Kalman Filtering for Multi-path Network Synchronization

G. Giorgi and C. Narduzzi

Department of Information Engineering, University of Padova
via G. Gradenigo 6/b, Padova I-35131, Italy
e-mail: {giada.giorgi, claudio.narduzzi}@dei.unipd.it

***Abstract*—The accuracy of time distribution over packet networks relies on propagation delay stability and two-way symmetry in the path between the master clock and a slave clock. Multi-path time synchronization provides a possibility to safeguard accuracy by exploiting path diversity, while improving security and fault-tolerance through inherent redundancy.**

In this paper we focus on the problem of combining time information received through multiple paths, for which we propose a Kalman filter-based algorithm. We show that the algorithm allows adaptive combination of time information, based on the associated measurement uncertainty and prove that the protocol and algorithm costs of redundancy can be repaid in terms of increased synchronization robustness.

I. Introduction

Effective time distribution over packet networks places specific requirements on the variability of network behavior. In this regard, clock synchronization accuracy depends on the stability of the propagation delay between the master clock and a slave clock, as well as on its symmetry in the two directions.

Where feasible, suitable protocols and/or dedicated hardware can ensure network compliance with these requirements, but the cost and effort may not always be justifiable. In some instances, provision of multiple network paths between the master and a slave can be considered a worthy alternative. In fact, multi-path synchronization provides the possibility of improving accuracy by exploiting path diversity, but can also improve security and fault-tolerance through inherent redundancy.

Multi-path time synchronization has received growing attention lately. The approach requires specific contributions in two main areas, namely, protocol aspects connected with the management and identification of network paths [1], and algorithms to either combine or select time information received through multiple paths. The use of time information from different sources in a slave clock algorithm is considered in the Network Time Protocol [2], which provides suitable filtering and selection algorithms. Combining algorithms for multiple path synchronization are discussed in [3], [4].

Given an accurate master clock, time information received by a slave exhibits variability patterns that are for the most part network-related. In general, the accuracy of time packets significantly depends on the path followed to reach the slave and packets might occasionally be discarded or time-out, particularly in a congested network. This shows that fusing time information received through multiple paths requires careful attention to a variety of factors.

In metrology, a related problem concerns the formation of a timescale out of ensemble readings of a set of physical clocks. This is approached by suitable adaptations of the Kalman filter (KF), that provide the basis for clock combining in timescale algorithms [5], [6]. Performances of a KF-based clock algorithm have been analyzed by the authors for a single-path master-slave connection [7], and consideration has further been given to the possibility of detecting the impairments that may typically affect time information received through a network [8]. In particular, resilience can be built into a slave clock algorithm by providing redundancy in the KF-based clock servo [9].

Progressing along this line of development, in this paper we propose a KF-based algorithm for multi-path clock synchronization, describe the adaptation of KF equations to this particular application and discuss initialization assumptions required for its successful implementation.

II. Multipath Time Protocol

Adaptation of time protocols to multi-path synchronization is the object of an Internet-Draft under discussion by the Internet Engineering Task Force (IETF) [10]. This can be applied to NTP as well as to the Precision Time Protocol (PTP), IEEE Std. 1588:2008 [11], and allows a time protocol to be run between master and slave through concurrent multiple paths. We briefly summarize the main features of multi-path PTP (MPPTP), considering for the sake of simplicity just one pair of nodes, where a single slave receives a time reference from the master node.

Like in plain PTP, the master sends to the slave node *Sync* messages containing a timestamp of the instant in which the message has been issued. This is indicated as $t_{M,sync}(k)$, where the index refers to the k-th synchronization message. In MPPTP, however, multiple copies of the message are delivered to the same slave node through different paths, identified by index $j = 1, \ldots, J$ with J being the total number of paths (see Fig. 1). Since network latencies are different, each copy will be received by the slave at a different time instant.

978-1-4799-2699-2/14/$31.00 ©2014 IEEE

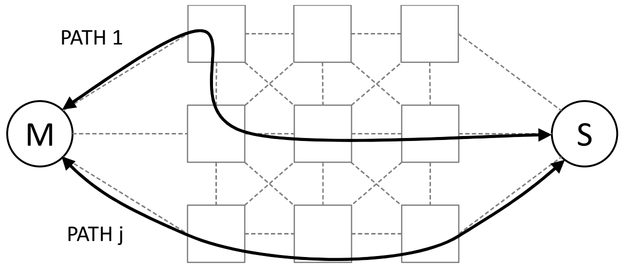

Fig. 1. Examples of master-slave synchronization through redundant paths. Square boxes represent network devices such as switches, routers and so on.

Let $t_{S,sync}(k,j)$ be the timestamp that the slave assigns to the k-th synchronization message on receiving it through the j-th path. The relationship between master and slave timestamps depends on both the network delay $d_{MS}(k,j)$ and the time offset $\theta(k)$ between slave and master:

$$t_{S,sync}(k,j) = t_{M,sync}(k) + d_{MS}(k,j) - \theta(k) \tag{1}$$

It is straightforward to see that a time offset calculated from this equation without compensating for network delay produces a biased estimate. To avoid this, the propagation delay contribution has to be corrected. The PTP procedure for its estimation involves the exchange of delay-request and delay-response messages beween the slave and master nodes. Likewise, in MPPTP a slave node will send to the master as many *Delay_Req* messages as the number of paths it is employing.

Let $t_{S,dreq}(i,j)$ be the timestamp measured by the slave before issuing the i-th delay-request message to the master through the j-th path. After a delay $d_{SM}(i,j)$ the master node will receive the message. The timestamp value obtained from the master at reception will be $t_{M,dreq}(i,j)$. This value will be returned to the slave through a *Delay_Resp* message. The relationship between master and slave timestamps is similar to (1):

$$t_{M,dreq}(i,j) = t_{S,dreq}(i,j) + d_{SM}(i,j) + \theta(i) \tag{2}$$

Then, the slave node can directly compute its own offset from the master time reference:

$$\begin{aligned}\hat{\theta}(k,j) \quad &= \quad \frac{(t_{M,sync}(k) - t_{S,sync}(k,j))}{2} + \\ &\quad + \frac{(t_{M,dreq}(i,j) - t_{S,dreq}(i,j))}{2}\end{aligned} \tag{3}$$

where $\hat{\theta}(k,j)$ is the time offset estimated by using timing information available through the j-th path. A combining algorithm is therefore necessary to fuse the available measurements before applying the resulting offset estimation to synchronize the local clock to the time reference.

III. Combining Algorithms: State of the Art

1) *Equal gain combining algorithm* – the first combining algorithm presented in [3] calculates a local time indication by averaging all the timestamps assigned by the slave node to each copy of the synchronization messages, that:

$$t^*_{S,sync}(k) = \frac{1}{J} \cdot \sum_{j=1}^{J} t_{S,sync}(k,j) \tag{4}$$

An offset estimate can be obtained by using this value in (3). This is equivalent to averaging raw time offsets that leads to the combined measurement:

$$\hat{\theta}^*(k) = \frac{1}{J} \cdot \sum_{j=1}^{J} \hat{\theta}(k,j) \tag{5}$$

It is important to note that: (i) no information about network latency over the different paths is employed, (ii) network paths are supposed to have the same statistical properties, (iii) the algorithm is memoryless: no information about the previous offset estimates are used to refine the current estimation.

2) *Switching algorithm* – this combining algorithm takes advantage of available information about the network delay introduced over a given path. This is simply obtained by calculating the absolute difference between the current estimation of the propagation delay, $d_{prop}(k,j)$ and its average value $\mathbf{E}[d_{prop}(j)]$, providing an estimate of the propagation delay standard deviation:

$$\hat{\sigma}_{d_{prop}(j)} = |d_{prop}(k,j) - \mathbf{E}[d_{prop}(j)]| \tag{6}$$

Only timing information obtained from the path having minimum variability are used.

3) *Dynamic algorithm* – in this latter case a hybrid approach is used to achieve better performances. The final estimate for the time offset is calculated by averaging only measurements provided by trusty paths, that are determined by exploiting propagation delay measurements.

Combining algorithms have been presented also in [4], where more sophisticated strategies are adopted to identify reliable information and discard instead unworthy measurements.

It is very important to note that all these algorithms perform a sort of weighted mean where weights depend on the variability of the propagation delay, possibly discarding too noisy measurements. This leads to consider a data fusion algorithm expressly developed for combining measurements, as well as other information available about the generating process, by considering their *uncertainty*. This algorithm, described in the following Section, exploits Kalman filtering capabilities.

IV. Combining by Kalman Filtering

In multi-path synchronization, measurements of time offset from master time are obtained through several paths, but they are all referred to the same local slave clock. Under this assumption, the local clock model remains unchanged and it is possible to apply well-known KF equations [7], which are summarized for completeness in Tab. I.

In particular, it is straightforward to see that the state transition matrix $\mathbf{A}$ and the corresponding process noise covariance matrix $\mathbf{Q}$ remain unchanged, since the behavior of the local clock with respect to its time reference is unaffected by the presence of multiple paths. If a time-invariant KF implementation is considered, we have:

$$\mathbf{A} = \left[\begin{array}{cc} 1 & T \\ 0 & 1 \end{array} \right] \tag{7}$$

where T is the synchronization period, and:

$$\mathbf{Q} = \left[\begin{array}{cc} \sigma_\theta^2 & 0 \\ 0 & \sigma_\gamma^2 \end{array} \right]. \tag{8}$$

Parameters σ_θ^2 and σ_γ^2 are the standard deviations for the two diffusion noise components associated, respectively, to the offset $\theta(k)$ and fractional frequency deviation $\gamma(k)$ components of the local clock state vector $\underline{x}(k) = [\theta(k)\ \gamma(k)]^T$ (superscript T denotes the transpose operator). Further details can be found in [7], [12], [13] where an in-depth analysis on how matrices $\mathbf{Q}$ and $\mathbf{R}$ have to be initialized is reported.

Let $\hat{\theta}(k,j)$ be the measurement of time offset between local clock and master time referred to the j-th path. The KF measurement vector $\underline{z}(k)$ contains the set of all available offset measurements, with $j = 1, \cdots, J$:

$$\underline{z}(k) = \left[\hat{\theta}(k,1), \cdots, \hat{\theta}(k,j), \cdots, \hat{\theta}(k,J)\right]^T \tag{9}$$

which leads to the measurement equation:

$$\underline{z}(k) = \mathbf{H} \cdot \left[\begin{array}{c} \theta(k) \\ \gamma(k) \end{array} \right] + \underline{v}(k). \tag{10}$$

The measurement matrix $\mathbf{H}$ is a $2 \times J$ matrix, where the number of columns corresponds to the number of clock state variables, while the number of rows corresponds to the number of different paths between master and slave. Since only offset measurements will be considered for KF updates, $\mathbf{H}$ can be written simply as:

$$\mathbf{H} = \left[\underline{1}^J, \underline{0}^J\right] \tag{11}$$

TABLE I
SUMMARY OF KALMAN FILTER EQUATIONS

Prediction equations

$\underline{\hat{x}}^-(k) = \mathbf{A} \cdot \underline{\hat{x}}(k-1)$
$\mathbf{P}^-(k) = \mathbf{A} \cdot \mathbf{P}(k-1) \cdot \mathbf{A}^T + \mathbf{Q}$

Update equations

$\underline{\hat{x}}(k) = \underline{\hat{x}}^-(k) + \mathbf{K}(k) \cdot \left(z(k) - \mathbf{H} \cdot \underline{\hat{x}}^-(k)\right)$
$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}(k) \cdot \mathbf{H}) \cdot \mathbf{P}^-(k)$
$\mathbf{K}(k) = \dfrac{\mathbf{P}(k)^- \cdot \mathbf{H}^T}{\mathbf{H} \cdot \mathbf{P}(k)^- \cdot \mathbf{H}^T + \mathbf{R}}$

Legend: $\underline{\hat{x}}^-(k)$ is the a-priori state estimate at the k-th time instant, $\underline{\hat{x}}(k)$ and $\underline{\hat{x}}(k-1)$ are the a-posteriori state estimates for respectively the k-th and $(k-1)$-th time instants. $\mathbf{P}^-(k)$ is the a-priori error covariance matrix at the time instant k-th while $\mathbf{P}(k)$ and $\mathbf{P}(k-1)$ are the a-posteriori error covariance matrices during time instants k and $k-1$. Finally $\mathbf{K}(k)$ is the Kalman gain matrix.

where $\underline{1}^J$ is a column vector having all elements equal to 1, while $\underline{0}^J$ identifies a column vector composed by all zeroes. Both vectors have equal size J.

$\underline{v}(k) = [v(k,1), \cdots, v(k,J)]^T$ is a *measurement* noise vector that represents the noise contributions for all the propagation paths. We make the hypothesis of independent paths, which allows to consider uncorrelated noise contributions. With this assumption, the corresponding measurement noise covariance matrix $\mathbf{R}$ results to be diagonal. The simplification is introduced in the interest of mathematical tractability although, in practice, paths may share common segments. In this case correlation can be caused by network traffic, unless the bandwidth of shared links is large enough to make the effect negligible. Investigation of this issue is left to future developments of this work.

The element of $\mathbf{R}$ in position (j,j) provides the variance of noise superposed to offset measurements related to the j-th path:

$$\mathbf{R} = \left[\begin{array}{ccc} \sigma_{v,1}^2 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & \sigma_{v,J}^2 \end{array} \right]. \tag{12}$$

In particular, $\sigma_{v,j}^2 = \sigma_{d_{prop}(j)}^2 + \sigma_{vt}^2$ where σ_{vt}^2 represents the uncertainty contribution due to the timestamping mechanism, while the contribution introduced through (3) by propagation delay asymmetry over the j-th path, is accounted for by $\sigma_{d_{prop}(j)}^2$.

It should be remembered that propagation delays may vary due to different network traffic conditions, therefore the measurement noise matrix $\mathbf{R}$ needs to be continuously updated to reflect the actual conditions over different paths.

The Kalman gain matrix $\mathbf{K}(k)$ results to be a $J \times 2$ matrix. Elements in the j-th row of this matrix represent the Kalman gain, respectively for offset and skew estimation, assigned to the measurement obtained through the j-th path.

An event-based implementation of the Kalman filter can also be derived for reducing computational costs, hence energy consumption, and for better dealing with aperiodic timing informations. More details are reported in [12]. Extension of this time-varying algorithm to multipath synchronization, as proposed here for the simpler time-invariant KF case, can also be obtained rather straightforwardly.

V. COMPARISON WITH COMBINING ALGORITHMS

The proposed KF-based algorithm implements a data-driven combining policy. It may be interesting to elaborate on this feature by an analysis of relationships with the combining algorithms introduced in Section III. To this aim, it is useful to explicitly write the update equation for the time offset estimate, as follows:

$$\underline{\hat{\theta}}(k) = \underline{\hat{\theta}}^-(k) + \sum_{j=1}^{J} K(1,j) \cdot \left(\theta(k,j) - \underline{\hat{\theta}}^-(k)\right) \tag{13}$$

where $K(1, j)$ is the Kalman gain related to the j-th offset measurement. This takes the form:

$$K(1,j) = \frac{\dfrac{1}{\sigma_{vj}^2}}{\dfrac{1}{p_{11}} + \sum_{i=1}^{J} \dfrac{1}{\sigma_{vi}^2}} \tag{14}$$

where $p_{11} = \mathbf{P}(1,1)$ and $p_{12} = \mathbf{P}(1,2) = \mathbf{P}(2,1)$ are elements of the a-posteriori covariance error matrix (defined in Table I).

It can be noted that gain $K(1, j)$ is inversely proportional to the variance of noise affecting offset measurements obtained through the j-th path. Therefore, the noisier a measurement is, the smaller its effect will be on the final estimate. Nevertheless, variance p_{11} depends on clock stability and on the time elapsed from the last synchronization message.

When measurements are very noisy compared with clock stability, the KF tends to retain the previously estimated value and neglect instead new measurements. In fact when $p_{11} \to 0$ it follows $K(1, j) \to 0$ for any j and, therefore, $\hat{\underline{\theta}}(k) \simeq \hat{\underline{\theta}}(k)^-$.

On the contrary, if the local clock is very unstable and/or timing messages are sent at a very low rate, p_{11} may take considerably high values. Under this assumption the Kalman gain becomes:

$$K(1,j) \simeq \frac{\dfrac{1}{\sigma_{vj}^2}}{\sum_{i=1}^{J} \dfrac{1}{\sigma_{vi}^2}} \tag{15}$$

and (13) then depends only on currently available measurements:

$$\hat{\underline{\theta}}(k) \simeq \sum_{j=1}^{J} K(1,j) \cdot \theta(k,j). \tag{16}$$

Being $\sum_{j=1}^{J} K(1, j) = 1$, in this way a memoryless Kalman filter is obtained. It is interesting to note that, if measurements are equally distributed, that is $\sigma_{vj}^2 = \sigma_v^2$ for $j = 1, \cdots, J$, the results provided by the Kalman filter with (16) are the same obtained with the *equal gain combining algorithm*.

An example of this situation is a synchronized system affected by high timestamping noise (e.g., when software timestamping is done). In this situation one has: $\sigma_{v,j}^2 \simeq \sigma_{v,t}^2$, therefore measurement matrix $\mathbf{R}$ no longer depends on network traffic conditions.

The *switching algorithm* can be obtained instead by suitably modifying the KF measurement noise matrix. In particular, if j^* results to be the path with minimum variability, the matrix $\mathbf{R}$ will be updated as follows:

$$\mathbf{R} = \begin{bmatrix} \infty & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & \sigma_{v,j^*}^2 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & \infty \end{bmatrix} \tag{17}$$

Only the element $\mathbf{R}(j^*, j^*)$ assumes a finite value, while other elements on the main diagonal are given infinite value. Consequently, for the Kalman gain one has: $K(1, j) = 0$ when $j \neq j^*$ and $K(1, j) = 1$ only when $j = j^*$. The resulting KF offset estimate becomes:

$$\hat{\underline{\theta}} = \theta(k, j^*) \tag{18}$$

that corresponds to the *switching algorithm* estimate.

Finally, the *dynamic combining algorithm* calculates the average of a restricted set of offset measurements that satisfy a given validation criterion. It is possible to reformulate also this combining algorithm in the Kalman filter framework when the following assumptions are satisfied:

- when links are working normally, that is, they aren't congested, timing messages are all affected in the same way, therefore $\sigma_{vj}^2 = \sigma_v^2$ for $j = 1, \cdots, J$, as in the *equal gain combining algorithm*.
- for measurements received through congested links the variance associated to propagation delay results instead to be very large. It follows, e.g, that for the congested j-th path: $1/\sigma_{d_{prop}(j)}^2 \simeq 0$, hence: $K(1, j) = 0$, thus the contribution due to the j-th measurement is nil.

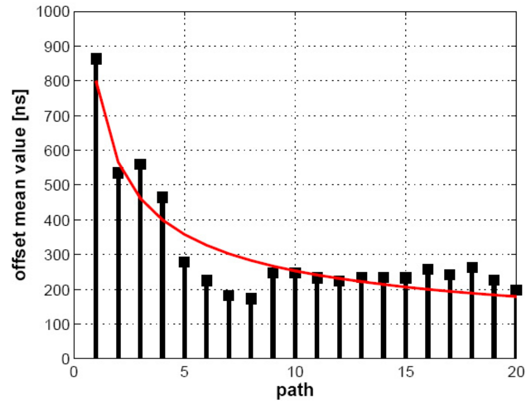

Fig. 2. Mean value of the time error between synchronized local clock and time reference with asymmetric path propagation delays. The x-axis shows the number of paths considered by the multipath synchronization protocol.

VI. Results

A. Asymmetric paths

Unbiased estimation of the actual time offset between master and slave can only be obtained under the hypothesis of symmetric propagation delays, that is when: $d_{MS}(k, j) = d_{SM}(i, j)$. Otherwise, when $d_{MS}(k, j) \neq d_{SM}(i, j)$ the synchronized local clock presents a bias with respect to its time reference.

Propagation delay can be modelled by the sum of a *packet delay variation* (PDV) random component and a purely deterministic term. The latter depends on the structure of a path, i.e., the number of links and switches that make up the end-to-end connection. In general, neither PDV nor deterministic delay are symmetric in the two directions, and the (bounded) average delay asymmetry is equally likely to be positive or negative. This justifies taking a mean value of

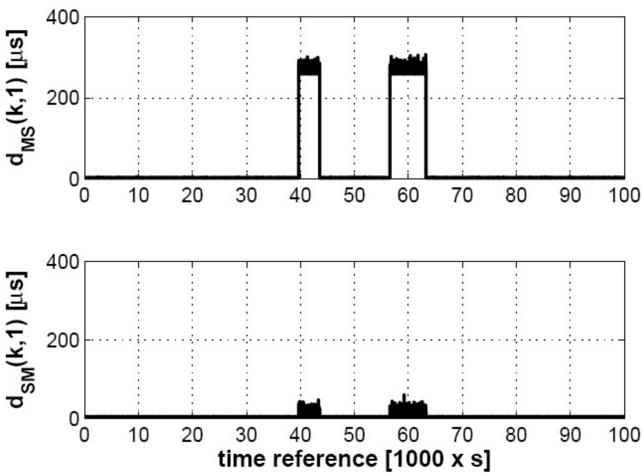

Fig. 3. Propagation delay over time in presence of bursts.

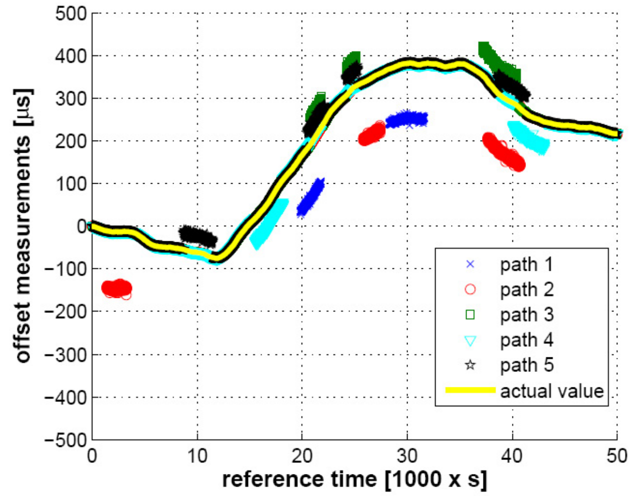

Fig. 4. Simulated offset measurements obtained by a MPPTP from different paths. The existing time offset between master clock and slave local clock is reported by a thick line for comparison.

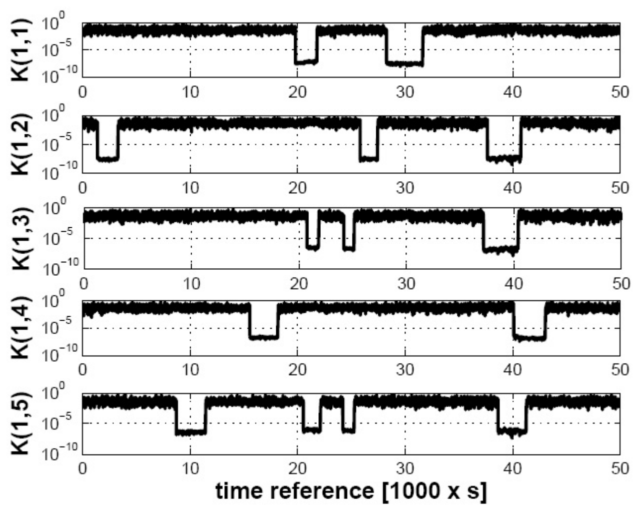

Fig. 5. Kalman gains over the time when 5 independent paths are considered.

the independently distributed delay asymmetry for different paths from a zero mean uniform distribution, as assumed in [3]. With this hypothesis, a multipath approach can effectively reduce bias error in the final estimate of offset between time reference and synchronized local clock.

Simulations have been done by generating exponentially distributed latency superposed to a time-constant contribution: $d(k,j) = d_{\mathcal{E}}(k,j) + \Delta_d(j)$ where $d_{\mathcal{E}}(k,j) \in \mathcal{E}(\lambda)$ and $\Delta_d(j) \in \mathcal{U}(\Delta_{min}, \Delta_{max})$. Parameters have been set respectively to: $\lambda^{-1} = 200$ ns, $\Delta_{min} = 1$ μs and $\Delta_{min} = 3$ μs.

Results obtained by combining measurements by a Kalman filter are reported in Fig. 2 that illustrates how the mean value of the time error varies according to the number of paths. In particular, a good fit can be found by assuming that the time error drops with the square root of the path number, as can be noted by comparing the experimental values with the curve fitting. Comparing these results with those illustrated in ([3], Fig. 10), it can be concluded that a KF-based combining algorithm produces qualitatively similar results.

B. Bursty traffic

Communication networks under real working conditions may occasionally become congested due to the presence of large data bursts. As discussed in [3], considering multiple paths may mitigate the effect on a synchronized local clock, under the assumption that these paths are independent. The effect of bursty traffic is now evaluated when a combining algorithm based on the Kalman filter is used.

When a network link becomes congested the delay experienced by timing messages may present a strong variability and a local increment of the mean value, as can be noted in Fig. 3. In general bursts mainly affect a propagation path only in one direction (between master to slave or viceversa), thus causing asymmetry. This results in a large measurement noise affecting time offset measurements, as can be noted in Fig. 4.

Measurements have been obtained by considering the same master-slave pair where the master clock is supposed to be absolutely stable, while the slave clock has been modeled by assuming $\sigma_\theta = 10$ ns and $\sigma_\gamma = 1$ ns/s. The noise introduced by the timestamping mechanism is supposed to be gaussian distributed with zero mean and standard deviation $\sigma_{vt} = 10$ ns.

Synchronization messages are sent by the master with a constant period $T_{sync} = 2$ s, while delay-request messages are delivered by the slave every 10 s. Network behavior has been considered by modeling the propagation delay d_{MS} affecting messages traveling from the master to the slave and similarly the propagation delay d_{SM} related to the inverse path between slave and master. In particular, $d = c + \mathcal{E}(\lambda)$ where c is a constant value and $\mathcal{E}(\lambda)$ is an exponentially distributed

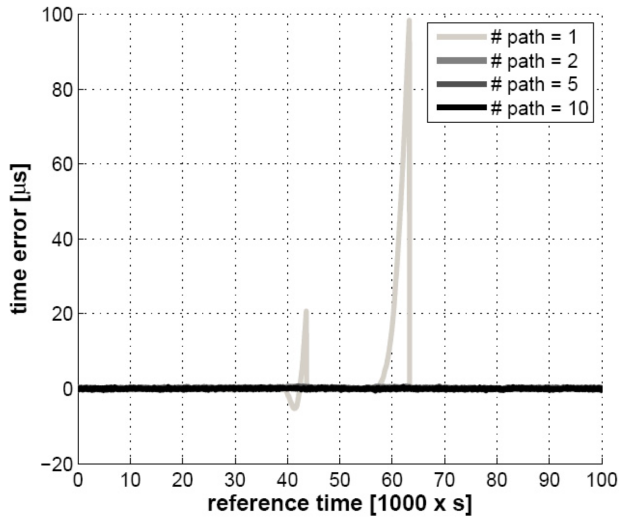

Fig. 6. Time error between local clock and time reference when a different number of independent paths are considered.

random variable with parameter λ [3]. Results reported in the following have been obtained assuming $c_{MS} = c_{SM} = 2$ μs and $\lambda^{-1} = 200$ ns under normal working conditions, while $c_{MS} \in \mathcal{U}(-300, +300)$ μs and $\lambda^{-1} = 4$ μs during network congestion. Burst duration may vary between a minimum of 30 minutes and a maximum of 2 hours, and occurrences are uniformly distributed within a day duration. Traffic on different paths is always supposed to be uncorrelated.

To track network variability, the measurement noise matrix $\mathbf{R}$ needs to be continuously updated. A very simple algorithm for estimating the variability of different propagation delay is assumed:

$$\hat{\sigma}_{d_{prop}(j)}(k) = \beta \cdot \hat{\sigma}_{d_{prop}(j)}(k-1) + (1-\beta) \cdot \sigma_{raw} \quad (19)$$

where $\sigma_{raw} = |d_{prop}(k, j) - \hat{\mu}[d_{prop}(j)]|$, as introduced in Eq. (6), $\hat{\mu}[d_{prop}(j)]$ is the estimated mean value of the propagation delay over the j-th path and β is a smoothing constant that can assume values between 0 and 1. Results have been obtained with $\beta = 0.6$.

A new estimate is therefore calculated each time a new synchronization message is received and it is used for updating the measurement matrix of the Kalman filter. In this way, KF gains are dynamically calculated and, as shown in Fig. 5, when measurements become worthless, their influence is reduced.

This approach is similar to the dynamic combining algorithm introduced in Sec. III which, however, is memoryless. The Kalman filter introduces instead a memory effect and previously estimated offset values are also considered to obtain a more accurate estimate.

Results have been reported in Fig. 6. This example shows that the use of a multipath protocol can effectively reduce the influence of large bursts of data in the network. Furthermore, the number of independent paths that need to be considered is not too large. In both examples we can see that significant improvements can be achieved by considering 5 paths instead of only a single path, while, on the contrary, adding other paths does not result in significantly better performances.

VII. Conclusion

A KF-based clock servo can be adapted for multi-path synchronization in a rather straightforward way. It has been shown in this paper that the algorithm allows adaptive combination of time information, based on the associated measurement uncertainty. Analysis of results using an established and well understood clock model evidences that the protocol and algorithm costs of redundancy can be repaid in terms of increased synchronization robustness.

Although comparatively large numbers of paths can be considered in simulation studies, it is clear from these results that a useful trade-off can already be obtained by the use of a realistically small set of alternative paths. It is also important to remember, however, that no significant improvement can be obtained when the intrinsic stability of a local clock is poor.

References

[1] A. Shpiner, Y. Revah, T. Mizrahi, "Multi-Path Time Protocols", Proc. 2013 Int. IEEE Symp. on Precision Clock Synchronization for Measurement Control and Communication (ISPCS), Lemgo, Germany, 25-27 Sept. 2013, pp. 1-6.

[2] D. Mills, U. Delaware, J. Martin, J. Burbank, W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", IETF Network Working Group, RFC 5905, June 2010.

[3] T. Mizrahi, "Slave diversity: Using multiple paths to improve the accuracy of clock synchronization protocols", Proc. 2012 Internat. IEEE Symp. on Precision Clock Synchronization for Measurement Control and Communication (ISPCS), San Francisco, CA, USA, 26-28 Sept. 2012, pp. 1-6.

[4] P. Ferrari, A. Flammini, S. Rinaldi, G. Prytz, "High availability IEEE 1588 nodes over IEEE 802.1aq shortest path bridging networks", Proc. 2013 Int. IEEE Symp. on Precision Clock Synchronization for Measurement Control and Communication (ISPCS), Lemgo, Germany, 25-27 Sept. 2013, pp. 35-40.

[5] C. A. Greenhall, "Forming stable timescales from the Jones-Tryon Kalman filter", Metrologia, vol. 40, no. 3, pp. S335-S341, June 2003.

[6] L. Galleani, P. Tavella, "Time and the Kalman filter", IEEE Control Systems Magazine, vol. , no. , pp. 44-65, April 2010.

[7] 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.

[8] M. Bertocco, G. Giorgi, C. Narduzzi, "Innovation-based Timestamp Validation for Reliable Sensor Network Synchronization", Proc. IEEE Internat. Instrum. Meas. Technol. Conf., I2MTC 2013, Minneapolis MN, 6-9 May 2013, pp. 784-789.

[9] G. Giorgi, C. Narduzzi, "A resilient Kalman-filter based servo clock", Proc. 2013 Internat. IEEE Symp. on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS 2013), Lemgo, Germany, 25-27 September 2013, pp. 59-64.

[10] A. Shpiner, R. Tse, C. Schelp, T. Mizrahi, "Multi-Path Time Synchronization", IETF Internet-Draft, Feb. 2014.

[11] International Standard IEEE1588, "Precision clock synchronization protocol for networked measurement and control systems", IEEE Std. 1588:2008.

[12] G. Giorgi, "An event-based Kalman filter for clock synchronization", Accepted for publication in *IEEE Trans. Instrum. Meas.*

[13] P. Ferrari, G. Giorgi, C. Narduzzi, S. Rinaldi, M. Rizzi, "Timestamp Validation Strategy for Wireless Sensor Networks based on IEEE 802.15.4 CSS", Accepted for publication in Trans. on Instrum. and Meas.