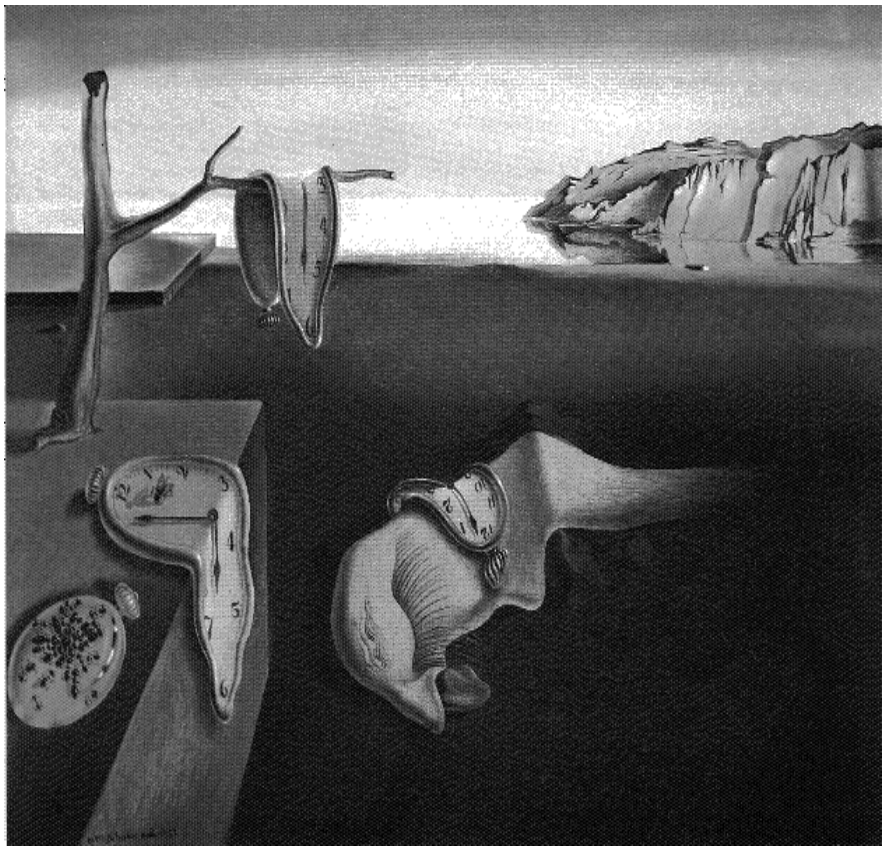
Institut für Automation
Abt. für Automatisierungssysteme

Technische Universität Wien

Projektbericht Nr. 183/1-80
July 1997

Interval-based Clock Synchronization Revisited

Ulrich Schmid and Klaus Schossmaier



Salvador Dali, "Die Beständigkeit der Erinnerung"

Interval-based Clock Synchronization Revisited

Ulrich Schmid, Klaus Schossmaier
Technische Universität Wien
Department of Automation
Treitlstraße 1, A-1040 Vienna
Email: {s, kmschoss}@auto.tuwien.ac.at

August 30, 1999

Abstract

This report[1] provides a few extensions and additions to the generic analysis of interval-based clock synchronization conducted in [SS97]. Apart from adding traditional accuracy and global rate to the major theorem, the accuracy analysis has been modified to circumvent the overly conservative bounds provided by the original framework.

Keywords: interval-based clock synchronization, fault-tolerant distributed real-time systems, convergence functions, precision, accuracy.

1 Introduction

Modern distributed systems usually run applications that rely on a global notion of time. In fact, algorithms for (fault-tolerant) distributed systems are considerably simplified and improved w.r.t. performance when mutually synchronized clocks are available, see [Lis93]. In addition, time rules daily life and hence most commercial computer applications, so system time should also have a well-defined relation to external time like *Universal Time Coordinated* (UTC).

Excluding non-fault tolerant solutions based on centralized clocks, we thus assume that each node p of a distributed system is equipped with a local clock $C_p(t)$ that continuously displays p's view of system time, satisfying

(1) *Precision requirement*: there is some fixed *precision* $\pi \geq 0$ such that $|C_p(t) - C_q(t)| \leq \pi$ for all nodes p, q that are non-faulty up to real-time t.

(2) *Accuracy requirement*: there is some *accuracy* $\alpha \geq 0$ such that $|C_p(t) - t| \leq \alpha$ for all nodes p that are non-faulty up to real-time t.

[1]This research is part of our project SynUTC (http://www.auto.tuwien.ac.at/Projects/SynUTC/), which has been supported by the Austrian Science Foundation (FWF) under grant no. P10244-ÖMA and the START programme Y41-MAT.

Note that we restrict our attention to $\alpha_p^+, \alpha_p^- = \mathcal{O}(t)$, with an implied constant[2] M in $\mathcal{O}$ being (much) smaller than 1; this excludes "degenerated" solutions like $C_p(t) \equiv 0$ since $C_p(t)$ must be within a linear envelope of real-time, cf. [DHS86].

Providing mutually synchronized local clocks is known as the internal synchronization problem, and numerous solutions have been worked out (at least in scientific research) under the term fault-tolerant clock synchronization, cf. [SLWL90], [RSB90] for an overview and [YM93] for a bibliography. Establishing an *accurate* global time in distributed systems is usually termed the external synchronization problem, due to the fact that UTC is provided externally to the system. A comprehensive collection of papers describing recent efforts in this area may be found in [Sch97c].

Among those is our research on *clock validation* introduced in [Sch95], which solves the external synchronization problem by verifying whether the highly accurate but possibly faulty "authoritative time" provided by UTC time sources is consistent with the low-accurate but reliable "validation time" formed by exchanging the information of the local (quartz) clocks in the system. If so, the authoritative time is accepted — if not, it is discarded and the nodes rely on the validation time instead. The latter situation, however, is encountered permanently during phases of total unavailability of accurate time information from UTC time sources. In this *flywheel operation*, clock validation obviously "degenerates" to internal synchronization. Consequently, the clock synchronization algorithm employed for computing the validation time must not only ensure precision but has to maintain high accuracy as well.

Traditional internal synchronization techniques turn out to be ill-suited for coping with this requirement. In fact, although worst case accuracy bounds have been computed for most existing synchronization algorithms, it is nevertheless true that a static worst case bound is not representative for the "average" execution. A promising alternative are interval-based algorithms introduced in Marzullo's thesis [Mar84] and further exploited in [Lam87], [Mar90], [OSF92], [Sch95], [Mil95], [BI96], [SS97], where local time information relating to real-time t is represented by an interval that is supposed to contain t. Given a set of such intervals from different nodes, a usually smaller interval that contains UTC can be determined by means of a suitable interval-valued convergence function. Since accuracy bounds are hence maintained dynamically ("on-line") here, they are of course representative for the "average" execution.

In [SS97], we presented and analyzed a generic interval-based clock synchronization algorithm suitable for computing the validation time in our clock validation framework. According to our exposition above, it maintains bounded precision when started from an initially synchronized state and takes care of accuracy intervals as well. The algorithm is generic in the sense that it left unspecified the convergence function employed for computing the clock adjustments. As in [Sch86], all results (worst-case precision, accuracy, maximum adjustment, etc.) are hence expressed in terms of a few characteristic parameters of the convergence function. In order to determine the performance of a particular instance of the algorithm, all that needs to be done is to determine the characteristic parameters of the particular convergence function and to plug it into the generic results, see e.g. [Sch97b], [Sch97a].

[2]Throughout this paper, we use the $\mathcal{O}(.)$-notation to characterize the order of magnitude of less important terms. For example, $\alpha_p^- = \mathcal{O}(t)$ means that there is some (reasonably small) constant $M > 0$ such that $|\alpha_p^-| \leq M|t|$.

This paper revisits and extends some part of the analysis in [SS97]. Apart from adding traditional accuracy and global rate to the major theorem, the accuracy analysis has been modified to circumvent the overly conservative bounds provided by the original framework. The remaining sections of our exposition are organized as follows: Section 2 briefly recollects the system model underlying our approach, Section 3 explains the generic interval-based clock synchronization algorithm. The major Section 4 outlines the overall analysis and provides the required modifications of the generic convergence function's Definition 8 and the major Theorem 1. A few concluding remarks in Section 5 eventually complete the paper.

2 System Assumptions

In this section, we will provide a brief recollection of the system model and its parameters. Similar models are well-known from the analysis of internal clock synchronization and other distributed algorithms for fault-tolerant systems. However, non-standard features are incorporated in the clock model, which is interval-based and deals with non-zero clock granularity, and in the model of the communication subsystem, which contains parameters for broadcast latencies and limited transmission bandwidth. See [SS97] for a detailed discussion of the assumptions stated below.

We consider a distributed system consisting of n *nodes*, which communicate with each other by message passing over a suitable communication network. Each node is equipped with a *processor* (with integer arithmetic only) that executes the clock synchronization algorithm, an adjustable *local clock*, and a *network interface*.

Assumption 1 (Execution Times) *A single (periodic) computation required for clock synchronization purposes at a non-faulty node p is completed within γ_p (real-time) seconds. Let $\gamma_{\max} \geq \gamma_p$, $\gamma_{\min} \leq \gamma_p$ be suitable uniform bounds on the execution time of all (non-faulty) nodes p.*

Assumption 2 (Local Clocks) *Each node p is endowed with a discrete* local clock $C_p(t)$ *of granularity $G > 0$, which allows state adjustment with a clock setting granularity $G_S = G/K$, for some integer $K \geq 1$. Any non-faulty node p's clock is characterized by its intrinsic inverse rate deviation bound $\boldsymbol{\rho}_p = [-\rho_p^-, \rho_p^+]$ and its maximum rate adjustment uncertainty*[3] *$\boldsymbol{u}_p = [-u_p^-, u_p^+]$, which are such that, in the absence of resynchronizations,*

$$(1 - \rho_p^-)(\Theta_i - \Theta_1) - u_p^- \leq \theta_i - \theta_1 \leq (1 + \rho_p^+)(\Theta_i - \Theta_1) + u_p^+ \tag{1}$$

with $\Theta_j = C_p(\theta_j)$ is guaranteed for any sequence of $i \geq 2$ successive clock ticks occuring at real-time θ_j, $1 \leq j \leq i$. Let $\boldsymbol{\rho}_{\max} = [-\rho_{\max}^-, \rho_{\max}^+] \supseteq \bigcup_p \boldsymbol{\rho}_p$ with $\rho_{\max} = \rho_{\max}^- + \rho_{\max}^+$ and $\boldsymbol{u}_{\max} = [-u_{\max}^-, u_{\max}^+] \supseteq \bigcup_p \boldsymbol{u}_p$ with $u_{\max} = u_{\max}^- + u_{\max}^+ = \mathcal{O}(G)$ be suitable uniform bounds for all (non-faulty) nodes p.

Note that a point in real-time θ is called *in synchrony* with node p's clock iff a clock tick takes place at θ; otherwise, it is called *arbitrary*.

[3] Note that there is a mistake in the statement of $\boldsymbol{u}_p$ for adder-based clocks in [SS97, p. 185], which should read $\boldsymbol{u}_p = [-G_{osc}^0 + \mathcal{O}(G\rho_{osc}), G_{osc}^0 + \mathcal{O}(G\rho_{osc})]$ valid for reasonable clock rate corrections $\delta G_s = \mathcal{O}(\rho_{osc})$.

Assumption 3 (Local Interval Clocks) *Each node p provides a* local interval clock

$$\boldsymbol{C}_p(t) = \begin{cases} [C_p(t) - \alpha_p^-(t), C_p(t) + \alpha_p^+(t)] & \text{if } t = \theta_i \text{ is in synchrony with } C_p(t), \\ [C_p(\theta_i) - \alpha_p^-(\theta_i), C_p(\theta_i) + \alpha_p^+(\theta_i)] & \text{for } \theta_i \leq t < \theta_{i+1} \text{ otherwise} \end{cases}$$

by maintaining an interval of accuracies $\boldsymbol{\alpha}_p(t) = [-\alpha_p^-(t), \alpha_p^+(t)]$ of granularity G_S, which is taken relatively to the node's local clock $C_p(t)$. It is controllable by means of the following operations:

- (Re-)Initializing $\boldsymbol{C}_p(\theta^R)$: *$C_p$ along with $\boldsymbol{\alpha}_p$ can be set atomically (all values, including reference point, being integer multiples of G_S) at any synchronous real-time θ^R.*
- Reading $\boldsymbol{C}_p(t)$: *C_p along with $\boldsymbol{\alpha}_p$ can be read atomically at any real-time t.*
- Deteriorating $\boldsymbol{C}_p(\theta)$: *$\boldsymbol{\alpha}_p$ is enlarged by $G\boldsymbol{\rho}_p$ at each clock tick θ of C_p. Moreover, at the first tick after (re-)initializing $\boldsymbol{C}_p$ (or at θ^R if the new reference point is not an integer multiple of G), an additional $\boldsymbol{u}_p$ is added to incorporate the rate adjustment uncertainty.*

Assumption 4 (Transmission Characteristics) *The system employs a synchronous message-based communication network with the following properties:*

(1) If a non-faulty node p of the distributed system initiates the broadcast of a clock synchronization message (CSM) at some real-time t_p^I, there is a uniform bound $\lambda_{\max} \geq 0$ on the possible delay up to time t_p^S when it actually starts the broadcast transmission; $\lambda_{\max}$ is called the maximum broadcast latency.

(2) If the broadcast of a non-faulty node p starts at real-time t_p^S, there is a bound $\omega_p \geq t_{pl}^A - t_p^S \geq 0$ on the delay up to time t_{pl}^A when the transmission to the last node l required for broadcasting is activated. Let $\omega_{\max} \geq \omega_p$ be a suitable uniform bound called the maximum broadcast operation delay. *Moreover, let*

$$B = \begin{cases} 1 & \text{if } \omega_{\max} = 0 \text{ (pure broadcast network)}, \\ 2 & \text{otherwise}, \end{cases} \tag{2}$$

be the "indicator" of presence of a pure broadcast network.

(3) If a node p activates its transmission to some node $q \neq p$ at time t_{pq}^A and no fault occurs, node q receives the message at time $t_q^p = t_{pq}^A + \delta'_{pq}$. The transmission delay *δ'_{pq} is assumed to satisfy*

$$\delta'_{pq} \in [\delta_{pq} - \varepsilon_{pq}^-, \delta_{pq} + \varepsilon_{pq}^+], \tag{3}$$

where $\delta_{pq} \geq \varepsilon_{pq}^- \geq 0$ represents the deterministic part and $\boldsymbol{\varepsilon}_{pq} = [-\varepsilon_{pq}^-, \varepsilon_{pq}^+]$ the maximum uncertainty of δ'_{pq}. Let $\boldsymbol{\varepsilon}_{\max} = [-\varepsilon_{\max}^-, \varepsilon_{\max}^+] \supseteq \bigcup_{p,q \neq p} \boldsymbol{\varepsilon}_{pq}$ with $\varepsilon_{\max} = \varepsilon_{\max}^- + \varepsilon_{\max}^+$ and $\delta_{\max} \geq \delta_{pq}$, $\delta_{\min} \leq \delta_{pq}$ be suitable uniform bounds for all (non-faulty) pairs of nodes p, $q \neq p$, with the additional technical condition

$$\delta_{\min}\boldsymbol{\rho}_{\max} \subseteq \boldsymbol{\varepsilon}_{\max}. \tag{4}$$

Note that this condition expresses the quite reasonable assumption that timekeeping by any non-faulty clock is more accurate than exploiting the synchronous network behavior.

(4) The accuracy interval in a CSM is transmitted with limited resolution. More specifically, it is assumed that the clock value $C_p(t_{pq}^A)$ is transmitted in a way that preserves its granularity G. Both lower and upper accuracy $\alpha_p^-(t_{pq}^A)$ and $\alpha_p^+(t_{pq}^A)$, however, are transmitted with finite resolution $R_A = LG_S$ for some integer $L \geq 1$ only; let

$$G_A = R_A - G_S \tag{5}$$

be the resulting loss in resolution.

Note that it is assumed that a CSM is "timestamped" with $\boldsymbol{C}_p(t_{pq}^A)$ at the moment of actual transmission t_{pq}^A, not at the moment of initiation t_p^I of the broadcast or at the moment t_p^S of actually starting it. This ensures that the relatively large maximum broadcast latency $\lambda_{\max}$ and/or the maximum broadcast operation delay $\omega_{\max}$ does not impair δ'_{pq} (and hence achievable precision and accuracy). Therefore, the model is suitable for both the extended capabilities provided by our UTCSU-ASIC (see [SSHL97] for details) and for traditional settings ($\omega_{\max} = \lambda_{\max} = 0$ and putting any uncertainty into ε_{pq}).

All assumptions above are only meaningful for those parts of the system that are fault-free. Dealing with fault-tolerant systems, a fault model is of course required, which specifies the allowed behavior of faulty system components. However, the generic algorithm itself puts only a few limitations on the severity of faults. Apart from the —usually more restrictive— additional requirements imposed by the particular convergence function employed (and hence listed there, see e.g. [Sch97b]), faulty nodes or network components may perform arbitrarily, including transmission and "reception" of any number of arbitrary messages without, however, being capable of causing "global" disturbance of system operation, e.g. by

- impersonating other nodes,
- flooding/jamming the network or non-faulty receiving nodes (violating $\lambda_{\max}$ and/or $\omega_{\max}$, or causing excessive transmission delays of other broadcasts).

Note that this is easily guaranteed in a fully-connected point-to-point network, but is difficult to ensure for a (non-redundant) broadcast channel.

We find it appropriate to conclude this section with a short table that gives a flavour of the order of magnitude of the various system parameters introduced so far. More specifically, the class of distributed systems that can be built atop of our *Network Time Interface* NTI (see [HSS98] for details) scores as follows:

Param.	**Value**		**Meaning**
n	100		number of nodes
γ_p	10	ms	overall execution time
G	100	ns	clock granularity
G_S	10^{-15}	s	clock setting granularity
ρ_p^+, ρ_p^-	10^{-6}		inverse rate deviation bound
u_p^+, u_p^-	100	ns	rate adjustment uncertainity
$\lambda_{\max}$	1	s	maximum broadcast latency
$\omega_{\max}$	0	s	maximum broadcast operation delay
δ_{pq}	10	μs	deterministic part transmission delay
ε_{pq}^-, ε_{pq}^+	1	μs	transmission delay uncertainity
G_A	100	ns	accuracy transmission loss

3 Generic State Synchronization Algorithm

In this section, we will provide a brief survey of the generic interval-based algorithm for clock state synchronization developed in [SS97]. We start with the basic notation: Bold letters like $\boldsymbol{I}$ are used to denote a real interval $\boldsymbol{I} = [x, y]$, $x \leq y$, with *lower edge* $x = \text{left}(\boldsymbol{I})$ and *upper edge* $y = \text{right}(\boldsymbol{I})$; the *empty interval* $\emptyset$ satisfies $\nexists t : t \in \emptyset$. A *set* of intervals is denoted by a calligraphic bold letter like $\boldsymbol{\mathcal{I}}$. For an interval $\boldsymbol{I} = [x, y]$, $|\boldsymbol{I}| = y - x$ denotes its *length* and $\text{center}(\boldsymbol{I}) = (x + y)/2$ its *centerpoint*. The *sum* of two intervals is defined as $[x, y] + [u, v] = [x + u, y + v]$, the *scalar product* by $s \cdot [x, y] = [sx, sy]$ for $s \geq 0$, and $\boldsymbol{I} + a = \boldsymbol{I} + [a, a] = [x + a, y + a]$ for some arbitrary scalar a. For two intervals $[x, y]$, $[u, v]$, the *intersection* reads $[x, y] \cap [u, v] = [\max(x, u), \min(y, v)]$ if $u \leq y$ and $v \geq x$, or $\emptyset$ otherwise, and the *union* is $[x, y] \cup [u, v] = [\min\{x, u\}, \max\{y, v\}]$. Note that this definition of the union is also valid for $[x, y] \cap [u, v] = \emptyset$, hence incorporates the closure of two disjoint intervals as well.

The core idea of the interval-based paradigm is to represent local time by a time-dependent *accuracy interval*. More specifically, a non-faulty accuracy interval $\boldsymbol{A} = \boldsymbol{A}(t)$ representing real-time t is an interval that satisfies $t \in \boldsymbol{A}$. Accuracy intervals are primarily provided by *interval clocks*, which are interval-valued functions $\boldsymbol{C}(t) = [L(t), U(t)]$ with the edges $L(t)$ resp. $U(t)$ forming the lower resp. upper "envelope" of real-time t. We have the following relations for accuracy intervals:

Definition 1 (Interval Relations [SS97, Def. 1]) *Accuracy intervals are categorized as follows:*

(1) Two accuracy intervals $\boldsymbol{I} = \boldsymbol{I}(t_1)$ and $\boldsymbol{J} = \boldsymbol{J}(t_2)$ are compatible iff they represent the same real-time $t_1 = t_2 = t$.

(2) Two compatible accuracy intervals $\boldsymbol{I}$ and $\boldsymbol{J}$ are consistent iff $\boldsymbol{I} \cap \boldsymbol{J} \neq \emptyset$.

(3) An accuracy interval $\boldsymbol{I} = \boldsymbol{I}(t)$ representing real-time t is accurate iff $t \in \boldsymbol{I}$.

In practice, interval clocks $\boldsymbol{C}(t) = [C(t) - \alpha^-(t), C(t) + \alpha^+(t)]$ are implemented as an ordinary clock $C(t)$ in conjunction with a time-dependent *interval of accuracies* $\boldsymbol{\alpha}(t) = [-\alpha^-(t), \alpha^+(t)]$ taken relatively to the clock's value. Most intervals $\boldsymbol{A} = [l, u]$ arising in our algorithms are interval clock readings[4] and hence not exhaustively specified by lower and upper edge l and u, but contain a distinguished *reference point* r as well, which partitions the interval in a *negative accuracy* α^- and a *positive accuracy* α^+:

$$\boldsymbol{I} = [l, u] = [r \pm \boldsymbol{\alpha}] = [r - \alpha^-, r + \alpha^+] \tag{6}$$

When referring to an interval $[r \pm \boldsymbol{\alpha}]$, $\boldsymbol{\alpha} = [-\alpha^-, \alpha^+]$ denotes its interval of accuracies and $\alpha = |\boldsymbol{\alpha}| = \alpha^+ + \alpha^-$ its length. For intervals $\boldsymbol{I} = [r \pm \boldsymbol{\alpha}]$ and $\boldsymbol{J} = [s \pm \boldsymbol{\beta}]$, we have $|\boldsymbol{I}| = \alpha^+ + \alpha^- = \alpha$, $\text{center}(\boldsymbol{I}) = r + (\alpha^+ - \alpha^-)/2$, $\boldsymbol{I} + \boldsymbol{J} = [r + s \pm \boldsymbol{\gamma}]$ where $\boldsymbol{\gamma} = \boldsymbol{\alpha} + \boldsymbol{\beta} = [-(\alpha^- + \beta^-), \alpha^+ + \beta^+]$, $\boldsymbol{I} + a = [r + a \pm \boldsymbol{\alpha}]$ for an arbitrary scalar a, and $s\boldsymbol{I} = [sr \pm \boldsymbol{\mu}]$ with $\boldsymbol{\mu} = s\boldsymbol{\alpha} = [-s\alpha^-, s\alpha^+]$ for any scalar $s \geq 0$. There is also a notation

[4] Note that something like $\boldsymbol{A}$ usually denotes a direct interval clock reading, whereas $\boldsymbol{I}$, $\boldsymbol{J}$ etc. represent modified or generic accuracy intervals.

to express intervals obtained from (6) by *swapping* its positive and negative accuracy, namely

$$\overline{\boldsymbol{I}} = \overline{[r \pm \boldsymbol{\alpha}]} = [r \mp \boldsymbol{\alpha}] = [r - \alpha^+, r + \alpha^-] = [r \pm \overline{\boldsymbol{\alpha}}] \tag{7}$$

where $\overline{\boldsymbol{\alpha}} = [-\alpha^+, \alpha^-] = -\boldsymbol{\alpha}$.

The generic clock synchronization algorithm of [SS97] employs the usual round-based structure of traditional internal synchronization algorithms. Periodically, every P (local time) seconds, the algorithm executes the following steps:

1. Initiation of a *full message exchange* (FME) to provide each node with the accuracy intervals of all other nodes within the system.

2. Preprocessing of the set of received accuracy intervals to make them all compatible.

3. Application of a suitable interval-valued convergence function to the set of preprocessed intervals to compute and subsequently apply a correction value for the local interval clock (starting the next round).

4. Keeping track of real-time by means of the local interval clocks.

These steps involve two basic operations called drift compensation and delay compensation. *Drift compensation* allows to shift ("drag") an accuracy interval in time by means of the local clock, without endangering accurateness of the resulting accuracy interval. Since the local clock can drift w.r.t. real-time, this requires sufficient enlargement ("deterioration") of positive and negative accuracy according to the definition below:

Definition 2 (Drift Compensation) *The result of drift compensation of an accurate interval $\boldsymbol{I} = \boldsymbol{I}(t_0)$ representing an arbitrary real-time t_0 (where $C(t_0) = T_0$) to some arbitrary real-time $t_1 \geq t_0$ (where $C(t_1) = T_1$) by means of a local clock with intrinsic inverse rate deviation bound $\boldsymbol{\rho} \subseteq \boldsymbol{\rho}_{\max}$ and rate adjustment uncertainty $\boldsymbol{u}$ is the accurate interval $\boldsymbol{I}' = \boldsymbol{I}'(t_1)$ defined by*

$$\boldsymbol{I}' = \boldsymbol{I} + T_1 - T_0 + (T_1 - T_0)\boldsymbol{\rho} + \boldsymbol{u} + I_{t_0 \neq \theta_0}\overline{\boldsymbol{G}} + I_{t_1 \neq \theta_1}\boldsymbol{G}_{\boldsymbol{\rho}}, \tag{8}$$

where

$$\boldsymbol{G}_{\boldsymbol{\rho}} = [0, G(1 + \rho^+_{\max})],\ \overline{\boldsymbol{G}} = [-G, 0],\ \boldsymbol{G} = [0, G], \tag{9}$$

and

$$I_{t \neq \theta} = \begin{cases} 0 & \text{if } t \text{ is is synchrony with } C(t), \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

The following Figure 1 shows an example of drift compensated intervals based upon some initial interval $[T_0 \pm \boldsymbol{\alpha}]$, at (equidistant) local times $T_i = T_0 + i\Delta T$, $i \geq 1$, in case of $\rho^- > \rho^+$, a fast (but deaccelerating) clock with $u, G \ll \Delta T$. For accurateness, deterioration must ensure that any resulting interval $\boldsymbol{I}'$ intersects with the line $T = t$. Note that dragging by a fast clock (accounted for by ρ^-) requires extending the lower envelope.

The second basic operation, namely, *delay compensation*, maintains accurateness of intervals that are transmitted over a network with variable transmission delays according to Assumption 4. Similar as above, positive and negative accuracy must be enlarged according to the definition below to account for the uncertainty in transmission delay:

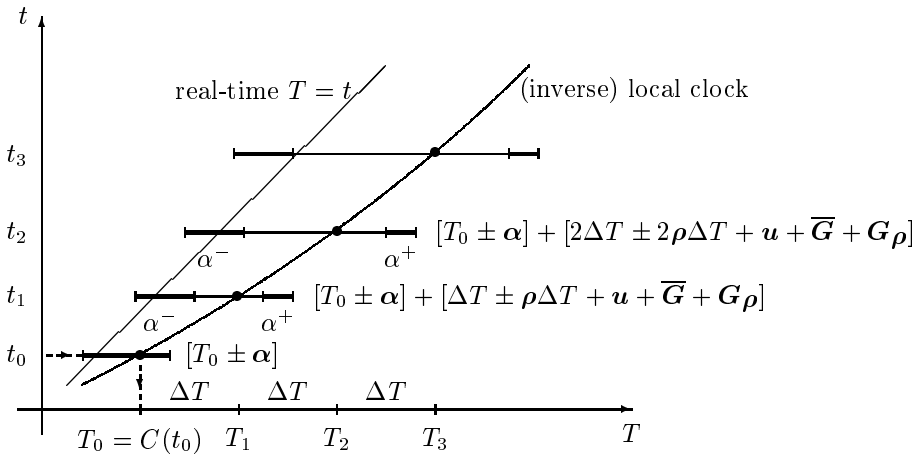


Figure 1: *Example of drift compensation in case of a fast (but deaccelerating) clock*

Definition 3 (Delay Compensation) *The result of delay compensation of an accurate interval* $\boldsymbol{I}$ *transmitted from node* p *to node* $q \neq p$ *in the absence of faults is the accurate interval*

$$\boldsymbol{I}'' = \boldsymbol{I} + \boldsymbol{2G}_A + [\delta_{pq} \pm \boldsymbol{\varepsilon}_{pq}], \tag{11}$$

where the loss in accuracy resolution during transmission is accounted for by $\boldsymbol{2G}_A = [-G_A, G_A]$.

Note that the effect of the non-zero clock granularity at the receiving node q is not considered here but rather in the drift compensation operation that takes place at q later on.

Delay compensation is illustrated in the following Figure 2, where it is assumed that the actual transmission delay is $\delta' > \delta$, $G_A \ll \varepsilon_{pq}$, and that the sender node p's clock is drift-free, i.e., progresses as real-time does. The middle time axis represents real-time, whereas the upper resp. lower one shows local time at node p resp. q. Interpreting Figure 2, one should consider the intervals $\boldsymbol{I}$ and $\boldsymbol{I}''$ as "fixed", whereas the reception time t' and hence the interval $\boldsymbol{I}'$ may vary with δ'.

Returning to the steps of our clock synchronization algorithm, drift compensation and delay compensation are required to make the received intervals compatible with each other while maintaining accurateness. More specifically, all intervals gathered at node q during an FME are transformed to intervals $\boldsymbol{I}_q^p$ representing a common point in time t_q^R as follows: For an accuracy interval $\boldsymbol{A}_p$ sent by node $p \neq q$, delay compensation (11) is applied to provide the receiver q with an initial interval $\boldsymbol{A}_p''$ that estimates $\boldsymbol{A}_p$ at the (non-synchronous) real-time of reception t_q^p (when $C_q(t_q^p) = T_q^p$). This interval $\boldsymbol{A}_p''$ is then dragged locally by the receiver's clock utilizing drift compensation (8) to some point in real-time t_q^R defined by $C_q(t_q^R) = T_q^R$ (so that t_q^R is in synchrony with node q's clock), resulting in

$$\boldsymbol{I}_q^p = \boldsymbol{A}_p + [T_q^R - T_q^p + \delta_{pq} \pm \boldsymbol{2G}_A + \boldsymbol{\varepsilon}_{pq}] + (T_q^R - T_q^p)\boldsymbol{\rho}_q + \boldsymbol{u}_q + \overline{\boldsymbol{G}}. \tag{12}$$

In addition to the intervals obtained from remote nodes $p \neq q$, there is also the own accuracy interval that must be considered. Of course, no actual transmission is

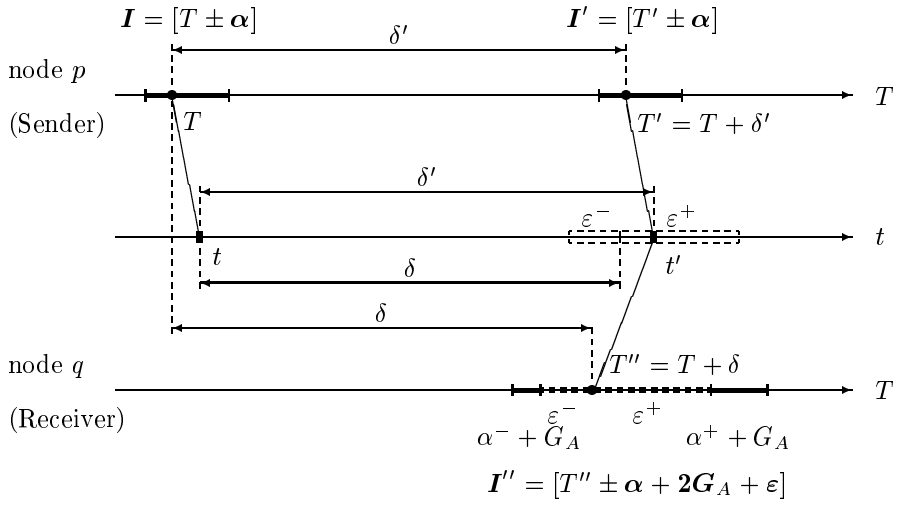


Figure 2: *Example of delay compensation when sending* $\boldsymbol{I}$ *from node* p *to* q

required here, we just have $\boldsymbol{I}_q^q = \boldsymbol{C}_q(t_q^R) = [T_q^R \pm \boldsymbol{\alpha}_q^R]$. For the sake of similarity to (12), however, one can imagine a zero-delay "loop-back transmission" of the accuracy interval $\boldsymbol{A}_q = \boldsymbol{C}_q(t_{qq}^A) = [T_{qq}^A \pm \boldsymbol{\alpha}_{qq}^A]$ at the time the FME is initiated, which "arrives" instantaneously ($T_q^q = T_{qq}^A$) at node q. We thus can write

$$\boldsymbol{I}_q^q = \boldsymbol{A}_q + T_q^R - T_q^q + (T_q^R - T_q^q)\boldsymbol{\rho}_q. \tag{13}$$

Provided that T_q^R is chosen large enough to ensure that the intervals of all non-faulty nodes can be received and processed, it follows by construction that any $\boldsymbol{I}_q^p$ is accurate if (1) $\boldsymbol{A}_p$ was accurate, (2) the transmission delay was not excessive, and (3) the receiver q is not faulty.

Now, given the set of node q's intervals $\boldsymbol{I}_q^p$, a function that provides a (small) interval that contains real-time t_q^R (i.e., is accurate) and enhances precision —despite of some possibly faulty $\boldsymbol{I}_q^p$'s— is required. Adhering to the terminology introduced in [Sch87], such a function is called a *convergence function*. The clock synchronization algorithm defined below was stated and analyzed for a generic convergence function $\boldsymbol{\mathcal{CV}}$, which can be any interval-valued function that satisfies certain properties given in Definition 8.

Putting together all the building blocks introduced before, the generic clock synchronization algorithm of [SS97] reads as follows:

Definition 4 (Generic Algorithm) *With the parameters required for the instance of the algorithm running at node* q*,*

- *node* q*'s intrinsic inverse rate deviation bound* $\boldsymbol{\rho}_q$ *and uniform bound* $\boldsymbol{\rho}_{\max} \supseteq \bigcup_p \boldsymbol{\rho}_p$ *with* $\rho_{\max} = |\boldsymbol{\rho}_{\max}| = \rho_{\max}^- + \rho_{\max}^+$ *(Assumption 2),*
- *clock granularity* G*, clock setting granularity* G_S*, node* q*'s maximum rate adjustment uncertainty* $\boldsymbol{u}_q = [-u_q^-, u_q^+]$*, and uniform maximum rate adjustment uncertainty* $\boldsymbol{u}_{\max} \supseteq \bigcup_p \boldsymbol{u}_p$ *with* $u_{\max} = u_{\max}^- + u_{\max}^+$ *(Assumption 2),*

- *transmission delay characteristics* δ_{sq}, $\boldsymbol{\varepsilon}_{sq}$ *for all nodes* $s \neq q$, *uniform bounds* $0 \leq \delta_{\min} \leq \min_{p,q}\{\delta_{pq}\}$, $\delta_{\max} \geq \max_{p,q}\{\delta_{pq}\}$, $\boldsymbol{\varepsilon}_{\max} \supseteq \bigcup_{p,q} \boldsymbol{\varepsilon}_{pq}$ *with* $\varepsilon_{\max} = |\boldsymbol{\varepsilon}_{\max}| = \varepsilon^-_{\max} + \varepsilon^+_{\max}$, *and accuracy transmission loss* G_A *(Assumption 4),*

- *computation delay compensation* Γ_q *(integer multiple of* G*) guaranteeing node* q*'s maximum computation time* γ_q *(Assumption 1), chosen according to*

$$\Gamma_q \geq \frac{\gamma_q + u^-_q}{1 - \rho^-_q},$$

  and uniform bounds $\Gamma_{\max} \geq \max_p\{\Gamma_p\}$ *and* $0 \leq \Gamma_{\min} \leq \min_p\{\Gamma_p\}$*; usually* $\Gamma_p = \Gamma_{\max} = \Gamma_{\min}$ *is the same for all nodes* p,

- *broadcast delay compensation* $\Lambda + \Omega$ *(integer multiple of* G*), chosen to satisfy*

$$\Lambda + \Omega \geq \frac{\lambda_{\max} + \omega_{\max} + u^-_{\max}}{1 - \rho^-_{\max}};$$

  in conjunction with Δ *below, it ensures that resynchronization starts only after all CSMs broadcast by non-faulty nodes during an FME have arrived (Assumption 4),*

- *transmission delay compensation* Δ *(integer multiple of* G*) chosen according to*

$$\Delta \quad \geq \quad \frac{\pi_0 + u_{\max} + \delta_{\max} + (P - \Gamma_{\min} + \pi^-)\rho_{\max} + \varepsilon^+_{\max}}{1 + \rho^+_{\max}}, \tag{14}$$

  where π_0 *and* π^- *depend on the particular convergence function employed (provided by the appropriate analysis),*

- *round period* $P \geq \Lambda + \Omega + \Delta + \Gamma_{\max}$ *(integer multiple of* G*),*

where all parameters are integer multiples of G_S *unless otherwise specified, our generic algorithm is defined as follows:*

(0) **Initial Synchronization***: At each node* q*, the local interval clock* $\boldsymbol{C}_q$ *must be initialized to the accuracy interval* $\boldsymbol{A}^0_q = [T^0_q - \alpha^{0-}_q, T^0_q + \alpha^{0+}_q]$ *at some synchronous real-time* t^0_q *by some external means. This initialization must ensure*

- $t^0_q \in \boldsymbol{A}^0_q$,
- $T^0_q \in [\Lambda + \Omega + \Delta + \Gamma_q \pm \boldsymbol{\pi}]$,
- $\boldsymbol{\alpha}^0_q \subseteq \boldsymbol{\pi}_0$,

  where $\boldsymbol{\pi}$ *and* $\boldsymbol{\pi}_0$ *depend on the particular convergence function employed (provided by the appropriate analysis).*

(1) **Periodic Synchronization***: Close to the end of each round* $k \geq 0$*, every node* q *in the system performs the following operations (the dependency of* T^I*,* t^I_q*, etc. upon round* k *is suppressed for brevity):*

(S) *CSM Send: Periodically at times* $C_q(t_q^I) = T^I = (k+1)P$, *node* q *initiates a broadcast. The message* M_{qp} *sent to node* p *at some real-time* t_{qp}^A *during that broadcast operation contains the accuracy interval* $\boldsymbol{A}_{qp} = [T_{qp}^A \pm \boldsymbol{\alpha}_{qp}^A] = \boldsymbol{C}_q(t_{qp}^A)$. *For the zero-delay "loop-back transmission" to the own node* q, $t_{qq}^A = t_q^I$ *so that* $T_{qq}^A = T^I = (k+1)P$.

(R) *CSM Reception: If a clock synchronization message* M_{pq} *from node* p *arrives at node* q *at real-time* t_q^p, *when* $C_q(t_q^p) = T_q^p$, *the interval given in (12), (13), namely*

$$\boldsymbol{I}_q^p = \begin{cases} \boldsymbol{A}_{pq} + [T_q^R - T_q^p + \delta_{pq} \pm \mathbf{2}\boldsymbol{G}_A + \boldsymbol{\varepsilon}_{pq}] + (T_q^R - T_q^p)\boldsymbol{\rho}_q + \boldsymbol{u}_q + \overline{\boldsymbol{G}} \\ \boldsymbol{A}_{qq} + T_q^R - T_{qq}^A + (T_q^R - T_{qq}^A)\boldsymbol{\rho}_q \qquad \textit{for } p = q \end{cases}$$

is computed and stored in a set $\boldsymbol{\mathcal{I}}_q$. *For the definition of the resynchronization time* T_q^R, *see step (T).*

(C) *Computation: At real-time* $t_q^{\Lambda+\Omega+\Delta}$ *defined by* $C_q(t_q^{\Lambda+\Omega+\Delta}) = T^{\Lambda+\Omega+\Delta} = (k+1)P + \Lambda + \Omega + \Delta$, *the convergence function* $\boldsymbol{\mathcal{CV}}$ *(cf. Definition 8) is applied to the compatible intervals stored in* $\boldsymbol{\mathcal{I}}_q$, *yielding the interval* $\boldsymbol{R}_q$. *In addition,* $\boldsymbol{\mathcal{I}}_q$ *is re-initialized to the empty set for the next round.*

(T) *Termination and Resynchronization: At real-time* t_q^R *defined by* $C_q(t_q^R) = T_q^R = (k+1)P+\Lambda+\Omega+\Delta+\Gamma_q$, *node* q*'s interval clock* $\boldsymbol{C}_q$ *is set to* $\boldsymbol{R}_q$ *(instantaneously or by continuous amortization).*

4 Generic Analysis of State Synchronization Revisited

Precision, accuracy and other characteristic quantities of the generic interval-based clock synchronization algorithm of Definition 4 were analyzed in [SS97]. In this subsection, we will briefly outline the most important issues and provide the announced extensions and improvements.

From the description of the algorithm in Section 3, it should be clear that accurateness of all non-faulty local interval clocks $\boldsymbol{C}_p(t)$ is maintained during all rounds. After all, any operation involved (drift compensation, delay compensation, and finally application of the convergence function) are explicitly designed to preserve accurateness of the intervals involved. Apart from the requirement of being accurate, which is a property of any single accuracy interval, however, we also have the precision requirement that applies to the reference points of a set of accuracy intervals. Precision and accuracy are in fact almost independent of each other, since the reference point can in principle be placed arbitrarily within the accuracy interval. This motivates the following interval-based definition of precision:

Definition 5 (Precision Intervals [SS97, Def. 2]) *Given some fixed* $\boldsymbol{\pi} = [-\pi^-, \pi^+]$ *with* $\pi^-, \pi^+ \geq 0$ *and* $\pi = |\boldsymbol{\pi}| = \pi^- + \pi^+$, *and a set of* $n \geq 2$ *compatible accuracy intervals* $\boldsymbol{\mathcal{I}} = \{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n\}$ *with* $\boldsymbol{I}_j = [r_j \pm \boldsymbol{\alpha}_j]$, *the* $\boldsymbol{\pi}$-precision interval $\hat{\boldsymbol{I}}_j$ *associated with* $\boldsymbol{I}_j$ *is defined as* $\hat{\boldsymbol{I}}_j = [r_j \pm \boldsymbol{\pi}]$. *The set* $\boldsymbol{\mathcal{I}}$ *is called* $\boldsymbol{\pi}$*-precise iff* $\bigcap_{j=1}^n \hat{\boldsymbol{I}}_j \neq \emptyset$.

Note that associated $\boldsymbol{\pi}$-precision intervals are not maintained "online", as are accuracy intervals, but rather computed from the reference points by adding the "static" values of $-\pi^-$, π^+.

The following lemma shows that $\boldsymbol{\pi}$-precision implies (traditional) precision $\pi = |\boldsymbol{\pi}|$:

Lemma 1 ($\boldsymbol{\pi}$-Precision vs. Precision) *Given a $\boldsymbol{\pi}$-precise set $\boldsymbol{\mathcal{I}} = \{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n\}$ of $n \geq 2$ compatible accuracy intervals $\boldsymbol{I}_j = [r_j \pm \boldsymbol{\alpha}_j]$, then*

(1) $|\hat{\boldsymbol{I}}_i \cup \hat{\boldsymbol{I}}_j| \leq 2\pi$ for any $1 \leq i, j \leq n$,

(2) $|r_i - r_j| \leq \pi$ for any $1 \leq i, j \leq n$.

Proof: See [SS97], Lemma 3. □

Although the definition of $\boldsymbol{\pi}$-precision is a key for precision analysis, there are only a few occasions where $\boldsymbol{\pi}$-precise intervals are encountered explicitly. In most cases, the slightly stronger predicate of *$\boldsymbol{\pi}$-accurateness* (implying $\boldsymbol{\pi}$-precision) is used, which is based on a suitable notion of an *internal global time* $\tau = \tau(t)$. More specifically, it was shown in [SS97] that it makes sense to stipulate an "artificial" internal global time $\tau^k = \tau^k(t)$ for round k that progresses as real-time does. Of course, like real-time, internal global time is not directly accessible and usually $\tau^k(t) \neq t$. Nevertheless, internal global time of any fixed round is equivalent to real-time when measuring time intervals.

The concept of internal global time makes it possible to define an analogon to accurateness as follows:

Definition 6 ($\boldsymbol{\pi}$-correctness [SS97, Def. 3]) *For $\boldsymbol{\pi} = [-\pi^-, \pi^+]$ with $\pi^-, \pi^+ \geq 0$,*

(1) an accuracy interval $\boldsymbol{I} = \boldsymbol{I}(t)$ is $\boldsymbol{\pi}$-accurate (w.r.t. internal global time $\tau^k = \tau^k(t)$ of round k) iff the $\boldsymbol{\pi}$-precision interval $\hat{\boldsymbol{I}} = \hat{\boldsymbol{I}}(\tau^k)$ associated with $\boldsymbol{I}$ satisfies $\tau^k \in \hat{\boldsymbol{I}}$,

(2) an accuracy interval $\boldsymbol{I}$ is $\boldsymbol{\pi}$-correct (w.r.t. internal global time of round k) iff $\boldsymbol{I}$ is both $\boldsymbol{\pi}$-accurate and accurate,

(3) a set $\boldsymbol{\mathcal{I}}$ of compatible accuracy intervals is $\boldsymbol{\pi}$-correct if all members are $\boldsymbol{\pi}$-correct.

Note that we can usually unambiguously write $\hat{\boldsymbol{I}} = \hat{\boldsymbol{I}}(t) = \hat{\boldsymbol{I}}(\tau^k)$ for $\tau^k = \tau^k(t)$, meaning that $\hat{\boldsymbol{I}}$ represents $\tau^k(t)$ iff $\boldsymbol{I}$ represents t.

The following Figure 3 shows an example of two $\boldsymbol{\pi}$-correct intervals.

Now we are ready for explaining how the generic algorithm given in Definition 4 maintains precision. The most important observation is that all steps of the algorithm, except the application of the convergence function, maintain precision automatically by maintaining accuracy. To understand why, assume that all members of the set $\boldsymbol{\mathcal{C}}(t)$ of non-faulty interval clocks are $\boldsymbol{\pi}_0$-correct at some real-time t^0, i.e., that their associated $\boldsymbol{\pi}_0$-precision intervals contain $\tau(t^0)$. In order to capture real-time t by $\boldsymbol{C}_p(t)$, it must be deteriorated (enlarged) properly to compensate for the drift of the local clock. However, if this is done properly to capture real-time $t' > t^0$, then the associated $\boldsymbol{\pi}$-precision interval $\hat{\boldsymbol{C}}_p(t')$ captures internal global time $\tau(t') > \tau(t^0)$ as well, provided that $\boldsymbol{\pi}$ is the result of enlarging $\boldsymbol{\pi}_0$ by the maximum amount any $\boldsymbol{C}_q$ has been enlarged. This is a simple consequence of the fact that internal global time progresses as real-time does.

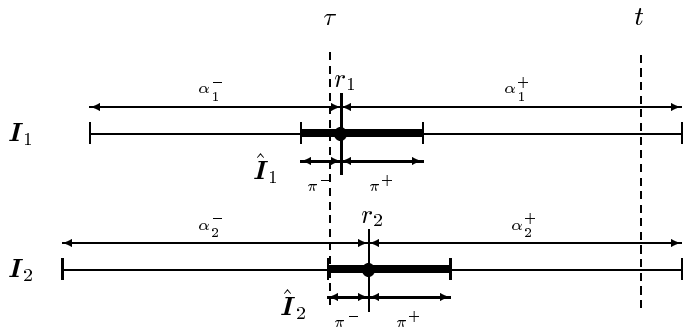


Figure 3: *Example of two $\boldsymbol{\pi}$-accurate intervals*

Note anyway that enlargement of precision intervals is just a matter of analysis — the algorithm need not deal explicitly with precision intervals at all.

Whereas enlarging $\boldsymbol{\pi}_0$ to $\boldsymbol{\pi}$ guarantees $\boldsymbol{\pi}$-correctness of all non-faulty $\boldsymbol{C}_p(t')$, this cannot ensure bounded precision for $t \to \infty$. Periodic resynchronizations are required for that purpose, giving raise to our round-based algorithm. More specifically, at the end of the k-th round, the nodes' current $\boldsymbol{\pi}$-correct local interval clocks are set to newly computed accuracy intervals that are $\boldsymbol{\pi}_0$-precise for some $\boldsymbol{\pi}_0 \subset \boldsymbol{\pi}$ (= precision enhancement provided by the convergence function). Note that we cannot safely assume $\boldsymbol{\pi}_0$-correctness here, since it will usually happen that round k's internal global time τ^k does not lie in the intersection of the new $\boldsymbol{\pi}_0$-precision intervals. However, if a new internal global time τ^{k+1} is defined for round $k+1$, $\boldsymbol{\pi}_0$-correctness w.r.t. τ^{k+1} can of course be guaranteed. Consequently, resynchronization starts the next round $k+1$, during which initial precision $\boldsymbol{\pi}_0$ again deteriorates to $\boldsymbol{\pi}$.

Note carefully that only the interval clocks' associated $\boldsymbol{\pi}$-precision intervals experience (precision) enhancement during resynchronization. The local interval clocks $\boldsymbol{C}_p$ itself must continuously track real-time t, so that the accuracies could be monotonically increasing; the accuracy in round k can in fact be viewed as an accumulation of the $\boldsymbol{\pi}$-precision intervals during round $0, \ldots, k$. This eventually explains why t and τ will usually be apart, as mentioned earlier.

The detailed analysis in [SS97] reflects the principal structure outlined above. In fact, a sequence of lemmas is provided that characterize how accuracy/precision intervals evolve in a single round: Lemma 11 in [SS97] describes the set of intervals fed into the convergence function, and Lemma 12 shows that the precision provided at the beginning of round k is reestablished at the beginning of round $k+1$. On top of that, a simple induction proof is conducted to derive the major Theorem 1.

Since any suitable convergence function can be plugged into the generic algorithm of Definition 4, all results obtained in [SS97] are generic in the sense that they are expressed in terms of a few characteristic parameters of the convergence function defined in [SS97, Def. 11], namely,

- accuracy preservation function $\boldsymbol{\aleph}(.)$, giving bounds on the provided interval of accuracies,
- precision preservation function $\boldsymbol{\Phi}(.)$, giving the precision of the provided accuracy interval w.r.t. τ^k,

- precision enhancement function $\Pi(.)$, giving the precision of the provided accuracy interval w.r.t. τ^{k+1}.

In our original Definition [SS97, Def. 11], we required $\boldsymbol{\mathcal{CV}}$ to be both translation invariant and weakly monotonic according to the following definition:

Definition 7 (Translation Invariance and Weak Monotonicity [SS97, Def. 10]) *Given two sets $\boldsymbol{\mathcal{I}} = \{\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n\}$ and $\boldsymbol{\mathcal{J}} = \{\boldsymbol{J}_1, \ldots, \boldsymbol{J}_n\}$ of $n \geq 1$ accuracy intervals, an interval-valued function $\boldsymbol{f}()$ of $n \geq 1$ interval arguments is called*

(1) weakly monotonic *iff $\boldsymbol{I}_j \subseteq \boldsymbol{J}_j$ with ref($\boldsymbol{I}_j$) = ref($\boldsymbol{J}_j$) for all $1 \leq j \leq n$ implies $\boldsymbol{f}(\boldsymbol{\mathcal{I}}) \subseteq \boldsymbol{f}(\boldsymbol{\mathcal{J}})$,*

(2) translation invariant *iff $\boldsymbol{f}(\boldsymbol{I}_1 + \Delta, \ldots, \boldsymbol{I}_n + \Delta) = \boldsymbol{f}(\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n) + \Delta$ for any real Δ.*

However, we recognized that there is no need to require weak monotonicity for $\boldsymbol{\mathcal{CV}}$ itself: Monotonicity is used in [SS97] to justify that bounds on input intervals carry over to bounds on the result. Still, there is no need to establish this property for $\boldsymbol{\mathcal{CV}}$, since it suffices to show that all characteristic functions are monotonic w.r.t. their interval arguments. The modified definition of the generic convergence function in Definition 8 now accounts for this fact.

More importantly, in the course of analyzing particular convergence functions [Sch97b], [Sch97a], we recognized that applying the generic framework of [SS97] literally provides overly conservative[5] worst case accuracy bounds. Intuitively, this is due to the fact that we simply added up the worst case enlargement of, say, α_p^+ —provided by the convergence function's accuracy preservation function $\aleph^+(.)$— in every round. The worst case enlargement of α_p^+, however, cannot occur successively in consecutive rounds. This is due to the fact that the occurence of the worst case setting usually yields an initial precision (i.e., after resnchronization) that is better than the worst case one. This means that the initial precision in the next round is smaller than the worst case one assumed for the previous round, which in turn prohibits the occurence of the worst case enlargement of α_p^+ then.

The key idea used for improving our analysis is to take into account the common intersection of those $\boldsymbol{\pi}$-precision intervals associated with the convergence function's input intervals that eventually determine the worst case setting for, say, α_p^+. By feeding its length ι as an additional parameter into the accuracy preservation function $\aleph^+(.)$, the worst case enlargement of α_p^+ can be conditioned on the common intersection actually present. Note that a lower bound was found to be sufficient for this purpose, since excessive adjustments happen for small intersections only.[6] An additional characteristic function $\Im^+(.)$ resp. $\Im^-(.)$ is introduced for keeping track of how the convergence function affects ι in worst case scenarios w.r.t. positive resp. negative accuracy. This (conditional) intersection enhancement function effectively determines ι for the next round, i.e., propagates the required information over multiple rounds.

Making this idea working in practice, however, is tricky for several reasons: First of all, it is the particular convergence function that determines how many/which input intervals

[5]Note that we do not mean the obvious lacking of a good worst case bound on $\alpha_p = \alpha_p^- + \alpha_p^+$ here. In fact, since the worst case scenario for negative and positive accuracy cannot occur simultaneously, simply adding the bounds on α_p^+ and α_p^- provides an overly conservative bound on α_p only.

[6]At least for any particular convergence function considered up to now.

are involved in the worst case accuracy setting. Leaving $\mathcal{CV}$ unspecified in our generic framework thus makes it necessary to supply all the information required for constructing the particular ι from our general expressions. Moreover, different ι^- resp. ι^+ are usually required for computing the worst case bound for α_p^- resp. α_p^+. In fact, it may even be the case that some ι depending upon p is required for determining the worst case accuracy at node p, see [Sch97a] for an example. It is worth noting, however, that something like $\iota^p < \iota^q$ implies that the computed worst case accuracy bounds for nodes p and q cannot both be tight.

To cope with those problems, we actually utilize individual lower bounds ι_s^+ resp. ι_s^-, $1 \leq s \leq n$, on the length of the intersection $\hat{\boldsymbol{I}}_p^s \cap \hat{\boldsymbol{I}}_q^s \cap \hat{\boldsymbol{I}}_p^{\min_p} \cap \hat{\boldsymbol{I}}_q^{\min_q}$ resp. $\hat{\boldsymbol{I}}_p^s \cap \hat{\boldsymbol{I}}_q^s \cap \hat{\boldsymbol{I}}_p^{\max_p} \cap \hat{\boldsymbol{I}}_q^{\max_q}$, where $\min_x$ resp. $\max_x$ represents that non-faulty node that leads to the leftmost right($\hat{\boldsymbol{I}}_x^{\min_x}$) resp. the rightmost left($\hat{\boldsymbol{I}}_x^{\max_x}$) for $x \in \{p, q\}$. Clearly, ι_s^+ resp. ι_s^- is meant for dealing with with the worst case accuracy setting for α_p^+ resp. α_p^-, so $\forall s : \iota_s^+$ resp. $\forall s : \iota_s^-$ are supplied as parameters to $\aleph^+(.)$ resp. $\aleph^-(.)$ and $\Im^+(.)$ resp. $\Im^-(.)$ in Definition 8.

In order to be able to inductively compute an expression for, say, ι_p^+, the conditional intersection enhancement function $\Im^+(.)$ provides an upper bound $\pi_0^{\iota_{pq}^+}$ on the mutual precision of the $\mathcal{CV}$'s results $\{\boldsymbol{R}_p, \boldsymbol{R}_q\}$ computed at node p, q under the worst case accuracy setting for α_p^+. Since $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$ are both $\boldsymbol{\pi}_0$-correct, this implies that the mutual intersection $\hat{\boldsymbol{R}}_p \cap \hat{\boldsymbol{R}}_q$ for any q (including $q = \min_x$) has length at least $\overline{\iota}_p^+ = \pi_0 - \max_q \pi_0^{\iota_{pq}^+}$. Now, since the drift and delay compensation operations applied for computing the input intervals fed into the convergence function are accuracy preserving, cf. (12), it follows that any initial intersection is necessarily preserved during a round. Thus, any intersection of source intervals like $\hat{\boldsymbol{I}}_p^s \cap \hat{\boldsymbol{I}}_q^{\min_q}$ must also have length at least $\overline{\iota}_s^+$, which implies that we can simply use $\iota_s^+ = \overline{\iota}_s^+$.

The above considerations lead to the following modified definition of the generic convergence function of [SS97]:

Definition 8 (Generic Convergence Function $\mathcal{CV}$, cf. [SS97, Def. 11]) *Let $\mathcal{I}_p = \{\boldsymbol{I}_p^1, \ldots, \boldsymbol{I}_p^n\}$ resp. $\mathcal{I}_q = \{\boldsymbol{I}_q^1, \ldots, \boldsymbol{I}_q^n\}$, $q \neq p$, be two ordered sets of n compatible intervals (all representing the <u>same</u> real-time t) obtained at nodes p resp. q at the end of a round, which are in accordance with a given fault model $\mathcal{F}$. Assuming that*

[1] any non-faulty $\boldsymbol{I}_p^i$ is $\boldsymbol{\pi}_p^i$-correct for $\boldsymbol{\pi}_p^i \in \boldsymbol{\mathcal{P}}_p = \{\boldsymbol{\pi}_p^1, \ldots, \boldsymbol{\pi}_p^n\}$ denoting a given set of precision bounds (and analogously for $\boldsymbol{I}_q^i$ with set of precision bounds $\boldsymbol{\mathcal{P}}_q = \{\boldsymbol{\pi}_q^1, \ldots, \boldsymbol{\pi}_q^n\}$),

[1'] $\boldsymbol{\mathcal{P}} = \{\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^n\}$ with $\boldsymbol{\pi}_p^i \cup \boldsymbol{\pi}_q^i \subseteq \boldsymbol{\pi}^i \subseteq \boldsymbol{\pi}^H$, for some suitable $\boldsymbol{\pi}^H$, denotes a set of uniform precision bounds ensuring $\boldsymbol{\pi}^i$-correctness of both $\boldsymbol{I}_p^i$ and $\boldsymbol{I}_q^i$ (if non-faulty),

[2] any pair of non-faulty intervals $\{\boldsymbol{I}_p^i, \boldsymbol{I}_q^i\}$ is $\boldsymbol{\pi}_I$-precise for some $\boldsymbol{\pi}_I \subseteq \boldsymbol{\pi}^H$,

[3] for any s with both $\boldsymbol{I}_p^s$ and $\boldsymbol{I}_q^s$ being non-faulty, the intersection of the associated precision intervals $\hat{\boldsymbol{I}}_p^s \cap \hat{\boldsymbol{I}}_q^s \cap \hat{\boldsymbol{I}}_p^{\min_p} \cap \hat{\boldsymbol{I}}_q^{\min_q}$ resp. $\hat{\boldsymbol{I}}_p^s \cap \hat{\boldsymbol{I}}_q^s \cap \hat{\boldsymbol{I}}_p^{\max_p} \cap \hat{\boldsymbol{I}}_q^{\max_q}$, where $\min_x$ resp. $\max_x$ represents that non-faulty node that leads to the leftmost right($\hat{\boldsymbol{I}}_x^{\min_x}$) resp. the rightmost left($\hat{\boldsymbol{I}}_x^{\max_x}$) for $x \in \{p, q\}$, has length at least $\iota_s^+ \geq 0$ resp. $\iota_s^- \geq 0$ (integer multiples of G_S),

[4] the accuracies of any non-faulty $\boldsymbol{I}_p^i = [T_p^i \pm \boldsymbol{\alpha}_p^i]$ *are integer multiples of* G_S *satisfying* $\boldsymbol{\alpha}_p^i \subseteq \boldsymbol{\beta}_p^i \in \boldsymbol{\mathcal{B}}_p$ *for a given set of accuracy bounds* $\boldsymbol{\mathcal{B}}_p = \{\boldsymbol{\beta}_p^1, \ldots, \boldsymbol{\beta}_p^n\}$ *(and analogously for* $\boldsymbol{I}_q^i$ *with set of accuracy bounds* $\boldsymbol{\mathcal{B}}_q$*),*

let

$$\begin{aligned} \boldsymbol{R}_p &= \boldsymbol{\mathcal{CV}}(\boldsymbol{\mathcal{I}}_p) = [T_p' \pm \boldsymbol{\alpha}_p'] \\ \boldsymbol{R}_q &= \boldsymbol{\mathcal{CV}}(\boldsymbol{\mathcal{I}}_q) = [T_q' \pm \boldsymbol{\alpha}_q'] \\ &. \end{aligned} \tag{15}$$

The generic convergence function $\boldsymbol{\mathcal{CV}}$ *must be translation invariant and should provide accurate intervals with reference point and accuracies being integer multiples of* G_S*. Its behavior is characterized by the following functions, which must be monotonic w.r.t. any interval argument:*

(1) Precision preservation function $\boldsymbol{\Phi}(.)$*, so that* $\boldsymbol{R}_p$ *is* $\boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}_p, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I)$*-correct and* $\boldsymbol{R}_q$ *is* $\boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I)$*-correct, with* $|\boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I)| = \mathcal{O}(\pi^H)$ *for* $\pi^H = |\boldsymbol{\pi}^H|$*.*

(2) Precision enhancement function $\Pi(.)$*, so that the set* $\{\boldsymbol{R}_p, \boldsymbol{R}_q\}$ *is* $\boldsymbol{\pi}_0$*-precise for any* $\boldsymbol{\pi}_0$ *satisfying* $|\boldsymbol{\pi}_0| = \pi_0 = \Pi(\boldsymbol{\mathcal{P}}, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I)$*, with* $\Pi(\boldsymbol{\mathcal{P}}, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I) < \pi^H = |\boldsymbol{\pi}^H|$*.*

(3) Conditional intersection enhancement functions $\Im^-(.)$ *resp.* $\Im^+(.)$*, so that the set* $\{\boldsymbol{R}_p, \boldsymbol{R}_q\}$ *is* $\boldsymbol{\pi}_0^{\iota_{pq}^-}$*-precise resp.* $\boldsymbol{\pi}_0^{\iota_{pq}^+}$*-precise with*

$$\begin{aligned} \pi_0^{\iota_{pq}^-} &= \Im^-(\boldsymbol{\mathcal{B}}_p, \boldsymbol{\mathcal{B}}_q, \boldsymbol{\mathcal{P}}_p, \boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^-) \\ \pi_0^{\iota_{pq}^+} &= \Im^+(\boldsymbol{\mathcal{B}}_p, \boldsymbol{\mathcal{B}}_q, \boldsymbol{\mathcal{P}}_p, \boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^+) \end{aligned}$$

for worst case accuracy settings w.r.t. α_p^- *resp.* α_p^+*.*

(4) Conditional accuracy preservation functions $\aleph^-(.)$*,* $\aleph^+(.)$*, so that*

$$\boldsymbol{\alpha}_p' \subseteq \left[-\aleph^-(\boldsymbol{\mathcal{B}}_p, \boldsymbol{\mathcal{P}}_p, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^-), \aleph^+(\boldsymbol{\mathcal{B}}_p, \boldsymbol{\mathcal{P}}_p, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^+)\right]$$

Note that both the precision enhancement function $\Pi(.)$ and the intersection enhancement functions $\Im^-(.)$, $\Im^+(.)$ provide a worst case precision, although for different classes of input scenarios: Whereas $\Pi(.)$ provides the precision for any input scenario, $\Im^-(.)$ resp. $\Im^+(.)$ is valid for scenarios leading to worst case α_p^- resp. α_p^+ according to item (4) only.

Another issue that was left open by the analysis of [SS97] is the determination of *traditional accuracy*, which gives the amount local time may drift from real-time during a given time interval Δt. Although worst case bounds on accuracy intervals obviously provide an upper bound on traditional accuracy as well, this usually leads to overly conservative estimates, see e.g. [Sch97b]. Therefore, an explicit expression for traditional accuracy is added to our major theorem. Moreover, taking the limit $\Delta t \to \infty$, traditional accuracy leads to the rate (and hence drift) of the synchronized clocks, which is more convenient for comparison. Note that appropriate worst case bounds for are available for most existing internal synchronization algorithms, see e.g. [LWL88], [MS85], [Sch87], [ST87], [VR92], [DB93], [FC95], [VRC97].

Our notion of internal global time makes it easy to deal with traditional accuracy. We only have to bound the maximum "jump" internal global time can experience when switching from one round to the next. This is sufficient since internal global time progresses as real-time does during a round, so that no additional deviation from real-time occurs in between synchronization instants.

The appropriately modified and extended major result of the generic analysis in [SS97] for the case of instantaneous adjustment[7] of the local clocks in step (T) of the algorithm in Definition 4 reads as follows:

Theorem 1 (Instantaneous Correction, cf. [SS97, Thm. 1]) *Running in a system complying to Assumptions 1–4, the clock synchronization algorithm of Definition 4 using the generic convergence function $\mathcal{CV}$ —characterized by accuracy preservation $\aleph^{\pm}(.)$, precision preservation $\Phi(.)$, precision enhancement $\Pi(.)$, and intersection enhancement $\Im^{\pm}(.)$ subject to a given fault model $\mathcal{F}$— guarantees accuracy and precision for all rounds $k \geq 0$ as follows:*

(0) The accuracy interval $\boldsymbol{A}_q^{k+1} = \boldsymbol{A}_q^{k+1}(t_q^{R,k}) = [T_q^{k+1} \pm \boldsymbol{\alpha}_q^{k+1}]$ provided by the local interval clock of a non-faulty node q at the beginning of round $k+1$, $k \geq 0$, satisfies $\boldsymbol{\alpha}_q^{k+1} \subseteq \boldsymbol{\beta}_q^{k+1}$ with

$$\begin{aligned} \boldsymbol{\beta}_q^{k+1} &= \left[-\aleph^-(\boldsymbol{\mathcal{B}}_q^{k+1}, \boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^{k,-}), \aleph^+(\boldsymbol{\mathcal{B}}_q^{k+1}, \boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^{k,+})\right] \quad (16) \\ \boldsymbol{\beta}_q^0 &= \boldsymbol{\alpha}_q^0, \end{aligned}$$

where $\boldsymbol{\mathcal{B}}_q^{k+1} = \{\boldsymbol{\beta}_q^{1,k+1}, \ldots, \boldsymbol{\beta}_q^{n,k+1}\}$ is defined by $\boldsymbol{\beta}_q^{p,k+1} = \boldsymbol{\beta}_p^k + \boldsymbol{\omega}_q^p$ with

$$\begin{aligned} \boldsymbol{\omega}_q^p &= \boldsymbol{u}_p + \boldsymbol{u}_q + \overline{\boldsymbol{G}} + 2\boldsymbol{G}_A + \boldsymbol{\varepsilon}_{pq} \\ &\quad + (P - \Delta - \Gamma_p)\boldsymbol{\rho}_p + (\Gamma_q + \Delta - \delta_{pq})\boldsymbol{\rho}_q \\ &\quad + (\Lambda + \Omega)[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}] \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_{\max} \qquad \textit{for } p \neq q, \quad (17) \\ \boldsymbol{\omega}_q^q &= \boldsymbol{u}_q + P\boldsymbol{\rho}_q + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_q, \quad (18) \end{aligned}$$

the set $\boldsymbol{\mathcal{P}}_q = \{\boldsymbol{\pi}_q^1, \ldots, \boldsymbol{\pi}_q^n\}$ of node q's precision bounds $\boldsymbol{\pi}_q^p \subseteq \boldsymbol{\pi}^p \subseteq \boldsymbol{\pi}^H$ —see item (1)— is defined by

$$\begin{aligned} \boldsymbol{\pi}_q^p &= \boldsymbol{\pi}_0 + \boldsymbol{u}_p + \boldsymbol{u}_q + \overline{\boldsymbol{G}} + \boldsymbol{\varepsilon}_{pq} + (P - \Delta - \Gamma_p)\boldsymbol{\rho}_p + (\Gamma_q + \Delta - \delta_{pq})\boldsymbol{\rho}_q \\ &\quad + (\Lambda + \Omega)[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}] \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_{\max} \qquad \textit{for } p \neq q, \quad (19) \\ \boldsymbol{\pi}_q^q &= \boldsymbol{\pi}_0 + \boldsymbol{u}_q + P\boldsymbol{\rho}_q + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_q, \quad (20) \end{aligned}$$

and

$$\begin{aligned} \iota_q^{k+1,-} &= \pi_0 - \max_j \Im^-(\boldsymbol{\mathcal{B}}_q^{k+1}, \boldsymbol{\mathcal{B}}_j^{k+1}, \boldsymbol{\mathcal{P}}_q, \boldsymbol{\mathcal{P}}_j, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^{k,-}) \geq 0, \quad (21) \\ \iota_q^{k+1,+} &= \pi_0 - \max_j \Im^+(\boldsymbol{\mathcal{B}}_q^{k+1}, \boldsymbol{\mathcal{B}}_j^{k+1}, \boldsymbol{\mathcal{P}}_q, \boldsymbol{\mathcal{P}}_j, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I, \forall s : \iota_s^{k,+}) \geq 0, \quad (22) \\ \iota_q^{0,-} = \iota_q^{0,+} &= \pi_0 - \max_j\{\alpha_j^0\} \geq 0. \quad (23) \end{aligned}$$

[7] Consult [SS97] for how to carry over this result to clock adjustment via continuous amortization.

(1) The interval clocks of non-faulty nodes are synchronized to the (observable) initial worst case precision (i.e., the precision at the beginning of each round of the slowest non-faulty clock)

$$\begin{aligned}\pi_{0,\max} &= \pi_0 + u_{\max} + G + (\Gamma_{\max} - \Gamma_{\min})\rho_{\max} \\ &\quad + \mathcal{O}(P\rho_{\max}^2 + G\rho_{\max} + \varepsilon_{\max}\rho_{\max}) \end{aligned} \tag{24}$$

with $\pi_0 = |\boldsymbol{\pi}_0|$*, where* $\boldsymbol{\pi}_0$ *is a solution of the equation*

$$|\boldsymbol{\pi}_0| = \Pi(\boldsymbol{\mathcal{P}}, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I) \tag{25}$$

for the set $\boldsymbol{\mathcal{P}} = \{\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^n\}$ *of uniform precision bounds* $\boldsymbol{\pi}^p \subseteq \boldsymbol{\pi}^H$ *defined by*

$$\begin{aligned}\boldsymbol{\pi}^p &= \boldsymbol{\pi}_0 + \boldsymbol{u}_p + \boldsymbol{u}_{\max} + \overline{\boldsymbol{G}} + \boldsymbol{\varepsilon}_{\max} \\ &\quad + (P - \Delta - \Gamma_p)\boldsymbol{\rho}_p + (\Gamma_{\max} + \Delta - \delta_{\min})\boldsymbol{\rho}_{\max} \\ &\quad + (\Lambda + \Omega)[-(\rho_{\max}^- - \rho_p^-), \rho_{\max}^+ - \rho_p^+] \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_{\max}, \end{aligned} \tag{26}$$

$$\boldsymbol{\pi}^{\overline{q}} = \boldsymbol{\pi}_0 + \boldsymbol{u}_q + P\boldsymbol{\rho}_q + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_q \tag{27}$$

$$\begin{aligned}\boldsymbol{\pi}^H &= \boldsymbol{\pi}_0 + 2\boldsymbol{u}_{\max} + \overline{\boldsymbol{G}} + \boldsymbol{\varepsilon}_{\max} + (P + \Gamma_{\max} - \Gamma_{\min} - \delta_{\min})\boldsymbol{\rho}_{\max} \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_{\max}, \end{aligned} \tag{28}$$

$$\begin{aligned}\boldsymbol{\pi}_I &= \boldsymbol{\varepsilon}_{\max} + B\boldsymbol{u}_{\max} + \overline{\boldsymbol{G}} + (\Lambda + \Omega + \Delta + \Gamma_{\max} - \delta_{\min})\boldsymbol{\rho}_{\max} \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_{\max}, \end{aligned} \tag{29}$$

where $\boldsymbol{\pi}^{\overline{q}} \subseteq \boldsymbol{\pi}^q$ *denotes node q's own (i.e., non-remote) precision bound.*

(2) The (observable) worst-case precision $\pi_{\max}$ *satisfies*

$$\begin{aligned}\pi_{\max} &= \max\Big\{\pi^- + u_{\max}^+ + (\Gamma_{\max} - \Gamma_{\min})\rho_{\max}^+, \\ &\qquad \pi^+ + u_{\max}^- + (\Gamma_{\max} - \Gamma_{\min})\rho_{\max}^-,\ \pi_0 + u_{\max} + P\rho_{\max}\Big\} \\ &\quad + G + \mathcal{O}(P\rho_{\max}^2 + G\rho_{\max} + \varepsilon_{\max}\rho_{\max}) \end{aligned} \tag{30}$$

with

$$\begin{aligned}\boldsymbol{\pi} &= \overline{\boldsymbol{\Phi}}(\boldsymbol{\mathcal{P}}, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I) + \boldsymbol{\pi}_0 + \boldsymbol{u}_{\max} + P\boldsymbol{\rho}_{\max} \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_{\max}, \end{aligned} \tag{31}$$

where $\overline{\boldsymbol{\Phi}}(.)$ *denotes the result of* $\boldsymbol{\Phi}(.)$ *with swapped positive and negative accuracy.*

(3) Resynchronization of any two non-faulty nodes p, q occurs within real-time $t_p^R - t_q^R$ *satisfying*

$$\begin{aligned}t_p^R - t_q^R &\subseteq \Gamma_p - \Gamma_q + [-\pi_0, \pi_0] + \boldsymbol{u}_p + \overline{\boldsymbol{u}}_q + P(\boldsymbol{\rho}_p + \overline{\boldsymbol{\rho}}_q) \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})[-\rho_{\max}, \rho_{\max}], \end{aligned} \tag{32}$$

where clock adjustments Υ_q *of at most* $\Upsilon_q \in \boldsymbol{\pi}_q \subseteq \boldsymbol{\pi}$ *defined by*

$$\boldsymbol{\pi}_q = \overline{\boldsymbol{\Phi}}(\boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I) + \boldsymbol{\pi}_0 + \boldsymbol{u}_q + P\boldsymbol{\rho}_q + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\boldsymbol{\rho}_{\max} \tag{33}$$

are applied to the clock of a non-faulty node q.

(4) Let $\boldsymbol{\Phi} = \bigcup_q \boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I) \subseteq \boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I)$. *For any round* $k \geq 0$, $\tau^{k+1}(t) - \tau^k(t) \in \overline{\boldsymbol{\Phi}} - \overline{\boldsymbol{\pi}}_0$, *and the traditional accuracy at the beginning of round* $k+1$ *satisfies*

$$T_q^{k+1} - t_q^{R,k} \in \overline{\boldsymbol{\pi}}_0 + (k+1)(\overline{\boldsymbol{\Phi}} - \overline{\boldsymbol{\pi}}_0). \tag{34}$$

The inverse rate $r_{q,syn}^{-1}$ *of the synchronized clock at any node* q *evaluates to*

$$r_{q,syn}^{-1} = \lim_{k\to\infty} \frac{t_q^{R,k} - t_q^0}{T_q^{k+1} - T_q^0} \in \Big[1 \pm \frac{\boldsymbol{\Phi} - \boldsymbol{\pi}_0}{P}\Big], \tag{35}$$

where $T_q^0 = C_q(t_q^0)$ *is node* q*'s local time at the beginning of round* $k = 0$.

Proof: With the exception of item (0) and item (4), the proof of Theorem 1 in [SS97] applies without modification. We just replaced π in certain remainder terms like (26) by the coarse bound $\pi = \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$ established in the proof of [SS97, Lem. 12], and added (27) according to [SS97, Eq. (58)], which gives the precision bound $\boldsymbol{\pi}^{\overline{q}}$ that applies for local (non-remote) intervals $\boldsymbol{I}_q^q$, cf. (13).

As far as item (0) is concerned, we found it convenient to introduce the additional abbreviation $\boldsymbol{\omega}_q^p$ in (17)/(18). Moreover, according to the modified Definition 8 of the generic convergence function, we splitteded up $\boldsymbol{\aleph}(.)$ in $\aleph^-(.)$ resp. $\aleph^+(.)$ and extended the list of parameters to include $\forall s : \iota_s^{k,-}$ resp. $\forall s : \iota_s^{k,+}$. The justification of the appropriate definitions (21)–(23) follows the line of reasoning outlined prior to Definition 8: An upper bound $\pi_0^{\iota_{qj}^+}$ on the precision of the set $\mathrm{shift}_{t'}(\{\boldsymbol{A}_q^k, \boldsymbol{A}_j^k\})$ at the beginning of a round k implies a lower bound $\iota_{qj}^+ = \pi_0 - \pi_0^{\iota_{qj}^+}$ on the length of the mutual intersection of the associated $\boldsymbol{\pi}_0$-precision intervals $\hat{\boldsymbol{A}}_q^k \cap \hat{\boldsymbol{A}}_j^k$. Taking the minimum over all j gives a lower bound $\iota_q^+ = \pi_0 - \max_j \pi_0^{\iota_{qj}^+}$ on that mutual intersection for arbitrary nodes j, including $j = \min_x$. The required $\pi_0^{\iota_{qj}^+}$, however, is given by the conditional intersection enhancement funtion $\Im^+(.)$ for rounds $k > 0$, and by the initial synchronization assumption in Definition 4 for round $k = 0$.

It only remains to confirm that the precision intervals associated with any (non-faulty) $\boldsymbol{I}_p^s$ and $\boldsymbol{I}_q^s$ —fed into the convergence function at node p and q— have a mutual intersection with either $\boldsymbol{I}_p^{\min_p}$ and $\boldsymbol{I}_q^{\min_q}$ of length at least ι_s^+ as well. However, the drift and delay compensation operations used to obtain round k's $\boldsymbol{I}_x^s$ from $\boldsymbol{A}_s^k$ according to (12) have been explicitly designed to preserve accurateness, so they must preserve any initial mutual intersection. Hence, choosing $\iota_q^{k+1,+}$ and $\iota_q^{k+1,-}$ according to (21)–(23) is legitimate.

Turning our attention to item (4), we know that $\boldsymbol{A}_q^{k+1}$ is $\boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}_q, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I)$-correct w.r.t. τ^k by virtue of item (2) of Definition 8 and hence $\boldsymbol{\Phi}$-correct. Note that $\boldsymbol{\Phi} \subseteq \boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}, \boldsymbol{\pi}^H, \boldsymbol{\pi}_I)$ is a simple consequence of weak monotonicity of $\boldsymbol{\Phi}(.)$. Moreover, from item (3) of Definition 8, it follows that all $\boldsymbol{A}_p^{k+1}$ are $\boldsymbol{\pi}_0$-precise. Hence it is possible to choose $\tau^{k+1} \in \bigcap_p \hat{\boldsymbol{A}}_p^{k+1}$, as justified by Definition 5, and we claim that we may in fact choose τ^{k+1} so that

$$-(\Phi^+ - \pi_0^+) \leq \tau^{k+1} - \tau^k \leq \Phi^- - \pi_0^-.$$

If this was not feasible, there would exist an interval $\hat{\boldsymbol{A}}_r^{k+1}$ of length π_0 with $\tau^{k+1} = \mathrm{left}(\hat{\boldsymbol{A}}_r^{k+1})$ or else $\tau^{k+1} = \mathrm{right}(\hat{\boldsymbol{A}}_r^{k+1})$ that satisfies $\tau^k \notin \hat{\boldsymbol{A}}_r^{k+1} + (\boldsymbol{\Phi} - \boldsymbol{\pi}_0) = \boldsymbol{\Phi}$. This, however, would contradict $\boldsymbol{\Phi}$-correctness of $\boldsymbol{A}_r^{k+1}$.

The bound (34) is a simple consequence of the fact that internal global time progresses as real-time does during a round, so that the maximum deviation between internal global time and real-time remains the same during any round. Therefore, we just have to add up the worst case internal global time "jumps" at each round. The initial deviation (in round 0) is zero since choosing $\tau^0(t) = t$ is legitimate due to the initial synchronization assumption in Definition 4. Hence, to complete the proof of (34), it only remains to add the maximum deviation between τ^{k+1} and the reference point T_q^{k+1} of $\boldsymbol{A}_q^{k+1}$, which is trivial since the latter is $\boldsymbol{\pi}_0$-correct.

To derive expression (35) for the rate of the synchronized clocks, we multiply (34) by -1 to arrive at

$$t_q^{R,k} - t_q^0 \in T_q^{k+1} - T_q^0 - (t_q^0 - T_q^0) + \boldsymbol{\pi}_0 + (k+1)(\boldsymbol{\Phi} - \boldsymbol{\pi}_0). \tag{36}$$

From the initial synchronization assumptions in item (0) of Definition 4, we gather $t_q^0 - T_q^0 \in \boldsymbol{\alpha}_q^0 \subseteq \boldsymbol{\pi}_0$. Moreover, from step (T) of the algorithm in conjunction with the fact that the maximum clock adjustment was shown to satisfy $\Upsilon_q \in \boldsymbol{\pi}$ in item (3) of this theorem, we obtain $T_q^{k+1} - T_q^0 = (k+1)P + \mathcal{O}(\pi)$. Plugging this into (36), we find

$$\frac{t_q^{R,k} - t_q^0}{T_q^{k+1} - T_q^0} \in 1 + \frac{[-\pi_0, \pi_0] + (k+1)(\boldsymbol{\Phi} - \boldsymbol{\pi}_0)}{(k+1)P + \mathcal{O}(\pi)}.$$

Taking the limit for $k \to \infty$ eventually provides (35) and completes the proof of our theorem. □

5 Final Remarks

We provided a few modifications and extensions of our generic analysis of interval-based clock synchronization conducted in [SS97]. Distinguishing features of our framework are conceptual beauty and high flexibility w.r.t. incorporating non-standard system parameters like clock granularity, broadcast latencies, etc.

Those advantages are primarily a consequence of our notion of internal global time and $\boldsymbol{\pi}$-accurateness, which allows us to reason about precision by considering each local interval clock separately, i.e., without explicitly relating it to the other clocks in the system. Even more, there is no need to consider the "absolute position" of intervals, i.e., local clock values, at all. In fact, any information required is provided in the interval of accuracies $\boldsymbol{\alpha}$ resp. in the associated $\boldsymbol{\pi}$-precision interval of an interval $\boldsymbol{I}(t) = [T \pm \boldsymbol{\alpha}]$, since all (non-faulty) accuracy intervals must contain real-time t resp. internal global time τ by construction, which thus serves as a "common reference" for relating different intervals. Of course, the particular reference point may lie anywhere in $[t - \alpha^+, t + \alpha^-]$ resp. $[\tau - \pi^+, \tau + \pi^-]$, according to the actually experienced clock drift, transmission delay, and initial accuracy, but there is no need for dealing with it explicitly.

References

[BI96] R. R. Brooks and S. S. Iyengar. Robust distributed computing and sensing algorithms. *IEEE Computer*, pages 53–60, June 1996.

[DB93] Rogério Drummond and Özalp Babaoğlu. Low-cost clock synchronization. *Distributed Computing*, 6:193–203, 1993.

[DHS86] Danny Dolev, Joseph Y. Halpern, and H. Raymond Strong. On the possibility and impossibility of achieving clock synchronization. *Journal of Computer and System Sciences*, 32:230–250, 1986.

[FC95] Christof Fetzer and Flaviu Cristian. An optimal internal clock synchronization algorithm. In *Proc. 10th Annual IEEE Conference on Computer Assurance*, Gaithersburg, MD, June 1995.

[HSS98] Martin Horauer, Ulrich Schmid, and Klaus Schossmaier. NTI: A Network Time Interface M-Module for high-accuracy clock synchronization. In *Proc. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'98)*, pages 1067–1076, Orlando, Florida, March-April 1998.

[Lam87] Leslie Lamport. Synchronizing time servers. Technical Report 18, Digital System Research Center, 1987.

[Lis93] Barbara Liskov. Practical uses of synchronized clocks in distributed systems. *Distributed Computing*, 6:211–219, 1993.

[LWL88] Jennifer Lundelius-Welch and Nancy A. Lynch. A new fault-tolerant algorithm for clock synchronization. *Information and Computation*, 77(1):1–36, 1988.

[Mar84] Keith A. Marzullo. *Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service*. PhD dissertation, Stanford University, Department of Electrical Engineering, February 1984.

[Mar90] Keith A. Marzullo. Tolerating failures of continuous-valued sensors. *ACM Transactions on Computer Systems*, 8(4):284–304, 1990.

[Mil95] David L. Mills. Improved algorithms for synchronizing computer network clocks. *IEEE Transactions on Networks*, pages 245–254, June 1995.

[MS85] Stephen R. Mahaney and Fred B. Schneider. Inexact agreement: Accuracy, precision, and graceful degradation. In *Proc. 4th ACM Symposium on Principles of Distributed Computing*, pages 237–249, Minaki, Canada, August 1985.

[OSF92] OSF. *Introduction to OSF DCE*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[RSB90] Parameswaran Ramanathan, Kang G. Shin, and Ricky W. Butler. Fault-tolerant clock synchronization in distributed systems. *IEEE Computer*, 23(10):33–42, October 1990.

[Sch86] Fred B. Schneider. A paradigm for reliable clock synchronization. In *Proc. Advanced Seminar of Local Area Networks*, pages 85–104, Bandol, France, April 1986.

[Sch87] Fred B. Schneider. Understanding protocols for byzantine clock synchronization. Technical Report 87-859, Cornell University, Department of Computer Science, August 1987.

[Sch95] Ulrich Schmid. Synchronized universal time coordinated for distributed real-time systems. *Control Engineering Practice*, 3(6):877–884, 1995. (Reprint from Proc. IFAC Workshop on Real-Time Programming WRTP'94, Lake Reichenau/Germany, 1994, p. 101–107.).

[Sch97a] Ulrich Schmid. Interval-based clock synchronization with optimal precision. Technical Report 183/1-78, Department of Automation, Technische Universität Wien, July 1997. (submitted).

[Sch97b] Ulrich Schmid. Orthogonal accuracy clock synchronization. Technical Report 183/1-77, Department of Automation, Technische Universität Wien, March 1997. (submitted).

[Sch97c] Ulrich Schmid, editor. *Special Issue on The Challenge of Global Time in Large-Scale Distributed Real-Time Systems*, J. Real-Time Systems 12(1–3), 1997.

[SLWL90] Barbara Simons, Jennifer Lundelius-Welch, and Nancy Lynch. An overview of clock synchronization. In Barbara Simons and A. Spector, editors, *Fault-Tolerant Distributed Computing*, pages 84–96. Springer Verlag, 1990. (Lecture Notes on Computer Science 448).

[SS97] Ulrich Schmid and Klaus Schossmaier. Interval-based clock synchronization. *J. Real-Time Systems*, 12(2):173–228, March 1997.

[SSHL97] Klaus Schossmaier, Ulrich Schmid, Martin Horauer, and Dietmar Loy. Specification and implementation of the Universal Time Coordinated Synchronization Unit (UTCSU). *J. Real-Time Systems*, 12(3):295–327, 1997.

[ST87] T. K. Srikanth and Sam Toueg. Optimal clock synchronization. *Journal of the ACM*, 34(3):626–645, July 1987.

[VR92] Paulo Veríssimo and Luís Rodrigues. A posteriori agreement for fault-tolerant clock synchronization on broadcast networks. In *Proc. 22nd International Symosium on Fault-Tolerant Computing*, Boston, Massachusetts, July 1992.

[VRC97] Paulo Veríssimo, Luís Rodrigues, and Antonio Casimiro. Cesiumspray: a precise and accurate global clock service for large-scale systems. *J. Real-Time Systems*, 12(3):243–294, 1997.

[YM93] Z. Yang and T. A. Marsland. Annotated bibliography on global states and times in distributed systems. *ACM SIGOPS Operating Systems Review*, pages 55–72, June 1993.

Glossary

Name	Meaning	Def.	Pg.
$\aleph^+(.)$, $\aleph^-(.)$	accuracy enhancement function	Def. 8	15
$\boldsymbol{\alpha} = [-\alpha^-, \alpha^+]$	interval of accuracies of $\boldsymbol{I} = [r \pm \boldsymbol{\alpha}]$	Asm. 3	6
$\alpha = \alpha^- + \alpha^+$	length of $\boldsymbol{\alpha}$	Sec. 3	6
$\boldsymbol{\alpha}_p(t) = [-\alpha_p^-(t), \alpha_p^+(t)]$	interval of accurracies at node p	Asm. 3	4
$\boldsymbol{\alpha}_p^0 = [-\alpha_p^{0-}, \alpha_p^{0+}]$	initial accurracies at node p	Def. 4	9
$B \in \{1, 2\}$	indicator of pure broadcast network	Asm. 4	4
$C(t), C_p(t)$	ordinary clock (node p)	Asm. 2	3
$\boldsymbol{C}_p(t) = [C_p(t) \pm \boldsymbol{\alpha}_p(t)]$	local interval clock of node p	Asm. 3	4
$\boldsymbol{\mathcal{CV}}$	generic convergence function	Def. 8	15
δ_{pq}	deterministic part trans. delay p to q	Asm. 4	4
$\delta_{\max}, \delta_{\min}$	uniform bounds det. part trans. delay	Asm. 4	4
Δ	transmission delay compensation	Def. 4	9
$\boldsymbol{\varepsilon}_{pq} = [-\varepsilon_{pq}^-, \varepsilon_{pq}^+]$	transmission delay uncertainity p to q	Asm. 4	4
$\varepsilon_{pq} = \varepsilon_{pq}^- + \varepsilon_{pq}^+$	length of $\boldsymbol{\varepsilon}_{pq}$	Asm. 4	4
$\boldsymbol{\varepsilon}_{\max}$	max. transmission delay uncertainity	Asm. 4	4
$\varepsilon_{\max}$	length of $\boldsymbol{\varepsilon}_{\max}$	Asm. 4	4
G	clock granularity	Asm. 2	3
$\boldsymbol{G} = [0, G]$	positive granularity interval	Def. 2	7
$\boldsymbol{G\rho} = [0, G(1 + \rho_{\max}^+)]$	max. positive granularity interval	Def. 2	7
$\overline{\boldsymbol{G}} = [-G, 0]$	negative granularity interval	Def. 2	7
G_S	clock setting granularity	Asm. 2	3
G_A	accuracy transmission loss	Asm. 4	4
$\boldsymbol{2G}_A = [-G_A, G_A]$	accuracy transmission loss interval	Def. 3	8
γ_p	real-time exec. time bound for node p	Asm. 1	3
$\gamma_{\max}, \gamma_{\min}$	uniform real-time execution time bounds	Asm. 1	3
Γ_p	logical time exec. time bound for node p	Def. 4	9
$\Gamma_{\max}, \Gamma_{\min}$	uniform logical time exec. time bounds	Def. 4	9
$I_{t \neq \theta}$	indicator function of non-synchrony	Def. 2	7
$\Im_-(.)$, $\Im_+(.)$	intersection enhancement function	Def. 8	15
$\boldsymbol{I}$, $\boldsymbol{A}$	(accuracy) intervals	Sec. 3	6
$\overline{\boldsymbol{I}}$	swapped interval $[r \mp \boldsymbol{\alpha}]$	Equ. 7	7
$\hat{\boldsymbol{I}}$	$\boldsymbol{\pi}$-precision interval associated with $\boldsymbol{I}$	Def. 5	11
$\boldsymbol{I}_q^p$	interval of p rec. at q	Equ. 12	8
$\boldsymbol{\mathcal{I}}_p$	ordered set received intervals at p	Def. 4	9
ι_p	mutual (pairwise) intersection	Def. 8	15
$\lambda_{\max}$	real-time maximum broadcast latency	Asm. 4	4
$\Lambda_{\max}$	logical time maximum broadcast latency	Def. 4	9
n	number of nodes	Sec. 2	3
$\omega_{\max}$	real-time max. broadcast oper. delay	Asm. 4	4
$\Omega_{\max}$	log. time max. broadcast oper. delay	Def. 4	9
P	round period	Def. 4	9
$\boldsymbol{\Phi}(.)$	precision preservation function	Def. 8	15

Name	Meaning	Def.	Pg.
$\boldsymbol{\pi} = [-\pi^-, \pi^+]$	(generic) interval of precision	Def. 5	11
$\pi = \pi^- + \pi^+$	length of $\boldsymbol{\pi}$	Def. 5	11
$\boldsymbol{\pi}_0 = [-\pi_0^-, \pi_0^+]$	ideal initial precision	Thm. 1	17
$\boldsymbol{\pi}^H = [-\pi^{H-}, \pi^{H+}]$	uniform precision exchanged intervals	Thm. 1	17
$\boldsymbol{\pi}_I$	uniform precision perceptions for s	Thm. 1	17
$\boldsymbol{\pi}^p$	precision perceptions of p	Thm. 1	17
$\boldsymbol{\pi}^{\overline{p}}$	precision perception of p at p	Thm. 1	17
$\boldsymbol{\pi}_q^p$	precision interval p received at q	Thm. 1	17
$\boldsymbol{\pi}$-accurate	correct w.r.t. t and $\boldsymbol{\pi}$-precise	Def. 5	11
$\boldsymbol{\pi}$-correct	correct w.r.t. both t and τ	Def. 6	12
$\boldsymbol{\pi}$-precise	precise intervals	Def. 5	11
$\Pi(.)$	precision enhancement function	Def. 8	15
ref($\boldsymbol{I}$)	reference point of interval $\boldsymbol{I}$	Sec. 3	6
$\boldsymbol{\rho}_p = [-\rho_p^-, \rho_p^+]$	inverse rate deviation bound node p	Asm. 2	3
$\rho_p = \rho_p^- + \rho_p^+$	length of $\boldsymbol{\rho}_p$	Asm. 2	3
$\boldsymbol{\rho}_{\max} = [-\rho_{\max}^-, \rho_{\max}^+]$	uniform inverse rate deviation bound	Asm. 2	3
$\rho_{\max} = \rho_{\max}^- + \rho_{\max}^+$	length of $\boldsymbol{\rho}_{\max}$	Asm. 2	3
T_{pq}^A, t_{pq}^A	log., real-time transmission p to q	Def. 4	9
T_p^R, t_p^R	log., real-time resynchronization at p	Def. 4	9
T_q^p, t_q^p	log., real-time reception of p at q	Def. 4	9
τ, τ^k	internal global time (of round k)	Def. 6	12
θ	real-time where oscillator tick occurs	Asm. 2	3
$\boldsymbol{u}_p = [-u_p^-, u_p^+]$	rate correction uncertainity node p	Asm. 2	3
$\boldsymbol{u}_{\max} = [-u_{\max}^-, u_{\max}^+]$	uniform rate correction uncertainity	Asm. 2	3
Υ_q	max. correction non-faulty clock q	Thm. 1	17