Fast Computation of Maximum Time Interval Error by Binary Decomposition

Stefano Bregni, *Senior Member, IEEE,* and Stefano Maccabruni

***Abstract*—Maximum time interval error (MTIE) is historically one of the main time-domain quantities for the specification of clock stability requirements in telecommunications standards. Nevertheless, plain computation of the MTIE standard estimator proves cumbersome in most cases of practical interest, due to its heavy computational weight. In this paper, MTIE is first introduced according to its standard definition. Then, a fast algorithm based on binary decomposition to compute the MTIE standard estimator is described. The computational weight of the binary decomposition algorithm is compared to that of the estimator plain calculation, showing that the number of operations needed is reduced to a term proportional to $N \log_2 N$ instead of N^2. A heavy computational saving is therefore achieved, thus making feasible MTIE evaluation based on even long sequences of time error (TE) samples.**

***Index Terms*—Clocks, digital communication, jitter, SONET, synchronization, synchronous digital hierarchy, time domain measurements.**

I. Introduction

A major topic of discussion in standard bodies dealing with network synchronization [1]–[4] is clock noise characterization and measurement. Among the quantities considered in international standards for specification of phase and frequency stability requirements, the maximum time interval error (MTIE) has played historically a major role for characterizing time and frequency performance in digital telecommunications networks [5]–[12]. Specifications in terms of MTIE are well suited to support the design of equipment buffer size.

In this paper, MTIE is first introduced according to its formal definition. Then, the main issue of its experimental measurement is pointed out: the heavy computational weight in most cases of practical interest, due to the number of operations nested in the direct, plain calculation of the MTIE standard estimator. Therefore, a fast algorithm to compute the MTIE standard estimator is described, thus making feasible MTIE evaluation based on even long sequences of Time Error (TE) samples. Finally, the computational weight of this fast algorithm is compared to that of the estimator plain calculation.

Manuscript received May 20, 1999; revised August 1, 2000. This paper is largely based on the paper "Fast Computation of Maximum Time Interval by Binary Decomposition," S. Bregni and S. Maccabruni, *Proc. IEEE ICC'99,* Vancouver, BC, Canada, June 1999.

S. Bregni is with the Department of Electronics and Computer Science, Politecnico di Milano, 20133 Milano, Italy (e-mail: bregni@elet.polimi.it).

S. Maccabruni was with the Department of Electronics and Computer Science, Politecnico di Milano, 20133 Milano, Italy. He is now with PRIDE, Milano, Italy.

Publisher Item Identifier S 0018-9456(00)09786-2.

II. Definition of MTIE

A thorough treatment of MTIE and of its properties can be found in [13]. A further specific analysis is reported in [14]. Here, solely the main definitions are summarized for the sake of understanding and to provide the reader with the background concepts.

A general expression describing a pseudo-periodic waveform which models the timing signal $s(t)$ at the clock output is given by [15]–[18]

$$s(t) = A \sin \Phi(t) \tag{1}$$

where A is the peak amplitude and $\Phi(t)$ is the *total instantaneous phase,* expressing the ideal linear phase increasing with t and any frequency drift or random phase fluctuation.

The generated *Time* function $\mathrm{T}(t)$ of a clock is defined, in terms of its total instantaneous phase, as

$$\mathrm{T}(t) = \frac{\Phi(t)}{2\pi\nu_{\mathrm{nom}}} \tag{2}$$

where ν_{nom} represents the oscillator nominal frequency. It is worthwhile noticing that for an ideal clock $\mathrm{T}_{\mathrm{id}}(t) = t$ holds, as expected. For a given clock, the time error function $\mathrm{TE}(t)$ [in standards also called $x(t)$] between its time $\mathrm{T}(t)$ and a reference time $\mathrm{T}_{ref}(t)$ is defined as

$$x(t) \equiv \mathrm{TE}(t) = \mathrm{T}(t) - \mathrm{T}_{\mathrm{ref}}(t). \tag{3}$$

The *Maximum Time Interval Error* function (MTIE) (τ, T) is the maximum peak-to-peak variation of TE in all the possible observation intervals τ (in former standards [5] and [6], denoted as S) within a measurement period T (see Fig. 1) and is defined as

$$\mathrm{MTIE}(\tau, T) = \max_{0 \le t_0 \le T-\tau} \left\{ \max_{t_0 \le t \le t_0+\tau} [\mathrm{TE}(t)] - \min_{t_0 \le t \le t_0+\tau} [\mathrm{TE}(t)] \right\}. \tag{4}$$

It should be noted, however, that the standards in force specify the MTIE limits simply as a function of τ (or S), thus implicitly assuming

$$\mathrm{MTIE}(\tau) = \lim_{T\to\infty} \mathrm{MTIE}(\tau, T). \tag{5}$$

III. Measuring MTIE

MTIE measurement is usually based on the time-domain measurement of the TE process $x(t)$ between the output of the clock under test (CUT) and a reference timing signal, which may be its input if the CUT is a slave clock (*synchronized clocks*

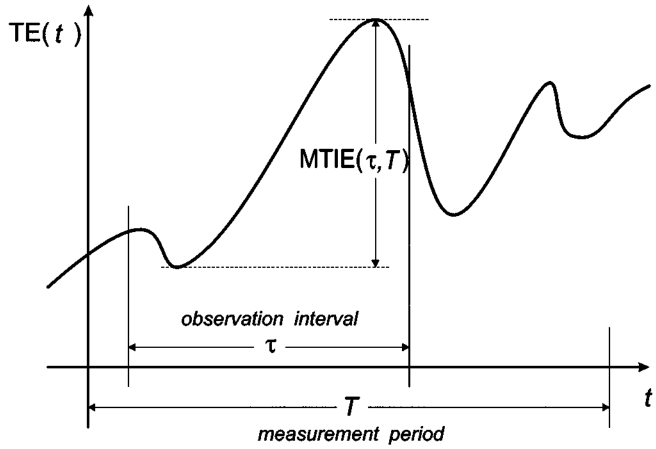

Fig. 1. Definition of MTIE(τ, T).

configuration), or the output of a second reference clock if the CUT is a free-running clock (*independent clocks configuration*) [7], [11]. Sequences of TE samples $\{x_i\}$, defined as

$$x_i = x(t_0 + (i-1)\tau_0) \qquad i = 1, 2, 3, \cdots \tag{6}$$

where t_0 is the initial observation time and τ_0 is the sampling period, are measured using digital counters and stored for numerical post-processing over a total measurement period T [12], [13]. The samples x_i are typically measured between two corresponding zero-crossings of the timing signals involved.

Starting from the sequence $\{x_i\}$ of TE samples measured, the definition (4) may be applied directly to compute MTIE(τ, T). Letting $N_T = T/\tau_0 + 1$ be the total number of available samples and $N_\tau = \tau/\tau_0 + 1$ be the number of samples available in a window (observation interval) of span τ, for *each* single value MTIE(τ, T) the following expression has to be computed:

$$\mathrm{MTIE}(\tau, T) = \max_{j=1}^{N_T - N_\tau + 1} \left[\max_{i=j}^{N_\tau + j - 1} (x_i) - \min_{i=j}^{N_\tau + j - 1} (x_i) \right]. \tag{7}$$

The above is the MTIE standard estimator recommended in [7], [11].

MTIE masks currently specified in standards span over a wide range of τ: four decades, namely, from 10^{-1} s up to 10^3 s. For a long time this range was even wider, from a few milliseconds up to 10^5 s. Furthermore, more specific studies may require investigation over different wide ranges.

As pointed out in [13], the number of samples N_T to process gets easily to the order of 10^5 in most cases of practical interest, if we are interested in a somehow accurate characterization of the clock noise. It obviously appears that the plain computation of the estimator (7) is unadvisable and quickly tends to be unmanageable, due to the number of operations nested in evaluation loops. Hence it comes the need of contriving a suitable algorithm effective in cutting down the computational weight of a plain implementation of the estimator (7).

IV. MTIE Computation by Binary Decomposition

The fast algorithm proposed in this paper is based on a binary decomposition of a TE sequence $\{x_i\}$ made of $N_T = 2^{k_{\mathrm{MAX}}}$ samples in nested windows made of $N_\tau = 2^k$ samples ($k = 1, 2, 3, \cdots, k_{\mathrm{MAX}}$). MTIE can be then evaluated recursively for each window size 2^k.

As the first step ($k = 1$), all the possible 2-points windows ($\tau = \tau_0$) are analyzed in the TE sequence: for each of them, the maximum and minimum values are stored. Their difference is the MTIE(τ_0) measured in that window, and the maximum of the MTIE values of all the 2-points windows is the resulting MTIE(τ_0, T) of the whole sequence. At this first step, there is no computational saving yet compared to the plain computation of the standard estimator.

Then, as second step ($k = 2$), all the possible 4-points windows ($\tau = 3\tau_0$) are considered. The maximum and minimum values of each of these windows can be obtained by comparing the maximum and minimum values of the two 2-points windows in which the 4-points window can be split. The difference between the maximum of the two maxima and the minimum of the two minima is the MTIE($3\tau_0$) measured in that 4-point window. The maximum of the MTIE values of all the 4-point windows is the resulting MTIE($3\tau_0$, T) of the whole sequence.

The next step ($k = 3$) is to consider all the possible 8-points windows ($\tau = 7\tau_0$), split in two 4-points windows. Then so on for increasing integer values of k. The computational saving of this algorithm, compared to the plain computation of the standard estimator, lies in avoiding the comparison of all the samples in the windows of size larger then 2. The price to pay is that we have to limit the evaluation of MTIE(τ, T) just to the $\log_2 N_T$ values corresponding to the windows made of $N_\tau = 2^k$ samples (this corresponds to a bit more than three MTIE values per decade on the τ axis, which may be considered sufficient in most practical applications).

More formally, starting from the TE sequence vector **x** made of $N_T = 2^{k_{\mathrm{MAX}}}$ TE samples x_i, two matrices $\mathbf{A}_{\mathrm{M}}$ and $\mathbf{A}_{\mathrm{m}}$ are built. Matrices are made of $N_T - 1$ columns (indexed by i) and $\log_2 N_T$ rows, indexed by k. The first $N_T - 2^k + 1$ elements of each kth row of the matrix $\mathbf{A}_{\mathrm{M}}$ contain the maximum values of all the possible 2^k-points windows sliding from left to right along the TE sequence $\{x_i\}$. The matrix $\mathbf{A}_{\mathrm{m}}$ contains, in an analogous fashion, the corresponding minimum values of the 2^k-points windows. Therefore, the set of all the possible 2^k-points windows in the whole TE sequence is completely described by the couple of vectors

$$\begin{aligned} \mathbf{a}_{\mathrm{M}/k} &= \{a_{\mathrm{M}/k,\,i}\} \\ \mathbf{a}_{\mathrm{m}/k} &= \{a_{\mathrm{m}/k,\,i}\} \end{aligned} \quad i = 1, 2, \cdots, N_T - 2^k + 1 \tag{8}$$

where $\mathbf{a}_{\mathrm{M}/k}$ and $\mathbf{a}_{\mathrm{m}/k}$ are the kth rows taken from the matrices $\mathbf{A}_{\mathrm{M}}$ and $\mathbf{A}_{\mathrm{m}}$, respectively.

The first row ($k = 1$) of matrices $\mathbf{A}_{\mathrm{M}}$ and $\mathbf{A}_{\mathrm{m}}$ is obtained directly by the TE sequence vector **x** as

$$\begin{aligned} a_{\mathrm{M}/1,\,i} &= \max(x_i, x_{i+1}) \\ a_{\mathrm{M}/1,\,i} &= \min(x_i, x_{i+1}) \end{aligned} \tag{9}$$

for $i = 1, 2, \cdots, N_T - 1$. The next rows ($k > 1$), instead, are obtained recursively as

$$\begin{aligned} a_{M/k,\,i} &= \max(a_{k-1,\,i}, a_{k-1,\,i+p}) \\ a_{m/k,\,i} &= \min(a_{k-1,\,i}, a_{k-1,\,i+p}) \end{aligned} \tag{10}$$

where $p = 2^{k-1}$, for $i = 1, 2, \cdots, N_T - 2^k + 1$.

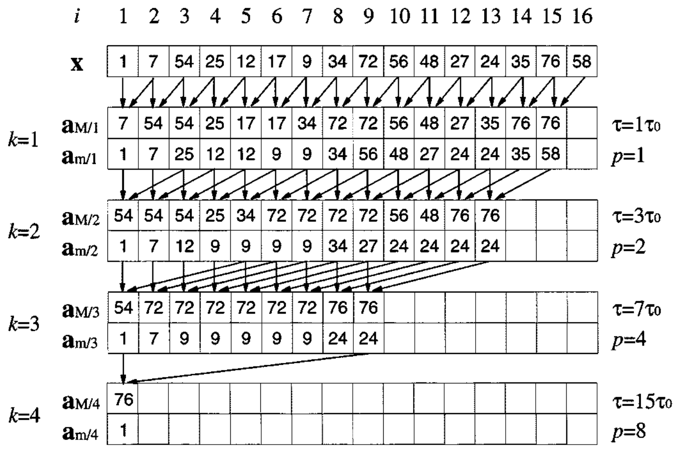

Fig. 2. Example of execution of the binary decomposition algorithm ($N_T = 16$).

Finally, the value MTIE(τ, T) for $\tau = (N_\tau - 1)\tau_0$ and $N_\tau = 2^k$ (here denoted as MTIE$_k$ for the sake of brevity) can be evaluated from the kth rows of the matrices $\mathbf{A}_{\rm M}$ and $\mathbf{A}_{\rm m}$ as

$$\text{MTIE}_k = \max_{i=1,\,\cdots,\,N_T-2^k+1} (a_{{\rm M}/k,\,i} - a_{{\rm m}/k,\,i}). \tag{11}$$

An example of the binary decomposition tree, applied on a TE sequence $\{x_i\}$ made of $N_T = 16$ samples ($k_{\rm MAX} = 4$), is shown in Fig. 2, which depicts the four couples of vectors $\mathbf{a}_{{\rm M}/k}$ and $\mathbf{a}_{{\rm m}/k}$ (for $k = 1, 2, 3, 4$) built recursively starting from the TE vector $\mathbf{x}$.

V. Computational Saving

The number of operations involved in the estimator plain computation and in the binary decomposition algorithm has been evaluated, in order to assess the resulting computational saving.

A. Plain Computation of the Estimator

As far as a plain computation of the estimator (7) is concerned, three nested loops can be identified:

1) an external loop increasing the observation interval τ, executed one time per each single value MTIE(τ, T) to compute;
2) a first internal loop executed, given τ, for each N_τ-points sliding window [the external $\max[\cdot]$ function in (7)], i.e. $N_T - N_\tau + 1$ times;
3) the most internal loop to find the maximum and minimum value in a set of N_τ samples, thus involving $2(N_\tau - 1)$ comparison test branches and a variable number of assignments according to the particular TE sequence (we neglect here the possibility to use a more efficient algorithm to extract the maximum and minimum values).

If we limit MTIE computation to one value per octave on the τ axis, as in the binary decomposition algorithm, then the first loop is executed $k_{\rm MAX} = \log_2 N_T$ times, the second loop $N_T - 2^k + 1$ times ($k = 1, 2, \cdots, k_{\rm MAX}$) and the third loop involves $2^{k+1} - 2$ branches. Thus, the computational weight results are approximately (from now on, N_T will be denoted simply as N for the sake of brevity):

$$\begin{array}{ll} \frac{4}{3}N^2 + \cdots & \text{comparison test branches} \\ 3(N\log_2 N - 2N) + \cdots & \text{assignments (best case)} \\ \frac{2}{3}N^2 + \cdots & \text{assignments (worst case)} \\ N\log_2 N - 2N + \cdots & \text{additions.} \end{array} \tag{12}$$

Comparison test branches are the most time-consuming operations.

It is worthwhile noticing that MTIE plain computation turned out to be a N^2-problem because we decided to limit MTIE computation to one value per octave on the τ axis. If MTIE is computed for all the possible $N-1$ values of τ, then the number of operations required becomes proportional to N^3 instead.

B. Binary Decomposition Algorithm

As far as the binary decomposition algorithm is concerned, on the other hand, the following loops can be identified:

1) a first loop initializing the first row ($k = 1$) of matrices $\mathbf{A}_{\rm M}$ and $\mathbf{A}_{\rm m}$ and then computing MTIE$_1$, involving in particular $2(N-1)$ comparison test branches;
2) a second main loop increasing the row index $k\,(k > 1)$, executed $\log_2 N - 1$ times;
3) a loop, internal to the previous one, to compute the next rows ($k > 1$) of matrices $\mathbf{A}_{\rm M}$ and $\mathbf{A}_{\rm m}$ and to evaluate the corresponding MTIE$_k$, involving in particular $3(N - 2^k + 1)$ comparison test branches.

Thus, the computational weight results are approximately:

$$\begin{array}{ll} 3N\log_2 N - 7N + \cdots & \text{comparison test branches} \\ 2N + \cdots & \text{assignments (best case)} \\ 3(N\log_2 N - 2N) + \cdots & \text{assignments (worst case)} \\ N\log_2 N - 2N + \cdots & \text{additions.} \end{array} \tag{13}$$

C. Comparison in Terms of Computational Weight

It is worthwhile noticing that, in the binary decomposition algorithm, the number of comparison test branches and worst-case assignments needed has been reduced to a term proportional to $N \log_2 N$ instead of the N^2 involved in the plain computation of the estimator (7).

The graph of Fig. 3 compares, on a logarithmic scale, the number of comparison test branches needed by the two algorithms considered as a function of the total number of available TE samples N, for $2^1 \le N \le 2^{25}$ [to build this graph, all the lower-order terms not shown in (12) and (13) have been taken into account]. Moreover, the ratio between the two numbers (i.e., the computational saving factor) is plotted as well for ease of comparison. It may be noticed that in the most common range $2^{14} \le N \le 2^{19}$ (i.e., $16\,384 \le N \le 524\,288$) the saving factor turns out to be in the remarkable order of 10^3 to 10^4.

Finally, it may be interesting to know how long do both algorithms take to execute with practical values of N on some average/low-power computer (any quantity intended for routine measurements should not require a supercomputer to be computed). To this purpose, both algorithms have been programmed in C language, compiled and run for testing on a SUN Sparc

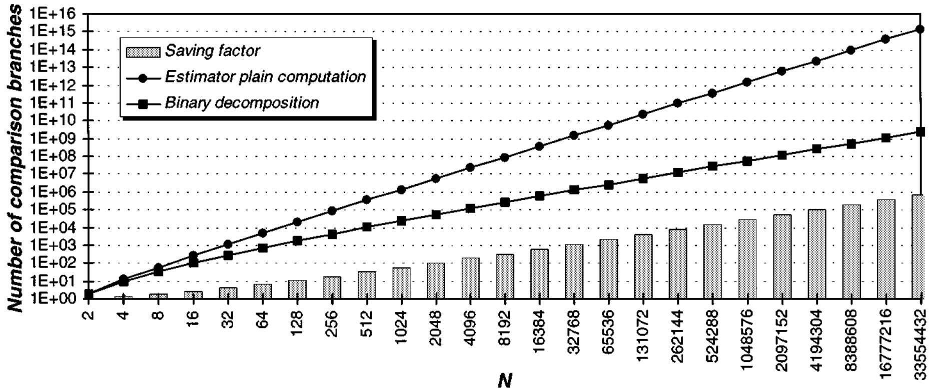

Fig. 3. Number of comparison test branches required by the MTIE estimator plain computation and the binary decomposition algorithm as a function of the total number of TE samples N.

Server 10 under UNIX operating system. The plain computation of the MTIE estimator, one value per octave as described in Section V-A, required about 1100 s of actual execution time (without any other CPU-time consuming processes) on a sequence of $N = 65\,536$ samples. The binary decomposition algorithm, on the same sample sequence, needed just something more than one second to complete execution. These results are in good agreement with the graph of Fig. 3, which for $N = 65\,536$ reports a saving factor in the order of 10^3.

VI. Conclusions

In this paper, a fast algorithm based on binary decomposition to calculate the MTIE standard estimator was proposed. The computational weight of the binary decomposition algorithm was compared to that of the estimator plain computation.

The proposed algorithm proved effective in achieving a strong computational saving, by reducing the number of comparison test branches and worst-case assignments needed to a term proportional to $N \log_2 N$ instead of N^2 (see the graph in Fig. 3). Moreover, as a perhaps unnecessary check, its effectiveness and correctness were confirmed in paper [19] by computing the MTIE of TE sequences generated by simulation of the so-called *power-law* noise, the model most frequently used to represent clock output noise in the frequency domain [16]–[18].

This binary decomposition algorithm allows fast MTIE evaluation in most practical situations: even very long sequences of TE samples do not require more than a few seconds of MTIE computation time. Therefore, it may be conveniently adopted by telecommunications engineers involved in time-domain measurement of clock stability.

Acknowledgment

The authors are first indebted to G. Zampetti with Symmetricom, San José, CA, who in past years was lavish with helpful suggestions on practical MTIE computation. Moreover, the authors thank N. Lazzari (classmate of S. Maccabruni at Politecnico di Milano) who contributed to software developing and G. Gilardi (student at Politecnico di Milano) who nicely helped them with software tests. Finally, the authors thank also Prof. E. Bava of Politecnico di Milano for his support.

References

[1] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann, and K. Dessouky, "Network synchronization," *Proc. IEEE*, vol. 73, pp. 1445–1467, Oct. 1985.

[2] P. Kartaschoff, "Synchronization in digital communications networks," *Proc. IEEE*, vol. 79, pp. 1019–1028, July 1991.

[3] J. C. Bellamy, "Digital network synchronization," *IEEE Commun. Mag.*, vol. 33, pp. 70–83, Apr. 1995.

[4] S. Bregni, "A historical perspective on network synchronization," *IEEE Commun. Mag.*, vol. 36, June 1998.

[5] ITU-T Rec. G.810, *Considerations on timing and synchronization issues, G.811; Timing Requirements at the Outputs of Primary Reference Clocks Suitable for Plesiochronous Operation of International Digital Links, G.812; Timing Requirements at the Outputs of Slave Clocks Suitable for Plesiochronous Operation of International Digital Links*. Geneva, Switzerland: Blue Book, 1988.

[6] *Synchronization Interface Standard*, ANSI T1.101-1994.

[7] ITU-T, Rec. G.810, "Definitions and terminology for synchronization networks,", Geneva, Switzerland, Aug. 1996.

[8] ITU-T, Rec. G.811, "Timing characteristics of primary reference clocks,", Geneva, Switzerland, Sept. 1997.

[9] ITU-T, Rec. G.812, "Timing requirements of slave clocks suitable for use as node clocks in synchronization networks,", Geneva, Switzerland, June 1998.

[10] ITU-T, Rec. G.813, "Timing characteristics of SDH equipment slave clocks (SEC),", Geneva, Switzerland, Aug. 1996.

[11] ETSI, "Transmission and multiplexing (TM); Generic requirements for synchronization networks,", Draft ETS 300 462, pt. 1–6, 1997.

[12] S. Bregni, "Clock stability characterization and measurement in telecommunications," *IEEE Trans Instrum. Meas.*, vol. 46, pp. 1285–1294, Dec. 1997.

[13] ——, "Measurement of maximum time interval error for telecommunications clock stability characterization," *IEEE Trans. Instrum. Meas.*, vol. 45, pp. 900–906, Oct. 1996.

[14] S. Bregni and F. Setti, "Impact of the anti-aliasing pre-filtering on the measurement of maximum time interval error," in *Proc. IEEE GLOBECOM '97*, Phoenix, AZ, Nov. 1997.

[15] J. A. Barnes, A. R. Chi, L. S. Cutler, D. J. Healey, D. B. Leeson, T. E. McGunigal, J. A. Mullen Jr., W. L. Smith, R. L. Sydnor, R. F. C. Vessot, and G. M. R. Winkler, "Characterization of frequency stability," *IEEE Trans. Instrum. Meas.*, vol. IM-20, May 1971.

[16] J. Rutman and F. L. Walls, "Characterization of frequency stability in precision frequency sources," *Proc. IEEE*, vol. 79, pp. 952–960, July 1991.

[17] M. Carbonelli, D. De Seta, and D. Perucchini, "Characterization of timing signals and clocks," *Eur. Trans. Telecommun.*, vol. 7, no. 1, pp. 9–24, Jan./Feb. 1996.

[18] S. Bregni, "Clocks in telecommunications," in *Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, Ed. New York: Wiley, 1999.

[19] S. Bregni and S. Maccabruni, "Fast computation of maximum time interval error by binary decomposition," in *Proc. IEEE ICC '99*, Vancouver, BC, Canada, June 1999.

Stefano Bregni (M'93–SM'99) was born in Milano, Italy, in 1965. He received the Dott. Ing. degree in telecommunications engineering from Politecnico di Milano and the M.Sc. degree from CEFRIEL Center, Milano.

In 1991, he joined SIRTI S.p.A., where he was mainly involved in SDH transmission system testing and in network synchronization issues, with special regard to clock stability measurement techniques. From 1994 to 1999, he was with CEFRIEL, as head of the Transmission Systems Department. Since 1999, he has been Assistant Professor at Politecnico di Milano, Department of Electronics and Computer Science. He is author or co-author of about 20 papers on several topics in this field, mostly published in IEEE conferences and journals. He is the author of the forthcoming book *Synchronization of Digital Communications Networks* (New York: Wiley, 2001).

Dr. Bregni served on ETSI and ITU-T committees on digital network synchronization. He is author of the paper "Clocks in Telecommunications" included in the *Encyclopedia of Electrical and Electronics Engineering* (New York: Wiley, 1999). He gave a full-day tutorial on synchronization of digital networks at IEEE ICC2000. He is Vice-Chairman of the Transmission, Access, and Optical Systems Committee of the IEEE Communications Society.

Stefano Maccabruni was born in Milano, Italy, in 1966. He received the Dott. Ing. degree in electronics engineering from Politecnico di Milano in 1998, with a graduation thesis on the subject of this paper.

Currently, he is a Software Developer.