Teemu Laine

New synchronization metrics for packet networks

Faculty of Electronics, Communications and Automation

Thesis submitted for examination for the degree of Master of Science in Technology

Espoo 16.8.2010

Supervisor:

Prof. Samuli Aalto

Instructor:

D.Sc. (Tech.) Antti Pietiläinen

Aalto University
School of Science and Technology

AALTO UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

ABSTRACT OF THE
MASTER'S THESIS

Author: Teemu Laine

Title: New synchronization metrics for packet networks

Date: 16.8.2010 Language: English Number of pages: 8+92

Faculty of Electronics, Communications and Automation

Department of Communications and Networking

Professorship: Communications and Networking Code: S-38

Supervisor: Prof. Samuli Aalto

Instructor: D.Sc. (Tech.) Antti Pietiläinen

This thesis introduces the challenging research area of network synchronization over packet-switched transport networks (also known as timing over packet (ToP)). Synchronization in telecom terminology consists of frequency, phase and time synchronization which require different approaches and performance metrics. Since the focus is on cellular network synchronization, technology requirements pose a strict demand for the new packet-based mobile backhaul replacing a traditional TDM-based infrastructure as a common timing reference distributor.

Consequently, synchronization metrics are the crucial performance indicators and estimators needed for system designing and performance verification related to ToP solutions. Packet metrics derived from the traditional synchronization metrics as well as completely new metrics such as MATIE and MAFE are analyzed and studied in great detail. Furthermore, an improved MATIE is also developed in a search for better performance.

Several measurements are conducted and from the several data recorder the implemented synchronization metrics are calculated and analyzed against simulated and real packet slave clock implementations etc. As a result, MAFE is proven to be the best frequency stability estimate and MATIE (or MATIE modification) one of the best phase stability estimates but not that applicable to satisfy the traditional MTIE masks. MTIE from simulated clock output would be the most applicable.

Keywords: Mobile backhaul synchronization, Timing over Packet (ToP), frequency sync, phase sync, time sync, MATIE, MAFE, TDEV, MTIE, PLL filter simulation

AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Teemu Laine

Työn nimi: Uudet synkronointimetriikat pakettiverkkoihin

Päivämäärä: 16.8.2010 Kieli: Englanti Sivumäärä: 8+92

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka Koodi: S-38

Valvoja: Prof. Samuli Aalto

Ohjaaja: TkT Antti Pietiläinen

Tämä työ esittelee verkkojen synkronoinnin haasteet pakettikytkentäisten siirtoverkkojen yli esimerkiksi mobiili-verkon tukiasemiin, joka tunnetaan nimellä ToP (timing over packet). Tässä yhteydessä synkronisaatio tarkoittaa yhteisen taajuuden, vaiheen tai ajan mahdollistamista, jotka vaativat hieman erilaisia menetelmiä ja metriikoita. Työssä keskitytään mobiili-verkkojen synkronointiin, joissa tekniikka asettaa tarkat vaatimukset synkronoinnin tarkkuudelle. Siirtoverkkojen muuttuminen aikajakoisista tekniikoista pakettikytkentäiseen vaatii synkronoinnin uudelleen miettimistä.

Synkronointimetriikat ovat tärkeitä suorituskykyindikaattoreita ja -ennustajia, joita tarvitaan ToP-järjestelmän suunnittelussa ja testauksessa. Perinteisistä kellometriikoista johdetut pakettimetriikat sekä täysin uudet pakkettimetriikat on tutkittu ja analysoitu tarkkaan. Näiden lisäksi uutta MATIE-metriikkaa yritettiin parantaa paremman suorituskyvyn toivossa.

Tutkimuksessa on tehty useita mittauksia ja tietokoneella toteutettuja metriikoita sovellettiin saatuihin mittaustuloksiin sekä verrattiin itse toteutettuihin simuloituihin orjakelloihin. Tuloksena saatiin, että MAFE osoittautui kestäväksi ja toimivaksi taajuusstabiilisuusmittariksi ja MATIE (tai paranneltu MATIE) on toimiva, mutta ei kovin käyttökelpoinen perinteisten maskien kanssa. Parhaaksi ratkaisuksi osoittautui MTIE-metriikka simuloidun orjakellon ulostulosta.

Avainsanat: Mobiili-verkkojen synkronointi, Timing over Packet (ToP), taajuus-, vaihe- ja aikasynkronointi, MATIE, MAFE, TDEV, MTIE, orjakellon simulointi

Preface

Thanks to the Nokia Siemens Networks for the wonderful opportunity to do my thesis for an international company and giving the challenging topic of packet synchronization. Especially, big thanks goes to my mentor and instructor Antti Pietiläinen from Nokia Siemens Networks for great guidance and being patient and extremely helpful throughout the whole writing process and past internships. In addition, I also appreciate the effort that my supervisor Prof. Samuli Aalto has put into this thesis and not to forget to thank Martin William for the spelling corrections. Last but not least, I would like to thank my parents for their unbreakable support and encouragement for studies.

Thank you,

Espoo, 16 August 2010

Teemu Laine

Table of contents

Abbreviations

ACR	Adaptive Clock Recovery
BC	Boundary Clock
BE	Best Effort
BMCA	Best Master Clock Algorithm
CAPEX	Capital Expenditure
CBR	Constant Bit Rate
CDMA	Code Division Multiple Access
CES	Circuit Emulation Services
CESoPSN	Circuit Emulation Services over Packet Switched Network
CPU	Central Processing Unit
CUT	Clock Under Test
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DVB-H	Digital Video Broadcasting Handhelds
EEC	(Synchronous) Ethernet Equipment Clock
EF	Expedited Forwarding
ESMC	Ethernet Synchronization Messaging Channel
FDD	Frequency Division Duplex
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HSPA	High Speed Packet Access
HSPA+	Evolved High Speed Packet Access
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIR	Infinite Impulse Response
ITU	International Telecommunication Union
IWF	Interworking Function
LTE	Long Term Evolution
MAFE	Maximum Average Frequency Error
MATIE	Maximum Average Time Interval Error
MBMS	Multimedia Broadcast and Multicast Services
MBSFN	Multi Media Broadcast over a Single Frequency Network
MDEV	Modified Allan deviation
MIMO	Multiple-input / Multiple-output
MPLS	Multiprotocol Label Switching
MRTIE	Maximum Relative Time Interval Error
MTIE	Maximum Time Interval Error
NGN	Next Generation Network
NNI	Network Node Interface
NTP	Network Time Protocol
NTR	Network Timing Reference
OCXO	Oven Controlled Crystal Oscillators
OPEX	Operational Expenditure

OTN	Optical Transport Network
PBCR	Packet Based Clock Recovery
PDF	Probability Density Function
PDH	Plesiochronous Digital Hierarchy
PDV	Packet Delay Variation
PHB	Per-Hop Behavior
ppb	part per billion
ppm	part per million
pps	packets per second
PRC	Primary Reference Clock
PTP	Precision Time Protocol
QoS	Quality of Service
SDH	Synchronous Digital Hierarchy
SHDSL	XX Digital Subscriber Line
SLA	Service Level Agreement
SP	Service Provider
SSU	Synchronization Supply Unit
TCO	Total Cost of Ownership
TCXO	Temperature-Compensated Crystal Oscillator
TDD	Time Division Duplex
TDEV	Time deviation
TDM	Time Division Multiplexing
TD-CDMA	Time Division Code Division Multiple Access
TD-SCDMA	Time Division Synchronous Code Division Multiple Access
TE	Time Error
TIE	Time Interval Error
ToP	Timing over Packet
UMTS	Universal Mobile Telecommunications System
UTRA	UMTS Terrestrial Radio Access
VDSL	Very high bitrate DSL
VoD	Video on Demand
WCDMA	Wideband CDMA
WiMAX	Worldwide Interoperability for Microwave Access
3GPP	3rd Generation Partnership Project

1. Introduction

Generally in telecommunications, synchronization refers to the common frequency among multiple equipment clocks. The common frequency is broadly referred to as the reference signal which is delivered to the network. Phase and time synchronization, on the other hand, are properties which cannot be derived from common frequency reference but in some cases are mandatory. Accurate synchronization in telecommunication networks is a crucial and compulsory requirement for non-deteriorated performance. Therefore, synchronization metrics are an essential part of the network design (e.g. mobile backhaul) process by estimating and verifying synchronization performance against technology requirements.

The purpose of this thesis is to familiarize the reader with the area of mobile backhaul synchronization with focus on timing over packet (ToP) synchronization solutions as opposed to competing technologies (e.g. synchronous Ethernet). The objective is to introduce packet synchronization metric principles and performance by first discussing existing methods, environmental effects and performance analysis. Then new packet synchronization metrics are introduced. The standardization process is ongoing in ITU-T and, therefore, this thesis reflects very recent and vibrant research area. The actual research question behind this thesis is what kind of metrics would fit best to describe and estimate packet timing performance in terms of frequency and phase synchronization. Packet networks introduce significant challenges to synchronization delivery. The greatest is packet delay variation (PDV) that must be taken into account in creating metrics as well as in the synchronization network design. The scope of the research is mainly limited to the frequency and phase synchronization investigation in the mobile backhaul environment, although, time synchronization issues are also briefly discussed at the end.

Frequency synchronization of base stations is needed especially to facilitate the handovers in the network. Phase or time synchronization is needed besides the frequency synchronization in some cellular technologies (e.g. CDMA2000). Currently time synchronization is distributed via GPS. Delivering accurate phase or time synchronization over network requires support from the all intermediate network elements. ITU telecommunication standardization sector consented the recommendations related to packet based frequency synchronization recently, in June 2010, and packet based time synchronization is due end of 2011.

The structure of this thesis is as follows. First, background information and principles of network synchronization are introduced along with the fundamental reasons for delivering synchronization. The fast mobile network evolution and the huge increase in mobile data usage drive the change also in mobile transport (backhaul) networks which forces service provider to reconsider the synchronization issues due to moving to asynchronous packet networks (e.g. Ethernet). Moreover, cellular network frequency and phase/time synchronization requirements are discussed at the end of Chapter 2.

Chapter 3 deepens the understanding by introducing methods for delivering the frequency or phase synchronization starting from legacy transmission technologies. Traditional PDH/SDH synchronization networks, still widely used, are stepping away from the new solutions designed for delivering synchronization in inherently asynchronous networks. Adaptive clock recovery, network timing reference, Synchronous Ethernet, and global satellite navigation systems are alternatives for timing over packet (ToP) synchronization.

Since the timing over packet (ToP) solution is the main method under research in this thesis, Chapter 4 elaborates the principles and the protocols behind ToP. Moreover, some business aspects related to ToP, packet slave clock design, architecture and implementation issues are discussed. Timing over packet solutions are actually deployed more and more by operators due to the lowest total cost of ownership (TCO). These are important elements regarding the understanding of the big picture of mobile network synchronization.

After discussing background issues from different perspectives, Chapter 5 introduces metrics related to the synchronization for the first time. This is the background for the research topic in question, the synchronization metrics. The traditional telecommunication metrics have been used for decades and the network synchronization limits have been expressed in the form of masks for the metrics at different interfaces in certain architectures. More specifically, the masks describe the boundaries for the metric outputs and the conformance with the masks indicates that the network should operate more or less flawlessly. Furthermore, the traditional clock metrics will be proven to be applicable in the packet environment as well as in legacy synchronization networks even though with the addition of packet selection methods.

However, the imperfection of the traditional synchronization metrics applied in packet environment leads to the introduction of new packet synchronization metrics MATIE and MAFE. Chapter 6 explains the new frequency and phase synchronization metrics for packet based environments. Moreover, the PLL (phase-locked loop) filter (i.e. the process of locking the oscillator to the timing reference) simulations are discussed since they have a significant role in evaluating the performance of packet metrics.

The research methods, measurement setups and data are discussed in Chapter 7. Fundamental packet delay stability measurements in the transport networks are the basis for the whole synchronization metric study. Some of the PDV measurements used in this thesis were not conducted by the author. Nevertheless, it is essential to comprehend the principles of the measurement setup and equipment in addition to the implementation of the actual metrics. The metrics were implemented as c-language scripts for mathematical software called Origin.

The results related to the performance of the new frequency synchronization metric MAFE are extensively expounded in Chapter 8. Verification of the performance of MAFE is extensively researched on several different data. In addition, MAFE has been tested against several different versions proposed by the members from the standardization body (ITU-T Q13/SG15).

Next, the performance of the phase stability metric MATIE has been evaluated in Chapter 9. For certain delay profiles it has been shown that the MATIE produces estimates of the clock stability slightly worse than desired, thus, one goal of the thesis was to develop an improved MATIE if possible. As the result, a MATIE modification is introduced and the performance of MATIE is compared with the MATIE modification.

Before concluding the research, the time synchronization issues are shortly presented in Chapter 10 with addition to the PTP (precision time protocol) protocol operation with on-path support when delivering accurate time synchronization. As mentioned, in addition to the frequency synchronization, the standardization of time synchronization is also in progress and completion is scheduled for 2011. Thus it is not the main focus of the thesis.

Finally, conclusions and observations are presented and discussed in the final chapter.

Reader should note! As a convention in this thesis, since several standard numbers appear through the text there are no citations after those sentences, however, every standard can be found from the reference list at the end of the thesis.

2. Principles of cellular network synchronization

Cellular backhaul networks experience evolution just like any other transport networks towards packet transport but concurrently bring new challenges for synchronization. A huge transition in cellular transport networks is occurring at the moment, which makes this topic current and relevant for service providers and telecommunication equipment manufacturers. Since synchronization is a compulsory feature in most cellular networks the selection of synchronization delivery method is also an important technology and business decision for service providers (SPs) due to different expenses. The total cost of ownership (TCO) is the key financial estimate that service providers want to minimize when choosing the technology strategy of the company [1]. The next sections elaborate upon mobile transport network evolution and the fundamental requirements for network synchronization.

2.1 Mobile transport network evolution

Mobile transport networks (i.e. mobile backhauls) are evolving from narrowband circuit-switched access networks based on TDM (time division multiplexing) to packet-switched networks based on Ethernet and IP (Internet protocol). The pressure on service providers to achieve better profitability, reduce costs and increase bandwidth at the same time is driving this significant change. Figure 1 depicts the change of backhaul technology markets towards Ethernet-based transport. Especially, fixed and mobile broadband and various video on demand (VoD), peer-to-peer (p2p), and IPTV services are vastly increasing the demand for more capacity [2]. Undoubtedly, the increased use of mobile broadband and the fast development of cellular network bitrates are posing a demand for new capacity in the mobile backhaul as well. Many operators are now investing in IP/Ethernet backhaul because of the HSPA/HSPA+ (high speed packet access) technology deployment and the ultimate need to lower backhaul costs. In addition, nearly every operator is moving to an IP/Ethernet mobile backhaul when deploying LTE (long-term evolution) [3]. There is also a need for flexibility to carry different types of services in the same transport network, i.e. network convergence. [4] [5]

Legacy TDM based transport networks (e.g. SONET/SDH or PDH networks) have inherently the property of carrying the timing reference at the physical layer. Traditional TDM networks also require frequency synchronization all over the networks in order to work properly which is technically well understood and implemented [6]. Timing signal jitter and wander limits are specified for TDM interfaces and the conformance with the limits is crucial in terms of TDM network performance. However, packet networks are inherently asynchronous by nature, i.e. their internal clocks can run freely without the need to obtain common timing. Nevertheless, there are still certain applications attached to the transport networks that require accurate frequency synchronization. For example, mobile base stations and TDM circuit emulation services (CES) require at least accurate frequency synchronization. Consequently, synchronization is required in packet networks only at the edge of the networks instead of the whole network. This introduces new kinds of challenges which require new methods and solutions. The next chapter introduces methods for delivering needed synchronization but first the requirements are explained in more detail.

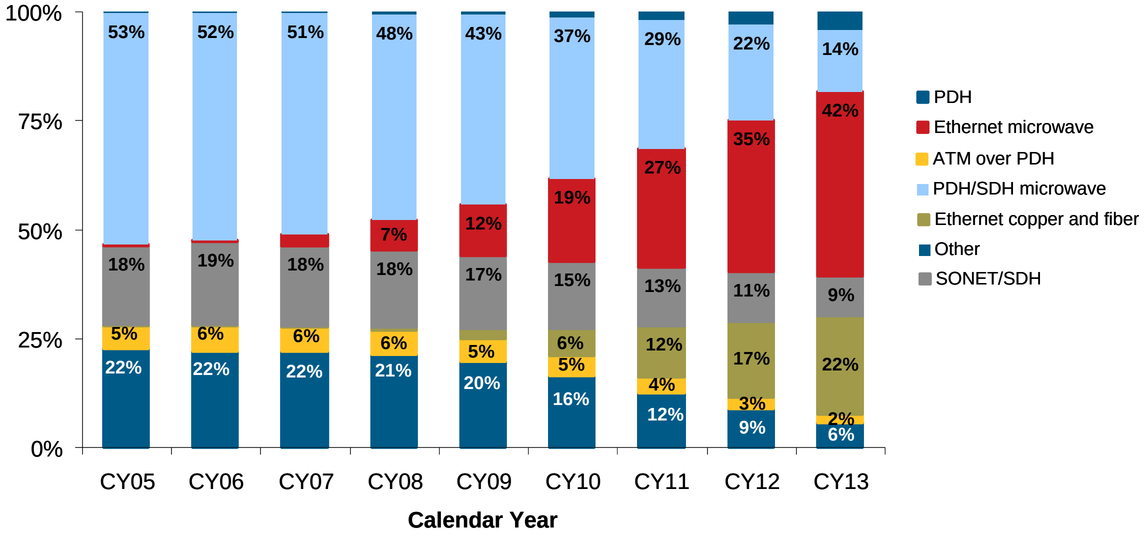


Figure 1: Mobile backhaul technology market shares by calendar years. Ethernet-based transport (microwave, fiber or copper) is becoming the dominant technology and previously dominant PDH/SDH technology is vanishing slowly. [7]

2.2 Synchronization requirements

Telecommunication network synchronization is the generic concept for distributing common phase, time or frequency to all the nodes that require it to coordinate their local time, phase or frequency. Therefore, the term, network synchronization, can be divided into three types of synchronization; frequency, phase and time synchronization.

Frequency synchronization is naturally the distribution of a common frequency reference to the nodes for controlling their local oscillators which is explained in the next section. Note that, two or more clocks can be frequency synchronized and at the same time have no common phase. The term phase synchronization implies that all the nodes have access to the reference timing signal of which rising edges occur at the same time. Figure 2 illustrates phase synchronization which is also in that case providing frequency synchronization. The term time synchronization obviously means distributing a common time reference to the network. Figure 3 depicts time synchronization between two systems. Note that time and phase synchronization are closely related and providing time synchronization is one way of achieving phase synchronization. For instance, the International Telecommunication Union (ITU) has specified standard time scale UTC (coordinated universal time) which is sometimes used in telecommunication systems and can be distributed using time of day (ToD) information. UTC is an atomic time scale designed to approximate Universal Time. However, at the moment in mobile backhaul time synchronization the GPS time is the most widely used timescale. [2]

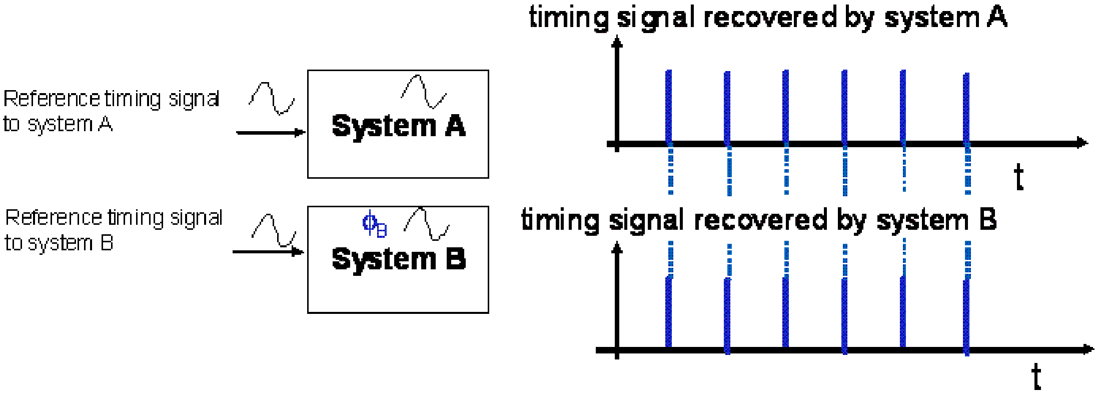

timing signal recovered by system A
Reference timing signal to system A
System A
t
timing signal recovered by system B
Reference timing signal to system B
ϕ_B
System B
t

Figure 2: The principle of phase synchronization. The "significant instants" of the signal (blue lines) are phase synchronized. Both systems are also frequency synchronized meaning the significant instants of the output timing signal occur at the same rate. [8]

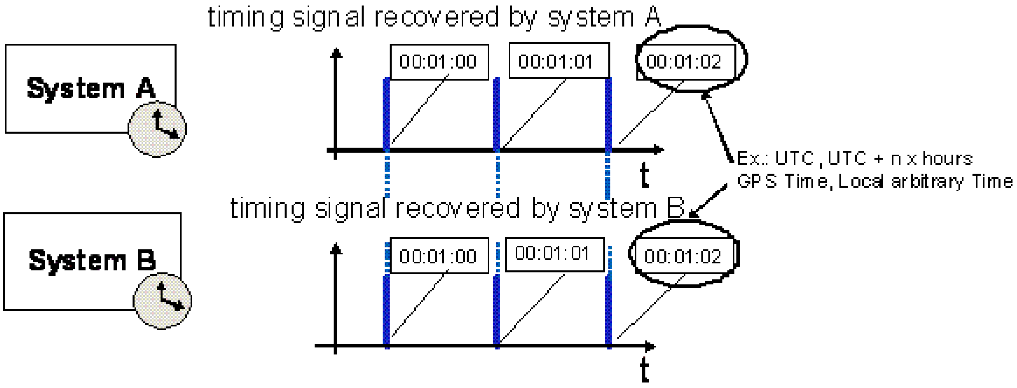

timing signal recovered by system A
System A
00:01:00
00:01:01
00:01:02
t
Ex.: UTC, UTC + n x hours
GPS Time, Local arbitrary Time
timing signal recovered by system B
System B
00:01:00
00:01:01
00:01:02
t

Figure 3: The principle of time synchronization. Nodes have a common reference for time and share a common timescale (e.g. UTC or GPS time). [8]

The simple purpose of synchronization in telecommunications is to make sure that equipments send and receive data at the same rate and possibly also at the same time. For example, if the TDM infrastructure is no longer synchronized, data would be lost, applications experience impairments and service quality would be degraded [4]. Time and phase synchronization are often additional requirements for certain applications or nodes. For example, mobile networks require accurate frequency synchronization for radio interfaces and in some cases also phase or time synchronization. In addition to the wireless cellular networks, circuit-emulation services (CES) set also very challenging requirements on synchronization. More precise requirements for synchronization are introduced hereafter, starting from frequency synchronization.

2.2.1 Frequency synchronization requirements

Radio networks and especially FDD (frequency division duplex) systems require always frequency stability which sets strict requirements for synchronization. Frequency stability must be guaranteed in the air interface in order to minimize disturbance, facilitate handover between base stations, and fulfill regulatory requirements (i.e. synchronization ensures that radio spectrum is not spread into the adjacent channels). In addition, in many existing networks accurate multiplexing and de-multiplexing of data requires frequency synchronization across the entire network.

Frequency synchronization requirements are usually expressed in parts-per notation due to the low-values. Parts-per notation is the ratio between the observed frequency and the reference frequency (i.e. measured frequency divided by reference frequency minus one). Parts-per-million (ppm) and parts-per-billion (ppb) are typically used. As an example, 3GPP (3rd generation partnership project) specifies that WCDMA (wideband code division multiple access) and LTE systems require frequency stability of the transmitted radio signal better than ± 50 ppb (parts per billion). Moreover, GSM (global system for mobile communications) and CDMA2000 specifications have the same frequency stability requirements. Table 1 lists the frequency accuracy requirements for different mobile network technologies. Note that the requirements are in the air interface (i.e. the radio interface), and the requirements inside the base station (i.e. at the input transmission interface) could be even more stringent. [2]

2.2.2 Phase and time synchronization requirements

In addition to frequency synchronization, some FDD systems require phase and time synchronization (e.g. in cellular technologies like CDMA2000 and CDMAone). Generally, TDD (time division duplex) systems need accurate time or phase synchronization. For instance, mobile systems based on Evolved UTRA-TDD or UTRA-TDD (UMTS terrestrial radio access – TDD) e.g. WCDMA TDD and TD-SCDMA and LTE-TDD base stations require phase instability of less than ± 1.5 µs (i.e. 3µs between base stations). WCDMA MBMS (multimedia broadcast and multicast services), in turn, does not require very accurate phase synchronization (only ± 20 ms) because it has been designed to work within the specifications defined in the original 3GPP standard. On the other hand, MBSFN (MBMS single frequency network) associated with LTE requires phase accuracy within a few microseconds. Table 1 lists also time synchronization requirements of the most common mobile network technologies that require it (e.g. mobile WiMAX). Strict phase synchronization is also needed in LTE’s MIMO (multiple-input/multiple-output) architecture; otherwise multiple signals can simply cancel out each other [9]. Moreover, the GSM radio access network can be bandwidth optimized by having accurate phase synchronization and thus enhance the radio network capacity. Relatively accurate time or frequency synchronization is needed as well in applications such as IP network delay monitoring, VoIP services (voice over IP), or IPTV, for example. [2]

Table 1: Mobile network synchronization requirements (frequency synchronization in both air and transmission interface and time synchronization) [1] [2]

	Frequency accuracy in Air Interface	**Clock accuracy in Transmission Interface**	**Time of Day requirement**
GSM	± 50 ppb	Not standardized (Vendor specific ±10 – 20 ppb)	N.A.
WCDMA	± 50 ppb	Not standardized (Vendor specific ±10 – 20 ppb)	FDD: N.A. MBMS: ± 20 ms TDD: ± 1.5 µs
TD-SCDMA	± 50 ppb	Not standardized (Vendor specific ±10 – 20 ppb)	FDD: N.A. TDD: ± 1.5 µs
CDMA2000	± 50 ppb	N.A.	±3 µs (±10 µs holdover)
Mobile WiMAX	± 15 ppb	Not standardized (Vendor specific less than ±10 ppb)	FDD: N.A. TDD: ± 0.7µs
LTE	± 50 ppb	Not standardized (Vendor specific ±10 – 20 ppb)	LTE MBMS: ± 5 µs preliminary value LTE TDD: ± 1.5 µs
DVB-H	Depends on radio frequency but might be a few ppb	N.A.	DVB-H/T SFN: within 1µs

2.3 Principle of clock recovery

The following chapters will focus on how synchronization can be delivered accurately enough. But before that, one should understand the principle of clock recovery (i.e. coordination of local oscillator according to the reference timing signal.

At the receiver of the timing signal a PLL (phase-locked loop) controls the frequency or the phase of the local oscillator according to the received reference timing signal. The function of PLL is to lock the local clock to the incoming reference signal. A simplified PLL is depicted in Figure 4. It consists of a comparator comparing the local timing signal to the incoming reference timing and a low-pass filter for removing jitter and wander before coordinating the oscillator. Typically, the cut-off frequency of PLL low-pass filter varies from a few Hz to sub-mHz depending on the situation and requirements.

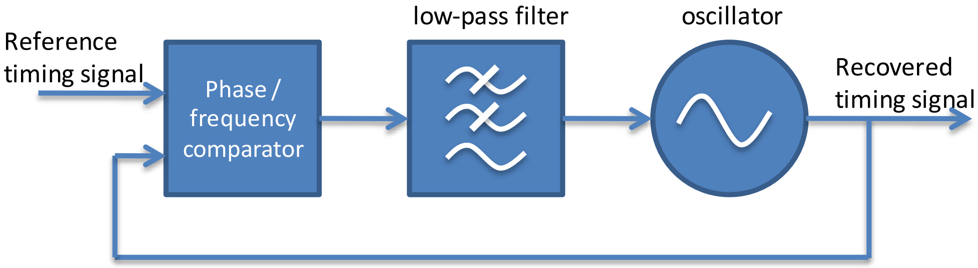


Figure 4: The simple block diagram of phase-locked loop (PLL) for controlling local oscillator.

Traditional TDM-based networks deliver the reference timing signal on physical layer (PHY) to the slaves. Generally, a conventional timing signal is pulse shaped and due to imperfections it experiences jitter and wander along the path. Rising edges of the rectangular signal are considered significant instants that are tried to align in synchronization among multiple clocks. The conventional rectangular timing signal is illustrated with possible jitter and wander in Figure 5.

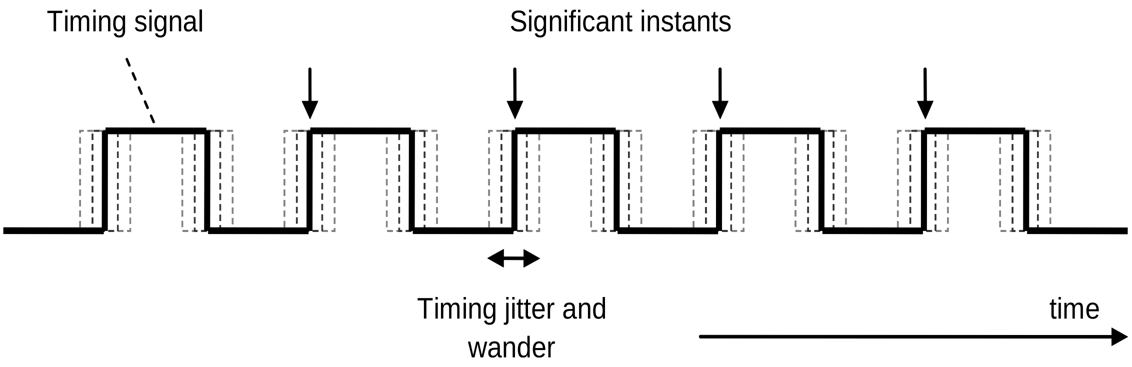


Figure 5: Conventional timing signal of which significant instants are the rising edges. Jitter and wander causes these significant instants to vary slightly from their ideal positions in time. [10] [8]

Consequently, in the context of mobile networks, the reference timing must be delivered directly to the base stations or over backhaul transport networks master-slave arrangements. The master nodes usually have access to a primary reference clock (PRC) traceable reference signal. PRC is a high accuracy clock of which long-term frequency stability error should be below 1 part per 10^{11}. Note that the quality of PRC is specified by ITU-T (ITU – telecommunication sector) in recommendation G.811 for digital network synchronization. [11]

3. Introduction to different methods for delivering synchronization

This chapter discusses several methods for distributing timing reference for the networks. Some of the technologies are suitable only for distributing frequency synchronization whereas some can handle both the frequency and time or phase synchronization delivery. However, note that the technologies presented are the most widely deployed synchronization methods for mobile backhaul purposes.

3.1 SDH/PDH synchronization

As it was explained previously, TDM-based PDH/SDH (plesiochronous digital hierarchy / synchronous digital hierarchy) networks can deliver and recover accurate timing on the physical layer. Since TDM-based networks need synchronization in order to work, thus, they can also deliver the timing reference for other applications like base stations (i.e. every base station connected via SDH/PDH can be synchronized with TDM timing). PDH E1/T1 links have been and are at the moment the most common method of delivering timing to the mobile network base stations but there is transition ahead toward other methods providing more capacity, flexibility and significant cost-savings [2]. And mostly, mobile operators are leasing E1/T1 copper lines from some other infrastructure owner, which leads to constant expenses.

SDH/SONET (synchronous optical networking) networks run on optical media and are strictly synchronized across the entire network, which reduces the need for buffers etc. SDH hierarchy introduces three levels, primary reference clock (PRC, characteristics specified in G.811), synchronization supply unit (SSU, characteristics specified in G.812) and SDH Equipment Clock (SEC, characteristics specified in G.813). The distribution of synchronization is based on the master-slave architecture and the synchronization is achieved by conveying the timing signal hop-by-hop basis (i.e. from one clock to another clock). Synchronization delivery paths begin always from PRCs. Figure 6 depicts the master-slave PRC reference delivering concept compared with distributed PRC.

Wander is caused, for example, by multiplexing and de-multiplexing in PDH systems or pointer movements in SDH/SONET systems [12]. SDH is synchronous whereas PDH networks are plesiochronous by nature which means that the network is nearly but not perfectly synchronized. PDH transmission generates more wander than SDH. Note that even in PDH the frequency of TDM stream is retained once it is de-multiplexed back from the aggregate streams.

Because of imperfections, ITU-T has specified certain metrics (introduced in recommendation G.810) and masks (specified by G.823 and G.824). Metrics are used to measure the synchronization performance and the compliance with the specified wander masks (limits) at the certain interfaces verifies the quality of synchronization. Specifically, base station vendors refer typically to the G.823 traffic mask which describes tolerable jitter and wander at the traffic interface since the reference signal is delivered along with the data. The masks and metrics are explained in more detail in Chapter 5.

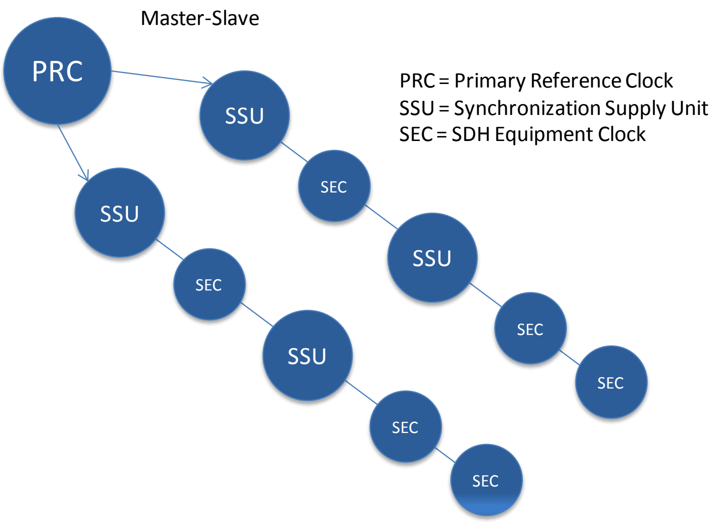

Master-Slave
PRC
SSU
SEC
PRC = Primary Reference Clock
SSU = Synchronization Supply Unit
SEC = SDH Equipment Clock

Figure 6: SDH/PDH clock chain is based on master-slave architecture (i.e. the most common timing distribution architecture). There may be up to 20 SECs between SSUs [13].

3.2 Timing over Packet

Timing over packet (ToP) methods distribute synchronization via packets that carry the time stamps generated by a master (server) that has access to a PRC reference. Figure 7 illustrates a simple ToP concept and architecture. The receiving equipment (slave) typically recovers the frequency synchronization by comparing the time provided by the timing packets (timestamps) with the local time. [2]

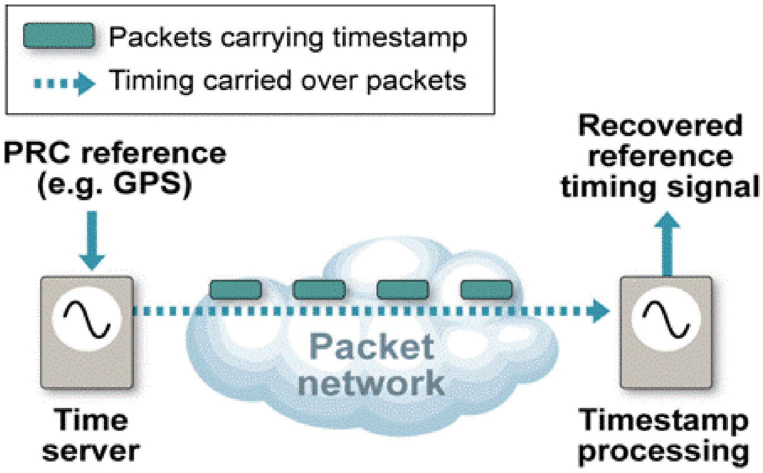

Packets carrying timestamp
Timing carried over packets
PRC reference (e.g. GPS)
Recovered reference timing signal
Packet network
Time server
Timestamp processing

Figure 7: Principle of ToP synchronization distribution. Time server acts as master (i.e. transmitter of the timing signal carried over packet network) and timestamp processing (i.e. timing recovery) happens at the slave end. [2]

The network between the master and slaves introduces multiple challenges for distributing timing since asynchronous and non-deterministic packet-switched networks introduce packet delay, variation and packet loss. Packet delay variation (PDV) is the typically used term to describe packet delay behavior in packet networks. [10] The effect of PDV to synchronization is elaborated in Chapter 4. In order to ensure that all network clocks will be synchronous (i.e., clocks are running at the same frequency), a robust method of recovering the synchronization in packet environment must be implemented. Moreover, to be able to guarantee frequency synchronization in the order of parts per billion (ppb), PDV and other undesirable packet network behavior must be taken into account and the effects in the receiving clock should be eliminated or minimized.

In contrast to the jitter and wander of a conventional timing signal (see Figure 5), PDV is essentially the same effect in packet environment as depicted in Figure 8 which illustrates the significant instant of the packet timing signal. The front edge of the packet header is the arrival time of the timing packet and with that information, only the less delayed packets can be chosen.

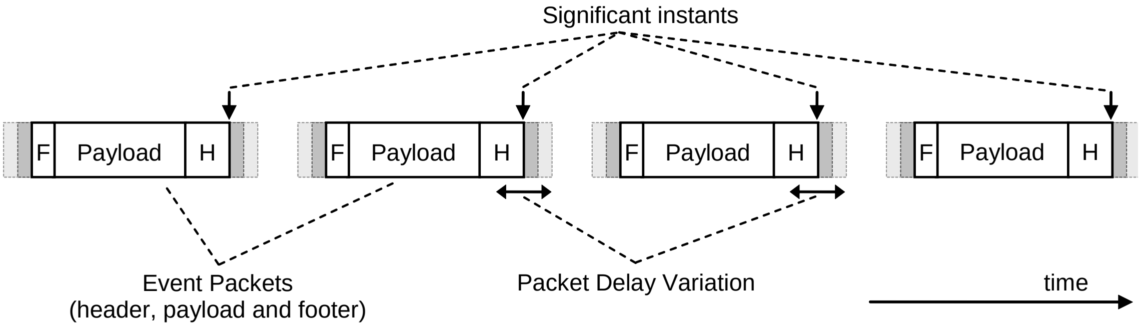


Figure 8: Packet timing signal, the significant instant is usually the front edge of the packet header. PDV illustrates the variation of significant instants.[8]

The slave must be able to filter out the effects of PDV (i.e. noise generated by network elements etc.). This poses also stability requirements for slave oscillators and they must be able to manage relatively long averaging periods (e.g. the oscillator should be stable for up to thousands of seconds). Temperature variation is the most important source of frequency drifting for oscillators. As a result, e.g. OCXO (oven-controlled crystal oscillator) based clock can tolerate much greater PDV than TXCO (temperature-compensated crystal oscillator) based clock by having better stability in varying conditions.

It is important to notice that in telecommunications, the offered services specify the required synchronization accuracy irrespective of the synchronization methods. For instance, PDH/SDH stability specifications (developed decades ago) have been used as a specification for base stations since the stability requirement of 18 ppb (specified by PDH mask, explained later) can be achieved with reasonable effort. Thus, the PDH/SDH synchronization specifications are still being stipulated in packet backhaul synchronization in some cases. Consequently, since the requirements of mobile base stations corresponding approximately to the network limits of PDH networks are rather challenging considering packet timing, it is crucial for operators to either cope with network conditions or try to negotiate an adequate SLA (service level agreement) due to the PDV sensitivity of ToP solutions. But SLA parameters are directly related to the amount of invested money. For instance, link capacities and buffers have direct relation to the delays that packets experience in the network. That is where the packet synchronization metrics come into play. There is a need for being able to determine whether the packet network is actually good enough for ToP solutions. These packet metrics are discussed in detail in Chapters 5 and 6.

The core of the ToP solutions is the clock recovery algorithm but the protocol creating and delivering the actual timing packets must be suitable for the purpose. Protocols like NTP (network time protocol) and IEEE 1588 are developed for synchronization purposes on packet networks. Note that the IEEE (Institute of Electrical and Electronics Engineers) 1588 standard defines precision time protocol, PTP. This acronym is used in this thesis hereafter. The advantage of PTP is that it has the ability to support very accurate time and phase synchronization with a boundary clocks or transparent clocks en route. The both clock types are discussed later in detail. Regardless of the protocol, ToP solutions also add a small amount of traffic (which is approximately 12 kbit/s traffic per synchronization stream) to the data links because timing packets travel along the data paths. However, typical packet based backhaul bandwidths start from ~ 10 Mbit/s so the ToP load is insignificant. [2]

Chapter 4 elaborates the critical and important issues of ToP synchronization solutions. Furthermore, the mechanisms, protocols and algorithms of ToP are discussed in more detail.

3.3 Synchronous Ethernet

Synchronous Ethernet (abbreviated as SyncE) is a timing distribution technology over Ethernet physical interfaces specified by ITU-T recommendations G.8261, G.8262 and G.8264. G.8261 specifies the overall SyncE architecture and network limits whereas the recommendations G.8262 and G.8264 specify equipment clock characteristics and synchronization status messages, respectively. For example, G.8262 specifies clock accuracy requirements, holdover performance and noise tolerance requirements for the Synchronous Ethernet equipment clock (EEC). Furthermore, the SyncE timing solution over optical transport networks (OTN) has been specified in ITU-T recommendations: G.709, G.798 and G.8251. Note that, SyncE is an extension which does not affect existing IEEE 802.3 Ethernet specifications. One of the original goals of the SyncE development was to make it interworking with SHD/PDH synchronization network. It brings a significant benefit for operators that they have to maintain and manage only a single synchronization network. [4]

The principle of SyncE is that EECs, synchronize to the frequency received from one interface and use the same frequency for all outgoing interfaces. This same procedure is used in the next hop along the synchronization path until the edge has been reached, i.e. SyncE is a hop-by-hop synchronization delivery protocol. In addition, SyncE introduces synchronization status messages (SSM) that contain indication of the quality level of the clock that is driving the synchronization chain. The messages are also used for controlling, maintaining and restoring synchronization chains. The Ethernet synchronization messaging channel (ESMC) protocol has been created for specifying the signaling packet formats and the transmission channel of packets. Note that the SyncE method can only be used for distributing frequency synchronization (i.e. it cannot provide time or phase synchronization) because it can deliver only the frequency reference on physical layer along with data streams. In that sense, the performance is quite similar to the SDH/PDH-based synchronization. [1] [4]

There are some significant advantages and disadvantages in using Synchronous Ethernet. First, an advantage is that SyncE is independent of PDV which is causing challenges for ToP solutions. Furthermore, the frequency accuracy achieved with SyncE is very high. However, the downside is that every network elements in the synchronization path between master and slave must support SyncE (i.e. have hardware and software support). Moreover, the operating expenses of managing synchronization network based on the physical layer are typically higher than for a flat upper layer network. A ToP solution without on-path support is also less expensive since any support from the transport network is not needed. SyncE signal could be passed from the master node (that has access to the PRC traceable reference) to sites which handle, for instance, TDM or CES services. Obviously, it could be used for mobile backhaul synchronization as well. [2]

Figure 9 illustrates the simple architecture of SyncE. The chain of EECs delivers the timing on per link basis (i.e. ESMC is terminated after single link).

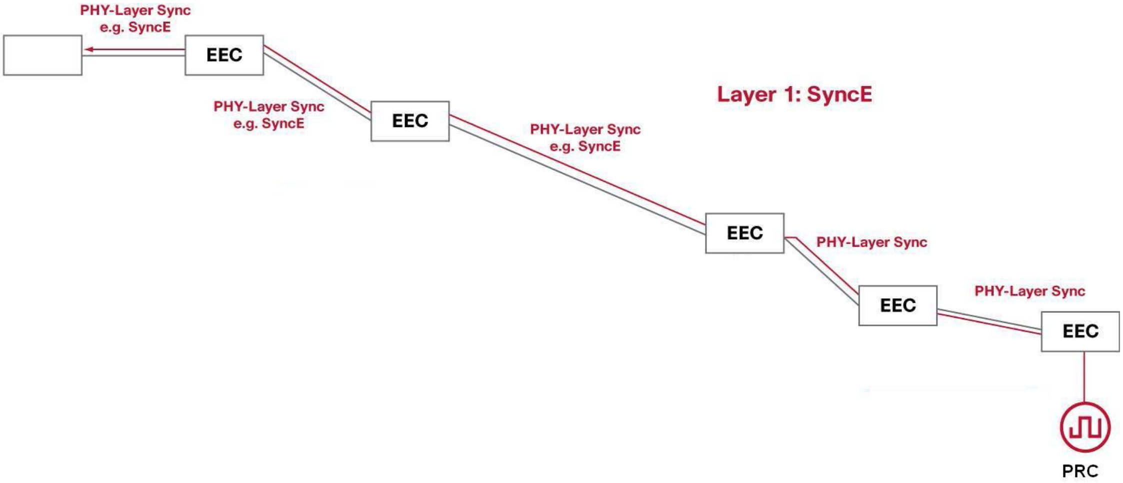


Figure 9: SyncE works on link-by-link principle. Timing signal is delivered along the chain of EECs. [1]

3.4 Global Navigation Satellite Systems

The global navigation satellite system (GNSS) is a generic name for satellite based navigation systems although global positioning system (GPS) governed by the United

States Department of Defence is the only fully operative system. The GPS system consists of about 24 satellites in orbit around the earth. Besides the actual location information (longitude, latitude, altitude), accurate timing reference can also be obtained with the GPS receiver. Each GPS satellite contains an atomic clock and the timing signal is constantly monitored by ground stations which can send back correction information if necessary. This procedure guarantees that the GPS system can provide timing accuracy as good as 100 nanoseconds and location within a few meters. GPS transmits additional information which makes it possible to translate the GPS time scale to UTC. Due to the high accuracy, GPS timing is often used to synchronize primary reference clocks (PRC) in telecommunication systems, NTP servers and cellular network base stations. [14]

In the context of base station synchronization, the downside of GPS timing is the need for an additional antenna and GPS receiver. The obvious advantage of GPS timing is high accuracy at least with a line-of-sight to the satellites. With a suitable oscillator (e.g. rubidium) one can average signal so that the frequency accuracy on the order of part in 10^{13} and time accuracy of few nanosecond can be achieved. The GPS system being subject to the control of US Department of Defense makes it a hard choice to rely only on that system. There are also some competing satellite systems (e.g. Galileo of the European Union, Compass of China and GLONASS of Russia) being assembled but nothing ready yet. [14]

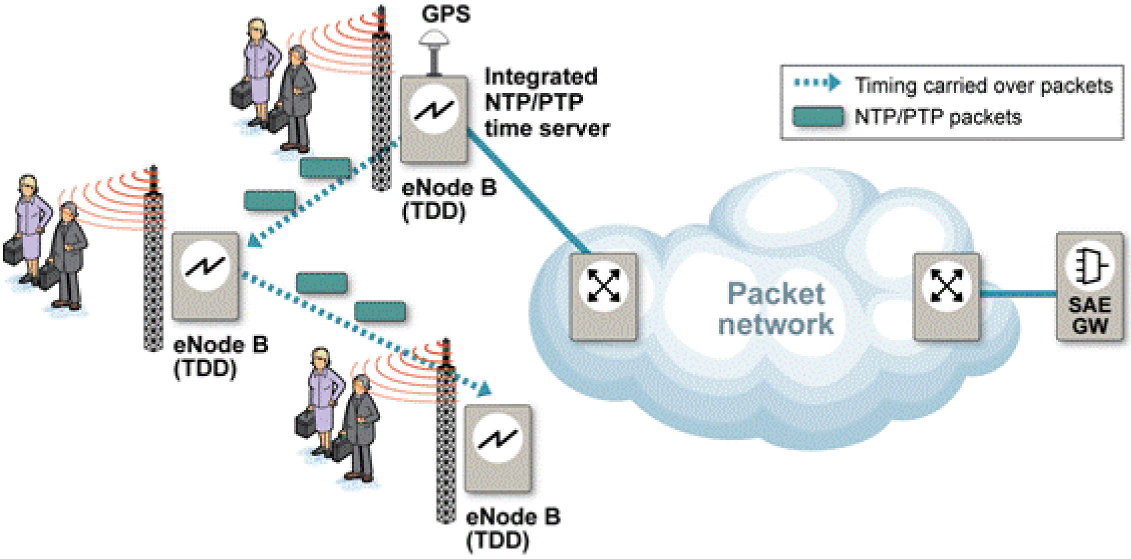


Figure 10: A possible scenario of GPS synchronization in mobile base station in conjunction with ToP solution. [2]

3.5 Adaptive Clock Recovery

Adaptive clock recovery (ACR) is a frequency synchronization method in which the synchronization is distributed over a packet-switched network as a constant bit rate (CBR) TDM pseudo wire stream. The clock is then recovered at the receiver end using the packets' time-of-arrival information independently of the physical layer. The clock stream format is a standard TDM pseudo wire flow. As one can see from Figure 11, adaptive clock recovery preserves the initial TDM timing over the packet network. [15]

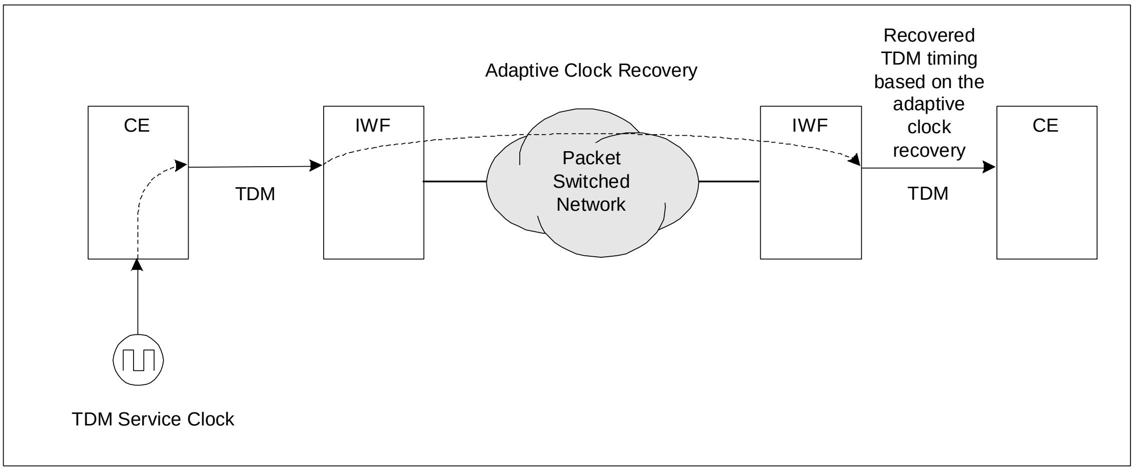


Figure 11: Example of adaptive clock recovery (ACR) in CES. TDM timing (frequency) is preserved over asynchronous packet network between two inter-working functions (IWFs) with ACR. [6]

3.6 Network Timing Reference

Network timing reference (NTR) is the frequency synchronization delivery method over DSL-based (digital subscriber line) last mile access. Hence, it is applicable in the situations where, e.g., the last mile of the mobile backhaul is implemented by the DSL technology. The reference clock can be distributed by mapping the frequency information to DSL modem transmission. The timing is delivered from the NTR-compatible DSLAM (Digital Subscriber Line Access Multiplexer) to the customer premise equipment which might be a base station. The advantages of NTR are high frequency synchronization accuracy and low-cost equipment. However, the PRC traceable reference signal is obviously needed in the DSLAM (digital subscriber line access multiplexer). [15]

4. ToP discussed in more detail

In this chapter timing over packet (ToP) solutions are discussed in more detail. The underlying technology, architecture and protocols are introduced, which is essential in order to understand synchronization metrics designed for packet networks. ToP synchronization offers a low-cost method for mobile backhaul synchronization which is also a clear business incentive for deploying the technology.

4.1 Business incentive

ToP solution introduces several benefits over other timing solutions for the mobile operators. First of all, the simple ToP solution eliminates the need for alternative mechanisms and equipments such as SyncE, GPS, or placing extremely stable and expensive oscillator at the receiving node. This obviously provides significant cost savings (CAPEX) for the mobile operators in terms of network equipment investments as well as possibly in installation and maintenance expenses (OPEX). Of course, ToP solutions need ToP-compliant network equipment at slave and master ends.

Actually, the year 2009 can be considered the breakthrough of ToP solutions due to technology development which has built up trust in ToP solutions although some service providers are still skeptical to the performance of packet synchronization. Nevertheless, the number of ToP synchronized mobile base stations is increasing rapidly. An important reason for deploying ToP synchronization has been the multiplication of data traffic in mobile networks forcing operators to start replacing their legacy TDM E1/T1 lines with less expensive and more flexible technologies (e.g. Ethernet). Increasing backhaul capacity with leased E1 lines produces more or less linear relation between capacity and expenses, which is intolerable for operators in the longer run [5].[3]

Ethernet is becoming the dominant mobile backhaul technology but synchronization can still be delivered with many different solutions as presented in the previous chapter [5]. Figure 12 shows that ToP solutions (PTP and NTP) are deployed rapidly in mobile backhauls due to the lower TCO for operators and service providers. PTP (denoted as IEEE 1588 in the figure) is expected to obtain the most significant market share of the Ethernet backhaul synchronization markets in coming years. The NTP and PTP protocols are discussed in more detail later. In addition to the synchronization technology market shares, Figure 13 illustrates a linear growth of the overall mobile backhaul market. The mobile backhaul market is growing due to the evolution to 4G capacities and new mobile network deployments in developing countries and rural areas. Thus, from the ToP point of view, the combination of market leader in synchronization technology and growing overall backhaul market is without a doubt a good business opportunity.

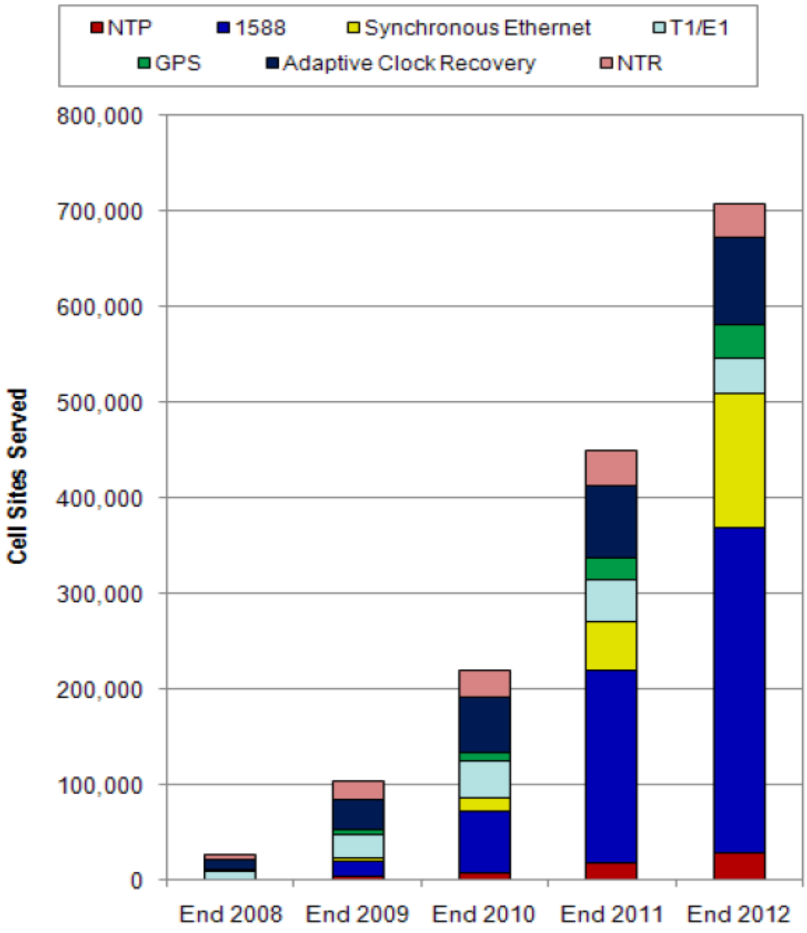


Figure 12: Ethernet backhaul sites synchronization technology forecast. IEEE 1588 will be dominant option in near future. [16]

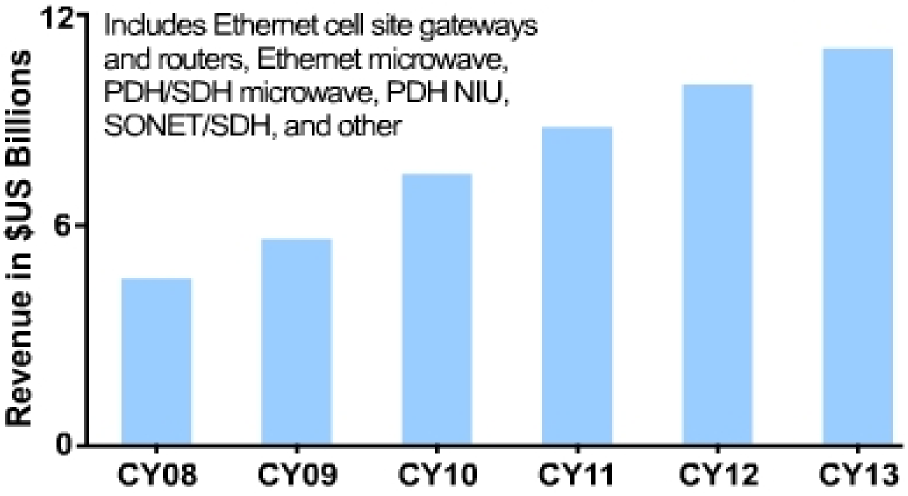


Figure 13: Mobile backhaul market is booming and growing more or less linearly year by year.[17]

4.2 Packet timing protocols

Several protocols have been created for timing purposes but the most widely used network time protocol (NTP) and precision time protocol (PTP) are introduced here.

4.2.1 Network Time Protocol

The previous network time protocol (NTP) version specified in RFC 1305 (NTPv3) by IETF (Internet engineering task force) is undeniably the most widely used synchronization protocol. The newest version 4 of NTP is recently completed and it is specified in RFC 5905 [18]. NTP was developed to synchronize clocks by messages (i.e. NTP packets) over the Internet. The nominal accuracy of NTP is in the level of tens of milliseconds in WAN (wide area network) and less than a millisecond in LAN (local area network). Its hierarchical client-server architecture consists of four stratum levels (0-3 levels). The top Stratum 0 consists of high accuracy reference clocks (i.e. the source clocks for accurate timing, usually access to atomic clock). Stratum 0 nodes act as servers for the Stratum 1 nodes (primary NTP servers). There are over 200 primary NTP servers located around the world. Stratum 3 is the end-user level, e.g. a home computer. NTP is so widespread that almost every computing platform and operating system understands it. The NTP standard specifies also algorithms used for recovering the synchronization from the messages. NTP filtering algorithms are used to reduce network-induced jitter and oscillator wander. Figure 14 illustrates the NTP implementation; one can see that NTP can use several servers for a reference and combine the signals to a single applicable timing reference. [14]

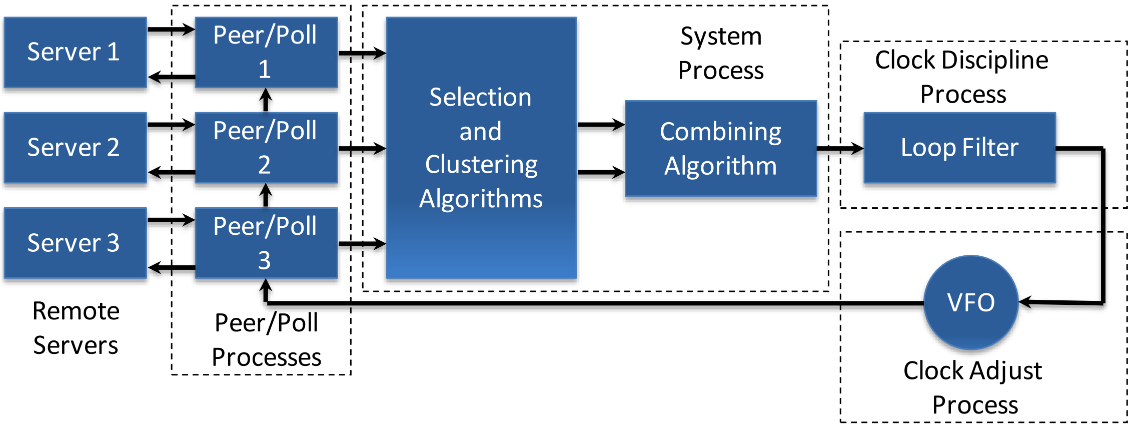


Figure 14: NTP block diagram describes the function blocks of the NTP client implementation. Reference can be a combination of several NTP server reference signals. [19][20]

4.2.2 IEEE 1588 Precision Time Protocol

IEEE 1588 is more recent than NTP and has received attention in telecom synchronization, although, the IEEE standard 1588 was not originally designed to provide accurate synchronization for telecommunication networks. The second version IEEE 1588-2008

(IEEE 1588 version 2 name is also used) was completed in 2008 to fit in better with telecommunication network synchronization purposes.

With PTP, one can achieve slave synchronization accuracies in the nanosecond or microsecond range, whereas, with NTP one can normally obtain client synchronization accuracies in the millisecond range. PTP's superiority is due to the specified timestamp point and on-path support specified in the standard. PTPv2, as opposed to PTPv1, has important additions for telecom purposes. These include unicast message transmission, shorter frames, and higher message rates. [21]

PTP specifies several message types. Messages are divided into two categories; event messages and general messages. Event messages are timestamped messages that need to be generated in both the receiving and the transmitting end. The instance when the timestamp point passes in or out of the interface is accurately recorded. General messages do not require timestamps but may contain one. Table 2 lists all the message types of PTP.

Table 2: IEEE 1588v2 message types are divided in two categories.[9]

Event messages	General messages
Sync	Announce
Delay_Req	Follow_Up
Pdelay_Req	Delay_Resp
Pdelay_Resp	Pdelay_Resp_Follow_Up
	Management
	Signaling

PTP operation can be implemented with or without on-path support (i.e. support from intermediate network elements). IEEE 1588 version 2 defines five different types of clock devices; ordinary clock, boundary clock, end-to-end transparent clock, peer-to-peer transparent clock and management node. Ordinary clocks, either grandmaster or slave, are the end points of synchronization trails. Boundary and transparent clocks are the elements that provide the on-path support for the end nodes, thereby ensuring better performance. [9]

Boundary clock is an intermediate node which can work as a master and a slave, i.e. it terminates and regenerates the packet timestamps which improves accuracy. Figure 15 illustrates the basic usage scenario of the PTP boundary clocks. Master-slave architecture is constructed hop-by-hop if all the intermediate nodes between the end-points are boundary clocks. Therefore, there is no end-to-end protocol message flow when using the boundary clock operation of PTP. [9]

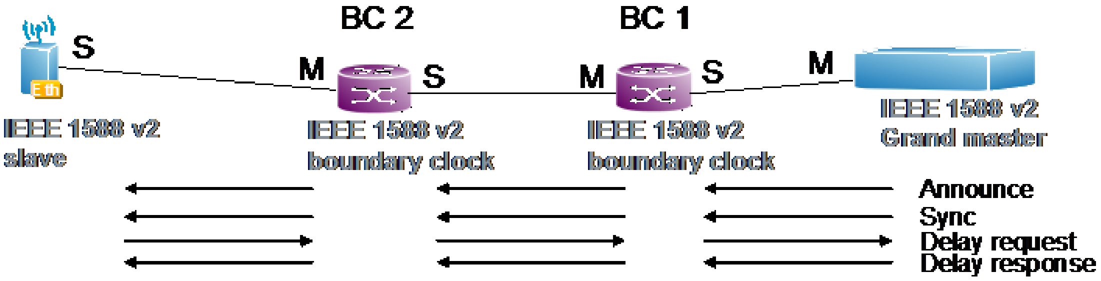

BC 2
BC 1
S
M
S
M
S
M
Eth
IEEE 1588 v2 slave
IEEE 1588 v2 boundary clock
IEEE 1588 v2 boundary clock
IEEE 1588 v2 Grand master
Announce
Sync
Delay request
Delay response

Figure 15: Precision time protocol boundary clock operation model in distributing time synchronization. Each BC act as master for next hop and slave for previous hop.

Another alternative for on-path support, transparent clock, can be neither a master nor a slave but a bridge between the two. Moreover, according to its name, it is basically transparent to the PTP messages with the exception that it corrects the timing messages by adding processing time (residence time) into a correction field of packets while forwarding them. There are actually two types of transparent clocks; end-to-end transparent clock and peer-to-peer transparent clock. The end-to-end transparent clock operation adds only the residence time of packet to the correction field, which does not require any hop-by-hop message transmission. Instead, peer-to-peer transparent clock adds also link delay correction information to the header of timing packets which obviously requires peer-to-peer message transmission. [9]

Precision time protocol operation has normally two phases. First, the master-slave hierarchy is established. Second, the synchronization process begins with a specified mechanism. For example, the most simple mechanism for time synchronization is first to determine the mean propagation delay between master and slave after which only one time stamped sync message is needed to make the clock corrections at the slave. The principle of propagation delay determination is presented in Figure 16. First, the slave receives the *Sync* packet and stores its accurate arrival time (t2). If the grandmaster is able to determine the accurate transmission instance only after the *Sync* has been sent, it sends a *Follow up* message containing the precise timestamp. *Delay request* is then sent back to the master that eventually responses with the *Delay response* message which contain the reception time of the request packet. The t3 timestamp records the sending time of *Delay response*, and respectively, t4 the receiving time at the master. The t3 timestamp in the *Delay request* message is not necessarily precise for the same reason as in the *Sync* message. However, after the message has been sent the slave will know the precise t3. After the sequence of four packets, the slave should know four timestamps which can be used in the mean propagation delay (mpd) calculations according to the simple Equation (1). [22]

$$mpd = \frac{(t2-t1)+(t4-t3)}{2} \qquad (1)$$

One should note that, RTT/2 is an approximation of the real one-way delay. Asymmetric delay, for example, affects the accuracy of time synchronization. On the other hand, time synchronization can be delivered easily when the one-way propagation delay is known for the receiver. The one-way propagation delay is summed to the time reference delivered by the timestamps inside the Sync packets.

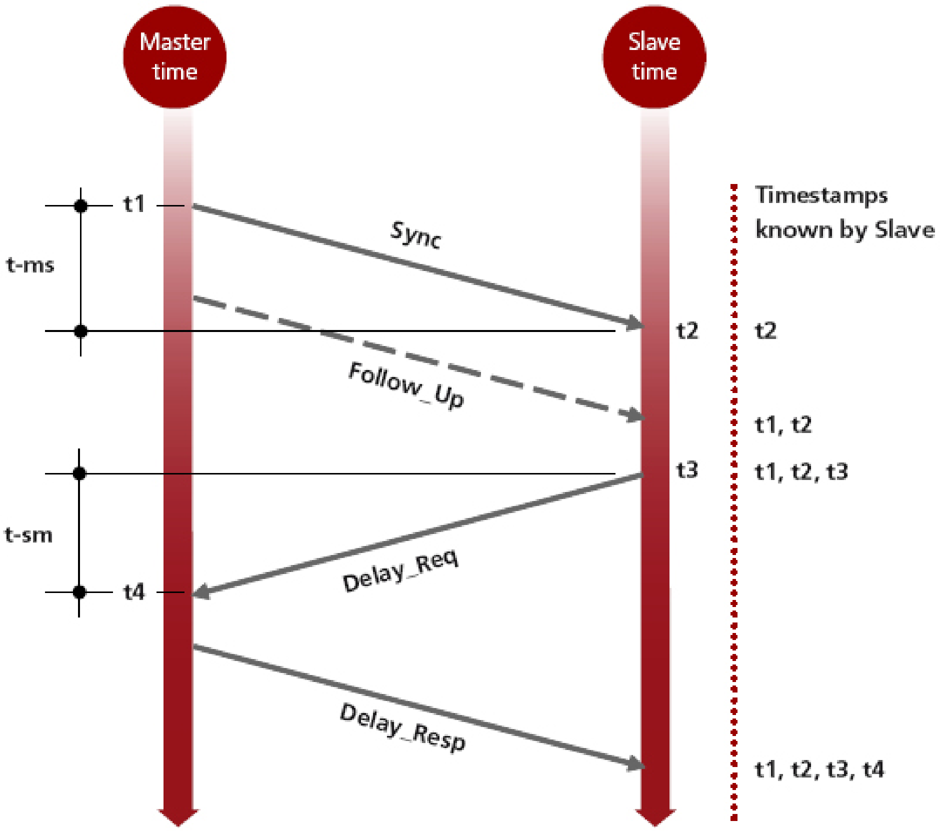


Figure 16: Basic propagation delay determination method using Delay Request-Response mechanism. The mean propagation delay cannot be calculated until the slave knows all the four timestamps (t1, t2, t3, t4). [9][23]

In PTP, there is a state machine at each port of a Boundary clock and an Ordinary clock. State machines use the BMCA (best master clock algorithm) to determine the best quality master for each slave port. Comparison can be done based on the information in Announce messages received from master clocks. [23]

Furthermore, the protocol is also quite flexible in terms of messages transportation. PTP messages can be transported over several different protocols. For example, PTP over UDP over IPv4/IPv6 or PTP over Ethernet plus a few other combinations are currently possible. [9]

4.3 Frequency synchronization without on-path support

The previous section introduced the protocols capable for accurate two-way timing distribution. Further, it described on-path support that has been specified for PTP for removing inaccuracies caused by queuing and other delays in network nodes. However, since the current packet environments do not actually provide on-path support and the frequency synchronization is the main requirement, time synchronization is not discussed anymore until Chapter 10. Moreover, for the sake of simplicity, hereafter only the one-way frequency (and phase) synchronization delivery without on-path support is investigated. Though, the two-way timing methods could be implemented with the same principles.

Figure 17 illustrates the basic ToP frequency synchronization delivery scenario in mobile backhaul networks. Timing packets are delivered from the timing server to the base stations over a packet network, which induces packet delay, jitter and loss. [24]

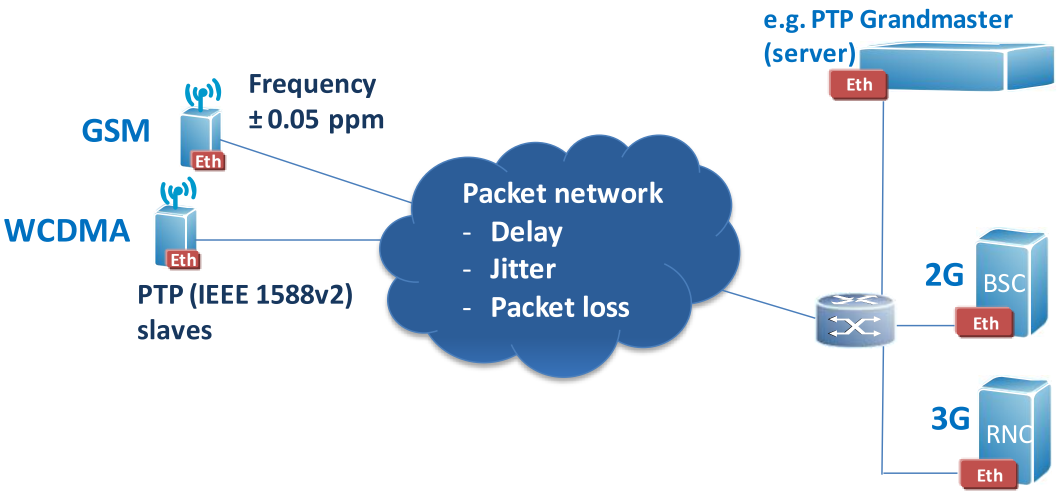


Figure 17: PTP as a frequency synchronization distribution method over Ethernet mobile backhaul. This kind of synchronization architecture is becoming more and more widespread.

Master typically sends the timing packets (with timestamps) to the slave at an average rate of 16 or 32 pps (packets per second) but not necessarily at fixed intervals. The receiver records the time when packets are received after which the local interval between two received packets is compared to the difference between the timestamps. Figure 18 illustrates the basic principle of distributing frequency synchronization over packet networks. The time stamps inside the consecutive packets are compared with local arrival times. Hence, in an ideal situation one can determine the frequency error of the local clock compared to the reference by sending two packets. In reality PDV makes it more challenging. Figure 19 depicts a simple disruptive situation caused by PDV where the slave mistakenly assumes that the frequency error is large. Therefore, averaging or packet selection methods are needed for stable performance. These issues are discussed in more detail later.

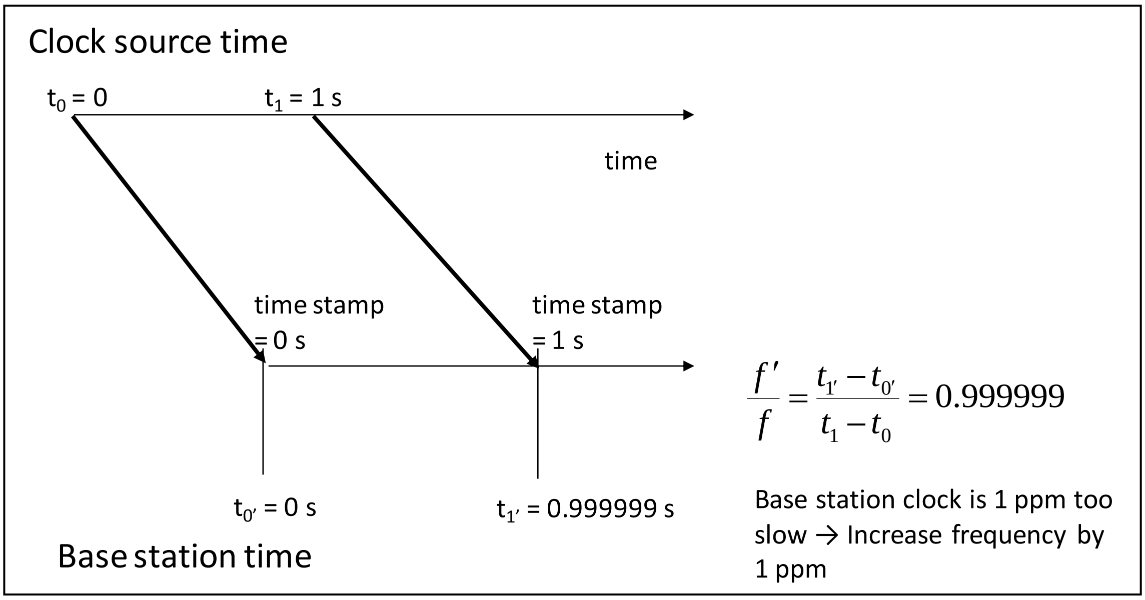

Clock source time
$t_0 = 0$
$t_1 = 1$ s
time
time stamp = 0 s
time stamp = 1 s
$\frac{f'}{f} = \frac{t_{1'} - t_{0'}}{t_1 - t_0} = 0.999999$
$t_{0'} = 0$ s
$t_{1'} = 0.999999$ s
Base station time
Base station clock is 1 ppm too slow → Increase frequency by 1 ppm

Figure 18: Principle of frequency synchronization over packet networks. Local time scale is compared with the time scale delivered in packet time stamps. In theory, two received packets are sufficient to make adjustments to the local clock.

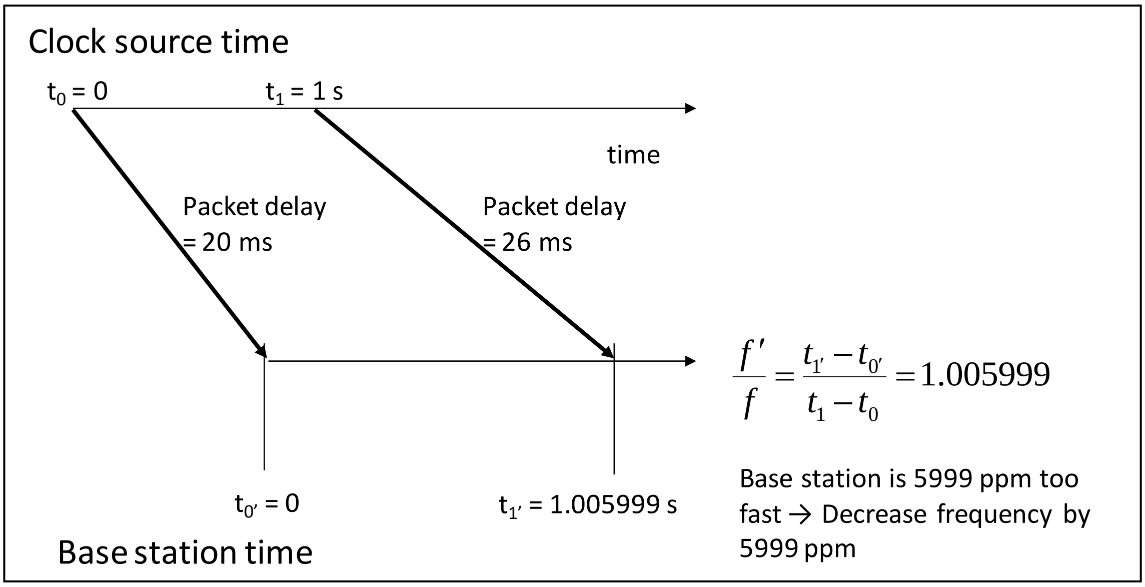


Figure 19: Effects of packet delay variation for simple frequency synchronization. The first timing packet experiences a delay of 20 ms and the second packet a delay of 26 ms. Therefore, the receiver node (e.g. base station) assumes that its own clock runs significantly faster that it should decrease the frequency by 5999 ppm (although the clock is actually 1 ppm too slow).

4.4 Challenges in packet networks

The basic principle of a packet network is to provide best effort (BE) service for every packet. Essentially the same resources are shared with every user in the network. Since a single path is shared among many streams and network elements are using the same queues to temporarily store the data before transmitting it to the same link, it is obvious that packets experience delay variation (i.e. jitter and wander) and possibly even packet loss. In addition, propagation and processing time in the network elements create the absolute minimum delay for traversing packets. Loss of a packet is an extreme situation which can happen, for instance, if queues are full, bit errors happen, etc. By definition, PDV means only the delay variation experienced by packets, not the whole packet delay. But due to the fact that only the delay variation is significant for determining the performance of ToP solutions, PDV data is used hereafter to refer to the actual packet delay data as a source of delay variation. Furthermore, PDV is a widely recognized term to describe the variation of packet delays as a generic concept and a challenge for synchronization rather than a mathematical term related to the variance of the packet delay distribution.

Although packet delays are dependent on time and other traffic, the queuing time for each packet is more or less independent of previous packets. Consequently, end points can perceive this PDV as randomly changing over time. Thus, in some network conditions the Gaussian probability distribution function (PDF) with certain mean and variance depicts closely the distribution function of a measured PDV data. [25]

Figure 20 depicts artificially created Gaussian distributed PDV data histogram (the red area) together with a real network delay distribution (specifically, a national backbone at ~30% load and ADSL as the access technology) in normal conditions. The latter histogram is shifted so that the minimum delays are equal. The real delay distribution resembles little bit a low variance Gaussian distribution. Practice has shown that relatively high load and connections with large number of hops create approximately Gaussian delay distribution.

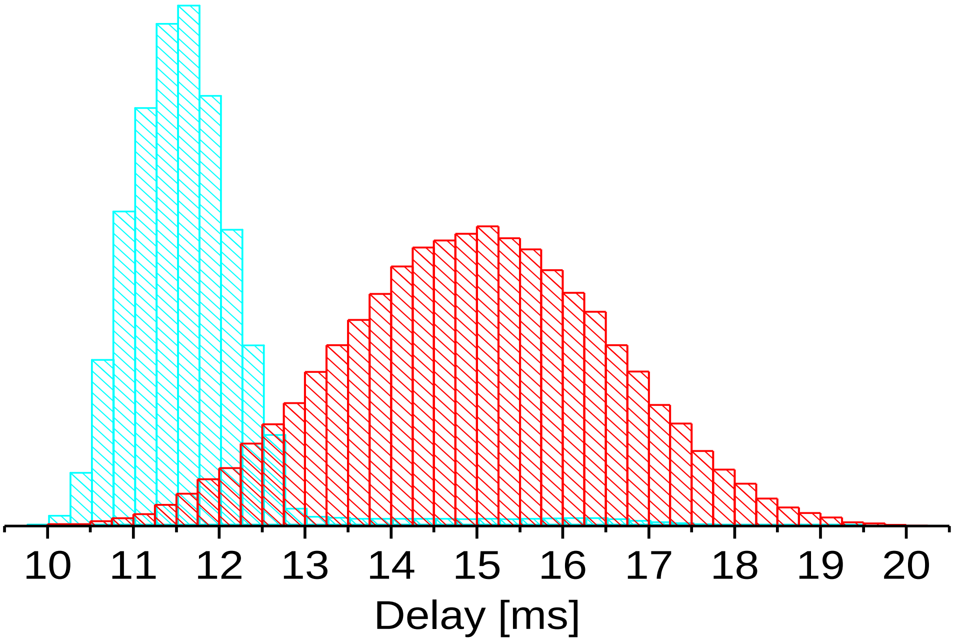


Figure 20: Example of Gaussian distributed packet delay data (PDV). The red area is the artificially created histogram of the Gaussian distributed delay variation and the blue area is the histogram of a real PDV measurement over a real production network. However, the real data is shifted approximately 11 ms to the left to match the minimum delays. Therefore, the mean values are not comparable, only the difference in variance is visualized. [26]

Figure 21 shows that, although, the PDV data distribution changes over the measurement due to different load scenarios, the minimum delay stays very stable throughout the measurement in the network setup in question. More specifically, the distribution of the first load scenario resembles approximately the symmetric distribution, because the mean delay is approximately in the middle of the delay variation. But clearly in the second phase, distribution is asymmetric, leaning more towards the minimum delay. As an explanation, in that particular 48 hour measurement, three different load scenarios were used over the ~10 Mbit/s SHDSL access link. In the first third of the measurement period the load was over 90% of the link capacity. In the second third the load was significantly reduced, and the last third was basically without any additional load, only the probe packet streams. It can be easily noticed that the minimum delay curve (the black curve) of small probe packets stays quite stable during the whole measurement. This supports the theory of small portion of packets traversing through the network almost without queuing delays. This fundamental discovery plays a significant role in ToP solutions and, consequently, in packet synchronization metrics discussed later.

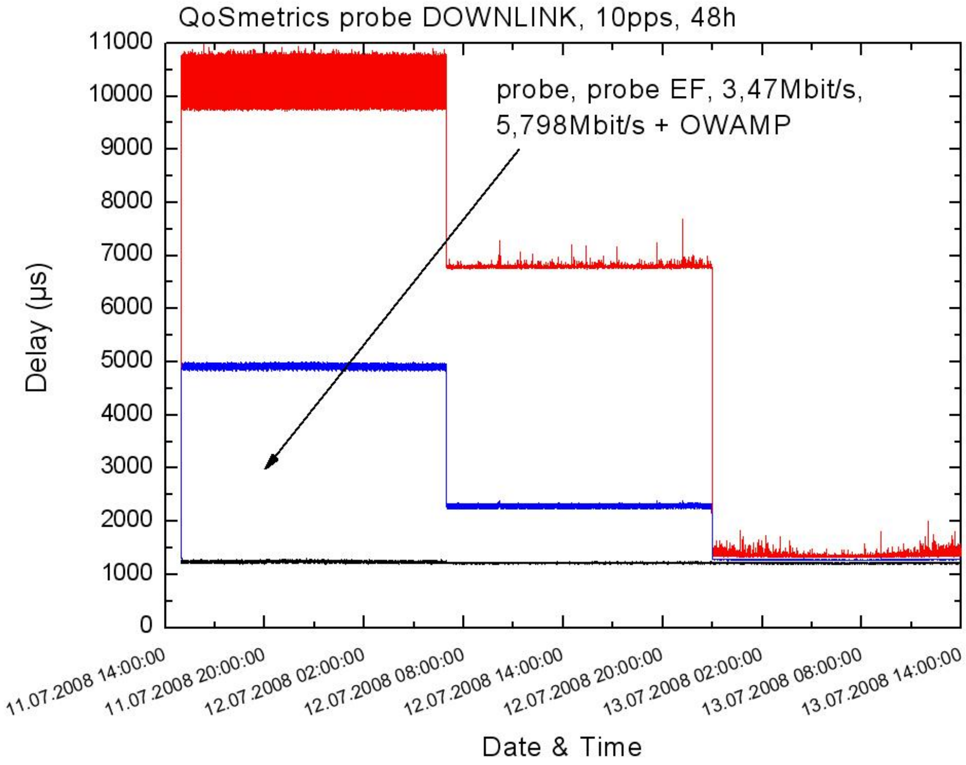


Figure 21: 48 hour measurement which shows the stability of minimum delays. Black curve depicts the minimum delay value, blue is mean and red is maximum. This particular measurement was conducted over operator's production network with ~10 Mbit/s SHDSL access network link as a bottleneck. Probe packets were sent at the rate of 10 packets per second (pps). The delays induced by the production network itself can be observed best in the last third.

In extreme load situations, the load of a certain link increases up to 100% where all packets have to queue. Hence, the minimum delay jumps as a result. This problem can be mitigated with prioritization of timing packets. For example, expedited forwarding (EF) solves the problem relatively effectively. This has been proven with real measurements and, thus, Figure 22 presents appropriate proof of the effectiveness of EF quality of service (QoS) class. EF is a PHB (per-hop behavior) protocol for differentiating services in the network. Prioritization class information can be carried inside the IP header in the 6-bit DSCP (differentiated services code point) field (e.g. the recommended EF code point binary value is 101110). Furthermore, EF is specified in RFC 3246 by IETF. [27]

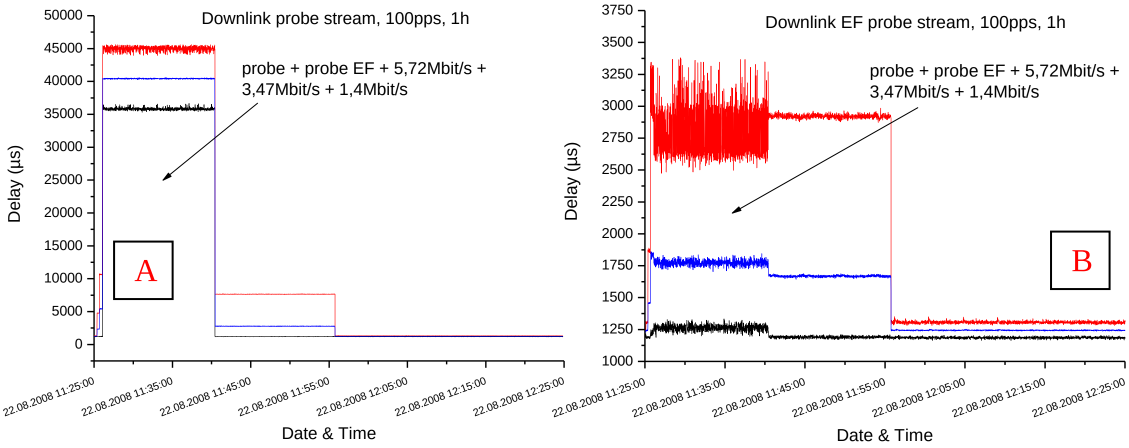


Figure 22: A three phase (load scenarios) measurement similar to the previous figure. The A figure depicts best effort probe stream delays and the B is the packet delay graph of simultaneous expedited forwarded (EF) probe stream. Significant difference in delays can be observed and most importantly for packet synchronization, the minimum delay stays relatively stable although the link is overloaded, i.e. without prioritization the minimum packet delay is around 35 ms whereas with EF it is approximately 1.2 ms.

All in all, when service providers are considering ToP solutions for delivering synchronization the SLA (service level agreement) parameters must be defined so that the ToP performance will be adequate. Metrics are designed to estimate the performance of the timing from the PDV data measured from the network. Thus, if metrics does not indicate required synchronization performance, the SLA parameters must be tightened or the network structure must be altered.

4.5 Measuring PDV

After explaining the important PDV concept, the process of obtaining PDV data is explained briefly. The packet delay measurement can be done either actively or passively. In active measurement a probe packet stream is injected into the data stream at the master location. Another device receives the probe stream for determining one-way delay. When measuring a packet slave clock performance at the same time it is more convenient to have passive PDV measurement because only one measurement device is needed. Figure 23 depicts a simple passive PDV measurement where there is a separate device (with GPS access for accurate time reference) for recording arrival time instances. The instances are compared with the timestamps to determine PDV. In the context of mobile backhaul, the actual PTP slave could be, for instance, a base station. [28]

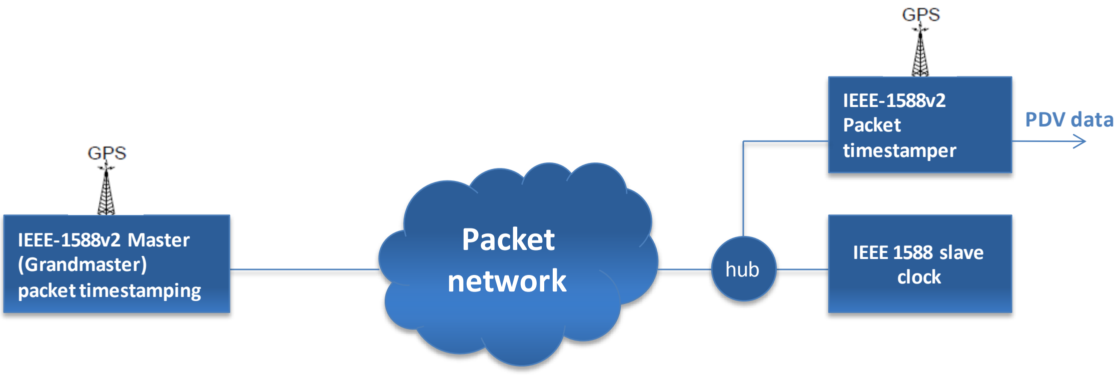

GPS
IEEE-1588v2 Master (Grandmaster) packet timestamping
Packet network
hub
GPS
IEEE-1588v2 Packet timestamper
PDV data
IEEE 1588 slave clock

Figure *23: The principle of accurate passive PDV measurement. Note: GPS timing at both ends is required in order to get delays with the accuracy of micro seconds or less. Packets have to be duplicated to the separate timestamper that records the delay variation.*

4.6 Clock recovery algorithms and slave architecture

To be able to understand ToP or any other synchronization distribution concept, one should know the principles of the slave architecture and clock recovery algorithms. The clock recovery algorithms are the most essential part of the slave clock implementation. There are neither standards nor rules how to implement the algorithms in practice, although, the NTP protocol specifies a clock algorithm commonly used when NTP is implemented. However, e.g. PTP (IEEE 1588v2) implementations are up to the vendors' own specifications.

In general, the heart of the slave clock is the oscillator producing frequency output which is phase-locked to the reference timing signal by PLL. The oscillator can only be characterized by measuring it against a reference clock, the errors of which are negligible with respect to the oscillator under test. Frequency drifting caused by environmental and inherent effects is due to changes in temperature, pressure, supply voltage and aging of oscillators. The environmental effects are well understood and compensated but there are also random variations involved. Metrics like TDEV, MDEV and MTIE introduced in the next chapter are common methods used to characterize oscillators' random variations. Today's oscillator technologies can be divided into two categories: quartz crystal resonances and rubidium, cesium and hydrogen atomic resonances. Quartz oscillators are based on mechanical vibrations of quartz crystal. Crystal oscillators (designated as XO) are the most widely used oscillators in timekeeping, and also in telecommunication applications with some compensation for external effects (e.g. TCXO and OCXO are compensating against temperature changes). The second category, rubidium, cesium and hydrogen oscillators are based on atomic resonances which makes them less susceptible to environmental effects. [14]

In the context of the packet slave architecture and ToP solution, the reference timing signal for local oscillator is transmitted over packet networks in the form of time stamps. For example, IEEE 1588v2 timestamps are processed at the input interface and time difference between the time stamp in packet and local clock is measured in a time comparator. Time differences corresponding to the fastest packets from a selection window are selected. The time differences are interpreted as a time error signal, which is then low-pass filtered to average out the short-term variation. The cut-off frequency of the PLL filter must be carefully selected to achieve adequate performance at the clock output. However, clocks cannot filter sudden jumps in input timing signal caused by e.g. route changes in timing packet distribution (i.e. the timing packets experience delay jump). Thus, clock recovery algorithms must be designed to neglect the large jumps in interpreted time error signal.

Figure 24 illustrates the architecture of a slave clock implementation. Cut-off frequencies of the PLL low-pass filters (i.e. -3dB frequency) that are used in reality are in the order of a mHz.

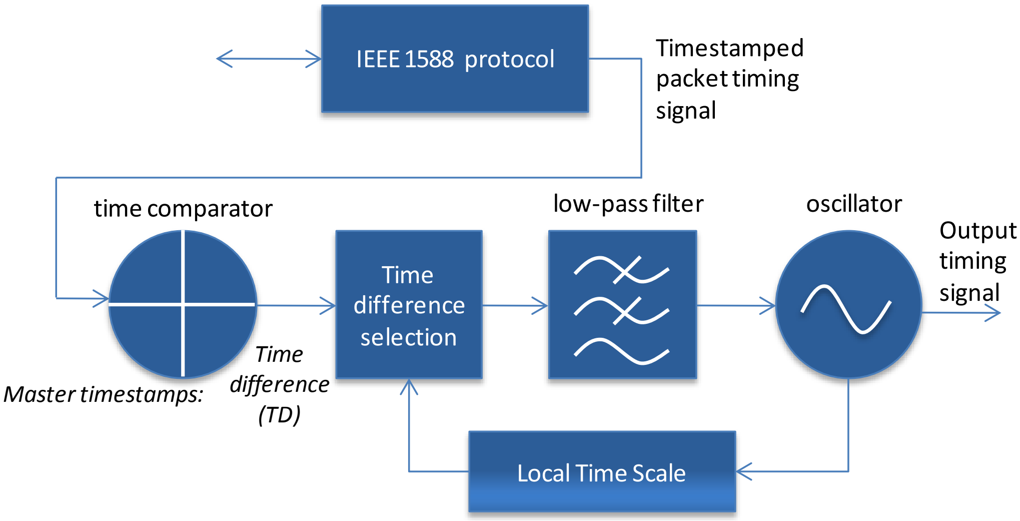


Figure 24: PTP packet slave clock block diagram at synchronization end point (e.g. at base station).

Another significant clock property is the holdover which is the ability of a clock to provide accurate timing in temporary absence of the reference timing, i.e. the clock is neither controlled nor adjusted during holdover. The feature is important in case of failures or interruptions in the network providing the timing signal. The better the oscillator the longer it remains stable without being adjusted. Without good holdover capabilities (stable oscillator), in case of link failure, low-cost slave clocks could start to drift and exceed specified limits rather soon. This would result in deteriorated quality of service. On the other hand, slave clocks with stable oscillators can provide an acceptable timing signal during outages of up to several hours or even days in e.g. cellular base stations. Holdover performance is a key property in achieving high service quality and availability. [24]

Generally, the PBCR (packet based clock recovery) algorithm is a term that includes the packet selection process and PLL mechanisms (i.e. low-pass filtering) related to the recovery of accurate timing from the reference timing signal carried by packets. Metrics introduced in the next two chapters help to determine and qualify the design parameters for clock recovery algorithms. Information can be used to develop better (i.e. more stable) or more optimal slave architectures and algorithms.

5. Introduction to metrics for estimating packet synchronization performance

This chapter discusses synchronization metrics which are mainly used to predict the stability (i.e. performance) of the packet slave clock and, in addition, to help in the design process of the slave clock that meets the strict requirements. First, concepts of time error (TE), time interval error (TIE) and the traditional clock metrics are introduced, since they are the basis for the packet synchronization metrics as well. Masks (i.e. the tolerable jitter and wander as a function of observation window) created for synchronization metrics compose a strong and relatively simple performance verification method.

5.1 TE and TIE concepts and measurement

The Telecommunication Standardization sector of ITU (International Telecommunication Union) specifies the concepts for measuring digital network synchronization in recommendation G.810. The time error function (denoted as *x(t)*) of a clock at certain time instant can be expressed as the difference between time T(*t*) generated by the clock and time $\mathrm{T_{ref}}(t)$ generated by a reference clock (see Equation (2)) [29].

$$x(t) = \mathrm{T}(t) - \mathrm{T_{ref}}(t) \qquad (2)$$

The time interval error (TIE) function is the difference between a measure of time interval of the clock and time interval of the reference signal. That definition can be simplified to the difference between the time error at starting point and the time error after certain interval. The mathematical definition of the TIE(*t*;τ) function at interval τ specified in G.810 is presented in Equation (3) [29].

$$\mathrm{TIE}(t;\tau) = [\mathrm{T}(t+\tau) - \mathrm{T}(t)] - [\mathrm{T_{ref}}(t+\tau) - \mathrm{T_{ref}}(t)] = x(t+\tau) - x(t) \quad (3)$$

Even though TIE is by definition different compared to TE, they are closely related. For instance, when assuming the initial time error to be zero ($x(t_0) = 0$), TIE$(t=t_0,\ \tau)$ as a function of τ is the same as the time error *x(t)* as a function of *t*. Therefore, TIE is the generic term used hereafter to measure the difference between the time interval of the clock and the time interval of a reference clock. Since several clock metrics can be calculated from TIE/TE data, it is required to understand the concepts of TIE or TE and how to actually measure it.

The TE or TIE measurements require always an external reference clock to be able to compare the clock under test (CUT) phase to the reference. Thereafter the measuring involves only calculating the difference between the rising edges of the intervals of two signals (i.e. the phase difference). Figure 25 illustrates the rather simple synchronized clock TIE/TE measurement, and Figure 26 shows the measurement principle of TIE in the independent clock case.

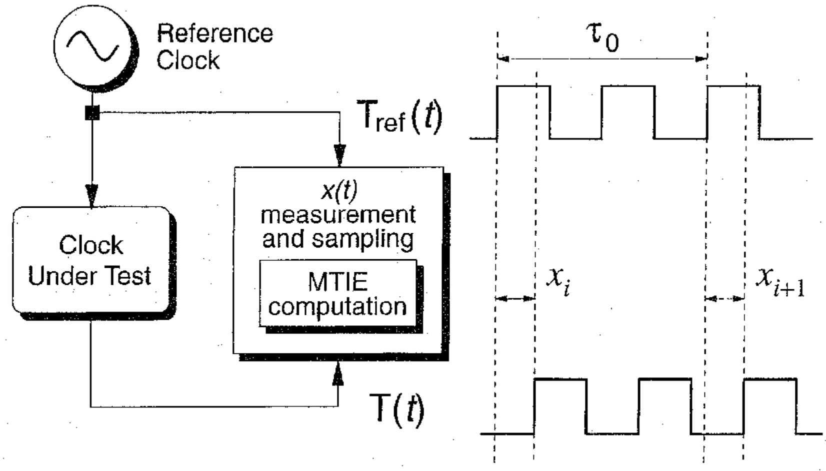
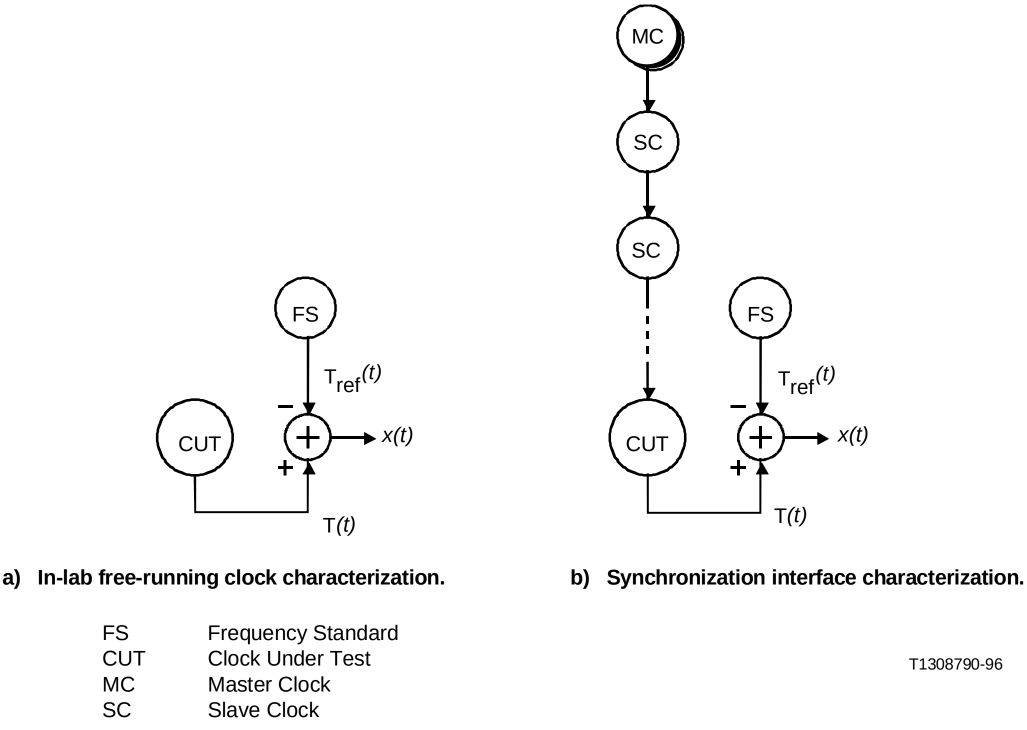


Figure 25: The principle of TIE measurement of a synchronized clock (i.e. the reference clock is generating the input timing signal of the clock as well as the reference timing signal). The phase (or time) error of two signals is calculated after every interval generating the TIE function. [30]



a) In-lab free-running clock characterization.

b) Synchronization interface characterization.

Figure 26: Independent clock TIE measurements. For example, in packet synchronization cases the timing for the local clock (e.g. base station local clock) comes usually through multi-hop timing path and external reference is needed for the TIE measurements. [29]

All in all, TIE simply shows the phase error of the slave clock compared to the reference clock (usually PRC traceable clock). Moreover, the frequency error between the local clock and the reference clock can be also derived from TIE by calculating the slope of the curve.

5.2 Traditional clock metrics - MTIE, TDEV

The metrics described below have been developed to specify design objectives and verify the performance of digital networks such as PDH and SDH networks where synchronization of the network is closely related to the actual performance of the network. Different errors occur easily if timing of the equipment is not accurate. [29]

Maximum time interval error (MTIE) is probably the most widely recognized and accepted synchronization metric. It finds the maximum peak-to-peak time error of the signal within an observation period. MTIE is easily understood as a sliding window function which searches for the maximum error within the sliding window. Consequently, MTIE calculation is normally used for determining the worst case scenario of the phase error of the clock. According to ITU-T recommendation G.810, MTIE is a function of observation interval ($n\tau_0$). The estimation formula is presented in Equation (4). [29]

$$\mathrm{MTIE}(n\tau_0) \cong \max_{1\le k\le N-n}\left[\max_{k\le i\le k+n} x_i - \min_{k\le i\le k+n} x_i\right], \quad n = 1,2,\ldots,N-1 \qquad (4)$$

Where:

x_i are time error samples;

k is the index for sliding observation window over the data;

N is the total number of samples;

τ_0 is the time error sampling interval;

n is the number of sampling intervals within the observation interval τ;

Furthermore, MTIE, by definition, measures the maximum peak-to-peak phase delay variation (i.e. time error between the local time and reference time) of a given signal with respect to an ideal reference signal. Thus, there is also MRTIE (maximum relative time interval error) specified in the G.810 recommendation which is the same metric with the exception that the maximum time interval error is calculated between the clock's output timing signal and a given input timing signal. [29]

Figure 27 depicts the calculation principle of the MTIE calculation.

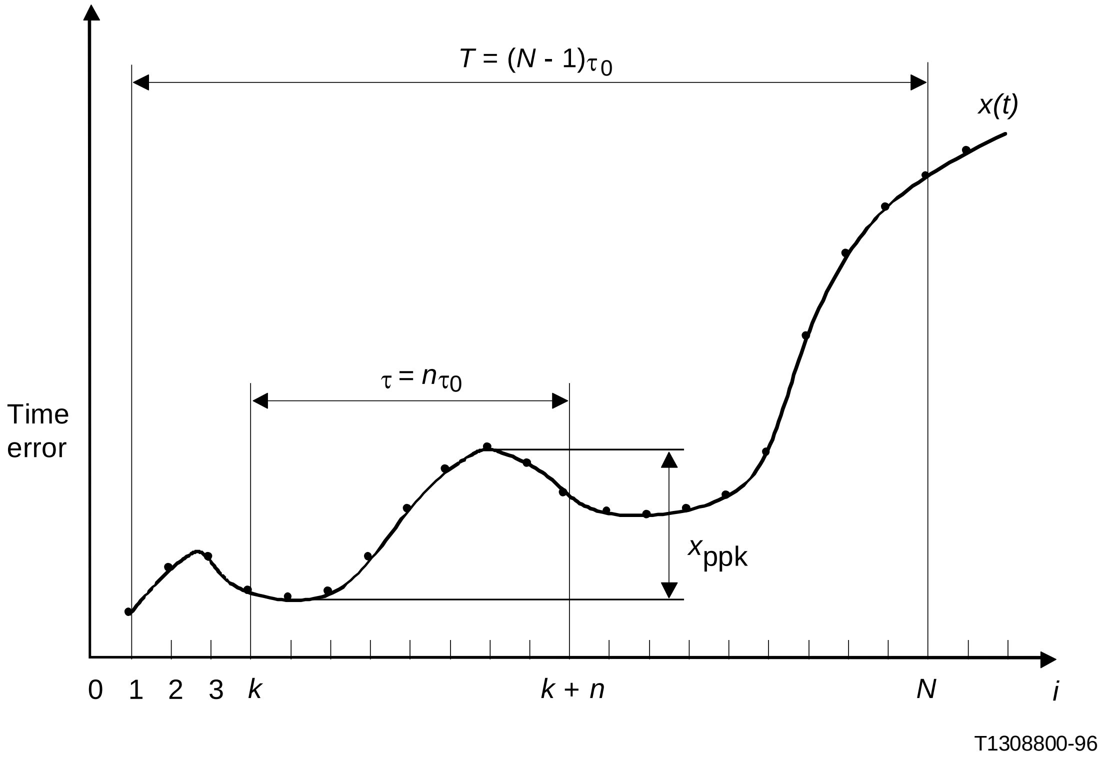


Figure 27: MTIE calculation example: y-axis is TE value (i.e. essentially the same as TIE with the assumption of the initial phase error being zero) and x-axis represents TE sample numbers. Observation window (τ) is sliding over the whole data step by step, one sample at a time. At each window position, maximum difference between minimum and maximum TE values (i.e. the peak-to-peak value within observation window [x_{ppk}*]) is searched and MTIE result at certain observation window is the maximum momentary difference (i.e. MTIE(τ) is the maximum *x_{ppk}* for all observations of length τ within T). In order to get the complete MTIE data as a function of observation window, the window size is increased after previous window has slid over the whole data. [29]*

Besides MTIE, the time deviation (TDEV) is another traditional metric for digital networks. It measures the expected signal time variation as a function of observation time (τ) correspondingly to MTIE. Thus, TDEV and MTIE are usually plotted as a function of observation window sizes (i.e. τ on x-axis). TDEV is a sliding window metric which uses three cascaded windows which are slid over the data. The TIE values inside each window are averaged and another averaging operation is conducted between the three sliding windows. The same procedure is repeated at different window sizes (τ).

TDEV is a phase stability metric which neglects the constant frequency offset of the clock compared to the reference (i.e. TDEV is zero in case of linear slope in time error of a clock). Based on the sequence of TE samples, estimated time deviation TDEV($n\tau_0$) is defined in Equation (5) and the principle of TDEV is elaborated in Figure 28.

$$\mathrm{TDEV}(n\tau_0) \cong \sqrt{\frac{1}{6n^2(N-3n+1)} \sum_{j=1}^{N-3n+1} \left[\sum_{i=j}^{n+j-1} \left(x_{i+2n} - 2x_{i+n} + x_i \right) \right]^2} \qquad (5),$$

n = 1, 2, ..., integer part (N/3)

Where:

x_i are time error samples;

j is the index to slide the windows;

N is the total number of samples;

τ_0 is the time error sampling interval;

n is the number of sampling intervals within the integration time τ;

τ $(n\tau_0)$ is the integration time, the independent variable of TDEV; Note, even in the case of TDEV, the term observation window will be used throughout this thesis to maintain consistency with the term used in MTIE.

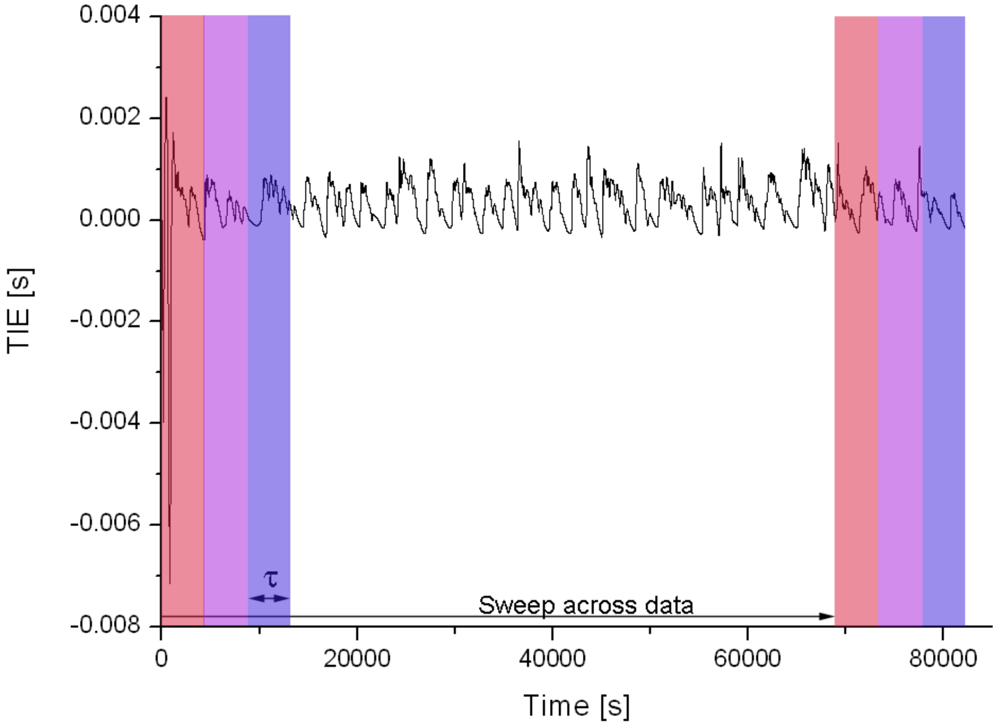


Figure 28: Principle of TDEV's sliding windows. Three consecutive same sized windows (τ-window) are summed together with weights (1, -2, 1) , are slid over the TIE data and finally an averaging operation of the sum of the windows is processed to get TDEV as a result.

Both MTIE and TDEV are phase stability metrics of a clock. It is important to distinguish the phase stability from the frequency stability of a clock although there is a relation between those two. The ITU-T recommendation also specifies frequency stability metric modified Allan deviation (MDEV) which is relative to TDEV according to the following Equation (6). [29]

$$MDEV(n\tau_0) = \frac{\sqrt{3} \times TDEV(n\tau_0)}{n\tau_0} \tag{6}$$

5.3 Masks for traditional metrics

The ITU-T recommendations G.823 and G.824 specify masks for MTIE and TDEV metrics for different signals in order to confirm that the network can satisfactorily control the amount of jitter and wander in the network elements, more specifically at network node interfaces (NNI)[13]. In other words, the phase stability of the clock must be below the mask to guarantee the required performance. The masks are originally defined for PDH networks to specify the jitter and wander limits in the network interfaces. By definition, short-term variations that are of frequency greater than or equal to 10 Hz are called jitter and, respectively, variations that are of frequency less than 10 Hz are called wander [29]. Mask for the different bit rates in the 1544-kbit/s hierarchy are defined in G.824 and masks for the 2048-kbit/s hierarchy in G.823.

Conformance with the mask guarantees the appropriate quality of service level for preventing slips (i.e. repetition or deletion of a block of bits) or single bit errors. Cellular base stations have been designed so that conformance with the MTIE mask combined with the frequency stability of the local oscillator assures that the frequency stability is adequate.

Note, the sheer MTIE mask does not guarantee the required stability, i.e., for instance, WCDMA frequency stability limit of 50 ppb is for a timeslot which corresponds to MTIE 3.3 * 10^{-11} s at a 0.67 ms observation window (see Figure 29). One can plot the frequency stability requirement curve (MTIE(τ)/τ = 50 ppb) in the same figure with the MTIE traffic interface mask (i.e. phase stability mask). By extending the 50 ppb line to cross the MTIE mask it indicates the required oscillator stability to average out the short-term frequency variation.

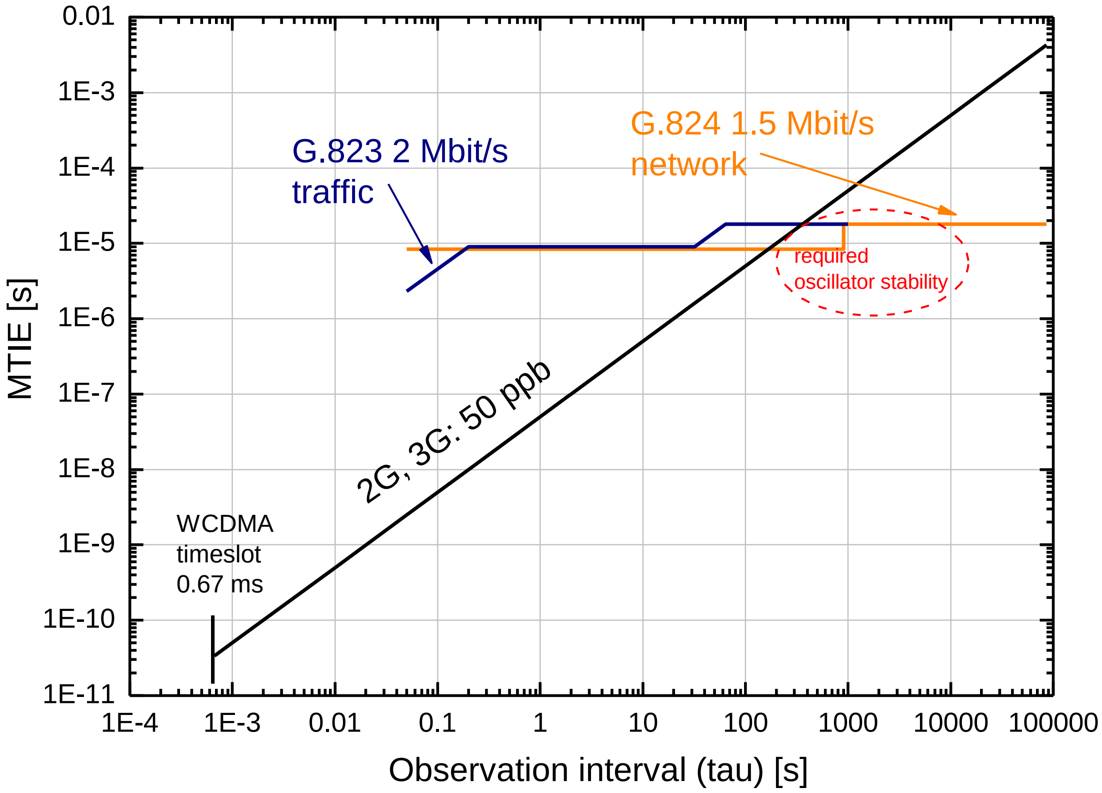


Figure 29: MTIE phase stability masks (2M and 1.5M hierarchies) with the frequency stability requirement curve (2G/3G: 50 ppb). The 50 ppb line is extended from the initial WCDMA timeslot requirement. Required oscillator stability is indicated in the picture (the red circle). The applicable oscillator stability must be around 400 s fro reaching 50 ppb stability using a 2M PDH service for synchronization.

The maximum phase error mask for PDH traffic (2048 kbit/s hierarchy) specified by ITU-T G.823 is depicted separately in Figure 30. The mask is for MRTIE (maximum relative time interval error) because it is relative to the PDH reference timing instead of the absolute reference clock. The figure indicates that the phase drift in traffic interface should be less than 18 µs in 1000 s. This same phase stability requirement is obviously applicable in a CES solution over asynchronous packet networks.

Figure 31, on the other hand, introduces a TDEV mask used in a 2048 kbit/s PDH hierarchy synchronization interface. It is rather irrelevant in the mobile backhaul case but presents a possible TDEV averaged phase stability limit. Due to the averaging capability of TDEV, the mask is a lot stricter than MTIE masks.

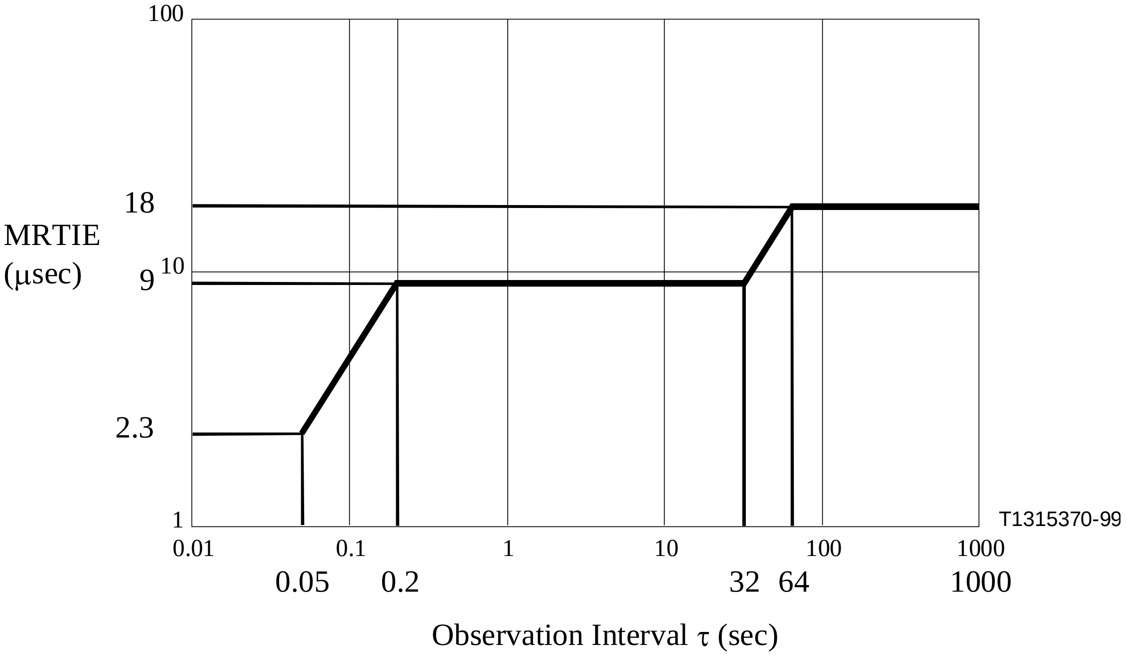


Figure 30: ITU-T specifies the wander limit (MTIE mask) for traffic interface in PDH networks. This mask is for MRTIE instead of MTIE since it is relative to the PDH reference instead of the absolute reference. [29]

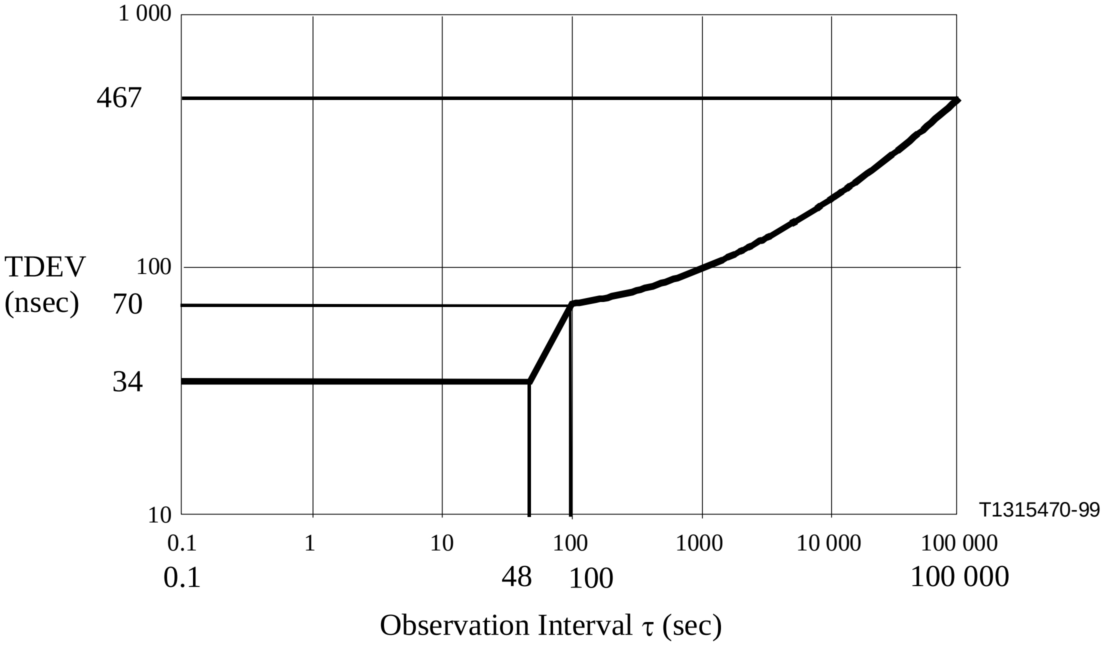


Figure 31: TDEV mask for sync interface in PDH network (2048 kbit/s PDH hierarchy). Due to synchronous interface and TDEV nature, the mask is relatively tight (i.e. the long term averaged wander should be less than 0.5 μs). [29]

5.4 Packet selection process

It has been proposed in the ITU-T standardization body to calculate TDEV from packet delays after selecting minimum delay packets instead of actual TIE/TE obtained from the output of the slave clock (as explained in Section 5.1) since both the PDV and TIE/TE data are phase information[31]. The packet selection resembles the operation of real packet clocks, see description below. In the case of the ToP solutions, the final averaging (i.e. noise reduction) before controlling the local clock is performed in the PLL filter of the slave. [32]

There are several packet selection methods proposed for packet-based timing which are divided into two approaches; integrated packet selection and pre-selection. Integrated packet selection means that the selection is integrated to the actual metric whereas the pre-selection process is executed before the metric. In both cases percentile, band, minimum picking selection methods can be used. The packet selection methods are based on the observation that a small fraction of all the packets experience the absolute minimum delay consisting of propagation and processing delays only. Furthermore, the minimum delay experienced by some packets is actually quite stable regardless of the link usage. An exception is, of course, the situation where the link usage is 100%. In that extreme situation every packet is queuing. Figure 32 illustrates the stability of minimum delays. More generally, also some other region of packet delays can be used which is then called band or range selection. [8]

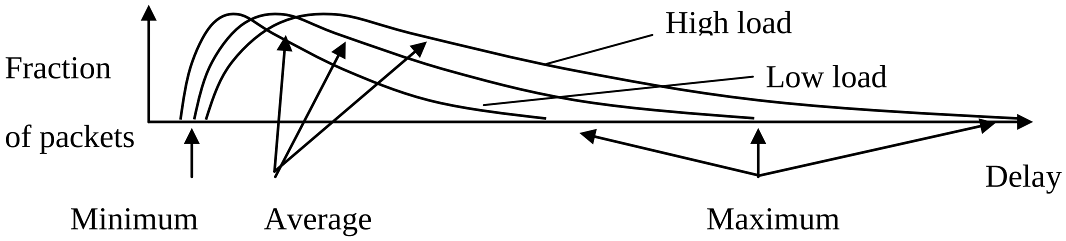


Figure 32: Stability of minimum delay stays stable in different load scenarios.

Different integrated or pre-selection methods are combined with the traditional metrics in order to achieve result from the PDV data corresponding to the metrics calculated from the actual slave clock output TIE data. For example, minTDEV, pTDEV (percentile) are metrics with integrated packet selection and, respectively, fpTDEV (fixed window percentile TDEV) is a metric with packet pre-selection. Although, the packet selection method makes the metric more suitable for a packet environment, it does not change the nature of the actual metric.

Next, all the proposed packet selection methods are discussed and finally listed in Table 3. The pre-selection essentially divides the input PDV data into same-sized fixed widows that a selection method is applied to. With this procedure, the original input data is reduced significantly since only one value is taken for the metric (e.g. averaged value, minimum value, etc.; depends on the selection method) from each fixed selection window. The principle of integrated packet selection is illustrated in Figure 33 and the packet pre-selection is depicted in Figure 34. [8]

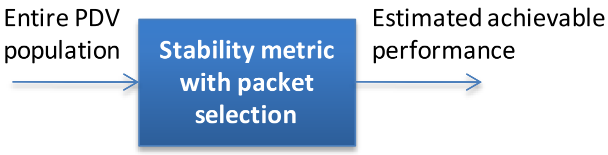

Entire PDV population
Stability metric with packet selection
Estimated achievable performance

Figure 33: Integrated packet selection method combines the packet selection and the metric (i.e. selection happens in the observation window (τ)), for example in Equation (5).

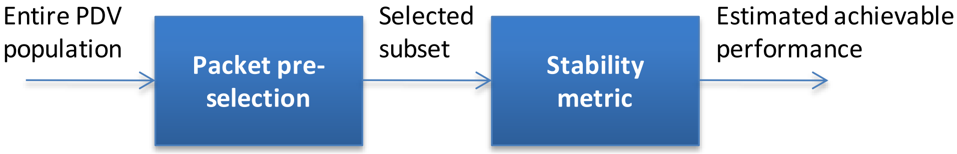


Figure 34: The principle of pre-selection processing. The entire PDV data is divided into fixed selection windows and the windows are processed according to a selection method. The pre-selection output produces a reduced subset of the original data. Note that after the pre-selection, the normal metric calculations are executed for the new reduced data.

The minimum selection (*min* prefix in integrated selection and *minimum-picking* prefix in pre-selection) method picks up only the smallest value (i.e. the fastest packet) from each section of data [8]. Thus, the noise filtering property of the minimum selection method is weak due to lack of averaging. In addition, in the integrated minimum packet selection method the smallest value of a window must be constantly updated since windows are sliding across the data. However, an efficient implementation has relatively small CPU load.

The percentile selection (*p* prefix in integrated selection and *fp* as in pre-selection) method solves the lack of averaging problem. In this method a certain predefined percentile of smallest values are taken into account and averaged from a section of data. In the integrated processing the values are taken and averaged from the sliding observation windows whereas in the pre-selection they are taken from the fixed selection windows (i.e. windows which do not slide across the data). [8] Generally, the percentile method is an effective way to use the advantage of stable minimum delay and still have averaging properties for removing noise effects. The downside of the integrated percentile selection method is the computational effort it requires since basically all the values inside sliding windows must be sorted continuously. The pre-selection method is less demanding for CPU and more applicable since the pre-selection procedure is done before the actual metric calculation to reduce the set of points significantly in the data while still preserving the minimum delay stability and averaging properties.

In contrast to the percentile selection method, the band selection (*band* prefix) method works similarly with the exception that it does not take the smallest percentile of delay values into account instead it takes values from a certain delay percentage band and averages them [8]. The selected band is obviously assumed to be stable all over in the data, thus, the method might be suitable for some delay behaviors. Nevertheless, this method is not used in the calculations presented in this thesis.

In addition, also cluster range selection method has been developed which is nearly equal to the band selection method. In the cluster range method the packet are selected and averaged from a predefined range of delay values instead of percentage. The range is the area between lower and upper limit within selection window. This packet selection method is neither used in this thesis. [8]

Note, the rate of the incoming timing packets and the fixed selection window size have significant influence on performance of the metric. From larger window and higher packet rate the smaller percentile can be selected and therefore better stability can be obtained. [33]

Table 3: Selection methods listed with computation effort evaluation. All selection methods are applicable for both integrated and pre-selection. [8] [34]

Selection method (acronym)	***Selection principle***	***Computation effort (CPU resources)***
percentile (*p*)	predefined % of the smallest values averaged from the sliding selection window	tedious, sorting needed constantly
minimum (*min*)	minimum value of the sliding selection window	relatively small
band (*band*)	certain band of values averaged from the sliding selection window	tedious, sorting needed constantly
minimum-picking (*min-picking*)	minimum value of the fixed selection window	small, because it reduces data for the metric

fixed percentile window (*fp*)	predefined percentile of values averaged from the fixed selection windows	small, because it reduces data for the metric
cluster range (*cluster range*)	values from the cluster range window (i.e. time/phase bounded range not percentile as in band method) are processed to obtain one value (e.g. averaged)	medium, no need for sorting since the bounds are fixed

5.5 Applicability of MTIE, TDEV and MDEV in packet environment

As discussed, the traditional clock metrics like MTIE, TDEV and MDEV could also be calculated from the delay data with the addition of the packet selection methods. However, the question is how good the metrics are to be calculated from the PDV data and what is the correlation between the metrics calculated from the PDV data and the packet slave clock output? The suitable metrics for estimating packet clock performance have been under standardization process for a couple of years now.

Therefore, as clarification and background for new metrics, a couple of pictures of how the traditional metrics with the packet selection process perform in reality are presented. In case of metrics with pre-selection, the effects of the two variables, the size of the pre-selection window and the percentile of selected packets, are and must be taken into account. In the integrated packet selection process, there is only averaging percentile parameter because the sliding observation window is also the selection window. The variation as a result of different selection parameters can be significant.

Figure 35 describes the performance of a different fpMTIE (fixed percentile pre-selection window MTIE) and the minimum-picking MTIE calculations with great variation in selection parameters. In the case of fpMTIE, the larger the pre-selection window and smaller percentile the better the performance compares with the MTIE of packet slave clock. The minimum-picking MTIE performs correspondingly when increasing the pre-selection window size although not as well as fpMTIE due to the lack of noise filtering capability. The green line in the figure depicts the 15 ppb frequency stability curve and the dashed red curve is the MTIE calculated from the packet slave clock output. A somewhat loose correlation can be found between the normal MTIE from the slave clock output and the packet fpMTIE with large pre-selection windows corresponding to the time constant of the PLL filter (e.g. 600 s or 1200 s) and none with the minimum-picking MTIE. However, the packet MTIE cannot be considered as a reliable packet synchronization metric since the results are highly dependent on parameters. MTIE is good worst case finder but the results of the packet MTIE are far larger than MTIE of a packet clock since inherently MTIE does not include any averaging. [32]

Besides the pre-selection methods, for instance, the percentile MTIE (pMTIE) could be also calculated which picks up a predefined percentile of smallest values from the sliding window and calculates MTIE from that. On the other hand, as explained in the previous section, the fixed percentile window method is nearly equal in terms of performance and less demanding for CPU, thus, it is used here.

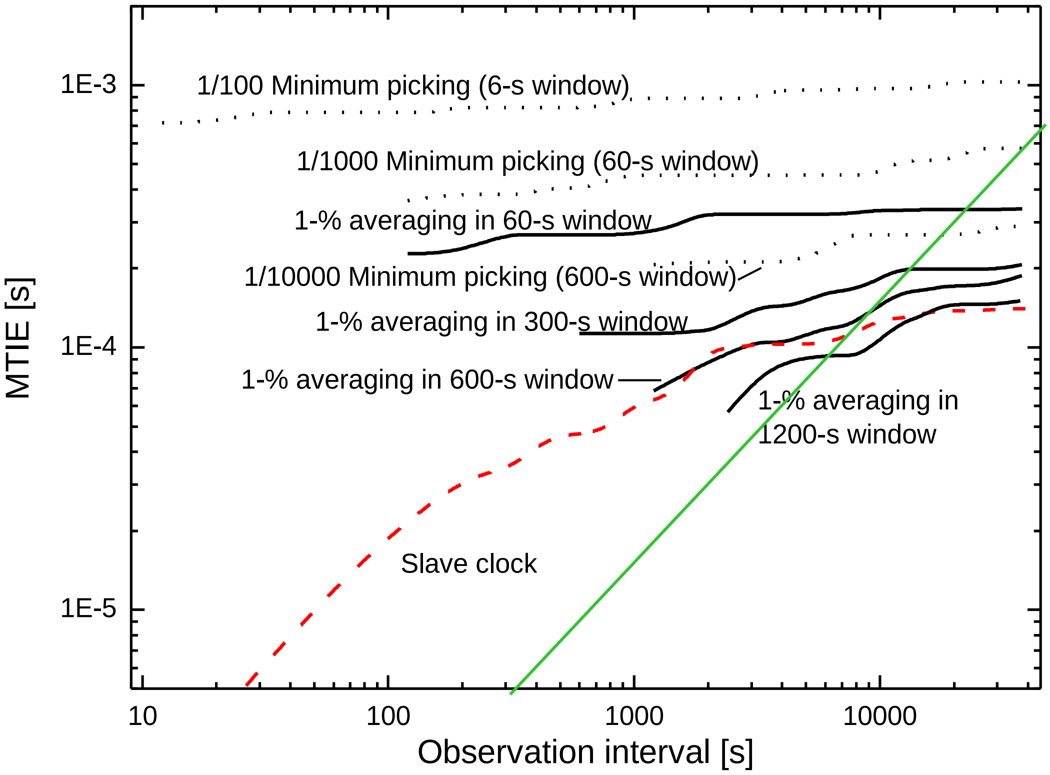


Figure 35: MTIE calculated from the slave clock output is compared against several packet MTIEs (fpMTIE and minimum-picking MTIE) with different packet pre-selection parameters. Best results can be obtained with a larger pre-selection window and smaller percentage of averaged minimum delays. [32]

The variation of different packet MTIEs is large which makes it rather difficult to find reliable parameters. Thus, packet slave clock simulation is probably the only option for making MTIE applicable in the packet environment (note, this method is explained and used later).

Contrary to the packet MTIE performance, TDEV calculated from the PDV data is not that sensitive to the variation of selection parameters due to the averaging properties of the metric. Figure 36 illustrates the normal TDEV and packet TDEV performance with different selection methods. One can see that the variation is rather modest (i.e. correlation between TDEV calculated from PDV and slave clock output is obvious). The reason for that is simply the behavior of the TDEV metric where averaging is carried out in the sliding windows. This resembles the low-pass filter behavior in packet clocks. TDEV is not a worst-case stability indicator. Instead, it studies the noise processes in the PDV data [33]. Therefore, TDEV cannot be used as a limit metric. However, packet TDEV can be valuable as an alternative metric in some situations where the worst case behavior is not of interest. Thus, TDEV with packet selection methods has been proposed for the appendix of the ITU-T recommendation G.8260.

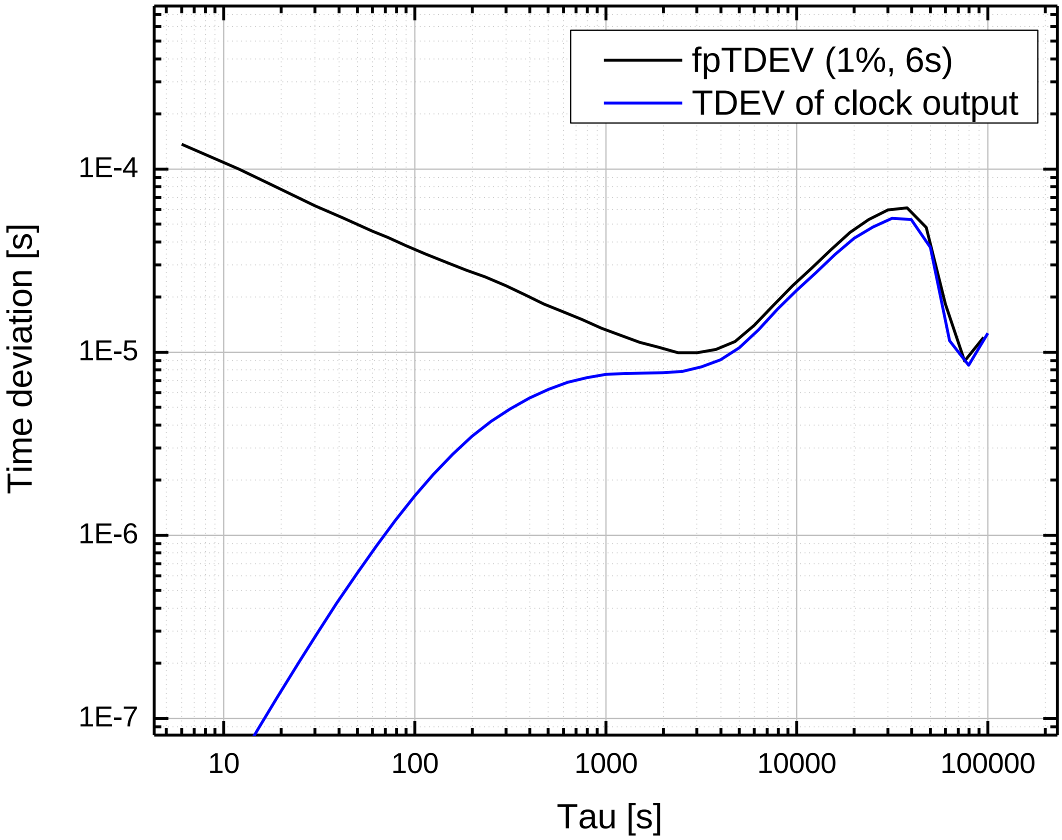


Figure 36: Fixed percentile TDEV metric calculated from timing packet PDV is compared against TDEV calculated from the packet slave clock output. The correlation is obvious at larger observation windows sizes (i.e. packet TDEV estimates the slave clock's TDEV).

Last, in Figure 37, MDEV (the frequency stability metric derived from TDEV) performance is compared with the maximum frequency error of the simulated slave clock (with PLL filters of 1st and 2nd order) output on the mathematically created input data (the artificially created data is explained and analyzed later on, see Figure 61). Correspondingly to TDEV in a phase stability context, MDEV is neither designed for detecting worst cases but finding long term averaged performance. Due to that, one can see that MDEV produces significantly lower frequency uncertainty value than the maximum frequency error of the simulated slave clock with 2nd order low-pass PLL filter. Packet slave clock with 2nd order filter is considered to perform adequately. Particularly, the left side of the MDEV curve (blue curve) shows that MDEV averages out the steep slope in the input TIE data which is creating the flat beginning of the maximum frequency errors curves. Thus, MDEV is not applicable for determination of obtainable frequency stability of certain network and slave clock implementation.

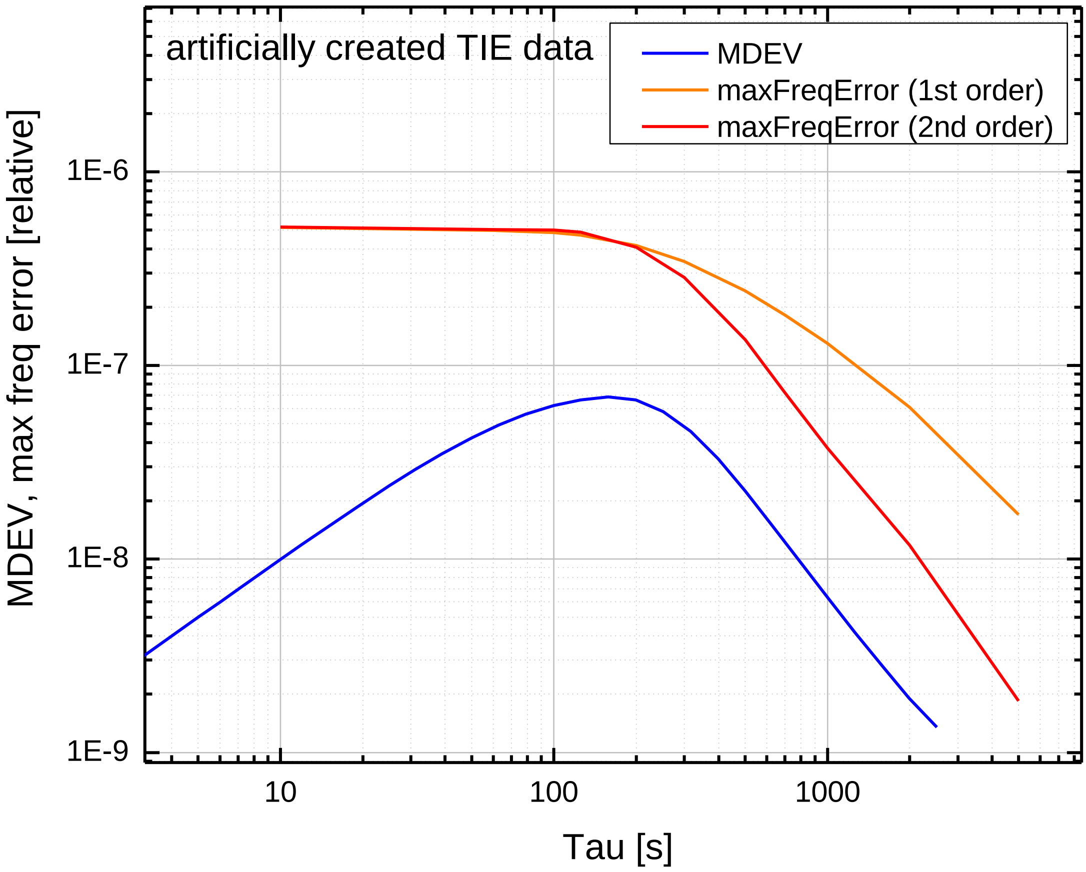


Figure 37: MDEV calculated from artificially created TIE data and compared with 1^st^, 2^nd^ order filter max frequency errors. MDEV goes much lower which indicates too good performance (i.e. a real packet slave would not be that good at any averaging window size). Hence, MDEV is definitely not good worst case finder.

All in all, TDEV and MDEV are suitable to be calculated from the PDV data but the nature of TDEV or MDEV is not applicable for determining the performance limits of the packet synchronization. On the other hand, MTIE is a good worst-case metric but is not optimal to be calculated from the PDV data due to missing averaging property of the metric. Hence, new reliable and robust worst case behavior packet metrics are needed.

6. New packet metrics

Since neither MTIE nor TDEV/MDEV are optimal metrics for determining the packet network limits (frequency and phase limits), new metrics are needed. This chapter introduces MATIE and MAFE which have been proposed for packet synchronization standardization. ITU-T Study Group 15/Question 13 has recently consented, June 2010, the architecture for frequency synchronization without on-path support and corresponding telecom profile for using PTP including recommendations G.8260, G.8261, G.8265, and G.8265.1. The recommendation G.8260 specifies the definitions and terminology for synchronization in packet networks and G.8261 specifies several test cases for determining the packet synchronization performance. The test cases are discussed in more detail when verifying the performance of MAFE.

Among packet TDEV metrics, MAFE and MATIE have been proposed for the appendix of G.8260 recommendation. The two packet selection approaches (pre-selection or integrated selection) and several selection methods have already been approved for the recommendation but not yet any metrics. Due to disagreement and too many contributions the metrics were postponed to a later version of G.8260. [8]

6.1 MATIE, MAFE

Maximum average time interval error (MATIE) is a newly developed packet synchronization metric but which afterwards was proven to be proposed already in 1992 as ZTIE (Z-transformed TIE), though, in a different context [35]. MATIE estimates the worst-case phase variation as a function of averaging capabilities of the slave clock. Regardless of the invention date, MATIE is hoped to solve the basic problems of the missing limit metric as discussed in the previous chapter. MATIE has two adjacent observation windows that slide through the data. The values inside each window are averaged and the maximum absolute value (i.e. unsigned value) of the difference between the two averages of adjacent windows is MATIE. Figure 38 depicts the sliding of two adjacent windows over the PDV or TIE data since like MTIE and TDEV, MATIE can be calculated from packet delays or from the clock output TIE. Moreover, observation window size (τ) is the independent variable as in TDEV and MTIE (i.e. the sweep is repeated at different window sizes to obtain MATIE at different τ values). MATIE is to certain extent a combination of MTIE and TDEV, i.e. combining averaging and maximum error calculations. MATIE resembles TDEV in a way that it averages the content of the sliding windows. On the other hand, MATIE resembles also MTIE because it searches for the maximum error value between the sliding windows whereas MTIE is searching it within one sliding window. [33]

MATIE is a phase stability metric as MTIE, and TDEV, respectively, MAFE is the frequency stability metric comparable to MDEV. The mathematical function of MATIE is presented in Equation (7). The variables are similar to the equations presented in Chapter 5. [36]

$$MATIE(n\tau_0) \cong \max_{1 \le k \le N-2n+1} \frac{1}{n} \left| \sum_{i=k}^{n+k-1} (x_{i+n} - x_i) \right| , \tag{7}$$

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

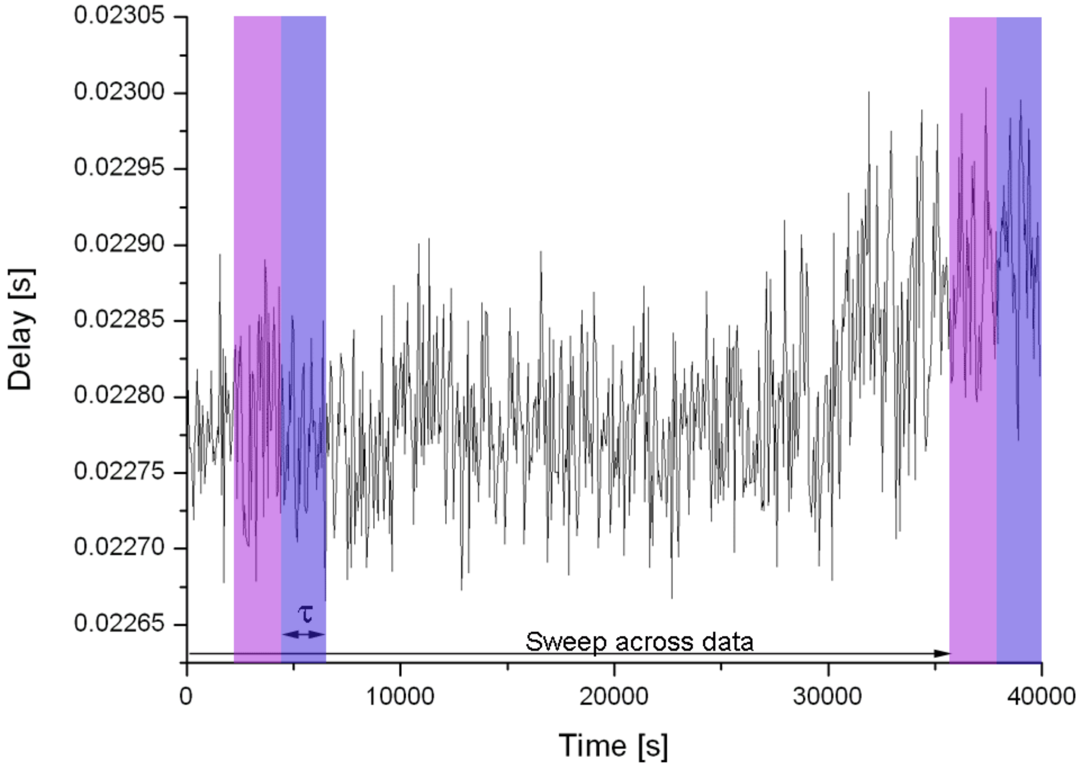


Figure 38: The principle of MATIE sliding windows. Two adjacent averaging windows are swept across the data and the maximum error between the windows is MATIE. Note that, this represents only one averaging window size. [36]

Equation (8) shows the relation between MATIE (phase stability metric) and MAFE (frequency stability metric). MAFE estimates the worst-case frequency error as a function of averaging capabilities of the slave clock. In MAFE, the unsigned difference is divided by observation window size (τ) which results in a slope that defines the average frequency error. Obviously, MAFE is the steepest slope at the certain observation window size when the combination of two windows is slid through the data. [36]

$$MAFE(n\tau_0) = \frac{MATIE(n\tau_0)}{n\tau_0} \tag{8}$$

Figure 39 illustrates MAFE calculation at certain sliding point and window size. The whole data is divided into red bars which represent MAFE sliding windows. The height of the bars depicts the average of the values inside each window. Thus, MAFE and

MATIE could be visualized and it is easy to determine the point where the maximum average error is found. For instance, the actual MAFE value (18 ppb) at τ = 3000 s is the worst case, i.e. the steepest slope between adjacent windows. However, one should note that the figure is simplified since in reality, windows are not fixed as the red bars, two windows are sliding point-by-point meaning that, for example, the MAFE slope is changing constantly and the largest value is chosen afterwards.

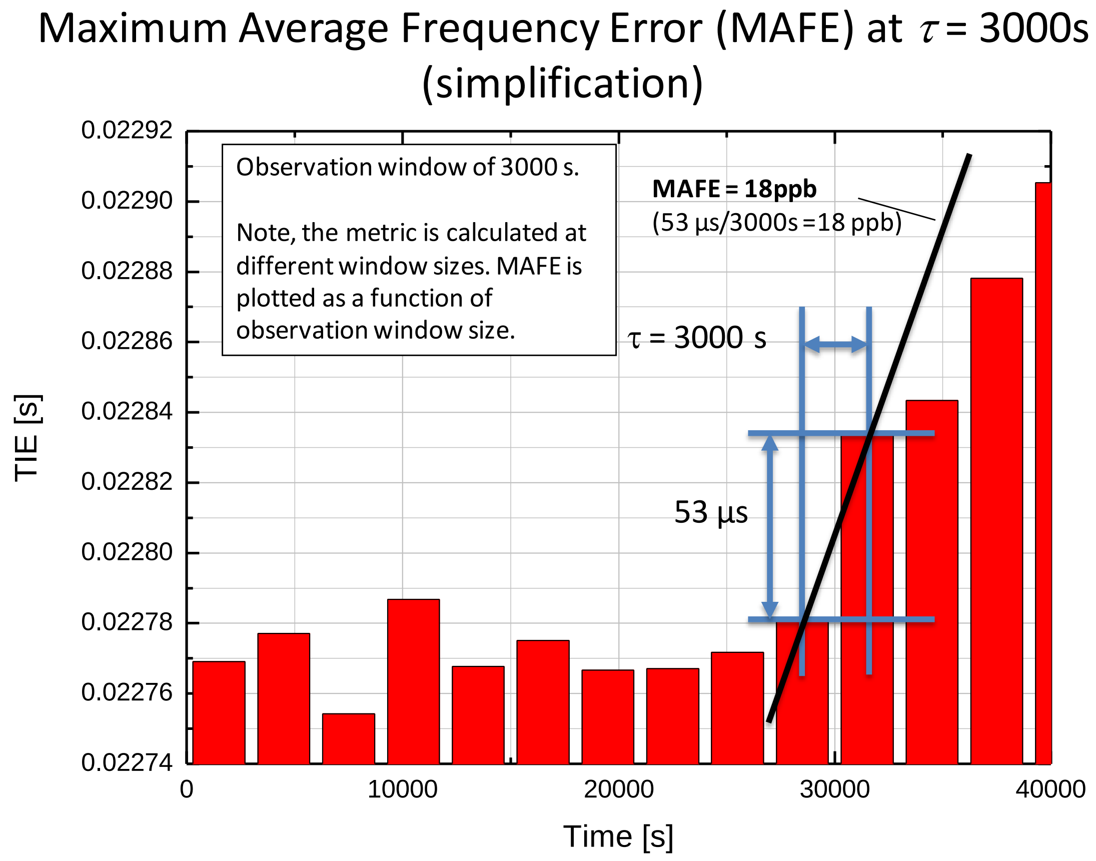


Figure 39: MAFE calculation example at τ (observation window) = 3000 s. Note, this example is a simplification. In reality the adjacent averaging windows slide over the data. This is only the situation at a certain point during the sliding operation. MATIE in this case would be 53 µs.

6.2 Packet clock simulations

For testing the applicability of, for instance, MAFE and MATIE, the maximum averaged errors predicted by the packet metrics should be compared with the maximum frequency or phase error produced by a real packet clock using the packets for timing. This section explains the purpose and principles of the packet slave clock simulations as an important method for performance verification. The slave clock simulation consists of packet selection and a PLL low-pass filter. Thereafter standard synchronization metrics are calculated from the simulated clock output.

As background information, previously conducted research has shown correlation between MATIE and the packet clock simulation with a 1st order low-pass filter[35]. To be more precise, MATIE actually performs slightly better than the simulated 1st order low-pass filter maximum phase error, and respectively, worse than the simulated 2nd order low-pass filter maximum phase error. On the other hand, MAFE correlates strongly with the maximum frequency error of a simulated packet clock with the 2nd order low-pass filter.

The reason why the 2nd order filters are of interest is because vendor specific phase-locked loops in real packet clocks use most likely minimally 2nd order low-pass filters, which produce significantly better noise filtering capability than 1st order low-pass filters. Thus, by comparing MATIE and MAFE against corresponding information from the 2nd order low-pass filter PLL implementation, the valuable relation between metrics and slave clock simulation can be obtained. Naturally, the results have to be comparable before interpretations. Thus, the process of unifying the results is illustrated here.

The simulated PLL filter implementation is elaborated later but the principle is introduced here. Since the synchronization metrics are functions of the observation window (i.e. filtering or averaging capability) the packet slave clock simulations should be corresponding. Thus, by calculating the filter outputs with different cut-off frequencies, the results as a function of filtering capability are obtained. The ratio between the observation window (τ) of metrics and the filter cut-off frequency was initially chosen to be $\tau = 1/f_{-3dB}$. However, it was noticed that MAFE estimates the maximum frequency error of the packet clock simulation with 2nd order low-pass filter when the simulation outputs were shifted along x-axis by the factor 3.2. Equation (9) presents the experimentally deduced relation between the observation window of MAFE and the cut-off frequency of the low-pass filter.

$$\tau = \frac{1}{3.2 f_{-3dB}} \tag{9}$$

Where:

τ is the observation window length used in the metrics calculation (e.g. MAFE);

$f_{\text{-3dB}}$ is the -3dB frequency (“corner/cut-off frequency”) of the low-pass filter;

Figure 40 illustrates the process of obtaining comparable results between the MAFE metric and the maximum frequency error detection from the slave clock simulation. As a result, both methods produce curves that can be plotted in the same graph.

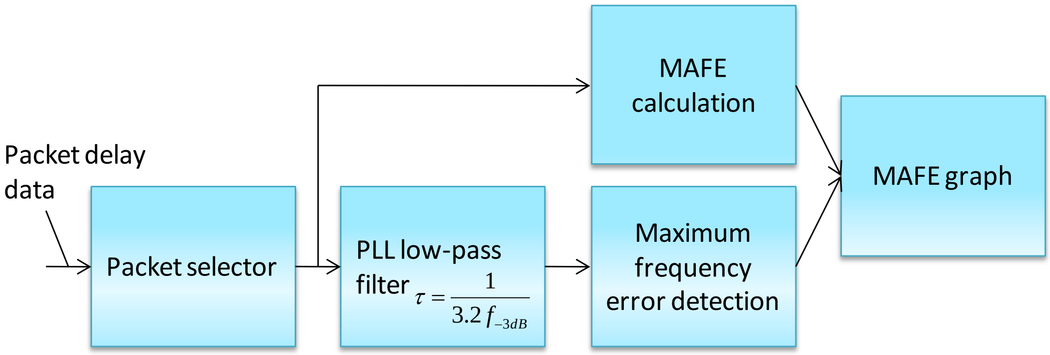

Packet delay data
Packet selector
PLL low-pass filter $\tau = \frac{1}{3.2 f_{-3dB}}$
MAFE calculation
Maximum frequency error detection
MAFE graph

Figure 40: Comparison principle of MAFE and maximum frequency error of simulated packet clock. The same packet selection methods are used for both approaches.

As a visual example, MAFE is depicted in Figure 41 with the maximum frequency error from the 2nd order filter slave clock simulations with two different shifting parameters (τ/3.2 and τ/2). The mathematically created input TIE function is introduced in Figure 61. The dashed red curve is the maximum frequency error matched with τ/2 which is significantly apart from the others at the corner point (around 200 s) which is crucial. The factor (τ/3.2) aligns the maximum frequency error curve nearly perfectly with the MAFE at the cornet point. Generally, MAFE performance is analyzed in Chapter 8.

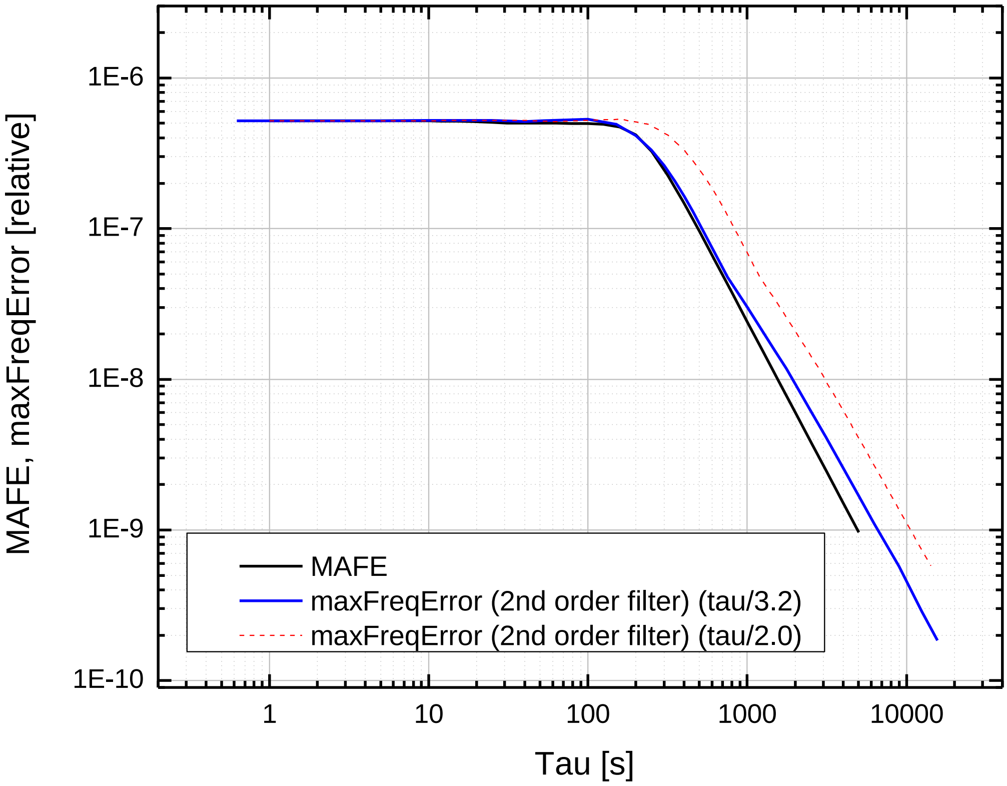


Figure 41: MAFE and the maximum frequency error of the slave clock simulations from the mathematically created TIE data. The figure indicates that τ/3.2 is the right shifting factor for the simulation outputs to match the MAFE.

On the other hand, in the context of phase stability, packet clock simulation outputs had to be shifted in line with the MATIE results by the factor τ/2 before the MTIE calculation matched with the MATIE results. The relation between τ and the cut-off frequency in the phase stability context is described in Equation (10). The coefficient 2.0 was deduced experimentally as 3.2 in the frequency stability context. The coefficient deductions were based on calculations on the artificially created data with distinguishable characteristics (discussed in Chapter 9). Clock simulation and packet metric produced otherwise similar phase stability curves but the peaks of the phase stability curves had to be aligned with each other.

$$\tau = \frac{1}{2.0\, f_{-3dB}} \tag{10}$$

Similarly to the MAFE case, Figure 42 depicts the process of obtaining comparable results between the MATIE metric and the maximum phase stability error of the slave clock simulation. MTIE is used to determine the maximum phase error of the packet slave clock simulation output.

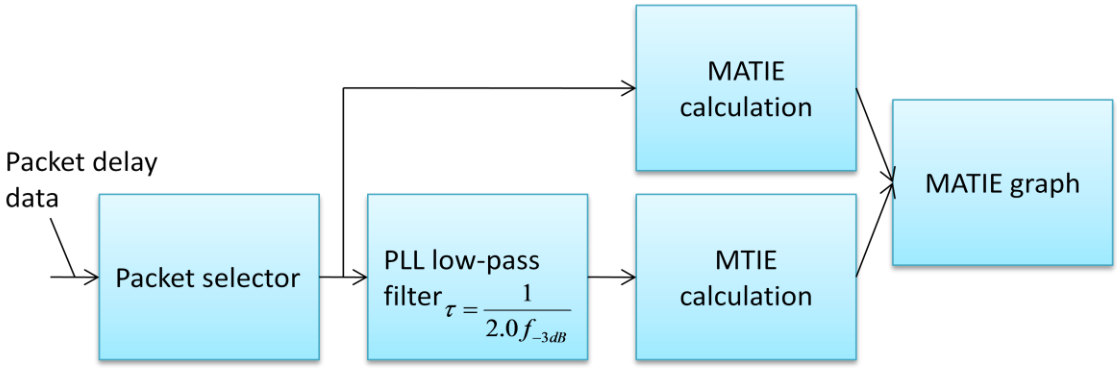


Figure 42: The principle of the process of obtaining comparable results between MATIE and the packet slave clock simulation. As a result, MATIE and the phase stability of the slave clock simulation can be plotted into the same figure.

As a conclusion, MAFE and MATIE are matched with the slave clock simulation with different coefficients (correspondingly τ/3.2 and τ/2). One should notice that in frequency synchronization the effect of that division is essentially different compared with a phase synchronization situation (i.e. because MTIE needs to be calculated with the new τ in case of phase synchronization). Hence, the change in τ does not only shift the curve horizontally along the x-axis, it actually changes the output values too (see Figure 43). The shifting factor moves the phase stability curve on diagonal direction while the ascending slope stays unaffected. This proves that the unifying factors have to be different with MATIE and MAFE.

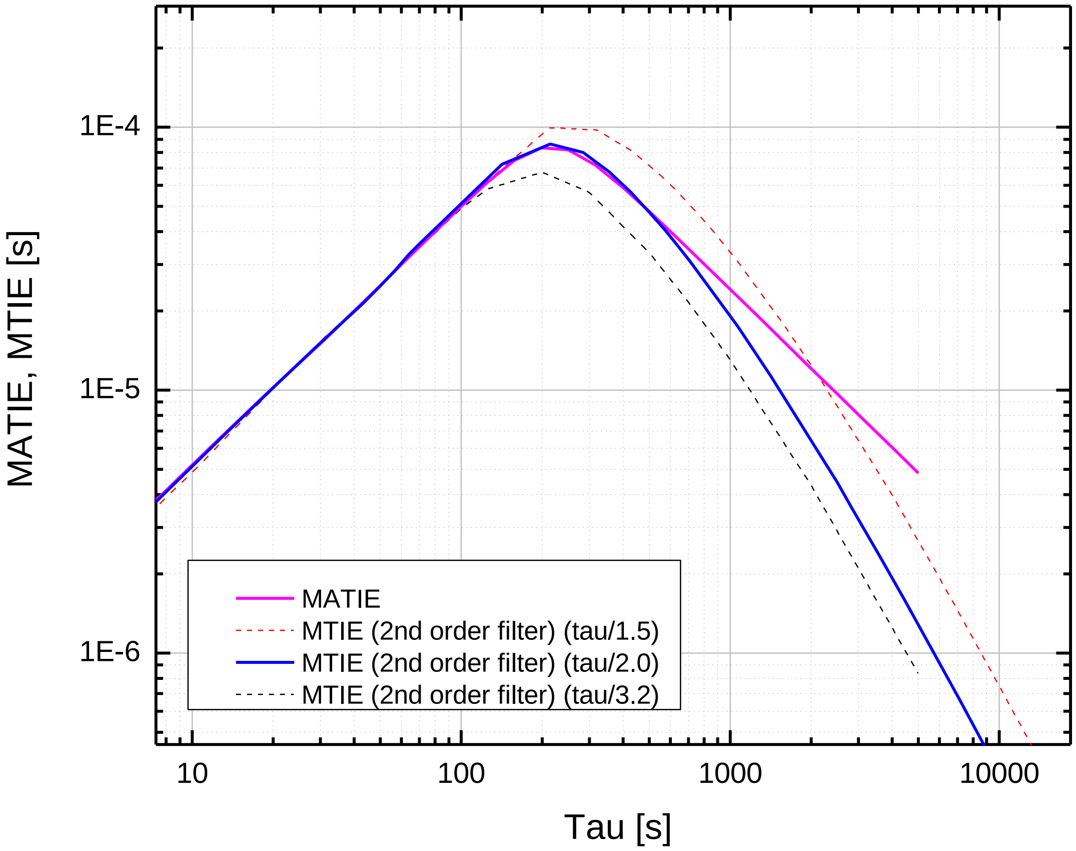


Figure 43: MTIE calculated from simulated slave clock output is aligned with MATIE with the presented factor (τ/2). Division shifts the MTIE curves in diagonal direction.

7. Measurement setups, data and research methods

In this research, several measurement data were used in the various calculations and analyses. Multiple measurements (e.g. active minimum delay stability and symmetry measurements) are conducted by the author whereas the actual PDV measurements and the phase error (TIE) measurements from the packet slave clock output were not executed by the author.

In addition to the measurements, the implementation of the metrics and the slave clock simulations are introduced here as the main research methods. The research conducted in this thesis is based on the results of the implemented metrics and the packet slave clock simulations on various data.

7.1 Minimum delay and delay symmetry measurements

As discussed earlier, the minimum delay of the small probe packets is the most stable. Therefore, the measurements concentrated mostly in this part of the delay distribution. As the setup could be used to measure delays in both directions, the delay asymmetry was also studied for determining achievable time synchronization performance.

In the active measurements, the packet delays are measured from the small probe packets streams which emulate the timing packets used by ToP protocols (e.g. PTP). Generally, understanding packet network behavior to some extent is mandatory when dealing with packet delay measurements. For instance, when excess traffic is generated with the same equipment and transmitted at fixed intervals, one must carefully evaluate the transmitting times of probe packets in contrast to the added load traffic in order to avoid systematic delay variation (also known as beating) [6]. Essentially, the beating means that every packet of a stream must wait for a free transmission timeslot or every packet passes without queuing depending on the time instant when the packets were sent. In real networks packets are more randomly timed and beating is uncommon. However, it is important in packet delay measurements that load streams would not cause beating and at least some percentile of packets can be transmitted without waiting for a free timeslot at the transmission interface of the network equipment.

In the measurement setup depicted in Figure 44, actual IP/MPLS (Internet protocol / multi protocol label switching) production network of an operator was used. The same core network was used with three different access technologies; 100 Mbit/s fiber, 10 Mbit/s SHDSL and 24 Mbit/s ADSL2+. Specifically, the four hop core consisted of 10 Gbit/s links with the aggregate length of approximately 40 km. Moreover, the use of a real production network is an advantage since the most realistic results can be obtained from the measurements over the actual production network (i.e. a network that is conveying all the subscriber traffic simultaneously to the probe and load traffic generated). Access technologies were undoubtedly the bottlenecks since the core used 10 Gbit/s links.

The purpose was to investigate the delay behavior of access technologies under significant stress as well as measure the delay difference between the uplink and downlink of theoretically symmetric connections. Probe packets were created and measured at the end points with QoSmetrics measurement equipment. The equipment with GPS timing as a reference clock could provide packet delay data with the accuracy of 1 µs. Moreover, the additional load was created also with the same equipment or with, for example, a 3G base station traffic generator.

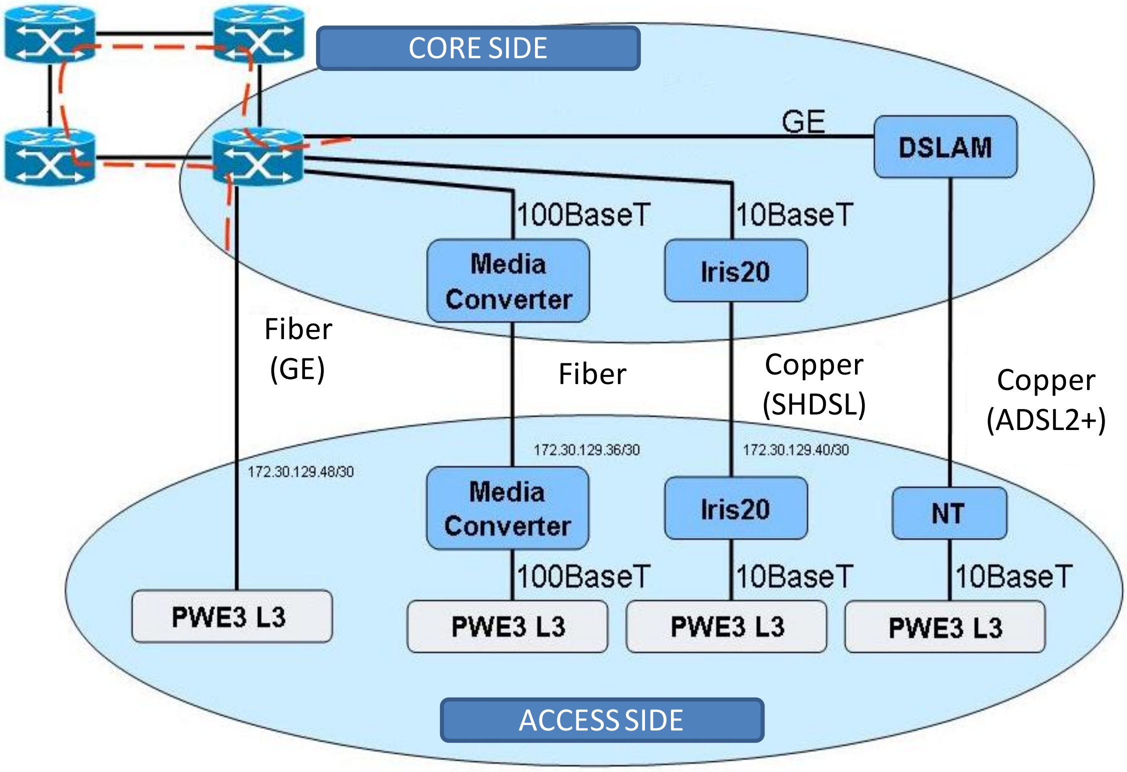


Figure 44: One measurement setup through a service provider's production network. The data path in core consisted of four IP/MPLS hops. Three different access technologies (Fiber, SHDSL, ADSL2+) used a last mile technology.

Figure 45 depicts the measurement setup in detail. The same measurement setup included also Linux PCs for ping and OWAMP (one-way active measurement protocol) measurements. However, ping and OWAMP measurement results are not important in the context of this thesis. Management traffic had the own subnet though it used the same switches.

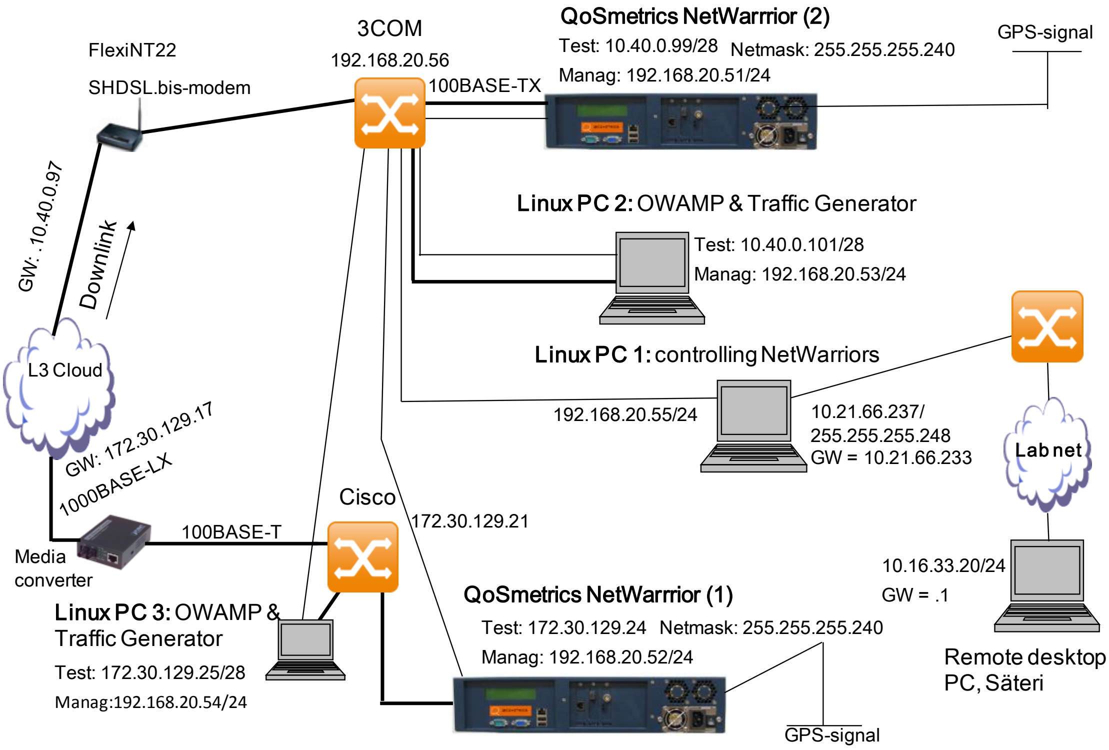
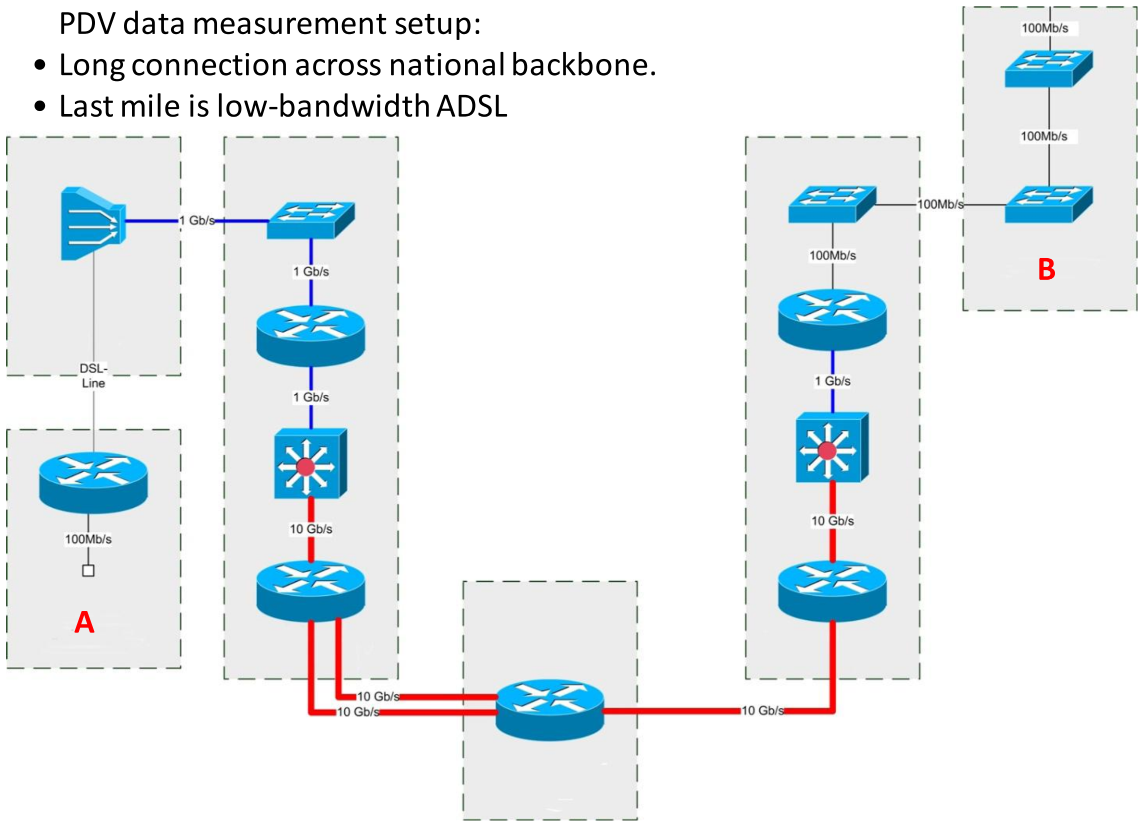


Figure 45: Active packet delay measurement setup over a production network. QoSmetrics NetWarrior's were used for load creation and sending the probe streams. This topology shows the measurement network and the separate management subnet used for configurations. Measurement data path is depicted with thick lines and management paths with thin lines.

7.2 PDV measurements

Since the packet metrics are calculated from the PDV data, several packet delay measurements are conducted at the ingress of the packet slave clock in real PTP packet synchronization test measurements (i.e. PDV data of real timing packets were recorded along with the actual TIE output of a packet slave clock). Consequently, this approach makes it possible to compare metrics with real packet clock behavior and performance.

Figure 46 presents another PDV measurement setup. The measured PDV data have been used in several metric calculations throughout the thesis and in packet environment research. These measurements were not executed by the author of this thesis. However, the measurement setup is presented for getting the right perception of the environment. In addition, this measurement lasted several days so that also the daily pattern in delays could be observed.



PDV data measurement setup:

- Long connection across national backbone.
- Last mile is low-bandwidth ADSL

Figure 46: Measurement setup between connection points A and B. High speed core network and ADSL as the access link technology.

7.3 Implementation of synchronization metrics

The principles of implementing the synchronization metrics in software are presented here. The packet synchronization metrics (e.g. MTIE, TDEV, MDEV, MATIE, and MAFE) have been implemented with various selection processes and methods introduced in Section 5.4. Moreover, some of the metrics like pTDEV (percentile TDEV) are extremely tedious for CPU which made it necessary to do some optimization work on the code, as well. All the different packet metrics and low-pass filters etc. are calculated from the measured PDV data with the scripts created for Origin mathematical software. Metrics are coded in Origin C-syntax. The software compiles and builds the C-language scripts.

As an example, the following block diagram depicts and elaborates fpMAFE and fpMATIE (fixed percentile window versions) scripts' functionality, see Figure 47. The same script calculates MATIE and MAFE at the same time due to the close relationship. After reading the parameters and the source PDV data, the fixed window percentile selection method (i.e. packet pre-selection) is executed. Thereafter, the sliding of two adjacent windows is implemented by maintaining the difference of sums of the two windows (i.e. new incoming value is replacing the removing value). The maximum difference is only maintained while windows are sliding through the data. The MAFE results are obtained from MATIE by straightforward divisions (see Equation (8)). The three first steps belong essentially to the pre-selection process after that the MATIE calculation is executed according to Equation (7).

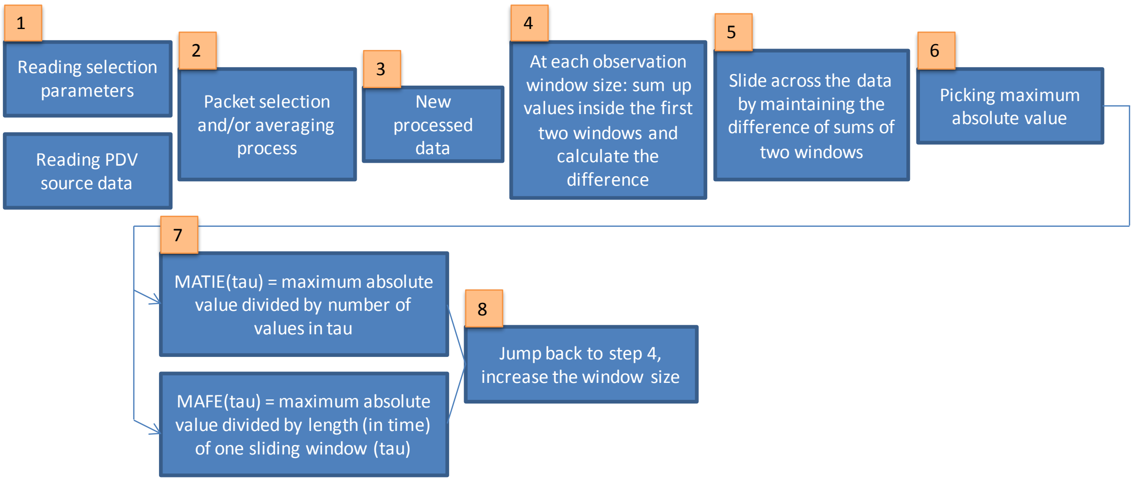

1
Reading selection parameters
Reading PDV source data
2
Packet selection and/or averaging process
3
New processed data
4
At each observation window size: sum up values inside the first two windows and calculate the difference
5
Slide across the data by maintaining the difference of sums of two windows
6
Picking maximum absolute value
7
MATIE(tau) = maximum absolute value divided by number of values in tau
MAFE(tau) = maximum absolute value divided by length (in time) of one sliding window (tau)
8
Jump back to step 4, increase the window size

Figure 47: The block diagram of fpMATIE and fpMAFE script implemented for Origin Math software.

7.4 Low-pass filter implementation

To be able to simulate actual slave clocks, the PLL low-pass filters must be implemented since they are essential part of the slave clock PLL architecture as explained earlier. For the research presented in this thesis, four recursive IIR (infinite impulse response) filters were implemented; 1st, 2nd, 3rd and 4th order Butterworth low-pass filters. Recursive filters were chosen due to easier implementation for Origin math software than non-recursive. Filter responses were calculated with different cut-off frequencies (i.e. with different real filters). Cut-off frequencies are related to the averaging capability of metrics (i.e. observation window. The smaller the cut-off frequency is the longer the observation period correspondingly. Metrics are functions of observation window (τ) length which makes it reasonable to have filter results also as a function of observation window (i.e. cut-off frequency in filter terminology).

For example, Figure 48 illustrates the 3rd order filter outputs when the input function is an artificially created TIE (pulse shape). The script calculates the filter output with 38 different cut-off frequencies (10 values per decade) which then can be plotted as a function of observation window.

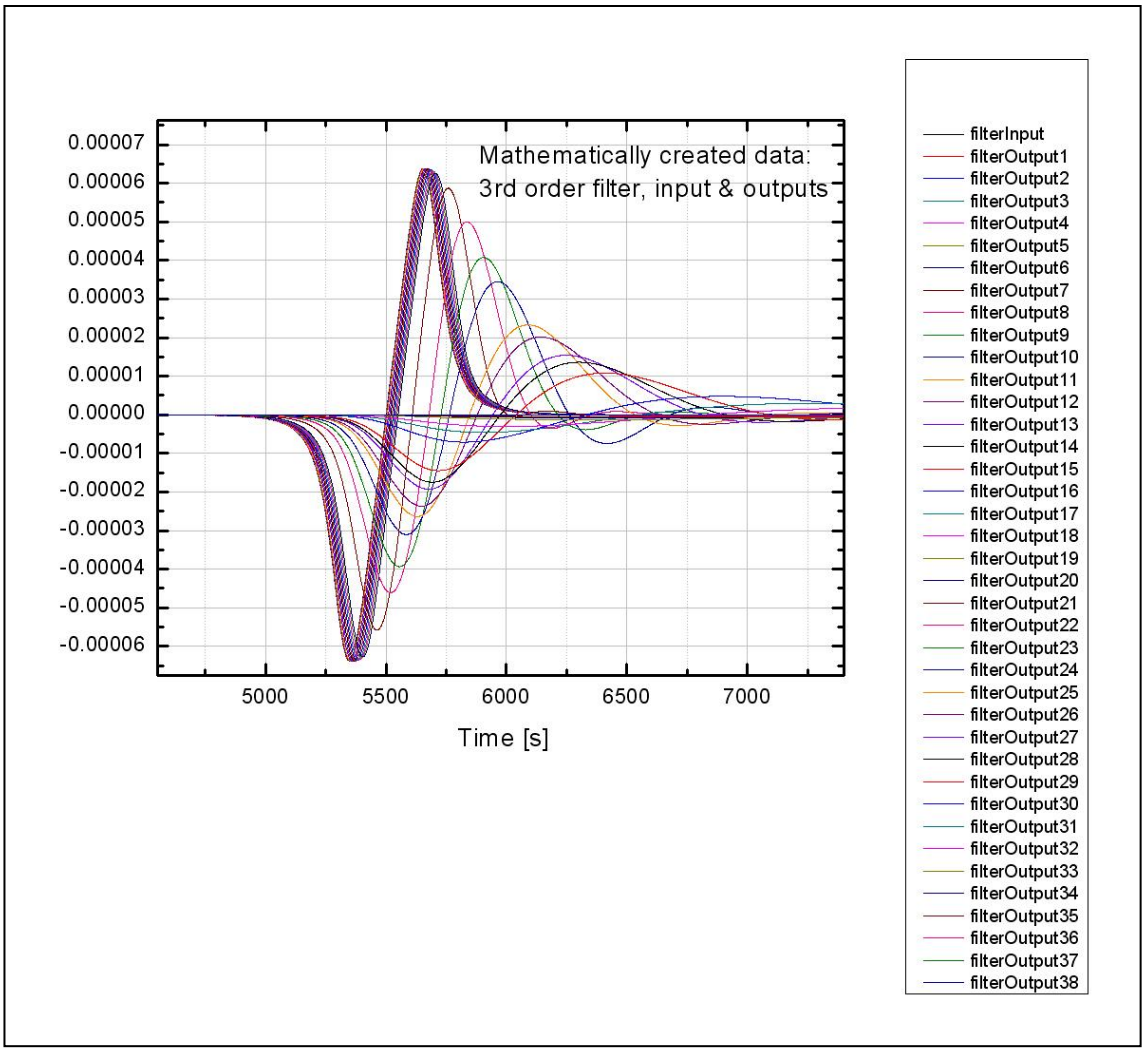


Figure 48: Mathematically created TIE function filtered with 38 different cut-off frequencies of the 3rd order Butterworth IIR low-pass filter. Each output represents simulated packet clock output, thus, MTIE or max frequency error detection can be calculated from them.

For each output, in the case of a frequency stability calculation, the maximum frequency error detection is executed. In other words, the maximum slope between two consecutive output values is searched. On the other hand, when phase stability is under investigation the MTIE calculation at the observation window size corresponding to the cut-off frequency is executed. Filters combined with the maximum frequency error detection are comparable with the results of MAFE and correspondingly filters together with MTIE calculation are comparable to MATIE as explained in Section 6.2.

A small disadvantage of recursive filters is the inherent feature of the time what it takes to adjust to the input signal. The smaller the cut-off frequency the longer it takes to adjust fully to the input signal, respectively, also the higher the order of the filter the longer it takes (depicted in Figure 49). This obviously has to be taken into consideration and compensated, for example, with seed values for filter coefficients or the reliable results must be taken only after the convergence. The latter has been chosen for every filter result presented in this thesis.

Equation (11) [37] presents the generic recursive IIR (infinite impulse response) 2nd order low-pass filter formula used in this thesis. The filter coefficients characterize the behavior of the IIR low-pass filter. Note that Butterworth filter coefficients are used in this thesis. One can notice that recursive filters use previous outputs for current calculation which naturally implies to the initial adjusting to the input signal. Seed values are not used but it would mean that the previous values at the beginning are fabricated to correspond to the real input as closely as possible.

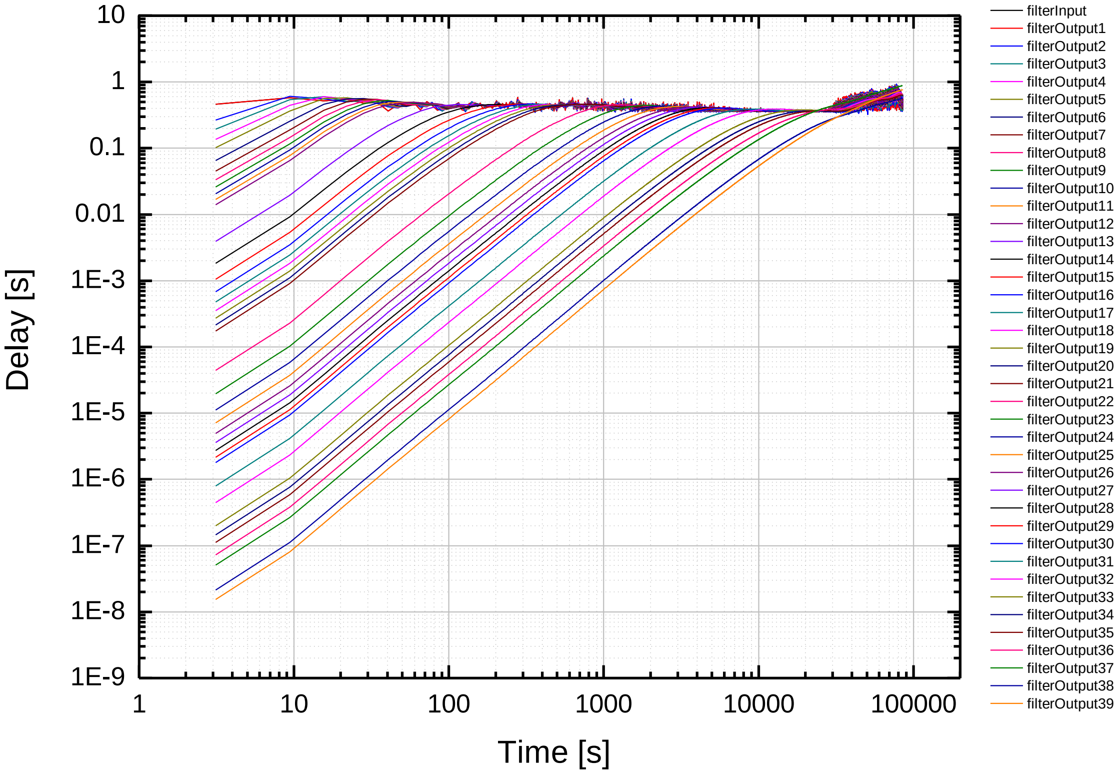


Figure 49: 2nd order low-pass filter adjusting to the input signal. In this case, there were no seed values, i.e. previous values at the start were set to zero.

$$filterOutput[i] = \frac{a[0]*input[i] + a[1]*input[i-1] + a[2]*input[i-2] - b[1]*output[i-1] - b[2]*output[i-2]}{b[0]} \tag{11}$$

Where:

a[0...2] are filter coefficients for delay input values;

b[0...2] are filter coefficients for previous output values;

input[i] is the input signal value at time instant i;

output[i] is the filter output signal at time instant i;

Furthermore, the outputs of the simulated filters are verified against a real proprietary low-pass filter. Figure 50 shows a real 3rd order PLL low-pass filter output and simulated 3rd order IIR low-pass filter output in the same picture with nearly the same cut-off frequencies. The input data is naturally the same for both filters but the simulated filter converge with input after small delay since no seed values were used for the recursive filter.

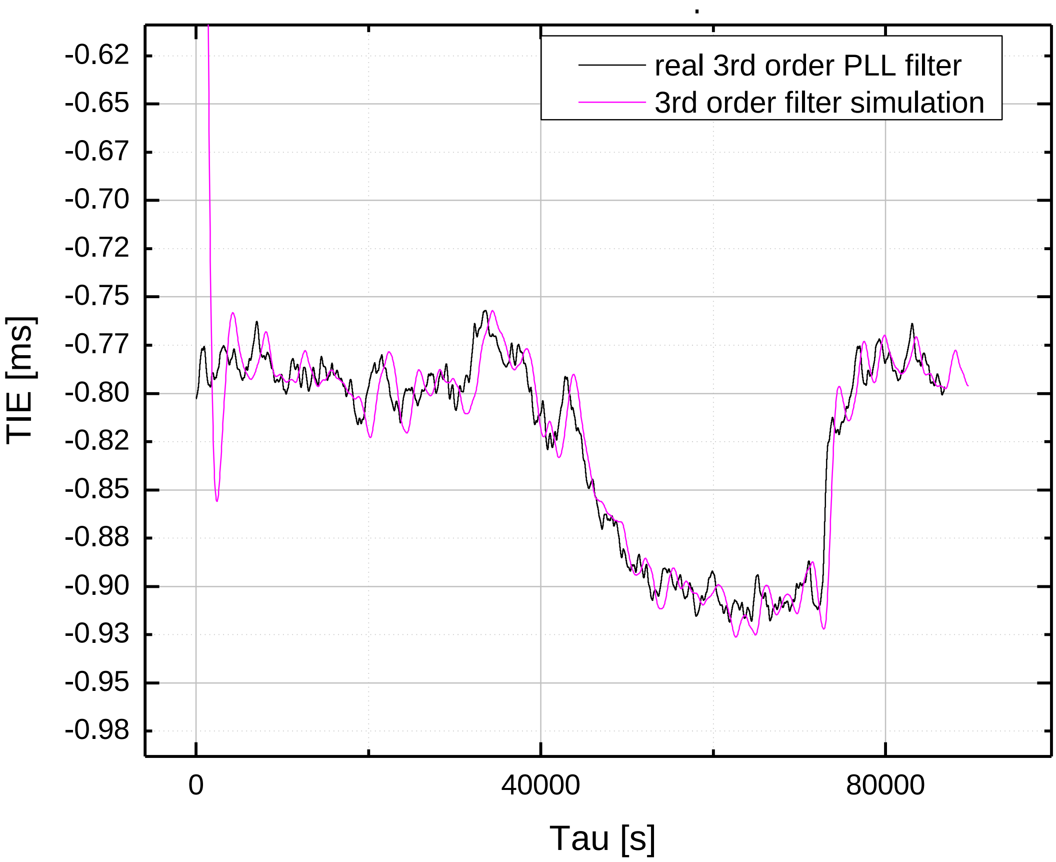


Figure 50: Real 3rd order low-pass filter compared with simulated PLL filter presented above. The cut-off frequency of the 3rd order PLL filter is not accurately known, thus, the filtering is slightly better in the simulated version. But that it irrelevant due to the fact the metric results (e.g. MTIE) are aligned precisely on the x-axis according to the cut-off frequency.

8. Verification of MAFE

MAFE (maximum average frequency error), described in Section 6.1, is one of the most promising frequency stability metric candidates for the new packet synchronization standardization governed by ITU-T. In this chapter, MAFE has been compared against several simulated PLL low-pass filters (i.e. packet slave clock simulations) as well as metrics calculated from real packet slave clock outputs. Besides the MAFE performance, some critics, remarks and counter arguments are discussed in this section. MAFE results have been compared with some other proposals from the synchronization standardization community.

As discussed earlier, MAFE can be calculated from the PDV data as well as from the output of the actual slave clock which gives the relation between the PDV data and the packet clock output [38]. The correlation (i.e. the applicability to estimate real packet slave clock performance) is investigated in this chapter.

8.1 MAFE compared with low-pass filters

Testing applicability of MAFE includes naturally a comparison with the performance (i.e. maximum frequency error) of the real packet slave clock receiving the timing in packets. Here the packet slave clock PLL functionality was simulated as introduced in the previous chapters. By experience, the slave PLL’s have at least a 2nd order low-pass filter for the noise reduction. Thus, it is important that MAFE predicts at least 2nd order filter performance. The next graphs are showing that MAFE actually estimates with even slightly better performance than the 2nd order filter.

One should remember that the simulated packet clock curves (e.g. max frequency error curves) represent filter performance as a function of cut-off frequencies. However, in reality, only one point is obviously significant, the point where the performance (i.e. frequency stability) is within specifications. Therefore, the simulations can be very useful when designing slave clocks due to the possibility of seeing the required filtering capability.

All in all, 1st, 2nd, 3rd, and 4th order IIR Butterworth low-pass filters were simulated. Filters were run with different packet selection parameters. The presented filter simulations are calculated either from the artificially created data or from real PDV data recorded at the input of the packet slave clock. Figure 51 depicts the performance of the simulated filters and MAFE on the artificially created data. At the small τ-values, filtering does not effect at all which results in the flat area on the left side of the curves. Different filtering capabilities can be observed with the larger τ-values (i.e. the smaller cut-off frequencies in low-pass filter and the larger the corresponding observation window in metrics).

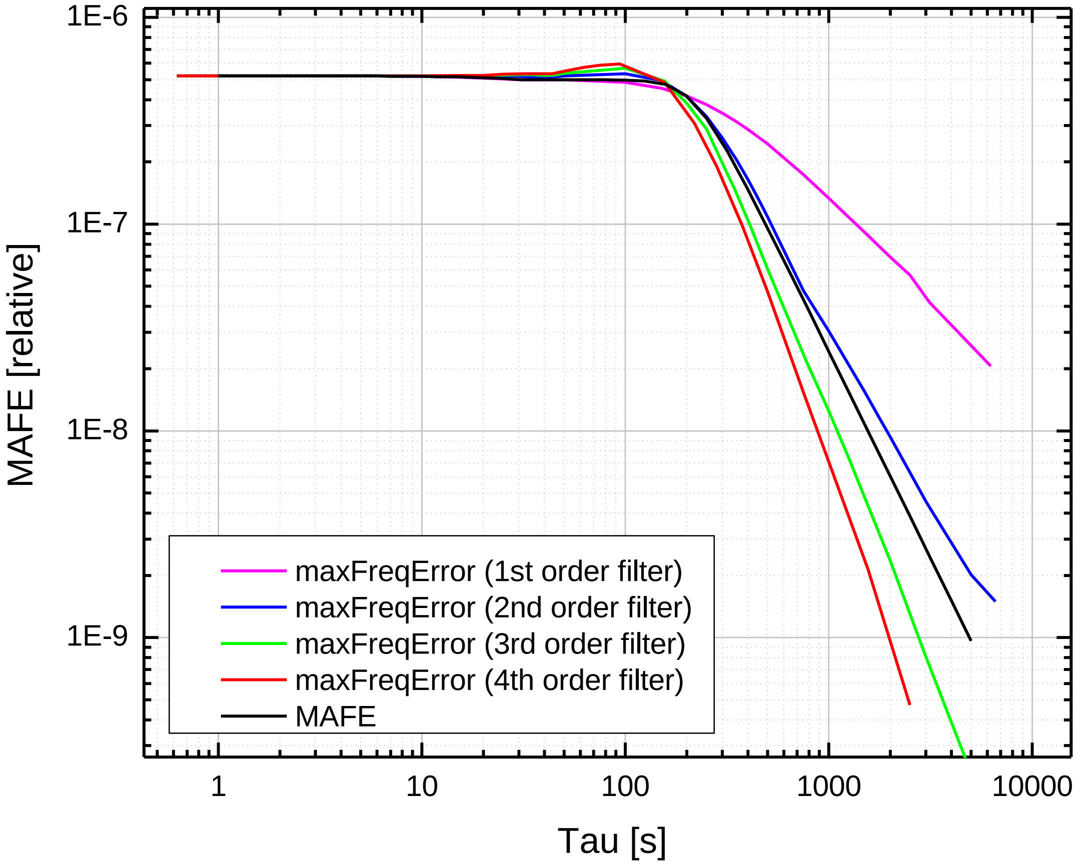


Figure 51: 1st, 2nd, 3rd and 4th order filters are compared with MAFE on the artificially created TIE data (explained in more detail later on). However, the filtering capability (i.e. the slope after the corner at approximately τ = 200 s) of MAFE is slightly better that the 2nd order filter but worse that 3rd order. Notably, 1st order filter is significantly worse than the others.

Figure 52 illustrates the performance in the case of a simulated delay data describing the combination of 4 hops with a daily load pattern. It can be seen that filters perform quite similarly in the realistic data with each other and fpMAFE (i.e. the differences between filters of different orders are smaller than for the artificially TIE data and fpMAFE estimates the performance well). The reason the equal performance is obvious, there are no such extreme situations in that particular data similar to the artificial TIE data. On the other hand, a slight difference is observable and obviously the 4th order performs the best due to the best filtering capability.

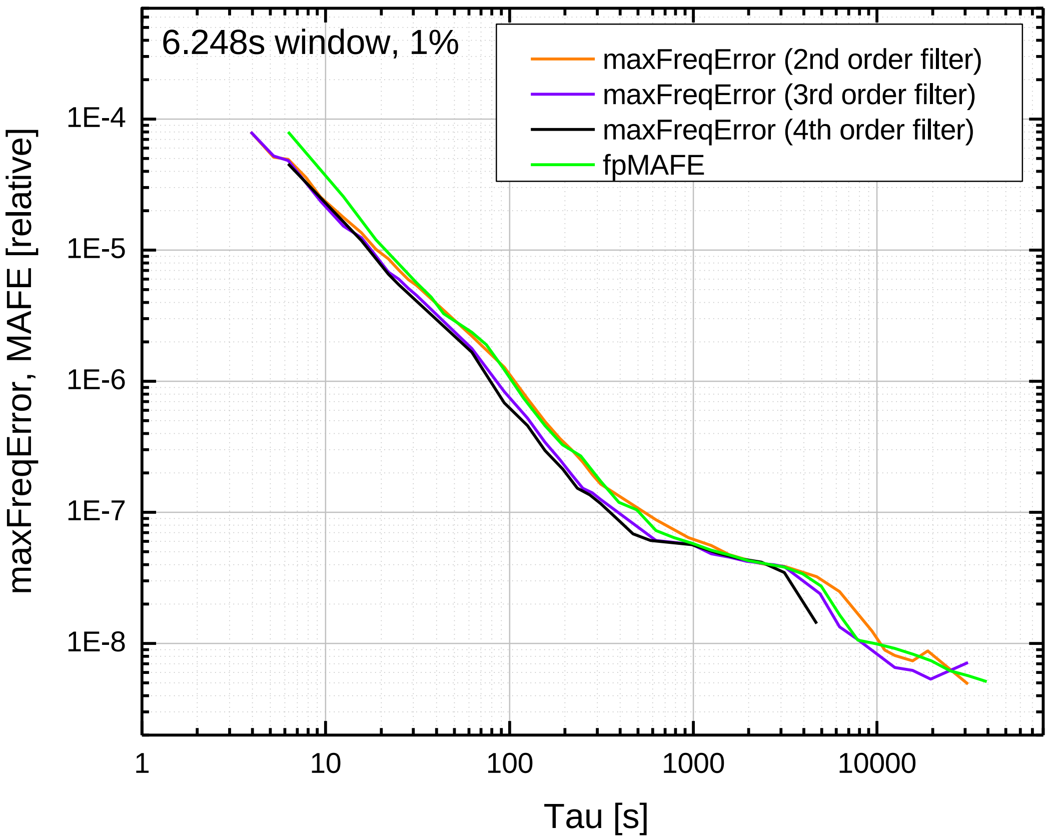


Figure 52: fpMAFE is compared against maximum frequency errors of 2nd, 3rd, and 4th order low-pass filter outputs on simulated data. Pre-selection process picked up the smallest value from blocks of 100 samples. It can be seen that the filtering capability of fpMAFE is nearly equal with 2nd order filter and does not lose significantly for either 3rd or 4th order low-pass filters.

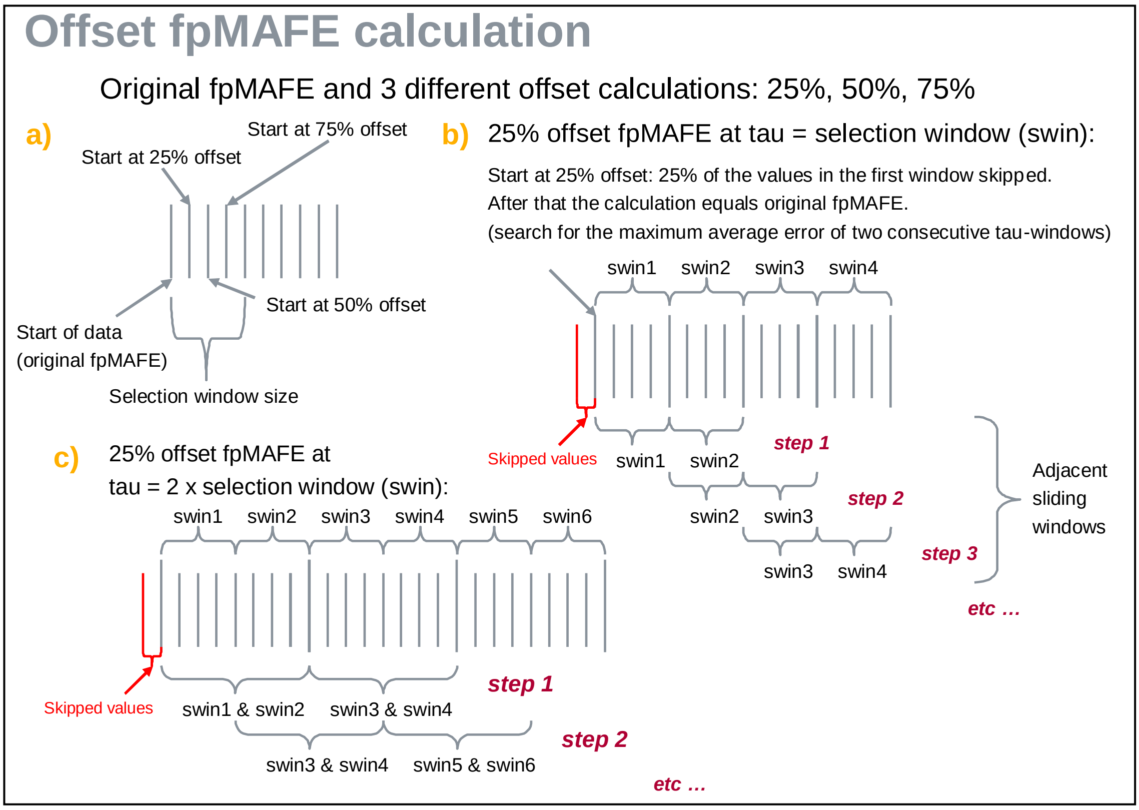
Generally, frequency stability metrics (e.g. MAFE) indicate the averaging period relationship to the obtained frequency stability. As introduced in the frequency requirement of cellular technologies, 50 ppb or better is a common requirement. A horizontal line can be drawn at the y = 50 ppb (5E-8) and the required averaging period can be obtained straight from the curves (i.e. when the curve goes below 50 ppb the averaging is adequate).

8.2 Offset and overlapping MAFE/MATIE

Although the MAFE and MATIE metrics are designed for estimating ToP performance, packet selection is still always required for obtaining good stability from PDV information. The most applicable pre-selection process has been the fixed window percentile method, which takes the smallest delays into account and averages them. This method decreases the data points significantly as explained in Chapter 5. Nevertheless, there have been some arguments against fpMAFE calculation. It has been questioned that how does the starting point of the fixed windows affect the outcome of fpMAFE. Therefore an offset version of fpMAFE was developed to investigate the effects and the results are presented below.

The principle of the offset fpMAFE is that some fraction of the first selection window is neglected and the following fixed windows are shifted accordingly so that each window has still the same size. This means that the first fixed selection window starts a little bit later, which, in turn, affects to the values of every fixed selection window, and therefore, results in a slightly different outcome. For instance, offset 25% skips 25 values of the selection window of length 100 values and starts fixed windows from that point.

Figure 53 elaborates the principle of offset fpMAFE and the same method can be applied for MATIE as well. Fixed selection windows are denoted as selection window number (swin #) notation. In addition, a few steps of the fpMAFE calculation are also depicted. Note that fpMAFE observation windows (τ) are sliding one fixed selection window at a time (see the steps in Figure 53). The actual script is implemented so that the function calculates 4 different offset cases (0%, 25%, 50% and 75%) at the same time. Note that 0% offset corresponds to the normal fpMAFE.


Offset fpMAFE calculation

Original fpMAFE and 3 different offset calculations: 25%, 50%, 75%

b) 25% offset fpMAFE at tau = selection window (swin):

Start at 25% offset: 25% of the values in the first window skipped.
After that the calculation equals original fpMAFE.
(search for the maximum average error of two consecutive tau-windows)

c) 25% offset fpMAFE at tau = 2 x selection window (swin):

Figure 53: Offset fpMAFE calculation explained. a) Beginning of the fixed windows, offset pointed in the picture. b) 25% offset fpMAFE at τ = selection window c) 25% offset fpMAFE at observation window (τ) = 2x selection window.

As mentioned, the observation windows are sliding through the data but jumping one whole selection window at a time. Counter arguments have been brought up that some information is probably lost due to those large jumps instead of smoother sliding. Offset fpMAFE is not solving that issue either. Therefore, another modification, the overlapping fpMAFE, is needed for proving that it actually does not matter.

The principle in the overlapping modification is that the function slides through the selection windows which are overlapping. The smoother the sliding is the more the fixed selection windows are overlapping each other. This naturally increases sliding steps which requires more computational effort. But on the other hand, the overlapping fpMAFE is a better worst case finder than the normal fpMAFE due to smoother sliding through the data. The more the selection windows are overlapping, the more it resembles MAFE/MATIE with integrated selection instead of pre-processing. Figure 54 demonstrates the fpMAFE calculation with overlapping selection windows. In that case, the two adjacent fixed selection windows have 75% of values in common (i.e. fpMAFE is jumping 25% of selection window at a time).

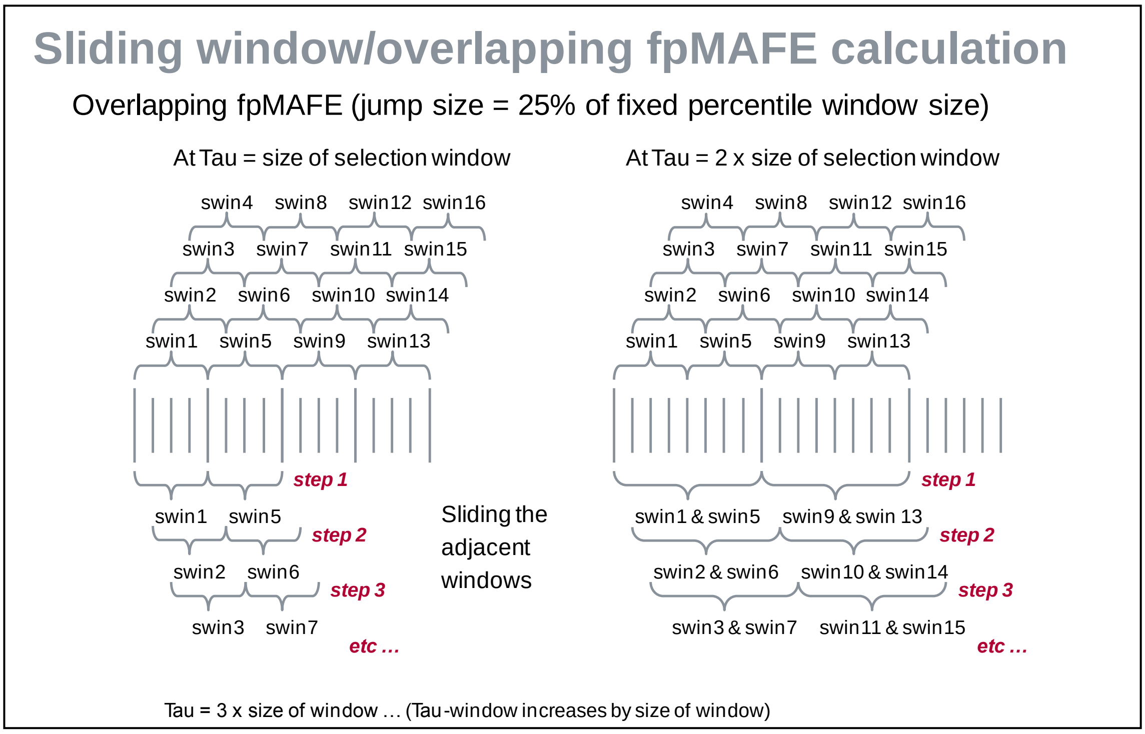

Sliding window/overlapping fpMAFE calculation
Overlapping fpMAFE (jump size = 25% of fixed percentile window size)
At Tau = size of selection window
At Tau = 2 x size of selection window
swin4 swin8 swin12 swin16
swin3 swin7 swin11 swin15
swin2 swin6 swin10 swin14
swin1 swin5 swin9 swin13
step 1
swin1 swin5
step 2
swin2 swin6
step 3
swin3 swin7
etc ...
Sliding the adjacent windows
swin1 & swin5 swin9 & swin 13
swin2 & swin6 swin10 & swin14
swin3 & swin7 swin11 & swin15
Tau = 3 x size of window … (Tau-window increases by size of window)

Figure 54: Overlapping fpMAFE is explained step by step. In this method the selection windows are overlapping each other. 75% of the values are the same in two adjacent selection windows. Left side of the figure depicts situation where observation window corresponds to one selection window, whereas, in the right side of the picture observation window is two selection windows so that those are not overlapping.

As one could theoretically affirm, 75% overlapping selection windows predicts the envelope (i.e. the worst case behavior) of the offset 25%, 50% and 75%. The worst case of the abovementioned offsets can be obtained with the overlapping fpMAFE since the function is sliding 1/4 of the fixed selection window at the time. Obviously the starting point for the overlapping fpMAFE and the offset fpMAFE has to be the same (i.e. so that the offset fpMAFE 0% starts from the same point as the overlapping fpMAFE with the jump size of 25% of the fixed window).

Figure 55 depicts the offset fpMAFE and the overlapping fpMAFE 25% on the same data. Slight differences between the normal fpMAFE and the offsets are observed with the smaller observation windows and the overlapping fpMAFE finds the worst case behavior of the four offsets as explained. However, the values with more importance are those at larger window sizes. To obtain the most reliable results out of fpMAFE; perhaps the average of those four offsets would be the best to describe the fair and less biased performance since the overlapping version being the ultimate worst case finder might not be a good average performance indicator of fpMAFE.

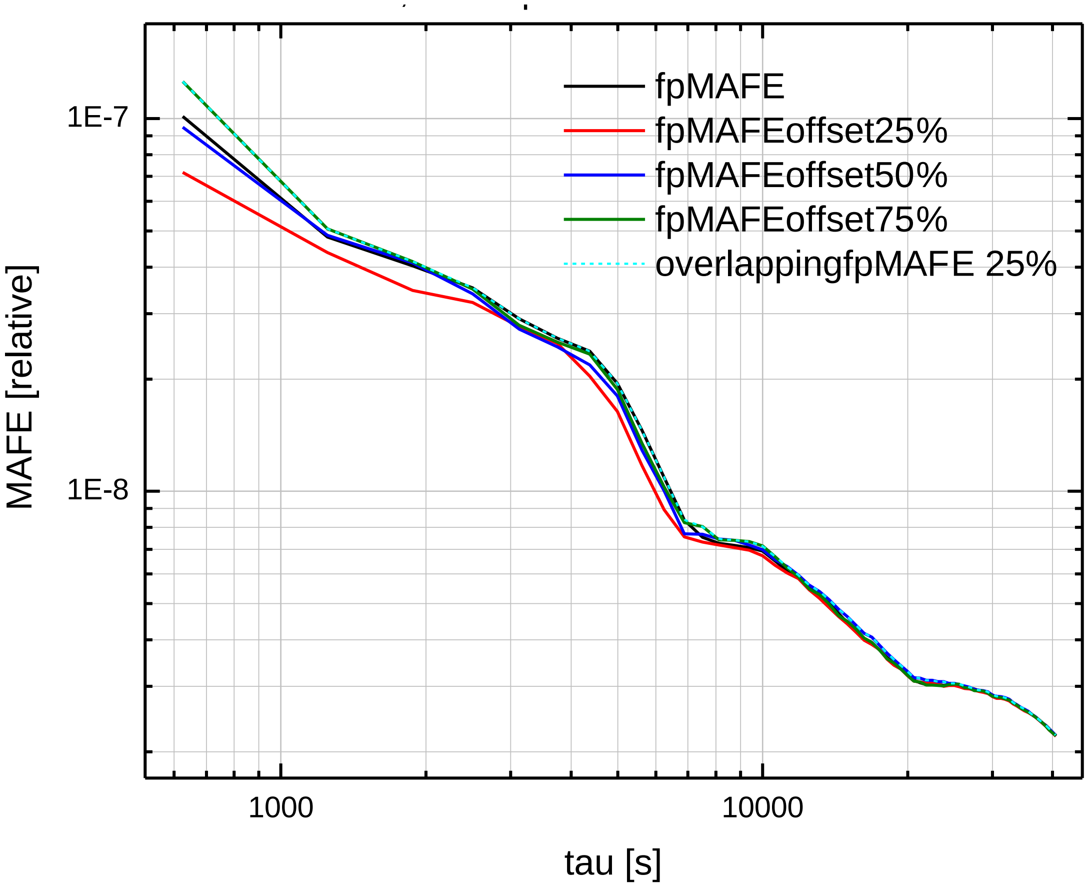


Figure 55: The normal fpMAFE, three offsets and the overlapping fpMAFE results in the same picture. The PDV data used for the calculation is simulated data and the length of selection window is 624.8 seconds (corresponds to 10 000 values) and 0.1% of the values are averaged.

8.3 MAFE compared with linear regression

In addition to the offset and overlapping proposals, another suggestion from the standardization community was the linear regression as a method for determining frequency error (i.e. a slope on the TIE data). This section presents comparison between the normal fpMAFE and the linear regression. The linear regression implemented here uses the same pre-selection windows as the fpMAFE (fixed selection window MAFE) calculation in order to maintain comparability (i.e. the whole data is divided into fixed sized pre-selection windows where the certain percentile of the smallest delays are averaged to obtain one value out of each window). Furthermore, the sliding process uses an equal amount of data (i.e. points after pre-selection) for the calculation of each slope which makes the result more reliable and comparable with fpMAFE. In other words, the linear regression slope at observation window (τ) is calculated from data of two times the observation window as in MAFE since MAFE(τ) uses also two observation windows for calculating the error between the sliding windows. Figure 56 elaborates the calculation principle of the linear regression as the frequency error metric.

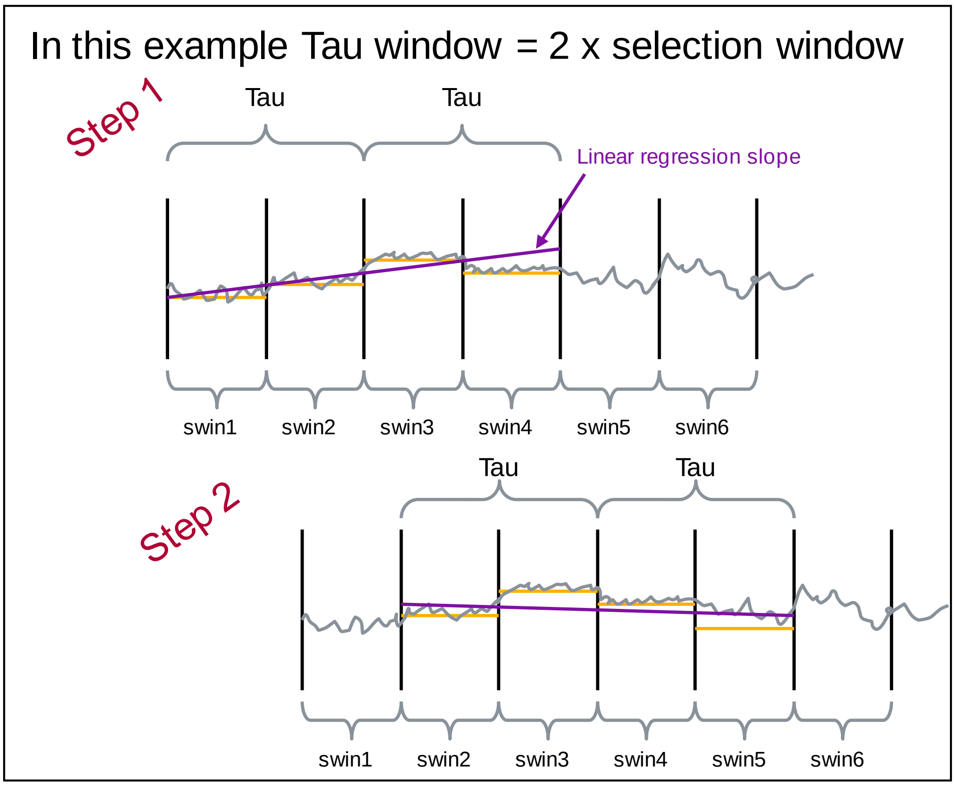


Figure 56: Linear regression calculation explained. Note that the data is first divided into fixed selection windows (swin# in the picture) just like in fpMAFE calculation. Observation window (τ) size is two selection windows in this example. MAFE averages those two selection windows inside τ window and calculates slope between two (τ) windows, whereas, linear regression calculates the slope through all the four windows.

The linear regression slope has been calculated with Equation (12). The comparison between the results of linear regression and fpMAFE from the simulated PDV data are presented in Figure 57. It can be seen that the linear regression and fpMAFE performs nearly perfectly equally. Moreover, other PDV data were tried as well and similar results have been obtained [38].

$$slope(j) = \frac{\left(N\sum_{i=j}^{j+N} X_i Y_i - \sum_{i=j}^{j+N} X_i \sum_{i=j}^{j+N} Y_i \right)}{\left(N\sum_{i=j}^{j+N} X_i^{\ 2} - \left(\sum_{i=j}^{j+N} X_i \right)^2 \right)} \tag{12}$$

Where:

N is the number of values taken into consideration;

j is the sliding index of the slope calculation;

i is the sum index at each sliding step;

X_i is time value at instant i;

Y_i is averaged fixed selection window value at time X_i;

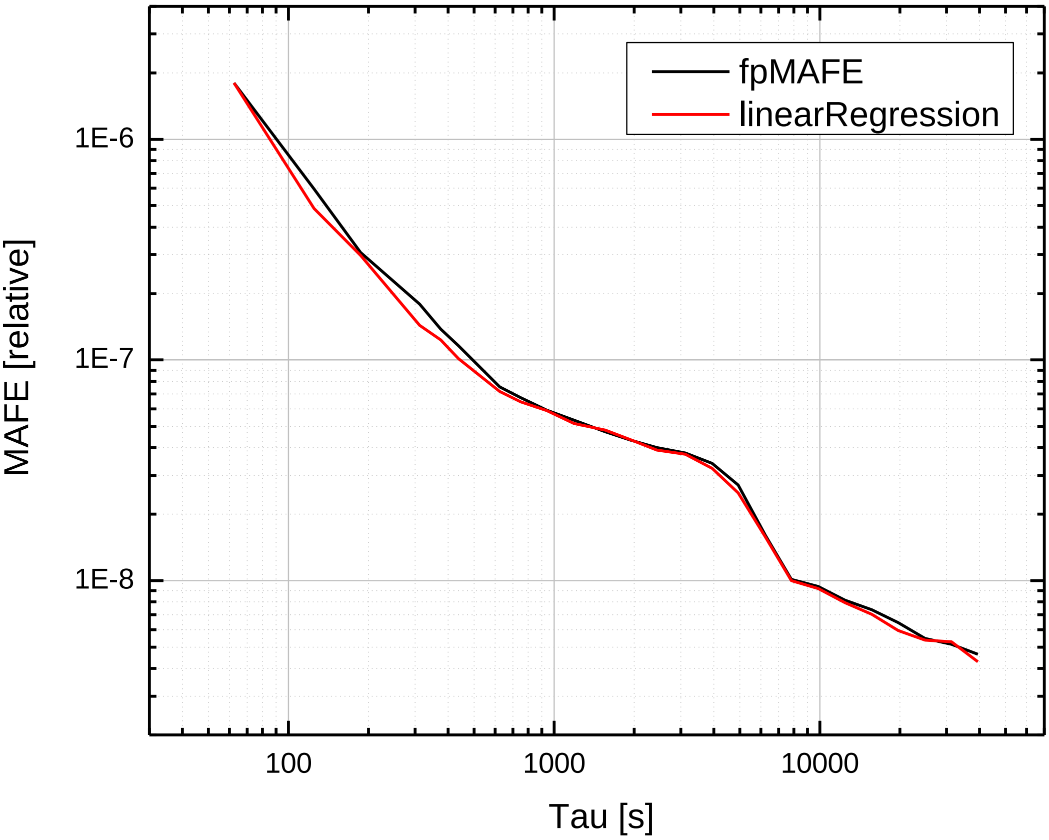


Figure 57: fpMAFE and the linear regression perform equally well on the same simulated data.

8.4 MAFE compared with real packet slave clock performance

Comparing the packet metrics with real packet clock output is the only possibility to prove the ability of metrics to estimate the clock stability from the input PDV data. Moreover, calculating the MAFE metric from the packet clock output TIE indicates two important things. First, the leftmost end of the MAFE curve of the packet clock shows the largest detected frequency error at the clock output (without significant averaging). Secondly, the slope in the right depicts potentially better stability by averaging more.

Figure 58 compares the MAFE calculated from the PDV data against the MAFE calculated from the outputs of two different packet slave clocks. The stability of the packet clock's oscillators can be determined from the picture by extending the horizontal lines (blue and black) to cross the packet MAFE curve (red curve). The stability of packet clock $_1$ is around 1000 seconds and the stability of oscillator $_2$ is approximately 13 000 seconds. The latter easily satisfies the requirement of 50 ppb (the purple mask in the picture, the green mask is for oscillator stability of 1000 s). The whole curve under the 50 ppb stability mask indicates that the required stability can be achieved with the oscillator in use.

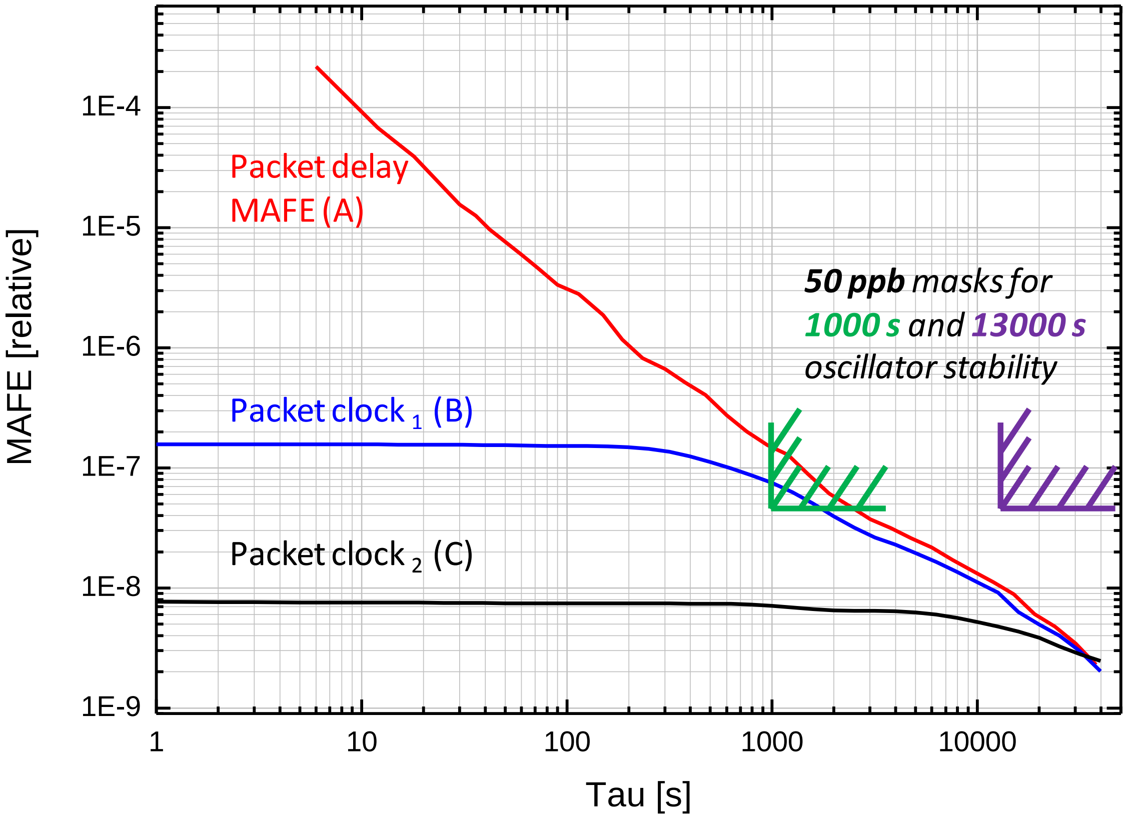


Figure 58: MAFE is calculated from the input packet delay data (A) and from two different oscillator outputs (B and C). 50 ppb masks for both oscillators are drawn in the picture. The required stability of 50 ppb is reached only with the more stable oscillator (C) at this delay profile.

8.5 Alternative methods for testing ToP performance

Although, MAFE has proven to be robust and applicable in different environments there are also some completely different alternatives for performance verification in terms of, for instance, frequency synchronization (i.e. frequency stability). ITU has already specified test cases for testing the performance and also a competing frequency stability metric has been proposed for the ongoing standardization. These alternative methods are briefly discussed here.

8.5.1 Test cases

One of the recommendations of the IEEE 1588 telecom profile standardized by ITU-T, G.8261 specifies test cases for determining the packet clock performance. Tests are alternative (i.e. competing) methods for characterizing packet network applicability for frequency synchronization.

Test cases specify a certain test topology (e.g. Gigabit Ethernet chain of 10 switches) for measurements and unique load scenarios of which timing performance is evaluated at the slave clock. Load scenarios disturb the performance of simultaneous synchronization flow. For example, the same traffic scenario could be added at each switch of the chain cumulatively by traffic generators to congest the network on purpose. Thus, if the clock stability stays within stability specifications, the system has passed the test case. Otherwise, the timing solution is not robust enough and has to be reconsidered. [6]

MAFE and other metrics are more flexible since they can be applied in any network with any load scenarios.

8.5.2 Another frequency stability metric

As mentioned, some competing frequency stability metrics have been proposed for the standardization body. In this section, MAFE is compared with a proposal by France Telecom, a member of the standardization organization. The proposal consists of taking a global minimum delay of the PDV data and draws a certain line (floor delay window) above it to indicate the limit of the tolerable minimum delay change (depicted in Figure 59). For example, a floor delay window of 60 µs above the minimum delay.

PDV data must be divided into time windows (i.e. observation windows) in order to obtain continuous results. The proposal states that the PDV tolerance is adequate if a certain number of packets over each fixed observation windows stay within the floor delay window. For instance, when targeting 50 ppb, within 2000 s observation windows, selected minimum packets must not exceed 100 µs delay from the global minimum due to 100 µs divided by 2000 s is 50ppb. Consequently, delay floor topping the floor delay window indicates performance worse than targeted which leads to deteriorated performance. [39]

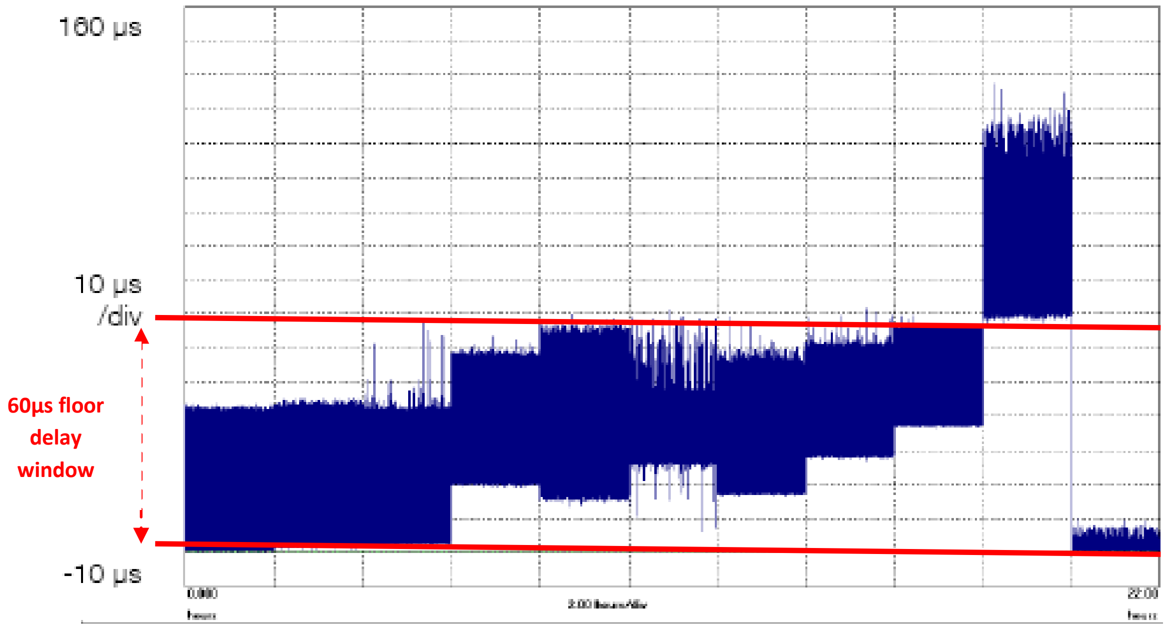


Figure 59: PDV measurement with relatively large delay floor jumps. Penultimate jump exceeds the limit which indicates deteriorated synchronization performance according to this metric. This figure is created by the author of the proposed metric. [39]

Hereafter, MAFE is proven to be a more suitable metric for estimating reliable frequency stability even in somewhat harsh environments. One of the PDV measurement data used in this thesis works perfectly as an indicator that MAFE can handle situations where the other metric struggles to meet the requirements. The France Telecom proposal is applied here so that 1% of the timing packets delays should be within the floor delay window at each observation window of 2000 s. The 100 µs floor delay window was applied for targeting the stability of 50 ppb. An envelope function was created for indicating the largest delay of 1% of the smallest delays (i.e. the delay limit of which includes the 1% of the smallest delays). In order to match the requirement of 50 ppb, the envelope should be within the floor delay window of 100 µs.

Figure 60 illustrates the performance interpretation of the proposed metric for the PDV data where MAFE produces desirable outcome, i.e. MAFE(τ =2000s) = 40 ppb. Also in the case of simulated slave clock, maximum frequency error at 2000 s averaging capability was 40 ppb. The black curve on the background is the minimum delay from every 100 values (i.e. the fixed window percentile of 1% from the window length of 100 delay values). The global minimum is derived loosely from the bottom edge of the black curve so that outliers are neglected. The France Telecom proposal included that the floor delay (i.e. the envelope of 1% in this case) amplitude should not exceed 100 µs limit. The red curve in the picture depicts envelope curve of 1% of the smallest values from 2000 s fixed observation windows (i.e. each point in the curve depicts the largest delay of the smallest 1% values within each observation time widow). The purple curve, in turn, depicts the average of the same 1% of the smallest delays from 2000 s time window. One can see clearly that the curves are 300 and 400 µs above the global minimum line (100 µs being the tolerable limit) proclaiming that 50 ppb frequency stability is not even close. Therefore, the lime-color curve was drawn at the minimum of 1% average curve instead of the global minimum which, nevertheless, did not meet the requirements either.

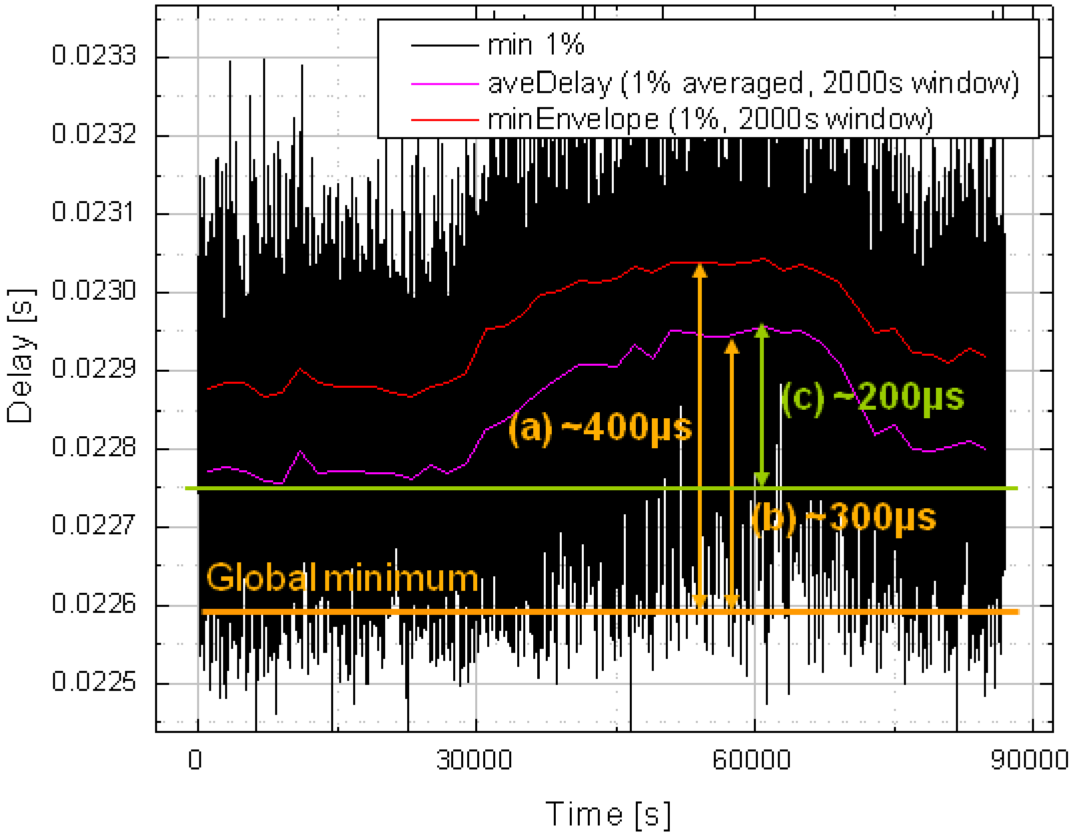


Figure 60: FT proposed a metric that would measure the certain percentile of minimum delays and compare that to the global minimum of the whole data. Thus, if the envelope of 1% of the smallest packets of 2000 s observation window exceeds 100 µs, the 50 ppb limit is outreached. The figure illustrates that neither of the cases would satisfy the 50 ppb limit in that particular data. MAFE, in turn, would fit into the limit of 50 ppb in that case.

9. MATIE and MATIE modification performances

Now that the frequency synchronization metric MAFE has been proven to be effective and applicable in various situations, the performance of the phase stability metric version, MATIE, is discussed. It was discussed in Chapter 6 that MATIE actually estimates slightly worse stability than the simulated packet clock with a 2nd order PLL filter on the artificially created data. Therefore, research on an improved MATIE is conducted as well, and the results are analyzed in this chapter. Target was to create and study a better version of MATIE (i.e. metric with equal filtering capability as 2nd order low-pass filter).

9.1 MATIE modification introduced

The study of an improved MATIE is started by introducing the artificially created (with mathematical formula) input TIE data (i.e. the input timing signal for a slave clock) which has been used already in previous examples, see Figure 37, Figure 41, and Figure 43. The artificially created input TIE data (depicted in Figure 61) corresponds to the input PDV data that would illustrate quite exceptional conditions in the network. Thus, it might not be very likely to find a similar situation in real life measurements but it provides an interesting and informative test scenario. The signal has equally high negative and positive spikes and a constant slope between the spikes. Outside of the pulse, all values are set to zero. The slope between the negative and the positive spikes is the significant part of the pulse for the metrics like MATIE and MAFE. With the small τ-windows, the maximum averaged error is always found from that slope until the length of two windows exceeds the length of the slope. Accordingly, the MATIE curve at small observation windows sizes is always an ascending linear slope.

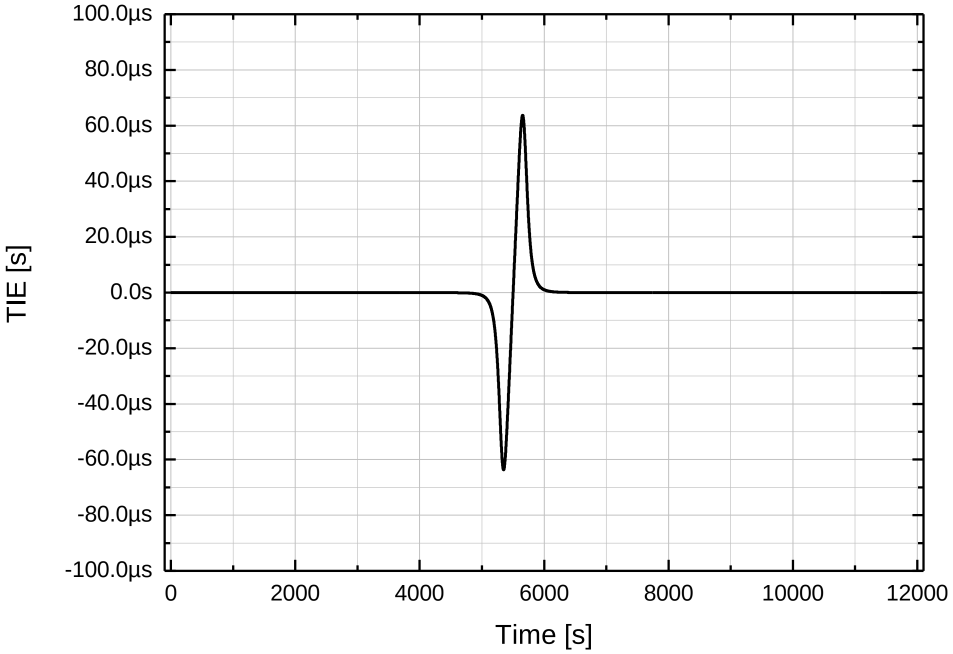


Figure 61: The artificially created input TIE signal which consists of 12 000 samples. The actual pulse is no longer than 1000 s.

Notably, due to the nature of MATIE, the maximum error value regardless of the τ-window length is always discovered exactly where windows are slid so that the center of the two adjacent windows is in the middle of the TIE data (i.e. where the negative values are turning to positive, at about time 5500 s). In that case, all the values in the first window are negative and, respectively, all the values in the second window are positive (see Figure 64). The discovery of the properties helped in finding a solution.

The simulated PLL low-pass filters used earlier were applied in the phase stability context as well. MATIE can be compared with the simulated slave clock performance by calculating MTIE from the simulated PLL filter output (explained in Section 6.2). As mentioned in Figure 43 and repeated Figure 62, MATIE is a slightly worse filter than a simulated 2nd order Butterworth low-pass filter in case of the mathematically created data.

Since the 2nd order low-pass filter performs better than MATIE only with larger observation window (τ) values the behavior of larger τ-windows needs to be investigated. First of all, different number and size of rectangular sliding windows (e.g. three sliding windows with the weights emphasizing the middle window) were tried out but turned out to be unsuccessful modifications. Rectangular windows would have the same slope after the peak regardless of the amount adjacent differently weighted windows. Hence, the fact that the maximum average error can always be found when the pulse is in the middle of the two sliding windows (see Figure 64), resulted in an idea that the window edges must be altered in order to increase the filtering capability at larger (τ) values. Therefore, several common shapes of window functions were implemented and tested while preserving the original MATIE principle of obtaining maximum average error of two consecutive sliding windows. For instance, triangular, Hamming, Hann, parabola and Tukey windows were implemented and the results are depicted in Figure 62. Many of the modification resulted in an even better filtering capability than the desired 2nd order low-pass filter. The steeper the downward slope of a phase stability curve is the better the filtering capabilities are.

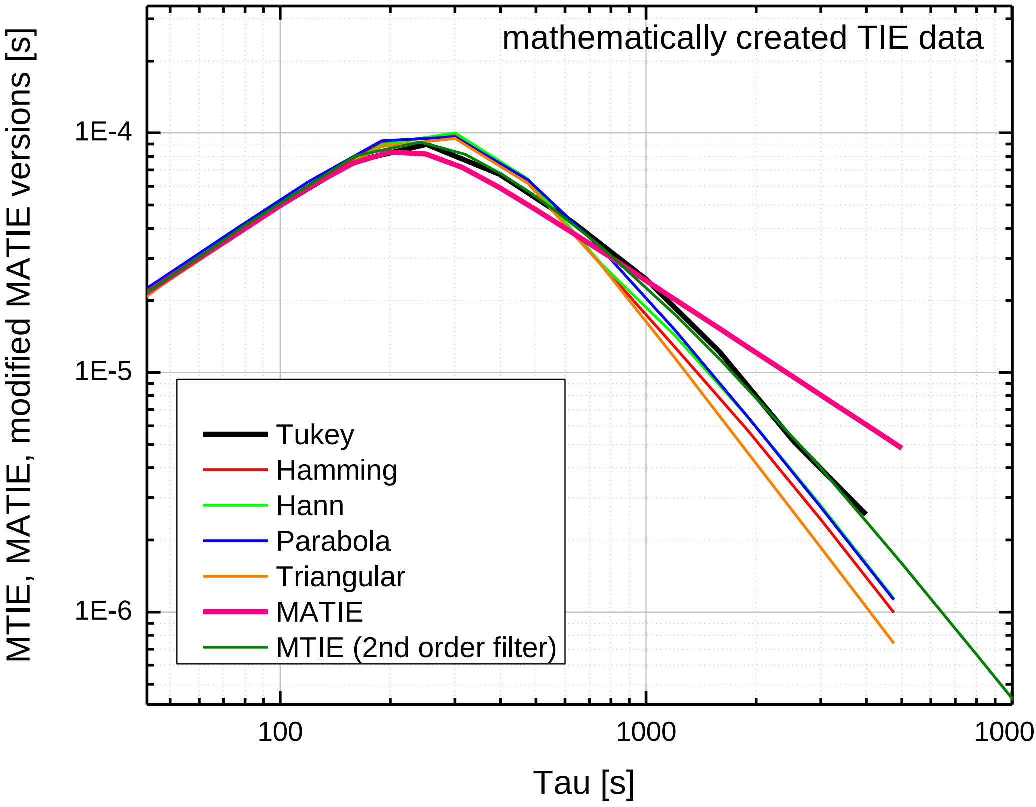


Figure 62: Tukey, Hamming, Hann, Parabola, triangular sliding windows, 2nd order filter MTIE and the normal (rectangular windows) MATIE were compared against the artificially created TIE data (Figure 61). Note that the peak of the curves goes higher in non-rectangular windows because the shape of the windows emphasizes the values in the middle of the window. Particularly, the peak signifies the situation where the observation window size is nearly half of the actual TIE pulse when the negative and positive spikes are approximately in the middle of the observation windows.

As the initial goal was to simulate the 2nd order filter, the performance of which is generally considered adequate, Tukey-window was found to have relatively equal filtering capability. Many real PLL filters are actually 2nd order filters, although, 3rd order filters are also implemented in a few packet slave PLL's.

Tukey is not the most commonly known window function but it is certainly known in the area of DSP (digital signal processing). The Tukey function is a generalization of a combination of a rectangular and Hann functions with a parameter α which can have values from 0 to 1. For example, with the value 1 the Tukey window is actually a rectangular window and, respectively, with the value 0, it is the Hann window function. The Hann window function is the so called "raised cosine" function. A picture of the Tukey window function (with parameter α = 0.5) is depicted in Figure 63.

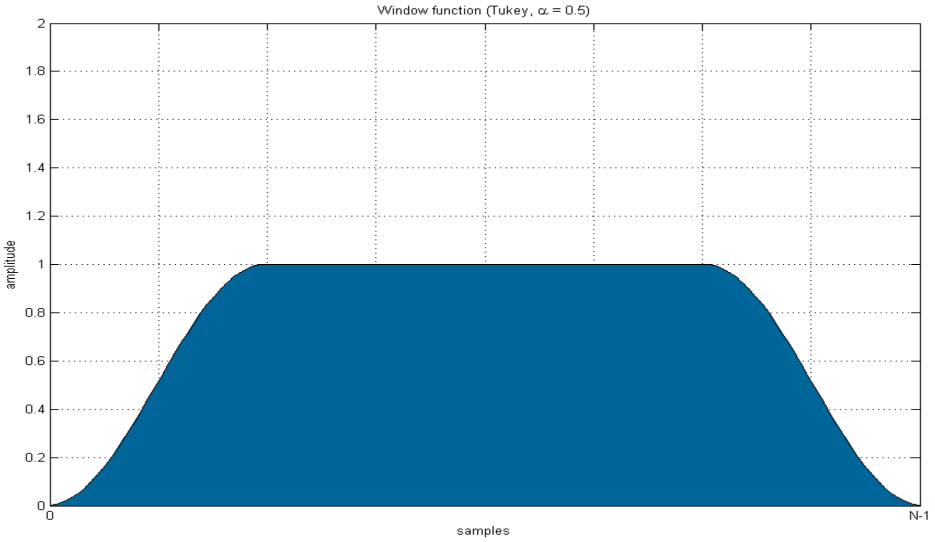


Figure 63: Tukey window function with the parameter α = 0.5. The figure illustrates the shape of the sliding window (i.e. weights for each value inside the window) used in MATIE modification.[40]

The mathematical formula of Tukey window function is presented in Equation (13) [40]. The outputs (*w(n)*) are used as the weight coefficients for the values inside sliding MATIE observation windows (τ). The Tukey version of MATIE is named MATIE modification hereafter.

$$w(n) = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\pi \frac{|n| - \alpha\frac{N}{2}}{(1-\alpha)\frac{N}{2}}\right)\right], \ when \ \ \frac{\alpha N}{2} \leq |n| \leq \frac{N}{2} \\ 1, \ when \ \ 0 \leq |n| \leq \frac{\alpha N}{2} \end{cases} \qquad (13)$$

Where:

N is the width of the window in number of samples;

n is the index running from –N/2 to N/2;

α is the Tukey coefficient;

w(n) is coefficient for each value inside window;

From a different point of view, MATIE can be also understood as a convolution of input data and a MATIE window function consisting of a positive and a negative rectangular window next to each other. The convolution essentially sums the overlapping area of the convolving and convolved function while the convolving function is sliding through the data. MATIE works similarly with the exception that the result of the convolution at each point is averaged by the number of samples in one sliding observation window. Therefore, the shape of MATIE curve is similar to the convolution result but scaled down all over the curve due to the averaging. Figure 64 illustrates the convolving function in case of the normal MATIE and MATIE modification. The artificially created data also plotted there at the point where maximum error between the two sliding windows would be found.

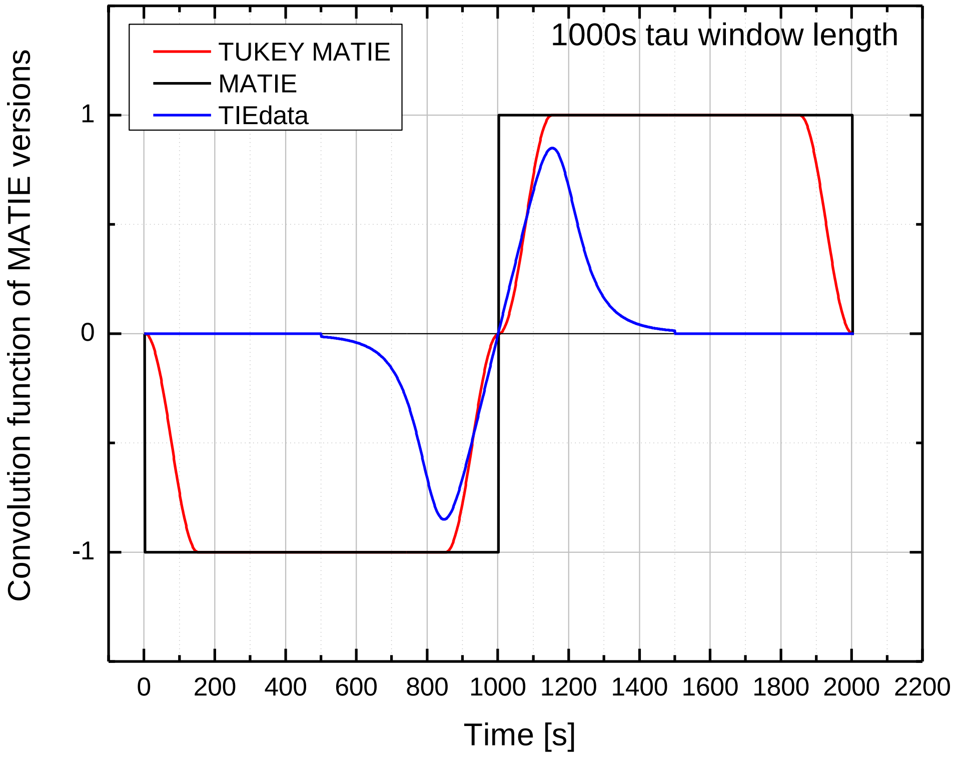


Figure 64: MATIE modification (Tukey) and the normal MATIE compared as convolving window functions. As window size increases, it is obvious that Tukey will perform better (i.e. the maximum average error will be less than compared with the normal MATIE).

9.2 Normalizing the MATIE modification results

As presented, MATIE calculated from the mathematically created input TIE data results in a nearly triangularly shaped MATIE curve as a function of the observation window with a rounded summit (depicted in Figure 62 and Figure 65). The ascending linear slope with low τ-values (i.e. the left side of the peak) is the implication from the steep slope between the spikes in the TIE data (i.e. a constant frequency error). Respectively, the descending slope (i.e. the right side of the curve) is the result of the filtering capability of the MATIE metric. The wider the sliding window the smaller the error becomes, due to the averaging factor (i.e. the number of zero values increases within the window).

The MATIE modification has better filtering capability than the normal MATIE but in order to compare the MATIE modification with MATIE, it is important to normalize the results at the same level (i.e. the effective areas of the windows were harmonized to correspond the weight of a rectangular window, τ * 1). As the area of Tukey window (see

Figure 64) is clearly less than the area of rectangular window, multiplying MATIE modification results by a normalizing coefficient is required. The exact coefficient is 1.1764 which can be mathematically calculated with the definite integral of the Tukey window. Without the coefficient, the left side slope would be below the normal MATIE which obviously is not correct behavior since the filtering capability should not make any difference with low τ-values. Note that the coefficient is used in previous and forthcoming graphs comparing MATIE and the MATIE modification.

In Figure 65, the MATIE modification is normalized as described and, as a result, the left side of the curve is exactly in-line with the normal MATIE. In addition, the 1st and 2nd order low-pass filter MTIE's are depicted also to verify the necessity of normalizing the results.

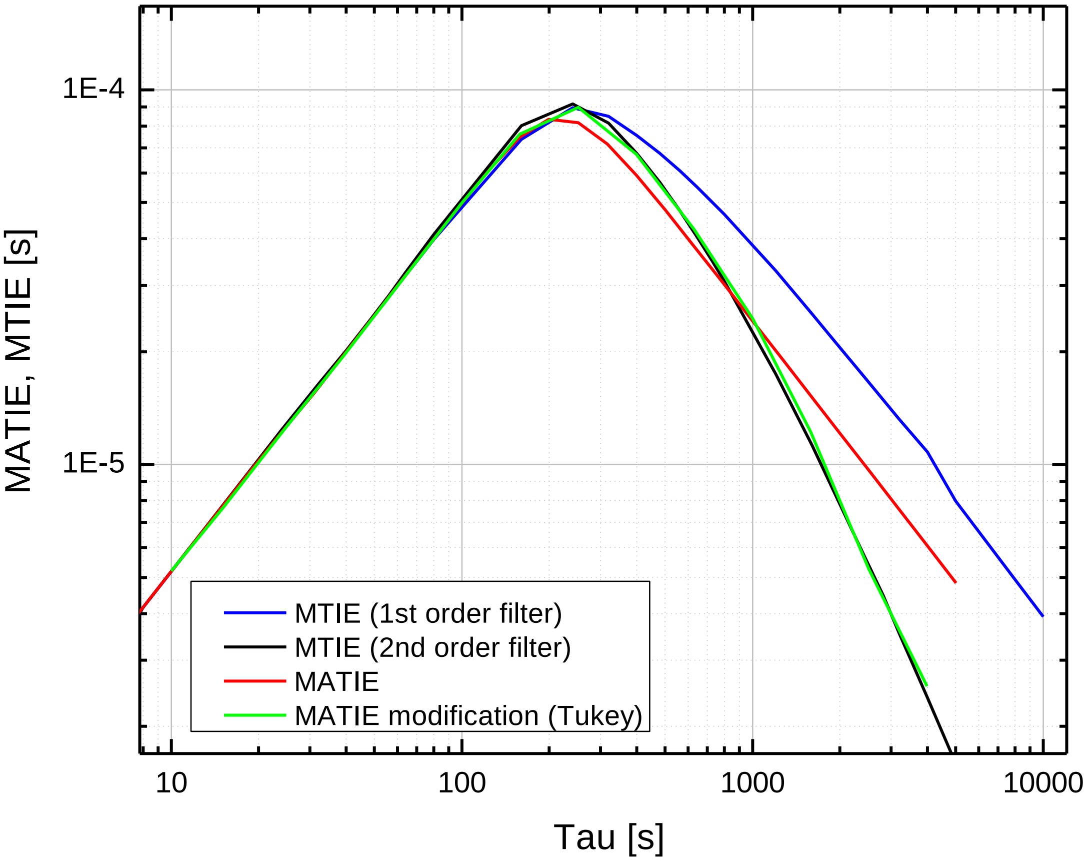


Figure 65: MATIE and the MATIE modification are compared with MTIE calculated from the simulated 1st and 2nd order PLL low-pass filter outputs. The artificially created TIE data is used as the input signal. The MATIE modification (Tukey with parameter alpha = 0.7) aligns perfectly with the 2nd order low-pass filter.

9.3 Research on MATIE and MATIE modification performances

In this section, the MATIE and MATIE modification performances are objectively investigated on real PDV data after stating that the MATIE modification performs desirably on the artificially created data. The performance research includes comparing the performance against real packet slave clock phase stability as well as simulated packet clock performance with the PLL low-pass filters and MTIE combination as explained previously.

9.3.1 MATIE and MATIE modification compared against simulated clocks

Comparison between MATIE, MATIE modification, and packet clock simulations with 1st, 2nd, 3rd and 4th order low-pass filter MTIEs are presented in this section. As previously, the main focus is on 2nd order low-pass filter performance. The MATIE, MATIE modification and slave clock simulations are calculated similarly as in artificial data case with the exception of adding the packet pre-selection before metrics and PLL filters.

The fpMATIE (fixed percentile window), the fpMATIE modification, and the 2nd order filter MTIE curves are depicted in Figure 66. The better filtering on artificially created data is vanished on real data (i.e. the better filtering of extreme situations is not being of use on real data). Thus, it seems that in practical cases the MATIE modification does not bring advantage.

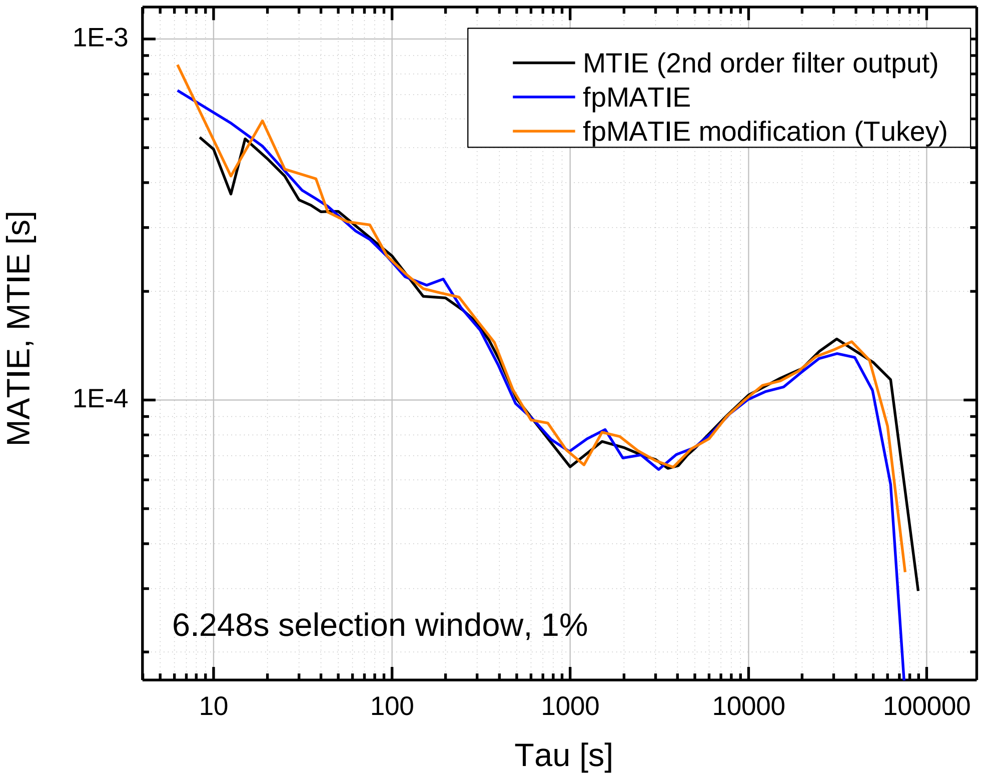


Figure 66: The 2nd order filter MTIE, the normal fpMATIE and the fpMATIE modification perform equally with the same selection parameters.

In theory, in the case of an extreme pulse in the PDV data, the MATIE modification and 2nd order MTIE should perform better than the normal MATIE when the observation window is clearly larger than the pulse length. However, the artificial data had the negative and the positive spikes close to each other, which make it a perfect maximum error indicator for MATIE metric as explained earlier but rather unlikely to come across in a real network behavior unless caused by a network malfunction.

Figure 67 shows that there would not be any substantial benefit of using 3rd or 4th order filters since the 2nd order filter performs very similarly to them. Notably, only the 1st order filter is worse on the data.

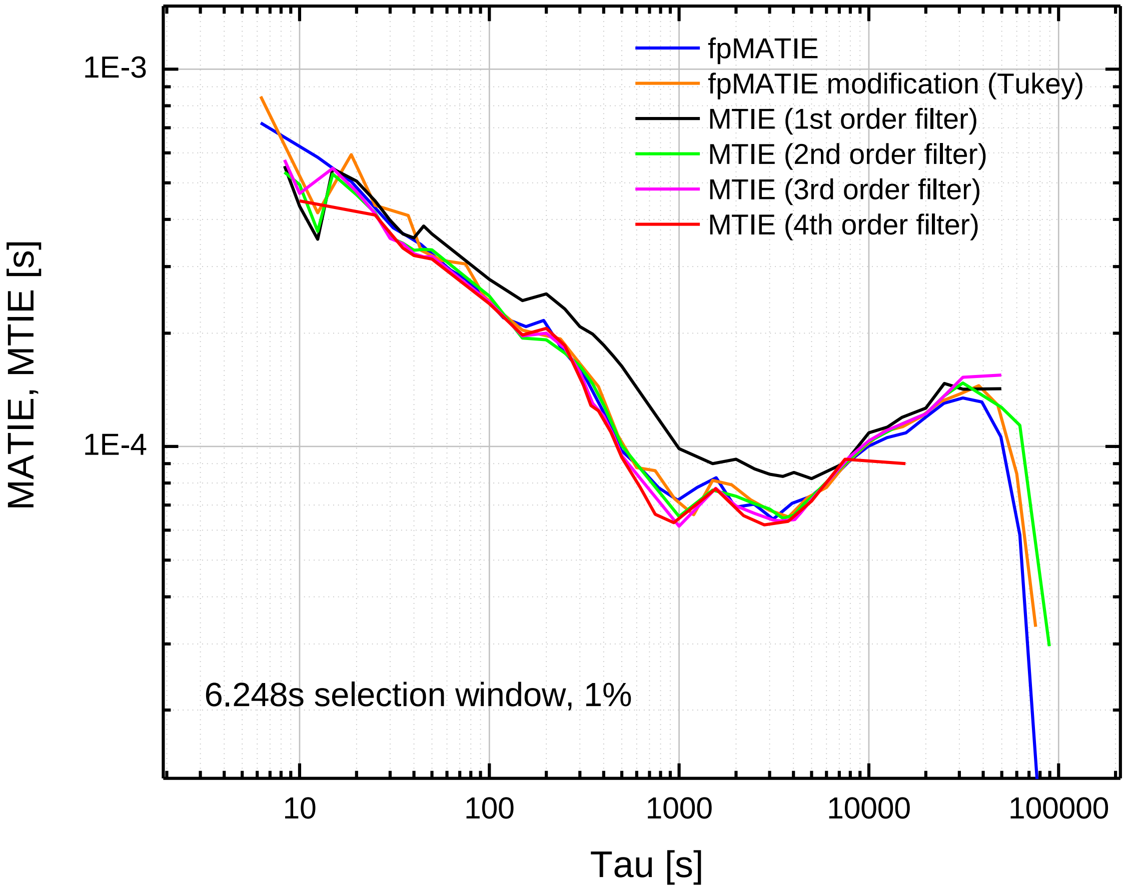


Figure 67: 2nd, 3rd and 4th order performs quite similarly on a real PDV data. 1st order is notably worse than others.

Although modification were made on MATIE for aiming at better phase stability metric, MAFE makes it possible to use the same modification in a frequency stability context. Figure 68 presents fpMAFE with the Tukey window modification against the 2nd order filter and the normal fpMAFE. Results are clear; the shape of the observation window does not affect the frequency metric with any significance. All the three methods indicate exactly similar performance as a function of the observation window (i.e. required oscillator stability for averaging).

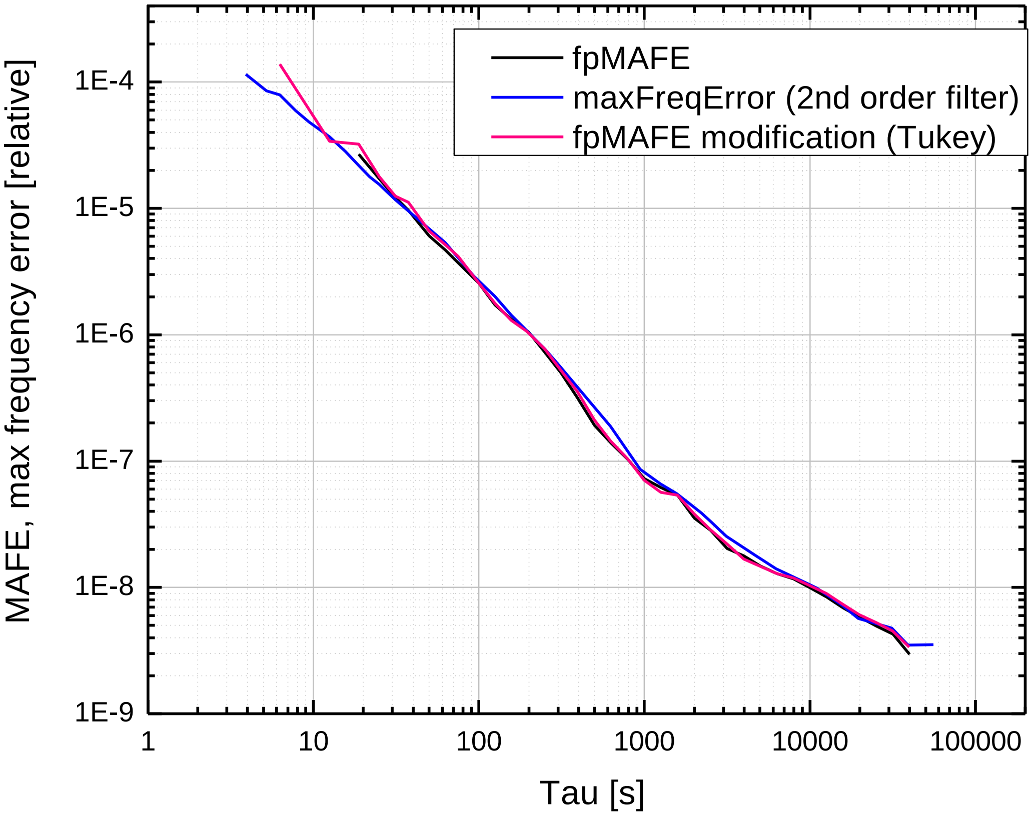


Figure 68: fpMAFE and 2^{nd} order filter max frequency error depicted with the fpMAFE modification (Tukey). Modification does not affect frequency stability with this real PDV data.

9.3.2 Packet MATIE compared with a packet slave clock

The MATIE of the packet delays can be compared with the MATIE calculated from a packet clock output. If the MATIE of the clock is equal or below the MATIE of packet delays then the clock is likely well-designed (i.e. the filtering capability of the slave clock is equal or better than 2^{nd} order and packet selection is adequate). Since the short-term noise and wander are filtered out in the PLL filter of a real packet clock, the metrics calculated from the clock output starts from much lower values than the metrics calculated from PDV even though most of the short-term variation is eliminated by packet selection. On the other hand, long-term variations are not filtered, hence, the curves align closely at the larger τ-values.

But on the other hand, daily network load variation patterns obviously influence the metrics. If the delay variation would be proportional to the variation in artificial data, the differences of MATIE and the MATIE modification could be observable with observation periods of significantly over one day.

Figure 69 depicts the behavior of packet metrics (MATIE and the MATIE modification) compared with the measured packet slave clock performance on the same 4 day measurement with the real packet synchronization equipment. The PDV of real timing packets and the output TIE signal of the packet slave clock were recorded during the measurement period. Note that the traditional clock metrics are calculated from the output TIE data naturally without packet selection methods and, correspondingly, packet metrics (i.e. metrics with packet selection) from the PDV data. Since the τ-window could be extended up to two days and thus longer than the daily variation pattern it was expected that the MATIE modification might bring some improvement. This corresponds to the case of mathematical pulse where the improved performance could be observed when the τ-window extended beyond the pulse length.

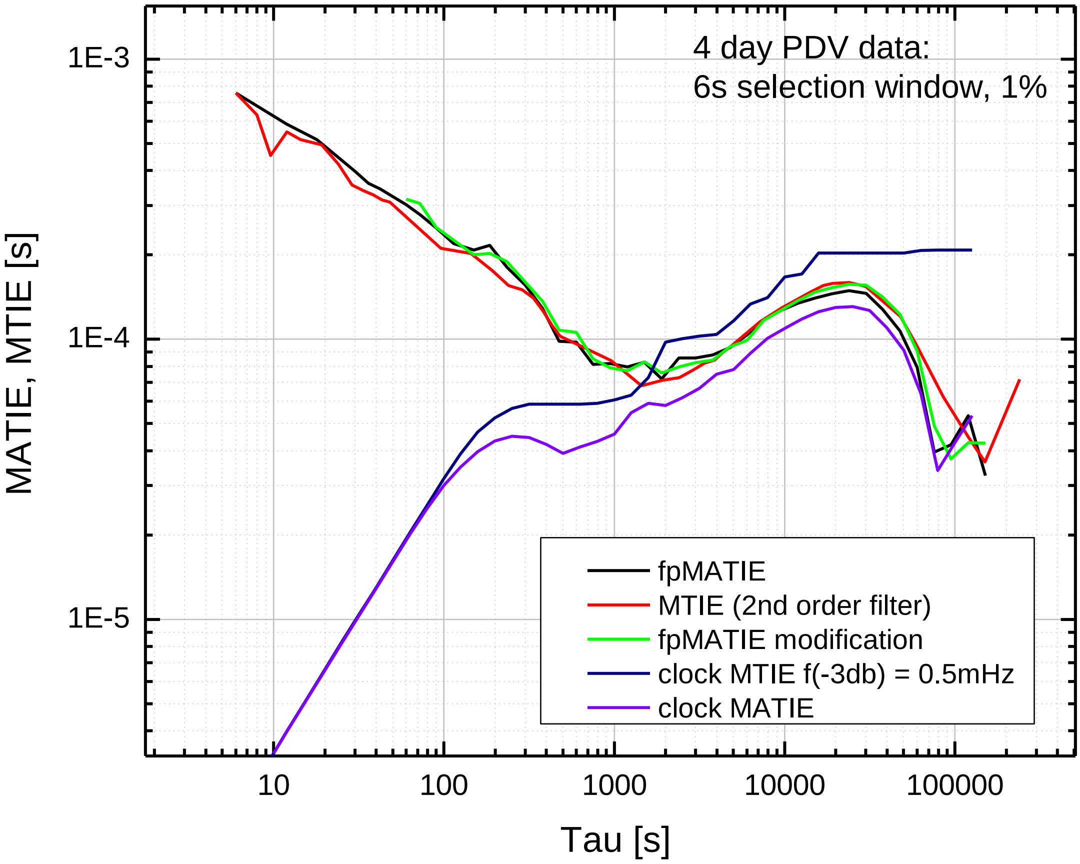


Figure 69: fpMATIE, the fpMATIE modification and the 2nd order low-pass filter MTIE are compared against the packet slave clock output phase stability expressed with MTIE and MATIE. Note, due nature of MTIE, the curve never descends. On the other hand, MATIE performs similar to the packet MATIE after 1000s (i.e. the clock stability in this measurement). The same packet selection parameters were used in every metric calculated from the PDV data; 6s fixed selection window and 1% of which is averaged to obtain a one value from each window.

Since the stability of the oscillator used in the packet slave clock is 1000 s, the curves calculated from the PDV data and the clock output should converge at 1000 s. However, there is a gap between the curves at 1000 s indicating that the packet metrics estimate actually slightly worse stability than the clock can actually produce. In that sense, neither MATIE nor the MATIE modification are not totally optimal but, nevertheless, estimates the performance of a real clock relatively well. From other point of view, clock metrics are below the packet metrics which indicates proper slave clock design. Note, the clock MATIE is lower than the others since it has been filtered once with the PLL filter and then a second time with the MATIE metric.

But on the other hand, a possible phase stability limitation must be addressed. As the PDH MTIE traffic mask indicates, 1000-second phase stability should be better than 18 µs at the traffic interface. MATIE may not be the best metric to satisfy that limit as can be seen from the figure. Although MATIE at different τ-values match the MTIE of a clock at the observation window corresponding to the filter stability, for finding out whether a clock with certain PDV input will satisfy the MTIE mask, a simulation of the actual clock may be needed. The clock simulation would try to perform as closely as possibly with the designed slave clock. Thus, reliable MTIE output can be obtained to see whether the mask is satisfied or not. If not, better network (i.e. less PDV) or better filtering capability is needed to obtain compliance with the mask. In another approach, satisfying the G.823 2M traffic mask (18 µs) could be verified using MAFE, i.e. if MAFE is below 18 ppb at τ-values larger than certain τ_x then a clock with a corresponding PLL filter cut-off frequency will satisfy the mask.

10. Time synchronization without on-path support

In Section 4.2, NTP and PTP were introduced. PTP was told to be able to deliver time synchronization in addition to frequency synchronization. Very high accuracy could be reached when using on-path support such as boundary clocks. As on-path support will not exist for some time there is interest to find out whether it would possible to reach the high accuracy with currently existing technology. The delivery of the absolute time reference is based on the calculation of mean propagation delay, meaning approximation of one-way delay with the four different timestamps, see Section 4.2.2. The method assumes constant and symmetric propagation delays (i.e. uplink and downlink delay should be equal), thus, delay symmetry is also important aspect for delivering the accurate time reference. [9]

Unlike physical medium, network nodes can easily create asymmetries which could be eliminated with on-path support. Some measurements have been conducted to obtain two-way delay data in a network without on-path support. The asymmetry was determined in a network that consisted of 4 high-speed core nodes with fiber and SHDSL access. Note that, the delays are averaged over 5000 s periods in each case in order to filter out the jitter and noise. Moreover, the difference curve is presented to help the interpretation. Figure 70 shows that with fiber access, the delay asymmetry stays around 1 µs producing a 500-ns error, which is tolerable since the time synchronization accuracy requirement in mobile base stations is commonly ± 1.5 µs.

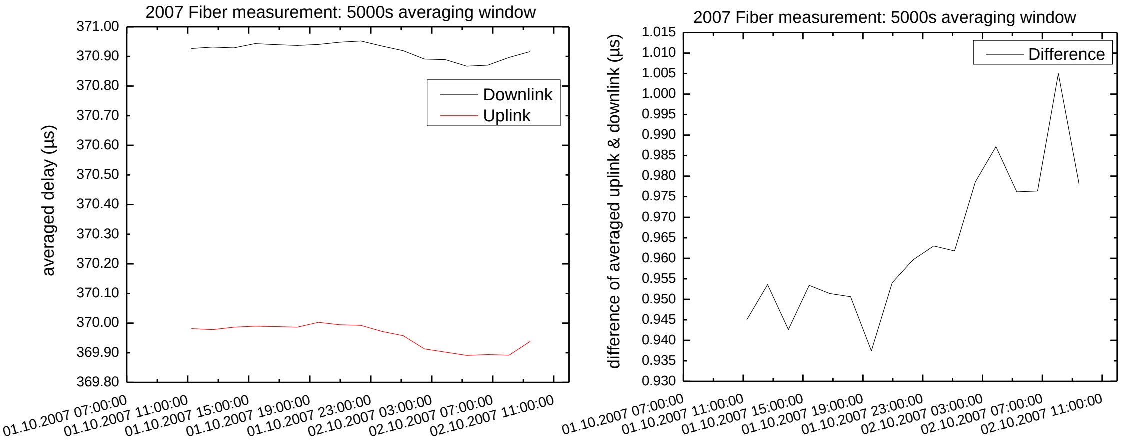


Figure 70: The same delay symmetry measurement with fiber as the last mile access technology. The rightmost figure depicts the difference (i.e. downlink – uplink). The delay asymmetry is approximately 1 µs.

Figure 71 depicts the same situation with SHDSL as the access technology. The delay asymmetry is significantly larger (i.e. around 4.5 µs). However, the daily variation is within 300 ns so the asymmetry could perhaps be compensated with fixed asymmetry correction.

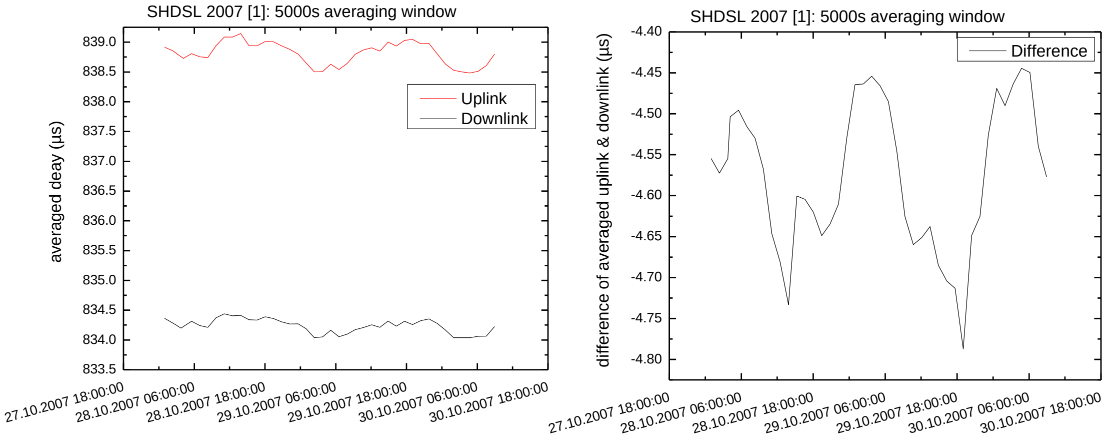


Figure 71: SHDSL delay symmetry, downlink data experiences approximately 4.5 µs less delay than uplink.

Figure 72 illustrates another SHDSL delay symmetry measurement. In this case the network consisted of 7 core nodes. In addition, the SHDSL equipments were different. The delay asymmetry is larger, this time uplink was faster than downlink, and the difference is not as stable as in Figure 71. The difference varies within 3 µs which is not tolerable anymore for accurate time synchronization with fixed asymmetry correction. Note, there was also a significant delay jump of 100 µs in that particular measurement in downstream only causing 50-µs error. This was removed before calculating the curves so that the daily asymmetry pattern could be determined more accurately. It proves that network can have unexpected behaviors which affect the synchronization and makes any fixed asymmetry corrections worthless if the jumps cannot be avoided.

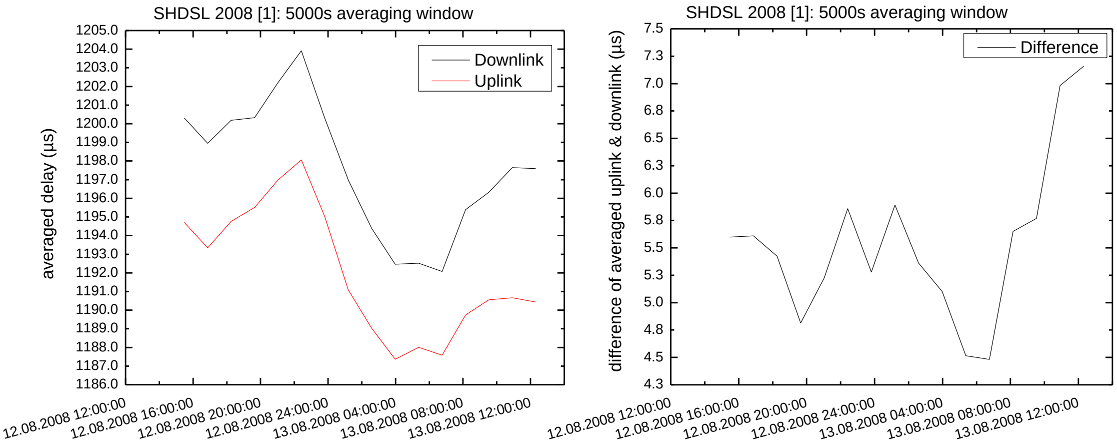


Figure 72: Another SHDSL delay symmetry measurement. In this case the delay asymmetry difference is varying more (i.e. approximately 3 µs).

Since the delay asymmetry is a significant problem already in the relatively simple network, on-path support for delivering time synchronization seems to be mandatory.

11. Conclusions

Since TDM-based PDH transport does not scale into larger bitrates with reasonable costs, it is evident that mobile backhaul is in transition towards packet based transports (e.g. Ethernet). This introduces new challenges for synchronization. Several technologies have been developed for synchronization delivery but timing over packet (ToP) solution has become the most attractive one. ToP has the lowest total cost of ownership since operators can abandon the traditional PDH and no GPS is needed.

Timing over packet has suffered from lack of trust since there has been no applicable metrics for performance verification. Operators have been skeptical about the possibility to deliver accurate synchronization over packet network. Thus, active research has been conducted in the area of packet synchronization and synchronization metrics.

During past three years, the behavior of several metrics has been investigated for the performance verification purposes and to help design proper slave clocks. The applicability of the traditional metrics such as TDEV and MTIE in the packet environment has been under discussion. Further, MAFE has been proposed as a limit metric for frequency stability.

This thesis began with general introduction to the cellular network synchronization methods and the mobile technology synchronization requirements. Thereafter, the focus was directed to timing over packet and the protocols capable of timing distribution e.g. PTP and NTP were introduced. Then the focus was further directed to frequency synchronization without on-path support and the challenges caused by the packet environment such as packet delay variation (PDV) were described. PDV measurements have been in important role in the metrics study to provide realistic data.

After the introduction to the packet synchronization, the traditional synchronization metrics were discussed in detail along with the timing packet selection methods which are necessary for the metrics to estimate the synchronization performance from packet timing signal. Thereafter, new packet metrics (MAFE and MATIE) were introduced. The applicability of MAFE was tested with several methods. MAFE was compared with simulated slave clock performance, linear regression and against another metric, floor delay window method, proposed for the same purpose. The fixed window pre-selection was investigated in detail. The study included offsetting the window positions and using overlapping selection windows. After proving the robustness of MAFE, the MATIE metric performance was investigated also against slave clock simulations and then a new modified MATIE was developed to enhance the performance. However, the better performance of the MATIE modification could only be seen on the artificially created test data.

Finally, after the metrics study, time synchronization without on-path support was discussed briefly. The time synchronization over packet network apparently requires on-path support from every node along the synchronization path.

All in all, since the standardization has been active in the area of the ToP synchronization, the value of the conducted research is significant. A number of the results presented in this thesis have been delivered to the ITU-T standardization. The telecommunication profile for frequency synchronization was standardized in June 2010. However, the discussion about the packet metrics (e.g. MAFE and MATIE) for the appendix was postponed until the end of 2010 due to the need to advance the time synchronization work. Regardless of that, MAFE and MATIE are one of the most promising estimates (i.e. metrics) for synchronization performance. Since time synchronization research and standardization are in progress, the issues have been only briefly discussed in this thesis. There are many possibilities for further research in the area of accurate time synchronization.

References

[1] Cisco. *Synchronous Ethernet: Achieving High-Quality Frequency Distribution in Ethernet NGNs*. White Paper. 2008. Accessed: 7. June 2010. Available: `http://www.cisco.com/en/US/prod/collateral/routers/ps9853/white_paper_c11-500360.pdf.`

[2] Ericsson. *Synchronization in packet-based networks: challenges and solutions.* White Paper. January 2009. Accessed: 10. March 2010. Available: `http://www.ericsson.com/res/docs/whitepapers/synchronization_in_packet_based_networks.pdf.`

[3] Infonetics Research. *Mobile Backhaul: 2009 in review, 2010 in view*. Market Research Paper. Accessed: 1. June 2010. Available: `www.infonetics.com`.

[4] Ferrant J-L., Gilson M., Jobert S., Mayer M., Ouellette M., Montini L., Rondrigues S and Ruffini S. *Synchronous Ethernet: A Method to Transport Synchronization.* IEEE Communications Magazine. October 2008. Available: `http://www.zarlink.com/zarlink/SyncE-IEEE-Sep08.pdf`

[5] Symmetricom. *Advances in Backhaul Synchronization - Maximizing ROI*. White Paper. June 2008. Accessed: 10. June 2010. Available: `http://www.symmetricom.com/resources/downloads/white-papers/.`

[6] ITU-T. *Timing and Synchronization aspects in Packet Networks.* 2008. Recommendation G.8261.

[7] Infonetics Research. *Mobile backhaul Equipment and Services. Biannual Worldwide and Regional Market Share, Size and Forecasts.* Market Analysis. November 2009. Accessed: 15. June 2010. Available: `www.infonetics.com`.

[8] ITU-T. *Draft of Recommendation G.8260 (no name yet).* 2010.

[9] JDSU & Calnex Solutions. Implementing IEEE 1588v2 for use in the mobile backhaul. *White Paper.* 2009. Accessed: 1. June 2010. Available: `http://www.jdsu.com/product-literature/IEEE1588v2 _wp_cpo_tm_ae.pdf.`

[10] Symmetricom. *Deployment of Precision Time Protocol for Synchronization of GSM and UMTS Base Stations*. White Paper. 2009. Accessed: 3. June 2010. Available: `http://www.symmetricom.com/resources/downloads/white-papers/.`

[11] ITU-T. *Timing characteristics of primary reference clocks.* 1998. Recommendation G.811.

[12] Agilent Technologies. *Understanding Jitter and Wander Measurements and Standards. Second Edition.* Technical Article. 2003. Accessed: 25. July 2010.

Available: http://cp.literature.agilent.com/litweb/pdf/5988-6254EN.pdf.

[13] ITU-T. *The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy.* 2000. Recommendation G.823.

[14] Eidson, J. C. *Measurement, Control and Communication Using IEEE 1588.* Book. London, Springer, 2006, pages 9-33.

[15] RAD data communications. *Synchronization and Timing in Packet-Based Mobile Backhaul.* White Paper. December 2009. Accessed: 1. June 2010. Available: http://www.rad.com/static-files/Static%20Files/MediaItems/11678_SyncToP.pdf.

[16] Heavy Reading. *Carrier Ethernet Switch/Router, Ethernet Backhaul Market Tracker.* Market analysis. July 2008.

[17] Infonetics. *Mobile Backhaul Equipment and Services, Biannual Worldwide and Regional Market Share, Size, and Forecasts.* November 2009. Available: www.infonetics.com.

[18] IETF. *Network Time Protocol Version 4: Protocol and Algorithms Specification.* 2010. RFC 5905.

[19] Mills, D. L. *NTP Architecture, Protocol and Algorithms*. July 2007. Accessed: 7. July 2010. Available: http://www.eecis.udel.edu/~mills/database/brief/arch/arch.pdf.

[20] Mills, D. L. *Internet Time Synchronization: the Network Time Protocol*. Article. University of Delaware, Electrical Engineering Department. Accessed: 1. June 2010. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.9287&rep=rep1&type=pdf.

[21] EndRun Technologies. *PTP/IEEE-1588 Frequently Asked Questions.* Web page. Accessed: 24. May 2010. Available: http://www.endruntechnologies.com/ptp-ieee-1588-faq.htm.

[22] Symmetricom. *IEEE 1588 Precision Time Protocol – Frequency Synchronization Over Packet Networks.* White Paper. 2008. Accessed: 31. May 2010. Available: http://www.symmetricom.com/resources/downloads/white-papers/.

[23] EndRun Technologies. *Precision Time Protocol (PTP).* White Paper. Accessed: 15. May 2010. Available: http://www.endruntechnologies.com/pdf/PTP-1588.pdf.

[24] Brilliant Telecommunication. S*ynchronization Deployment Considerations for IP RAN Backhaul Operators.* White Paper. March 2010. Accessed: 15. May 2010.

Available: http://www.brillianttelecom.com/lib/solutions/white_papers/bti_sync_deplymt_4_ip-ran_ops_wp_100308.pdf.

[25] RAD Data Communications. *Timing over Packet*. White Paper. January 2008. Accessed: 1. March 2010. Available: http://www.rad.com/static-files/Static%20Files/ MediaItems/5360_timing.pdf.

[26] Pietiläinen A. *Defining ToP SLA for Packet Networks.* NSN internal document. June 2009.

[27] IETF. *An Expedited Forwarding PHB (Per-Hop Behavior).* RFC 3246. 2002.

[28] Cosart, L. *Timing Measurements in Packet Networks*. ITSF presentation slides. November 2007. Available: www.symmetricom.com.

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

[30] Bregni, S. *Measurement of Maximum Time Interval Error for Telecommunications Clock Stability Characterization.*. Article: IEEE Transactions on Instrumentation And Measurement, Vol. 45. October 1996.

[31] Symmetricom. *Calculating minTDEV for Packet Timing Data Measurements.* ITU-T contribution (C 360). Geneva. June 2007.

[32] Nokia Siemens Networks. *Packet MTIE* . ITU-T Standardization working document (WD21). Miami. April 2008

[33] Symmetricom & Nokia Siemens Networks. *xTDEV/MAFE Definitions/Usage for G.8260 Appendix.* ITU-T Standardization working document (WD24). San Jose. March 2010.

[34] Nokia Siemens Networks. *Packet selection methods for performance metrics - a chain of switches.* ITU Standardization working document (WD35). Edinburgh. June 2009.

[35] Nokia Siemens Networks. *Comparison between synchronization metrics.* ITU-T contribution (C 302). Geneva. November 2008.

[36] Nokia Siemens Networks. *Maximum average time interval error.* ITU-T Standardization working document (WD60). Rome. September 2008.

[37] Digital Signal Processing. *Coefficients of recursive (IIR) digital filters.* Web page. Accessed: 10. February 2010. Available: http://www.dsptutor.freeuk.com/dfilt12.htm.

[38] Nokia Siemens Networks. *MAFE as a packet clock network limit metric.* ITU-T Standardization working document (WD77). Lannion. November 2009.

[39] France Telecom. *Continuation of the consideration on the performance aspects of a packet slave clock targeting 50ppb and based on a physical clock with a large time constant and PDV accumulation discussion.* ITU-T Standardization working document (WD37). Lannion. November 2009.

[40] MathWorks. *Documentation - Window functions.* Web page. Accessed: 1. July 2010. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/signal/tukeywin.html.

[41] ITU-T. *Timing characteristics of SDH equipment slave clocks (SEC).* 2003. Recommendation G.813.

[42] ITU-T. *Timing requirements of slave clocks suitable for use as node clocks in synchronization networks.* 2004. Recommendation G.812.

[43] IEEE. *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.* 2008. IEEE1588-2008 (1588v2).

[44] ITU-T. *Architecture and requirements for packet based frequency delivery.* 2010. Draft of Recommendation G.8265.

[45] ITU-T. *IEEE1588 profile for telecom.* 2010. Draft of Recommendation G.8265.1.

[46] ITU-T. *Timing characteristics of synchronous Ethernet equipment clock (EEC).* 2007. Recommendation G.8262.

[47] ITU-T. *Distribution of timing trough packet networks.* 2008. Recommendation G.8264.