gives information on network packet delay noise processes but is not suitable for frequency offset characterization.

Like TDEV and minTDEV, clusterTDEV is sensitive to systematic effects which could mask noise components.

Unlike TDEV and like minTDEV, clusterTDEV is sensitive to frequency offsets. Frequency offsets may be more difficult to ascertain precisely when neither a well-populated floor nor ceiling exists.

Two additional parameters, the cluster range and the cluster rule, must be selected for clusterTDEV.

The definition of the precise aspects that create the potential sensitivities listed above and the subsequent method of handling these when applying this metric are for further study.

It may be helpful to use the representation clusterTDEV(τ, δ, *clusterType*) where τ is the observation interval, δ the cluster range, and the *clusterType* provides the rule used to generate the *anchor value*. Generally the rule is available from context and in that case need not be included in the representation.

For example:

$$\mathrm{minTDEV}(\tau) = \mathrm{clusterTDEV}(\tau, 0, clusterType_{\min}) \tag{I-32}$$

I.4.2.1.5 pktfilteredTDEV

pktfilteredTDEV is the TDEV of the filtered-packet time error sequence, obtained from the appropriate formula given in [ITU-T G.810] for the definition or estimator.

Definition

pktfilteredTDEV($n\tau_0$) is defined as:

$$\mathrm{pktfilteredTDEV}(n\tau_0) = \sqrt{\frac{1}{6n^2}\left\langle\left[\sum_{i=1}^{n}\left(y_{i+2n} - 2y_{i+n} + y_i\right)\right]^2\right\rangle}, \tag{I-33}$$

where the angle brackets denote an ensemble average.

Estimator formula

pktfilteredTDEV($n\tau_0$) may be estimated by:

$$\mathrm{pktfilteredTDEV}(n\tau_0) \cong \sqrt{\frac{1}{6n^2(N-3n+1)}\sum_{j=1}^{N-3n+1}\left[\sum_{i=j}^{n+j-1}\left(y_{i+2n} - 2y_{i+n} + y_i\right)\right]^2} \tag{I-34}$$

for $n = 1, 2, ...,$ integer part $\left(\frac{N}{3}\right)$

I.4.3 Metrics estimating frequency accuracy

I.4.3.1 Maximum average frequency error

Definition

There is a simple relationship between MAFE and the MATIE metric defined above.

$$\mathrm{MAFE}(n\tau_0) = \frac{\mathrm{MATIE}(n\tau_0)}{n\tau_0} \tag{I-35}$$

Thus, MAFE($n\tau_0$) is defined as a specified percentile, β, of the random variable:

$$Z_n = \frac{\max\limits_{1\le k\le N-2n+1} \frac{1}{n}\left|\sum_{i=k}^{n+k-1}\left(x_{i+n} - x_i\right)\right|}{n\tau_0} \tag{I-36}$$

for $n = 1, 2, ...,$ integer part (N/2)

where x_i is the packet time error sequence (and is a random sequence), $n\tau_0$ is the observation window length, n is the number of samples in the window, τ_0 is the sample interval, N is the number of samples in the data set, and k is incremented for sliding the window. MAFE is a dimensionless, normalized frequency ($\Delta f/f$).

Estimator formula

MAFE($n\tau_0$) may be estimated by:

$$\mathrm{MAFE}(n\tau_0) \cong \frac{\max\limits_{1 \le k \le N-2n+1} \frac{1}{n} \left| \sum_{i=k}^{n+k-1} (x_{i+n} - x_i) \right|}{n\tau_0} \qquad \text{(I-37)}$$

for $n = 1, 2, ...,$ integer part ($N/2$)

The above is a point estimate, and is obtained for measurements (i.e., samples x_i of the packet time error sequence or physical clock signal time error sequence, which represents the data values) over a single measurement period (see Figure II.1 of [ITU-T G.810]). Estimates of MAFE (for specified N, $\tau = n\tau_0$, and β), and their respective degrees of statistical confidence, may be obtained from measured data if measurements are made for multiple measurement periods (see clause II.5 of [ITU-T G.810]).

Usage

When applied to the time error sequence of a physical clock signal data corresponding to the clock output in Figure I.1, at small τ values where the MAFE calculation does not do any further filtering to the clock signal, MAFE expresses the peak frequency error of the clock, see Figure I.13. At larger τ values where MAFE represents narrower bandwidth than in the clock servo producing the signal, MAFE presents how the maximum frequency error could be reduced by further averaging of the clock signal.

The recommended usage for determining the applicability of a network for packet synchronization is to apply the metric to pre-processed delay data, corresponding to the selected subset in Figure I.4, MAFE predicts the maximum frequency error calculated from the largest difference in averaged time interval error observed between adjacent averaging windows of width $n\tau_s$, see Figure I.13.

$$\mathrm{MAFE}(n\tau_s) \cong \frac{\max\limits_{1 \le k \le N-2n+1} \frac{1}{n} \left| \sum_{i=k}^{n+k-1} (x'_{i+n} - x'_i) \right|}{n\tau_s} \qquad \text{(I-38)}$$

for $n = 1, 2, ...,$ integer part ($N/2$)

Figure I.13 shows an example of the effect of MAFE applied to pre-processed delay data and to the time error data of one particular physical clock that uses an averaging period of the order of 1 000 s. The definition of averaging period is for further study.

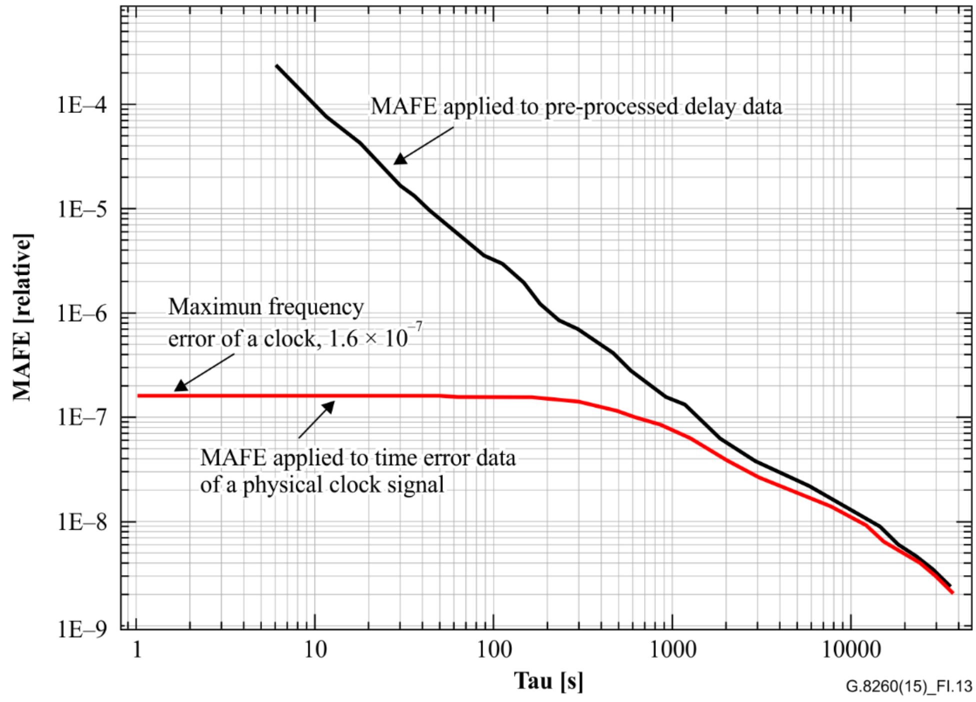

Figure I.13 – MAFE applied to pre-processed delay data and to time error data of physical clock

I.4.3.1.1 Pros and cons of MAFE and MATIE

MAFE is well suited to the characterization of frequency error.

The behavior of MAFE and MATIE as a function of the nature of the time error is examined below. Note that MAFE and MATIE do not converge for noise types of higher order than WFM, e.g., FFM and RWFM.

Like the minTDEV and percentileTDEV metrics, MAFE and MATIE with floor-based selection is sensitive to a small number of low-lying outliers.

The definition of the precise aspects that create the potential sensitivities listed above and the subsequent method of handling these when applying this metric are for further study.

Properties of MAFE and MATIE

Suppose that the time error sequence for which the MAFE (or MATIE) is computed is described by the sequence $\{x(n\tau_0)\}$ or (to simplify notation), $\{x(n)\}$. The coefficient μ below is the *constant time error*. Consider the cases where the dominant component of the time error is:

a. Simple frequency offset

In this case the time error can be written as (frequency offset = b):

$$x(k) = \mu + b \cdot k \cdot \tau_0 \tag{I-39}$$

and it can be shown that:

$$\mathrm{MATIE}(\tau = n \cdot \tau_0) = b \cdot \tau_0 \cdot n \tag{I-40}$$

Note that the frequency offset is related to MATIE in this case via the relationship:

$$b = \frac{\mathrm{MATIE}(\tau = n \cdot \tau_0)}{n \cdot \tau_0} = \mathrm{MAFE}(\tau = n \cdot \tau_0) \tag{I-41}$$

That is, when the time error is dominated by a frequency offset, the value of MAFE will be a constant, equal to the frequency offset (fractional frequency units).

b. White phase noise

In this case:

$$x(k) = \mu + \varphi_{\mathrm{WPM}}(k) \tag{I-42}$$

where $\varphi_{\mathrm{WPM}}(k)$ is a white noise sequence with power (variance) σ_φ^2. It can be shown that whereas a closed form expression for MATIE does not exist it can be approximated by:

$$\mathrm{MATIE}(\tau = n \cdot \tau_0) \approx \left(4 \cdot \sqrt{2 \cdot \tau_0} \cdot \sigma_\varphi\right) \cdot \left(\frac{1}{\sqrt{n \cdot \tau_0}}\right) \tag{I-43}$$

Equation I-43 assumes that the noise distribution is approximately Gaussian, based on the central limit theorem. The maximum value is approximated as four times the standard deviation. This is represented by the factor of 4 in Equation I-43. For a Gaussian distribution, four standard deviations correspond to the upper 6.33×10^{-5} quantile (2-sided).

In terms of MAFE:

$$\mathrm{MAFE}(\tau = n \cdot \tau_0) = \frac{\mathrm{MATIE}(\tau)}{\tau} \approx \left(4 \cdot \sigma_\varphi \cdot \sqrt{2 \cdot \tau_0}\right) \cdot \tau^{\left(-\frac{3}{2}\right)} \tag{I-44}$$

Thus, the impact of white PM on values of MATIE/MAFE becomes less important for large values of the observation interval τ. When plotted on a log-log scale, the graph of MATIE appears as a straight line of slope –0.5 (–1.5 for MAFE).

c. Flicker phase noise

In this case:

$$x(k) = \mu + \varphi_{\mathrm{FPM}}(k) \tag{I-45}$$

Generally speaking, flicker lies between white noise and random walk. If C is a constant that is related to the strength of the flicker random process, then we can write (note that this is not a formal proof.):

$$\begin{aligned} \mathrm{MATIE}(\tau = n \cdot \tau_0) &\approx 4 \cdot C \\ \mathrm{MAFE}(\tau = n \cdot \tau_0) &\approx 4 \cdot C \cdot \tau^{(-1)} \end{aligned} \tag{I-46}$$

The factor of 4 represents four standard deviations as described in the white phase noise description above.

d. Random walk phase noise

In this case:

$$x(k) = \mu + \varphi_{\mathrm{WFM}}(k) \tag{I-47}$$

and it can be shown that for large n, we can write

$$\begin{aligned} \mathrm{MATIE}(\tau = n \cdot \tau_0) &\approx 4 \cdot \sigma_\varepsilon \cdot \left(\sqrt{\frac{2}{3 \cdot \tau_0}}\right) \cdot (n \cdot \tau_0)^{0.5} \\ \mathrm{MAFE}(\tau = n \cdot \tau_0) &\approx 4 \cdot \sigma_\varepsilon \cdot \left(\sqrt{\frac{2}{3 \cdot \tau_0}}\right) \cdot (n \cdot \tau_0)^{-0.5} \end{aligned} \tag{I-48}$$

The factor of 4 represents four standard deviations as described in the white phase noise description above.

Thus the impact of random walk PM on values of MATIE/MAFE becomes very important for large values of the observation interval τ. When plotted on a log-log scale, the graph of MATIE appears as a straight line of slope +0.5 (–0.5 for MAFE). The term σ_ε represents the standard deviation of the white-noise sequence underlying the generation of the random walk sequence.

I.4.3.2 minMAFE

The packet selection operation can also be integrated in the MAFE calculation. The definitions and estimator formulas for minMAFE are given as follows:

Definition

minMAFE($n\tau_0$) is defined as a specified percentile, β, of the random variable:

$$V_n = \frac{\max\limits_{1 \le k \le N-2n+1} \left| x_{\min}(k+n) - x_{\min}(k) \right|}{n\tau_0} \tag{I-49}$$

for n = 1, 2, ..., integer part (N/2)

where $x_{\min}(k) = \min[x_j] \text{ for } (k \le j \le k+n-1)$, where x_i is the packet time error sequence (and is a random sequence), $n\tau_0$ is the observation window length, n is the number of samples in the window, τ_0 is the sample interval, N is the number of samples in the data set, and k is incremented for sliding the window.

Estimator formula

minMAFE($n\tau_0$) may be estimated by:

$$\text{minMAFE}(n\tau_0) \cong \frac{\text{minMATIE}(n\tau_0)}{n\tau_0} = \frac{\max\limits_{1 \le k \le N-2n+1} \left| x_{\min}(k+n) - x_{\min}(k) \right|}{n\tau_0}, \tag{I-50}$$

for n = 1, 2, ... integer part (N/2)

The above is a point estimate, and is obtained for measurements (i.e., samples x_i of the packet time error sequence, which represent the data values) over a single measurement period (see Figure II.1 of [ITU-T G.810]. Estimates of minMATIE (for specified N, $\tau = n\tau_0$, and β), and their respective degrees of statistical confidence, may be obtained from measured data, if measurements are made for multiple measurement periods (see clause II.5 of [ITU-T G.810]).

I.4.3.3 pktfilteredFFO

pktfilteredFFO is the fractional frequency offset of the filtered-packet time error sequence, $y(t)$, substituted into the formula defined in [b-GR-1244-CORE]. Refer to clauses 4.5.2 and I.2 of [ITU-T G.810] for the definition and a description of FFO.

Estimator formula

pktfilteredFFO(t; $M\tau_s$), a dimensionless quantity, may be estimated by:

$$\text{pktfilteredFFO}(t;\, M\tau_s) \cong \frac{6}{M\tau_s} \sum_{i=1}^{M} y_{i+m-1} \left(\frac{2i}{(M^2-1)} - \frac{1}{M-1} \right), \; m = 1,2,3, \ldots, N-M+1 \tag{I-51}$$

where:

$t = m \cdot \tau_s$

τ_s	*s* the sampling period in seconds of the time samples after preselection and filtering.
M	is the number of samples in the calculation interval.
$M \cdot \tau_s$	is the calculation interval.
N	is the total number of samples of the time data.
y_m	are samples of time data in units of seconds, after preselection and filtering.

I.4.4 Metrics estimating time error

I.4.4.1 Packet-selected two-way time error (pktSelected2wayTE)

The packet-selected two-way time error sequence, $x_C'(t)$ in Figure I.11 of clause I.3.4, is a sequence that can be used directly as a metric estimating time error. It is referred to as pktSelected2wayTE. A graphical representation of the packet-selected two-way time error sequence is shown in Figure I.13a:

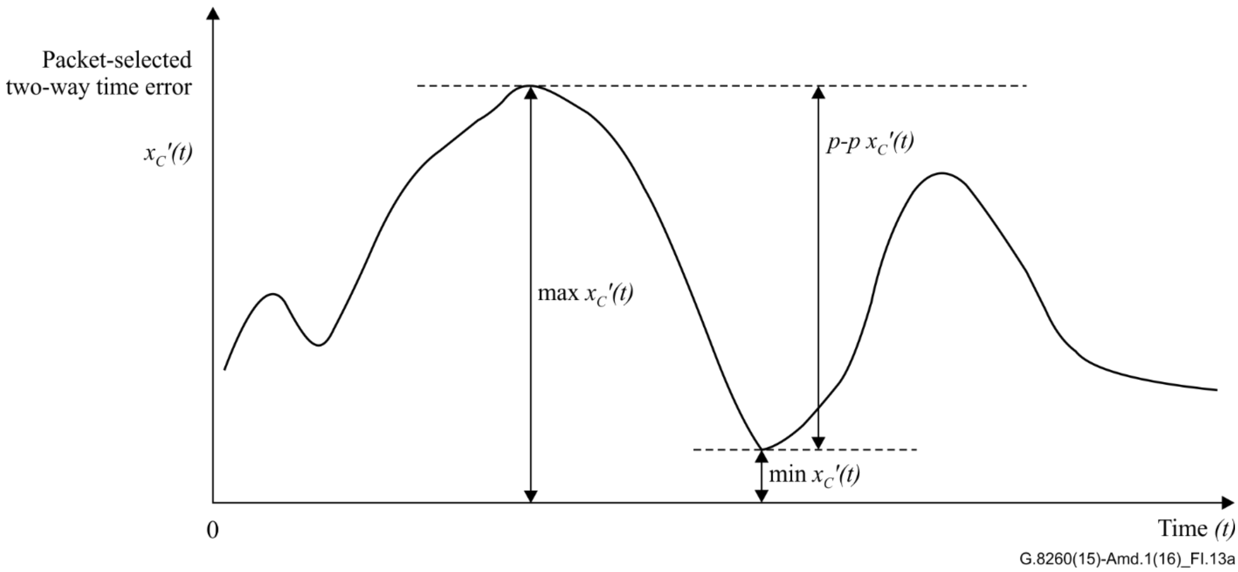

Figure I.13a – Packet-selected two-way time error sequence and derived values

Peak-to-peak packet-selected two-way time error:

$$\text{peak-to-peak(pktSelected2wayTE)} = \max(\text{pktSelected2wayTE}) - \min(\text{pktSelected2wayTE}) \quad \text{(I-51a)}$$

Maximum absolute packet-selected two-way time error:

$$\max|\text{pktSelected2wayTE}| = \max(\ |\max(\text{pktSelected2wayTE})|, |\min(\text{pktSelected2wayTE})|\ ) \quad \text{(I-51b)}$$

I.4.4.2 Packet filtered two-way time error (pktFiltered2wayTE)

The packet-filtered two-way time error sequence, $y(t)$ in Figure I.11 of clause I.3.4, is a sequence that can be used directly as a metric estimating time error. It is referred to as pktFiltered2wayTE.

Maximum absolute packet-filtered two-way time error:

$$\max|\text{pktFiltered2wayTE}| = \max(\ |\max(\text{pktFiltered2wayTE})|, |\min(\text{pktFiltered2wayTE})|\ ) \quad \text{(I-51c)}$$

I.4.4.3 Maximum/Minimum/Peak-to-peak average time error (maxATE, minATE, ppATE)

Definitions

For estimating how a time clock could further filter the noise in the two-way time error sequence, an averaging function can be slid over the data in a similar manner as in the bandwidth filtering function (clause I.3.1.2) used for pktTIE (clause I.4.1.3) and other metrics.

maxATE($n\tau_S$) ("Maximum Average Time Error") is defined as a specified percentile, β, of the random variable:

$$X_n^{\max} = \max_{1 \le k \le N-n+1} \frac{1}{n} \sum_{i=k}^{n+k-1} \left(x_C'(i)\right) \tag{I-52}$$

for $n = 1, 2, ..., N$

where $x_C'(i)$ is the packet-selected two-way time error (which represents time error and is a random sequence), $n\tau_S$ is the observation window length, n is the number of packet selection windows in the observation window and consequently the number of pre-processed samples in the observation window, τ_S is the packet selection window length and consequently the time interval between delay samples after the pre-processing step of packet selection, N is the total number of pre-processed samples, and k is incremented for sliding the observation window. maxATE describes the maximum value of average packet-selected two-way time error over an observation interval of length $n\tau_S$.

Similarly, minATE($n\tau_S$) ("Minimum Average Time Error") is defined as a specified percentile, β, of the random variable:

$$X_n^{\min} = \min_{1 \le k \le N-n+1} \frac{1}{n} \sum_{i=k}^{n+k-1} \left(x_C'(i)\right) \tag{I-53}$$

for $n = 1, 2, ..., N$

where the variables are defined as above. minATE describes the minimum value of average packet-selected two-way time error over an observation interval of length $n\tau_S$.

Finally, ppATE($n\tau_S$) ("Peak-to-peak Average Time Error") is defined as a specified percentile, β, of the random variable:

$$X_n^{pp} = X_n^{\max} - X_n^{\min} = \max_{1 \le k \le N-n+1} \frac{1}{n} \sum_{i=k}^{n+k-1} \left(x_C'(i)\right) - \min_{1 \le k \le N-n+1} \frac{1}{n} \sum_{i=k}^{n+k-1} \left(x_C'(i)\right) \tag{I-54}$$

for $n = 1, 2, ..., N$

where the variables are defined as above. ppATE describes the peak-to-peak value of average packet-selected two-way time error over an observation interval of length $n\tau_S$.

Estimator formulas

For calculating maxATE the sliding window size is varied by sequencing n and determining the maximum two-way time error for each value of n, thus creating maxATE as a function of sliding averaging window width. In a similar fashion, minATE is calculated by determining the minimum two-way time error as a function of sliding averaging window width. ppATE is calculated through a point-by-point subtraction of maxATE minus minATE elements.

$$\text{maxATE}\left(n\tau_S\right) \cong \max_{1 \le k \le N-n+1} \frac{1}{n} \sum_{i=k}^{n+k-1} \left(x_C'(i)\right), \tag{I-55}$$

$$\text{minATE}\left(n\tau_S\right) \cong \min_{1 \le k \le N-n+1} \frac{1}{n} \sum_{i=k}^{n+k-1} \left(x_C'(i)\right), \tag{I-56}$$

$$\text{ppATE}\left(n\tau_S\right) = \text{maxATE}\left(n\tau_S\right) - \text{minATE}\left(n\tau_S\right) \tag{I-57}$$

for $n = 1, 2, ..., N$

where τ_S is the time interval between delay samples after packet selection. Thus τ_S is the same as the selection window size. N is the total number of samples and $x_C'(i)$ is the packet-selected two-way time error after packet selection. The resulting maxATE and minATE curves are shown in Figure I.14 along with the value of ppATE for $n = 1$.

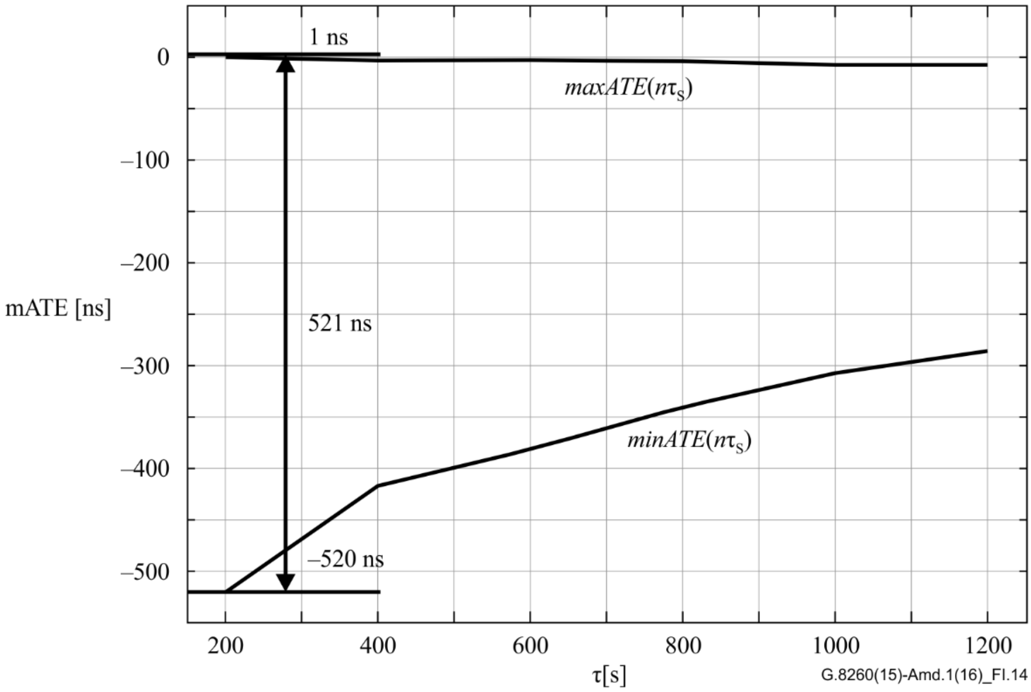

Figure I.14 – Example of average time error calculation

I.5 PDV metrics studying floor delay packet population

The objective of this category of PDV metrics is to study the population of timing packets within a certain fixed cluster range starting at the observed floor delay. The population of timing packets can then be compared with acceptance or rejection thresholds. The main idea here is to ensure that at least a minimum number of packets, or alternately a minimum percentage of packets, always remains within the specified fixed cluster range starting at the observed floor delay.

As an example, consider Figure I.15. The packet delay values are shown as a function of time. Some packets arrive within a certain range of the smallest observed delay (those below the red line) and others arrive outside that range. In each window interval, those packets arriving within the range are counted. This count is compared against an acceptance criterion for each window interval. If all window intervals meet the acceptance criterion, then the network has met the PDV network limit.

The windows depicted in Figure I.15 are shown as non-overlapping, but contiguous. This is also referred to as a "jumping window" approach. The "sliding window" approach considers windows that are shifted by 1 packet (sample). Intermediate approaches consider different levels of overlap, where the degree of overlap is known as the step size. Thus, overlapping windows include both sliding windows and step-overlapping windows. Using sliding windows detects all non-stationary and short transient failure events. Implementations may choose to use a jumping window or an overlapping window approach (including both sliding and step-overlapping).

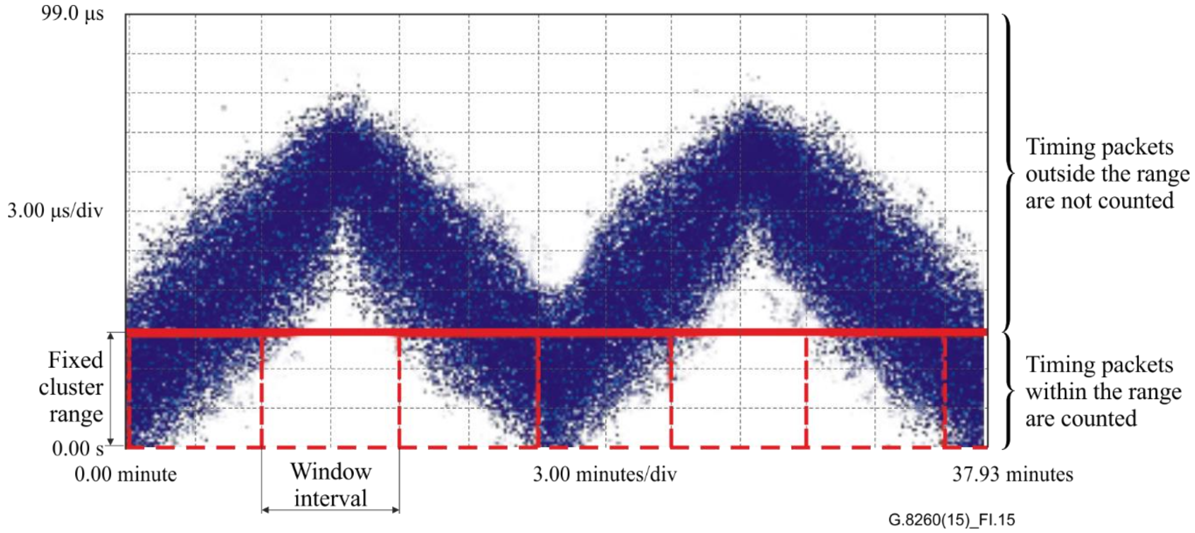

Figure I.15 – Example of PDV metric studying the population of packets within a fixed cluster range starting at the observed floor delay

PDV limits specified in terms of these metrics are considered as met if at least *m* packets, or alternately at least *p*% of packets, are observed for any window interval of *t* s within a fixed cluster range starting at the observed floor delay and having a size δ. If fewer than *m* packets are observed, or alternately less than *p*% of packets, then the PDV limit is considered as not met.

This process can be described in the following way:

Let $x[i]$ represent the measured latency of timing packet *i*, where $0 \le i < N$. That is, there are *N* packets in the measurement data set. Let the nominal time between timing packets be represented by τ_P. Let δ represent the cluster range and let *W* represent the window interval in units of time, which can also be expressed as *K* samples, where $K = W/\tau_P$. *K* represents the (nominal) number of packets transmitted in the window interval.

NOTE – It is assumed that the packet rate of the timing flow is nominally constant. The case for a variable rate of packet transmission is for further study.

Define the minimum observed delay as:

$$d_{\min} = \min_{0 \le i < N} x[i] \tag{I-59}$$

The observed $d_{\min}$ given by Equation I-59 is an estimate of the absolute minimum latency that a packet may experience. If a better estimate of the absolute minimum latency is available, for example from previous measurement data, that alternate value may be used. In all cases, Equations I-60 to I-65 are valid for choices of minimum delay less than or equal to the observed $d_{\min}$.

Then, define the indicator function which performs floor packet selection:

$$\phi_F(i,\delta) = \begin{cases} 1; & \text{if } \; x[i] \le d_{\min} + \delta \\ 0; & \text{otherwise} \end{cases} \quad \text{for } 0 \le i < N \tag{I-60}$$

Note that Equation I-60 assumes that packet delay is always greater than $d_{\min}$.

The convention followed in Equations I-61, I-62 and I-63 is that sample index *n* is associated with the end of the window. That is, the floor packet metrics are based on complete windows and consequently values of *n* less than $(K-1)$ are not defined.

Then define the *Floor Packet Count* (FPC) sequence with parameters n, W ($W = K{\cdot}\tau_P$) and δ:

$$\mathrm{FPC}(n,W,\delta) \;=\; \sum_{j=n-(K-1)}^{n} \phi_F(j,\delta) \quad \text{for } (K-1) \le n < N \tag{I-61}$$

Define the *Floor Packet Rate* (FPR) sequence with parameters n, W and δ:

$$\mathrm{FPR}(n,W,\delta) \;=\; \frac{\mathrm{FPC}(n,W,\delta)}{W} \quad \text{for } (K-1) \le n < N \tag{I-62}$$

Define the *Floor Packet Per cent* (FPP) sequence with parameters n, W and δ:

$$\mathrm{FPP}(n,W,\delta) = \left(\frac{\tau_P}{W}\right) \times \mathrm{FPC}(n,W,\delta) \times 100\,\% \;\; \text{for } (K-1) \le n < N \tag{I-63}$$

The *floor packet per cent* is applicable to defining network limits. That is, the network performance is acceptable if

$$\min_{(K-1)\le n<N} \{\mathrm{FPP}(n,W,\delta)\} \ge p\% \tag{I-64}$$

where the network acceptance criterion is p%, and the parameters W and δ are provided in the appropriate Recommendation, for example [b-ITU-T G.8261.1].

The *floor packet rate* (equivalently *floor packet count*) is a suitable metric for identifying the slave clock tolerance limit. That is, the slave clock must meet its specified output performance if

$$\min_{(K-1)\le n<N} \{\mathrm{FPC}(n,W,\delta)\} \ge m \tag{I-65}$$

where the parameters m, W and δ are provided in the appropriate recommendation as applicable.

Equations I-62, I-63, I-64, and I-65 are general and appropriate for sliding window approaches. Jumping and overlapping window calculations can be obtained by sub-sampling the sliding window samples.

For the jumping window case, estimates are derived every K samples. That is, the jumping window samples are simply the sliding window estimates under-sampled by a factor of K. Over the full measurement interval, there are $M = (N/K)$ jumping window samples and consequently the index for the jumping window sequence ranges from 0 through $(M - 1)$.

The jumping window approach is suitable when network conditions are stationary and spectral, and probability density parameters do not change rapidly. The sliding windows may be more appropriate, for example, for short term transient or rapidly changing events.

NOTE 1 – This category of PDV metrics requires a long enough measurement period such that the observed floor delay would give a good enough estimation of the absolute floor delay. The minimum measurement period depends on the type of network considered. Long measurement periods, for instance over one or several days, should be favored in order to study diurnal PDV effects.

NOTE 2 – Like minTDEV and MAFE, these metrics may be sensitive to a small number of low-lying outliers. The definition of the precise aspect that creates the potential sensitivity and the subsequent method of handling this when applying this metric is for further study.

NOTE 3 – This category of PDV metrics is sensitive to non-stationary network conditions as described in clause I.3.3 that produce floor delay steps of significant amplitude, which may occur for instance during network re-routing events. The handling of floor delay steps is for further study.

NOTE 4 – These metrics are mainly intended to be used as post-processing metrics. The use of these metrics for real-time processing is for further study.

NOTE 5 – These metrics can be used to study the PDV noise produced independently by the forward or the reverse direction of a packet timing flow. Consideration of the combined effect of both directions is for further study.

I.5.1 Determination of observed floor delay

When calculating the Floor Population metrics, it is first necessary to determine the value of the observed floor delay. Whereas it is permissible for the user to specify a suitable value for the floor delay, two data-driven methods of determining this value are described here. The first method, called the *overall minimum* method, is to use the minimum delay observed over the entire measurement period. The second method, called the *progressive minimum* method, is to use the minimum observed delay in the measurement period up to the time window over which the individual Floor Population metric value is calculated. Refer to clause I.5.1.3 for information concerning the impact of packet network re-route events on the determination of observed floor delay.

I.5.1.1 Minimum floor delay over the entire measurement

In the overall minimum method, the observed floor delay used in computing the Floor Population metrics is the minimum delay value over the entire measurement data set following Equation I-59.

As the overall minimum delay for a measurement period may not be known until the end of the period, calculation of Floor Population metric values over a given time window may depend upon delay values that have not yet been observed. This dependency upon future observations makes it more difficult to provide an early indication of Floor Population conformance for long-term tests.

I.5.1.2 Progressive determination of floor delay

In applications where it is not practical to wait till the end of the measurement period to determine the observed floor delay, the following causal estimation procedure can be used. At each Floor Population metric computation point n, the observed floor delay is estimated as the smallest delay value in the measurement period up to (and including) the window over which the metric is computed. This running estimate of floor delay is then used when calculating the Floor Population metric. This enables calculation of the Floor Population metric value at any given time, depending only upon delay values that have already been observed.

To accommodate the dynamic notion of observed floor delay, the floor population metrics defined in clause I.5 are modified to use the current retrospective estimate of the floor delay rather than the minimum over the whole data set. The terminology used is $FPx_M[n,W,\delta,d_{min}(n)]$ where x represents the metric ("count", "rate", "per cent"), the subscript M indicates that the formula used is a modified form of the ITU-T G.8260 definition, and $d_{min}(n)$ is the current running estimate of the floor delay.

In terms of Equation I-59, the observed floor delay at time n (where n is always a sample index at the end of an observation window) can be estimated as:

$$d_{\min}(n) = \min_{0 \le i < n} x[i] \qquad \text{(I-66)}$$

The value of the floor packet metrics in Equations I-61 to I-63 at time n are then calculated using $d_{min}(n)$ instead of d_{min}, as shown in Equation I-66.

As an example for a specific implementation, the floor delay at time n can be iteratively estimated according to the following algorithm:

1. Denote $d(n)$ *as* the minimum packet delay of the most recent observation window;
2. Compare $d(n)$ to the current estimate of the observed floor delay $d_{min}(n)$
   a. If $d(n) < d_{min}(n) \;\Rightarrow\; d_{min}(n) = d(n)$
   b. Otherwise$\Rightarrow d_{min}(n)$ remains unchanged (I-67)

The progressive floor determination method continually refines the estimate of the observed floor delay value during the measurement period. At each Floor Population metric computation point n, the observed floor delay is estimated as the smallest delay value, $d_{min}(n)$, in the measurement period up to (and including) the window over which the metric is computed. The running value of this estimate is then used when calculating the (estimated) Floor Population metric. The windows could be sliding, overlapping, or jumping.

Starting from the first observation window and for each subsequent window, the Floor population metrics are computed as $FPx_M[n,W,\delta,d_{min}(n)]$.

An initial *settling-time* (S s) should be allowed to ensure a good estimate of the floor has been established, resulting in valid calculations only occurring after this time has elapsed. The objective is to select a value of S after which the estimation of the observed floor delay would be close enough to the true delay floor, allowing for reliable *FPx* conformance tests to be calculated for the subsequent observation windows. The settling time can be either based on a predefined fixed value (for example, on some worst-case assumption) or calculated in real time based on the specific properties of the packet delay. The specific method and value are for further study. It has been demonstrated that for [ITU-T G.8261] Appendix VI test cases, a value of S = 600 s should suffice.

I.5.1.3 Re-route events and impact on observed floor delay

I.5.1.3.1 Re-route events

Re-route events are defined as a change in the path taken by packets through a network. Such a re-route event can result in a change to the observed floor delay of a given PTP flow.

Significant re-route events that occur in a given network are expected to have the following attributes:

– These are infrequent events. In a well-engineered telecom network, such re-route events (usually as a result of some equipment malfunction or routine maintenance) should be quite rare.

– A single re-route event may cause multiple floor delay changes over a short time period (e.g., a network maintenance activity that results in a short-term network re-route event that is restored through a second re-route event after a relatively short period of time). Such a series of proximate re-route events should be regarded as a single event. It is typically assumed that such network route instability can last between a few minutes (due to software upgrade of the routers) up to a few hours (as a result of some failure in the network).

– The consequent floor delay changes are abrupt. The floor delay changes occur within only a few packet interval durations (as it is very short, the specific duration is not critical).

I.5.1.3.2 Re-route event impact on packet network limit

Unlike network loading variations, congestion and other extreme conditions, observed floor delay changes due to re-route events are not part of the packet network limit and do not consume part of the packet network limit budget.

When a re-route event occurs, the packet network is considered to not comply with the packet network limit. The packet network limit measurement should be re-started and a new observed floor delay computed. It should be noted that between re-route events, the network observed floor delay is considered to be fixed and thus, the FPP calculation procedures described in clause I.5.1 apply.

I.5.1.3.3 Determination of re-route event

The methods of determination of re-route event occurrences and related observed floor delay change are for further study. These methods are applicable to network analysis equipment and are not applicable to packet slave clocks.

Generally speaking such methods could be classified into the following:

– Based on information sent to the probe from the network management entity alerting that a reroute event took place. Such an alert would derive the probe to search for a new observed floor delay (OFD) based on one of the methods described in clause I.5.1.

– Based on information sent to the probe from an active network OAM mechanism (whether that is part of PTP or relying on external OAM techniques) alerting that a reroute event took place. As in the previous case, such an alert would derive the probe to search for a new OFD based on one of the methods described in clause I.5.1.

– In the absence of the above methods, the use of an automatic OFD classification estimation method is an efficient way to allow for meaningful OFD computation in network with re-route events.

A mixture of all or part of these approaches could also be beneficial.

I.5.1.3.4 Determination of re-route events using an OFD estimation method

This section describes an observed floor delay estimation method that automates the location of re-routes events by post-processing packet capture data and compute the in-between observed floor delay. This approach is suggested for use when an existing network management entity is not available to make such a determination.

Being an estimation for re-route detection, this method may not fully conform to the requirements in clause I.5.1.3.2 that requires the observed floor delay to be re-calculated every re-route event. Specifically, this approach may be subject to both false alarms (incorrectly determining that a re-route event occurred when one did not actually take place) and mis-detections (failing to notice a re-route event occurrence). Since an OFD section determination is made only after a stable minimum floor is observed for a long enough time (the *S* parameter), false alarms should be quite rare. Mis-detections of small magnitude re-route events may still occur from time to time. A typical reason for mis-detecting a re-route may be that that the magnitude was too small compared with the data set noise; in such a case, such a failure may not have a material impact on the network analysis when the target network or synchronization performance is very relaxed.

An observed floor delay estimation method for finding the observed floor delay between re-route occurrences may comprise of the following recursive steps:

1) A full packet capture is performed, with minimum capture duration, creating the entire packet section. The capture must be based on an accurate time reference (e.g., GPS).

2) The packet section is analysed to find the overall floor delay, and is then divided into sliding or jumping windows.

3) The floor delay of the first window of the packet section must be close to the overall floor delay.

4) The floor delay of the last window of the packet section must be close to the overall floor delay.

5) There must not be another packet sub-section between the first and last window of the original packet section that itself:

   a) meets above criteria 2, 3 & 4 and

   b) where the overall floor delay of this sub-set packet sequence is not close to the original overall floor delay.

6) If such a sub-section is not detected, the original packet section comprises a single OFD section and its delay minimum is the OFD.

7) On the other hand, if such sub-section(s) (one or more) are detected, the entire original section is then divided down to its sub-sections and the entire process [1) to 7)] is then repeated for every subsection independently.

This procedure is based on the following mathematical characterization of a legitimate OFD section:

A group of observation windows that share the same observed floor delay property can be defined as the *largest* group of N consecutive windows (jumping or overlapping to some degree) $W^{(1)}$ to $W^{(N)}$ that meet the following requirements:

- The overall group duration is not shorter than S s AND not longer than 86 400 s.
- Given $j = \underset{k}{argmin}\left(d_{min}^{(k)}\right), 1 \leq k \leq N$, ($W^{(j)}$ is the window, within the group, having the lowest delay) the following conditions must be met:

$$\begin{aligned}\left|d_{\min}^{(1)} - d_{\min}^{(j)}\right| \leq \varepsilon \\ \left|d_{\min}^{(N)} - d_{\min}^{(j)}\right| \leq \varepsilon\end{aligned}, \qquad \text{(I-68)}$$

Where ε, in seconds, is the degree of tightness in classifying groups of distinct OFDs that we require.

- There is no subgroup of M consecutive windows $W^{(l)}$ to $W^{(l+M-1)}$ contained within $W^{(1)}$ to $W^{(N)}$ ($1 < l < l + M - 1 < N$) that meets the following requirement:
  - The overall subgroup duration is not shorter than S s AND not longer than 86 400 s.
  - Given $i = \underset{k}{argmin}\left(d_{min}^{(k)}\right), l \leq k \leq l + M - 1$, ($W^{(i)}$ is the window, within the subgroup, having the lowest delay) the following conditions are met:

$$\begin{aligned}\left|d_{\min}^{(l)} - d_{\min}^{(i)}\right| \leq \varepsilon \\ \left|d_{\min}^{(l+M-1)} - d_{\min}^{(i)}\right| \leq \varepsilon \\ \left|d_{\min}^{(j)} - d_{\min}^{(i)}\right| > \delta\end{aligned} \qquad \text{(I-69)}$$

$W^{(k)}, 1 \leq k \leq N$ can be either non-overlapping or overlapping (to whatever degree) 200 s windows.

The determination of the parameters S, ε, δ is for further study.

I.5.2 Use of floor delay packet population PDV metrics for two-way protocol flows

The floor delay packet population PDV metrics may be used with two-way protocol flows.

One usage, following Equation I-59, would establish a unique minimum path delay, d_{min}, for each direction of a two-way protocol flow, resulting in the forward path delay, $d_{\text{min-fwd}}$ and the reverse path delay, $d_{\text{min-rev}}$. These minimum path delays may then be processed to compute the path delay asymmetry, based on minimum path delay:

$$path\_delay\_asymmetry = (d_{\text{min-rev}} - d_{\text{min-fwd}})/2 \qquad \text{(I-70)}$$

These minimum path delays, $d_{\text{min-fwd}}$ and $d_{\text{min-rev}}$ may also be used to compute FPP according to Equation I-63 for the forward path, $\text{FPP}_{fwd}(n,W,\delta)$, and the reverse path, $\text{FPP}_{rev}(n,W,\delta)$. The $\text{FPP}_{fwd}(n,W,\delta)$ and $\text{FPP}_{rev}(n,W,\delta)$ may then be applied against defined performance limits.

I.5.3 Exceptional events and impact on packet network limit

Exceptional events and other severe, unexpected network phenomena may occur from time to time in a packet network. These events may reduce the number of packets that arrive within the defined cluster range, thereby causing a temporary failure of the packet network to comply with the defined FPP network limit. To accommodate such exceptional events, a small number of non-overlapping failing windows (X) may be allowed over a measurement period (Y). To ensure that such exceptional events have limited time duration, the maximum number of consecutive non-overlapping failing windows may also be specified (Z). The values of X, Y and Z are defined in the relevant recommendations.

Bibliography

[b-ITU-T G.8261.1] Recommendation ITU-T G.8261.1/Y.1361.1 (2012), *Packet delay variation network limits applicable to packet-based methods (Frequency synchronization)*.

[b-ITU-T G.8263] Recommendation ITU-T G.8263/Y.1363 (2017), *Timing characteristics of packet-based equipment clocks.*

[b-GR-1244-CORE] GR-1244-CORE (2009), *Clocks for the synchronized network: Common generic criteria*, Telcordia Technologies Generic Requirements, (Issue 4, October 2009).

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Published in Switzerland
Geneva, 2023