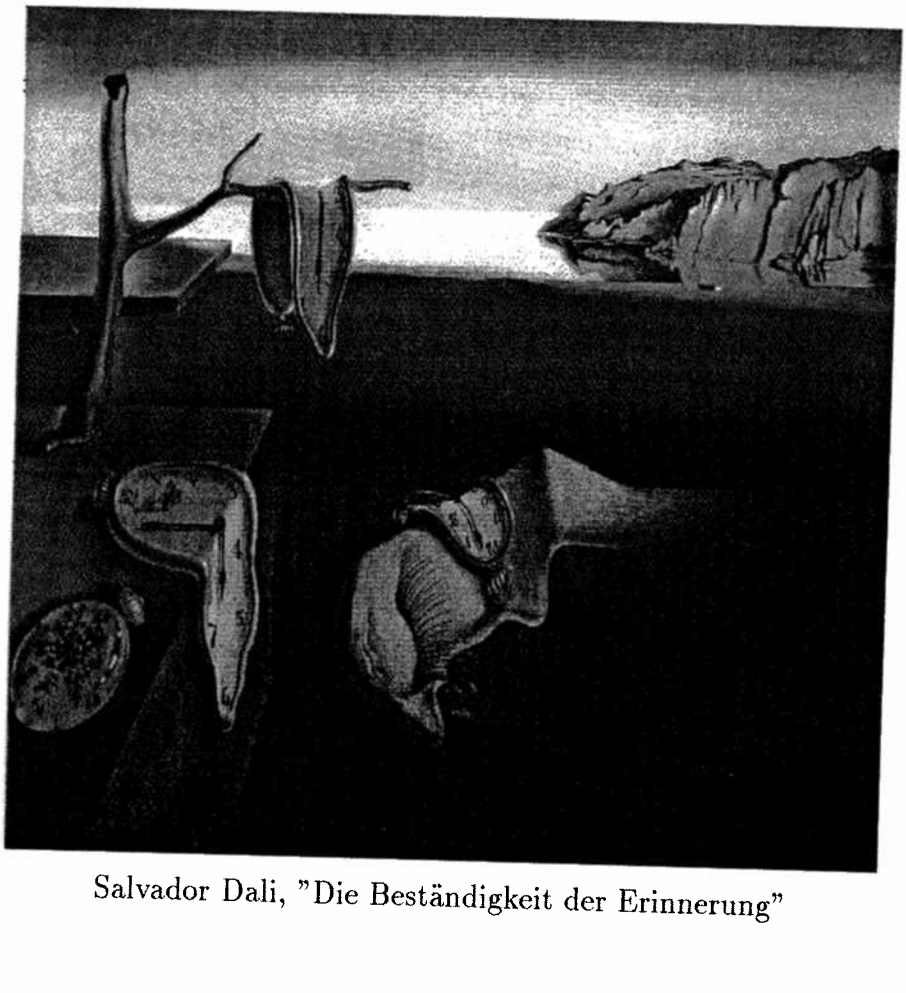
Institut für Automation
Abt. für Automatisierungssysteme

Technische
Universität
Wien

Projektbericht Nr. 183/1-64
December 1995

# Interval-Based Clock Synchronization

*Ulrich Schmid and Klaus Schossmaier*



Salvador Dali, "Die Beständigkeit der Erinnerung"

Special Issue on the Challenge of Global Time in Large-Scale Distributed Real-Time Systems, , 1–59 ()


# Interval-based Clock Synchronization *

ULRICH SCHMID s@auto.tuwien.ac.at
KLAUS SCHOSSMAIER kmschoss@auto.tuwien.ac.at
*Dept. of Automation, Technical University of Vienna, Treitlstraße 3, A-1040 Vienna, Austria*

Editor: W.A. Halang

**Abstract.** In this paper, we develop and analyze a simple interval-based algorithm suitable for fault-tolerant external clock synchronization. Unlike usual internal synchronization approaches, our convergence function-based algorithm provides approximately synchronized clocks maintaining both precision and accuracy w.r.t. external time. This is accomplished by means of a time representation relying on intervals that capture external time, providing accuracy information encoded in interval lengths. The algorithm, which is generic w.r.t. the convergence function and relies on either instantaneous correction or continuous amortization for clock adjustment, is analyzed by utilizing a novel, interval-based framework for establishing worst-case precision and accuracy bounds subject to a fairly detailed system model. Apart from individual clock rate and transmission delay bounds, our system model incorporates non-standard features like clock granularity and broadcast latencies as well. Relying on a suitable notion of internal global time, our analysis unifies treatment of precision and accuracy, ending up in striking conceptual beauty and expressive power.

**Keywords:** external clock synchronization, fault-tolerant distributed real-time systems, universal time coordinated (UTC), convergence functions, generic precision analysis, accuracy intervals, clock granularity, continuous amortization.

## 1. Introduction and Overview

One important problem that needs to be addressed when dealing with distributed real-time systems is the issue of global time. To support very large —world-wide— distributed applications, like automatic (instrument) landing systems (ILS), for example, each computing node in the system should have local access to a reliable system time satisfying the following two application requirements:

(R1) **Accuracy** (= maximum deviation of local clock reading from real-time)

Time rules daily life and for that reason most commercial computer applications, e.g. a flight reservation system. Hence, system time must have a well-defined relation to the only official and legal world-wide standard *Universal Time Coordinated* (UTC). It is made publicly available world-wide primarily by means of radio transmission, most notably by the NAVSTAR *Global Positioning System* (GPS), which has changed the world of accurate time and position measurement completely, see [3] in this special issue for an up-to-date overview.

* This work is part of our project SynUTC, supported by the Austrian Science Foundation (FWF) under contract no. P10244-ÖMA. Further project information may be found under http://www.auto.tuwien.ac.at/~kmschoss/synutc.html

Providing an accurate global time in distributed systems is usually termed as the *external clock synchronization problem*, due to the fact that UTC has to be provided externally to the system. The probably most well-known solution is the *Network Time Protocol* NTP, which was designed to establish a global time related to UTC in the Internet, see [12], [13]. NTP provides its clients with an average accuracy below 10 $ms$, which amply fulfills the modest accuracy requirements of typical applications.

(R2) **Precision** (= maximum difference of simultaneous local clock readings)

Algorithms for (fault-tolerant) distributed systems are usually considerably simplified and improved when approximately synchronized local clocks are available, see [7], [20] for some examples. Distributed real-time systems depend upon precise global time even at a very low level of operation. For example, timestamping is often employed for establishing a global order of (external) events occurring at different nodes.

Providing mutually synchronized local clocks is known as the *internal clock synchronization problem*, and numerous solutions have been worked out (at least in scientific research) under the term *fault-tolerant clock synchronization*, see e.g. [25] for an overview. In fact, there are more than 60 papers listed in the 1993 bibliography [27] of clock synchronization in distributed systems. The actual precision requirements of typical applications are in the range below 1 $ms$, although increasingly demanding applications like an airborne flight control system supporting ILS will certainly push these requirements down to the $\mu s$-range. Aiming at that high precision is also advantageous w.r.t. improving the performance of most distributed algorithms.

Different components (i.e., applications) within a heterogeneous distributed system have usually different requirements concerning accuracy and/or precision. However, it should be clear that a "uniform" global time that satisfies both requirements simultaneously is preferable over a solution that provides "translations" between parts of the system employing their own idea of precise/accurate time, cf. [6].

Unfortunately, it turns out that establishing a precise global time that also relates to some external time standard like UTC in fault-tolerant distributed systems is not a simple matter of combining techniques from (R1) and (R2). Informally, it is difficult to add accuracy to existing solutions for internal synchronization, since such algorithms are necessarily reluctant to obey "authoritative" information of a few UTC time sources due to fault-tolerance. On the other hand, whereas high precision is of course implied by high accuracy, it is usually not feasible to build a fault-tolerant time service that is purely based on accuracy, since highly accurate UTC is not continuously available.

Although it has been recognized early that the problem of *fault-tolerant external clock synchronization* constitutes a research topic in its own right, cf. [2], it did not receive much attention until recently. [18] provides a short overview of existing earlier work, and the papers in this special issue form a quite representative collection of more recent efforts. Among the latter is our research on *interval-based clock validation* (introduced in [17]), which aims at a solution of the external clock synchronization problem for large-scale, fault-tolerant real-time systems. Clock validation algorithms are based on the idea of verifying whether highly accurate but possibly faulty "authoritative time" provided by UTC time sources is consistent with some less accurate but reliable "validation time" formed by exchanging information of all local clocks in the system. If so, the distinguished time is accepted, otherwise, it is discarded and the nodes rely on the validation time instead.

There might of course be phases of unavailability of accurate time information from the UTC time sources. For clock validation, this means that the system automatically undergoes a transition to internal synchronization, and a transition back to normal operation when UTC is available again. This *flywheel operation* implies, however, that the clock synchronization algorithm (employed for computing the validation time) must not only ensure precision but should maintain high accuracy as well.

When system time must have a defined relation to external time, there is a promising alternative to the "one-dimensional" point of view sufficient for internal synchronization, viz. the *interval-based paradigm* introduced in Marzullo's thesis [9]; see also [11]. Interval-based algorithms represent time information relating to an external standard like UTC by intervals that are known (better to say supposed) to contain UTC. Given a set of such intervals from different sources, a usually smaller interval that actually contains UTC may be determined, even if some of the source intervals are faulty.

We consider the interval-based paradigm as being exceptionally suitable for dealing with fault-tolerant external clock synchronization. Since accuracy is maintained dynamically ("on-line") here, it provides an average case behavior that is much better than the worst case one. By contrast, worst-case accuracy bounds for internal synchronization algorithms are necessarily static in nature, allowing no improvement w.r.t. average case at all. Surprisingly enough, however, interval-based approaches did not receive much attention in research. To our knowledge, there is only Lamport's technical report [6], Marzullo's work [10] on replicated sensors, and our paper [18] introducing interval-based clock validation that further exploit ideas of [9]. However, it is worth mentioning that both DTS, the *Digital Time Synchronization Service* of OSF/DCE, and newer versions of NTP are built upon the interval-based paradigm, see [14] and [13].

This paper provides description and analysis of a simple interval-based algorithm suitable for being used in clock validation, which is generic w.r.t. the convergence function employed. Thus, our results are given in terms of some characteristic parameters of the convergence function, cf. [15]. In order to determine precision and accuracy of a particular instance of our algorithm, it only remains to determine the characteristic parameters of the particular convergence function, and to plug them into the generic results. Lacking space prohibited us from including a sample analysis in this paper; consult [19] for this purpose.

The outline of the rest of the paper is as follows: Section 2 introduces the interval-based paradigm, focussing on both accuracy and precision intervals. The system model for processors, local clocks, and communications in conjunction with a discussion of the (generic) fault model is contained in Section 3. Section 4 eventually provides our clock synchronization algorithm and outlines some of its properties. The generic analysis of precision and accuracy, along with the definition of internal global time and the treatment of continuous amortization, is given in Section 6. Finally, some conclusions and directions of further research are appended in Section 7.

## 2. The Interval-Based Paradigm

To introduce the interval-based paradigm, we have to establish some basic notation and operations on intervals first. We use bold letters like $\boldsymbol{I}$ to denote a real interval $\boldsymbol{I} = [x, y]$, $x \leq y$, with lower and upper *edge* $x$ and $y$, respectively; 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*. More generally, given some $\boldsymbol{\pi} = [-\pi^-, \pi^+]$ with $\pi^-, \pi^+ \geq 0$ and $\pi^- + \pi^+ = \pi$, the (asymmetric) $\boldsymbol{\pi}$-center —characterizing a "centerpoint" according to the proportion of $\pi^- : \pi^+$— of an interval $\boldsymbol{I}$ reads $\boldsymbol{\pi}\text{-center}(\boldsymbol{I}) = \frac{x\pi^+ + y\pi^-}{\pi}$. The *sum* of two intervals is defined by $[x, y] + [u, v] = [x + u, y + v]$, the *scalar product* by $b \cdot [x, y] = [bx, by]$ for $b \geq 0$, 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 $[x, y] \cup [u, v] = [\min\{x, u\}, \max\{y, v\}]$, even valid for $[x, y] \cap [u, v] = \emptyset$.

Most of the intervals encountered in our setting contain a distinguished *reference point* that partitions the interval in a *negative* and a *positive accuracy*. To that end, we introduce the notation

$$\boldsymbol{I} = [r \pm \boldsymbol{\alpha}] = [r - \alpha^-, r + \alpha^+] = [x, y]_r, \tag{1}$$

where

$$\begin{aligned}
r &= \text{ref}(\boldsymbol{I}) && \boldsymbol{I}\text{'s reference point (also called midpoint)},\\
\alpha^+ &= \text{acc}^+(\boldsymbol{I}) \geq 0 && \boldsymbol{I}\text{'s positive (upper) accuracy},\\
\alpha^- &= \text{acc}^-(\boldsymbol{I}) \geq 0 && \boldsymbol{I}\text{'s negative (lower) accuracy},\\
x &= \text{left}(\boldsymbol{I}) = r - \alpha^- && \boldsymbol{I}\text{'s lower edge (envelope)},\\
y &= \text{right}(\boldsymbol{I}) = r + \alpha^+ && \boldsymbol{I}\text{'s upper edge (envelope)}.
\end{aligned}$$

Note that we usually suppress the subscript $r$ in intervals of the form $[r - \alpha^-, r + \alpha^+]_r$ for brevity.

When referring to an interval $[r \pm \boldsymbol{\alpha}]$, $\boldsymbol{\alpha} = [-\alpha^-, \alpha^+]$ denotes its *interval of accuracies* and $\alpha = |\boldsymbol{\alpha}| = \alpha^+ + \alpha^-$ this interval's length. For intervals $\boldsymbol{I} = [r \pm \boldsymbol{\alpha}]$ and $\boldsymbol{J} = [s \pm \boldsymbol{\beta}]$, we have $|\boldsymbol{I}| = \alpha^+ + \alpha^- = \alpha$, $\text{center}(\boldsymbol{I}) = r + (\alpha^+ - \alpha^-)/2$,

$I + J = [r + s \pm \gamma]$ where $\gamma = \alpha + \beta = [-(\alpha^- + \beta^-), \alpha^+ + \beta^+]$, $I + a = [r + a \pm \alpha]$ for an arbitrary scalar $a$, and $bI = [br \pm \mu]$ with $\mu = b\alpha = [-b\alpha^-, b\alpha^+]$ for any scalar $b \geq 0$. There is also a notation to express intervals obtained from (1) by *swapping* the positive and negative accuracy, namely

$$\overline{I} = \overline{[r \pm \alpha]} = [r \mp \alpha] = [r - \alpha^+, r + \alpha^-] = [r \pm \overline{\alpha}] \tag{2}$$

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

### 2.1. Accuracy Intervals

The core idea of the interval-based paradigm introduced in [9] is to represent time information by a time-dependent *accuracy interval*. More specifically, an accuracy interval $\boldsymbol{A} = \boldsymbol{A}(t)$ representing real-time $t$ is an interval satisfying $t \in \boldsymbol{A}$. Accuracy intervals are primarily provided by *interval clocks*, which are interval-valued functions $\boldsymbol{C}(t) = [L(t), U(t)]$ with the edges $L(t)$ and $U(t)$ forming lower and upper *envelope*, respectively, of real-time $t$. In practice, interval clocks are implemented as an ordinary clock $C(t)$ in conjunction with a time-dependent interval $\boldsymbol{\alpha}(t) = [-\alpha^-(t), \alpha^+(t)]$ of *accuracies* taken relatively to the clock's value, hence $\boldsymbol{C}(t) = [C(t) - \alpha^-(t), C(t) + \alpha^+(t)]$.

DEFINITION 1 (INTERVAL RELATIONS) *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 both 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 $t \in \boldsymbol{I}$.*

Note that two compatible and accurate accuracy intervals are consistent, whereas two compatible consistent accuracy intervals $\boldsymbol{I}, \boldsymbol{J}$ are not necessarily accurate since possibly $t \notin \boldsymbol{I} \cap \boldsymbol{J}$. Moreover, consistency is not transitive in general since non-empty intersections $\boldsymbol{I} \cap \boldsymbol{J}$ and $\boldsymbol{J} \cap \boldsymbol{K}$ do not imply $\boldsymbol{I} \cap \boldsymbol{K} \neq \emptyset$. However, we have the following lemma:

LEMMA 1 (CONSISTENCY AND INTERSECTION) *If $n \geq 2$ compatible accuracy intervals $\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n$ are* mutually consistent *in the sense that they are all pairwise consistent, then $\bigcap_{j=1}^{n} \boldsymbol{I}_j \neq \emptyset$.*

Proof: Using induction, we first have $\boldsymbol{I}^1 = \boldsymbol{I}_1 \neq \emptyset$, and provided that $\boldsymbol{I}^{k-1} = \bigcap_{i=1}^{k-1} \boldsymbol{I}_i \neq \emptyset$ we conclude non-emptiness for $\boldsymbol{I}^k$: Assuming the contrary, $\boldsymbol{I}_k$ would lie entirely left (or right) of $\boldsymbol{I}^{k-1}$ so that it cannot be consistent with the interval $\boldsymbol{I}_l$, $1 \leq l \leq k-1$ whose left (right) edge delimits $\boldsymbol{I}^{k-1}$. □

### 2.2. Precision Intervals

Apart from the requirement of being accurate, a property of a single accuracy interval, we also have to deal with the precision requirement that applies to (the reference points of) a set of accuracy intervals. In the traditional framework, a set of ordinary clocks is called *precise* with *precision* $\pi$ during some time interval $D$, iff $|C_p(t) - C_q(t)| \leq \pi$ for $t \in D$ (and clocks progress linearily with $t$). In our setting, precision can easily be added to accuracy, since none of the interval relations given in Definition 1 involves the reference point; it is not required for accuracy purposes and can in principle be set to any point within the accuracy interval.

Unlike traditional approaches, we utilize a definition of precision that is based on intervals, inspired by the following lemma:

LEMMA 2 (PRECISION EQUIVALENCE) *Given some* $\boldsymbol{\pi} = [-\pi^-, \pi^+]$ *with* $\pi^-, \pi^+ \geq 0$ *and* $\pi = |\boldsymbol{\pi}| = \pi^- + \pi^+$, *and* $n \geq 2$ *non-negative real numbers* $r_1, \ldots, r_n$, *we have*

$$|r_i - r_j| \leq \pi \quad \text{for all } i,j \qquad \Longleftrightarrow \qquad \bigcap_{j=1}^{n} [r_j \pm \boldsymbol{\pi}] \neq \emptyset.$$

**Proof:** To show direction $\Leftarrow$, we note that there is some $t' \in \bigcap_{j=1}^n [r_j \pm \boldsymbol{\pi}]$ so that immediately $\pi \geq \max_{1 \leq i \leq n}(r_i) - \min_{1 \leq j \leq n}(r_j) \geq |r_i - r_j|$ for all $i, j$. The other direction follows from Lemma 1, since $|r_i - r_j| \leq \pi$ for all $i, j$ implies that the intervals $[r_j \pm \boldsymbol{\pi}]$ are mutually consistent. □

DEFINITION 2 (PRECISION INTERVALS) *Given* $\boldsymbol{\pi} = [-\pi^-, \pi^+]$ *with* $\pi^-, \pi^+ \geq 0$ *and* $\pi = |\boldsymbol{\pi}| = \pi^- + \pi^+$, *and a set of* $n \geq 2$ *compatible accuracy intervals* $\mathcal{I} = \{I_1, \ldots, I_n\}$ *with* $I_j = [r_j \pm \alpha_j]$, *the* $\pi$-precision interval $\hat{I}_j$ *associated with* $I_j$ *is defined as*

$$\hat{I}_j = [r_j \pm \boldsymbol{\pi}].$$

*The set* $\mathcal{I}$ *is called* $\pi$*-precise iff* $\bigcap_{j=1}^n \hat{I}_j \neq \emptyset$.

Keep in mind that the associated $\pi$-precision interval $\hat{I}$ of an accuracy interval $I$ is not separately maintained, but rather computed from the reference point of $I$. Thus, precision and accuracy are orthogonal issues here. Note also that we cannot safely assume $\hat{I}_p \subseteq I_p$ in case of small accuracies.

Our definition of $\pi$-precision has several immediately apparent consequences, most importantly, that $\pi$-precision implies precision $\pi$:

LEMMA 3 ($\pi$-PRECISION VS. PRECISION) *Given a* $\pi$*-precise set* $\mathcal{I} = \{I_1, \ldots, I_n\}$ *of* $n \geq 2$ *compatible accuracy intervals* $I_j = [r_j \pm \alpha_j]$ *and* $\pi = |\boldsymbol{\pi}|$, *then*

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

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

**Proof:** The first assertion follows from the fact that $\hat{I}_i$ and $\hat{I}_j$ are consistent and that $|\hat{I}_l| \leq \pi$ for $1 \leq l \leq n$, according to the definition of the associated $\pi$-precision intervals. The second assertion is an immediate consequence of Definition 2 and Lemma 2. □

LEMMA 4 (PRECISION BY ACCURATENESS) *If the* $n \geq 2$ *compatible accuracy intervals* $\mathcal{I} = \{I_1, \ldots, I_n\}$ *with* $I_j = [r_j \pm \alpha_j]$ *are accurate and* $\alpha_j^- \leq \alpha^-$, $\alpha_j^+ \leq \alpha^+$ *for all* $j$, *then* $\mathcal{I}$ *is also* $\pi$*-precise for any* $\pi \supseteq [-\alpha^-, \alpha^+]$.

**Proof:** Using $\pi \supseteq [-\alpha^-, \alpha^+]$ in Definition 2, we have $\hat{I}_j \supseteq I_j$ and the statement of the lemma follows from $t \in \bigcap_{j=1}^n I_j \neq \emptyset$. □

LEMMA 5 (COMPOSITION OF PRECISIONS) *Let* $\mathcal{I} = \{I_1, \ldots, I_n\}$ *and* $\mathcal{J} = \{J_1, \ldots, J_m\}$ *be two sets of compatible accuracy intervals that are* $\pi$*-precise and* $\pi'$*-precise, respectively. If* $\left(\bigcap_{j=1}^n \hat{I}_j\right) \cap \left(\bigcap_{i=1}^m \hat{J}_i\right) \neq \emptyset$, *then the set* $\mathcal{I} \cup \mathcal{J}$ *is* $\pi \cup \pi'$*-precise.*

**Proof:** Since the $\pi$-precise set $\mathcal{I}$ and the $\pi'$-precise set $\mathcal{J}$ are also $\pi \cup \pi'$-precise, the statement follows immediately from Definition 2. □

Our definition of $\pi$-precision is a key issue in our novel interval-based framework for precision analysis. Nevertheless, there are only a few occasions where we actually face $\pi$-precise intervals according to Definition 2. In most cases, we employ the slightly stronger predicate of *$\pi$-correctness* as provided in Definition 3. It characterizes the $\pi$-precision interval $\hat{I}$ associated with an accuracy interval $I$ as being accurate w.r.t. an appropriately defined *internal global time* $\tau = \tau(t)$. Actually, $I = I(t)$ is called $\pi$-correct iff both $\tau \in \hat{I}$ and $t \in I$, in other words, iff $I$ is accurate w.r.t. both $\tau$ and $t$, as shown in Figure 1.

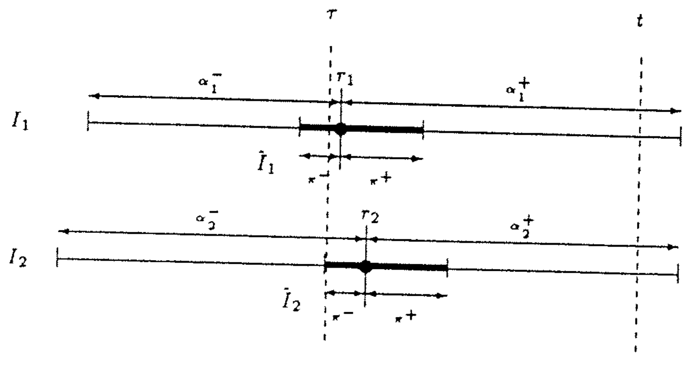



*Figure 1.* Accuracy and Precision Intervals

At this point there is no need to elaborate on how internal global time is actually maintained; consult Section 5.1 for details. Intuitively, we exploit the fact that our clock synchronization algorithm maintains the set $\mathcal{C} = \mathcal{C}(t)$ of non-faulty interval clocks $C_p(t)$ so that it is $\pi_0$-precise at (periodic) resynchronization real-times $t^{R,k}$, characterizing the beginning of the $k+1$-st *round* ($k \geq 0$). This property allows us to define round $k$'s unique internal global time $\tau^k = \tau^k(t)$ by $\tau^k(t) = \tau^k(t^{R,k}) + (t - t^{R,k})$ for any real-time $t$; the "fixed point" $\tau^k(t^{R,k})$ is some arbitrary value satisfying $\tau^k(t^{R,k}) \in \bigcap_{C_j \in \mathcal{C}} \hat{C}_j(t^{R,k}) \neq \emptyset$. For large accuracies, it is likely that $\tau^k(t) \neq t$, although Lemma 4 justifies that choosing $\tau(t) = t$ is possible if accuracies are sufficiently small (i.e., $\pi > \alpha$). However, internal global time of any round progresses as real-time does, so that it can be used interchangeably with real-time if one is interested in measuring durations only.

DEFINITION 3 ($\pi$-CORRECTNESS) *For* $\pi = [-\pi^-, \pi^+]$ *with* $\pi^-, \pi^+ \geq 0$,

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

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

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

LEMMA 6 (RELATION $T$ AND $\tau$) *If the intervals* $I_1(t_1) = [T_1 \pm \alpha_1]$ *and* $I_2(t_2) = [T_2 \pm \alpha_2]$ *are* $\pi_1$*-accurate and* $\pi_2$*-accurate (w.r.t. internal global time of the same round), respectively, and* $\tau_1 = \tau(t_1)$, $\tau_2 = \tau(t_2)$, *then* $\tau_1 - \tau_2 = t_1 - t_2 \in T_1 - T_2 + \pi_1 + \overline{\pi}_2$.

**Proof:** Since, for $i = 1, 2$, $\tau_i \in \hat{I}_i(\tau_i)$ according to the asserted $\pi_i$-correctness of $I_i$, we have $T_i - \pi_i^- \leq \tau_i \leq T_i + \pi_i^+$. Subtracting those inequalities and recalling the notion of swapped intervals easily provides the statement of the lemma; note that $\tau_1 - \tau_2 = t_1 - t_2$ since internal global time (of the same round) progresses as real-time does. □

In the following we will frequently employ the abbreviation $\tau = \tau(t) = \tau^k(t)$ as above when the particular round $k$ is clear from the context. Moreover, we can usually unambiguously write $\hat{I} = \hat{I}(t) = \hat{I}(\tau^k)$ for $\tau^k = \tau^k(t)$, meaning that $\hat{I}$ represents $\tau^k(t)$ iff $I$ represents $t$.

The following lemma is a simple corollary of Lemma 6 for $t_1 = t_2 = t$:

LEMMA 7 (DISTANCE REFERENCE POINTS) *If the compatible intervals* $I_1(t) = [T_1 \pm \alpha_1]$ *and* $I_2(t) = [T_2 \pm \alpha_2]$ *are* $\pi_1$*-accurate and* $\pi_2$*-accurate (w.r.t. internal global time of the same round), respectively, then* $T_2 - T_1 \in \pi_1 + \overline{\pi}_2$. □

Note that this result, which is of central importance for precision analysis, can be viewed as a consequence of the fact that both $\hat{I}_1(\tau)$ and $\hat{I}_2(\tau)$ must contain $\tau$, so that $T_2 - T_1 \in \left[-(\pi_2^+ + \pi_1^-), \pi_2^- + \pi_1^+\right] = \pi_1 + \overline{\pi}_2$.

It should be obvious that a set of compatible $\pi$-correct intervals $\mathcal{I} = \{I_1, \ldots, I_n\}$ is $\pi$-precise due to $\tau \in \bigcap_{j=1}^n \hat{I}_j \neq \emptyset$. Therefore, Lemmas 3–5 are valid for $\pi$-correct sets as well. Of course, since our precision analysis is primarily based upon those lemmas, it would in principle be sufficient to deal with $\pi$-precise intervals. Introducing internal global time and $\pi$-accurateness, however, allows to reason about precision by considering each local interval clock separately, i.e., without explicitly relating it to the other clocks in the system, which greatly simplifies the analysis.

As a consequence, most of the intervals encountered in our analysis are $\pi$-correct ones, so that it does not make much sense to adhere to a strict separation of accuracy/precision terminology. In the remaining sections, will use *interval* as a standard term of generic meaning, while accuracy interval or precision interval is only used if we want to stress the particular "instance" of the interval in question.

## 3. System Modeling

In this section, we will provide 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 our discrete 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.

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

As we will see later on, all computations required for clock synchronization purposes are essentially periodic and require integer arithmetic only. As far as the execution speed of the processor executing the algorithm is concerned, we assume the following:

ASSUMPTION 1 (EXECUTION TIMES) *A single (periodic) computation required for clock synchronization purposes at a non-faulty node p is completed within $\gamma_p$ (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.*

Note that we do not make further assumptions about application tasks the processor might perform concurrently with clock synchronization; a fault-free node $p$ must solely guarantee the bound $\gamma_p$.

In the following two subsections, we will develop the system model for local clocks and for the communication subsystem without considering faults. The fault model is discussed in the last subsection.

## 3.1. Discrete Local Interval Clocks

The local clock of a node is assumed to be built upon a physical clock (usually driven by a quartz oscillator) of non-zero granularity $G$ (micro-)seconds, which allows adjustment of rate and state. Non-zero granularity implies that the clock is incremented by $G$ at discrete points in real-time (called *clock ticks*) only, posing a particular challenge to system modeling.

A clock is usually modeled as a monotonic function $C : t \to T$ mapping *real-time* $t$ to *logical time* $T = C(t)$. Note that we use the convention of writing upper-case names (like $T$) for logical time and lower-case ones (like $t$) for real-time values throughout the paper. Most often, $C(t)$ is assumed to be a continuous (differentiable) function, although existing clocks are modeled appropriately by discrete step-functions only. Up to our knowledge, however, discretization and the adverse effect of non-zero clock granularity has been investigated only in [24].

In our analysis, we employ an alternative discretization that is more suitable for the interval-based paradigm. Instead of considering clocks $T = C(t)$ in the first place, we start with *inverse clocks* $t = c(T)$ mapping logical time to real-time. More specifically, "inverting" the approach of [16], we assume that real-time $t$ advances instantaneously a (varying) real value $g$ at each logical tick of the clock, and remains constant everywhere else. Clock ticks take place every (fixed) $G > 0$ logical time seconds, modeling non-zero clock granularity.

$c(T)$ is of course only meaningful for $T = kG$ being a multiple of the clock granularity $G$. Unfortunately, when $c(T)$ is actually defined as the inverse of the step-function modeling clock $C(t)$, it is multi-valued (infinitely many values, representing the progress of real-time) at $T = kG$. We enforce a proper function, however, by defining $c(kG)$ to be the value before advancing the clock by $g$, i.e., $\lim_{T' \to T^-} c(T') = c(T) = \lim_{T' \to T^+} c(T') - g$ for $T = kG$. The following Figure 2 illustrates this.

Note that this definition of $c(T)$ calls for setting the clock $C(\theta)$'s value at real-time $\theta$, where a tick takes place, to the value after advancing $C$ by $G$, i.e., $\lim_{t' \to \theta^-} C(t') + G = C(\theta) = \lim_{t' \to \theta^+} C(t')$. This is due to the fact that $C(t)$ and $c(T)$ must be inverse at real-time $\theta$ where $C(t)$ ticks, thus $\theta = c(\Theta)$ for $\Theta = C(\theta)$. In fact, we assume that discrete local time approximates continuous local time by a leading (majorizing) step-function.

In practice, a local clock $C_p(t)$ is primarily characterized by its *intrinsic rate* $r_p = G/g$, which gives the amount of logical time seconds the clock advances per real-time second. More specifically, one usually assumes that there is some $\rho_p^+, \rho_p^- \ll 1$ —accounting for the clock's rate deviations due to oscillator frequency offset, aging, temperature dependency, noise, etc.— such that

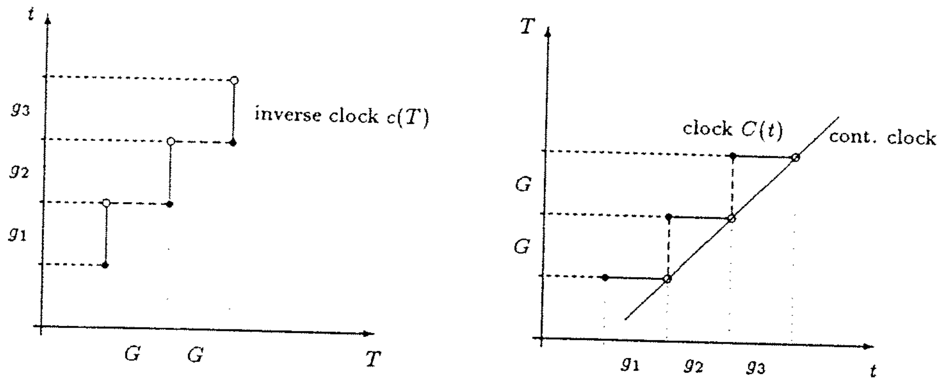



*Figure 2.* Discrete Clocks

$$1 - \rho_p^+ + \mathcal{O}(\rho_p^{+2}) = \frac{1}{1+\rho_p^+} \le r_p \le \frac{1}{1-\rho_p^-} = 1 + \rho_p^- + \mathcal{O}(\rho_p^{-2}). \tag{3}$$

In our clock model, we will employ the equivalent condition $1-\rho_p^- \le r_p^{-1} \le 1+\rho_p^+$ on the *intrinsic inverse rate* $r_p^{-1}$, i.e., the rate of the inverse clock, which gives the amount of real-time seconds the inverse clock $c_p(T)$ advances per logical time second.

Unfortunately, granularity $G$ and intrinsic inverse rate $r_p^{-1}$ are not sufficient to describe completely the behavior of discrete clocks. First, *rate adjustment uncertainities* are introduced when local clocks utilize an "artificial" rate generated by discrete rate adjustment techniques. Since it is difficult to fine-tune the frequency of an ordinary quartz oscillator (as opposed to a voltage controlled oscillator, where this is easy), techniques have been developed that allow rate adjustment of a clock by occasionally tampering with raw oscillator ticks. This type of clocks tick at the intrinsic rate most of the time, whether they are adjusted or not. However, when the accumulated deviation between intended and observed local time is about to exceed some bound $u$, the next tick is modified: If the clock is to be slowed down resp. speeded up, the next regular tick is delayed resp. advanced. Of course, this causes an additional uncertainity in the relation between logical time and real-time not explained by the intrinsic clock rate, which must be taken into account explicitly.

In addition, we have to account for the fact that practical clocks cannot be state adjusted with infinite resolution, but only with a certain *clock setting granularity* $G_S \le G$. Whereas $G_S = G$ is easily provided by making the clock register writable, it is considerably more expensive to implement fine-grained clock setting capabilities ($G_S < G$). Apart from employing state adjustment controlled by a continuous amortization algorithm (see Section 5.3), instantaneous state correction could be implemented directly by utilizing a clock register with higher (internal) resolution $G_S$ or, alternatively, by delaying/advancing the time of setting the clock.

In any case, $G_S$ should be considered as the "internal" granularity of the clock, as opposed to the (coarser) granularity $G$ available for external clock reading.

Note that fine-grained clock setting and discrete rate adjustment should not be considered independent of each other, as it might be the case in a sub-optimal clock design. More specifically, we assume that the error between intended and observed local time caused by rate adjustment is added to the initial clock setting error prior to deciding when oscillator ticks are to be modified. For example, in case of $u = G$, if a clock driven by a slow oscillator is set to $G + G/2$, an additional clock tick should be introduced as soon as the accumulated error of the subsequent clock ticks becomes $G/2$, otherwise the total error will exceed $u = G$ by then.

With these preparations, we are ready for stating our basic model of local clocks. It does not incorporate explicit rate adjustment capabilities required for continuous amortization, which are added in Section 5.3. However, rate adjustments may already be incorporated here for fine-tuning of the intrinsic rate.

Assumption 2 (Local Clocks) *Each node p is endowed with a discrete* local clock *$C_p(t)$, which increments by $G > 0$ (micro-)seconds at each clock tick and allows state adjustment with clock setting granularity of at least $G_S = G/K$ seconds, for some integer $K \geq 1$. In the absence of resynchronizations, intrinsic inverse rate and rate adjustment uncertainity of the clock of a non-faulty node p are such that*

$$(1 - \rho_p^-)(\Theta_i - \Theta_1) - u_p^- \leq \theta_i - \theta_1 \leq (1 + \rho_p^+)(\Theta_i - \Theta_1) + u_p^+ \tag{4}$$

*with $\Theta_j = C_p(\theta_j)$ is guaranteed for any sequence of $i \geq 2$ successive clock ticks $\theta_j$, $1 \leq j \leq i$. For $\rho_p = [-\rho_p^-, \rho_p^+]$ denoting the clock's intrinsic inverse rate deviation bounds (in $[\frac{sec}{Sec}]$) and $u_p = [-u_p^-, u_p^+]$ its maximum rate adjustment uncertainity (in [sec]), let $\rho_{\max} = [-\rho_{\max}^-, \rho_{\max}^+] \supseteq \bigcup_p \rho_p$ with $\rho_{\max} = \rho_{\max}^- + \rho_{\max}^+$ and $u_{\max} = [-u_{\max}^-, u_{\max}^+] \supseteq \bigcup_p u_p$ with $u_{\max} = u_{\max}^- + u_{\max}^+ = \mathcal{O}(G)$ be suitable uniform bounds for all (non-faulty) nodes p.*

Of course, inequality (4) is also valid for $i = 1$, although the bounds are not particularly meaningful, besides from the fact that $u_p$ can be used to account for the fractional clock setting value ($< G$) in case of fine-grained clock setting.

It seems appropriate here to sketch how conceivable clock implementations map to the above model, i.e., how $G$, $\rho_p$, $G_S$ and $u_p$ are to be chosen for a certain implementation. First of all, we note that $u_p^- = u_p^+$ for all discrete rate adjustment techniques we are aware of. More specifically, although $u_p^+$ bounds the logical vs. real-time deviation $\Delta$ caused by a fast clock (say, $\Delta > 0$), whereas $u_p^-$ is meaningful for a slow clock ($\Delta < 0$), it is apparent that both $\Delta > 0$ and $\Delta < 0$ occurs when slowing down or speeding up the clock via rate adjustment. In fact, the actual sign of the deviation $\Delta$ depends on whether a modified tick is at the beginning ($j = 1$) or beyond the end ($j > i$) of the sequence of ticks under consideration. Consequently, it is the maximum value of any scenario that determines $u_p^- = u_p^+$.

The particular clock models considered are as follows:

- *Counter Clock:* Implemented by means of a fixed-frequency oscillator that increments a counter by $G = 1/f_{osc}$ (usually, $G = 2^{-k}$ for binary counters or $G = 10^{-k}$ for decimal ones) at each tick, it follows that $\rho_p$ is the intrinsic inverse rate deviation bound $\rho_{osc}$ of the oscillator, $G_S = G$, and $u_p = [0, 0]$ since there are no rate adjustment capabilities.

- *Voltage Controlled Oscillator (VCO):* Replacing the fixed-frequency oscillator above by a VCO adds (continuous) rate adjustment capabilities, so that $G = 1/f_{osc}^0$ for the intrinsic (non-adjusted) oscillator frequency $f_{osc}^0$, $\rho_p = \rho_{osc}^0$, $u_p = [0, 0]$, and $G_S \ll G$ provided state correction is done by continuous amortization.

- *Tick Advancing/Delaying:* One technique for discrete rate adjustment, employed in the CSU of [5], is based on a fixed-frequency oscillator running at a multiple $f_{osc} = mf_{clock}$, $m \geq 2$, of the desired clock frequency $f_{clock} = 1/G$. Without rate adjustment, every $m$-th oscillator tick is used to increment the counter, so that $G = 1/f_{clock} = m/f_{osc}$. To speed up the clock, the $m - 1$-th oscillator tick is used occasionally, that is, when the accumulated deviation between intended and observed logical time is about to exceed $G/m = 1/f_{osc}$. Similarly, the $m + 1$-th oscillator tick is used occasionally if the clock is to be slowed down. Therefore, it is immediately apparent that $\rho_p = \rho_{osc}$, $G_S = G/m$ because clock setting may be delayed in multiples of $G/m$ (both for instantaneous state correction and continuous amortization), and $u_p = [-G/m + \mathcal{O}(G\rho_{osc}), G/m + \mathcal{O}(G\rho_{osc})]$, where the remainder terms account for the deviation between real-time and (observed) logical time.

- *Tick Insertion/Deletion:* This approach is very similar to the above one and is most efficient for $f_{osc} = 2f_{clock}$, so that $G = 1/f_{clock} = 2/f_{osc}$. However, instead of shifting all (future) oscillator ticks as a consequence of any single correction instant, it just inserts an additional oscillator tick between two regular ones in case of speeding up, and suppresses a regular tick if slowing down the clock. Therefore, future ticks occur at the same instants as they would have occurred if no rate correction took place. It follows immediately that $\rho_p = \rho_{osc}$, $G_S = G$ (both for instantaneous state correction and continuous amortization) since slowing down only works in multiples of $G$ (although speeding up would allow $G/2$), and $u_p = [-G + \mathcal{O}(G\rho_{osc}), G + \mathcal{O}(G\rho_{osc})]$.

- *Adder-Based Clock:* This novel clock architecture utilized in our UTCSU-ASIC (see [22] or [23]) uses a fixed-frequency oscillator that drives an adder instead of a counter. At each oscillator tick, a high-resolution ($G_S \ll G_{osc}^0 = 1/f_{osc}$) clock register is incremented by a programmable register STEP, which usually contains the intrinsic value $G_{osc}^0$. It can be modified to any value $G_{osc}^0 + \delta G_S$ for rate adjustment purposes, with $\delta G_S > 0$ speeding up and $\delta G_S < 0$ slowing down the clock. Hence, an adder-based clock has in fact variable internal granularity, equal to the content of STEP. State correction (with clock setting granularity $G_S$) can be carried out by continuous amortization if STEP holds appropriate increments.

The clock's actual granularity $G$, which should satisfy $G \geq G^0_{osc}$ to be meaningful, is imposed by the fact that the clock register is externally accessible with resolution $G$ only, resulting in a reading error of up to $G$. That is, the "fractional part" with resolution $G_S \ll G$ is not visible externally, but nevertheless (continuously) maintained internally. It follows that $\rho_p = \rho_{osc}$ and $u_p = [-G + \mathcal{O}(G\rho_{osc}), G + \mathcal{O}(G\rho_{osc})]$ since the clock register is occasionally incremented by 2G in case of speeding up the clock (STEP $> G$), and not incremented at all in case of slowing it down (STEP $< G$).

Now we will establish a relation between real-time and logical time intervals as measured on a non-faulty local clock. Recalling the definition of $c(T)$ from the beginning, we have for any $t$ and $T = C(t)$

$$c(T) \leq t < c(T + G). \tag{5}$$

Therefore, we easily obtain

$$c(T) - c(T_0 + G) < t - t_0 < c(T + G) - c(T_0) \tag{6}$$

for any $t, t_0$ and $T = C(t), T_0 = C(t_0)$. If $t = \theta$ and/or $t_0 = \theta_0$ denotes some real-time where clock $C(t)$ ticks, we can use the stronger relation $\theta = c(\Theta)$ instead of (5), so that the corresponding $G$ in (6) may be dropped (however, $<$ has to be replaced by $\leq$). The following definition helps in unifying this situation.

DEFINITION 4 (SYNCHRONY) *Real-time $t$ is in synchrony with a node's local clock $C(t)$ iff $t = \theta$ for some real-time $\theta$ where $C(t)$ ticks. Let the indicator function of non-synchrony be defined as*

$$I_{t \neq \theta} = I_\theta(t) = \begin{cases} 0 & \textit{if } t \textit{ is in synchrony with } C(t), \\ 1 & \textit{otherwise.} \end{cases} \tag{7}$$

With the help of this definition, we can generalize (6) to

$$c(T) - c(T_0 + I_{t_0 \neq \theta_0} G) \leq t - t_0 \leq c(T + I_{t \neq \theta} G) - c(T_0). \tag{8}$$

For a non-faulty node $p$, (4) immediately provides

$$(1 - \rho_p^-)\Delta T - u_p^- \leq c_p(T + \Delta T) - c_p(T) \leq (1 + \rho_p^+)\Delta T + u_p^+, \tag{9}$$

for any $T$, $\Delta T$ being integer multiples of $G$. Combining this with (8), we easily obtain the following inequalities estimating the real-time interval $t - t_0$ in terms of the corresponding logical time interval $T - T_0 = C_p(t) - C_p(t_0)$ for a non-faulty clock in the absence of resynchronizations:

LEMMA 8 (DURATION ESTIMATIONS) *Let $t_0$ and $t \geq t_0$ be two arbitrary points in real-time and $T = C_p(t)$, $T_0 = C_p(t_0)$ the corresponding points in logical time at node p. If clock $C_p(t)$ is non-faulty and if there are no adjustments, we have*

$$\begin{aligned} t - t_0 &\geq (T - T_0)(1 - \rho_p^-) - u_p^- - I_{t_0 \neq \theta_0} G(1 - \rho_p^-) \\ t - t_0 &\leq (T - T_0)(1 + \rho_p^+) + u_p^+ + I_{t \neq \theta} G(1 + \rho_p^+) \end{aligned} \tag{10}$$

*and the converse*

$$\frac{t - t_0 - u_p^+}{1 + \rho_p^+} - I_{t \neq \theta} G \leq T - T_0 \leq \frac{t - t_0 + u_p^-}{1 - \rho_p^-} + I_{t_0 \neq \theta_0} G. \tag{11}$$

**Proof:** Due to monotonicity of $C_p(t)$, we always have $T \geq T_0$ since we assumed $t \geq t_0$. If $T > T_0 + G$, (10) follows immediately from plugging in (9) at both sides of (8). Moreover, it is immediately apparent that (10) is also valid for $T = T_0$ and $T = T_0 + G$, although the lower bound is not particularly meaningful. Finally, the converse relation follows by trivial algebraic manipulations. □

Neglecting terms of order $\mathcal{O}(\rho_p^2)$ and $\mathcal{O}(G\rho_p)$ in (11), we easily obtain the common formula

$$(t - t_0)(1 - \rho_p^+) - u_p^+ - I_{t \neq \theta} G \leq T - T_0 \leq (t - t_0)(1 + \rho_p^-) + u_p^- + I_{t_0 \neq \theta_0} G;$$

note that $\rho_p^-, \rho_p^+$ are swapped here. Apart from $u_p^-, u_p^+$, this estimate is the one of [24] improved w.r.t. the quite usual case where $t$ or/and $t_0$ is in synchrony with the ticks of $C_p(t)$. In that case, there is no need to spoil the appropriate upper/lower bound by $G$, actually halving (or even ruling out completely) the adverse effects of non-zero clock granularity.

We link the above clock model with the interval-based paradigm introduced in Section 2 by means of the most important ***drift compensation operation***, cf. [6]: Consider an accurate interval $\boldsymbol{I} = \boldsymbol{I}(t_0) = [T_0 \pm \alpha]$ that somehow appears at a node at some arbitrary real-time $t_0$. In order to provide an interval $\boldsymbol{I}' = \boldsymbol{I}'(t_1)$ representing some arbitrary real-time $t_1 \geq t_0$ based on $\boldsymbol{I}$ locally at the node, one should move (the reference point of) $\boldsymbol{I}$ to the right by $t_1 - t_0$, thus providing the obviously accurate interval $\boldsymbol{I}' = \boldsymbol{I} + (t_1 - t_0)$. Unfortunately, $t_1 - t_0$ is not available, but can be approximated via local clock $C(t)$. Shifting an interval $\boldsymbol{I}$ from $t_0$ to $t_1$ using $C(t)$ is called *dragging*; it provides the interval $\boldsymbol{I}' = \boldsymbol{I} + C(t_1) - C(t_0) = [T_1 \pm \alpha]$. That interval's accurateness, however, might be violated due to the error in approximating $t_1 - t_0$ by $C(t_1) - C(t_0)$.

*Deterioration* is required to maintain accurateness of the dragged interval. This is done by "blowing up" the dragged interval's reference point to an interval accounting for the maximum possible approximation error given by (10), which amounts to enlarge the positive and negative accuracy of interval $\boldsymbol{I}$ according to (12) in Definition 5. Observe that dragging by a fast clock (accounted for by $\rho^-$) requires extending the lower envelope of $\boldsymbol{I}'$.

DEFINITION 5 (DRIFT COMPENSATION) *The result of drift compensation of an accurate interval $I = I(t_0)$ 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 $\rho \subseteq \rho_{\max}$ and rate adjustment uncertainity $u \subseteq u_{\max}$ is the accurate interval $I' = I'(t_1)$ defined by*

$$I' = I + T_1 - T_0 + (T_1 - T_0)\rho + u + I_{t_0 \neq \theta_0}\overline{G} + I_{t_1 \neq \theta_1} G_\rho, \tag{12}$$

*where*

$$G_\rho = [0, G(1 + \rho^+_{\max})], \ \overline{G} = [-G, 0], \ \textit{and}\ G = [0, G]. \tag{13}$$

Figure 3 shows an example of drift compensated intervals based upon some initial interval $[T_0 \pm \alpha]$, at (equidistant) local times $T_i = T_0 + i\Delta T$, $i \geq 1$, in case of $\rho^- > \rho^+$, a fast (but deaccelerating clock), and $u, G \ll \Delta T$. For accurateness, deterioration must ensure that the resulting interval $I'$ intersects with the line $T = t$.

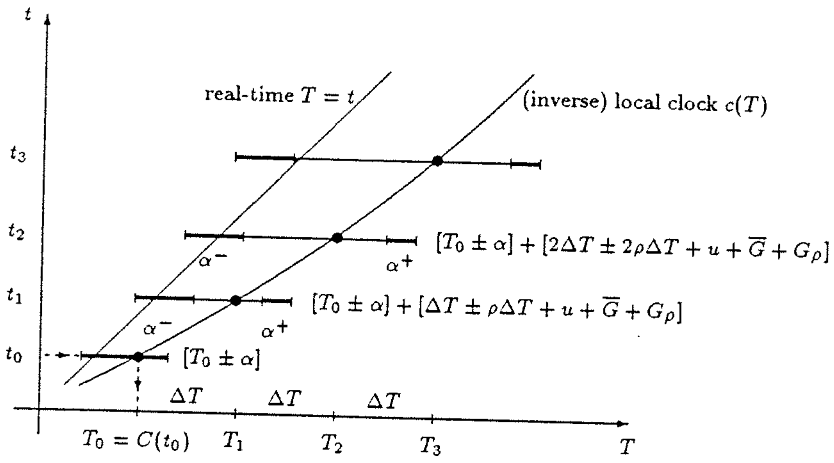



*Figure 3.* Drift Compensation

A few remarks on drift compensation are appropriate here:

- It is important to realize that the errors due to rate adjustment uncertainities (accounted for via $u$) do not add up *in a single (uninterrupted)* drift compensation operation. Unfortunately, they can add up in subsequent drift compensations that are separated by some other operation, like network transmission, drift compensation at another node, or even computation of the convergence function. Therefore, it will turn out that $u_{\max}$ spoils achievable worst case accuracy and precision even more than granularity does(!), cf. Theorem 1. However, we should note that it is very unlikely in practice to have executions where the worst case behavior is actually attained.

- Almost any meaningful $t_1$ is in synchrony with the local clock, since activities of the clock synchronization algorithm are usually initiated when the local clock reaches some predefined value. Therefore, the term involving $G\rho$ in (12) is encountered in our analysis only a few times, for example, when determining maximum precision, see Theorem 1.
- Inequality (12) is valid for $T_1 \geq T_0$, although the negative accuracy of $\boldsymbol{I}'$ is not particularly meaningful (unnecessarily large) when $T_1 = T_0$ or $T_1 = T_0 + G$, recall the proof of Lemma 8. Hence, periods of drift compensation lasting at least $2G$ are preferable w.r.t. tightness of the bounds.

Equipped with those prerequisites, we can eventually introduce the clock model suitable for the interval based paradigm: Each node $p$ has to provide a *local interval clock* $\boldsymbol{C}_p(t)$, which is implemented by an interval of accuracies $\boldsymbol{\alpha}_p$ relative to the nodes instantaneous local clock value $C_p(t)$. In the absence of adjustments (i.e., when running at its intrinsic rate), $\boldsymbol{C}_p(t)$ must be accurate despite of the fact that $C_p(t)$ may drift away from real-time. Hence, $\boldsymbol{\alpha}_p$ must be maintained according to (12):

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

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

*via a local interval of accuracies $\boldsymbol{\alpha}_p(t) = [-\alpha_p^-(t), \alpha_p^+(t)]$ of resolution $G_S$ taken relatively to the node's local clock $C_p(t)$, which is maintained by means of the following operations:*

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

Note that it does not make much sense to maintain accuracies with a resolution below $G_S$, since the clock setting error spoils accuracy by the same amount it spoils the clock value. Choosing $G_S$ as the resolution for accuracies is also advantageous due to the fact that all computations of the clock synchronization algorithm can be performed by using integer arithmetic, provided that all non-integer parameters compiled into the algorithm are integer multiples of $G_S$.

Of course, the clock synchronization algorithm is responsible for periodically reinitializing $C_p(t^R)$ in an accurate way, so that $t \in C_p(t)$ for all $t \geq t^R$ is guaranteed for a non-faulty node $p$ by accurateness of deterioration, cf. (12). The correctness of this statement involves a subtle issue, though, if applications are allowed to read $C_p(t)$ at arbitrary (non-synchronous) real-times $t$. More specifically, in implementations where any reading of the local clock is synchronized to clock ticks, as is the case when using our UTCSU-ASIC, it is of course sufficient to guarantee accurateness for synchronous real-times (so that only $u_p$ must be incorporated). However, in settings that allow reading of the local clock at arbitrary real-times $t$, reading $\alpha_p(t)$ must (explicitly or implicitly) incorporate $G\rho$ as well, since $C_p(t)$ has to be accurate for any $\theta \leq t < \theta + g$ here.

Deteriorating $C_p$ can either be performed by adding $G\rho_p$ at each clock tick or, in an accumulated fashion, by adding $(C(t_a) - C(t_{a-1}))\rho_p$ at the $a$-th clock reading access at real-time $t_a$. For the adder-based clock in our UTCSU-ASIC (cf. the description following Assumption 2), we implemented an approximation of the former technique in hardware: Instead of adding $G\rho_p$ at any clock tick, we add $G^0_{osc}\rho_p$ at any oscillator tick. This approximation introduces an error of at most $\mathcal{O}(G\rho_p)$ in $\alpha_p$, which vanishes in the already present remainder terms, see Theorem 1.

### 3.2. Network Communications

To model the communication subsystem, we first recall that our clock synchronization algorithm operates in periodic rounds, taking place every $P$ (logical time) seconds. At the end of each round, *clock synchronization messages* (CSM) are exchanged among the $n$ nodes of the distributed system in a *full message exchange* (FME). More specifically, in a single FME, each node $p$ transmits a CSM consisting of the accuracy interval $A_p(t^A_{pq}) = [C_p(t^A_{pq}) - \alpha^-_p(t^A_{pq}), C_p(t^A_{pq}) + \alpha^+_p(t^A_{pq})]$ to node $q$ at some real-time $t^A_{pq}$. Applying a suitable convergence function to the set of received and preprocessed accuracy intervals, each node eventually computes a local clock correction value that enforces precision and accuracy.

The required communication primitive for this setting is a basic (unreliable) broadcast operation, something that is easily implemented by means of $n$ send operations in a fully connected point-to-point network, or even provided in hardware by modern broadcast-type networks. The most important requirements are upper and lower bounded transmission delays, i.e., synchronous behavior.

ASSUMPTION 4 (TRANSMISSION CHARACTERISTICS) *We assume a synchronous data network exhibiting the following properties:*

*(1) If a non-faulty node p of the distributed system initiates its broadcast at some arbitrary time $t^I_p$, 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 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, the "indicator function" of making use of a pure broadcast network is*

$$B = \begin{cases} 1 & \text{if } \omega_{\max} = 0 \text{ (pure broadcast network)}, \\ 2 & \text{otherwise.} \end{cases} \tag{14}$$

*(3) If some node p activates its transmission to some node* $q \neq p$ *at time* $t_{pq}^A$ *and no transmission fault occurs, node q receives the message at time* $t_q^p$*, with the* transmission delay $\delta'_{pq} = t_q^p - t_{pq}^A$ *satisfying*

$$\delta'_{pq} \in [\delta_{pq} - \varepsilon_{pq}^-, \delta_{pq} + \varepsilon_{pq}^+], \tag{15}$$

*where* $\delta_{pq}$ *represents the deterministic part and* $\boldsymbol{\varepsilon}_{pq} = [-\varepsilon_{pq}^-, \varepsilon_{pq}^+]$ *the maximum uncertainity of* $\delta'_{pq}$ *(of course,* $\delta_{pq} \geq \varepsilon_{pq}^-$*). 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*

$$\delta_{\min} \boldsymbol{\rho}_{\max} \subseteq \boldsymbol{\varepsilon}_{\max}. \tag{16}$$

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

*(4) The accuracy interval in the CSM is transmitted with limited resolution. More specifically, we assume that the local 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* $\alpha_p^+(t_{pq}^A)$ *are transmitted with finite resolution* $R_A = LG_S$ *for some integer* $L \geq 1$*; let*

$$G_A = R_A - G_S \tag{17}$$

*be the loss in resolution.*

We assume that a CSM is "timestamped" with $C_p(t^A_{pq})$ at the moment of actual transmission $t^A_{pq}$, not at the moment of initiation $t^I_p$ of the broadcast or at the moment $t^S_p$ 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 $\delta'_{pq}$ and hence achievable precision and accuracy. Therefore, we can cope with both the extended capabilities provided by our UTCSU-ASIC (see [23] for details) and with traditional settings ($\omega_{\max} = \lambda_{\max} = 0$ and including any uncertainity in $\varepsilon_{pq}$).

Our model can be adopted to a wide variety of different networks: $\lambda_{\max} > 0$ and $\omega_{\max} = 0$ models broadcast-type networks, whereas $\lambda_{\max} = 0$ and $\omega_{\max} > 0$ is appropriate for point-to-point networks without hardware broadcast capabilities. Note that we can even deal with approaches that stagger CSM transmissions in time to avoid peak network load, simply by making $\omega_{\max}$ large enough to cover the whole period of transmissions.

In the interval-based paradigm, a *delay compensation operation* is responsible for coping with transmission delays (cf. [6]). Basically, delay compensation maintains accurateness of intervals that are transmitted over a network experiencing variable transmission delays according to Assumption 4. If an accurate interval $\boldsymbol{I} = [T \pm \boldsymbol{\alpha}]$ is sent from node $p$ to $q \neq p$, experiencing some transmission delay $\delta' \in [\delta \pm \varepsilon]$, an accurate interval $\boldsymbol{I}''$ (which covers the unobservable $\boldsymbol{I}'$ representing the real-time of reception at the sender-node $p$) is constructed at the receiving node $q$ by shifting the original interval $\boldsymbol{I}$ by $\delta$, and blowing up the shifted interval's reference point to an interval $\varepsilon$ in order to compensate for uncertainities in the transmission delay. In addition, we have to account for the effects of finite transmission resolution $R_A$ of accuracies. Since accuracies $\alpha^-$, $\alpha^+$ of the sending node $p$ are always multiples of $G_S$ according to Assumption 3, truncation to $R_A = LG_S$ for some integer $L \geq 1$ introduces an error of at most $G_A = R_A - G_S$.

DEFINITION 6 (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*

$$\boldsymbol{I}'' = \boldsymbol{I} + \boldsymbol{2}G_A + [\delta_{pq} \pm \varepsilon_{pq}], \tag{18}$$

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

Note that we did not consider the effect of non-zero clock granularity $G$ at the receiving node $q$ here, since a drift compensation operation takes place at $q$ later on. Bear in mind, however, that the real-time of reception $t'$ at $q$ is usually not in synchrony with $q$'s clock.

Figure 4 should make delay compensation straightforward. We assume that the experienced transmission delay is $\delta' > \delta$ and $G_A \ll \varepsilon_{pq}$. The middle time axis corresponds to real-time, whereas the upper and lower ones display logical time at

$p$ and $q$, respectively. The sender node $p$'s rate $r_p$ is presumed to be 1, i.e., its local clock progresses as real-time does. Interpreting Figure 4, one should consider the intervals $I$ and $I''$ as "fixed", whereas the reception time $t'$ and hence the interval $I'$ may vary with $\delta'$.

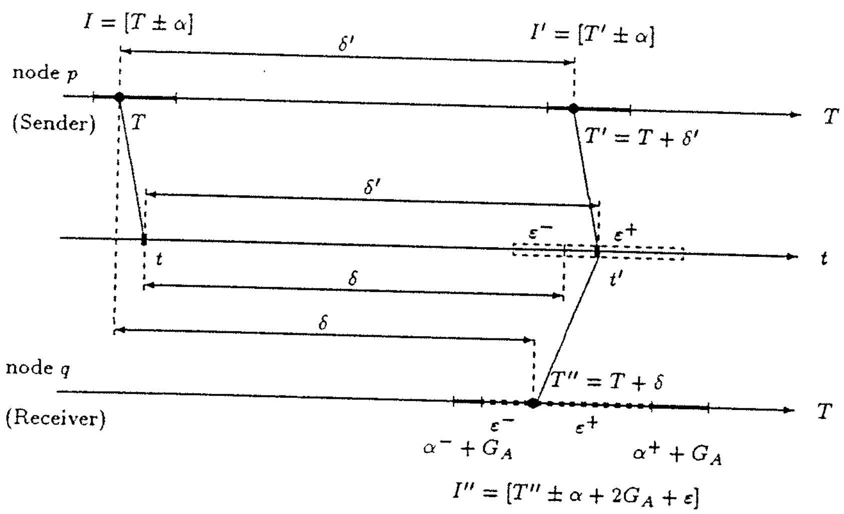



*Figure 4.* Delay Compensation

### 3.3. Fault Model

All assumptions in the previous subsections are meaningful for the fault-free case only. Dealing with fault-tolerant systems, a pertinent fault model is required. However, since we are considering a generic algorithm and its analysis, it does not make much sense to stipulate a particular fault model here — not even an advanced hybrid one as in [26] or [1]. After all, it is the convergence function that is primarily concerned about faults. Consequently, we will assume that an *abstract fault model* $\mathcal{F}$ is provided along with a particular convergence function. $\mathcal{F}$ is abstract in the sense that it gives information on faults not in terms of faulty system components, but rather by classifying the intervals $I_q^p$ provided to the convergence function at node $q$ as a result of reception and preprocessing of the broadcast(s) of node $p$, see (20).

Eventually, any $\mathcal{F}$ must incorporate a (convincing) way of tracing back abstract faults to component faults in order to be meaningful in practice. The following issues are to be considered here:

- Our generic algorithm imposes only a few limitations on the severity of faults. More specifically, faulty nodes or network components may in principle perform arbitrarily, including transmission (and "reception") of any number of arbitrary messages without, however, being capable of causing (serious) "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.

- Viewed from a single (non-faulty) receiving node $q$'s perspective, an interval $I_q^p$ resulting from reception and preprocessing of node $p$'s broadcast(s) during an FME can be faulty in various ways:

  - *Omission faults*, caused by an omissive faulty node $p$ or transient errors during message reception, resulting in $I_q^p = \emptyset$.
  - *Timing faults*, due to a faulty node/clock $p$ or excessive transmission delays, resulting in a non-accurate and/or non-$\pi$-accurate $I_q^p$.
  - *Clock (value) faults*, caused by a faulty node $p$ or a damaged message, resulting in a non-accurate and/or non-$\pi$-accurate $I_q^p$.

  Apart from those faults, which arise in traditional clock synchronization as well, we face additional *accuracy faults* that are unique in the interval-based paradigm. Adopting the terminology of [10], we distinguish three different types:

  - *Truncated accuracy faults*, caused by accuracies being too small, resulting in a non-accurate $I_q^p$.
  - *Bounded accuracy faults*, due to accuracies that are too large but bounded (usually in a way that makes them indistinguishable from accuracies provided in a non-faulty broadcast), resulting in an accurate but not particularly meaningful $I_q^p$.
  - *Unbounded accuracy faults*, resulting in an accurate but meaningless $I_q^p$.

  Obviously, accuracy faults do not affect $\pi$-precision intervals, since $\pi$ is not transmitted but rather compiled into the algorithm. However, one has to account for the possibility that $\hat{I}_q^p$ is $\pi$-accurate despite of the fact that $I_q^p$ suffers from a truncated accuracy fault, and also the opposite situation where $\hat{I}_q^p$ is not $\pi$-accurate although it is accurate because of a bounded/unbounded accuracy fault.

- Viewed from the perspective of corresponding intervals $I_p^s$, $I_q^s$ at two (non-faulty) nodes $p$ and $q$ (for the same sending node $s$) in a single FME, we encounter the following faults:

- *Arbitrary faults* covering (almost, see first item) any kind of faulty behavior, including a byzantine (two-faced, "asymmetric", cf. [1]) one. Arbitrary faults may be caused by nodes sending different messages to different receivers or by excessive transmission delays at the receiving ends. Note that both faults can also occur in broadcast-type networks, since the elementary broadcast operation is not assumed to be reliable, see [18]. Of course, some receiving nodes may experience an omission or deliver a non-faulty interval in an arbitrarily faulty broadcast as well.
- Depending on the convergence function, there are usually one or more classes of faults that may be considered as *restricted faults*, in the sense that they can be tolerated "easier" than arbitrary ones. For example, tolerating $f$ consistently perceived timing faults ("symmetric" faults, cf. [1]), usually requires $n \geq 2f + 1$ nodes (instead of $n \geq 3f + 1$ in case of arbitrary faults). Again, some receiving nodes may experience an omission fault instead of providing a faulty interval; delivery of a non-faulty interval, however, usually turns the fault into an arbitrary one.
- *Omission faults* are usually perceived differently at different receiving nodes $p$ and $q$. Traditionally, they are attributed to sending nodes ("strictly omissive asymmetric faults", cf. [1]), although most receive omissions occur (independently!) at the receiving nodes. Hence, viewed globally, they cannot be traced back to (a reasonably small number of) omissive sending nodes.
- *Crash faults* (and other "benign" faults according to the terminology of [1]) are consistently detectable at all nodes. However, a node that crashes during a broadcast operation produces (at least) restricted faults due to inconsistent reception.

Note finally that the four types of faults above, which are well-known from traditional clock synchronization, are only meaningful for $\pi$-precision intervals, not for accuracy intervals.

## 4. The Algorithm

This section contains the description of our generic clock synchronization algorithm. It employs the common round-based structure from other internal synchronization algorithms. Periodically, every $P$ (logical 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 (involving delay compensation operations, recall Definition 6).
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.

4. Keeping track of real-time by means of the local interval clocks of the nodes (involving drift compensation operations, recall Definition 5) up to the next round.

The abovementioned basic operations, i.e., delay compensation and drift compensation, are required to make the exchanged intervals compatible with each other while maintaining $\pi$-correctness. More specifically, all intervals gathered at node $q$ in an FME are preprocessed to represent 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 (18) 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 $C_q(t_q^p) = T_q^p$. That interval $\boldsymbol{A}_p''$ is then dragged locally by means of the receiver's clock, utilizing drift compensation (12), to some common, synchronous point in real-time $t_q^R$ defined by $C_q(t_q^R) = T_q^R$. Therefore, we arrive at the compatible intervals

$$\boldsymbol{I}_q^p = \boldsymbol{I}_q^p(t_q^R) = \boldsymbol{A}_p + 2\boldsymbol{G}_A + [\delta_{pq} \pm \varepsilon_{pq}] + T_q^R - T_q^p + (T_q^R - T_q^p)\boldsymbol{\rho}_q + \boldsymbol{u}_q + \overline{\boldsymbol{G}}. \tag{19}$$

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 is immediately apparent that $\boldsymbol{I}_q^p$ is accurate —and, as we will justify in Section 5, $\pi$-accurate for some suitable $\pi$ as well— if (1) $\boldsymbol{A}_p$ was accurate, (2) transmission delay was not excessive, and (3) the receiver $q$ is not faulty; recall our discussion of the abstract fault model $\mathcal{F}$ in Section 3.3.

In addition to the intervals obtained from remote nodes $p \neq q$, there is also the accuracy interval of the own node $q$ that needs to be considered. Of course, no actual transmission is required here, so we just have $\boldsymbol{I}_q^q = \boldsymbol{C}_q(t_q^R) = [T_q^R \pm \alpha_q^R]$. Observe carefully that it is possible to compute $\boldsymbol{C}_q(t_q^R)$ in advance by exploiting knowledge of some $\boldsymbol{C}_q(\theta) = [T \pm \alpha]$ with $T < T_q^R$ from the same round and without continuous amortization being active: Since exactly $T_q^R - T$ clock ticks must occur between $t$ and $t_q^R$, we obviously have $\boldsymbol{C}_q(t_q^R) = \boldsymbol{C}_q(t) + T_q^R - T + (T_q^R - T)\boldsymbol{\rho}_q$ due to intrinsic deterioration; recall Assumption 3. Therefore, we can imagine a zero-delay "loop-back transmission" of the accuracy interval $\boldsymbol{A}_q = \boldsymbol{C}_q(t_{qq}^A) = [T_{qq}^A \pm \alpha_{qq}^A]$ at the FME initiation, which "arrives" instantaneously at node $q$ (hence $T_q^q = T_{qq}^A$). Incorporating this and rearranging (19), we finally arrive at

$$\boldsymbol{I}_q^p = \begin{cases} \boldsymbol{A}_p + [T_q^R - T_q^p + \delta_{pq} \pm 2\boldsymbol{G}_A + \varepsilon_{pq}] + (T_q^R - T_q^p)\boldsymbol{\rho}_q + \boldsymbol{u}_q + \overline{\boldsymbol{G}}, & p \neq q, \\ \boldsymbol{A}_q + T_q^R - T_q^q + (T_q^R - T_q^q)\boldsymbol{\rho}_q, & p = q. \end{cases} \tag{20}$$

Now, given the set of node $q$'s intervals $\boldsymbol{I}_q^p$ above, a function that provides a (small) interval that both contains real-time $t_q^R$ and enhances precision —despite of some possibly faulty ones— is required. Adhering to the terminology introduced in [16], we call such a function a convergence function. The clock synchronization algorithm defined below is stated (and subsequently analyzed) on top of a generic convergence function $\mathcal{CV}$, which can be any interval-valued function that satisfies certain properties stated in Definition 11.

DEFINITION 7 (GENERIC ALGORITHM) *With the parameters required for the instance of the algorithm running at node p,*

- *total number n of nodes,*
- *node p's intrinsic inverse rate deviation bound $\boldsymbol{\rho}_p$ and uniform bound $\boldsymbol{\rho}_{\max} \supseteq \bigcup_p \boldsymbol{\rho}_p$ with $\rho_{\max} = |\boldsymbol{\rho}_{\max}| = \rho^-_{\max} + \rho^+_{\max}$ (defined in Assumption 2),*
- *clock granularity G, clock setting granularity $G_S$, node p's maximum rate adjustment uncertainity $\boldsymbol{u}_p = [-u^-_p, u^+_p]$, and uniform maximum rate adjustment uncertainity $\boldsymbol{u}_{\max} \supseteq \bigcup_p \boldsymbol{u}_p$ with $u_{\max} = u^-_{\max} + u^+_{\max}$ (defined in Assumption 2),*
- *transmission delay characteristics $\delta_{sp}$, $\boldsymbol{\varepsilon}_{sp}$ for all nodes s, uniform bounds $0 \leq \delta_{\min} \leq \min_{p,q} \delta_{pq}$, $\delta_{\max} \geq \max_{p,q} \delta_{pq}$ and $\boldsymbol{\varepsilon}_{\max} \supseteq \bigcup_{p,q} \boldsymbol{\varepsilon}_{pq}$ with $\varepsilon_{\max} = |\boldsymbol{\varepsilon}_{\max}| = \varepsilon^-_{\max} + \varepsilon^+_{\max}$, and accuracy transmission loss $G_A$ (defined in Assumption 4),*
- *computation delay compensation (integer multiple of G) guaranteeing node p's maximum computation time $\gamma_p$ (defined in Assumption 1), chosen according to*

  $$\Gamma_p \geq \frac{\gamma_p + u^-_p}{1 - \rho^-_p},$$

  *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 chosen to be the same at 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 $\Delta$ below, it ensures that resynchronization starts only after all CSMs broadcast by non-faulty nodes during an FME have arrived (cf. Assumption 4),*
- *transmission delay compensation $\Delta$ (integer multiple of G) chosen according to*

  $$\Delta \geq \frac{\pi_0 + u_{\max} + \delta_{\max} + (P - \Gamma_{\min} + \pi^-)\rho_{\max} + \varepsilon^+_{\max}}{1 + \rho^+_{\max}} \tag{21}$$

  *(defined in Lemma 11), where $\pi_0 = |\boldsymbol{\pi}_0|$ and $\pi^-$ (defined in Theorem 1) depend on the convergence function employed,*

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

*where all parameters are integer multiples of* $G_S$ *(cf. Assumption 3) 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* $\boldsymbol{A}_q^0 = [T_q^0 - \alpha_q^{0-}, T_q^0 + \alpha_q^{0+}]$ *at some synchronous real-time* $t_q^0$ *by some external means. This initialization must ensure*

- $t_q^0 \in \boldsymbol{A}_q^0$,
- $T_q^0 \in [\Lambda + \Omega + \Delta + \Gamma_q \pm \pi]$,
- $\alpha_q^0 \subseteq \pi_0$,

*where* $\pi$ *and* $\pi_0$ *depend on the convergence function (defined in Theorem 1). Note that we assume here w.l.o.g. that real-time and logical time start at* $t = 0$ *and* $T = 0$*, respectively, at the beginning of round 0.*

(1) **Periodic Synchronization**: *Near the end of each round* $k \geq 0$*, every node q in the system performs the following operations:*

(S) CSM Send: *Periodically at times* $C_q(t_q^I) = T^I = (k+1)P$ *(the dependency of* $T^I$, $t_q^I$*, etc. upon round k is suppressed for brevity), 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* $\boldsymbol{A}_{qp} = [T_{qp}^A \pm \alpha_{qp}^A] = \boldsymbol{C}_q(t_{qp}^A)$*. 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: *When a CSM* $M_{pq}$ *from node p arrives at node q at real-time* $t_q^p$ *where* $C_q(t_q^p) = T_q^p$ *(with* $T_q^q = T_{qq}^A$*), the interval given in (20), namely*

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

*is computed and stored in an ordered set* $\mathcal{I}_q$*. For the definition of the resynchronization time* $T_q^R$ *turn to item (T).*

(C) Computation: *At real-time* $t_q^{\Lambda+\Omega+\Delta}$ *defined by* $C_q(t_q^{\Lambda+\Omega+\Delta}) = T^{\Lambda+\Omega+\Delta} = (k+1)P + \Lambda + \Omega + \Delta$ *in round k, the (generic) convergence function* $\mathcal{CV}$ *(cf. Definition 11) is applied to the compatible intervals stored in* $\mathcal{I}_q$*, yielding the result* $\boldsymbol{R}_q$*. For any node p not present in* $\mathcal{I}_q$ *by time* $t_q^{\Lambda+\Omega+\Delta}$*, an omission fault is assumed; the empty interval* $\boldsymbol{I}_q^p = \emptyset$ *is provided to the convergence function in this case. Finally,* $\mathcal{I}_q$ *is re-initialized to the empty set for the next round* $k+1$*.*

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

A few remarks are certainly appropriate here:

- A number of parameters defined in our system model (Assumptions 1–4) must be provided (statically or dynamically) to the instance of the algorithm running at a node $p$, e.g., rate deviation bound $\rho_p$, transmission delay parameters $\delta_{pq}$, $\varepsilon_{pq}$, and quantities $\Delta$, $\Lambda$, $\Omega$, $\Gamma_p$, etc. related to $\lambda$, $\omega$, and $\gamma_p$. The particular convergence function might also require some parameters. Therefore, our algorithm depends on those parameters not implicitly as traditional synchronization algorithms do, but rather explicitly; see our comments in Section 6.

- The needed initial synchronization is automatically provided when the algorithm is used in our clock validation framework, cf. Section 1. Clock validation assumes that there are some *primary nodes* having their physical clock disciplined by an UTC time source of high accuracy, which may, however, fail arbitrarily. In normal operation, a primary node $p$'s local clock $C_p$ provides UTC with some *a priori* accuracy $\alpha^0$, such that $t \in [C_p(t) - \alpha^0, C_q(t) + \alpha^0]$, for all real-times $t \geq 0$. Since the analysis of this paper is entirely devoted to phases of total unavailability of UTC, excluding normal or "mixed" operation, we may assume that local clocks are initially synchronized when flywheeling begins.

- Steps (S), (C), and (T) of the algorithm are triggered when the local clock reaches some specific point in (logical) time, so that they are effectively sequenced. Step (R), however, takes place asynchronously when a CSM drops in. Note also that the execution time required for computing the convergence function is usually smaller when the latter is scattered among the CSM receptions, i.e., when performing piecewise computation in step (R). Of course, if $\gamma_p$ is chosen appropriately, our results apply to this situation as well.

- The set $\mathcal{I}_q$ is an ordered set, i.e., a vector. Each element of an ordered set has assigned a unique (global) number permitting total ordering. In our particular framework, we use the identifier of the node that originated the interval. Note that most sets in our analysis are ordered ones, so *set* should usually be read as *ordered set* throughout the paper.

- Resetting the local interval clock instantaneously in step (T) of the algorithm would cause non-monotonicity and non-continuity of local time. This is circumvented in practice by means of the continuous amortization algorithm of [21]: To perform state correction of the local clock, its rate is modified by a fixed amount $\pm\psi$ for a programmable period until the clock has changed its value as desired. This algorithm, which is supported in hardware by our UTCSU-ASIC, is particularly attractive as it does not impair the worst-case precision and accuracy obtained for instantaneous correction if $\psi$ is chosen suitably, see Theorem 2 for details.

In the remaining sections, we will analyze worst case accuracy and precision of the algorithm given in Definition 7, stating expressions in terms of characteristic parameters of the convergence function. The major results obtained herein will be summarized in Theorem 1 (instantaneous correction) and Theorem 2 (continuous amortization).

## 5. Precision and Accuracy Analysis

From the description of the algorithm in the preceding section, it should be reasonably obvious that accurateness of the local interval clocks $C_p(t)$ of all non-faulty nodes is maintained during all rounds. In fact, all operations involved (drift compensation, delay compensation, and application of the convergence function) are explicitly designed to preserve accurateness of the intervals involved. This is not that obvious for precision, however, so we find it appropriate to give an informal overview how precision is actually maintained.

First of all, recall that any local interval clock $C_p(t)$ is defined by three values, namely accuracies $\alpha_p^-(t)$, $\alpha_p^+(t)$ and local clock value $C_p(t)$ as reference point. Dealing solely with accuracy, the lower and the upper edge would be sufficient. Hence, the reference point can be set appropriately to achieve the orthogonal goal of maintaining the precision condition $|C_p(t) - C_q(t)| \leq \pi$ for all non-faulty nodes $p$ and $q$. From Lemma 3 in Section 2.2, we know that this is achieved when the interval clocks of all non-faulty nodes are kept $\pi$-precise.

The actual approach taken is particularly attractive due to the fact that precision during a round is automatically maintained when accuracy is. To understand how this works, assume that all members of the set $\mathcal{C}(t)$ of non-faulty interval clocks are $\pi_0$-correct at some real-time $t^0$, i.e., that their associated $\pi_0$-precision intervals contain $\tau(t^0)$, and recall that internal global time $\tau$ progresses as real-time does. When trying to capture real-time $t$ by $C_p(t)$, we must deteriorate (enlarge) the accuracy interval in order to compensate for the drift of the local clock. However, if this is done properly to capture real-time $t' > t^0$, it is clear that the associated $\pi$-precision interval $\hat{C}_p(t')$ captures internal global time $\tau(t') > \tau(t^0)$ as well, provided that $\pi$ is the result of enlarging $\pi_0$ by the maximum amount any $C_j$ has been enlarged. It is important to understand, though, that enlargement of precision intervals is just a matter of analysis, since the algorithm does not deal with precision intervals at all. Anyway, it follows that $\mathcal{C}(t')$ must be $\pi$-correct.

Whereas enlarging $\pi_0$ to $\pi$ guarantees that $\tau(t)$ lies in the intersection of the $\pi$-precision intervals of all non-faulty nodes, this cannot ensure <u>bounded</u> 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 $\pi$-correct local interval clocks are reinitialized to newly computed accuracy intervals that are $\pi_0$-precise for $\pi_0 \subset \pi$ (= precision enhancement). Note that we cannot safely assume $\pi_0$-correctness here, since it may be the case that internal global time $\tau^k$ for round $k$ does not lie in the intersection of the new $\pi_0$-precision intervals. However, if we define a new internal global time $\tau^{k+1}$ for round $k+1$, independently of its predecessor $\tau^k$, we have of course $\pi_0$-correctness w.r.t.

$\tau^{k+1}$. Consequently, the resynchronization launches the next round $k+1$ during which initial precision $\pi_0$ again deteriorates to $\pi$.

Keep in mind that only $\pi$-precision intervals $\hat{C}_p$ are affected by precision enhancement. The local interval clocks $C_p$ itself must continuously track real-time $t$, so that the accuracies could grow. Actually, accuracy in round $k$ can be viewed as an accumulation of the $\pi$-precision intervals during round $0, \ldots, k$. This eventually explains why $t$ and $\tau$ will usually be apart, as mentioned in Section 2.2.

### 5.1. Internal Global Time

When trying to formalize the concept of internal global time, a number of technical difficulties arise. First of all, we have to establish a notion that allows us to deal with multivalued local time. Near a resynchronization event at node $p$, occurring at real-time $t_p^R$, one could be interested in local time $T_p^R = C_p(t_p^R)$ taken from the clock before resynchronization, and/or in $T_p^{R'} = C_p'(t_p^R)$ read from the already resynchronized clock. To express this situation unambiguously, we employ the well-thumbed technical device of *virtual clocks*: When round $k$ at node $p$ commences at real-time $t_p^{R,k-1}$, a virtual clock $C_p^k$ is instantiated that progresses according to the physical clock $C_p$ up to time $t_p^{R,k}$, when the $k+1$-th resynchronization event takes place. At this instant, a new virtual clock $C_p^{k+1}$ (initialized with a value based on the convergence function applied to the intervals taken from the FME) is instantiated, which then proceeds concurrently with $C_p^k$ in the same way. Needless to say, it is the set of virtual clocks of round $k$ that defines the instance $\tau^k = \tau^k(t)$ of internal global time.

The probably most awkward problem when trying to define internal global time, however, arises from the fact that the contributing intervals reside at different nodes. After all, resynchronizations at different nodes do not occur simultaneously. Albeit two nodes are within the same round $k$ most of the time, there are short periods where one has already resynchronized (and thus started its round $k+1$) while the other one has not. Nevertheless, accuracy intervals from different nodes must be made compatible to form a $\pi$-precise set. For practical purposes, this requires dragging by means of the local clock and utilizing drift compensation (see Definition 5). For the purpose of analysis, however, there is no need to make intervals residing at different nodes compatible in a practicable way. Rather than using dragging and drift compensation, it is sufficient to employ a simple, ideal shift-operation:

DEFINITION 8 (SHIFTING) *The result of* shifting *an interval $\boldsymbol{I} = \boldsymbol{I}(t_1)$ to some point in time $t' \geq t_1$ is the interval $\boldsymbol{J} = \boldsymbol{J}(t') = shift_{t'}(\boldsymbol{I}) = \boldsymbol{I} + t' - t_1$. For a set of $n \geq 1$ intervals $\mathcal{I} = \{\boldsymbol{I}_1(t_1), \ldots, \boldsymbol{I}_n(t_n)\}$, the shifted set $\mathcal{J}$ of compatible intervals all representing some arbitrary $t' \geq \max_{1 \leq i \leq n} t_i$ is defined by $\mathcal{J} = shift_{t'}(\mathcal{I}) = \{\boldsymbol{I}_1(t_1) + t' - t_1, \ldots, \boldsymbol{I}_n(t_n) + t' - t_n\}$.*

Of course, any interval in $\mathcal{J}$ above represents $t'$. However, keep in mind that they are artificial constructions, i.e., that they could only be provided by dragging $I$ with an ideal, continuous real-time clock. Note that restricting $t'$ to a point in time greater than any $t_j$ of the intervals in $\mathcal{I}$ is not really necessary.

LEMMA 9 (PRECISION SHIFTED INTERVALS) *Let $\mathcal{I} = \{I_1(t_1), \ldots, I_n(t_n)\}$ be a set of intervals with $I_i(t_i) = [T_i \pm \alpha_i]$ representing real-time $t_i$ for $1 \leq i \leq n$, and $\mathcal{J} = \{J_1(t'), \ldots, J_n(t')\} = shift_{t'}(\mathcal{I}) = \{I_1(t_1) + t' - t_1, \ldots, I_n(t_n) + t' - t_n\}$ for some arbitrary $t' \geq \max_{1 \leq i \leq n} t_i$.*

*(1) If, for any $i$, the interval $I_i$ in $\mathcal{I}$ is $\pi_i$-correct w.r.t. internal global time of the same round $k$, then $J_i = I_i + t' - t_i$ is also $\pi_i$-correct, and the whole shifted set $\mathcal{J}$ is $\pi'$-correct for $\pi' = \bigcup_i \pi_i$,*

*(2) if $\mathcal{J}$ is $\pi$-correct w.r.t. the internal global time of round $k$, then $\tau_i = \tau^k(t_i) \in \hat{I}_i(\tau_i)$ for any $i$.*

**Proof:** To prove the first statement of the lemma, we note that $\pi_i$-correctness of $I_i$ implies $\pi_i$-correctness of $J_i = I_i + t' - t_i$, since $\hat{J}_i = \hat{I}_i + \tau' - \tau_i$ due to the fact that internal global time (of the same round) progresses as real-time does, i.e., $\tau' - \tau_i = t' - t_i$. The asserted $\bigcup_i \pi_i$-correctness follows from $n-1$ applications of Lemma 5 to the union of the singletons $\{I_j + t' - t_j\}$ forming $\mathcal{J}$.

To prove the second statement of the lemma, we use the same argument as before to derive $\hat{J}_i(\tau_i) = \hat{I}_i(\tau_i) + \tau' - \tau_i$ for $\tau' = \tau(t')$, so that the set of $\pi$-precision intervals associated with $\mathcal{J}$ reads $\hat{\mathcal{J}} = \{\hat{I}_1(\tau_1) + \tau' - \tau_1, \ldots, \hat{I}_n(\tau_n) + \tau' - \tau_n\}$. The asserted $\pi$-correctness of $J_i$ implies $\tau' \in \hat{I}_i(\tau_i) + \tau' - \tau_i$, hence $\tau_i \in \hat{I}_i(\tau_i)$. □

It is helpful to view the precision of shifted intervals as an *ideal* one, in contrast to the *observable* precision obtained by applying drift compensation. We have the following relation between ideal and observable precision:

LEMMA 10 (SHIFTING VS. DRIFT COMPENSATION) *Let $\mathcal{I} = \{I_1(t_1), \ldots, I_n(t_n)\}$ be a set of intervals $I_i(t_i) = [T_i \pm \alpha_i]$ residing at node $p_i$ and representing real-time $t_i$ being in synchrony with clock $C_{p_i}$. For some arbitrary $t' \geq \max_{1 \leq i \leq n} t_i$, define $\mathcal{I}' = \{I'_1(t'), \ldots, I'_n(t')\}$ to be the set of compatible intervals $I'_i(t')$ obtained from $I_i(t_i)$ by applying drift compensation at node $p_i$.*

*(1) If the shifted set $\mathcal{J} = shift_{t'}(\mathcal{I}) = \{I_1(t_1) + t' - t_1, \ldots, I_n(t_n) + t' - t_n\}$ of compatible intervals all representing $t'$ is $\pi$-precise, then $\mathcal{I}'$ is $\pi'$-precise for*

$$\pi' = \pi + \bigcup_i (T'_i - T_i)\rho_{p_i} + u_{p_i} + G_\rho, \tag{22}$$

*where $T'_i - T_i = C_{p_i}(t') - C_{p_i}(t_i)$ satisfies*

$$\frac{t' - t_i - u^+_{p_i}}{1 + \rho^+_{p_i}} - G \le T'_i - T_i \le \frac{t' - t_i + u^-_{p_i}}{1 - \rho^-_{p_i}}. \tag{23}$$

*(2) If, for all $i$, the interval $I_i$ is $\pi_i$-correct, then $\mathcal{I}'$ is $\pi'$-correct for*

$$\pi' = \bigcup_i \pi_i + (T'_i - T_i)\rho_{p_i} + u_{p_i} + G_\rho. \tag{24}$$

**Proof:** Recalling Definition 5 of drift compensation, (22) and also (24) follow immediately, since (ideal) $\pi$ must be enlarged to capture the maximum deviation from real-time that can arise when dragging the intervals $I_i$ from $t_i$ to $t'$; recall that $t_i$ is in synchrony with node $p_i$'s clock. The bounds on $T'_i - T_i = C_{p_i}(t') - C_{p_i}(t_i)$ given by (23) are obtained directly from (11). □

The above lemma allows us to carry over any result involving shifted intervals to the situation where those intervals are actually compatible in the real system, i.e., when they are read at some common point in real-time $t'$. Apart from the inevitable effect of granularity, there is an additional enlargement essentially proportional to $(t' - t_i)\rho_{p_i}$.

With these preparations, we are ready for the precise definition of internal global time:

DEFINITION 9 (INTERNAL GLOBAL TIME) *Let $\mathcal{C}^{k+1}$ for $k \ge 0$ be the set of the non-faulty nodes' virtual interval clocks $C_j^{k+1}(t_j^{R,k})$ of round $k+1$ at their $k+1$-th resynchronization instants $t_j^{R,k}$, when switching from round $k$ to $k+1$ takes place at node $j$. If $shift_{t^{R,k}}\mathcal{C}^{k+1}$ for some $t^{R,k} \ge \max_q t_q^{R,k}$ is $\pi_0$-precise, we define* internal global time *$\tau^{k+1}(t)$ for round $k+1$ by*

$$\begin{aligned} \tau^{k+1}(t) &= \tau^{k+1}(t^{R,k}) + (t - t^{R,k}) \qquad \textit{for } k \ge 0, \\ \tau^0(t) &= t \end{aligned}$$

*valid for all $t$, where $\tau^{k+1}(t^{R,k}) \in \bigcap_{J \in shift_{t^{R,k}}\mathcal{C}^{k+1}} \hat{J} \neq \emptyset$.*

Note that defining $\tau^0(t) = t$ is justified by applying Lemma 4 to the initial synchronization assumption given in Definition 7.

### 5.2. Generic Interval-Based Analysis

In this subsection, we will provide our interval-based framework for analyzing worst case precision and accuracy of the generic algorithm given in Definition 7. As in [16], we will describe convergence functions by a few characteristic parameters (functions) and derive expressions in terms of those. To obtain the final results for a particular instance of the algorithm, it is only necessary to determine the characteristic functions of the particular convergence function and to plug them into Theorems 1 and 2.

The general outline of our generic analysis is quite straightforward: We provide a sequence of lemmas that describe how accuracy/precision intervals evolve in a single round. Starting from Lemma 11 describing the set of intervals fed into the convergence function, Lemma 12 guarantees 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 can be conducted to establish our major Theorem 1, which provides results for instantaneous correction. Finally, adopting the achievements of [21], it follows that those results are also valid in case of continuous amortization.

Our interval-based framework surpasses traditional approaches to precision analysis due to its conceptual beauty and high flexibility w.r.t. incorporating features like clock granularity, broadcast latencies, etc. This is primarily a consequence of our notion of internal global time, which allows to focus on the behavior of a single clock instead of being forced to consider the whole ensemble. Even more, in our analysis, there is no need to consider the "position" of intervals (i.e., local clock values) at all. In fact, any information required is encoded in the interval of accuracies $\alpha$ (resp. in the associated $\pi$-precision interval) of an interval $I(t) = [T \pm \alpha]$, since all (non-faulty) accuracy intervals must contain real-time $t$ (resp. internal global time $\tau$) by construction, which thus serves as a "common reference" for relating them. Of course, the particular reference point may lie anywhere in $[t - \alpha^+, t + \alpha^-]$ (resp. $[t - \pi^+, t + \pi^-]$), according to the actually experienced drift, transmission delay, and initial accuracy, but we do not have to deal with it explicitly.

The following first major lemma describes how precision evolves during a round, including local drift compensation and interval dissemination in the FME. Keep in mind that the resulting intervals $I_q^p$ are fed into the convergence function.

LEMMA 11 (FME DISSEMINATION) *Let $A_p = [T_p \pm \alpha_p] = C_p^k(t_p^{R,k-1})$ be the accuracy interval of node p's interval clock at real-time $t_p = t_p^{R,k-1}$, when round k (for some fixed $k \geq 0$) starts, and denote by $\mathcal{A}$ the subset of the $A_p$'s of those nodes p that remain non-faulty during round k. Let $T_q^R = (k+1)P + \Lambda + \Omega + \Delta + \Gamma_q$ be the logical time when the $k+1$-th resynchronization instant —happening at real-time $t_q^R = t_q^{R,k}$ and terminating round k— is scheduled at node q, and let $I_q^p = I_q^p(t_q^R)$ be the interval (20) that is obtained at node q as the result of delay and drift compensation of a node p's accuracy interval transmitted during the FME in round k.*

*If, at the beginning of the round,*

*[0] the accuracies of any $A_p = [T_p \pm \alpha_p] \in \mathcal{A}$ are integer multiples of $G_S$,*

*[1] $shift_{t'}(\mathcal{A})$, $t' \geq \max_p t_p$ arbitrary, is $\pi_0$-correct for some $\pi_0 = [-\pi_0^-, \pi_0^+]$ with $\pi_0 = \pi_0^- + \pi_0^+$,*

*[2] there is some* $\pi = [-\pi^-, \pi^+]$ *with* $\pi = \pi^- + \pi^+ \geq \pi_0$ *such that*

$$T_p \in [kP + \Lambda + \Omega + \Delta + \Gamma_p \pm \pi] \tag{25}$$

*for any* $\boldsymbol{A}_p \in \boldsymbol{\mathcal{A}}$,

*[3] transmission delay, broadcast delay, computation time compensation, and round period are integer multiples of* $G$ *satisfying*

$$\begin{aligned}
\Delta &\geq \frac{\pi_0 + u_{\max} + \delta_{\max} + (P - \Gamma_{\min} + \pi^-)\rho_{\max} + \varepsilon^+_{\max}}{1 + \rho^+_{\max}}, && (26)\\
\Lambda + \Omega &\geq \frac{\lambda_{\max} + \omega_{\max} + u^-_{\max}}{1 - \rho^-_{\max}}, && (27)\\
\Gamma_p &\geq \frac{\gamma_p + u^-_p}{1 - \rho^-_p} \qquad \text{for node } p, && (28)\\
\Gamma_{\max} &\geq \max_p \Gamma_p,\\
0 \leq \Gamma_{\min} &\leq \min_p \Gamma_p,\\
P &\geq \Lambda + \Omega + \Delta + \Gamma_{\max},
\end{aligned}$$

*we have the following results:*

*(1) Any non-faulty receiving node* $q$ *is able to form its set* $\mathcal{I}_q$ *of intervals* $\boldsymbol{I}^p_q = \boldsymbol{I}^p_q(t^R_q) = [T^{p,R}_q \pm \boldsymbol{\alpha}^{p,R}_q]$ *—given by (20)— at least* $\gamma_q$ *real-time seconds before resynchronization takes place at time* $t^R_q$. *By default,* $\boldsymbol{I}^p_q = \emptyset$ *if node* $p$*'s CSM did not arrive in time. Any* $\boldsymbol{I}^p_q \in \boldsymbol{\mathcal{N}}_q \subseteq \mathcal{I}_q$ *(contained in the subset* $\boldsymbol{\mathcal{N}}_q$ *of non-faulty intervals) is accurate with accuracies being integer multiples of* $G_S$, *which satisfy*

$$\begin{aligned}
\boldsymbol{\alpha}^{p,R}_q \subseteq\ & \boldsymbol{\alpha}_p + \boldsymbol{u}_p + \boldsymbol{u}_q + \overline{\boldsymbol{G}} + 2\boldsymbol{G}_A + \boldsymbol{\varepsilon}_{pq}\\
& + (P - \Delta - \Gamma_p)\boldsymbol{\rho}_p + (\Gamma_q + \Delta - \delta_{pq})\boldsymbol{\rho}_q\\
& + (\Lambda + \Omega)[-\max\{\rho^-_q - \rho^-_p, 0\}, \max\{\rho^+_q - \rho^+_p, 0\}]\\
& + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}
\end{aligned} \tag{29}$$

*and*

$$\begin{aligned}
\boldsymbol{\alpha}^{p,R}_q \supseteq\ & \boldsymbol{\alpha}_p + \boldsymbol{u}_p + \boldsymbol{u}_q + \overline{\boldsymbol{G}} + 2\boldsymbol{G}_A + \boldsymbol{\varepsilon}_{pq}\\
& + (P - \Delta - \Gamma_p)\boldsymbol{\rho}_p + (\Gamma_q + \Delta - \delta_{pq})\boldsymbol{\rho}_q\\
& - (\Lambda + \Omega)[-\max\{0, \rho^-_p - \rho^-_q\}, \max\{0, \rho^+_p - \rho^+_q\}]\\
& + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}
\end{aligned} \tag{30}$$

*for* $p \neq q$, *and*

$$\alpha_q^{q,R} = \alpha_q + u_q + P\rho_q + \mathcal{O}(\pi)\rho_q. \tag{31}$$

*Moreover, any $I_q^p \in \mathcal{N}_q$ is $\pi_q^p$-correct with*

$$\begin{aligned}\pi_q^p \subseteq\ & \pi_0 + u_p + u_q + \overline{G} + \varepsilon_{pq} + (P - \Delta - \Gamma_p)\rho_p + (\Gamma_q + \Delta - \delta_{pq})\rho_q \\ & + (\Lambda + \Omega)\big[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}\big] \\ & + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}\end{aligned} \tag{32}$$

*for $p \neq q$, and*

$$\pi_q^q = \pi_0 + u_q + P\rho_q + \mathcal{O}(\pi)\rho_q. \tag{33}$$

*The whole set $\mathcal{N}_q$ is $\pi_q^H$-correct for*

$$\begin{aligned}\pi_q^H \subseteq\ & \pi_0 + u_{\max} + u_q + \overline{G} + \varepsilon_{\max} \\ & + (P - \Delta - \Gamma_{\min})\rho_{\max} + (\Gamma_q + \Delta - \delta_{\min})\rho_q \\ & + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}.\end{aligned} \tag{34}$$

*(2) The set $\mathcal{N} = \bigcup_{q \text{ non-faulty}} \mathcal{N}_q$ containing all non-faulty intervals $I_q^p$ at all non-faulty receiving nodes $q$ has the property that $\mathit{shift}_{t''}(\mathcal{N})$ for some arbitrary $t''$ satisfying $t'' \geq \max_q t_q^R$ is $\pi^H$-correct for*

$$\begin{aligned}\pi^H \subseteq\ & \pi_0 + 2u_{\max} + \overline{G} + \varepsilon_{\max} + (P + \Gamma_{\max} - \Gamma_{\min} - \delta_{\min})\rho_{\max} \\ & + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}.\end{aligned} \tag{35}$$

*(3) Finally, the set $\mathcal{N}^p$ formed by the non-faulty nodes' "perceptions" $I_q^p$ of the accuracy interval transmitted by a single non-faulty node $p$ has the property that $\mathit{shift}_{t''}(\mathcal{N}^p)$ for $t'' \geq \max_q t_q^R$ is both $\pi^p$-correct and $\pi_I$-precise with*

$$\begin{aligned}\pi^p \subseteq\ & \pi_0 + u_p + u_{\max} + \overline{G} + \varepsilon_{\max} \\ & + (P - \Delta - \Gamma_p)\rho_p + (\Gamma_{\max} + \Delta - \delta_{\min})\rho_{\max} \\ & + (\Lambda + \Omega)\big[-(\rho_{\max}^- - \rho_p^-), \rho_{\max}^+ - \rho_p^+\big] \\ & + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max},\end{aligned} \tag{36}$$

$$\begin{aligned}\pi_I \subseteq\ & \varepsilon_{\max} + Bu_{\max} + \overline{G} + (\Lambda + \Omega + \Delta + \Gamma_{\max} - \delta_{\min})\rho_{\max} \\ & + \mathcal{O}(\pi_0 + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}.\end{aligned} \tag{37}$$

**Proof:** From the description of the clock synchronization algorithm in Definition 7, we know that any node $p$ initiates its broadcast in round $k$ when its local clock reaches time $(k+1)P$. Due to broadcast latency $\lambda_{\max}$ and broadcast operation delay $\omega_p \leq \omega_{\max}$ according to Assumption 4, the CSM to node $q$ is actually transmitted when the local clock of the transmitting node $p$ reads time $T_{pq}^A = C_p(t_{pq}^A)$ satisfying

$$(k+1)P \leq T^A_{pq} \leq (k+1)P + \frac{\lambda_{\max} + \omega_{\max} + u^-_{\max}}{1-\rho^-_{\max}} \leq (k+1)P + \Lambda + \Omega, \quad (38)$$

recall (11) in conjunction with (27). Of course, $t^A_{pq}$ is the real-time when the CSM (containing the accuracy interval $A^A_{pq} = [T^A_{pq} \pm \alpha^A_{pq}]$) is actually transmitted. Combining (38) with (25), we easily obtain

$$\begin{aligned} T^A_{pq} - T_p &\leq (k+1)P + \Lambda + \Omega - (kP + \Lambda + \Omega + \Delta + \Gamma_p - \pi^-) \\ &= P - \Delta - \Gamma_p + \pi^- \end{aligned} \quad (39)$$

and similarly

$$T^A_{pq} - T_q \leq P - \Delta - \Gamma_q + \pi^-. \quad (40)$$

The CSM from node $p$ is received (we consider non-faulty intervals here) at node $q$ delayed by $\delta'_{pq}$, hence at real-time $t^p_q = t^A_{pq} + \delta'_{pq}$, when the local clock of node $q$ reads $T^p_q$. We will now establish bounds on $T^p_q - T^A_{pq}$.

Relating the points in real-time involved in the evolution of a round (from the beginning to reception of the FME) yields

$$t^p_q - t_q = t^p_q - t_p - (t_q - t_p) = t^A_{pq} + \delta'_{pq} - t_p - (t_q - t_p). \quad (41)$$

Since $t_q$, $t_p$, and $t^A_{pq}$ (but not $t^p_q$) are in synchrony with their respective clocks, applying (10) to $t^p_q - t_q$ and $t^A_{pq} - t_p$ provides

$$\begin{aligned} (T^p_q - T_q)(1-\rho^-_q) - u^-_q &\leq (T^A_{pq} - T_p)(1+\rho^+_p) + u^+_p + \delta'_{pq} \\ &\quad - (t_q - t_p) \\ (T^p_q - T_q)(1+\rho^+_q) + G(1+\rho^+_q) + u^+_q &\geq (T^A_{pq} - T_p)(1-\rho^-_p) - u^-_p + \delta'_{pq} \\ &\quad - (t_q - t_p). \end{aligned}$$

Some algebraic manipulations produce

$$\begin{aligned} (T^p_q - T^A_{pq})(1-\rho^-_q) &\leq (T_q - T^A_{pq})(1-\rho^-_q) + (T^A_{pq} - T_p)(1+\rho^+_p) \\ &\quad + \delta'_{pq} - (t_q - t_p) + u^-_q + u^+_p \\ &\leq T_q - t_q - (T_p - t_p) + (T^A_{pq} - T_q)\rho^-_q + (T^A_{pq} - T_p)\rho^+_p \\ &\quad + \delta_{pq} + \varepsilon^+_{pq} + u^-_q + u^+_p \end{aligned}$$

and

$$\begin{aligned} G(1+\rho^+_q) + (T^p_q - T^A_{pq})(1+\rho^+_q) &\geq (T_q - T^A_{pq})(1+\rho^+_q) \\ &\quad + (T^A_{pq} - T_p)(1-\rho^-_p) + \delta'_{pq} \\ &\quad - (t_q - t_p) - u^+_q - u^-_p \\ &\geq T_q - t_q - (T_p - t_p) - (T^A_{pq} - T_q)\rho^+_q \\ &\quad - (T^A_{pq} - T_p)\rho^-_p + \delta_{pq} - \varepsilon^-_{pq} - u^+_q - u^-_p \end{aligned}$$

Abbreviating $\mu_{pq} = T_q - t_q - (T_p - t_p)$ and recalling (39) and (40), we eventually obtain

$$
\begin{aligned}
T_q^p - T_{pq}^A \leq \; & \frac{1}{1-\rho_q^-}\Big[\mu_{pq} + \delta_{pq} + (P - \Delta - \Gamma_q + \pi^-)\rho_q^- \\
& \qquad + (P - \Delta - \Gamma_p + \pi^-)\rho_p^+ + \varepsilon_{pq}^+ + u_q^- + u_p^+\Big], \qquad (42) \\
T_q^p - T_{pq}^A \geq \; & \frac{1}{1+\rho_q^+}\Big[\mu_{pq} + \delta_{pq} - (P - \Delta - \Gamma_q + \pi^-)\rho_q^+ \\
& \qquad - (P - \Delta - \Gamma_p + \pi^-)\rho_p^- - \varepsilon_{pq}^- - u_q^+ - u_p^-\Big] - G. \qquad (43)
\end{aligned}
$$

Since all $\boldsymbol{A}_i = [T_i \pm \alpha_i] \in \mathcal{A}$ are $\pi_0$-correct according to precondition [1] of our lemma, Lemma 6 provides $-\pi_0 \leq \mu_{pq} \leq \pi_0$ since $\pi_0 + \overline{\pi}_0 = [-\pi_0, \pi_0]$, so that it follows from (42) that

$$
\begin{aligned}
T_q^p \; & \leq \; T_{pq}^A + \frac{\pi_0 + \delta_{pq} + (P - \Delta - \Gamma_{\min} + \pi^-)(\rho_q^- + \rho_p^+) + \varepsilon_{pq}^+ + u_q^- + u_p^+}{1-\rho_q^-} \\
& \leq \; (k+1)P + \Lambda + \Omega + \Delta - \Delta \\
& \qquad + \frac{\pi_0 + \delta_{\max} + (P - \Delta - \Gamma_{\min} + \pi^-)\rho_{\max} + \varepsilon_{\max}^+ + u_{\max}}{1 - \rho_{\max}^-} \\
& = \; (k+1)P + \Lambda + \Omega + \Delta \\
& \qquad + \frac{\pi_0 + \delta_{\max} + (P - \Gamma_{\min} + \pi^-)\rho_{\max} + \varepsilon_{\max}^+ + u_{\max}}{1 - \rho_{\max}^-} \\
& \qquad - \Delta\Big[\frac{\rho_{\max}}{1-\rho_{\max}^-} + 1\Big] \\
& \leq \; (k+1)P + \Lambda + \Omega + \Delta;
\end{aligned}
$$

the last step is confirmed by plugging in the definition of $\Delta$ according to (26). This result eventually assures that $\boldsymbol{I}_q^p$ from any non-faulty node $p \neq q$ is available for computing the convergence function at node $q$ at latest when $q$'s clock reads $(k+1)P + \Lambda + \Omega + \Delta$, leaving a logical time duration $\Gamma_q$ up to time $T_q^R$ when resynchronization will take place. This is also true for the interval $\boldsymbol{I}_q^q$ from the own node, which can in fact be (pre-)computed at any time, recall the remark following (19). Remembering (28), this means that at least $\gamma_q$ real-time seconds are available for computing the convergence function, as asserted in item (1) of our lemma.

Apart from that, inequalities (42) and (43) may be condensed into the more convenient form

$$
T_q^p - T_{pq}^A = \delta_{pq} + \mathcal{O}(\pi_0 + P\rho_{\max} + G + \varepsilon_{\max}), \tag{44}
$$

where we used $u_{\max} = \mathcal{O}(G)$ according to Assumption 2.

With this preparatory work, we can attack the results given in item (1) of the lemma. According to our expositions in Section 4, each receiving node $q$ relies on (20) to compute the interval

$$I_q^p = I_q^p(t_q^R) = A_{pq}^A + [T_q^R - T_q^p + \delta_{pq} \pm \varepsilon_{pq} + 2G_A] + (T_q^R - T_q^p)\rho_q + u_q + \overline{G} \tag{45}$$

from the accuracy interval $A_{pq}^A$ received from node $p \neq q$. Incorporating the term that accounts for local deterioration (cf. Assumption 3) at node $p$ from local time $T_p$ where the round started (with the accuracy interval $A_p = [T_p \pm \alpha_p]$) up to transmission of the CSM to node $q$ at time $T_{pq}^A$, we easily obtain an expression for the interval of accuracies $\alpha_q^{p,R}$ of $I_q^p = [T_q^{p,R} \pm \alpha_q^{p,R}]$ from (45):

$$\begin{aligned} \alpha_q^{p,R} &= \alpha_p + (T_{pq}^A - T_p)\rho_p + u_p + \varepsilon_{pq} + 2G_A + (T_q^R - T_q^p)\rho_q + u_q + \overline{G} \\ &= \alpha_p + u_p + u_q + \overline{G} + 2G_A + \varepsilon_{pq} \\ &\qquad + (T_{pq}^A - T_p)\rho_p + (T_q^R - T_{pq}^A)\rho_q - (T_q^p - T_{pq}^A)\rho_q. \end{aligned} \tag{46}$$

Since we assumed in Definition 7 that all parameter values appearing in (46) are integer multiples of $G_S$, which is true for $\alpha_p$ according to precondition [0] of our lemma as well, $\alpha_q^{p,R}$ is also an integer multiple of $G_S$ as asserted in item (1). Considering

$$f(c) = (c-a)r_p + (b-c)r_q \leq \begin{cases} (c_{\max} - a)r_p + (b - c_{\max})r_q & \text{if } r_p \geq r_q, \\ (c_{\min} - a)r_p + (b - c_{\min})r_q & \text{if } r_p \leq r_q \end{cases}$$

for $r_p, r_q \geq 0$ (and similarly —with $c_{\max}$ and $c_{\min}$ exchanged— for the lower bound), which follows immediately from $f(c) = c(r_p - r_q) - ar_p + br_q$, we obtain

$$\begin{aligned} f(c) &\leq (c_{\max} - a)r_p + (b - c_{\max})r_q + \max\{r_q - r_p, 0\}(c_{\max} - c_{\min}) \\ f(c) &\geq (c_{\max} - a)r_p + (b - c_{\max})r_q - \max\{0, r_p - r_q\}(c_{\max} - c_{\min}) \end{aligned} \tag{47}$$

Using the above upper bound in equation (46) for $\alpha_q^{p,R}$ and recalling (39), $T_q^R = (k+1)P + \Lambda + \Omega + \Delta + \Gamma_q$ in conjunction with (38), and relation (44), we obtain the upper bound

$$\begin{aligned} \alpha_q^{p,R} \subseteq\; & \alpha_p + u_p + u_q + \overline{G} + 2G_A + \varepsilon_{pq} \\ & + (P - \Delta - \Gamma_p + \pi^-)\rho_p + (\Gamma_q + \Delta)\rho_q \\ & + (\Lambda + \Omega)[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}] \\ & - \delta_{pq}\rho_q + \mathcal{O}(\pi_0 + P\rho_{\max} + G + \varepsilon_{\max})\rho_q; \end{aligned} \tag{48}$$

remembering that $\pi_0 \leq \pi$ according to precondition [2] of our lemma, the result stated in (29) follows. For (30), we just have to employ in the lower bound (47) instead of the upper bound in the derivation above, which amounts to replacing the term $(\Lambda + \Omega)[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}]$ in (48) by $-(\Lambda + \Omega)[-\max\{0, \rho_p^- - \rho_q^-\}, \max\{0, \rho_p^+ - \rho_q^+\}]$.

We still have to investigate $\alpha_q^{p,R}$ in case of $p = q$. Incorporating the term $(T_{qq}^A - T_q)\rho_q + u_q$, which accounts for deterioration at node $q$ from local time $T_q$ up to $T_{qq}^A = (k+1)P$, the instant of the virtual "loop-back transmission", in the expression for $I_q^q$ in (20) provides

$$\alpha_q^{q,R} = \alpha_q + (T_q^R - T_q)\rho_q + u_q,$$

from which (31) follows immediately.

The above derivations for accuracy immediately carry over to $\pi_q^p$-correctness of $I_q^p$ for a suitably chosen $\pi_q^p$: Given that $A_p$ was $\pi_0$-correct at the beginning of round $k$ (at real-time $t_p$), we only have to add the maximum uncertainities caused by the drift and delay compensation operations. Recall that internal global time for any fixed round progresses as real-time does, so that maintaining accuracy w.r.t. real-time by enlarging the interval automatically maintains "accuracy" w.r.t. internal global time as well. Note that the term $2G_A$ accounting for limited accuracy transmission resolution (see Assumption 4) can be ignored here, since precision intervals like $\pi_0$ are not handled by the algorithm but rather computed in our analysis. Adopting (46) appropriately, literally the same derivation that lead to (29) resp. (31) provides (32) resp. (33).

Moreover, by virtue of Lemma 5, we find that the set $\mathcal{N}_q$ of all non-faulty $I_q^p$ at node $q$ is $\pi_q^H$-correct for $\pi_q^H = \bigcup_{p \neq q} \pi_q^p$. Straightforward majorizations of (32) easily confirm the value given in (34); note that setting $\rho_p = \rho_{\max}$ causes the term involving $[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}]$ to vanish. Finally, $\pi_q^q \subseteq \pi_q^H$ is also true by virtue of the technical condition $\delta_{\min}\rho_{\max} \subseteq \varepsilon_{\max}$ according to (16), so that genuinely $\pi_q^H = \bigcup_p \pi_q^p$ as required.

To prove item (2), we note that the asserted $\pi^H$-correctness of the shifted set $\mathcal{N}$ is a straightforward consequence of Lemma 5 applied to $\bigcup_{q \text{ non-faulty}} \text{shift}_{t''} \mathcal{N}_q$. The bound stated in (35) follows from

$$\begin{aligned}
\pi^H &= \bigcup_q \pi_q^H \\
&\subseteq \pi_0 + 2u_{\max} + \overline{G} + \varepsilon_{\max} \\
&\qquad + (P - \Delta - \Gamma_{\min})\rho_{\max} + (\Gamma_{\max} + \Delta - \delta_{\min})\rho_{\max} \\
&\qquad + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}.
\end{aligned}$$

Finally, drawing our attention to the set $\mathcal{N}^p$ of intervals obtained at different (non-faulty) receivers for the same broadcast (of non-faulty node $p$), Lemma 5 applied to $\bigcup_{q \neq p} \text{shift}_{t''}\{I_q^p\}$ yields $\pi^p = \bigcup_{q \neq p} \pi_q^p$, and trivial majorizations easily provide (36). Again, $\pi_q^q \subseteq \pi^p$ by virtue of (16), so that $\pi^p = \bigcup_q \pi_q^p$ as required.

In addition, it is clear that almost the same accuracy interval $A_{pq}^A$ appears in $I_q^p$ of any receiver $q$. More specifically, the only difference is the deterioration that occurs at the sender $p$ between $T_{\min}^p = \min_q T_{pq}^A$ and the particular $T_{pq}^A$ under consideration. Conceptually, this may be viewed as if node $p$ has commenced with an interval $[T_{\min}^p \pm o]$ of length 0. Starting from an equation adopted from (45) in a similar way, we obtain

$$\pi_I \subseteq \bigcup_{q, p \neq q} \Big[ (T_{pq}^A - \min_q T_{pq}^A)\rho_p + (B-1)u_p + \varepsilon_{pq}$$

$$+ (T_q^R - T_{pq}^A)\rho_q - (T_q^p - T_{pq}^A)\rho_q + u_q\Big] + \overline{G}$$
$$\subseteq \varepsilon_{\max} + Bu_{\max} + \overline{G} + \bigcup_{q,p\neq q} (T_q^R - \min_q T_{pq}^A)\rho_{\max} - \delta_{\min}\rho_{\max}$$
$$+ \mathcal{O}(\pi_0 + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}$$

by majorizing $\rho_p$, $\rho_q$ by $\rho_{\max}$ and using (44). Note that $T_{pq}^A - \min_q T_{pq}^A = 0$ in case of a broadcast network ($B = 1$, cf. Assumption 4), so that no deterioration occurs here. The result stated in (37) follows from recalling that $T_q^R - \min_q T_{pq}^A \leq \Lambda + \Omega + \Delta + \Gamma_{\max}$ by virtue of (38) and the definition of the logical resynchronization time $T_q^R$. Note that the contribution $(T_q^R - (k+1)P)\rho_q = (\Lambda + \Omega + \Delta + \Gamma_q)\rho_q$ of the own node $q$ to $\pi^I$ is again covered by expression (37) due to the technical condition (16). This eventually completes the proof of Lemma 11. □

It is important to understand that the set $\mathcal{N}^p$ is $\pi^H$-correct and $\pi_I$-precise, but not necessarily $\pi_I$-correct. After all, we cannot assume that internal global time $\tau^k(t)$ of the current round $k$ lies in the intersection of the (quite small) $\pi_I$-precision intervals associated with the elements of $\mathcal{N}^p$.

Next we will provide the abstract properties of the class of convergence functions considered in this paper. We start with the following definition:

DEFINITION 10 (TRANSLATION INVARIANCE, WEAK MONOTONICITY) *Given two sets $\mathcal{I} = \{I_1, \ldots, I_n\}$ and $\mathcal{J} = \{J_1, \ldots, J_n\}$ of $n \geq 1$ accuracy intervals, an interval-valued function $f()$ of $n \geq 1$ interval arguments is called*

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

*(2)* translation invariant *iff $f(I_1 + \Delta, \ldots, I_n + \Delta) = f(I_1, \ldots, I_n) + \Delta$ for any real $\Delta$.*

Note that a weakly monotonic function satisfies this property for both accuracy intervals and associated $\pi$-precision intervals, hence $\hat{I}_j \subseteq \hat{J}_j$ with ref$(\hat{I}_j)$ = ref$(\hat{J}_j)$ for all $j$ implies $f(\hat{\mathcal{I}}) \subseteq f(\hat{\mathcal{J}})$.

DEFINITION 11 (INTERVAL CONVERGENCE FUNCTION) *Let $\mathcal{I}_p = \{I_p^1, \ldots, I_p^n\}$ and $\mathcal{I}_q = \{I_q^1, \ldots, I_q^n\}$ be two ordered sets of $n$ compatible intervals representing t each, which are in accordance with a given abstract fault model $\mathcal{F}$. Provided that*

*[0] the accuracies of any non-faulty $I_p^i = [T_p^i \pm \alpha_p^i]$ are integer multiples of $G_S$ satisfying $\alpha_p^i \subseteq \beta_p^i \in \mathcal{B}_p$ for a given set of accuracy bounds $\mathcal{B}_p = \{\beta_p^1, \ldots, \beta_p^n\}$ (and analogous for $I_q^i$ with set of accuracy bounds $\mathcal{B}_q$),*

*[1] any non-faulty $I_p^i$ is $\pi_p^i$-correct for $\pi_p^i \in \mathcal{P}_p = \{\pi_p^1, \ldots, \pi_p^n\}$ denoting a given set of precision bounds (and analogous for $I_q^i$ with set of precision bounds $\mathcal{P}_q = \{\pi_q^1, \ldots, \pi_q^n\}$),*

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

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

*define*

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

*The generic convergence function* $\mathcal{CV}$ *is assumed to be translation invariant and weakly monotonic, computing accurate intervals with reference point and accuracies being integer multiples of* $G_S$*, characterized by*

*(1) weakly monotonic (w.r.t. any interval argument)* accuracy preservation function $\aleph()$*, which ensures that* $\boldsymbol{\alpha}_p' \subseteq \aleph(\mathcal{B}_p, \mathcal{P}_p, \pi^H, \pi_I)$ *and* $\boldsymbol{\alpha}_q' \subseteq \aleph(\mathcal{B}_q, \mathcal{P}_q, \pi^H, \pi_I)$*,*

*(2) weakly monotonic* precision preservation function $\boldsymbol{\Phi}()$*, which guarantees that* $\boldsymbol{R}_p$ *is* $\boldsymbol{\Phi}(\mathcal{P}_p, \pi^H, \pi_I)$*-correct and* $\boldsymbol{R}_q$ *is* $\boldsymbol{\Phi}(\mathcal{P}_q, \pi^H, \pi_I)$*-correct, with* $|\boldsymbol{\Phi}(\mathcal{P}, \pi^H, \pi_I)| = \mathcal{O}(\pi^H)$ *for* $\pi^H = |\pi^H|$*,*

*(3) (weakly) monotonic* precision enhancement function $\Pi()$*, which ensures that the set* $\{\boldsymbol{R}_p, \boldsymbol{R}_q\}$ *is* $\pi_0$*-precise for some (=any)* $\pi_0$ *satisfying* $|\pi_0| = \pi_0 = \Pi(\mathcal{P}, \pi^H, \pi_I)$*, with* $\Pi(\mathcal{P}, \pi^H, \pi_I) < \pi^H = |\pi^H|$.

Note that it is possible to provide additional arguments to any of the characteristic functions. For example, the set of accuracy bounds $\mathcal{B}_p$ could be provided to $\boldsymbol{\Phi}()$ as well; we reduced the number of arguments for brevity.

It is worth mentioning that our precision preservation function $\boldsymbol{\Phi}()$ replaces "accuracy" $\alpha$ in the (non interval-based) characterization of convergence functions of [16]. Informally, $\boldsymbol{\Phi}()$ quantifies how well a new clock fits to the clocks of the current round, which seems to be the core property characterized by $\alpha$.

The following lemma describes the result of applying the generic convergence function to the intervals resulting from an FME as set forth by Lemma 11:

LEMMA 12 (PRECISION PRESERVATION) *Let* $\boldsymbol{A}_p = [T_p \pm \boldsymbol{\alpha}_p] = \boldsymbol{C}_p^k(t_p^{R,k-1})$ *be the accuracy interval of node p's interval clock at real-time* $t_p = t_p^{R,k-1}$*, when round k (for some fixed* $k \geq 0$*) starts, and denote by* $\mathcal{A}$ *the subset of the* $\boldsymbol{A}_p$*'s of those nodes p that remain non-faulty during round k w.r.t. a given abstract fault model* $\mathcal{F}$*. Let* $T_q^R = (k+1)P + \Lambda + \Omega + \Delta + \Gamma_q$ *be the logical time when the k + 1-th resynchronization instant —happening at real-time* $t_q^R = t_q^{R,k}$*— is scheduled at node q, and let* $\boldsymbol{I}_q^p = \boldsymbol{I}_q^p(t_q^R)$ *be the interval (20) that is obtained at node q as the result of delay and drift compensation of a node p's accuracy interval transmitted during the FME in round k.*

*Assume that the set $\mathcal{I}_q$ of intervals $I_q^p$ available at node $q$ is subsequently fed into a convergence function $\mathcal{CV}$ characterized by accuracy preservation $\aleph()$, precision preservation $\Phi()$, and precision enhancement $\Pi()$.*

*If, at the beginning of round $k$,*

*[0] the accuracies of any $A_p = [T_p \pm \alpha_p] \in \mathcal{A}$ are integer multiples of $G_S$ bounded according to*

$$\alpha_p \subseteq \beta_p \in \mathcal{B} \tag{49}$$

*for a given set of accuracy bounds $\mathcal{B} = \{\beta_1, \ldots, \beta_n\}$,*

*[1] $shift_{t'}(\mathcal{A})$, $t' \geq \max_p t_p$ arbitrary, is $\pi_0'$-correct w.r.t. internal global time of round $k$ for some $\pi_0' \subseteq \pi_0$, where $\pi_0$ is a solution of the equation*

$$|\pi_0| = \Pi(\mathcal{P}, \pi^H, \pi_I) \tag{50}$$

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

$$\begin{aligned}
\pi^p &= \pi_0 + u_p + u_{\max} + \overline{G} + \varepsilon_{\max} \\
&\quad + (P - \Delta - \Gamma_p)\rho_p + (\Gamma_{\max} + \Delta - \delta_{\min})\rho_{\max} \\
&\quad + (\Lambda + \Omega)\big[-(\rho_{\max}^- - \rho_p^-), \rho_{\max}^+ - \rho_p^+\big] \\
&\quad + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, && (51)\\
\pi^H &= \pi_0 + 2u_{\max} + \overline{G} + \varepsilon_{\max} + (P + \Gamma_{\max} - \Gamma_{\min} - \delta_{\min})\rho_{\max} \\
&\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, && (52)\\
\pi_I &= \varepsilon_{\max} + Bu_{\max} + \overline{G} + (\Lambda + \Omega + \Delta + \Gamma_{\max} - \delta_{\min})\rho_{\max} \\
&\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, && (53)
\end{aligned}$$

*[2] any $A_p \in \mathcal{A}$ satisfies $T_p \in [kP + \Lambda + \Omega + \Delta + \Gamma_p \pm \pi']$ for $\pi' \subseteq \pi$ defined by*

$$\pi = \overline{\Phi}(\mathcal{P}, \pi^H, \pi_I) + \pi_0 + P\rho_{\max} + u_{\max} + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, \tag{54}$$

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

*[3] broadcast delay $\Lambda + \Omega$, transmission delay $\Delta$, computation time compensation $\Gamma_p$, and round period $P$ are as defined in item [3] of Lemma 11,*

*then the set $\mathcal{R}$ of intervals*

$$R_q = R_q(t_q^R) = [T_q' \pm \alpha_q'] = \mathcal{CV}(\mathcal{I}_q)$$

*provided by the application of the convergence function $\mathcal{CV}$ to the set $\mathcal{I}_q$ of compatible intervals (20) resulting from the FME at a non-faulty node $q$ satisfies*

*(0)* $R_q$ *is accurate with accuracies being integer multiples of* $G_S$ *bounded according to*

$$\alpha'_q \subseteq \aleph(\mathcal{B}_q, \mathcal{P}_q, \pi^H, \pi_I),$$

*where the set* $\mathcal{B}_q = \{\beta_q^1, \ldots, \beta_q^n\}$ *of node q's accuracy bounds is defined by*

$$\begin{aligned}\beta_q^p \;=\; & \beta_p + u_p + u_q + \overline{G} + 2G_A + \varepsilon_{pq} \\ & + (P - \Delta - \Gamma_p)\rho_p + (\Gamma_q + \Delta - \delta_{pq})\rho_q \\ & + (\Lambda + \Omega)\big[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}\big] \\ & + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}\end{aligned} \tag{55}$$

*for* $p \neq q$ *and*

$$\beta_q^q = \beta_q + u_q + P\rho_q + \mathcal{O}(\pi)\rho_q \tag{56}$$

*with* $\beta_p \in \mathcal{B}$*; the set* $\mathcal{P}_q = \{\pi_q^1, \ldots, \pi_q^n\}$ *of node q's precision bounds* $\pi_q^p \subseteq \pi^p \subseteq \pi^H$ *is defined by*

$$\begin{aligned}\pi_q^p \;=\; & \pi_0 + u_p + u_q + \overline{G} + \varepsilon_{pq} + (P - \Delta - \Gamma_p)\rho_p + (\Gamma_q + \Delta - \delta_{pq})\rho_q \\ & + (\Lambda + \Omega)\big[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}\big] \\ & + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}\end{aligned} \tag{57}$$

*for* $p \neq q$ *and*

$$\pi_q^q = \pi_0 + u_q + P\rho_q + \mathcal{O}(\pi)\rho_q. \tag{58}$$

*(1)* $R_q$ *is* $\Phi(\mathcal{P}_q, \pi^H, \pi_I)$*-correct w.r.t. internal global time of round k,*

*(2)* $shift_{t'}(\mathcal{R})$*,* $t' \geq \max_p t_p^R$ *arbitrary, is* $\pi_0$*-correct w.r.t. the newly defined internal global time of round* $k+1$*,*

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

$$\begin{aligned}t_p^R - t_q^R \;\subseteq\; & \Gamma_p - \Gamma_q + [-\pi_0, \pi_0] + P(\rho_p + \overline{\rho}_q) + u_p + \overline{u}_q \\ & + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})[-\rho_{\max}, \rho_{\max}]\end{aligned} \tag{59}$$

*for* $\pi_0 = |\pi_0|$*,*

*(4) the maximum clock adjustment* $\Upsilon_q$ *applied to the clock of any non-faulty node q satisfies* $\Upsilon_q \in \pi_q$ *for* $\pi_q \subseteq \pi$ *defined by*

$$\pi_q = \overline{\Phi}(\mathcal{P}_q, \pi^H, \pi_I) + \pi_0 + P\rho_q + u_q + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, \tag{60}$$

*hence* $T'_q \in [(k+1)P + \Lambda + \Omega + \Delta + \Gamma_q \pm \pi_q]$ *for any* $R_q \in \mathcal{R}$*.*

Proof: First of all, we establish some coarse *a priori* bounds on the various precision values given in precondition [1] of our lemma, which will be required for applying Lemma 11: Since $\pi_0$ satisfies (50), it follows from $\Pi() < \pi^H$ according to item (3) of Definition 11 that $\pi^H > \pi_0 = C\pi^H$ for $C < 1$. Therefore, (52) reveals $(1-C)\pi^H = \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$, so that this remainder term applies for $\pi^H$ and $\pi_0$ as well. Moreover, plugging in $|\Phi(\mathcal{P}_p, \pi^H, \pi_I)| \leq |\Phi(\mathcal{P}, \pi^H, \pi_I)| = \mathcal{O}(\pi^H)$ —as justified by weak monotonicity of $\Phi()$ and the bound from item (2) of Definition 11— into (54) resp. (60) reveals that $\pi_p \leq \pi = \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$ for any $p$ as well.

Now, since our preconditions are the same ones as required by Lemma 11, it follows from (29) resp. (31) that the accuracy of a (non-faulty) interval $I_q^p \in \mathcal{N}_q \subseteq \mathcal{I}_q$ is bounded according to (55) resp. (56), which defines the set of accuracy bounds $\mathcal{B}_q$ required for $\mathcal{CV}$. Similarly, we know from (32) resp. (33) that any $I_q^p \in \mathcal{N}_q$ is $\pi_q^{p\prime}$-correct for $\pi_q^{p\prime} \subseteq \pi_q^p$ given by (57) resp. (58). In addition, (35) implies that any $I_q^p \in \mathcal{N}$ is $\pi^{H\prime}$-correct with $\pi^{H\prime} \subseteq \pi^H$ defined in (52), and (36) establishes that any $I_q^p \in \mathcal{N}^p$ is $\pi^{p\prime}$-correct for $\pi^{p\prime} \subseteq \pi^p$ given by (51); note that $\mathcal{O}(\pi') \leq \mathcal{O}(\pi) = \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$, as shown above. Moreover, from (37) it follows that $\mathcal{N}^p$ is also $\pi_I'$-precise with $\pi_I' \subseteq \pi_I$ given by (53); again, $\mathcal{O}(\pi_0') \leq \mathcal{O}(\pi_0) = \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$.

Therefore, we have established bounds on all the arguments of the characteristic functions of $\mathcal{CV}$. Hence, the statements asserted in item (0) and (1) of our lemma follow immediately. Note that weak monotonicity of $\mathcal{CV}$ (and $\aleph()$ as well) is required here to carry over bounds on the source intervals to bounds on the result. Since (50) in conjunction with item (3) of Definition 11 implies that $\text{shift}_{t'}(\mathcal{R})$, $t' \geq \max_p t_p^R$ arbitrary is $\pi_0$-precise, we can define internal global time $\tau^{k+1} = \tau^{k+1}(t')$ for the new round $k+1$ to be an arbitrary point that lies in the intersection of the intervals in $\text{shift}_{t'}(\mathcal{R})$ (recall Definition 9), so that this set is actually $\pi_0$-correct w.r.t. $\tau^{k+1}$, as asserted in item (2) of the lemma.

To prove the statement of item (3), we recall that node $p$ resynchronizes at real-time $t_p^R$ when the virtual clock $C_p^k$ used in round $k$ displays $T_p^R = (k+1)P + \Lambda + \Omega + \Delta + \Gamma_p$. Since round $k$ started when $C_p^k$ read $T_p \geq kP + \Lambda + \Omega + \Delta + \Gamma_p - \pi^-$ according to precondition [2] of our lemma, our usual argument provides that the initial precision $\pi_0$ has deteriorated to

$$\begin{aligned}\pi_{o,p} &= \pi_0 + (T_p^R - T_p)\rho_p + u_p \subseteq \pi_0 + u_p + (P + \pi^-)\rho_p \\ &\subseteq \pi_0 + u_p + P\rho_p + \mathcal{O}(\pi)\rho_{\max} \qquad (61)\end{aligned}$$

when $t_p^R$ is reached; that is, the virtual clock $C_p^k(t_p^R) = [T_p^R \pm \alpha_p^R]$ of a non-faulty node $p$ is $\pi_{o,p}$-correct. Therefore, Lemma 6 immediately provides

$$t_p^R - t_q^R \subseteq T_p^R - T_q^R + \pi_{o,p} + \overline{\pi}_{o,q}$$

for all non-faulty nodes $p$ and $q$. Plugging in the definition of the logical resynchronizaton times $T_p^R$ and $T_q^R$ and (61) while recalling $\pi + \overline{\pi} = [-\pi, \pi]$ for any $\pi$, item (3) of our lemma follows.

Turning our attention to item (4), we recall that the new virtual clock $C_q^{k+1}(t_q^R)$ at node $q$ is initialized to $R_q = [T_q' \pm \alpha_q']$, so that it is $\Phi(\mathcal{P}_q, \pi^H, \pi_I)$-correct w.r.t. $\tau^k$ by virtue of item (1) of our lemma. On the other hand, the virtual clock $C_q^k(t_q^R)$ was established above to be $\pi_{o,q}$-correct w.r.t. $\tau^k$ according to (61). Lemma 7 thus yields

$$\Upsilon_q = T_q' - T_q^R \in \pi_{o,q} + \overline{\Phi}(\mathcal{P}_q, \pi^H, \pi_I). \tag{62}$$

Using $\Phi(\mathcal{P}_q, \pi^H, \pi_I) \subseteq \Phi(\mathcal{P}, \pi^H, \pi_I)$ due to weak monotonicity, (60) and also $\pi_q \subseteq \pi$ follows, which in turn justifies our choice of (54). The bound on $T_q'$ given in item (4) is an immediate consequence of $\Upsilon_q = T_q' - T_q^R$, eventually completing the proof of our lemma. □

By virtue of the Lemma 12, it is not difficult to provide the major result of this section:

THEOREM 1 (PRECISION REF. ACCURACY INSTANTANEOUS CORRECTION) *Running in a system complying to Assumptions 1–4, the clock synchronization algorithm of Definition 7 using the generic convergence function $\mathcal{CV}$ —characterized by accuracy preservation $\aleph()$, precision preservation $\Phi()$, and precision enhancement $\Pi()$ subject to a given abstract fault model $\mathcal{F}$— guarantees accuracy and precision for all rounds as follows:*

*(0) The accuracy interval $A_q^{k+1} = [T_q^{k+1} \pm \alpha_q^{k+1}]$ provided by the local interval clock of a non-faulty node $q$ at the beginning of round $k+1$, $k \geq 0$, satisfies $\alpha_q^{k+1} \subseteq \beta_q^{k+1}$ for*

$$\begin{aligned} \beta_q^{k+1} &= \aleph(\mathcal{B}_q^{k+1}, \mathcal{P}_q, \pi^H, \pi_I), \\ \beta_q^0 &= \alpha_q^0, \end{aligned} \tag{63}$$

*where $\mathcal{B}_q^{k+1} = \{\beta_q^{1,k+1}, \ldots, \beta_q^{n,k+1}\}$ is defined recursively by*

$$\begin{aligned} \beta_q^{p,k+1} &= \beta_p^k + u_p + u_q + \overline{G} + 2G_A + \varepsilon_{pq} \\ &\quad + (P - \Delta - \Gamma_p)\rho_p + (\Gamma_q + \Delta - \delta_{pq})\rho_q \\ &\quad + (\Lambda + \Omega)[-\max\{\rho_q^- - \rho_p^-, 0\}, \max\{\rho_q^+ - \rho_p^+, 0\}] \\ &\quad + O(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max} \end{aligned} \tag{64}$$

*for $p \neq q$ and*

$$\beta_q^{q,k+1} = \beta_q^k + u_q + P\rho_q + O(\pi)\rho_q. \tag{65}$$

*(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 last non-faulty clock)*

$$\begin{aligned} \pi_{0,\max} &= \pi_0 + u_{\max} + G + (\Gamma_{\max} - \Gamma_{\min})\rho_{\max} \\ &\quad + O(P\rho_{\max}^2 + G\rho_{\max} + \varepsilon_{\max}\rho_{\max}) \end{aligned} \tag{66}$$

*with $\pi_0 = |\pi_0|$, where $\pi_0$ is a solution of the equation*

$$|\pi_0| = \Pi(\mathcal{P}, \pi^H, \pi_I) \tag{67}$$

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

$$\begin{aligned}
\pi^p &= \pi_0 + u_p + u_{\max} + \overline{G} + \varepsilon_{\max} \\
&\quad + (P - \Delta - \Gamma_p)\rho_p + (\Gamma_{\max} + \Delta - \delta_{\min})\rho_{\max} \\
&\quad + (\Lambda + \Omega)\big[-(\rho^-_{\max} - \rho^-_p), \rho^+_{\max} - \rho^+_p\big] \\
&\quad + \mathcal{O}(\pi + P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, & (68)\\
\pi^H &= \pi_0 + 2u_{\max} + \overline{G} + \varepsilon_{\max} + (P + \Gamma_{\max} - \Gamma_{\min} - \delta_{\min})\rho_{\max} \\
&\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, & (69)\\
\pi_I &= \varepsilon_{\max} + Bu_{\max} + \overline{G} + (\Lambda + \Omega + \Delta + \Gamma_{\max} - \delta_{\min})\rho_{\max} \\
&\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}. & (70)
\end{aligned}$$

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

$$\begin{aligned}
\pi_{\max} &= \max\Big\{\pi^- + u^+_{\max} + (\Gamma_{\max} - \Gamma_{\min})\rho^+_{\max}, \\
&\qquad \pi^+ + u^-_{\max} + (\Gamma_{\max} - \Gamma_{\min})\rho^-_{\max}, \ \pi_0 + u_{\max} + P\rho_{\max}\Big\} \\
&\quad + G + \mathcal{O}(P\rho^2_{\max} + G\rho_{\max} + \varepsilon_{\max}\rho_{\max}) & (71)
\end{aligned}$$

*with*

$$\begin{aligned}
\pi &= \overline{\Phi}(\mathcal{P}, \pi^H, \pi_I) + \pi_0 + u_{\max} + P\rho_{\max} \\
&\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}, & (72)
\end{aligned}$$

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

*(3) Resynchronization of any two non-faulty nodes $p$, $q$ occurs within real-time $t^R_p - t^R_q$ satisfying*

$$\begin{aligned}
t^R_p - t^R_q &\subseteq \Gamma_p - \Gamma_q + [-\pi_0, \pi_0] + u_p + \overline{u}_q + P(\rho_p + \overline{\rho}_q) \\
&\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})[-\rho_{\max}, \rho_{\max}], & (73)
\end{aligned}$$

*where adjustments of at most $\Upsilon_q \in \pi_q \subseteq \pi$ defined by*

$$\pi_q = \overline{\Phi}(\mathcal{P}_q, \pi^H, \pi_I) + \pi_0 + u_q + P\rho_q + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}$$

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

**Proof:** The above results are established by conducting an induction proof on the round $k$: Assuming that the accuracy intervals $A_p = [T_p \pm \alpha_p] = C^k_p(t^{R,k-1}_p)$ of all non-faulty nodes $p$ are $\pi_0$-correct at the beginning of round $k$, in the sense that [1] $\mathrm{shift}_{t'}(\mathcal{A})$, $t' \geq \max_p t_p^{R,k-1}$ arbitrary, is $\pi_0$-correct, [2] $T_p \in [kP + \Lambda + \Omega + \Delta + \Gamma_p \pm \pi]$, and [3] $\alpha_p \subseteq \beta_p^k$, we show that the accuracy intervals $A'_p = [T'_p \pm \alpha'_p] = C_p^{k+1}(t_p^{R,k})$ at the beginning of the succeeding round $k+1$ satisfy these precision properties (and the accuracy bounds $\alpha'_p \subseteq \beta_p^{k+1}$) as well. However, this —and hence most results stated in our theorem— follows directly from Lemma 12. The initial case $k = 0$ is immediately implied by the initial synchronization assumption in the algorithm's Definition 7. More specifically, the recursive definitions (63) and (64) follow directly from item (0) of Lemma 12; $\alpha_q^0$ denotes the initial accuracy of node $q$ at the beginning of round 0 according to Definition 7. Similarly, item (3) of our theorem just combines item (3) and (4) of Lemma 12.

It only remains to derive the expressions for the observable precisions $\pi_{0,\max}$ and $\pi_{\max}$ given in item (1) and (2) of our theorem. Let $\mathcal{C}$ be a certain set of just resynchronized non-faulty virtual interval clocks $C_p^{k+1}(t_p^R)$ at their respective resynchronization times $t_p^R = t_p^{R,k}$. From item (3) of Lemma 12 in conjunction with $\pi_0 = \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$ established in its proof, it follows that $t_p^R - t_q^R \leq \Gamma_{\max} - \Gamma_{\min} + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$ for any two non-faulty nodes $p$ and $q$, so that item (2) of Lemma 10 provides us with the precision $\pi_{x,\max}$ of the clocks in $\mathcal{C}$ at real-time $t_l^R = \max_p t_p^R$, when the last non-faulty node in the system resynchronizes: Provided that any $C_p^{k+1}(t_p^R) \in \mathcal{C}$ is $\pi_x$-correct, we obtain

$$\begin{aligned}\pi_{x,\max} &\subseteq \pi_x + \bigcup_p \frac{t_l^R - t_p^R + u_p^-}{1 - \rho_p^-}\rho_p + u_p + G_\rho \\ &\subseteq \pi_x + (\Gamma_{\max} - \Gamma_{\min})\rho_{\max} + u_{\max} + G \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}; \end{aligned} \tag{74}$$

recall $u_{\max} = \mathcal{O}(G)$ from Assumption 2 and the definition of $G_\rho$ in (13).

To establish $\pi_{0,\max}$, we have to consider the set $\mathcal{C}$ of all non-faulty virtual clocks $C_p^{k+1}(t_p^R)$. Since $\mathrm{shift}_{t_l^R}\mathcal{C}$ is $\pi_0$-correct w.r.t. internal global time $\tau^{k+1}$ due to item (2) of Lemma 12, we have to plug in $\pi_x = \pi_0$ in (74), which provides (66).

Before we can attack overall precision $\pi_{\max}$, we need some technical preparations. Consider some $t_0$, $t_1 \geq t_0$ being in synchrony with a node's clock $C(t)$, and denote by $I'(t)$ the interval obtained by drift compensation of a $\pi_0$-correct (initial) interval $I_0 = I(t_0)$ dragged from $t_0$ to $t$ at that node. Then, it is not difficult to see that $I'(t)$ for <u>any</u> $t$ within $t_0 \leq t \leq t_1$ is $\pi$-correct for

$$\pi = \pi_0 + (T_1 - T_0)\rho + u + G_\rho, \tag{75}$$

where $T_0 = C(t_0)$ and $T_1 = C(t_1)$. In fact, monotonicity of $C(t)$ implies $T_1 \geq T$ so that (75) follows immediately from (12) for $t_1 = t$. However, note the subtle fact that we cannot always infer $\pi$-correctness of $I'(t)$ for any $t \leq t_1$ from $\pi$-correctness of $I'(t_1)$. Whereas this is justified when $t_1$ is <u>not</u> in synchrony with $C(t)$, we must explicitly account for clock granularity (which amounts to adding $G_\rho$) if $t_1$ is synchronous.

Returning to our problem of determining the maximum observable overall precision, we know from (61) that the virtual interval clock $C_p^k(t_p^R) = [T_p^R \pm \alpha_p^R]$ of a non-faulty node $p$ is $\pi_{o,p}$-correct at $t_p^R$. Therefore, by using majorization in (61) and recalling (75), it follows that $C_p^k(t)$ of any non-faulty node $p$, for any $t \leq t_p^R$ in round $k$, is $\pi_{o,.}$-correct with

$$\pi_{o,.} = \pi_0 + P\rho_{\max} + u_{\max} + G_\rho + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}. \tag{76}$$

Assuming that the last resynchronizing node $l$ in round $k$ actually attains this maximum, we now consider the set $C$ containing just a single virtual clock $C_p^{k+1}(t_p^R)$, of any non-faulty node $p \neq l$. From item (1) of Lemma 12 and weak monotonicity of $\Phi()$, we know that $C_p^{k+1}(t_p^R)$ is $\Phi(\mathcal{P}, \pi^H, \pi_I)$-correct w.r.t. internal global time of round $k$. Plugging this into (74) provides the observable precision

$$\begin{aligned}\pi^M &= \Phi(\mathcal{P}, \pi^H, \pi_I) + (\Gamma_{\max} - \Gamma_{\min})\rho_{\max} + u_{\max} + G \\ &\quad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max}\end{aligned} \tag{77}$$

of $C_p^{k+1}(t_l^R)$ at time $t_l^R$, when the last node $l$ is about to resynchronize. Of course, by the above reasoning, this precision is in fact valid for $C_p^{k+1}(t)$ for any $t$ satisfying $t_p^R \leq t \leq t_l^R$ as well, since granularity compensation $G_\rho$ is already incorporated in $\pi^M$.

Applying Lemma 7 to $C_l^k(t)$ and $C_p^{k+1}(t)$ while using (76) and (77) reveals that the distance $\Upsilon(t) = \mathrm{ref}(C_p^{k+1}(t)) - \mathrm{ref}(C_l^k(t))$ for any $t$ with $t_p^R \leq t \leq t_l^R$ is bounded by

$$\begin{aligned}\Upsilon(t) &\in \pi_{o,.} + \overline{\pi}_M \\ &\subseteq \pi_0 + P\rho_{\max} + u_{\max} + G + \overline{\Phi}(\mathcal{P}, \pi^H, \pi_I) \\ &\qquad + (\Gamma_{\max} - \Gamma_{\min})\overline{\rho}_{\max} + \overline{u}_{\max} + \overline{G} \\ &\qquad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})(\rho_{\max} + \overline{\rho}_{\max}) \\ &= \pi + (\Gamma_{\max} - \Gamma_{\min})\overline{\rho}_{\max} + \overline{u}_{\max} + [-G, G] \\ &\qquad + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})[-\rho_{\max}, \rho_{\max}],\end{aligned}$$

recall the definition of $\pi$ in (72).

Of course, the maximum of the positive and negative accuracies of the interval above gives the maximum observable precision for the "mixed" case, where virtual clocks of round $k$ and $k+1$ are simultaneously alive. Thus, to determine $\pi_{\max}$, it only remains to find the maximum observable precision for the case where all nodes are still in round $k$, and to take the maximum of both cases. However, we established already that $C_p^k(t)$ and $C_q^k(t)$ for any $t \leq \min\{t_p^R, t_q^R\}$ are at most $\pi_{o,.}$-correct, so that Lemma 7 in conjunction with (76) provides

$$\mho(t) = \mathrm{ref}(C_p^k(t)) - \mathrm{ref}(C_q^k(t))$$

$$\in [-\pi_0, \pi_0] + P[-\rho_{\max}, \rho_{\max}] + [-u_{\max}, u_{\max}] + [-G, G] + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})[-\rho_{\max}, \rho_{\max}].$$

Taking the maximum values of positive and negative accuracies of $\Upsilon(t)$ and $\mho(t)$ eventually provides (71), completing the proof of our theorem. □

### 5.3. Continuous Amortization

Theorem 1 provides results for instantaneous correction, where the local interval clocks —displaying $C_p^k(t_p^{R,k})$— are re-initialized to $C_p^{k+1}(t_p^{R,k})$ at the end of round $k$. Since this simple approach could cause non-continuity and non-monotonicity of local time, applications usually demand some kind of continuous amortization. Such techniques are based on the idea of smoothing out the difference $C_p^{k+1}(t_p^{R,k}) - C_p^k(t_p^{R,k})$ by means of a suitable rate "boost". Continuous amortization has been studied in some detail in [16] and, in particular, in [21], where the non-interval based variant of our algorithm has been introduced and analyzed.

Adopting existing continuous amortization techniques to the interval-based framework involves intricate issues. First of all, continuous amortization only applies to the reference point $C_p(t)$ of a local interval clock $C_p(t) = [L_p(t), U_p(t)] = [C_p(t) - \alpha_p^-(t), C_p(t) + \alpha_p^+(t)]$, whereas the upper and lower envelope can be set instantaneously. However, since the envelopes are not maintained explicitly but rather implicitly via $\alpha_p^-(t)$ and $\alpha_p^+(t)$, it is necessary to compensate any change of $C_p(t)$ caused by continuous amortization by changing $\alpha_p(t)$ appropriately.

Apart from that, it might also happen that instantaneous setting of lower or upper edge cause "negative" values of the accuracies $\alpha_p^-$ or $\alpha_p^+$, since the reference point is not changed to its new value simultaneously with $\alpha_p$. Of course, eventually, accuracies will become positive again, but one should ensure that applications read "negative" accuracy values as zero. Furthermore, the worst case accuracy bounds for the beginning of a round provided by Theorem 1 are not particularly meaningful anymore, except for the total length $\alpha_p = \alpha_p^- + \alpha_p^+$.

Since general applicability of our results would be sacrificed if we tailored our considerations to a particular continuous amortization algorithm, we utilize the following abstract specification:

DEFINITION 12 (CONTINUOUS AMORTIZATION ALGORITHM) *Setting node p's local interval clock $C_p(t)$ displaying $[T \pm \alpha]$ at real-time $t$ to $[T' \pm \alpha']$ (with $T' = T + \Upsilon$ being an integer multiple of $G_S$) is accomplished by*

*(1) adjusting the (intrinsic) inverse rate $r_p^{-1}$ of the local clock $C_p$ to $(1-\psi)r_p^{-1}$ for the next $A$ clock ticks (according to the amortizing clock), where the* amortization rate deviation *$\psi$ and (local time) duration $A \cdot G$ of the* amortization period *are related by*

$$\psi = \frac{\Upsilon}{AG},$$

*(2) instantaneously setting the clock's interval of accuracies to* $\alpha_p = \Upsilon + \alpha' = [-(\alpha'^- - \Upsilon), \alpha'^+ + \Upsilon]$, *and modifying (intrinsic) deterioration, clock reading, etc. appropriately to keep away any effect of continuous amortization from the envelopes* $L_p(t) = C_p(t) + \alpha_p^-(t)$ *and* $U_p(t) = C_p(t) + \alpha_p^+(t)$.

A few additional comments are appropriate here:

- Real-time duration $t_e - t_s$ and logical time one $T_e - T_s = AG$ of the amortization period are related by the adjusted inverse rate $r_p^{-1}(1-\psi)$, namely, $AG = \frac{r_p}{1-\psi}(t_e - t_s)$. It thus follows immediately that $r_p(t_e - t_s) = AG(1-\psi) = AG - \Upsilon$, so that the amortizing clock gains $\Upsilon$ w.r.t. the non-amortizing clock during $t_e - t_s$.
- $T'$ can have higher resolution than $G$, since we assume rate adjustment capabilities providing a clock setting granularity of at least $G_S \leq G$, cf. Assumption 2.
- The way intrinsic deterioration is to be modified to satisfy item (2) of Definition 12 depends on the implementation of the rate adjustment capabilities. If the non-adjusted clock is also available during continuous amortization for maintaining the envelopes, as is the case with the adder-based clock in our UTCSU-ASIC (see [23]), it suffices to subtract the additional $G\psi$ put on any clock tick from $\alpha_p$. For settings that provide the amortizing clock only, e.g., for a VCO-based architecture, it is also necessary to modify the intrinsic deterioration of $\alpha_p$ by replacing $G\rho_p$ with $G(1-\psi)\rho_p$, cf. Assumption 3.

Continuous amortization requires that local clocks are fine-grained rate adjustable. The local clock model introduced in Assumption 2 was provided without discussing the impact of explicit rate adjustment. Adding the required features, however, is relatively straightforward: We simply assume that a local clock guarantees (4) for all sequences of clock ticks, whether being adjusted or not.

ASSUMPTION 5 (RATE ADJUSTED LOCAL CLOCKS) *For any number* $R \geq 1$ *of successive periods of rate adjustment with rate deviation* $\psi_r$ *during* $A_r \geq 1$ *successive clock ticks,* $1 \leq r \leq R$, *the local clock* $C_p(t)$ *of a non-faulty node p satisfies*

$$\theta_{i_R} - \theta_1 \;\geq\; (1-\rho_p^-)\sum_{r=1}^{R}(1-\psi_r)(\Theta_{i_r} - \Theta_{i_{r-1}}) - u_p^-, \tag{78}$$

$$\theta_{i_R} - \theta_1 \;\leq\; (1+\rho_p^+)\sum_{r=1}^{R}(1-\psi_r)(\Theta_{i_r} - \Theta_{i_{r-1}}) + u_p^+, \tag{79}$$

*where* $i_0 = 1$, $i_r = 1 + \sum_{1 \leq l \leq r} A_l$ *for* $1 \leq r \leq R$, *and* $\Theta_j = C_p(\theta_j)$ *for clock tick* $\theta_j$, $1 \leq j \leq i_R$.

Note that we actually face alternating amortization periods ($\psi \neq 0$) followed by a period where the clock runs at its intrinsic rate ($\psi = 0$). The rate adjustment uncertainity in case of $\psi \neq 0$ should be viewed as the maximum real-time deviation of a tick $\theta_j$ from the ideal one (as produced by an ideal continuous clock —running at (inverse) rate $r^{-1} = 1-\psi$— when reaching $\Theta_j$). Since all discrete rate adjustment techniques discussed in conjunction with Assumption 2 are using the intrinsic (non-adjusted) clock for tick modification, we assumed $u_p$ to be independent of $\psi$. Note that rate adjustment uncertainities do not add up during consecutive periods of rate adjustment.

The following lemma is our basic tool for investigating continuous amortization. In what follows, we will denote the amortizing local clock (controlled by the continuous amortization algorithm) by $C_p^{\psi}$, to distinguish it from the instantaneously set clock $C_p$ employed in the previous sections.

LEMMA 13 (DURATION ESTIMATIONS) *Let the synchronous real-times $t_s$ and $t_e$ delimit a single amortization period of an amortizing clock $C_p^{\psi}$, and let $t_0 < t_e$, $t \geq t_s$ be arbitrary, possibly non-synchronous real-times, with $T_s = C_p^{\psi}(t_s)$ and similarly for $T_e$, $T_0$, $T$ denoting the corresponding points in logical time. If $C_p^{\psi}$ is non-faulty during $t - t_0$, we have*

$$\begin{aligned} t - t_0 &\geq (T - T_0)(1 - \rho_p^-) - (T_e' - T_s')(1 - \rho_p^-)\psi \\ &\quad - I_{t_0 \neq \theta_0} G(1 - \rho_p^-)(1 - \psi') - u_p^-, \\ t - t_0 &\leq (T - T_0)(1 + \rho_p^+) - (T_e' - T_s')(1 + \rho_p^+)\psi \\ &\quad + I_{t \neq \theta} G(1 + \rho_p^+)(1 - \psi'') + u_p^+, \end{aligned}$$

*where $T_s' = \max\{T_0, T_s\}$, $T_e' = \min\{T, T_e\}$, $\psi' = \psi$ if $T_0 \geq T_s$ or 0 otherwise, and $\psi'' = \psi$ if $T < T_e$ or 0 otherwise.*

*The converse relations are given by*

$$T - T_0 \geq \frac{t - t_0 - u_p^+}{1 + \rho_p^+} + (T_e' - T_s')\psi - I_{t \neq \theta}(1 - \psi'')G, \tag{80}$$

$$T - T_0 \leq \frac{t - t_0 + u_p^-}{1 - \rho_p^-} + (T_e' - T_s')\psi + I_{t_0 \neq \theta_0}(1 - \psi')G; \tag{81}$$

*the (equivalent) conditions for $\psi' = \psi$ and $\psi'' = \psi$ in the real-time domain are $t_0 \geq t_s$ and $t < t_e$, respectively.*

**Proof:** Using $R = 3$ with $\psi_1 = 0$, $\psi_2 = \psi$, and $\psi_3 = 0$ in Lemma 13 and rewriting (78) and (79) accordingly provides

$$\begin{aligned} \theta_{i_3} - \theta_1 &\geq (1 - \rho_p^-)(\Theta_{i_3} - \Theta_1) - (1 - \rho_p^-)(\Theta_{i_2} - \Theta_{i_1})\psi - u_p^-, \\ \theta_{i_3} - \theta_1 &\leq (1 + \rho_p^+)(\Theta_{i_3} - \Theta_1) - (1 + \rho_p^+)(\Theta_{i_2} - \Theta_{i_1})\psi + u_p^+. \end{aligned}$$

Recalling $\theta_{i_R} - \theta_1 = c(\Theta_{i_R}) - c(\Theta_1)$ and substituting the notation of our lemma, it is not difficult to verify that

$$\begin{aligned} c(T) - c(T_0 + I_{t_0 \neq \theta_0} G) &\geq (T - T_0 - I_{t_0 \neq \theta_0} G)(1 - \rho_p^-) - (T_e' - T_s')(1 - \rho_p^-)\psi \\ &\quad + I_{t_0 \neq \theta_0} G(1 - \rho_p^-)\psi' - u_p^- \end{aligned} \tag{82}$$

and similarly

$$\begin{aligned} c(T + I_{t\neq\theta}G) - c(T_0) \;\leq\; & (T + I_{t\neq\theta}G - T_0)(1 + \rho_p^+) - (T_e' - T_s')(1 + \rho_p^+)\psi \\ & - I_{t\neq\theta}G(1 + \rho_p^+)\psi'' + u_p^+. \end{aligned} \tag{83}$$

Plugging this into (8) provides the lower and upper bound given in the first statement of our lemma. The converse relations (80) and (81) are easily established by a simple algebraic manipulation; the equivalent conditions for $\psi'$ and $\psi''$ in the real-time domain follow from (5). □

Using this lemma, it is easy to show that an amortizing clock $C_p^\psi$ satisfies the same drift bounds as an instantaneously corrected clock. Note that this implies that an amortizing clock is "indistinguishable" from an instantaneously corrected one as far as worst case drift is concerned.

LEMMA 14 (AMORTIZING VS. INSTANTANEOUS CLOCKS) *Let $C_p^\psi(t)$ be an amortizing local.clock makes a state correction $\Upsilon$ during an amortization period of $A$ clock ticks, so that $\psi = \Upsilon/(AG)$. If $t_0 \leq t_s$ denotes some synchronous real-time not later than the time $t_s$ when amortization commences, with $T_0 = C_p^\psi(t_0) = C_p(t_0)$, we have for any real-time $t \geq t_s$*

$$\frac{t - t_0 - u_p^+}{1 + \rho_p^+} - I_{t\neq\theta}G \leq C_p^\psi(t) - C_p(t_0) \leq \frac{t - t_0 + u_p^-}{1 - \rho_p^-} + \Upsilon \tag{84}$$

*in case of $\Upsilon > 0$, and*

$$\frac{t - t_0 - u_p^+}{1 + \rho_p^+} + \Upsilon - I_{t\neq\theta}G \leq C_p^\psi(t) - C_p(t_0) \leq \frac{t - t_0 + u_p^-}{1 - \rho_p^-}. \tag{85}$$

*for $\Upsilon < 0$.*

**Proof:** By a trivial minorization of (80) and by using $(T_e' - T_s')\psi = (T_e' - T_s)\psi \leq AG\psi = \Upsilon > 0$ in (81), relation (84) follows immediately. Applying majorization of (81) in case of $\Upsilon < 0$, the upper bound in (85) is also easily established. For the lower bound, we distinguish two cases for (80): For $t \geq t_e$, we have $\psi'' = 0$ so that (85) follows immediately. For $t < t_e$, we have $\psi'' = \psi < 0$ but also $T_e' - T_s \leq (A-1)G$, so that

$$\begin{aligned} (T_e' - T_s)\psi - I_{t\neq\theta}(1 - \psi'')G &= \left[(A-1)G\psi - \Upsilon\right] + \Upsilon - I_{t\neq\theta}G + I_{t\neq\theta}G\psi \\ &= \Upsilon - I_{t\neq\theta}G - (1 - I_{t\neq\theta})G\psi \geq \Upsilon - I_{t\neq\theta}G. \end{aligned}$$

This confirms (85) also for this case, completing the proof of the lemma. □

Now we are ready for our final theorem. It establishes that continuous amortization does not impair worst cast precision and accuracy provided that $|\psi| = |\Upsilon|/(AG)$ is chosen large enough; note that the other results provided by Theorem 1 are not affected by continuous amortization. Therefore, our theorem imposes an upper bound on the length of the amortization period $A$. Note carefully that this makes it impossible to conceal the impairment of clock rate from applications completely.

THEOREM 2 (PRECISION REF. ACCURACY CONTINUOUS AMORTIZATION) *Given the clock synchronization algorithm of Definition 7 employing the continuous amortization algorithm of Definition 12 for setting the local interval clock in step (T), with*

*[1] amortization rate deviation $\psi_p$ of any non-faulty node p's clock satisfying $1 > |\psi_p| \geq \rho_{max}$,*

*[2] $P \geq \Lambda + \Omega + \Delta + \Gamma_{\max} + A_{\max}G$ with*

$$A_{\max} \geq \left\lceil \frac{\max\{\pi^-, \pi^+\}}{G|\psi_{\min}|} \right\rceil, \tag{86}$$

*where $|\psi_{\min}| = \min_p |\psi_p|$ and $\pi = [-\pi^-, \pi^+]$ is the maximum adjustment applied to a non-faulty clock given by (72),*

*we obtain*

*(1) maximum precision $\pi^{\psi}_{\max}$ given by*

$$\pi^{\psi}_{\max} = \pi_{\max} + \frac{\psi_{\max} u_{\max}}{(1 - \bar{\rho}_{\max})(1 - \psi_{\max})} + \mathcal{O}(G\rho_{\max}), \tag{87}$$

*for $\psi_{\max} = \max_p \psi_p$, where $\pi_{\max}$ defined in (71) is the maximum precision of the instantaneously corrected variant of the algorithm. The additional terms vanish in the remainder if $\psi_{\max} = \mathcal{O}(\rho_{\max})$.*

*(2) The accuracy interval $A^{R,k+1}_q = [T^{R,k+1}_q \pm \alpha^{R,k+1}_q]$ provided by the local interval clock of a non-faulty node q at the end of (and during) round k + 1, $k \geq 0$, satisfies*

$$\alpha^{R,k+1}_q \subseteq \beta^{k+1}_q + P\rho_q + u_q + G + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})\rho_{\max},$$

*where $\beta^{k+1}_q$ is given by (63). Note that this is the same bound as obtained for the instantaneously corrected variant of the algorithm.*

**Proof:** It should be clear that continuous amortization can only impair precision if instantaneous correction would bring the clocks of two non-faulty nodes $p$ and $q$ closer together, i.e., if the (instantaneous) correction value $\Upsilon$ applied to the larger clock is negative or the one applied to the smaller clock is positive: Since continuous amortization delays clock correction, improving the clocks' precision is also delayed in this case. On the other hand, if clock correction happens to occur in the opposite direction, continuous amortization can at most improve precision w.r.t. instantaneous correction. After all, $\pi_{\max}$ is the maximum of both the precision achieved when all clocks are still in round $k$ (defining maximum precision in the former case) and the "mixed situation" (defining maximum precision in the latter case), where some clocks are already in round $k+1$. In any case, when continuous amortization has terminated, the resulting clock values are indistinguishable from the ones obtained by instantaneous correction according to Lemma 14.

Therefore, it suffices to investigate the growth of the difference of the amortizing clocks $C_f^{\psi_f}(t)$, $C_l^{\psi_l}(t)$ of two non-faulty nodes $f$ and $l$ for $t \geq t_f^R$, where $t_f^R \leq t_l^R$ denote the respective real-times of resynchronization (i.e., when switching from the virtual clocks of round $k$ to the ones of round $k+1$ —and hence beginning of amortization— takes place). More specifically, defining the clock differences $D(t) = C_f^{\psi_f}(t) - C_l(t)$ and $D'(t) = C_f^{\psi_f}(t) - C_l^{\psi_l}(t)$ with the abbreviations $C_l(t) = C_l^k(t)$ and $C_f(t) = C_f^k(t)$, we can restrict our attention to the case where the correction values $\Upsilon_f = C_f^{k+1}(t_f^R) - C_f^k(t_f^R)$ and $\Upsilon_l = C_l^{k+1}(t_l^R) - C_l^k(t_l^R)$ satisfy

$$\operatorname{sgn}\Upsilon_f = -\operatorname{sgn}D(t_f^R), \tag{88}$$

$$\operatorname{sgn}\Upsilon_l = \operatorname{sgn}D'(t_l^R); \tag{89}$$

note that $C_f^{\psi_f}(t_f^R) = C_f(t_f^R)$ and $C_l^{\psi_l}(t_l^R) = C_l(t_l^R)$.

Given the way we established $\pi_{\max}$ in Theorem 1 in conjunction with the reasoning above, we can deduce that overall precision is not impaired if $D(t) \cdot \operatorname{sgn}D(t_f^R)$ is monotonically decreasing for $t \geq t_f^R$ (and analogous w.r.t. $D'(t)$ for $t \geq t_l^R$). Note that there is no need to consider the case where a clock difference changes sign and grows in the opposite direction; Lemma 14 guarantees that this cannot go beyond the difference of the new (corrected) virtual clocks and hence remains within $\pi_{\max}$.

Now we will establish an expression for the worst case difference of amortizing clocks. First of all, setting $t_0 \geq t_s$ and $t < t_e$ in Lemma 13 implies $T_e' = T$, $T_s \leq T_s' = T_0$, and $\psi' = \psi'' = \psi$, so that (80) may be rearranged to

$$(T - T_0)(1-\psi) \geq \frac{t - t_0 - u_p^+}{1+\rho_p^+} - I_{t\neq\theta}(1-\psi)G.$$

Combining this with a similar rearrangement of (81) easily provides

$$\frac{t - t_0 - u_p^+}{(1+\rho_p^+)(1-\psi)} - I_{t\neq\theta}G \leq C_p^\psi(t) - C_p^\psi(t_0) \leq \frac{t - t_0 + u_p^-}{(1-\rho_p^-)(1-\psi)} + I_{t_0\neq\theta_0}G. \tag{90}$$

Note that this relation is of course the one of (11) in case of $\psi = 0$. Subtracting (90) for $C_f^{\psi_f}(t)$ from the same expression for $C_l^{\psi_l}(t)$ and applying straightforward majorizations yields the upper bound

$$\begin{aligned} C_f^{\psi_f}(t) - C_l^{\psi_l}(t) \;\leq\; & C_f^{\psi_f}(t_0) - C_l^{\psi_l}(t_0) + 2G \\ & + \frac{u_f^-}{(1-\rho_f^-)(1-\psi_f)} + \frac{u_l^+}{(1+\rho_l^+)(1-\psi_l)} \end{aligned}$$

$$
\begin{aligned}
&\quad + (t - t_0)\left[\frac{1}{(1-\rho_f^-)(1-\psi_f)} - \frac{1}{(1+\rho_l^+)(1-\psi_l)}\right] \\
&= C_f^{\psi_f}(t_0) - C_l^{\psi_l}(t_0) + 2G + u_f^- + u_l^+ \qquad (91)\\
&\quad + \frac{u_f^- \psi_f}{(1-\rho_f^-)(1-\psi_f)} + \frac{u_l^+ \psi_l}{(1+\rho_l^+)(1-\psi_l)} + \mathcal{O}(G\rho_{\max}) \\
&\quad + (t - t_0)\left[\frac{\rho_l^+ + \rho_f^- + \psi_f - \psi_l - \rho_f^- \psi_f - \rho_l^+ \psi_l}{(1-\rho_f^-)(1-\psi_f)(1+\rho_l^+)(1-\psi_l)}\right],
\end{aligned}
$$

where we used the simple identity

$$
\frac{u}{(1-\rho)(1-\psi)} = u + \frac{\rho u}{1-\rho} + \frac{\psi u}{(1-\rho)(1-\psi)}
$$

in conjunction with $u_{\max} = \mathcal{O}(G)$ according to Assumption 2. Similarly, we obtain the lower bound

$$
\begin{aligned}
C_f^{\psi_f}(t) - C_l^{\psi_l}(t) &\geq C_f^{\psi_f}(t_0) - C_l^{\psi_l}(t_0) - 2G \\
&\quad - \frac{u_f^+}{(1+\rho_f^+)(1-\psi_f)} - \frac{u_l^-}{(1-\rho_l^-)(1-\psi_l)} \\
&\quad + (t - t_0)\left[\frac{1}{(1+\rho_f^+)(1-\psi_f)} - \frac{1}{(1-\rho_l^-)(1-\psi_l)}\right] \\
&= C_f^{\psi_f}(t_0) - C_l^{\psi_l}(t_0) - 2G - u_f^+ - u_l^- \qquad (92)\\
&\quad - \frac{u_f^+ \psi_f}{(1+\rho_f^+)(1-\psi_f)} - \frac{u_l^- \psi_l}{(1-\rho_l^-)(1-\psi_l)} + \mathcal{O}(G\rho_{\max}) \\
&\quad + (t - t_0)\left[\frac{-\rho_l^- - \rho_f^+ - \psi_l + \psi_f + \rho_f^+ \psi_f + \rho_l^- \psi_l}{(1+\rho_f^+)(1-\psi_f)(1-\rho_l^-)(1-\psi_l)}\right].
\end{aligned}
$$

For $D(t)$, we have to set $t_0 = t_f^R$ and $\psi_l = 0$ in the bounds above. Now, the terms in the first line in both the upper bound (91) and the lower bound (92) are covered by $\pi_{\max}$ according to our earlier expositions; $2G$ and also the rate adjustment uncertainities are of course included. Hence, we just have to ensure that the bracketed term in (91) and (92) is $\leq 0$ and $\geq 0$, respectively. Recalling (88) and $\operatorname{sgn}\Upsilon_f = \operatorname{sgn}\psi_f$, it is apparent that we only have to consider $\psi_f < 0$ for the upper and $\psi_f > 0$ for the lower bound, respectively, since e.g. the upper bound is only meaningful if $D(t)$ is positive. Hence it follows that $\psi_f \leq -\rho_l^+ - \rho_f^-$ resp. $\psi_f \geq \rho_l^- + \rho_f^+$; choosing $|\psi_f| \geq \rho_{\max}$ guarantees both conditions to hold. For $t_f^R \leq t < t_l^R$, we thus obtain

$$
|D(t)| \leq \pi_{\max} + \frac{u_{\max}\psi_{\max}}{(1-\rho_{\max}^-)(1-\psi_{\max})} + \mathcal{O}(G\rho_{\max}).
$$

Similarly, for $D'(t)$, we have $t_0 = t_l^R$ and $\psi_l \neq 0$ (irrespectively of $\psi_f$, which can be taken out of the brackets and included in $\pi_{\max}$ if it has the "wrong" sign). By virtue of (89), we only have to consider $\psi_l > 0$ for the upper and $\psi_l < 0$ for the lower bound, respectively. The same reasoning as above shows that $|\psi_l| \geq \rho_{\max}$ is sufficient here as well, which finally justifies the lower bound given in precondition [1] of our theorem; note that the upper bound $|\psi_p| < 1$ given there comes from $1 - \psi_p$ arising in the denominator in the bounds involved. Similar as above, we easily obtain

$$|D'(t)| \leq \pi_{\max} + \frac{u_{\max}\psi_{\max}}{(1-\bar{\rho}_{\max})(1-\psi_{\max})} + \mathcal{O}(G\rho_{\max}),$$

which is valid for all $t \geq t_l^R$ within the amortization periods of interest. This eventually confirms (87).

Finally, recalling the bounds $\pi$ on the maximum clock adjustment $\Upsilon$ applied to a non-faulty clock from item (3) of Theorem 1 in conjunction with the relation $\psi = \Upsilon/(AG)$, the condition on $A_{\max}$ in precondition [1] follows; note that $\Upsilon$ is actually an integer multiple of $G$. This completes the proof of the precision part of our theorem.

For the accuracy part, we assumed in the definition of the continuous amortization algorithm that the envelopes of the accuracy interval are not affected. Hence, total length of the accuracy interval $\alpha_q^{k+1}$ present at the beginning of round $k+1$ is the same as for the instantaneous correction case. However, since the reference point is not set simultaneously with the edges of the interval, it is not possible to give useful bounds on positive resp. negative accuracy. However, when continuous amortization has finished, the situation of instantaneous correction is reestablished, according to Lemma 14, so that the worst case accuracy interval at the end of round $k+1$ is the same for both continuous amortization and instantaneous correction. Now, a round (as measured at node $q$'s clock) has length $L \in P + \pi$ according to (54), hence $L = P + \mathcal{O}(P\rho_{\max} + G + \varepsilon_{\max})$ by virtue of the order term established for $\pi = |\pi|$ in the proof of Lemma 12. Applying (75) established in the proof of Theorem 1 easily provides (2), completing the proof of Theorem 2. □

## 6. Conclusions and Further Research

We introduced and rigorously analyzed a simple interval-based algorithm suitable for external clock synchronization. Unlike usual internal synchronization approaches, our convergence function-based algorithm (dynamically) maintains both precision and accuracy w.r.t. an external time standard like UTC. To that end, each node keeps track of time by means of a local interval clock $C(t)$, which is made up of an interval of accuracies taken relatively to the ordinary local clock's value. Our algorithm periodically exchanges local interval clock readings among all nodes in the system and employs an interval-valued convergence function to obtain and subsequently apply a clock correction value that enforces precision and accuracy. Clock correction can be done either instantaneously or by means of a certain continuous amortization algorithm.

The comprehensive analysis presented in the previous sections is generic w.r.t. the particular convergence function and provides results that are immediately applicable in practice. It relies on a system model that incorporates many aspects usually abstracted away, like non-zero clock granularity, rate-adjustment uncertainities, and broadcast latencies. Technically, the analysis is based upon a novel, interval-based framework for providing worst case bounds for various parameters in terms of the characteristic functions of the convergence function. It manifests a striking conceptual beauty and expressive power, primarily by utilizing a suitable notion of internal global time. The results obtained include worst case bounds for initial and maximum precision, accuracy, maximum clock correction, and resynchronization tightness, for both instantaneous correction and linear continuous amortization. One of the surprising facts revealed by our analysis is the worse effect of rate adjustment uncertainities —arising when discrete rate adjustment techniques are employed— as well as clock granularity upon all parameters.

Current theoretical work is primarily devoted to the analysis of particular instances of our generic algorithm. Apart from the orthogonal accuracy algorithms dealt with in [19], we are also working on a promising generalization of the fault-tolerant midpoint algorithm of [8]. In addition, we are investigating interval-based algorithm(s) operating in conjunction with on-line measurement/control of characteristic system parameters, like rate deviation bounds or transmission delays. After all, interval-based algorithms have the inevitable shortcoming that they depend on certain system parameters explicitly. Most importantly, rate deviation bounds $\rho_p$, transmission delay bounds $\varepsilon_{pq}$ and appropriate maximum values $\rho_{\max}$, $\varepsilon_{\max}$ are required for the instance of the algorithm running at node $p$. However, this does not mean that such parameters can only be provided by compiling them statically into the algorithm — on-line measurement is in fact the most appealing alternative.

Future theoretical research will be primarily devoted to the problem of integrating our algorithm(s) in the clock validation framework, which raises issues ranging from incorporating accuracy intervals from primary nodes over managing the transition to/from UTC loss up to suitable failure detectors for certain (unbounded accuracy) faults. Other subjects of interest include a more general investigation of the problem of optimality w.r.t. precision/accuracy and the most challenging task of providing an average case analysis of precision and accuracy.

As far as practice is concerned, we are currently working on a fully engineered implementation (software + hardware) on top of our UTCSU-ASIC, which will be mainly used for experimental evaluation. Besides confirming theoretical results experimentally, it will help us not to have overlooked important practical issues. Moreover, demonstrating the suitability of our concepts is mandatory for a certain industrial pilot application that could be carried out by using our approach.

## References


1. M.H. Azadmanesh, R.M. Kieckhafer. *New Hybrid Fault Models for Asynchronous Approximate Agreement*, to appear in IEEE Transactions on Computers, 1996.
2. F. Cristian. *Probabilistic Clock Synchronization*, Distributed Computing, 3, 1989, p. 146–158.
3. P. H. Dana. *Global Positioning System (GPS) Time Dissemination for Real-Time Applications*, to appear in Real-Time Systems (this issue), 1996.
4. D. E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, Vol. 1, 2nd ed., Addison Wesley, Reading, Mass., 1973.
5. H. Kopetz, W. Ochsenreiter. *Clock Synchronization in Distributed Real-Time Systems*, IEEE Transactions on Computers, C-36(8), August 1987, p. 933–940.
6. L. Lamport. *Synchronizing Time Servers.* Technical Report Digital System Research Center 18, p. 1–33, 1987.
7. B. Liskov. *Practical uses of synchronized clocks in distributed systems*, Distributed Computing 6, 1993, p. 211–219.
8. J. Lundelius-Welch, N. Lynch. *A New Fault-Tolerant Algorithm for Clock Synchronization*, Information and Computation, 77(1), 1988, p. 1–36.
9. K. Marzullo. *Maintaining the Time in a Distributed System: An Example of a Loosely-Coupled Distributed Service.* Ph.D. thesis, Dept. of Electrical Engineering, Stanford University, 1984.
10. K. Marzullo. *Tolerating Failures of Continuous-Valued Sensors*, ACM Transactions on Computer Systems, 8(4), November 1990, p. 284–304.
11. K. Marzullo, S. Owicki. *Maintaining the Time in a Distributed System.* ACM Operating System Review 19(3), p. 44–54, 1983.
12. D. L. Mills. *Internet Time Synchronization: The Network Time Protocol*, IEEE Transactions on Communications, 39(10), October 1991, p. 1482–1493.
13. D. L. Mills. *Improved Algorithms for Synchronizing Computer Network Clocks*, IEEE Trans. Networks, June 1995, p. 245–254.
14. OSF. *Introduction to OSF DCE*, Prentice Hall, Englewood Cliffs, NJ, 1992.
15. F. B. Schneider. *A Paradigm for Reliable Clock Synchronization*, Proc. Advanced Seminar of Local Area Networks, Bandol, France, April 1986, p. 85–104.
16. F. B. Schneider. *Understanding Protocols for Byzantine Clock Synchronization*, Technical Report Dept. Computer Science, Cornell University 87-859, August 1987.
17. U. Schmid. *Synchronized UTC for Distributed Real-Time Systems*, Proc. IFAC Workshop on Real-Time Programming WRTP'94, Lake Reichenau/Germany, 1994, p. 101–107.
18. U. Schmid. *Synchronized Universal Time Coordinated for Distributed Real-Time Systems*, Control Engineering Practice, 3(6), 1995, p. 877–884. (Reprint of [17]).
19. U. Schmid. *Orthogonal Accuracy Clock Synchronization*, (manuscript), 1996.
20. U. Schmid, A. Pusterhofer. *SSCMP: The Sequenced Synchronized Clock Message Protocol*, Computer Networks and ISDN Systems, 27, 1995, p. 1615–1632.
21. F. Schmuck, F. Cristian. *Continuous Clock Amortization Need Not Affect the Precision of a Clock Synchronization Algorithm*, 9th ACM Symposium on Principles of Distributed Computing, 1990, p. 133–143.
22. K. Schossmaier, U. Schmid. *UTCSU Functional Specification*, Technical Report Dept. of Automation, Technical University of Vienna, no. 183/1-56, 1995.
23. K. Schossmaier, U. Schmid, M. Horauer, D. Loy. *Specification and Implementation of the Universal Time Coordinated Synchronization Unit (UTCSU)*, to appear in Real-Time Systems (this issue), 1996.
24. W. Schwabl. *Der Einfluß zufälliger und systematischer Fehler auf die Uhrensynchronisation in verteilten Echtzeitsystemen*, Dissertation an der Technisch-Naturwissenschaftlichen Fakultät, Technical University of Vienna, October 1988.
25. B. Simons, J. Lundelius-Welch, N. Lynch. *An Overview of Clock Synchronization*, in B. Simons, A. Spector (eds.): Fault-Tolerant Distributed Computing, Springer Lecture Notes on Computer Science 448, 1990, p. 84–96.

26. P.M. Thambidurai, Y.K. Park. *Interactive Consistency with Multiple Failure Modes*, Proc. 7th Reliable Distributed Systems Symposium, October 1988.
27. Z. Yang, T. A. Marsland. *Annotated Bibliography on Global States and Times in Distributed Systems*, ACM SIGOPS Operating Systems Review, June 1993, p. 55–72.

## Glossary

| Name | Meaning | Def. | Pg. |
|---|---|---|---|
| $A$ | length (in ticks) amortization period | Def. 12 | 48 |
| $A_{\max}$ | maximum length amortization periods | Thm. 2 | 52 |
| $\boldsymbol{A}$, $\boldsymbol{I}$ | (accuracy) intervals | Sec. 2 | 4 |
| $\overline{\boldsymbol{A}}$ | swapped interval $[r \mp \boldsymbol{\alpha}]$ | Equ. 2 | 5 |
| $\hat{\boldsymbol{A}}$ | $\pi$-precision interval associated with $\boldsymbol{A}$ | Def. 9 | 31 |
| $\aleph$ | accuracy enhancement function | Def. 11 | 39 |
| $\boldsymbol{\alpha} = [-\alpha^-, \alpha^+]$ | interval of accurracies of $\boldsymbol{I} = [r \pm \boldsymbol{\alpha}]$ | Asm. 2 | 4 |
| $\alpha = \alpha^- + \alpha^+$ | length of $\boldsymbol{\alpha}$ | Sec. 2 | 4 |
| $\boldsymbol{\alpha}_p(t) = [-\alpha_p^-(t), \alpha_p^+(t)]$ | interval of accurracies at node $p$ | Asm. 3 | 17 |
| $\boldsymbol{\alpha}_p^0 = [-\alpha_p^{0-}, \alpha_p^{0+}]$ | initial accurracies at node $p$ | Def. 7 | 25 |
| $B \in \{1, 2\}$ | indicator of pure broadcast network | Asm. 4 | 18 |
| $C(t), C_p(t)$ | ordinary clock (node $p$) | Sec. 3.1 | 10 |
| $C_p^{\psi}(t)$ | amortizing ordinary clock node $p$ | Lem. 13 | 50 |
| $c(T)$ | inverse clock | Sec. 3.1 | 10 |
| $\boldsymbol{C}_p(t) = [C_p(t) \pm \boldsymbol{\alpha}_p(t)]$ | local interval clock of node $p$ | Asm. 3 | 17 |
| $\mathcal{CV}$ | generic convergence function | Def. 11 | 39 |
| $\delta_{pq}$ | deterministic part trans. delay $p$ to $q$ | Asm. 4 | 18 |
| $\delta_{\max}, \delta_{\min}$ | uniform bounds det. part trans. delay | Asm. 4 | 18 |
| $\Delta$ | transmission delay compensation | Def. 7 | 25 |
| $\boldsymbol{\varepsilon}_{pq} = [-\varepsilon_{pq}^-, \varepsilon_{pq}^+]$ | transmission delay uncertainity $p$ to $q$ | Asm. 4 | 18 |
| $\varepsilon_{pq} = \varepsilon_{pq}^- + \varepsilon_{pq}^+$ | length of $\boldsymbol{\varepsilon}_{pq}$ | Asm. 4 | 18 |
| $\boldsymbol{\varepsilon}_{\max}$ | max. transmission delay uncertainity | Asm. 4 | 18 |
| $\varepsilon_{\max}$ | length of $\boldsymbol{\varepsilon}_{\max}$ | Asm. 4 | 18 |
| $g$ | real-time duration corresponding to $G$ | Asm. 2 | 12 |
| $G$ | clock granularity | Asm. 2 | 12 |
| $\boldsymbol{G} = [0, G]$ | positive granularity interval | Def. 5 | 16 |
| $\boldsymbol{G}\rho = [0, G(1 + \rho_{\max}^+)]$ | positive granularity interval | Def. 5 | 16 |
| $\overline{\boldsymbol{G}} = [-G, 0]$ | negative granularity interval | Def. 5 | 16 |
| $G_S$ | clock setting granularity | Asm. 2 | 12 |
| $G_A$ | accuracy transmission loss | Asm. 4 | 18 |
| $2\boldsymbol{G}_A = [-G_A, G_A]$ | accuracy transmission loss interval | Def. 6 | 20 |
| $\gamma_p$ | real-time exec. time bound for node $p$ | Asm. 1 | 9 |
| $\gamma_{\max}, \gamma_{\min}$ | uniform real-time exec. time bounds | Asm. 1 | 9 |
| $\Gamma_p$ | logical time exec. bound for node $p$ | Def. 7 | 25 |
| $\Gamma_{\max}, \Gamma_{\min}$ | uniform logical time exec. bounds | Def. 7 | 25 |

| Name | Meaning | Def. | Pg. |
|---|---|---|---|
| $I_{t \neq \theta}$ | indicator function of non-synchrony | Def. 4 | 14 |
| $I_q^p$ | interval of $p$ rec. at $q$ | Equ. 19 | 24 |
| $\mathcal{I}_p$ | ordered set received intervals at $p$ | Def. 7 | 25 |
| $\lambda_{\max}$ | real-time maximum broadcast latency | Asm. 4 | 18 |
| $\Lambda_{\max}$ | logical time max. broadcast latency | Def. 7 | 25 |
| $n$ | number of nodes | Sec. 3 | 9 |
| $\omega_{\max}$ | real-time max. broadcast oper. delay | Asm. 4 | 18 |
| $\Omega_{\max}$ | log. time max. broadcast oper. delay | Def. 7 | 25 |
| $P$ | round period | Def. 7 | 25 |
| $\Phi()$ | precision preservation function | Def. 11 | 39 |
| $\pi = [-\pi^-, \pi^+]$ | (generic) interval of precision | Def. 2 | 6 |
| $\pi = \pi^- + \pi^+$ | length of $\pi$ | Def. 2 | 6 |
| $\pi_0 = [-\pi_0^-, \pi_0^+]$ | ideal initial precision | Lem. 12 | 40 |
| $\pi^H = [-\pi^{H-}, \pi^{H+}]$ | uniform precision exchanged intervals | Lem. 11 | 32 |
| $\pi_I$ | uniform precision perceptions | Lem. 11 | 32 |
| $\pi^p$ | precision perceptions of $p$ | Lem. 11 | 32 |
| $\pi_q^p$ | precision interval $p$ received at $q$ | Lem. 12 | 40 |
| $\pi$-accurate | correct w.r.t. $t$ and $\pi$-precise | Def. 2 | 6 |
| $\pi$-center | asymmetric reference point setting | Sec. 2 | 4 |
| $\pi$-correct | correct w.r.t. both $t$ and $\tau$ | Def. 9 | 31 |
| $\pi$-precise | precise interval set | Def. 2 | 6 |
| $\Pi()$ | precision enhancement function | Def. 11 | 39 |
| $\psi$ | inverse amortization rate deviation | Def. 12 | 48 |
| $[r \pm \alpha]$ | interval with reference point | Equ. 1 | 4 |
| ref($\boldsymbol{I}$) | reference point of interval $\boldsymbol{I}$ | Sec. 2 | 4 |
| $\boldsymbol{\rho}_p = [-\rho_p^-, \rho_p^+]$ | inverse rate deviation bound node $p$ | Asm. 2 | 12 |
| $\rho_p = \rho_p^- + \rho_p^+$ | length of $\boldsymbol{\rho}_p$ | Asm. 2 | 12 |
| $\boldsymbol{\rho}_{\max} = [-\rho_{\max}^-, \rho_{\max}^+]$ | uniform inverse rate deviation bound | Asm. 2 | 12 |
| $\rho_{\max} = \rho_{\max}^- + \rho_{\max}^+$ | length of $\boldsymbol{\rho}_{\max}$ | Asm. 2 | 12 |
| $t$ | real-time value | Sec. 3.1 | 10 |
| $T$ | logical time value | Sec. 3.1 | 10 |
| $T_p$, $t_p$ | log., real-time beginning round at $p$ | Lem. 11 | 32 |
| $T_{pq}^A, t_{pq}^A$ | log., real-time transmission $p$ to $q$ | Def. 7 | 25 |
| $T_p^R, t_p^R$ | log., real-time resynchronization at $p$ | Def. 7 | 25 |
| $T_q^p, t_q^p$ | log., real-time reception of $p$ at $q$ | Def. 7 | 25 |
| $\tau$, $\tau^k$ | internal global time (of round $k$) | Def. 9 | 31 |
| $\theta$ | real-time when clock tick occurs | Def. 4 | 14 |
| $\Theta$ | logical time when clock tick occurs | Def. 4 | 14 |
| $\boldsymbol{u}_p = [-u_p^-, u_p^+]$ | rate correction uncertainity node $p$ | Asm. 2 | 12 |
| $\boldsymbol{u}_{\max} = [-u_{\max}^-, u_{\max}^+]$ | uniform rate correction uncertainity | Asm. 2 | 12 |
| $\Upsilon_q$ | max. correction non-faulty clock $q$ | Lem. 12 | 40 |