

Clock Synchronization in Telecommunications via PTP (IEEE 1588)

Dr. André Vallat, Dominik Schneuwly
Oscilloquartz SA, CH-2002 Neuchâtel, Switzerland
vallat@oscilloquartz.com, schneuwly@oscilloquartz.com

Abstract - The transfer of time and frequency over packet switched networks is a difficult task because packet networks introduce large and highly variable packet delays. This paper looks into performance aspects of a new time transfer protocol called Precise Time Protocol Version 2, also known under the name of the corresponding standard IEEE 1588 v2. The study makes use of discrete event simulations which track the forwarding and queuing processes of individual packets in a simple network. The simulation results show that it is indeed possible to transfer time and frequency over limited size packet networks with accuracy levels commonly required by telecommunication equipment and applications. The simulation results support that PTP is suitable for a range of packet network types such as Metro Area Networks, base-station backhaul networks and access aggregation networks.

1. INTRODUCTION

Packet switching is becoming the dominant switching technique in modern telecommunication networks. The transfer of time and frequency is an important network service, since it is needed for the correct functioning of telecommunication equipment and user applications alike. In packet networks information is carried by packets which are forwarded asynchronously through the network. The propagation delays of the individual packets are highly variable. These are difficult conditions for the design of accurate time and frequency transfer systems.

This work addresses the question of whether it is at all possible, and if yes, under what conditions, to transfer time and frequency over packet switched telecom networks with the accuracies commonly required by telecom equipment and applications. Typical requirements include time transfer with microsecond accuracy (e.g. for the synchronization of mobile network's radio base-station, see [12]), and frequency transfer aiming at short-term stabilities expressed as TDEV of a few tens of nanoseconds at for example 50 seconds of observation interval (e.g. requirements for standard synchronization interfaces as per [13], section 6.4.2: TDEV ($\tau = 50$ s) = 34 ns). There is also the question whether such performance levels can be

achieved in all possible network implementations, or whether there are some constraints regarding network structure, network size, traffic load, etc. In the same vein there is the important question whether the implementation of new time and frequency transfer systems only concerns the end-points, or whether the switches and routers of the network also need to be upgraded or replaced.

The work presented herein is based on a new time transfer protocol called Precise Time Protocol (PTP). The second version of PTP is in the process of being standardized by the Institute of Electrical and Electronics Engineers (IEEE) under the designation IEEE 1588 v2. PTP v2 is presently the most advanced of such protocols.

2. TIME TRANSFER TECHNIQUES FOR PACKET NETWORKS

2.1. Definitions

The packet delay $\delta_{AB}(i)$ of the i -th packet of a given packet stream is the time interval between the instant when the start of the packet is leaving end-system A and the instant when it is entering end-system B.

As will become clear later on, we also need to define the packet delay asymmetry $A(j)$ of a pair of packets which are part of a simple bi-directional packet exchange: a 'request packet' is being sent from A to B, and a 'response packet' is being sent from B to A in response to and shortly after reception of the request packet:

$$A(j) = \delta_{AB}(j) - \delta_{BA}(j),$$

where $\delta_{AB}(j)$ is the delay of the request packet, and $\delta_{BA}(j)$ is the delay of the response packet.

2.2. Precise Time Protocol (IEEE 1588)

A variety of time transfer protocols adapted to the packet network context has been developed and standardized by the industry. The discussion in this section is based on the Precise Time Protocol (PTP) which was standardized by the Institute of Electrical and Electronics Engineers (IEEE) under the

designation IEEE 1588. The discussion refers to version 2 of the standard (see [7] and [8]).

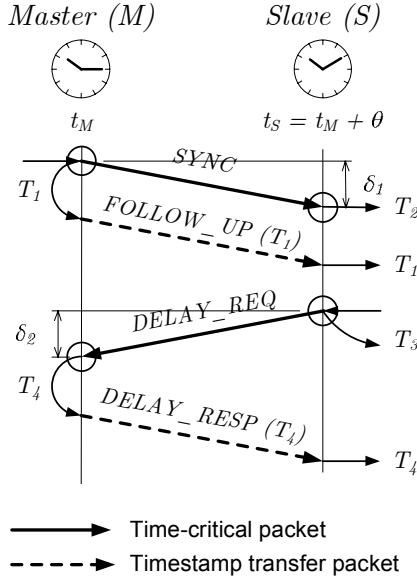


Figure 1: Two-Way Time Transfer in PTP

2.3. Two-Way Time Transfer

In order to compensate packet delays, PTP uses the well known Two-Way Time Transfer (TWTT) technique. Figure 1 shows the TWTT packets which are exchanged between the PTP master clock and the PTP slave clock in order for the slave to determine and correct its time offset relative to the master. The diagram shows the two time-critical packets typical of all TWTT-based protocols. In PTP they are called *SYNC* and *DELAY_REQ*. The two additional PTP packets *FOLLOW_UP* and *DELAY_RESP* will be explained later. The slave exploits the measured transmit and receive times of the two time-critical packets in order to calculate the clock offset $\bar{\theta}$ (θ represents the real offset, $\bar{\theta}$ represents the estimate). The calculation uses the well known formula below, which is based on the assumption that the delays in both directions δ_1 and δ_2 are equal.

$$\bar{\theta} = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

Because of the presence of a packet delay asymmetry the calculated offset is affected by the following error (see [7]):

$$\theta_{ERROR} = \theta - \bar{\theta} = \frac{\delta_1 - \delta_2}{2} = \frac{A}{2}$$

In a packet network the queuing part of the total packet delay is highly asymmetrical, except when the network runs at very low traffic loads.

2.4 Mitigating Packet Delay Asymmetry

How does PTP mitigate the detrimental effects of packet delay asymmetry? The basic idea is to somehow remove or compensate those delay components which are likely to be asymmetrical. The delays generated by the links are approximately symmetrical. The node delays, however, are often asymmetrical because of the queues and because of asymmetrical traffic loads. PTP uses a combination of techniques in order to remove or compensate these delay components.

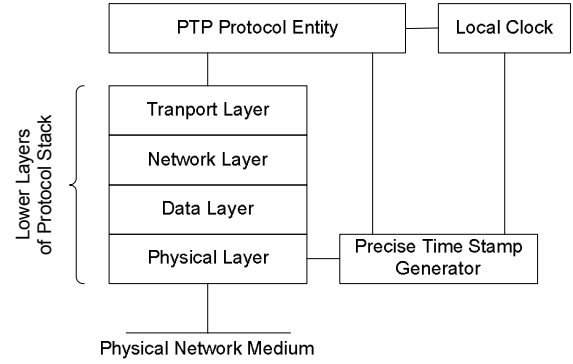


Figure 2: PTP Hardware assistance

The first technique consists in measuring the four time-stamps of a TWTT interrogation (T_1 , T_2 , T_3 and T_4) as close as possible to the physical port, as shown in Figure 2. The figure shows the protocol stack associated with one of the input or output ports of a node or an end-system. The time-stamps are measured by the 'Precise Time Stamp Generator' when the packet traverses the Physical Layer of the protocol stack. Any delay generated while the packet is moving up or down the protocol stack, including queuing, is singled out, since the TWTT exchange takes place virtually between the physical layers of the master and the slave.

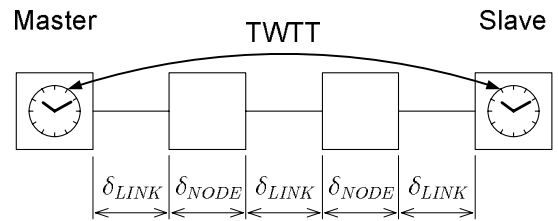


Figure 3: PTP Transparent Clocks

The second technique is illustrated in Figure 3. The figures show a PTP path with two end-systems and two network nodes. Here the PTP-related working mode of the network nodes is called 'End-to-end Transparent

Clock' mode (there exist other operation modes in PTP which are not discussed here). The PTP protocol session is between the two end-systems. The two time-critical packets *SYNC* and *DELA_REQ* experience both the link delays and the node delays. The node delays are measured by the network nodes and the measured values ($\bar{\delta}_{NODE,1}$ for *SYNC*, and $\bar{\delta}_{NODE,2}$ for *DELA_REQ*) are transferred to the slave via the additional packets *FOLLOW_UP* and *DELA_RESP*. These measured node delays are called residence times in PTP parlance. With this information the slave calculates an estimate of the asymmetry and uses it for the calculation of the slave clock's offset:

$$\bar{A} = \sum \bar{\delta}_{NODE,1} - \sum \bar{\delta}_{NODE,2}$$

$$\bar{\theta} = \frac{(T_4 - T_3) - (T_2 - T_1)}{2} - \frac{\bar{A}}{2}$$

The remaining error is now half the difference between the actual and the estimated asymmetry:

$$\theta_{ERROR} = \frac{A}{2} - \frac{\bar{A}}{2}$$

The formula above gives the error for a single TWTT interrogation. PTP slaves use sophisticated processing and filtering techniques applied to consecutive TWTT interrogations in order to further reduce the residual clock offset.

3. SIMULATION TOOL

3.1. Objective

As was mentioned earlier, the accuracy of packet-based time transfer techniques depends primarily on the packet delay asymmetry present in the network. The work presented herein aims at evaluating the potential accuracy of these techniques by studying the statistical properties of delay asymmetry via simulations.

3.2. Overall structure of the simulator

The simulation tool is a program written in C. It implements a discrete event simulation based on a sampled time-scale with sampling period T_S . The simulated events are packets being transmitted, received, stored and switched through the nodes and links of the network. Figure 4 shows the block diagram of the simulated network. It consists of two end-systems, i.e. the PTP master and the PTP slave, of six network nodes (switches or routers), 12 traffic sources & sinks, and a number of links. The horizontal links have a data rate of 1 Gbit/s. The vertical lines represent aggregations of 12 links with 1 Gbit/s data rate each. Because of the negligible contribution of link delay to

the total delay in real networks, the simulated links have zero delay.

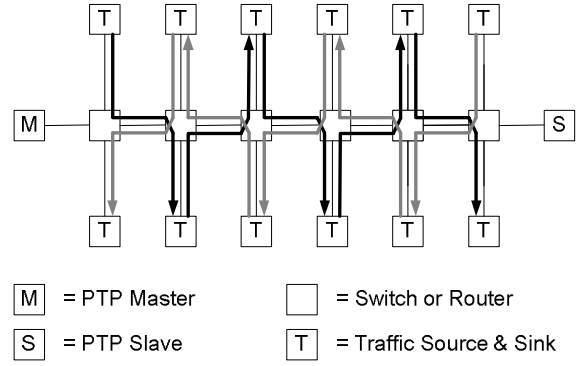


Figure 4: Topology of the simulated network

3.3. Network Nodes

The internal structure of a switch or router is shown in Figure 5. It consists simply of a common input queue, a switching matrix and an output queue for each output port. All queues have a FIFO (first in, first out) queuing discipline. The service time of the switching matrix is constant and equal to T_S . This mirrors the fact that modern switches and routers have fast switching times; they do not contribute substantially to the node delays, queuing delays being much larger contributions. There are two node types. The first type is PTP-capable because it implements physical layer time-stamping and Transparent Clock function (see section 3.3 above). The second type is an ordinary switch or router with no PTP capability. The non-PTP-capable (ordinary) node simply forwards PTP packets in the same way it forwards other packet traffic. The PTP-capable nodes measure the delays PTP packets experience while traversing the nodes (residence times), and send the results to the PTP slave.

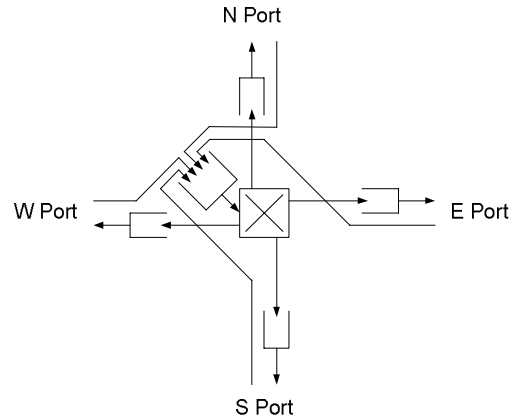


Figure 5: Internal structure of a switch or router

3.4. PTP End-systems

During a simulation run the two PTP end-systems exchange *SYNC* and *DELAY_REQ* packets as shown in Figure 1. These packet pairs are generated periodically at a constant rate R_I called interrogation rate. In order to keep complexity within limits, the two additional packets *FOLLOW_UP* and *DELAY_RESP* are not actually sent through the simulated network. Instead, node delays measured by the PTP-capable nodes are provided directly to the PTP slave. The PTP slave calculates the asymmetry estimate \bar{A} as described in section 2.3.

3.5. Traffic Sources and Traffic Flows

Each traffic source in Figure 4 represents 12 traffic links with 1 Gbit/s data rate each. According to [9] and [10], Ethernet and Internet traffic is bursty, and packet inter-arrival times contain both long-range dependent and time-independent components. The traffic generation algorithm described below is a simple way of generating traffic flows with such time-dependencies. The algorithm is described below, with a representing a parameter related to traffic load:

At any time instant $k \cdot T_s$ there are two possibilities:

- 1) With a probability of p , one of the 12 traffic ports starts sending a new packet
- 2) With a probability of $(1 - p)$, none of the ports starts sending a new packet.

The probability p is time dependent. For each time instant $k \cdot T_s$, $p(k)$ is calculated anew starting with the draw of a pseudo-random number $r(k)$ with uniform distribution within the interval $[-a/2, +a/2]$. The calculation continues as follows:

If $r(k) > 0$ then $\Delta p(k) = r(k)[1 - p(k-1)]$

Else $\Delta p(k) = r(k)$;

$p(k) = |p(k-1) + \Delta p(k)|$.

The black and grey bold arrows in Figure 4 show how the traffic flows generated by the traffic sources traverse the network. The arrangement is such that there is non-PTP traffic in both directions superposed with the PTP traffic on each node-to-node hop between the two end-systems.

3.6. Simulator Output

The goal of the simulations is to study the slave clock's remaining error after the clock has been adjusted to the master's time via PTP. As mentioned in section 2.3, this error depends on the residual delay asymmetry after correction by the estimated asymmetry:

$$\theta_{ERROR} = \frac{A}{2} - \frac{\bar{A}}{2}$$

In view of this, the simulator's main output is the residual asymmetry

$$A_R(j) = A(j) - \bar{A}(j),$$

where $j = 1, 2, 3, \dots =$ interrogation sequence number

The simulator also provides the packet delays of all *SYNC* packets. The indication of packet delay gives useful information about the traffic conditions in the network. The simulator provides its output data in the following formats:

- Raw data
- Histogram
- Standard Deviation
- Time Deviation (TDEV)

Refer to [11] for a discussion of TDEV.

It is important to note, that the residual packet delay asymmetry $A_R(j)$ influences the final accuracy of the time transfer only indirectly, since the PTP slave applies processing and filtering techniques to further reduce the errors (see section 2.3). However, $A_R(j)$ is the raw input data for these post-processing techniques. Knowledge of the properties of $A_R(j)$ can help optimizing said techniques.

4. SIMULATIONS

4.1 Simulation Scenarios

Six simulation runs were executed. Three of them were done with a traffic source parameter $a = 0.02$ (see section 3.4), and another three with $a = 0.06$. The value $a = 0.02$ correspond to a nominally loaded network (a traffic load for which the network was designed), whereas $a = 0.06$ corresponds to a network near total congestion. For each of the two a -values, three simulations were executed with different numbers of PTP-capable and non-PTP-capable (ordinary) network nodes on the path between the two PTP end-systems. The number of non-PTP-capable nodes N_N in the three simulations is 2, 4 and 6. The case $N_N = 0$ (i.e. all network nodes are PTP-capable) is not simulated here. Since the tool simulates mainly queuing delays and since these are exactly compensated by the Transparent Clock mechanism, the results for $N_N = 0$ are trivial. In reality the time transfer accuracy for $N_N = 0$ is determined by second order factors other than queuing. Table 1 summarizes the six simulation scenarios.

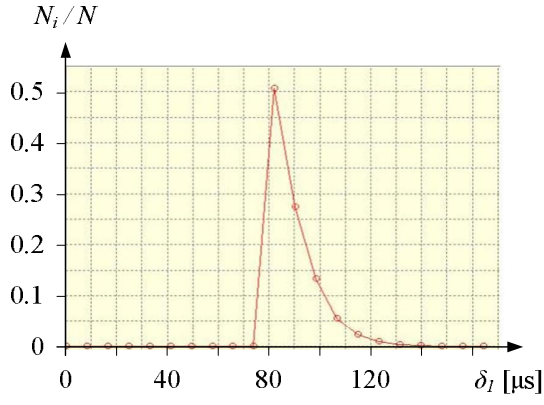


Figure 6: Histogram of packet delay δ_I , $a = 0.02$

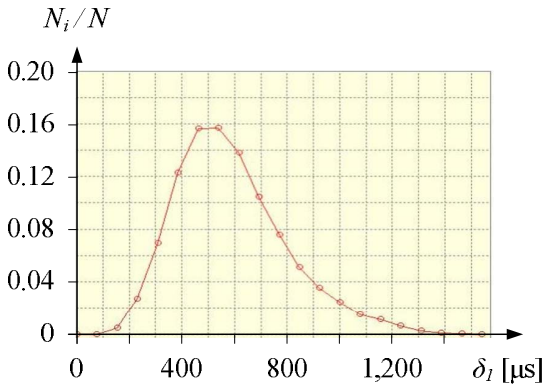


Figure 7: Histogram of packet delay δ_I , $a = 0.06$

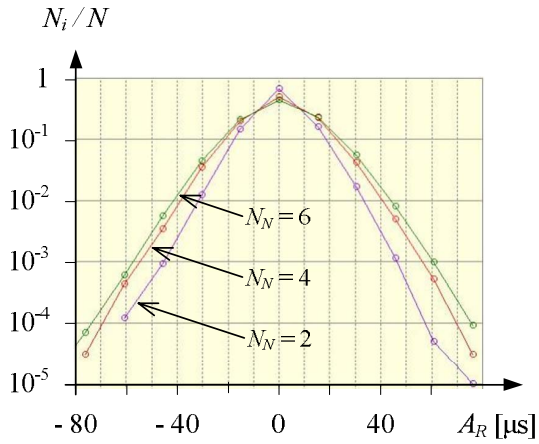


Figure 8: Histogram of residual asymmetry, $a = 0.02$

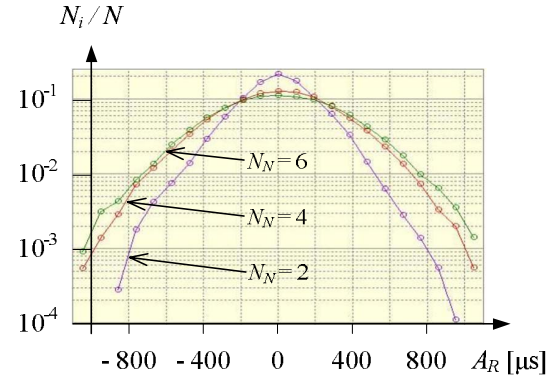


Figure 9: Histogram of residual asymmetry, $a = 0.06$

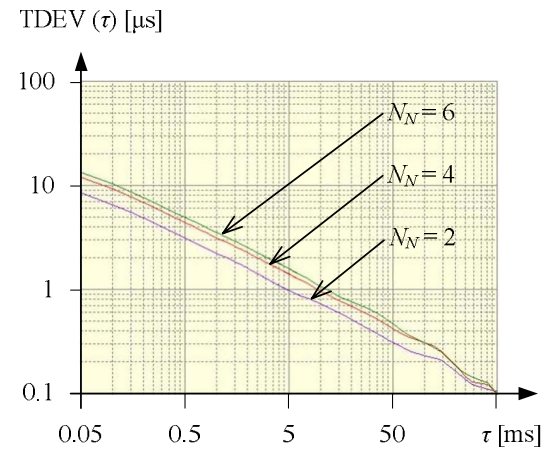


Figure 10: TDEV of residual asymmetry, $a = 0.02$

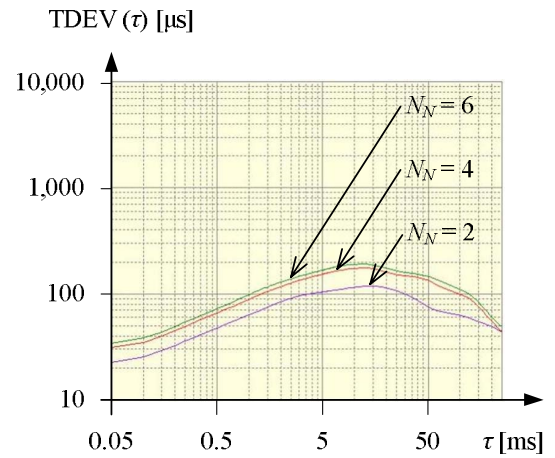


Figure 11: TDEV of residual asymmetry, $a = 0.06$

Table 1: Simulation scenarios

Run no.	X = non-PTP-capable node O = PTP-capable node	N_N	a
1	- O - X - X - O - O - O -	2	0.02
2	- O - X - X - X - X - O -	4	
3	- X - X - X - X - X - X -	6	
4	- O - X - X - O - O - O -	2	0.06
5	- O - X - X - X - X - O -	4	
6	- X - X - X - X - X - X -	6	

The other simulation parameters were fixed as follows:

- Simulation sampling period $T_S = 1 \mu\text{s}$
- Simulation length = 5,000,000 time samples
- Packet length = 1,500 bytes
- Size of input queues = 150,000 bytes
- Size of output queues = 150,000 bytes
- PTP interrogation rate $R_I = 20,000$ per s

The chosen interrogation rate is higher than what is usually the case in PTP. The impact on the results is negligible, but there is the advantage that exploitable results can be obtained with much shorter simulation runs.

4.2 Simulation Results and Discussion

Figures 6 and 7 show histograms of the delays experienced by PTP packets flowing from the master to the slave (the *SYNC* packets in Figure 1). These delays do not enter as such into the expression of the slave clock's residual time error θ_{ERROR} (see section 2.2). However, these histograms draw a picture of the network state in terms of traffic level and queue occupancies. One can compare the two histograms with known network delay patterns such as those published in [1], [2], [3], [4], [5] and [6]. The histogram of Figure 6 corresponds to a medium to high traffic load. The histogram of Figure 7 has its peak at 500 μs , whereas the peak in Figure 6 is at 85 μs . The histogram of Figure 7 is typical for a network which is close to the condition of network congestion. This allows one to say that the case $a = 0.02$ corresponds to a nominally loaded network (a traffic load for which the network was designed), whereas the case $a = 0.06$ simulates a network which is close to congestion.

Figures 8 to 11 show the histograms and TDEV functions of the residual asymmetry A_R for all six simulation runs. Table 2 summarizes the standard deviations.

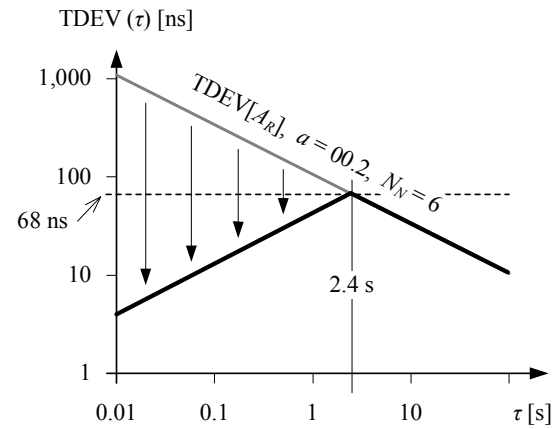
Table 2: Standard Deviations of A_R

Run no.	a	N_N	Standard Deviation [μs]
1	0.02	2	9
2		4	13
3		6	14
4	0.06	2	210
5		4	310
6		6	340

The histograms and the standard deviations reveal some interesting facts:

- The change from nominal traffic load towards congestion has a high impact on the results; the standard deviation increases by a factor of more than 20.
- Whether only 2, or 4, or all network nodes are non-PTP-capable does not have a strong impact on the results; the difference between $N_N = 2$ and $N_N = 4$ is small, and the difference between $N_N = 4$ and $N_N = 6$ is negligible.

These two findings are true for all cases where some or all of the intervening network nodes (switches or routers) are non-PTP-capable (i.e. physical layer time-stamping and Transparent Clock function not implemented). Of course, the relationships might be different in networks which are equipped exclusively with PTP-capable nodes.

**Figure 12:** Low-pass filtering

TDEV representations are useful in two ways. First, some of the synchronization quality specifications for telecommunication networks are specified as TDEV limits; hence the TDEV results obtained herein can be compared directly with these requirement specifications. Secondly, TDEV allows one to predict the effect of applying linear filtering to the data, as in

the case of using a Phase Locked Loop (PLL) as low-pass filter. A good starting point for the discussion is the TDEV limit specified for standard synchronization interfaces in ITU-T Recommendation G.823, section 6.2.4 (see [13]). For observation intervals between 0.1 s and 50 s the TDEV limit is 34 ns. The question is, whether low-pass filtering can attenuate the jitter and wander resulting from A_R and evidenced in the TDEV curves to the requested level of 34 ns. While highly problematic for the case $a = 0.06$, this is indeed possible for the case $a = 0.02$. The TDEV curves in Figure 10 are proportional to $\tau^{-1/2}$ (White Phase Modulation noise). A first-order low-pass filter with a cut-off frequency f_c will lower the TDEV curve in the region $\tau < 0.3/f_c$, and will leave the TDEV untouched in the region $\tau > 0.3/f_c$ (according to [14], Appendix I, the relationship between observation interval τ and Fourier frequency f is: $\tau = 0.3/f$). In the region below $0.3/f_c$ a first-order low-pass filter will change the TDEV curve so as to make it proportional to $\tau^{1/2}$. The attenuation must be such that the TDEV of θ_{ERROR} falls below 34 ns, or the TDEV of A_R below $2 \cdot 34 \text{ ns} = 68 \text{ ns}$. The TDEV of A_R for the case $N_N = 0.06$ is approximately

$$\text{TDEV}(\tau) \cong 10^{-7} \cdot s^{3/2} \cdot \tau^{-1/2}$$

It is easy to see that

$$\text{TDEV}(\tau = 2.4 \text{ s}) \cong 68 \text{ ns}.$$

So the cut-off frequency must be chosen to satisfy

$$f_c < \frac{0.3}{2.4 \text{ s}} = 125 \text{ mHz}.$$

In other words: in order to obtain slave clock synchronization compliant with the TDEV specification mentioned above, the raw data provided by the PTP time transfer process must be low-pass filtered (e.g. PLL) with a cut-off frequency of $f_c \leq 125 \text{ mHz}$. Experience shows that it is perfectly possible to realize PLLs with 125 mHz bandwidth using good quartz crystal oscillators.

5. CONCLUSIONS

The results of these first simulation runs lead to a number of interesting conclusions; they are valid for packet networks with limited size, since the simulated network path had a length of only six nodes:

- With PTP IEEE 1588 it is possible to transfer time and frequency with the accuracy levels commonly required by telecommunication applications and equipment, under the condition that traffic load does not approach congestion. This is achievable with PTP slave clocks implementing some adequate sample selection algorithm and jitter low-pass filtering with a maximum bandwidth of 125 mHz.
- In networks of limited size and controlled traffic load, the required performance level can be achieved with only the end-systems, i.e. the PTP master and slave clocks, implementing full PTP functionality. It is not necessary that the switches or routers provide full PTP support (i.e. physical layer time-stamping, Transparent Clock function). Of course, the full potential of PTP can only be exploited if all intervening switches or routers do provide the full PTP functionality.
- In cases where some or all of the switches or routers do not provide full PTP support, the traffic load has a significant impact on the accuracy of PTP time transfer. The performance degrades strongly when the network approaches congestion.

Many of today's network types do have the size and congestion control characteristics which make them suitable targets for PTP synchronization. The simulation results support that PTP is suitable for network types such as Metro Area Networks, base-station backhaul networks, access aggregation networks, and the like.

REFERENCES

- [1] G. Iacovoni, I. Bartoli; "Emulation of IP impairments on real-time services"; *Third International Conference on Performance Modeling and Evaluation of Heterogeneous Networks, (HET-NET 2005)*; Ilkley, West Yorkshire, UK; 2005.
- [2] S. Moon, J. Kurose, P. Skelly, D. Townsley; "Correlation of Packet and Loss in the Internet"; *Technical Report UM-CS-1998-11*; University of Massachusetts, Amherst, MA; 1998.
- [3] D. Loguinov, R. Hayder; "Correlation of Packet and Loss in the Internet"; *ACM Computer Communication Review*, vol. 21, no. 1; 2002.
- [4] A. Mukherjee; "On the Dynamics and Significance of Low-Frequency Components of Internet Load"; *Interworking: Research and Experience*, vol. 5, no. 4; 1994.
- [5] Q. Li, D. Mills; "On the Long-range Dependence of Packet Round-trip Delay in Internet"; *Proceedings of IEEE ICC'98*, vol. 2, pp. 1185-1192; Atlanta, GA; 1998.

- [6] Q. Li, D. Mills; “Investigating the Scaling Behavior, Crossover and Anti-persistence of Internet Packet Delay Dynamics”; *Proceedings IEEE GLOBECOM Symposium 1999*, pp. 1843-1852; 1999.
- [7] J. Edison; “Measurement, Control and Communication Using IEEE 1588”; Springer, London; 2006.
- [8] J. Eidson; “IEEE-1588 Standard Version 2, A Tutorial”; *2006 Conference on IEEE 1588*; Gaithersburg, MA; October 2 to 4, 2006.
- [8] W. Leland, M. Taqqu, W. Willinger, D. Wilson; “On the Self-Similar Nature of Ethernet Traffic”; *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1-15; 1994.
- [10] J. Cao, W. Cleveland, D. Lin, D. Sun; “On the Nonstationarity of Internet Traffic”; *Proceedings ACM SIGMETRICS 2001*, pp. 102-112; 2001.
- [11] M. Carbonelli, D. De Seta, D. Perucchini; “Characterization of Timing Signals and Clocks”; *European Transactions on Telecommunications*, vol. 7, no. 1; January/February 1996; pp. 9-24.
- [12] 3rd Generation Partnership Project; “Technical Specification TS 25.105 V6.0.0: Base Station (BS) radio transmission and reception (TDD)”; Valbonne, France; December 2003.
- [13] International Telecommunication Union; “ITU-T Recommendation G.823: The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy”; Geneva; March 2000.
- [14] International Telecommunication Union; “ITU-T Recommendation G.812: Timing requirements of slave clocks suitable for use as node clocks in synchronization networks”; Geneva; June 2004.