Precision Packet-based Frequency Transfer based on Oversampling

Giada Giorgi, Claudio Narduzzi
Department of Information Engineering
University of Padova, Padova, Italy
Email: [giada, narduzzi]@dei.unipd.it

***Abstract*—Frequency synchronization of a distributed measurement system requires the transfer of an accurate frequency reference to all nodes. The use of a general-purpose, packet-based network for this aim is analyzed in the paper, where oversampling is considered as a means to counter the effects of packet delay variation on time accuracy. A comprehensive analysis, that includes the stability of the local clock, is presented and shows that frequency transfer through a packet network of this kind is feasible, with an accuracy level that can be of interest to a number of distributed measurement applications.**

***Index Terms*—Frequency transfer, synchronization, measurement uncertainty.**

I. Introduction

In distributed measurement systems a common time scale is needed to allow the exchange of time-referenced values among nodes. Accurate alignment is assumed, for instance, in sensor fusing for time-dependent applications [1] and in synchrophasor-based, smart grid wide-area measurement systems [2]. Synchronization is also a very important issue in a variety of other application areas, such as telecommunication networks [3], [4] and industrial automation.

Dissemination of a time reference to distributed measurement nodes is a challenging task. A Global Navigation Satellite System (GNSS), such as the Global Positioning System (GPS) [5] is widely employed for this purpose, for instance, to align commercial phasor measurement units (PMUs) to within less than 1 μs. Accurate GNSS-based synchronization requires good reception of signals from a number of satellites, which can be hard to achieve in certain conditions. Short-term variability, of the order of a few tens of ns in GPS [6], may need consideration. Very precise synchronization can also be obtained through packet-based transmission systems. The Precision Time Protocol (PTP) [7] was developed for this purpose and requires dedicated PTP-compliant network equipment (e.g., boundary clocks, transparent clocks), to accurately measure and/or compensate for packet network delays.

Synchronization involves both frequency transfer and time distribution, that can be seen as two coordinated but distinct functions in packet networks [3], [8]. In some instances (e.g., the *White Rabbit* project [9]), they are implemented through different means. Accurate frequency transfer can be realized at the physical level by means of Synchronous Ethernet [10], provided node clocks throughout the network satisfy suitably tight conditions on frequency accuracy [11]. In this paper we focus on the problem of frequency transfer and the achievement of frequency synchronization by the exchange of packets with embedded timestamps. This has close similarities with time recovery techniques adopted in circuit emulation services (CES) over packet-switched networks [4] where, however, timing packets are continuously issued at a constant rate.

Without specific physical-level timing support, packet delay variation (PDV) makes the propagation of accurate time information more difficult. However, most networks are currently engineered to provide Quality of Service (QoS) support, that can be exploited to transport time packets as *highest priority* items, taking precedence over best-effort traffic. This enables adoption of the *oversampling* technique, whereby packets are broadcast at high rate by the master node providing the time reference. Slave nodes can thus collect a large set of measurements over an interval that can still be considered short, compared to clock stability and network routing dynamics. The packet flow employed in oversampling should be regarded as a measurement probe with respect to the PDV behavior of an end-to-end network connection. This snapshot of statistical variability can be processed to counter PDV effects and ultimately obtain the required accurate time information, reducing the need for dedicated equipment. The approach can be attractive for broadening the reach of wide-area monitoring systems, where more specialized solutions would be impracticable in terms of cost-effectiveness.

In this paper we analyze the performance of the oversampling approach in precision packet-based frequency transfer. We discuss the effects of PDV on the estimation of frequency offset, evidencing critical aspects and design trade-offs that need to be considered to achieve a target accuracy. For this purpose, we propose a novel packet selection algorithm that provides robustness to the effects of traffic load variation and introduce an analytical model of PDV, that allows tuning for good frequency transfer accuracy. The aims of the study are:

1) determine best frequency offset estimates, using packet latency measurements in conditions of PDV-induced variability;
2) evaluate the uncertainty associated to those estimates;
3) discuss the accuracy levels achievable by syntonization through a general-purpose packet network.

II. Time Protocol Measurement

A time protocol is based on the exchange of packets between a master node containing the reference clock, and slave nodes whose clocks need to be synchronized to the master. The master node generates timestamps that are broadcast within suitable packets. Timestamp value $t_1(n)$ associated to the n-th packet is sent at time t_n and, neglecting master clock uncertainty, $t_1(n) = t_n$. When the packet is received by a slave node, a local timestamp $t_2(n)$ is generated.

A. Latency

The *latency* of the n-th packet is the time difference $l_S(n) = t_2(n) - t_1(n)$ measured by the slave and provides an estimate of the slave clock time offset $x(n)$ that, however, also includes the master-to-slave network delay $d_{MS}(n)$ experienced by the packet and slave timestamping uncertainty $w_S(n)$:

$$l_S(n) = x(n) + d_{MS}(n) + w_S(n) \tag{1}$$

Master-to-slave network delay can be decomposed as: $d_{MS}(n) = D_{MS} + q_{MS}(n)$. The first term is the sum of constant propagation and processing delays along a given network path. It may vary only when packet routing is changed, usually producing a step change in $d_{MS}(n)$, but the rate of occurrence of these events is typically low.

The term $q_{MS}(n)$ represents PDV, that is, packet delay variability associated to packet queuing phenomena. It is determined by variable traffic conditions through the network, depending on traffic load, the kind of traffic (cross traffic or in-line traffic), the number of switches along the path, etc.. Accordingly, we consider $q_{MS}(n)$ as a random process. PDV is the main obstacle to the use of a general-purpose packet network for frequency transfer, since it prevents accurate measurement of the slave clock fractional frequency offset $y(n)$. In fact, given two consecutive couples of timestamp values, the estimate is provided by the ratio:

$$\hat{y}(n) = \frac{l_S(n) - l_S(n-1)}{t_1(n) - t_1(n-1)} \tag{2}$$

whose uncertainty mainly depends on the first-difference of the random process $q_{MS}(n)$. As the contribution of $w_S(n)$ is practically negligible in comparison, we shall dispense with it in the following.

B. Oversampling

Time offset of a slave clock must be regarded as a continuous-time random process $x(t)$. The rationale for oversampling is the assumption that $x(t)$ varies very slowly in comparison to network delay, allowing the slave to collect a large set of timestamps in a time window T_W short enough that the mean value:

$$\bar{x}(t^*) = \frac{1}{T_W} \int_{t^*}^{t^*+T_W} x(t)dt \tag{3}$$

can be associated to *any* time packet in the window as the actual time offset.

Let W be the number of packets received within the i-th window. To simplify notation, we introduce index $j = n - [(i-1)W]$ and set: $l_j(i) = l_S(n)$ and $q_j(i) = q_{MS}(n)$ for $(i-1)W < n \leq iW$, assuming no dead time occurs between consecutive windows. Within the i-th window, $x(n) \equiv \bar{x}(i)$, therefore latency values $l_j(i)$ can be considered measurements of $\bar{x}(i)$, to which a random disturbance is superposed:

$$l_j(i) = \bar{x}(i) + D_{MS} + q_j(i) \tag{4}$$

The set of PDV values $q_j(i)$, with $J = 1, \ldots, W$ is an independent, identically distributed (i.i.d.) random vector $\underline{\mathbf{q}}(i)$. Consequently, the set of latencies $l_j(i)$ also forms a suitably shifted i.i.d. random vector, $\underline{\mathbf{l}}(i)$.

Experimental observations of background network traffic show that patterns remain constant over several hundred seconds [12]. Given the time intervals considered in this work, it is acceptable to assume that D_{MS} and random process parameters are stationary within a window.

C. Packet selection

PDV effects on measurement uncertainty can be minimized by *packet selection*, that is, by processing the set of latency values measured within a given time window, to provide a better estimate of the actual master-to-slave time offset.

The packet selector output referred to the i-th time window, $L(i) = g\left(\underline{\mathbf{l}}(i)\right)$, can be either the latency of a specific single packet, or a statistical parameter (e.g., average, median) related to a selected subset of packets, or to the whole set. Packet selectors discussed in the following Sections can be shown to be all invariant to a time shift of the pdf, therefore:

$$L(i) = g\left(\underline{\mathbf{l}}(i)\right) = \bar{x}(i) + D_{MS} + g\left(\underline{\mathbf{q}}(i)\right). \tag{5}$$

A suitable time estimate $T(i)$ has to be associated with $L(i)$, so that a more accurate estimate of fractional frequency offset $\hat{y}(i)$ can be obtained by replacing l_S and t_1 in (2), respectively, with L and T:

$$\hat{y}(i) = \frac{L(i) - L(i-1)}{T(i) - T(i-1)}. \tag{6}$$

Depending on the packet selector, the difference $T(i) - T(i-1)$ might not be constant, unlike in (2). However, its mean value is equal to T_W and, for the purpose of this analysis, it can be assumed that $T(i) - T(i-1) \cong T_W$. Substitution of (5) into (6) yields:

$$\hat{y}(i) = y_{T_W}(i) + \frac{g\left(\underline{\mathbf{q}}(i)\right) - g\left(\underline{\mathbf{q}}(i-1)\right)}{T_W} \tag{7}$$

where $y_{T_W}(i) = [\bar{x}(i) - \bar{x}(i-1)]/T_W$ is the slave clock average fractional frequency offset in the time interval T_W, whereas the second term depends on PDV. The mean value of $\hat{y}(i)$ is:

$$\mathrm{E}[\hat{y}(i)] \cong \mathrm{E}[y_{T_W}(i)] + \frac{\mathrm{E}\left[g(\underline{\mathbf{q}}(i))\right] - \mathrm{E}\left[g(\underline{\mathbf{q}}(i-1))\right]}{T_W}. \tag{8}$$

and the estimate is unbiased when the second term in (8) vanishes.

III. Evaluation of Frequency Uncertainty

To evaluate the uncertainty of the refined estimate (6), we first note that condition $\mathrm{E}\left[g(\underline{\mathbf{q}}(i))\right] = \mathrm{E}\left[g(\underline{\mathbf{q}}(i-1))\right]$ is satisfied over time spans where network traffic patterns do not change significantly. Uncertainty can then be associated to the variance $\mathrm{var}[\hat{y}(i)]$. From the stationarity and i.i.d. assumptions made for $q_{MS}(n)$ it follows:

$$\mathrm{var}[\hat{y}(i)] = \mathrm{var}[y(i)]_{T_W} + 2\frac{\mathrm{var}\left[g(\underline{\mathbf{q}}(i))\right]}{T_W^2}, \tag{9}$$

where the last term depends on the variance of the packet selector output. For simplicity, in the following we use the symbol: $\mathrm{var}[L(i)] = \mathrm{var}\left[g(\underline{\mathbf{q}}(i))\right]$.

A. Slave clock stability

Since $y(t)$ is a non-stationary process, its variance depends on time. With oversampling, an updated estimate of $\hat{y}(i)$ can only be produced after a whole time window has elapsed, that is every T_W seconds. It follows:

$$\mathrm{var}[\hat{y}(i)] = \frac{\sigma_1^2}{T_W} + \frac{\sigma_2^2}{3}T_W + 2\,\frac{\mathrm{var}\left[L(i)\right]}{T_W^2} \tag{10}$$

where σ_1^2 and σ_2^2 are respectively the variances of the mutually uncorrelated, zero-mean diffusion components corresponding to white phase modulation (WPM) noise and white frequency modulation (WFM) noise, as introduced in [13].

In the ideal case $\mathrm{var}\left[L(i)\right] = 0$ (that is, no PDV) estimation variance is minimised when $T_{W(opt)} = \sqrt{3}(\sigma_1/\sigma_2)$. With the parameter values usually associated to a suitably stable clock, however, the resulting window length would be too long and, remembering (3), in the following we shall assume for T_W an order of magnitude not greater than 10^2 s. The contribution to $\mathrm{var}[\hat{y}(i)]$ of the term proportional to σ_2^2 thus becomes negligible in most cases.

Frequency uncertainty also affects the stability of disciplined slave clocks. When frequency is estimated by (6) the resulting time offset variance is given by:

$$\mathrm{var}[x(t)]_\tau \cong \frac{\sigma_2^2}{3}\tau^3 + 2\,\mathrm{var}\left[L(i)\right]\left(\frac{\tau}{T_W}\right)^2 \tag{11}$$

where τ is the time elapsed from initial alignment to the reference clock and the approximation holds for $\tau \gg T_W$. Given a time synchronization bound $\pm\Delta$, the condition:

$$k\sqrt{\mathrm{var}[x(t)]_\tau} \leq \Delta \tag{12}$$

with a suitable value of the integer k, determines the length of time that a slave clock can meet the synchronization requirement without receiving any update.

B. Packet selection uncertainty

A variety of packet selection methods have been considered in the literature (e.g., [8], [14]–[16]). We briefly discuss three of them, assuming end-to-end PDV is described by a Gamma random variable $f_q(a;\alpha,\beta)$ whose shape and scale parameters α and β (both positive) depend on network traffic load. This model is widely employed in networks [12], [17], [18] and yields as the mean and variance of $q_{MS}(n)$, respectively: $\mathrm{E}[q_{MS}] = (\alpha/\beta)$ and $\mathrm{var}[q_{MS}] = (\alpha/\beta^2)$.

a) Minimum packet selection: the minimum latency value within an oversampling window corresponds to the index $j_{min} = \arg\left\{\min_j\left[l_j(i)\right]\right\}$. For $W \to \infty$ the latency estimate $L_{min}(i) = l_{j_{min}}(i)$ tends to $\bar{x}(i) + D_{MS}$. In practice a time window contains a finite number W of packets and variability is significant, unless a very large number W is considered. Minimum packet selection performs very well for lightly loaded networks (approximately up to 20% of maximum traffic capacity [16]) and could be regarded as a benchmark in these conditions.

b) Mean value packet selection: At high traffic loads, averaging the full set of latency values provides the greatest reduction in packet selector output variance [16]. The mean and variance for this selector, indicated by $L_\mu(i)$, are:

$$\mathrm{E}\left[L_\mu(i)\right] = \bar{x}(i) + D_{MS} + \frac{\alpha}{\beta} \qquad \mathrm{var}\left[L_\mu(i)\right] = \frac{1}{W}\frac{\alpha}{\beta^2} \tag{13}$$

c) Quantile packet selection: Packet selection based on a quantile threshold is suggested in [8]. Let L_δ be the latency value corresponding to quantile δ from a set of W packets. Typically $\delta = 1\%$ in ITU-T Recommendations. The set of indices: $J_\delta = \{j : l_j(i) \leq L_\delta\}$ comprises the δW fastest packets, whose master-to-slave delay component is described by a truncated and suitably scaled Gamma pdf $f_\delta(a;\alpha,\beta)$. The packet selector output for the i-th time window, $L_\delta(i)$, is obtained by averaging the latencies of the selected packets:

$$L_\delta(i) = \frac{1}{\delta W}\sum_{j \in J_\delta} l_j(i). \tag{14}$$

If W is large enough that the δ-quantile Q_δ can be considered independent of the actual set $\underline{\mathbf{l}}(i)$, the expected value of $L_\delta(i)$ is:

$$\mathrm{E}\left[L_\delta(i)\right] = \bar{x}(i) + D_{MS} + \frac{\alpha}{\beta}\left[1 - \frac{Q_\delta}{\alpha}f_\delta(Q_\delta;\alpha,\beta)\right] \tag{15}$$

confirming that the "fast packets" subset has a reduced mean PDV. However, elements in this subset also have a somewhat greater variance, since the truncated pdf is more skewed than the original Gamma pdf. For $j \in J_\delta$ and neglecting higher-order terms:

$$\mathrm{var}\left[q_j(i)\right] \cong \frac{\alpha}{\beta^2}\left[1 + \frac{Q_\delta}{\alpha}f_\delta(Q_\delta;\alpha,\beta)\left(\alpha + 1 + \beta Q_\delta\right)\right] \tag{16}$$

Averaging by (14) produces only a modest reduction, since $\delta W \ll W$.

Performance of $L_\delta(i)$ is good with moderate network traffic, but variance increases at higher loads [16]. To achieve optimal results the quantile value δ would have to be adjusted as traffic load varies, which is hardly practical.

IV. Boltzmann (soft-min) packet selector algorithm

Variances of packet selectors in Section III-B change significantly with traffic load, which limits their usefulness for frequency transfer through a general purpose network. In this Section we introduce a novel algorithm, that is designed to provide robustness to varying traffic load conditions while retaining a simple structure. It is based on the weighted mean:

$$L_B(i) = \sum_{j=1}^{W} w(l_j(i)) \cdot l_j(i) \tag{17}$$

where the weights:

$$w(l_j) = \frac{-e^{\theta l_j}}{\sum_{j=1}^{W} e^{-\theta l_j}} = \frac{e^{-\theta l_j}}{Z} \qquad \text{with:} \quad \theta \geq 0 \tag{18}$$

depend on a common parameter θ, whose choice is discussed later on. The denominator Z provides weight normalization.

If one considers $\{l_j(i) : j = 1, \dots, W\}$ as the set of possible latency "states" within the i-th window, the weights (17) can be interpreted as the probabilities of each state according to a Boltzmann distribution (hence the subscript B). In machine learning literature (17) is also known as a "soft-min" function (e.g., [19]).

To our knowledge, this approach has not been considered before for packet selection, therefore we briefly introduce its statistical properties. It can be verified that weights (18) are invariant to time shifts, therefore condition (5) holds also in this case. The expected value of $L_B(i)$ is:

$$\mathrm{E}\left[L_B(i)\right] = \bar{x}(i) + D_{MS} + \frac{\alpha}{\beta}\left(\frac{\beta}{\beta+\theta}\right) \tag{19}$$

and the corresponding variance is:

$$\mathrm{var}\left[L_B(i)\right] = \frac{1}{W}\frac{\alpha}{\beta^2} \cdot \kappa(\alpha,\beta,\theta)\left(\frac{\beta}{\beta+2\theta}\right)^2 \tag{20}$$

where:

$$\kappa(\alpha,\beta,\theta) = \left[1 + \alpha\left(\frac{\theta}{\beta+\theta}\right)^2\right]\left[1 + \frac{\theta^2}{\beta^2+2\beta\theta}\right]^{\alpha}. \tag{21}$$

To analyze the selector variance and the way parameter θ can be employed to tune the algorithm, we consider the ratio $\mathrm{var}\left[L_B(i)\right]/\mathrm{var}\left[L_\mu(i)\right]$. Its behaviour is plotted in Fig. 1, where each curve refers to a different traffic load value. When $\theta = 0$ the algorithm simply computes the mean value (since all weights in (18) are equal to $1/W$) and $L_B(i) = L_\mu(i)$. As long as $\theta/\beta \leq 0.1$, the multiplier is always < 1 but the gain using $L_B(i)$ in place of $L_\mu(i)$ is negligible. In the range $0.1 < \theta/\beta \leq 1$ behaviour depends more significantly on traffic load. For network loads between 5% and 45%, packet selector output variance can be reduced by up to an order of magnitude, whereas with higher loads it tends to increase.

Fig. 1 suggests that, as θ increases, behavior of the Boltzmann selector goes from mean packet selection (best high traffic load performance) to approaching minimum packet selection (low traffic load benchmark). A suitable choice of θ ensures that the algorithm will outperform $L_\delta(i)$ in most cases, while keeping variance upper bounded by that of $L_\mu(i)$.

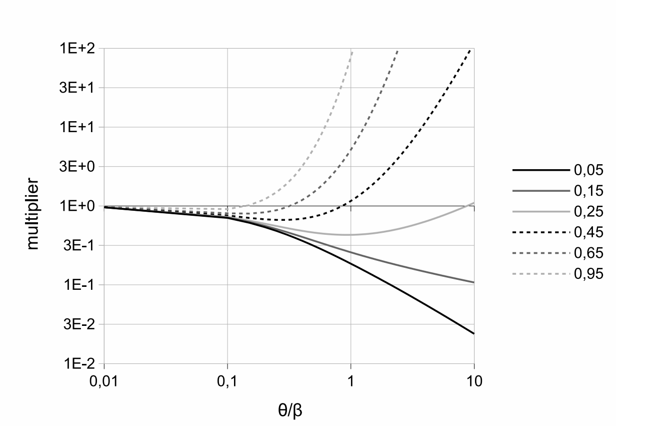

Fig. 1. Plot of function $\kappa(\alpha,\beta,\theta)\frac{\alpha}{\beta^2}\cdot\left(\frac{\beta}{\beta+2\theta}\right)^2$. Values are given as functions of the ratio (θ/β). Curves are obtained for different values of α, for which the corresponding traffic load, according to [12], is shown in the legend.

An adaptive selector would be impractical, however θ can be employed as a tuning parameter. Plots in Fig. 1 show that very good performance can be achieved, up to 45% traffic load, by setting $\theta = \beta$. Performance remains roughly comparable up to about 60%, then frequency transfer accuracy progressively degrades if network traffic increases beyond this level. However, average traffic load is usually rather far from 100% and very high loads should, arguably, be a temporary condition in a network.

V. Analysis, Modelling and Simulation

Although a PDV model based on the Gamma pdf can provide accurate fits of experimental network data, it relies on two parameters whose values are non-linear functions of end-to-end traffic load. Thus it is not readily generalizable, as it is hard to directly relate α and β to the network configuration. We propose a PDV model that provides a simple explanation for first- and second-order moments of end-to-end PDV experimental data, allowing better insight for the analysis of frequency transfer. For this reason, we start with the analysis of experimental data.

A. Experimental data

An assessment of PDV for a typical network environment can be obtained using experimental parameter values provided in [12] and obtained from a network conforming to the ITU-T hypothetical reference model HRM-1, where PTP *sync* packets were used for timing. The model refers to high-speed optical fiber links and assumes an end-to-end packet network connection using links with 1 Gb/s and 10 Gb/s bit rates, as would suit a telecommunications network. In Fig. 2 grey lines plot PDV mean and standard deviation values as functions of traffic load, expressed in percent of link capacity [12]. The mean value increases approximately in proportion to the load, but standard deviation grows at a much slower rate. Eventually, both curves drop slightly, as the network approaches traffic load saturation. Numerical values in Fig. 2 are specific of the test set-up, but these general trends are shared by most network connections.

HRM-1 measurement data are a very useful reference for this analysis, since requirements of distributed measurement systems are not dissimilar from those of the telecommunications field. For instance the 1-μs time synchronization tolerance, required for PMUs in a smart grid wide-area monitoring system, is similar to a requirement for fourth generation Long Term Evolution (4G LTE) wireless network base stations.

B. PDV profiling by PTP

Although, in principle, oversampling is not dependent on a specific time protocol, PTP packets are particularly well-suited as measurement probes to profile PDV behavior. On account of the small size of PTP *sync* packets and comparatively low rate, a highest-priority PTP flow has negligible effect on traffic, so that queueing delays affecting latency depend on the characteristics of the non-priority packet flow. We note that:

1) a PTP *sync* packet represents an ethernet payload of just 80 bytes, whereas typical payload can be up to 1500 bytes long. It is reasonable to assume that average non-priority packet size in any general-purpose network is, on average, nearly two orders of magnitude greater (in [12] packets in the measured bulk flow were up to 2000 bytes long);
2) *Sync* packets, having the highest priority, are allowed to overcome ordinary traffic at the earliest opportunity, remaining in a buffer only until the current transmission is completed. When delay occurs, it can be assumed to be the result of collision with a non-priority packet accessing the physical interface at a node;
3) typical PTP *sync* packet oversampling rate could be $R = 32$ packets per second (pkt/s), resulting in a corresponding bit rate of about 32 kb/s, whose impact would be negligible in most network links.

TABLE I
ITU-T TRAFFIC PROFILES

	TM1			TM2		
s	1	2	3	1	2	3
b_s	64	576	1518	64	576	1518
p_s	80	5	15	30	10	60
$P[s]$	16.6	9.4	74.0	2	5.8	92.2

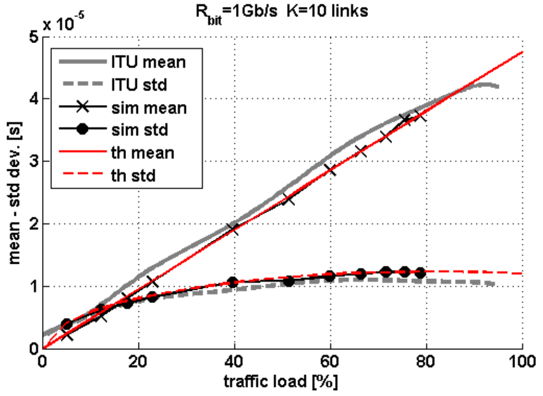

Fig. 2. Mean value and standard deviation of PDV versus traffic load expressed in percent of capacity, obtained from ITU-T experimental data referring to a HRM-1 end-to-end connection [12] (continuous lines). Red lines represent the fits obtained by the PDV model proposed in Section V-C, symbols represent simulation results.

C. PDV analytical model

The model we propose is based on the following set of assumptions:

1) an end-to-end connection is composed of K links and the total PDV $q_{MS}(n)$ experienced by a time packet is the sum of variable delays $q_l(n)$ (with $l = 1, \ldots, K$) experienced at each link;
2) traffic on the l-th link is described by two characteristics: the traffic load ρ_l expressed as a fraction of capacity, and the traffic profile TMx = $\{[b_s, p_s], s = 1, 2, \ldots\}$. The latter defines the amount of packets of different sizes in the flow, specifying the percentage p_s of packets with byte length b_s for each size index s. Data for traffic profiles specified by ITU-T are reported in Table I;
3) the probability of collision between a PTP packet and an ordinary packet in a network switch is equal to the incoming link traffic load relative to its capacity, ρ_l;
4) non-preemptive switches are considered, therefore when a collision occurs, the highest-priority packet has to wait for completion of the current packet transmission. Consequently, the random wait time is uniformly distributed between 0 and the s-type packet duration τ_s;
5) the probability $P[s]$ that collision involes a s-type packet is equal to the percentage of s-type bytes in the overall flow: $P[s] = b_s p_s \left(\sum_s b_s p_s\right)^{-1}$.

Accordingly, delay length q_l for the l-th link is described by a random variable with probability density function:

$$f_{q_l}(a) = (1 - \rho_l)\delta(a) + \rho_l \sum_s f_{q_l|s}(a) P[s] \qquad (22)$$

where $f_{q_l|s}(a)$ is the uniform pdf of wait time for a s-type packet.

Assuming each link shares the same traffic profile, we assign an equal value ρ to traffic load at each link. Accordingly, $\mathrm{E}[q_l] = \rho\bar{\tau}/2$ for all l, where $\bar{\tau} = \sum_s P[s]\tau_s$ is the average packet duration. The mean end-to-end delay $\mathrm{E}[q_{MS}]$ experienced by a PTP packet is then:

$$\mathrm{E}[q_{MS}] = \frac{\bar{\tau}}{2} K\rho. \qquad (23)$$

End-to-end delay variance can be straightforwardly obtained from (22) as:

$$\mathrm{var}[q_{MS}] = \frac{\bar{\tau}^2}{3} K\rho \left[\frac{\sum_s P[s]\tau_s^2}{\left(\sum_s P[s]\tau_s\right)^2} - \frac{3}{4}\rho \right]. \qquad (24)$$

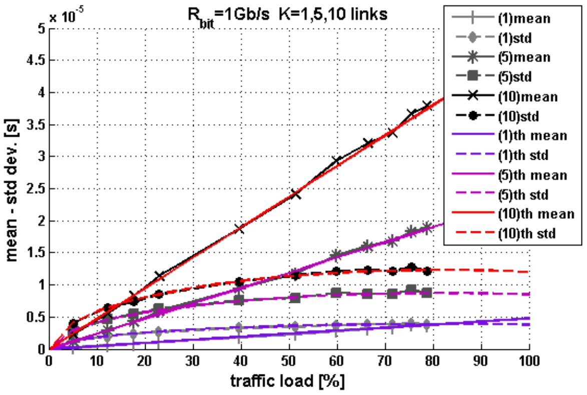

Fig. 3. PDV mean values and standard deviations of versus traffic load (in percent of capacity) for end-to-end paths composed by 1, 5 and 10 links. Symbols represent simulation results, coloured lines refer to theoretical results.

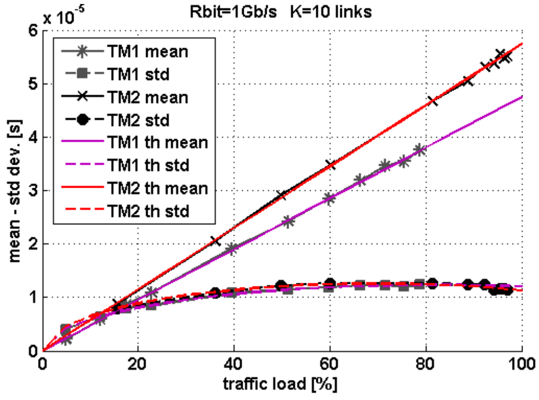

Fig. 4. PDV mean values and standard deviations versus traffic load (in percent of capacity) for end-to-end paths with two different traffic profiles. Symbols represent simulation results, coloured lines refer to theoretical results.

This general expression depends only on network and traffic parameters. Therefore can be applied to any traffic profile within the model assumptions given above.

Plots generated by Eqs. (23) and (24), shown by red lines in Fig. 2, evidence very good agreement in spite of the simplifying assumptions. The linear regression correlation coefficient of the PDV mean value is extremely close to 1 until traffic saturation effects come into play (at about 70% of capacity)[1]. From experimental data we extrapolated a mean packet duration slightly shorter than 10 μs, corresponding to a mean packet size of about 1200 bytes at 1 Gb/s, in very good accordance with the description of traffic by a TM1 profile.

The analytical model requires few parameters to describe network behaviour, that is, the number of links, the traffic load and traffic profiles. It can be very useful in packet selector design and tuning. For instance, Fig. 3 shows end-to-end delay mean values and variances for networks composed of different numbers of links. On the other hand, Fig. 4 takes into account the impact of different traffic profiles. It can be seen in this case that variance remains approximately unchanged between TM1 and TM2. Symbols drawn over the plots refer to simulation analysis results, that further validate the approach.

VI. Performance analysis

Results presented in this Section are obtained by reproducing in a combined simulation the effects of both packet network and slave clock behaviour. We emphasize the importance of a comprehensive analysis, that includes the stability of the local clock. To represent this, we employed the discrete-time two-variable state-space oscillator model described in [20]. On the other hand, since time information are carried by high-priority UDP packets, there is little need to reproduce protocol aspects and it suffices to generate long streams of latency values where the PDV component is varied to represent different traffic load conditions. This set-up allows the test of different packet selector algorithms. Their output is employed to discipline the slave clock, whose model parameters σ_1 and σ_2 can also be changed to represent clocks with different phase noise and stability properties.

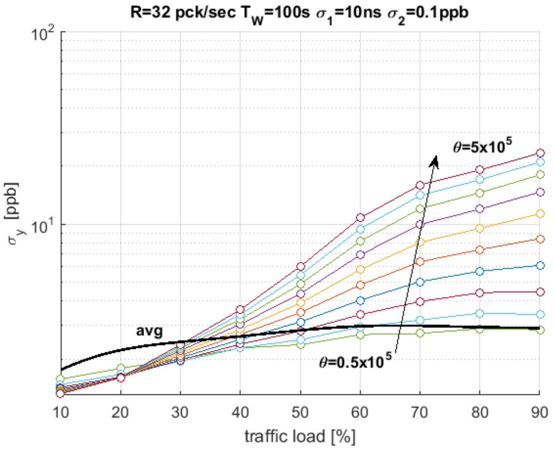

Fig. 5. Variation of the standard deviation $\sqrt{\mathrm{var}[\hat{y}(i)]}$ with traffic load, for different values of the Boltzmann weighting parameter θ.

A. Optimization of the Boltzmann packet selector

It was shown in Section III-B that performance of the Boltzmann packet selector depends on the weighting parameter θ, whose optimal value is related to β. The model we introduced allows to find an approximate estimate in the form:

$$\hat{\beta} = \frac{\mathrm{E}[q_{MS}]}{\mathrm{var}[q_{MS}]} = \frac{3}{2}\cdot\frac{1}{\bar{\tau}}\left[\frac{\sum_s P[s]\tau_s^2}{\left(\sum_s P[s]\tau_s\right)^2} - \frac{3}{4}\rho\right]^{-1} \qquad (25)$$

This is inversely proportional to the average packet duration $\bar{\tau}$ (or, proportional to the packet rate of the end-to-end network connection at full load). Fig. 5 shows the dependence of $\sqrt{\mathrm{var}[\hat{y}(i)]}$ on traffic load and shows in better detail the effect of the choice of θ for packet selector $L_B(i)$, using the experimental parameters related to Fig. 2.

These results suggest that a suitably simple choice for $L_B(i)$ can be the constant value: $\theta \cong 1/\bar{\tau}$. This simple criterion is

[1] An even better fit can be obtained by assigning specific values of p_l to each link.

easy to apply and can be assumed to match *prevailing* load conditions in the network.

B. Packet selector comparison

The design of packet-based frequency transfer requires an appropriate choice of values for a number of interdependent parameters. The results given here should help provide a sufficiently detailed understanding.

Plots in Fig. 6 show the standard deviation $\sqrt{\mathrm{var}[\hat{y}(i)]}$ of the fractional frequency offset estimate, with oversampling window length T_W presented on the abscyssa. Clock parameters σ_1 and σ_2 are shown at the top of the figure. PDV standard deviation of 10 μs and packet rate value $R = 32$ packets per second were considered.

Continuous lines refer to theoretical curves, obtained from (10) using the appropriate values of $\mathrm{var}[L(i)]$ derived in Section III-B for packet selectors $L_\mu(i)$ and $L_B(i)$. Symbols instead refer to results obtained by simulation and agree very well with the theory. The trace labelled 'uniform' represents the no oversampling case, with a single *sync* packet sent at uniform time intervals T_W. Therefore, packet rate is $R = 1/T_W$ and $\hat{y}(n)$ is obtained by (2). For the other two packet selectors, oversampling rate R is indicated in the figure. It can be noticed that curves referring to $L_\mu(i)$ and $L_B(i)$ have an increased slope and the enhancement provided by oversampling is proportional to $\sqrt{RT_W}$, up to the optimal window length. The selected value for the tuning parameter of $L_B(i)$ is $\theta = 10^5$, that brings the ratio θ/β to about 0.5. Actual results therefore agree with Fig. 1 which shows both packet selector algorithms having similar performance at this traffic level.

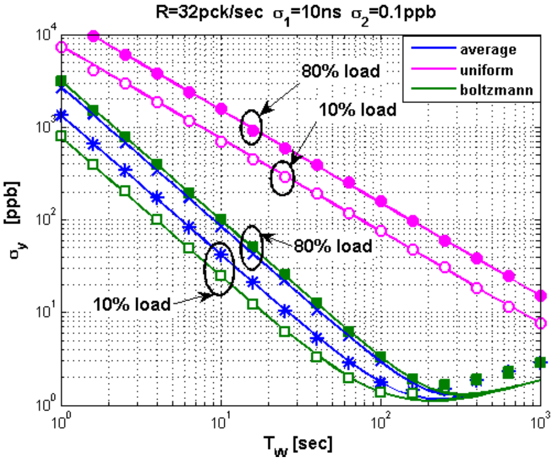

Fig. 6. Effect of traffic load variation on the standard deviation $\sqrt{\mathrm{var}[\hat{y}(i)]}$ of the fractional frequency estimate. Values given refer to 10% and 80% of the network capacity.

The effect of traffic load variations, again, is consistent with the fact that, even with a fixed value of θ, the estimate standard deviation obtained with $L_B(i)$ is significantly reduced in a lightly loaded network.

C. A numerical exercise

A packet-based frequency transfer scheme can be designed employing the set of relationships provided in Sections III and V-C. As an example we consider a generic slave clock modelled by parameters $\sigma_1 = 10^{-8}$ and $\sigma_2 = 10^{-10}$. These yield a stability of approximately 3.5 parts per billion (ppb) over a one-hour interval. The optimal window for the estimation of fractional frequency deviation is $T_{W(opt)} \cong 170$ s. We employ this value, trying to approach the best performance for the given clock.

Let us assume that time packets through a general purpose network are characterized by a PDV standard deviation $\sigma_{PDV} \cong 10$ μs. For oversampling, we consider a typical packet rate $R = 32$ pkt/s, which results in $W = 5440$ time packets within an oversampling window. Using packet selector $L_\mu(i)$, this yields $\mathrm{var}[L(i)]^{\frac{1}{2}} = 135$ ns and, from (10), the contribution of PDV to $\mathrm{var}[\hat{y}(i)]$ is about twice that associated with σ_1, the resulting standard deviation being 1.3 ppb.

A rather well-balanced situation results here since, from (11), we obtain a time offset standard deviation of 13.3 μs over a one-hour interval. This means the slave clock frequency is now synchronized to within a few ppb, while stability remains basically unchanged (3.7 ppb standard deviation for a one-hour interval). The window length T_W determines the adjustment rate of the slave clock, therefore time uncertainty within this interval $U_t = k\sqrt{\mathrm{var}[\hat{y}(i)]}T_W$ is equal to to $\pm 0.5\mu$s with $k = 2.6$ (99% confidence assuming Gaussian approximation).

Application requirements determine performance acceptability. For instance, a slave clock with these characteristics could be acceptable for the 1 μs PMU synchronization target, provided a continuous flow of *sync* packets can be maintained. Any interruption or degradation, possibly caused by network congestion, would cause degradation in a short time. A slave clock with better stability can improve the reliability of the distributed measurement system.

VII. Conclusions

This work provides some insight and useful results for designing oversampling in a packet-based frequency transfer scheme, evaluating all possible sources of uncertainty. To the authors' knowledge, no works so far have analyzed the interdependence between slave clock behavior and PTP packet queuing delays to show their effect on synchronization accuracy. Therefore, optimization criteria and packet selector performance comparisons presented in the paper should help in the choice of packet selection methods.

Our results show it is possible to employ a general-purpose packet network to precisely transfer frequency, with an accuracy level that may prove of interest to a number of distributed measurement applications. A few issues can be highlighted:

1) the design of packet-based frequency transfer requires careful consideration of a number of interacting issues, involving the stability of clocks in measurement nodes, the stationarity of network configuration and the variability of traffic patterns;

2) oversampling produces a limited amount of additional network traffic and only requires suitable adaptation of clock synchronization algorithms;
3) a packet selection algorithm is essential for packet-based frequency transfer. It should be designed to minimise estimation variance first, then to provide robustness towards variations of traffic parameters;
4) since frequency transfer accuracy is degraded under high traffic load, specifications should allow for the possibility of network congestion. Clock algorithms should be able to detect this condition, that can be assimilated to signal degradation in GPS-based synchronization. Since usually high traffic loads are not long-lasting conditions, stability specifications of current clocks should be adequate to deal with the issue as a transient impairment;
5) whereas PPS is usual for GPS clock synchronization, optimal measurement rate for packet-based transfer is slower, intervals ranging from tens to hundreds of seconds. Consequently, local clock stability plays a more important role in achieving measurement accuracy. Using longer intervals, though seemingly attractive, could be made pointless by network configuration changes.

The analytical model proposed in Section V enhances the understanding of PTP packet propagation, allowing to relate end-to-end PDV model parameters with traffic profile information. In turn, this allows tuning of the robust packet selector proposed in Section IV. Although the results presented in this paper are based on a limited number of case studies, they do provide the basis for the design and deployment of effective frequency transfer through non-dedicated packet networks.

References

[1] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, "IEEE 1588-Based Synchronization System for a Displacement Sensor Network", IEEE Trans. Instrum. Meas., vol. 57, no. 2, Feb. 2008, pp. 254-260,

[2] A. Carta, N. Locci, C. Muscas, "GPS-Based System for the Measurement of Synchronized Harmonic Phasors", IEEE Trans. Instrum. Meas., vol. 58, no. 3, Mar. 2009, pp. 586-593.

[3] Recommendation ITU-T G.8261/Y.1361 (2013), *Timing and synchronization aspects in packet networks*, ITU, Geneva, 2014.

[4] J.L. Ferrant, S. Ruffini, "Evolution of the Standards for Packet Network Synchronization", IEEE Comm. Magazine, Feb. 2011, pp.132-138.

[5] W. Lewandowski, C. Thomas, "GPS time transfer", Proc. IEEE, vol. 79, no. 7, July 1991, pp. 991-1000.

[6] P. Daly, I. D. Kitching, D. W. Allan, T. K. Peppler, "Frequency and Time Stability of GPS and GLONASS Clocks", Int. J. Satellite Comm., vol. 9, pp. 11-22, 1991.

[7] IEEE Std 1588$^{\text{TM}}$-2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE, New York, 2008.

[8] Recommendation ITU-T G.8260 (2012), *Definitions and terminology for synchronization in packet networks*, ITU, Geneva, Switzerland, 2012.

[9] P. Moreira, J. Serrano, T. Wlostowski, P. Loschmidt, G. Gaderer, "White Rabbit: Sub-Nanosecond Timing Distribution over Ethernet", Int. IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2009, Brescia, Italy, Oct. 12-16, 2009, pp. 58-62.

[10] M. Rizzi, S. Rinaldi, P. Ferrari, A. Flammini, D. Fontanelli, D. Macii, "Enhancing Accuracy and Robustness of Frequency Transfer Using Synchronous Ethernet and Multiple Network Paths", to appear in: IEEE Trans. on Instr. and Meas., DOI: 10.1109/TIM.2016.2559950

[11] Recommendation ITU-T G.8262/Y.1362 (2010), *Timing characteristics of a synchronous Ethernet equipment slave clock*, ITU, Geneva, 2010.

[12] Recommendation ITU-T G.8263/Y.1363 (2012) – Amendment 2, *Timing characteristics of packet-based equipment clocks*, ITU, Geneva, 2014.

[13] C. Zucca and P. Tavella, "The clock model and its relationship with the Allan and related variances", IEEE Trans. Ultrason., Ferroelect., Freq. Contr., vol. 52, no. 2, pp. 289296, 2005.

[14] I. Hadžić, D.R. Morgan, "On packet selection criteria for clock recovery", Int. IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2009, Brescia, Italy, Oct. 12-16, 2009, pp. 35-40.

[15] M. Anyaegbu, C. Wang, W. Berrie, "Dealing with packet delay variation in IEEE 1588 synchronization using a sample-mode filter", IEEE Intelligent Transportation Systems Magazine, Winter 2013, pp. 20-27.

[16] G. Giorgi, C. Narduzzi, "On the Accuracy of Packet-based Measurements in the PTP Telecom Profile", IEEE Int. Workshop on Measurement and Networking, M&N 2015, Coimbra, Portugal, Oct. 12-13, 2015, pp. 134-139.

[17] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, P. Van Mieghem, "Analysis of End-to-End Delay Measurements in Internet", Proc. Passive and Active Measurement Workshop, PAM 2002, Fort Collins, CO, USA, 25-26 March 2002, pp. 1-8

[18] P. Holleczek, R. Karch, R. Kleineisel, S. Kraft, J. Reinwand, V. Venus, "Statistical Characteristics of Active IP One Way Delay Measurements", IEEE Int. Conf. on Networking and Services, ICNS 2006, Silicon Valley, CA, USA, 16-18 July 2006, pp. 1-5.

[19] L.B. Romdhane, "A soft-min based neural model for causal reasoning", IEEE Trans. on Neural Networks, vol. 17, no. 3, May 2006, pp. 732-744.

[20] G. Giorgi, C. Narduzzi, "Performance analysis of Kalman filter-based clock synchronization in IEEE 1588 networks", IEEE Trans. on Instr. and Meas., vol. 60, n. 8, pp. 2902-2909, Aug. 2011.