A precise clock synchronization protocol

Citation for published version (APA):

Document status and date:
Published: 01/01/1993

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
A precise clock synchronization protocol

by

E.J. Luit and J.M.M. Martin

93/45
This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science Eindhoven University of Technology. Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review. Copies of these notes are available from the author.

Copies can be ordered from:
Mrs. M. Philips
Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN
The Netherlands
ISSN 0926-4515

All rights reserved
editors: prof.dr.M.Rem
prof.dr.K.M.van Hee.
A precise clock synchronization protocol

E.J. Luit and J.M.M. Martin

Department of Mathematics and Computing Science,
Eindhoven University of Technology, Eindhoven, The Netherlands

Abstract

A distributed clock synchronization protocol is presented which achieves a very high precision without the need for very frequent resynchronizations. The protocol tolerates failures of the clocks: clocks may be too slow or too fast, exhibit omission failures and report inconsistent values. Synchronization takes place in synchronization rounds as in many other synchronization protocols. At the end of each round, clock times are exchanged between the clocks. Each clock applies a convergence function (CF) to the values obtained. This function estimates the difference between its clock and an average clock and corrects its clock accordingly. Clocks are corrected for drift relative to this average clock during the next synchronization round. The protocol is based on the assumption that clock reading errors are small with respect to the required precision of synchronization.

It is shown that the CF resynchronizes the clocks with high precision even when relatively large clock drifts are possible. The CF of the algorithm exploits the properties of a matrix that contains the differences between all clocks in the system. All correct processors have access to identical copies of this matrix which is disseminated by an underlying reliable message protocol. It is shown that the drift-corrected clocks remain synchronized until the end of the next synchronization round. The maximum length of a synchronization round mainly depends on the precision with which the clocks are resynchronized and on the maximum magnitude of the second time derivative of the function that describes a clock. The stability of the protocol is proven, i.e., it is shown that drift of the clocks with respect to physical time is bounded.

Keywords

Real-time Distributed Systems, Fault Tolerance, Physical Clock Synchronization
1 Introduction

Physical clock synchronization is essential in distributed real-time systems. The physical process controlled by the system may impose timing relations between tasks that execute on different processors. For example, in a copying machine, there are timing relations between the separation of a blank sheet, the separation of an original and the exposure of an original. When the clocks are not synchronized precisely, it is not possible to implement strict timing relations. Also, clock synchronization provides a global time base that allows agreement on the ordering of events in the system. This is important for other system components such as a reliable communication service.

The clock synchronization (CS) protocol presented achieves a very precise synchronization of the clocks without the need for frequent resynchronizations. Precision is defined here as the maximum difference between the values of any pair of correct clocks at any time. The protocol was developed to achieve a precision of 100 µs given a worst-case clock drift rate (or drift, for short) of $10^{-4}$ and a clock reading error of approximately 2 µs. With these parameters, most published protocols, e.g., [LaMe85], [MaSch85] and [SrTo87], require very frequent resynchronizations. Cristian [Cris89] describes a protocol that avoids these frequent resynchronizations because the clocks are corrected for local drift. However, he only describes a master-slave protocol. This paper describes a distributed protocol that also applies drift corrections.

Drift corrections are evidently not useful if the relative drifts of the clocks cannot be estimated rather precisely. Local drift must be estimated with respect to some suitably defined average clock. When a clock is resynchronized, an algorithm called the Convergence Function (CF) [Schn86] calculates a correction for the clock. This correction can be interpreted as the best estimate of the difference between this clock and the average clock. The correction applied to synchronize a clock is thus also a measure of its drift relative to the average clock. Therefore, drift corrections can only be applied if the maximum difference between any two clocks immediately after a resynchronization is small enough. This maximum difference is called the resynchronization precision here. The resynchronization precision is only guaranteed directly after a resynchronization, in contrast to the precision which is guaranteed at all times. When drift corrections are not used, the interval between resynchronizations equals half the quotient of the difference between precision and resynchronization precision and the worst-case drift rate.

Drift corrections are only useful when the ratio of the resynchronization precision and the precision is smaller than $\frac{1}{3}$. This is discussed in more detail in section 5. Existing software synchronization protocols do not resynchronize the clocks with sufficient precision. For example, the CFs described in [MaSch85] achieve resynchronization precision to precision ratios of $\frac{3}{4}$ and $\frac{1}{3}$. Of the CFs described in [LaMe85], only algorithm CSM achieves a ratio that is just below $\frac{1}{3}$ with the parameters given above. However, when details such as clock setting errors are taken into account, the ratio increases above $\frac{1}{3}$. The resynchronization precision of the protocol described in [SrTo87] equals the time needed to 1) prepare a message, 2) send it to all other processors and 3) process this message. Therefore, this
protocol can not be used either.

In this paper, a hardware clock is described by two mathematical clocks. One of these mathematical clocks represents the uncorrected hardware clock and the second one represents the drift-corrected hardware clock. Only this second clock is synchronized within the precision. Clock differences and drift corrections are estimated from the uncorrected clock, while applications only have access to the drift-corrected clock. A clock reading is described as an estimate of the mathematical clock with a certain maximum reading error.

The resynchronization precision of the protocol described in this paper depends mainly on the size of the errors in the readings of both local and remote clocks. If these reading errors are much smaller than the precision required, the clocks can be resynchronized very precisely. The reading error of local clocks is small if the hardware clocks can be read out in small units. A remote clock can only be read with a small error if uncertainties in message transmission delays are small. Almost constant transmission delays can be realized with Cristian's probabilistic synchronization [Cris89]. Almost constant transmission delays are also possible in a statically scheduled system such as the DEpendable Distributed Operating System (DEDOS) [SRLH91], [VLHJ91], [LuMo92].

In most published protocols, the CF calculates a new clock time from some fault-tolerant average of all clock values exchanged instead. The CF described in this paper examines the consistency of the clock differences exchanged. If exchanged values are not consistent, pairs or triples of clocks are removed from a set of clocks that defines average clock time. This procedure is repeated until a consistent subset of clocks remains. The procedure removes at least one incorrect clock with each pair or triple removed. Correct clocks are also removed with this procedure. It is shown that the consistent subset of clocks can never be empty. The procedure imposes a strong bound on the influence of incorrect clocks on the new clock times.

It is necessary to show that the protocol is stable, i.e., the rate at which the clocks drift away from physical time must be bounded. This property is called accuracy in this paper. When an external source of physical time is not available, proof of the accuracy can only be based on the maximum drift specified by the clock manufacturer. Therefore, the resynchronization is based on true hardware clock differences and not on the mathematical drift-corrected clock. Notice that these true hardware clock differences can be very large compared to the synchronization precision.

Since overheads increase rapidly with the number of clocks in the system, a hierarchical organization of the protocol is foreseen. In a hierarchically organized protocol, the synchronization between components in the same group is better than between the different groups. This paper only describes the synchronization within one group.

The effectiveness of drift corrections was investigated in an experiment with rapidly changing drifts that was conducted at the Eindhoven University of Technology (TUE). A simple master-slave configuration was synchronized with a probabilistic protocol and drift corrections were applied to the slave. The experiment showed that even when the drift changed rapidly, a high precision was obtained at low cost. In the experiment, the master was kept at a constant temperature while the cooling air of the slave was blocked until thermal equilibrium was reached. Then, the cooling air was restored. The behaviour
of the relative drift between the clocks during the experiment is shown in fig. 1. Notice that the relative drift changed by almost two orders of magnitude during the experiment. Nevertheless, the difference between the drift-corrected clock of the slave and the clock of the master was never larger than 660 $\mu$s even though the slave was resynchronized only once every two minutes. The clock reading error was approximately 6.2 $\mu$s.

2 Informal description of the protocol

The protocol periodically resynchronizes the clocks, like many published synchronization protocols [LaMe85], [MaSch85], [Schn86], [SrTo87], [Cris89], [OISh91], [VeRo92]. It is assumed that incorrect clocks may be present in the system. These clocks may run too slow or too fast, exhibit omission failures or report inconsistent values. Initially, correct clocks are assumed to be synchronized within the precision. This must be accomplished by a different protocol. For example, the time can be set on one of the clocks to which all other clocks synchronize. Initial estimates of the drift are assumed to be available as well.

The synchronization round of length $T_S$ is divided into three phases (fig. 2). The first phase of synchronization round $n$ starts around physical time $T_{E-1}^n$ when the clocks have just been resynchronized. During this phase, only the drift corrections calculated for round $n - 1$ are applied to the clocks. The differences between all clocks are measured during the second phase around time $T_M^n$. This phase is assumed to be short. During the last phase, the CF calculates the corrections required to resynchronize the system. Also, the
drift corrections that must be applied during the next round are calculated. However, the main reason to distinguish this phase is to allow a hierarchical organization of the CS protocol. The protocol at the higher levels can be executed during this last phase. A hierarchical organization of the protocol is necessary when large systems must be synchronized. Otherwise the overheads become intolerable because a large number of messages must be exchanged to tolerate incorrect clocks.

During the first phase, single tick drift corrections are applied to the hardware clock. Corrections are made a single tick at the time because the precision depends linearly on the size of the corrections (see section 5). The times at which the corrections must be applied are estimated with a version of Bresenham’s line discretization algorithm [Bres65]. This algorithm guarantees that the corrections are always within one clock tick of the intended correction.

At the end of the first phase, the measurement period starts around time $T_M$. Each clock sends its clock value to all other clocks. When a clock value is received, the receiver immediately reads its local clock and determines the difference between the two clock values. This difference is corrected for the communication delay incurred. The length of the measurement period is assumed to be small enough to ignore drift of the clocks during this period. Therefore, clock differences can be treated as if the clocks were read out at the same physical time.

During the last period of length $T_S$, each clock first disseminates the clock differences determined during the measurement phase by a reliable message protocol. Therefore, correct clocks have access to identical copies of a matrix $\Delta t_{ij}$ of differences between all clocks $i$ and $j$. Then, the difference between each clock and an average clock at time $T_M$ is estimated from this matrix. Drift corrections continue to be applied during the last phase. At the end of this phase, around time $T_E$, the clocks are resynchronized. However, since the clock differences were determined at time $T_M$, the latter point constitutes a virtual synchronization point for the following round. Therefore, not only the correction calculated by the CF for time $T_M$ is applied to the clocks. Also, the drift corrections that were applied during the last phase are subtracted from each clock and the newly calculated drift corrections for this period are added. After time $T_E$, the clocks then seem to have been synchronized at time $T_M$. The precision must thus be proven for the time intervals $[T_{M-1}, T_E]$ and not only for the interval $[T_{E-1}, T_E]$. The mathematical clocks that describe the hardware clocks must likewise be defined for this same interval as shown in fig. 2. Each physical clock is described by two mathematical clocks. One clock $c_i^n$ describes the uncorrected hardware clock $i$ during round $n$. A second clock $c_i^n$ describes the drift-corrected clock.

Figure 2: Phases of the synchronization protocol and validity of mathematical clocks
Both local and remote clock reading errors are assumed to be small. A remote clock reading error includes both the local clock reading error and the uncertainty in the transmission delay. Almost constant transmission delays can be obtained with probabilistic CS because round-trip messages are exchanged until the delays incurred are close enough to the minimum possible delay. Almost constant delays are also possible in a statically scheduled system because receivers know when synchronization messages are expected. Since the network is reserved for these communications, synchronization messages do not suffer unexpected delays on the network. A receiver waits for the arrival of a synchronization message during a period that is long enough to receive a message from any correct clock. A synchronization message that arrives outside this interval is treated as an omission failure.

Readings of correct clocks are described as estimates of a continuous clock with a certain maximum reading error. This model is used for all quantities of interest. For example, it was found to be very convenient to model an incorrect clock in the same manner. Likewise, when a clock calculates its difference with an average clock, this is described as an estimate of this average clock. The resynchronization precision is then essentially twice the maximum error in the estimate.

Since measurement errors are assumed to be small, other errors such as clock setting errors may become important. Therefore, these errors are modelled and some implementation details are included. The results show that these errors can not be entirely neglected although rough estimates of the precision can be obtained without considering these errors. However, it is important to model some errors. For example, the linear dependence of the precision on the units in which drift corrections are applied shows that these units must be small.

The next section enumerates the assumptions made and the notation is defined. In section 4, the resynchronization precision is proven. Proof of the precision and accuracy are given in section 5 and 6 respectively. Finally, in section 7, the results are discussed, conclusions are drawn and future work is outlined.

3 Assumptions and notation

The system consists of $N$ processors each of which has its own clock. The first assumption is the fault hypothesis that limits the number of incorrect clocks in the system.

Assumption 1. In a system of $N$ clocks, at most $k$ clocks are incorrect during round $n$ with $N \geq 3k + 1$

A clock that is correct during round $n$ is defined as follows:

Definition 1 A clock is called correct during round $n$ 1) if it is correctly synchronized at the beginning of round $n - 1$, 2) if its drift satisfies the assumptions 5 and 6 given below during rounds $n$ and $n - 1$, 3) if it sends all messages required by the protocol and 4) if
it does not report or calculate incorrect or inconsistent values. A clock is called incorrect otherwise.

No distinction is made between a clock and the processor on which it resides. Therefore, a failure of a clock may actually be a failure of its processor.

Hardware clocks can only be read in discrete units. However, it is more convenient to describe a clock as a continuous function.

**Definition 2** A correct clock \( i \) that is valid in round \( n \) is approximated by a continuous clock function \( c_i^n(T) \) which maps physical time \( T \) to local time during round \( n \). This clock describes the true hardware clock without drift corrections. A second clock \( c_i^n(T) \) describes the drift-corrected clock \( i \).

The clocks presented to the system are the drift-corrected clocks \( c_i^n(T) \). Only these clocks are synchronized with high precision. However, resynchronizations must be based on readings of the uncorrected clocks \( c_i^n(T) \). The clock readings that are exchanged during the measurement phase are therefore always estimates of the true hardware clock. Drift corrections that were applied to a hardware clock during period \( n \) are undone before a value is transmitted.

Because static scheduling or probabilistic CS is assumed, the following assumption is made about the transmission delay.

**Assumption 2** The transmission delay is equal to a known constant up to a small error.

It is only important that the transmission delay is constant when clock differences are established. Therefore, the transmission delay is defined as the time between the readings of the two clocks whose difference is established. Notice that the transmission delay needs to be constant but it needs not be the smallest possible delay. It may be necessary to increase the delay in order to obtain a constant delay. However, since the protocol is executed infrequently, the resulting inefficiency should be insignificant.

The following assumption is made for simplicity:

**Assumption 3** Each pair of clocks can communicate directly with each other.

The network that connects the clocks can thus have, e.g., a fully connected or a bus topology. Therefore, a synchronization message can not be delayed by a system component other than the sender or the receiver.

**Definition 3** A reading of a clock \( i \) in round \( n \) is described as an estimate \( \text{est}(c_i^n(T)) \) of the clock function \( c_i^n \) at physical time \( T \) with a reading error of at most \( \text{err}(\text{est}(c_i^n)) \). An estimate thus satisfies

\[
|\text{est}(c_i^n(T)) - c_i^n(T)| \leq \text{err}(\text{est}(c_i^n))
\]
Assumption 4 A correct clock $i$ satisfies $\text{err}(\text{est}(c_i^n)) = \varepsilon$

The error $\varepsilon$ includes both the actual reading error and remaining uncertainties in the transmission delay. For simplicity, the same error describes local and remote clock readings. This implies that the results obtained could be improved slightly.

Since any set of values can be represented by an estimate and a reading error, an incorrect clock can be described by an estimate and a reading error as well. However, the reading error of an incorrect clock can be arbitrarily large.

Definition 4 A clock $i$ that distributes inconsistent values is described by a clock function $c_i^n(T)$ that is read out with a reading error $\text{err}(\text{est}(c_i^n)) = \zeta > \varepsilon$.

For convenience, a clock function $c_i^n(T)$ that describes an incorrect clock may be assumed to satisfy all assumptions made for correct clocks apart from assumption 4. In particular, it may be assumed that it satisfies the precision. This last assumption leads to a larger value of $\zeta$ because the smallest value of $\zeta$ is obtained when the clock readings are described by an average and a maximum deviation from this average. Actually, it is not necessary to make the assumption about the precision. The conclusions of the next section are unchanged if it is only assumed that an incorrect clock has a drift smaller than the maximum drift assumed. The CF described in the next section bounds the value of $\zeta$ of incorrect clocks that remain among the set of clocks which defines the average clock.

Definition 5 The absolute drift $\rho_i(T)$ of a correct clock $i$ is the rate at which the clock drifts away from physical time. Therefore, the clock defined during round $n$ for $T \geq T_M^{n-1}$ is related to the absolute drift by

$$c_i^n(T) = c_i^n(T_M^{n-1}) + \int_{T_M^{n-1}}^{T}(1 + \rho_i(T))\,dT$$

where $T_M^{n-1}$ is the virtual synchronization point of round $n$ and $c_i^n(T_M^{n-1})$ the value to which the clock was set at time $T_M^{n-1}$.

A drift-corrected clock can be defined analogously, with the drift $\rho_i(T)$ replaced by an apparent drift $\rho_i'(T)$.

Proof of the precision obviously requires that the drift of the hardware clocks is bounded.

Assumption 5 The drift $\rho_i(T)$ of a correct clock $i$ is a differentiable function that satisfies

$$|\rho_i(T)| < \rho_{\text{max}}$$

where $\rho_{\text{max}}$ is the maximum drift specified by the manufacturer.
Drift corrections can only be based on estimates of the drift observed during previous rounds. However, the drift may vary with time due to changes of the environment temperature. Therefore, a bound must be assumed on the derivative of the drift in order to prove the precision.

Assumption 6 The first derivative of the drift of a correct clock $i$ is bounded:

$$\left| \frac{dp_i(T)}{dT} \right| < R_{\text{max}}$$

A bound on $R_{\text{max}}$ is derived in section 5 which depends mainly on the precision, the clock reading error and the length $T_S$ of the synchronization round. When variations of the drift due to changes in the environment temperature are small, long synchronization rounds can be realized.

Definition 6 The absolute time at which clock $i$ reaches the $n^{th}$ measurement time is denoted by $T^n_{M,i}$.

Definition 7 The difference between the clock reading reported by clock $i$ to clock $j$ and the reading by clock $j$ on receipt of $i$'s value is denoted by $\Delta t_{ij}$.

Since the difference is corrected for the transmission delay and because assumption 3 is made,

$$\Delta t_{ij} = \text{est}(c^n_i(T^n_{M,i})) - \text{est}(c^n_j(T^n_{M,i}))$$

(3.1)

Notice that the estimates exchanged are the clock readings without drift correction, i.e. the value of $\Delta t_{ij}$ is determined by the uncorrected hardware clock differences.

Assumption 7 Clock values and clock differences are distributed to all correct clocks by a reliable message protocol that does not require that all clocks are synchronized at all times.

Since this assumption is made, all correct clocks have access to identical copies of an $N$ by $N$ matrix that contains the time differences between all clocks in the system. An example of a protocol that realizes assumption 7 is the one described in [StClAl93].

Assumption 8 Clock differences are measured in a time span that is short enough to neglect drift during this period.
If this assumption can not be made, the clock differences must be interpolated to a fixed point in time based on the estimated drifts relative to the average clock. This merely complicates the formulae derived in section 5 but it does not present any fundamental problems [Lamm92]. However, it is assumed that some time elapses between the moment that the clock differences are measured and the moment that the clocks are resynchronized. The main reason for this assumption is that future research will be directed towards a hierarchical synchronization protocol. In such a protocol, the time elapsed between the measurements and the synchronization of the different groups at the lowest level may not be negligible.

A minor but non-negligible influence on the precision that can be achieved is the precision with which the hardware clock can be set. This is expressed by a clock setting error.

**Assumption 9** A correct clock $i$ can be set with an error that is less than or equal to $\varepsilon$. This maximal clock setting error is denoted by $\text{err}(\text{set}(c))$. The actual error made when clock $i$ is set during round $n$ at time $T$ is denoted by $\text{err}(\text{set}(c_i^n(T)))$.

## 4 Resynchronization Precision

In this section, it is shown that correct clocks resynchronize with a precision $\pi_0 = 12\frac{2}{3}\varepsilon$. Correct clocks are assumed to be synchronized within the precision $\pi$, i.e. the clocks $c_i^n(T)$ that have been corrected for local drift satisfy

$$|c_i^n(T) - c_j^n(T)| < \pi$$

for all correct clocks $i$ and $j$ during synchronization round $n$.

The CF exploits the properties of the $N$ by $N$ matrix of clock differences. Incorrect clocks are removed from the set of clocks which defines the average clock. It is shown that the influence on the average clock of incorrect clocks that remain in this set is bounded.

The matrix defined by $\Delta t_{ij} = c_i^n(T_{M_i}^n) - c_j^n(T_{M_j}^n)$ for an idealized set of correct clocks that can be read without errors satisfies several relations. The matrix is antisymmetric and there are relations between triples of clock differences, for example

$$\Delta t_{ij} + \Delta t_{jl} + \Delta t_{li} = 0$$

The actual relations between the matrix elements must take reading errors into account. Each matrix element requires two independent clock readings. In addition, the absolute value of a matrix element is bounded by the maximum drift rate and the resynchronization precision. Therefore, correct clocks $i$, $j$ and $l$ are required to satisfy the following predicates:
\begin{align}
\text{consistent}(i,j) &= |\Delta t_{ij} + \Delta t_{ji}| \leq 4\varepsilon \land |\Delta t_{ij}| \leq 2\rho_{max} \cdot T_S + \pi_0 \quad (4.2) \\
\text{consistent}(i,j,l) &= |\Delta t_{ij} + \Delta t_{ji} + \Delta t_{il}| \leq 6\varepsilon \land \\
&\quad |\Delta t_{ij} - \Delta t_{il} - \Delta t_{il}| \leq 6\varepsilon \land \\
&\quad |\Delta t_{ij} - \Delta t_{ij} - \Delta t_{il}| \leq 6\varepsilon \\
&\quad (4.3)
\end{align}

In 4.2, the term \(\pi_0\) is assumed to be small with respect to \(2\rho_{max} \cdot T_S\). The relation \(|\Delta t_{il}| \leq 2\varepsilon\) is not used since this is not useful at all. Relations between clock differences that involve four or more clocks are not used either, because this does not improve the results.

The convergence function first constructs a set \(C^n\) of clocks that satisfy 4.2 and 4.3. This set defines the average clock. Since all correct clocks have access to the same data (assumption 7), all correct clocks agree on the set \(C^n\). The error \(\zeta\) that describes incorrect clocks that remain in \(C^n\) is bounded by the restrictions 4.2 and 4.3.

It is possible to identify incorrect clocks as well. However, this is computationally much more complex and this task is not performed by the CS protocol. Identification of incorrect clocks requires that incorrect matrix elements are identified first. One or both of two clocks \(i\) and \(j\) must be incorrect if \(\Delta t_{ij}\) does not satisfy 4.2 or if the number of indices for which 4.3 is not satisfied exceeds \(k\), the maximum number of incorrect clocks in the system. Such a matrix element is clearly inconsistent. If for a clock \(i\) the number of inconsistent indices \(j\) for which \(\Delta t_{ij}\) is inconsistent exceeds \(k\), clock \(i\) must be incorrect. Details of this procedure are a subject for future research.

The set \(C^n\) is constructed as follows. First, \(C^n\) is initialized to the set of all clocks. Then, pairs of clocks that do not satisfy 4.2 are removed from the set \(C^n\) to a set \(P\). Once a clock is removed from \(C^n\), matrix elements that involve this clock are disregarded. Then, triples of clocks that do not satisfy 4.3 are removed in the same manner to a set \(T\). Matrix elements that involve clocks removed to \(T\) are disregarded as well. With each pair or triple removed, at least one incorrect clock is removed. Since \(N \geq 3k + 1\), this procedure always leaves at least one correct clock in \(C^n\). All clocks then estimate the average clock from the set \(C^n\). It is shown below that correct clocks in \(C^n\) estimate the average clock with an error of \(3\frac{1}{3}\varepsilon\), while correct clocks that are not in \(C^n\) estimate the average clock with an error of \(5\frac{1}{3}\varepsilon\). Before the proof can be given, two lemmas are required.

**Lemma 1** The ratio of correct clocks in \(C^n\) to incorrect clocks in \(C^n\) exceeds 2.

**Proof:**

Denote the number of pairs in \(P\) by \(|P|\) and the number of triples in \(T\) by \(|T|\). With each triple or pair, at least one incorrect clock is removed. Therefore, the number \(N_{corr}\) of correct clocks in \(C^n\) satisfies

\[
N_{corr} \geq 2k + 1 - 2|T| - |P| \\
(4.4)
\]

11
The number $N_{\text{incorr}}$ of incorrect clocks in $C^n$ satisfies

$$N_{\text{incorr}} \leq k - |T| - |P| \quad (4.5)$$

From this, the lemma follows directly. □

**Lemma 2** If two clocks $i$ and $j$ are synchronized with precision $\pi$, the absolute time at which local drift-corrected clocks reach the same value differs by at most $\pi + o(3\rho_{\text{max}} \cdot \pi)$.

**Proof:**

Denote the times at which the drift-corrected clocks reach the same value by $T_i$ and $T_j$. Let $T_A = \frac{1}{2}(T_i + T_j)$. Then, with definition 5, it follows that the drift-corrected clocks satisfy

$$c_i^\alpha(T_i) - c_j^\alpha(T_j)$$

$$= c_i^\alpha(T_A) - c_j^\alpha(T_A) + T_i - T_j + \int_{T_A}^{T_i} \rho_i(\tau) d\tau - \int_{T_A}^{T_j} \rho_j(\tau) d\tau$$

$$= 0$$

or

$$T_i - T_j = c_j^\alpha(T_A) - c_i^\alpha(T_A) + \int_{T_A}^{T_j} \rho_j(\tau) d\tau - \int_{T_A}^{T_i} \rho_i(\tau) d\tau \quad (4.6)$$

Here, $\rho_i'$ and $\rho_j'$ denote the drift of the drift-corrected clocks. With 4.2, definition 10 given below and definition 11 of a drift-corrected clock given in section 5, it is trivial to show that $|\rho_i(T)| < 3\rho_{\text{max}}$ for a correct clock $i$. Therefore, it follows from 4.6 that

$$|T_i - T_j| \leq |c_i^\alpha(T_A) - c_j^\alpha(T_A)| + 3\rho_{\text{max}} \cdot |T_i - T_j|$$

By assumption, the clocks are synchronized within the precision, therefore

$$|T_i - T_j| \leq \frac{\pi}{1 - 3\rho_{\text{max}}} = \pi + o(3\rho_{\text{max}} \cdot \pi)$$

□

**Theorem 1** Readings $\text{est}(c_i^\alpha(T_M^n))$ of an incorrect clock $i \in C^n$ can be described by a clock $c_i(T_M^n)$ that is read with a reading error $\zeta$ of $5\varepsilon$. 12
Proof:
The value of $\zeta$ is determined by the constraints imposed by 4.2 and 4.3 on $err(est(c^n_i))$. In order to describe a clock with an estimate and a reading error, the differences between all values reported by the clock must be bounded. The predicate $consistent(i, j)$ in 4.2 only restricts the difference between pairs of readings that involve two clocks, i.e., the difference between $est(c^n_i(T_{M_j}^n)) - est(c^n_j(T_{M_i}^n))$ and $est(c^n_j(T_{M_j}^n)) - est(c^n_i(T_{M_i}^n))$. Therefore, $consistent(i, j)$ merely ensures the consistency of the readings exchanged between clocks $i$ and $j$. Constraint 4.3 restricts differences between pairs of readings that involve three clocks such as the difference between $est(c^n_i(T_{M_i}^n)) - est(c^n_j(T_{M_j}^n))$ and $est(c^n_j(T_{M_j}^n)) - est(c^n_i(T_{M_i}^n))$. Constraint 4.3 thus relates all values reported by a clock and therefore ensures the consistency of all clock readings. Constraint 4.3 also imposes a bound on the readings exchanged between two clocks. However, this bound is much weaker than the bound that follows from 4.2.

Since the two constraints relate different pairs of readings of an incorrect clock, the weakest constraint determines the value of $\zeta$. It is easily shown that the weakest constraint is 4.3. As an example, it is shown that if $i$ is an incorrect clock and $j$ and $l$ are correct clocks, the relation

$$|\Delta t_{ij} + \Delta t_{jl} + \Delta t_{li}| \leq 6\varepsilon$$

restricts $\zeta$ to $5\varepsilon$. Proof of the restrictions from other constraints is completely analogous. Notice that if there is an incorrect clock in $C^n$, there are always more than two correct clocks in $C^n$ according to lemma 1. Therefore, the two correct clocks $j$ and $l$ in 4.7 can always be found. With equation 3.1,

$$|\Delta t_{ij} + \Delta t_{jl} + \Delta t_{li}| = |est(c^n_i(T_{M_j}^n)) - est(c^n_j(T_{M_i}^n)) +
est(c^n_j(T_{M_j}^n)) - est(c^n_i(T_{M_i}^n)) +
est(c^n_i(T_{M_j}^n)) - est(c^n_j(T_{M_i}^n))|$$

(4.8)

All clocks in $C^n$ are consistent with the precision. The differences between the absolute times $T_{M_j}^n$ and $T_{M_l}^n$ at which clocks $j$ and $l$ reach the same value thus differ by at most $\pi$ according to lemma 2. Since an incorrect clock is described by a clock that is read with a larger reading error according to definition 4, this clock is also correctly synchronized. Therefore, relations of the following type

$$est(c^n_x(T_{M_j}^n)) = est(c^n_y(T_{M_j}^n)) + (T_{M_y}^n - T_{M_x}^n) + o(\rho_{\max} \cdot \pi)$$

(4.9)

are valid for all clocks $x$ and $y$. When 4.9 is substituted in 4.8, the terms $T_{M_y}^n - T_{M_x}^n$ cancel. The two independent estimates $est(c^n_x(T_{M_j}^n))$ of each clock remain. Since these estimates differ by at most $2err(est(c^n_i))$, 4.7 reduces to
or $\zeta \leq 5\varepsilon$ since $i$ is assumed to be incorrect and $j$ and $l$ are assumed to be correct. Similarly, 4.2 leads to

$$2\text{err}(\text{est}(c_i^n)) \leq 4\varepsilon + 2\text{err}(\text{est}(c_j^n))$$

or $\zeta \leq 3\varepsilon$.

Evaluation of constraints in which two or more incorrect clocks occur only put stronger bounds on $\zeta$. For example, when both clock $i$ and clock $j$ in 4.7 are assumed incorrect, $\zeta$ is bounded by

$$2\text{err}(\text{est}(c_i^n)) + 2\text{err}(\text{est}(c_j^n)) \leq 6\varepsilon + 2\text{err}(\text{est}(c_l^n))$$

or $\zeta \leq 2\varepsilon$.

The values distributed to clocks in $C^n$ by an incorrect clock in $C^n$ are thus described by a clock that is read with an error of $5\varepsilon$.

\begin{definition}
The average absolute time at which the clock differences are determined is defined by

$$T^n_M = \frac{1}{|C^n|} \sum_{i \in C^n} T^n_{M_i}$$

From assumption 5 and lemma 2, it follows that for a correct clock $i \in C^n$

$$|T^n_M - T^n_{M_i}| < \pi + o(\rho_{\text{max}} \cdot \pi) \quad (4.10)$$

\end{definition}

\begin{definition}
The average system clock at absolute time $T^n_M$ is defined by

$$c^n_A(T^n_M) = \frac{1}{|C^n|} \sum_{i \in C^n} c^n_i(T^n_M)$$

The average clock $c^n_A(T)$ can be described by an initial value and a drift $\rho^n_A(T)$ as in definition 5. The drift of the average clock is defined as the average drift of the clocks in $C^n$ during the previous round thus $|\rho^n_A(T)| < \rho_{\text{max}}$. Consequently, the difference between a correct clock $i$ and the average clock equals

$$c^n_i(T^n_M) - c^n_A(T^n_M) = c^n_i(T^n_{M_i}) - c^n_A(T^n_{M_i}) + o(\rho_{\text{max}} \cdot \pi) \quad (4.11)$$

\end{definition}
Definition 10 A correct clock $i \in C^n$ obtains an estimate $\Delta t^{n}_{iA}$ of its difference with the average clock from

$$\Delta t^{n}_{iA} = \frac{1}{|C^n|} \sum_{j \in C^n \land j \neq i} \Delta t_{ij}$$

The estimate is denoted by $\Delta t_{iA}$ if the round number is unambiguous.

Notice that when reading errors are disregarded,

$$\Delta t_{iA} = \frac{1}{|C^n|} \sum_{j \in C^n} c^n_i(T^J_M) - c^n_j(T^J_M) = c^n_i(T_M) - c^n_A(T_M)$$

According to 4.11, this is equal to the difference between clock $i$ and the average clock at time $T^n_M$ up to terms of order $\rho_{max} \cdot \pi$. Therefore, when clock $i$ applies the correction $\Delta t_{iA}$ at time $T^n_M$, this can be described as if clock $i$ sets its clock equal to the average clock at time $T^n_M$.

In reality, reading errors cannot be ignored. The following theorem bounds the error in the estimate of $c^n_i(T^n_M) - c^n_A(T^n_M)$.

Theorem 2 A correct clock $i \in C^n$ estimates its difference with the average clock $c^n_A(T^n_M)$ with an error of $3\frac{1}{3} \varepsilon$.

Proof:

The error in the estimate $\Delta t_{iA}$ is determined by

$$|c^n_i(T^n_M) - c^n_A(T^n_M) - \Delta t_{iA}| = \left| c^n_i(T^n_M) - c^n_A(T^n_M) - \frac{1}{|C^n|} \sum_{j \neq i \land j \in C^n} \Delta t_{ij} \right|$$

(4.12)

according to 4.11 and definition 10. With equation 3.1 for $\Delta t_{ij}$ and definition 9 of the average clock, this can be rewritten as

$$\left| c^n_i(T^n_M) - \frac{1}{|C^n|} \sum_{j \in C^n} c^n_j(T^n_M) - \frac{1}{|C^n|} \sum_{j \in C^n \land j \neq i} \text{est}(c^n_i(T^n_M)) + \frac{1}{|C^n|} \sum_{j \in C^n} \text{est}(c^n_j(T^n_M)) \right|$$

(4.13)

Since the clocks are synchronized within the precision, an estimate $\text{est}(c^n_i(T^n_M))$ can be approximated by
\[
\text{est}(c_i^n(T^n_{M_i})) = \text{est}(c_i^n(T^n_M)) + (T^n_{M_i} - T^n_M) + o(\rho_{\text{max}} \cdot \pi)
\]

The terms \((T^n_{M_i} - T^n_M)\) from the last two terms on the right-hand side of 4.13 cancel. Therefore,

\[
|c_i^n(T^n_M) - c_i^n(T^n_M) - \Delta t_{iA}| = |c_i^n(T^n_M) - \frac{1}{|C^n|} \sum_{j \in C^n} c_j^n(T^n_M) + \frac{1}{|C^n|} \sum_{j \in C^n} \text{est}(c_j^n(T^n_M)) - \frac{1}{|C^n|} \sum_{j \in C^n} \text{est}(c_i^n(T^n_M))| + o(\rho_{\text{max}} \cdot \pi)
\]

With definition 3 of \(\text{err}(\text{est}(c_i^n))\),

\[
|c_i^n(T^n_M) - c_i^n(T^n_M) - \Delta t_{iA}| \leq \text{err}(\text{est}(c_i^n)) + \frac{1}{|C^n|} \sum_{j \in C^n} \text{err}(\text{est}(c_j^n)) + o(\rho_{\text{max}} \cdot \pi)
\]

By assumption, clock \(i\) is correct, therefore the contribution of the first term equals \(\epsilon\). When the number of incorrect clocks in \(C^n\) is denoted by \(\kappa\), the second term can be bounded according to theorem 1 by

\[
\frac{1}{|C^n|} \sum_{j \in C^n} \text{err}(\text{est}(c_j^n)) \leq \frac{1}{|C^n|} (|C^n| - \kappa) \cdot \epsilon + \kappa \cdot 5\epsilon = \frac{1}{|C^n|} (|C^n| \cdot \epsilon + \kappa \cdot 4\epsilon)
\]

It follows from lemma 1 that \(\frac{\kappa}{|C^n|} < \frac{1}{3}\). Therefore

\[
\frac{1}{|C^n|} \sum_{j \in C^n} \text{err}(\text{est}(c_j^n)) < 2 \cdot \frac{1}{3} \epsilon
\]

From 4.14, 4.15 and 4.16, it follows that \(\text{err}(\text{est}(c_i^n(T^n_M^n) - c_i^n(T^n_M))) < 3\frac{1}{3} \epsilon\).

A correct clock \(i \not\in C^n\) estimates its difference with the average clock by the following procedure. First, it calculates for each clock \(m \in C^n\) the value of \(\Delta t_{im} + \Delta t_{mA}\). Then it removes the \(k - |T| - |P|\) clocks with the highest and the \(k - |T| - |P|\) clocks with the lowest value of \(\Delta t_{im} + \Delta t_{mA}\). Therefore, either all incorrect clocks are removed or there are correct clocks both among the clocks with the highest and among the clocks with the lowest values of \(\Delta t_{im} + \Delta t_{mA}\). The clock then estimates its difference with the average clock from any clock \(j\) of the remaining clocks:

\[
\Delta t_{iA} = \Delta t_{ij} + \Delta t_{jA}
\]
Theorem 3  The error in the estimate of the deviation from the average clock of a correct clock \( i \notin \mathcal{C}^n \) is bounded by \( 5\frac{1}{3}\varepsilon \). This bound is denoted by \( \text{err}(\text{est}(\Delta t)) \).

Proof:  
The clock \( j \) from which \( i \) determines its difference with the average clock is either correct or incorrect. If \( j \) is not correct, there must be correct clocks \( l \) and \( m \in \mathcal{C}^n \) among the \( k - |\mathcal{T}| - |\mathcal{P}| \) clocks with the highest respectively lowest values of \( \Delta t_{im} + \Delta t_{mA} \) with

\[
\begin{align*}
\Delta t_{ij} + \Delta t_{jA} &\leq \Delta t_{il} + \Delta t_{lA} \quad (4.18) \\
\Delta t_{ij} + \Delta t_{jA} &\geq \Delta t_{im} + \Delta t_{mA} \quad (4.19)
\end{align*}
\]

If \( j \) is correct, \( l \) and \( m \) can be chosen equal to \( j \). The error in the estimate of \( c_i^n(T_M) - c_A^n(T_M) \) is bounded from above by 4.18 and from below by 4.19. Since the derivations are completely analogous, only the upper bound is derived. The error in the estimate of the true difference between clock \( i \) and the average clock is given by

\[
\begin{align*}
\Delta t_{iA} - c_i^n(T_M) + c_A^n(T_M) \\
= & \quad [ \text{definition 4.17 for } \Delta t_{iA} ] \\
\triangleq & \quad \Delta t_{jA} + \Delta t_{ij} - c_i^n(T_M) + c_A^n(T_M) \\
\leq & \quad [4.18] \\
\Delta t_{iA} + \Delta t_{il} - c_i^n(T_M) + c_A^n(T_M) \\
= & \quad [ \text{equation 4.10 and assumption 5 about the drift rate} ] \\
\triangleq & \quad \Delta t_{iA} + \text{est}(c_i^n(T_M)) - \text{est}(c_i^n(T_M)) - c_i^n(T_M) + c_A^n(T_M) \\
= & \quad [ \text{rearranging terms} ] \\
(\Delta t_{iA} - c_i^n(T_M) + c_A^n(T_M)) + (\text{est}(c_i^n(T_M)) - c_i^n(T_M)) + (c_i^n(T_M) - \text{est}(c_i^n(T_M))) \\
\leq & \quad [ \text{definition 3 and definition 10} ] \\
\text{err}(\text{est}(c_i^n(T_M)) - c_A^n(T_M))) + \text{err}(\text{est}(c_i^n)) + \text{err}(\text{est}(c_i^n)) + o(\rho_{\text{max}} \cdot \pi) \\
< & \quad [ \text{theorem 2 and assumption 4; } l \in \mathcal{C}^n \text{ and } i \text{ are by assumption correct} ] \\
5\frac{1}{3}\varepsilon + o(\rho_{\text{max}} \cdot \pi) \quad (4.20)
\end{align*}
\]
The clocks thus resynchronize with a precision of $\pi_0 = 12\frac{2}{3}\varepsilon$ since it is assumed that $\text{err}(\text{set}(c)) = \varepsilon$. Notice that the clocks can be made to synchronize to any value wanted by adding a constant term to each clock in addition to the estimated difference with the average clock.

5 Precision

The proof in this section shows that drift-corrected clocks $c_i^n$ are synchronized within the precision during round $n$. It is also shown that the maximal correction applied to any clock at the end of a round is at most $\frac{1}{2}(\pi + \pi_0)$. A correct clock $i$ measures its difference with other clocks at the end of round $n - 1$ at time $T_{M_i}^{n-1}$ at which the drift-corrected clock $c_i^{n-1}(T_{M_i}^{n-1}) = (n-1)T_S$. At time $T_{E_i}^{n-1}$ defined by $c_i^{n-1}(T_{E_i}^{n-1}) = (n-1)T_S + T'_S$, the new clocks $c_i^n$ and $c_i^n$ are started. When these clocks are started, drift corrections applied during the period $[T_{M_i}^{n-2}, T_{E_i}^{n-1}]$ are undone. The estimated difference $\Delta t_{i,n}^n$ with the average clock at the end of round $n - 1$ is subtracted from the clocks. In addition, the new drift correction for the period $[T_{M_i}^{n-1}, T_{E_i}^{n-1}]$ is applied to the drift-corrected clocks. It is shown that the corrected clocks remain synchronized up to time $T_{E_i}^n$.

A constant term $\sum_{j \in C^{n-1}} \Delta t_{jA}^n$ is subtracted from all clocks at a resynchronization. Due to this term, the average clock at the beginning of round $n$ has the value $(n-1)T_S + T'_S$. This will be shown after the precision is proven. Since this term requires that a clock $i \in C^{n-1}$ has been able to obtain an estimate of its difference with the average clock at time $T_{M_i}^{n-2}$, this requirement is an additional condition for a clock $i \in C^{n-1}$. It is simple to verify that all results derived before, e.g. lemma 1, remain valid when a third set $S$ of single clocks excluded from $C^{n-1}$ is introduced.

For the proof of the precision, it is convenient to describe the clocks $c_i^n$ and $c_i^n$ as if they were started at time $T_{M_i}^{n-1}$. It is argued below that this is allowed.

Definition 11 The clock function $c_i^n(T)$ of a correct clock $i$ and the corresponding drift corrected clock $c_i^n$ are defined for $T \geq T_{M_i}^{n-1}$ by

\begin{align}
    c_i^n(T_{M_i}^{n-1}) &= c_i^{n-1}(T_{M_i}^{n-1}) - \Delta t_{iA}^{n-1} - \frac{1}{|C^{n-1}|} \sum_{j \in C^{n-1}} \Delta t_{jA}^n + \text{err}(\text{set}(c_i^n(T_{M_i}^{n-1}))) \quad (5.1) \\
    c_i^n(T) &= c_i^n(T_{M_i}^{n-1}) + \int_{T_{M_i}^{n-1}}^{T} (1 + \rho_i(\tau))d\tau \quad (5.2) \\
    c_i^n(T) &= c_i^n(T) - \frac{c_i^n(T) - c_i^n(T_{M_i}^{n-1})}{T_S} \cdot \Delta t_{iA}^n \quad (5.3)
\end{align}

However, the new clocks are actually started at time $T_{E_i}^{n-1}$. These clocks are described by
\[ c_i^n(T_{E_i}^{n-1}) = c_i^n(T_{E_i}^{n-1}) + \frac{T_S + T'_S}{T_S} \cdot \Delta t_i^{n-2} - \Delta t_i^{n-1} - \frac{1}{|C_{n-1}|} \sum_{j \in C_{n-1}} \Delta t_j^{n-2} + \text{err}(\text{set}(c_i^n(T_{E_i}^{n-1}))) \]  
\[ c_i^n(T) = c_i^n(T_{E_i}^{n-1}) + \int_{T_{E_i}^{n-1}}^{T} (1 + \rho_i(\tau)) d\tau \]  
\[ c_i^n(T) = c_i^n(T) - \frac{c_i^n(T) - c_i^n(T_{E_i}^{n-1}) + T'_S}{T_S} \cdot \Delta t_i^{n-1} \]

for \( T \geq T_{E_i}^{n-1} \). Notice that 5.4 accurately describes how the new clock value is based on the value of the drift-corrected clock which is maintained in hardware. The uncorrected clock is derived from the drift-corrected clock and the drift corrections applied between \( T_{M}^{n-2} \) and \( T_{E_i}^{n-1} \). The reason for this arrangement is efficiency: the value of the drift-corrected clock is required more often than the value of the uncorrected clock.

**Lemma 3**

The clocks described by equations 5.1 to 5.3 are equivalent with the clocks described by equations 5.4 to 5.6, i.e. they correspond to independent settings of a hardware clock to equivalent mathematical clocks.

**Proof:**

It may be assumed that the clocks are identical during round \( n-1 \) up to time \( T_{M}^{n-1} \). After this time, the clocks 5.1 to 5.3 are resynchronized while the clocks 5.4 to 5.6 continue to run until time \( T_{E_i}^{n-1} \) and are then resynchronized. From equation 5.6 for round \( n-1 \) at time \( T_{E_i}^{n-1} \), it follows that

\[ c_i^{n-1}(T_{E_i}^{n-1}) = c_i^{n-1}(T_{E_i}^{n-1}) + \frac{T_S + T'_S}{T_S} \cdot \Delta t_i^{n-2} \]  

because \( c_i^{n-1}(T_{E_i}^{n-1}) - c_i^{n-1}(T_{E_i}^{n-2}) = T_S \). Therefore, disregarding clock setting errors, the first two terms on the right-hand side of 5.4 are exactly equal to \( c_i^{n-1}(T_{E_i}^{n-1}) \). When this is substituted in 5.5, the right-hand sides of 5.2 and 5.5 are easily seen to be equal in terms of the uncorrected clock of the previous round for \( T \geq T_{E_i}^{n-1} \) with the aid of the relation

\[ c_i^{n-1}(T_{E_i}^{n-1}) = c_i^{n-1}(T_{M}^{n-1}) + \int_{T_{M}^{n-1}}^{T_{E_i}^{n-1}} (1 + \rho_i(\tau)) d\tau \]

To show that the drift-corrected clocks 5.3 and 5.6 are equivalent for \( T \geq T_{E_i}^{n-1} \), it is thus sufficient to show that the drift-correction terms on the right-hand sides of 5.3 and 5.6 are equivalent disregarding clock setting errors. Equation 5.3 can be rewritten as

\[ c_i^n(T) \left( 1 + \frac{\Delta t_i^{n-1}}{T_S} \right) = c_i^n(T) + \frac{c_i^n(T_{M}^{n-1})}{T_S} \Delta t_i^{n-1} \]
According to equation 5.3 for time $T_M^{n-1}$, $c_i^{n-1}(T_M^{n-1}) = c_i^n(T_M^{n-1})$. With this and with 5.1, equation 5.8 can be written as

$$c_i^n(T) \left(1 + \frac{\Delta t_{iA}^{n-1}}{T_S} \right) = c_i^n(T) + \frac{c_i^{n-1}(T_M^{n-1}) - \Delta t_{iA}^{n-1} - \frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} \Delta t_{iA}^{n-2}}{T_S} \Delta t_{iA}^{n-1} \quad (5.9)$$

Likewise, 5.6 can be rewritten as

$$c_i^n(T) \left(1 + \frac{\Delta t_{iA}^{n-1}}{T_S} \right) = c_i^n(T) + \frac{c_i^n(T_{Ei}^{n-1}) - \frac{T_E'}{T_S} \Delta t_{iA}^{n-1}}{T_S} \quad (5.10)$$

Equation 5.6 for time $T_{Ei}^{n-1}$ reads

$$c_i^n(T_{Ei}^{n-1}) = c_i^n(T_{Ei}^{n-1}) - \frac{T_E'}{T_S} \Delta t_{iA}^{n-1}$$

Here, $c_i^n(T_{Ei}^{n-1})$ can be replaced by the right-hand side of equation 5.4, again disregarding clock setting errors:

$$c_i^n(T_{Ei}^{n-1}) = c_i^{n-1}(T_{Ei}^{n-1}) + \frac{T_E}{} + \frac{T_E'}{T_S} \Delta t_{iA}^{n-2} - \Delta t_{iA}^{n-1} - \frac{1}{|C^{n-1}|} \sum_{j \in C^{n-1}} \Delta t_{jA}^{n-2} - \frac{T_E'}{T_S} \Delta t_{iA}^{n-1} \quad (5.11)$$

Since $c_i^{n-1}(T_{Ei}^{n-1}) = c_i^{n-1}(T_{Mi}^{n-1}) + T_E'$, this can be rewritten as

$$c_i^n(T_{Ei}^{n-1}) = c_i^{n-1}(T_{Mi}^{n-1}) - \Delta t_{iA}^{n-1} + T_E' + \frac{T_E}{} + \frac{T_E'}{T_S} \Delta t_{iA}^{n-2} - \Delta t_{iA}^{n-1} - \frac{1}{|C^{n-1}|} \sum_{j \in C^{n-1}} \Delta t_{jA}^{n-2} - \frac{T_E'}{T_S} \Delta t_{iA}^{n-1}$$

According to 5.6 for round $n - 1$ at time $T_M^{n-1}$, $c_i^{n-1}(T_{Mi}^{n-1}) = c_i^{n-1}(T_{Mi}^{n-1}) - \Delta t_{iA}^{n-2}$. Thus

$$c_i^n(T_{Ei}^{n-1}) = c_i^{n-1}(T_{Mi}^{n-1}) - \Delta t_{iA}^{n-1} - \frac{1}{|C^{n-1}|} \sum_{j \in C^{n-1}} \Delta t_{jA}^{n-2} + \frac{T_E'}{T_S} (\Delta t_{iA}^{n-2} - \Delta t_{iA}^{n-1}) + T_E'$$

Therefore, the drift correction term of 5.10 equals

$$c_i^{n-1}(T_{Mi}^{n-1}) - \Delta t_{iA}^{n-1} - \frac{1}{|C^{n-1}|} \sum_{j \in C^{n-1}} \Delta t_{jA}^{n-2} + \frac{T_E'}{T_S} (\Delta t_{iA}^{n-2} - \Delta t_{iA}^{n-1})$$

There are two differences between the drift-correction terms of 5.9 and 5.10. The first difference is caused by the times at which the clocks are started, reflected by $c_i^{n-1}(T_{Mi}^{n-1})$ in 5.11 and $c_i^{n-1}(T_{Mi}^{n-1})$ in 5.9. There is also a minor difference due to the different estimates of the drift during the period $[T_M^{n-1}, T_{E}^{n-1})$. This is reflected by the term $\frac{T_E'}{T_S} (\Delta t_{iA}^{n-2} - \Delta t_{iA}^{n-1})$. The resulting difference between the drift-corrected clocks is of order $p_{\text{max}}^2 \cdot T_S$ and can thus be ignored. □
for(;;){ /* loop over synchronization periods */
est = - int('Δt_{IA}' + 0.5); /* rounded number of clock ticks */
nslice = nslices / est; /* nslices is the number of clock periods */
tend = nstep * (nslices/nstep); /* scaled units of the time axis */
tcurr = 0; t = 0; /* end time iteration */
est = nslices - 2 * est * nstep /* time and waiting time */
while ( tcurr != tend){
  if ( es < 0 ){
    es += -2 * nstep * est + 2 * nslices;
    t += nstep; tcurr += nstep;
    'wait(t)';
    'advance clock by 1 tick';
    t = 0
  }
  else {
    es -= 2 * nstep * est;
    t += nstep ; tcurr += nstep;
  }
}
t = nslices - tcurr; /* last interval of synchronization period */
'wait(t)';
if ( es < 0 ) 'advance clock by 1 tick';
}

Figure 3: Modified Bresenham algorithm to estimate drift corrections

The continuous part of 5.3 is estimated with a version of Bresenham's line discretization algorithm [Bres65] that can be executed by a separate thread or process. The algorithm shown in figure 3 only works for negative values of Δt_{IA}, but can easily be adapted to handle positive values as well.

The algorithm guarantees that the difference between the actual and intended correction is always less than ε if the coordinates of the end points are integers. Since the number of ticks to correct may be fractional, this introduces another error of ε. The maximum error in the estimate of the drift correction is thus equal to 2ε and is denoted by err(est(dc)).

The algorithm has been transformed to minimize the number of iterations. It is assumed that the line in the clock period-drift correction plane is in the first octant. If this is not the case, the optimized algorithm can not be used. However, in most cases the relative drifts will be of the order of 10^-5 to 10^-6. With a clock period of 20 ms and ε of 2μs, the angle of the line is normally close to 0°. For a line near 0°, many iterations are needed for a correction of a single clock tick. Therefore, the unit of the time axis is chosen such that the line is as close to 45° as possible. In most cases, a single iteration then suffices to determine when the next single tick correction must be made. Occasionally, two iterations are necessary.
Before the precision can be proven, a definition and a few lemmas are required.

**Definition 12** Each correct clock estimates its drift with respect to the drift of the average clock at time $T_{M}^{n-1}$ from the estimate of its difference with average time:

$$est(p_{iA}^{n}(T_{M}^{n-1})) = \frac{\Delta t_{iA}^{n-1}}{T_{S}}$$

(5.12)

The error in this estimate is determined by 1) the errors in the estimates of the difference with the average clock at $T_{M}^{n-2}$ and $T_{M}^{n-1}$, 2) the clock setting error $err(set(c_{i}^{n}(T_{M}^{n-2})))$, and 3) the maximum rate $R_{max}$ at which the drift can change during round $n-1$. For the proof of the precision, it is most convenient to calculate the maximum difference between two correct clocks $i$ and $j$ at time $T_{M}^{n}$. The proof requires the following lemma:

**Lemma 4** The ratio of the actual and intended length of a synchronization round is equal to 1 up to terms of order $\rho_{max}$ and $\varepsilon_{\varepsilon}$:

$$\left| \frac{T_{M}^{n} - T_{M}^{n-1}}{T_{S}} \right| = 1 + o(\rho_{max}) + o(\frac{\varepsilon}{T_{S}})$$

Proof:

Let $i$ be a correct clock $\in C^{n}$.

$$\left| \frac{T_{M}^{n} - T_{M}^{n-1}}{T_{S}} \right| \leq \left| \frac{T_{M}^{n} - T_{M}^{n-1}}{T_{S}} \right| + \left| \frac{T_{M}^{n} - T_{M}^{n}}{T_{S}} \right|$$

(5.13)

With definition 11 for time $T_{M}^{n}$,

$$c_{i}^{n}(T_{M}^{n}) = c_{i}^{n}(T_{M}^{n}) - c_{i}^{n}(T_{M}^{n}) - c_{i}^{n}(T_{M}^{n}) = c_{i}^{n}(T_{M}^{n}) - c_{i}^{n}(T_{M}^{n}) + \int_{T_{M}^{n-1}}^{T_{M}^{n}} (1 + \rho_{i}(\tau))d\tau - \Delta t_{iA}^{n-1}$$

where the second equality follows from $c_{i}^{n}(T_{M}^{n}) - c_{i}^{n}(T_{M}^{n}) = T_{S}$. Since $c_{i}^{n}(T_{M}^{n}) - c_{i}^{n}(T_{M}^{n}) = c_{i}^{n}(T_{M}^{n}) - c_{i}^{n}(T_{M}^{n})$, integration, rearranging terms and division by $T_{S}$ leads to

$$\frac{T_{M}^{n} - T_{M}^{n-1}}{T_{S}} = 1 - \frac{1}{T_{S}} \int_{T_{M}^{n-1}}^{T_{M}^{n}} \rho_{i}(\tau) + \frac{1}{T_{S}} \Delta t_{iA}^{n-1}$$

With assumption 5 about the drift rate and $|\Delta t_{iA}^{n-1}| < 2\rho_{max} \cdot T_{S} + \pi_{0}$, which follows from 4.2 and definition 10:

$$\frac{T_{M}^{n} - T_{M}^{n-1}}{T_{S}} < 1 + \rho_{max} \cdot \frac{T_{M}^{n} - T_{M}^{n-1}}{T_{S}} + 2\rho_{max} + o(\frac{\varepsilon}{T_{S}})$$
This can be rewritten as
\[ \frac{T_{M_1}^{n} - T_{M_1}^{n-1}}{T_S} < 1 + o(3\rho_{max}) + o(\frac{\varepsilon}{T_S}) \]

The lower bound on this term can be derived analogously. This bounds the first term of 5.13. A bound on the second term of 5.13 follows from the definition of \( T_{M_1}^{n} \) and from constraint 4.2 that is satisfied by all clocks in \( C^n \). Neglecting small terms, \( |T_{M_1}^{n} - T_{M_1}^{n}| \leq 2\rho_{max} \cdot T_S \) for \( j \in C^n \). Therefore, the second term of 5.13 is bounded by \( o(2\rho_{max}) \). With these results, lemma 4 is derived.

The following lemma bounds the error in the estimate of the relative drift \( \rho_{ij} \) at time \( T_{M_1}^{n-1} \).

**Lemma 5** The error in the estimate of the relative drift \( \rho_{ij} \) between correct clocks \( i \) and \( j \) satisfies

\[
\text{err}(\text{est}(\rho_{ij}(T_{M_1}^{n-1}))) < R_{max} \cdot T_S + \frac{2\pi_0 - 2\text{err}(\text{set}(c))}{T_S}
\]

up to terms of order \( \rho_{max}^2 \) and \( \rho_{max} \cdot R_{max} \cdot T_S \).

**Proof:**

The drift that is actually estimated is the average drift during the previous round, relative to the drift of the average clock. With definition 12

\[
\text{est}(\rho_{ij}(T_{M_1}^{n-1})) = \text{est}(\rho_{iA}(T_{M_1}^{n-1})) = \text{est}(\rho_{jA}(T_{M_1}^{n-1})) = \frac{\Delta t_{iA}^{n-1} - \Delta t_{jA}^{n-1}}{T_S}
\]

The error in the relative drift can then be bounded as follows:

\[
\text{err}(\text{est}(\rho_{ij}(T_{M_1}^{n-1}))) = |\rho_{ij}(T_{M_1}^{n-1}) - \text{est}(\rho_{ij}(T_{M_1}^{n-1}))| = |\rho_{ij}(T_{M_1}^{n-1}) - \frac{\Delta t_{iA}^{n-1} - \Delta t_{jA}^{n-1}}{T_S}| \leq |\text{theorem 3}|
\]

\[
|\rho_{ij}(T_{M_1}^{n-1}) - \frac{c_{i}^{n-1}(T_{M_1}^{n-1}) - c_{A}^{n-1}(T_{M_1}^{n-1}) + c_{A}^{n-1}(T_{M_1}^{n-1}) - c_{j}^{n-1}(T_{M_1}^{n-1})}{T_S}| + \frac{2\text{err}(\text{est}(\Delta t))}{T_S}
\]

\[
= |\text{definition 11}|
\]

\[
|\rho_{ij}(T_{M_1}^{n-1}) - \frac{c_{i}^{n-1}(T_{M_1}^{n-2}) - c_{j}^{n-1}(T_{M_1}^{n-2})}{T_S} - \frac{1}{T_S} \int_{T_{M_1}^{n-2}}^{T_{M_1}^{n-1}} (\rho_i(\tau) - \rho_j(\tau)) d\tau| + \frac{2\text{err}(\text{est}(\Delta t))}{T_S}
\]

23
\begin{align*}
&\leq [\text{inclusion of } \rho_i(T_M^{n-1}) \text{ under the integral, lemma 4}] \\
&\quad \frac{1}{T_S} \left| \Delta i^{(n-1)}(T_M^{n-2}) - \Delta j^{(n-1)}(T_M^{n-2}) \right| + \frac{1}{T_S} \left| \int_{T_M^{n-2}}^{T_M^{n-1}} (\rho_i(T_M^{n-1}) - \rho_i(\tau) - \rho_j(T_M^{n-1}) + \rho_j(\tau)) d\tau \right| \\
&\quad + \frac{2\text{err}(\text{est}(\Delta t))}{T_S} + o(\rho_{\text{max}}^2) \\
&\quad + \frac{2\text{err}(\text{est}(\Delta t))}{T_S} + o(\rho_{\text{max}}^2)
\end{align*}

(5.15)

A bound on the first term of 5.15 follows from theorem 3:

\[ \left| \frac{\Delta i^{(n-1)}(T_M^{n-2}) - \Delta j^{(n-1)}(T_M^{n-2})}{T_S} \right| \leq \frac{\pi_0 \pi_{1}}{T_S} \]

The second term of 5.15 is bounded by assumption 6 on the derivative of the drift:

\[ \int_{T_M^{n-2}}^{T_M^{n-1}} (\rho_i(T_M^{n-1}) - \rho_i(\tau)) d\tau < \int_{T_M^{n-2}}^{T_M^{n-1}} R_{\text{max}}(T_M^{n-1} - \tau) d\tau = \frac{1}{2} R_{\text{max}} \cdot T_S^2 + o(R_{\text{max}} \cdot \rho_{\text{max}} \cdot T_S) \]

Therefore, the second term is bounded by \( R_{\text{max}} \cdot T_S + o(\rho_{\text{max}} \cdot R_{\text{max}} \cdot T_S) \). Combining these results,

\[ \text{err}(\text{est}(\rho_i(T_M^{n-1}))) < R_{\text{max}} \cdot T_S + \frac{2\pi_0 - 2\text{err}(\text{est}(\Delta c))}{T_S} + o(\rho_{\text{max}}^2) + o(\rho_{\text{max}} \cdot R_{\text{max}} \cdot T_S) \]

Notice that \( R_{\text{max}} \) is very small; in the experiment described above, the largest average value of the derivative of the drift was \( 5 \cdot 10^{-8} \).

\[ \square \]

**Theorem 4** The maximum difference between two correct clocks \( c_i^n \) and \( c_j^n \) is less than the precision \( \pi \) provided that

\[ R_{\text{max}} < \frac{\pi - (\pi_0 - \text{err}(\text{est}(\Delta c))) \cdot (1 + 2\frac{T_0 + T_4}{T_S}) - \text{err}(\text{est}(\Delta c)) - 2\pi_0}{(T_S + T_5) \cdot (2T_5 + T_5)} \]

(5.16)

**Proof:**

In order to prove the precision, the maximum difference between two correct clocks at time \( T_E^p \) must be calculated first. With definition 11 of a drift-corrected clock,

\[ c_i^n(T_E^p) = c_i^n(T_M^{n-1}) + \int_{T_M^{n-1}}^{T_E^p} (1 + \rho_i(\tau)) d\tau = c_i^n(T_M^{n-1}) - c_i^n(T_M^{n-1}) \cdot T_S \]

(5.17)

The third term on the right-hand side of 5.17 can be rewritten with the approximation

\[ c_i^n(T_E^p) - c_i^n(T_M^{n-1}) = c_i^n(T_M^{n-1}) - c_i^n(T_M^{n-1}) + o(\rho_{\text{max}} \cdot (T_S + T_5)) \]

24
Since \( c_i^n(T_E^n) - c_j^n(T_M^{n-1}) = T_S + T'_S \),
\[
\frac{c_i^n(T_E^n) - c_j^n(T_M^{n-1})}{T_S} \cdot \Delta t_{iA}^{n-1} = \frac{T_S + T'_S}{T_S} \cdot \Delta t_{iA}^{n-1} + o\left(\rho_{\text{max}} \cdot (T_S + T'_S) \cdot (2\rho_{\text{max}} \cdot T_S + \pi_0)\right)
\]
\( (5.18) \)

In this formula, the error introduced by the estimate of the drift correction is not expressed. Below, this error is accounted for.

The difference between the drift-corrected clocks \( i \) and \( j \) can now be bounded.
\[
|c_i^n(T_E^n) - c_j^n(T_E^n)|
\leq [5.17 \text{ and } 5.18; \text{ the term } \pi_0 \text{ in } 5.18 \text{ is neglected with respect to } \rho_{\text{max}} \cdot T_S]
\]
\[
|c_i^n(T_M^{n-1}) - c_j^n(T_M^{n-1}) + \int_{T_M^{n-1}}^{T_E^n} \rho_{ij}(\tau) d\tau - \frac{T_S + T'_S}{T_S} \cdot (\Delta t_{iA}^{n-1} - \Delta t_{jA}^{n-1}) + 2\text{err(est}(dc)) + o(\rho^2_{\text{max}} \cdot (T_S + T'_S))
\]
\[
< [\text{ with assumption } 6 \text{ about the derivative of the drift, theorem } 3 \text{ and lemma } 4]
\]
\[
\int_{T_M^{n-1}}^{T_E^n} (\rho_i(T_M^{n-1}) - \rho_j(T_M^{n-1})) d\tau + 2\int_{T_M^{n-1}}^{T_E^n} R_{\text{max}}(\tau - T_M^{n-1}) d\tau - (T_S + T'_S)\text{est}(\rho_{ij}(T_M^{n-1})) + \pi_0 + 2\text{err(est}(dc)) + o(\rho^2_{\text{max}} \cdot (T_S + T'_S))
\]
\[
\leq [\text{integration; } T_E^n - T_M^{n-1} = T_S + T'_S + o(\rho_{\text{max}}(T_S + T'_S))]
\]
\[
|\rho_{ij}(T_M^{n-1}) \cdot (T_S + T'_S) + R_{\text{max}} \cdot (T_S + T'_S)^2 - \text{est}(\rho_{ij}(T_M^{n-1})) \cdot (T_S + T'_S)| + \pi_0 + 2\text{err(est}(dc)) + o(\rho^2_{\text{max}} \cdot (T_S + T'_S))
\]
\[
= [\text{ definition } 12 \text{ of the estimate of the drift correction}]
\]
\[
|\text{err(est}(\rho_{ij}(T_M^{n-1}))) \cdot (T_S + T'_S) + R_{\text{max}} \cdot (T_S + T'_S)^2| + \pi_0 + 2\text{err(est}(dc)) + o(\rho^2_{\text{max}} \cdot (T_S + T'_S))
\]
\[
< [\text{ lemma } 5]
\]
\[
R_{\text{max}} \cdot T_S \cdot (T_S + T'_S) + R_{\text{max}} \cdot (T_S + T'_S)^2 + (2\pi_0 - 2\text{err(est}(c))) \cdot \frac{T_S + T'_S}{T_S} + \pi_0 + 2\text{err(est}(dc)) + o(\rho^2_{\text{max}} \cdot (T_S + T'_S)) + o(R_{\text{max}} \cdot \rho_{\text{max}} \cdot T_S \cdot (T_S + T'_S))
\]
\[
= [\text{ rearranging terms }]
\]
\[
R_{\text{max}} \cdot (T_S + T'_S) \cdot (2T_S + T'_S) + (\pi_0 - \text{err(set}(c))) \cdot (1 + 2\frac{T_S + T'_S}{T_S}) + 2\text{err(est}(dc)) + o(\rho^2_{\text{max}} \cdot (T_S + T'_S)) + o(R_{\text{max}} \cdot \rho_{\text{max}} \cdot T_S \cdot (T_S + T'_S))
\]

Essentially, the difference between the clocks is bounded by \( 2R_{\text{max}} \cdot T_S^2 + 3\pi_0 \) (with \( T'_S = 0 \)). The first term represents the deviation due to changes in the drift during the two synchronization rounds involved. The second term results from 1) the errors in the estimates of average time at the beginning and end of round \( n - 1 \) and 2) the resynchronization precision at the beginning of round \( n \).
Proof of the precision can now be given trivially since inequality 5.16 follows directly from the derivation above. Since the maximum possible difference between two drift-corrected clocks is attained around time $T^n_E$, $|c^n_i(T^n_E) - c^n_j(T^n_E)|$ must be smaller than $\pi$. Notice that the difference between clocks with $T^n_{E_i} > T^n_{E_j}$ can increase only marginally between $T^n_{E_i}$ and $T^n_{E_j}$ by terms of order $\rho_{max} \cdot (T_S + T'_S)$.

The value for $R_{max}$ is determined by the environment of the system which determines the stability of the drift. The length of the synchronization period that can be attained thus depends on the stability of the drift. With $\pi = 100\mu s$, $T_S = 10s$, $T'_S = 0$, $\varepsilon = 2 \mu s$, $err(set(c)) = \varepsilon$ and $err(est(dc)) = 2\varepsilon$, the maximum value of $R_{max} = 10^{-7}$. Notice that this is more than sufficient to guarantee the precision under the circumstances of the experiment mentioned before where the maximum average rate of change of drift was approximately $5 \cdot 10^{-8}$.

There are cases when it is not possible to apply drift corrections. Obviously, this is the case when $\pi \leq 3\pi_0$. In this case, the length of the synchronization period is determined by $T_S < (\pi - \pi_0)/2\rho_{max}$. Also when $\pi > \pi_0$, it may not be useful to apply drift corrections as the length of the resulting synchronization round may be smaller than $(\pi - \pi_0)/2\rho_{max}$. In particular, this can be the case when the assumed $\rho_{max}$ is small.

The term $\frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} \Delta t_{iA}^{n-2}$ in 5.1 ensures that both the corrected and uncorrected clocks have values around $(n-1)T_S$ at the beginning of round $n$. This is shown by the following lemma.

**Lemma 6** The average clock $c^n_A$ has the value $(n-1)T_S$ at the beginning of round $n$ up to terms of order $\varepsilon$ and $\rho_{max} \cdot \pi$.

**Proof:**

From 5.3 at the end of round $n - 1$, it follows that

$$c^{n-1}_i(T^{n-1}_M) = c^{n-1}_i(T^{n-1}_M) - \Delta t_{iA}^{n-2} = (n-1)T_S$$

Or

$$c^{n-1}_i(T^{n-1}_M) = (n-1)T_S + \Delta t_{iA}^{n-2}$$

Therefore

$$\frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} c^{n-1}_i(T^{n-1}_M) = (n-1)T_S + \frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} \Delta t_{iA}^{n-2}$$

According to 5.1, the average clock of round $n$ at time $T^{n-1}_M$ equals

$$\frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} c^n_i(T^{n-1}_M) = \frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} c^{n-1}_i(T^{n-1}_M) - \frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} \Delta t_{iA}^{n-1} + \frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} \Delta t_{iA}^{n-2} + \frac{1}{|C^{n-1}|} \sum_{i \in C^{n-1}} err(set(c_i))$$
The second term on the right-hand side of this equation is easily shown to be equal to 0. The first term on the right-hand side of the equation can be approximated with the aid of the definition of a clock:

\[
\frac{1}{|C_{n-1}|} \sum_{i \in C_{n-1}} c_i^{n-1}(T_M^{n-1}) = \frac{1}{|C_{n-1}|} \sum_{i \in C_{n-1}} \left(c_i^{n-1}(T_M^{n-1}) + T_M^{n-1} - T_M^{n-1}\right) + o(\rho_{\text{max}} \cdot \pi) \\
= \frac{1}{|C_{n-1}|} \sum_{i \in C_{n-1}} c_i^{n-1}(T_M^{n-1}) + o(\rho_{\text{max}} \cdot \pi)
\]

(5.21)

where the last equality follows from the definition of \(T_M^{n-1}\). When this result is substituted in equation 5.20, it follows with equation 5.19 that

\[
\frac{1}{|C_{n-1}|} \sum_{i \in C_{n-1}} c_i^{n}(T_M^{n-1}) = (n - 1)T_S + \text{err}(\text{set}(c)) + o(\rho_{\text{max}} \cdot \pi)
\]

The clocks are thus reset to a multiple of \(T_S\) at the end of a round.

\(\Box\)

At time \(T_M^n\), the drift-corrected clocks are synchronized within \(\frac{1}{2}\pi\) from the average drift-corrected clock if 5.16 is satisfied. It is simple to show that the average drift-corrected clock equals \(n \cdot T_S\) at this time, neglecting terms of order \(\rho_{\text{max}} \cdot \pi\). Both corrected and uncorrected clocks are then resynchronized within \(\frac{1}{2}\pi_0\) to the new average clock which has the value \(n \cdot T_S\) at time \(T_M^n\) according to lemma 6. Therefore, a drift-corrected clock is corrected by at most \(\frac{1}{2}(\pi + \pi_0)\) at a resynchronization. This correction can be made by shortening or lengthening the clock period so that a monotonous clock is obtained.

6 Accuracy

A bound is derived on the drift of the average clock. This drift is determined by the average drift of the drift-corrected clocks. Since the drift-corrected clocks advance by \(T_S\) during a synchronization round, the drift of the average clock is determined by the ratio of \(T_S\) to the actual length of a round. A bound on this ratio was already derived in section 5. The following theorem improves this ratio.

**Theorem 5** The average drift \(\rho_A^n\) of the drift-corrected clocks satisfies

\[
|\rho_A^n| < 2\rho_{\text{max}}
\]

neglecting small terms.

**Proof:**

Let \(i\) be a clock in \(C^n\). Therefore, \(i\) has been able to obtain an estimate of its difference with
the average clock at time $T_M^n - 1$. The average drift is calculated for round $n - 1$. According to equation 5.3,

$$n \cdot T_S = c_i^n(T_M^n) = c_i^n(T_M^n) - \Delta t_{i_A}^{n-1}$$

$$= c_i^n(T_M^n) + \int_{T_M^{n-1}}^{T_M^n} (1 + \rho_i(\tau))d\tau - \Delta t_{i_A}^{n-1}$$

$$= c_i^n(T_M^n) + (T_M^n - T_M^{n-1}) + \int_{T_M^{n-1}}^{T_M^n} \rho_i(\tau)d\tau - \Delta t_{i_A}^{n-1}$$  \(6.1\)

This expression can be summed over all clocks $i \in C^n$ and divided by $|C^n|$. According to lemma 6 and disregarding small terms,

$$\frac{1}{|C^n|} \sum_{i \in C^n} c_i^n(T_M^n) = (n-1)T_S + \frac{1}{|C^n|} \sum_{i \in C^n} (c_i^n(T_M^n) - c_A^n(T_M^n))$$

Therefore, when equation 6.1 is summed over all clocks $i \in C^n$ and divided by $|C^n|$, the following expression results:

$$n \cdot T_S = (n-1)T_S + \frac{1}{|C^n|} \sum_{i \in C^n} (T_M^n - T_M^{n-1}) + \frac{1}{|C^n|} \sum_{i \in C^n} \int_{T_M^{n-1}}^{T_M^n} \rho_i(\tau)d\tau$$

$$- \frac{1}{|C^n|} \sum_{i \in C^n} \Delta t_{i_A}^{n-1} + \frac{1}{|C^n|} \sum_{i \in C^n} (c_i^n(T_M^n) - c_A^n(T_M^n))$$

The second term on the right-hand side of this equation equals $T_M^n - T_M^{n-1}$. Rearranging terms, division by $T_M^n - T_M^{n-1}$ and ignoring small terms leads to

$$\rho_A^{n-1} = \frac{T_S}{T_M^n - T_M^{n-1}} - 1 = \frac{1}{|C^n|} \cdot T_S \sum_{i \in C^n} \int_{T_M^{n-1}}^{T_M^n} \rho_i(\tau)d\tau - \frac{1}{|C^n|} \cdot T_S \sum_{i \in C^n} \Delta t_{i_A}^{n-1}$$

$$+ \frac{1}{|C^n|} \cdot T_S \sum_{i \in C^n} (c_i^n(T_M^n) - c_A^n(T_M^n))$$

This proves the theorem since the first two terms on the right-hand side of this expression are bounded by $\rho_{max}$. The last term is bounded by $\pi_0/(2T_S)$ and can thus be ignored.

$\Box$

7 Conclusion

It is interesting to examine the possibilities and restrictions of the CS protocol described. The protocol periodically resynchronizes the clocks with a resynchronization precision of $12/3$ times the clock reading error. If the ratio of the required precision and the resynchronization precision is larger than approximately 3, it is possible to apply drift corrections
and a long synchronization period can be obtained. It is argued in section 5 that drift corrections cease to be useful in any protocol if this ratio is less than 3. Obviously, the protocol can not be used at all if this ratio is less than 1.

If the ratio is between 1 and 3, the length of the synchronization period is determined by the difference between the precision and the resynchronization precision and by the maximum drift assumed. For example, if \( \pi = 50 \mu s, \varepsilon = 2 \mu s, \rho_{max} = 10^{-6} \) and \( T_s' = 0, \) \( \pi_0 = 25\frac{1}{2} \mu s. \) The length of the synchronization period is then limited to approximately 12 seconds.

If the ratio exceeds 3, drift corrections can be applied. In a statically scheduled system or with probabilistic clock synchronization, the above parameters are quite realistic. The length of the synchronization period is determined in this case by the stability of the drift. Lamport and Melliar-Smith give an example in the conclusion of their paper [LaMe85] with \( \pi = 50 \mu s, \varepsilon = 1 \mu s \) and \( \rho_{max} = 0.5 \cdot 10^{-6}. \) Notice that their \( \rho \) and \( \varepsilon \) differ by a factor 2 from the \( \rho_{max} \) and \( \varepsilon \) as used here. With these parameters, \( \pi_0 = 12.5 \mu s. \) Assuming \( R_{max} = 5 \cdot 10^{-8} \) and \( T_s' = 0, \) a synchronization period of approximately 15 seconds is obtained. Without drift corrections, a synchronization period of 37 seconds is obtained. Therefore, it is useless to apply drift corrections in this case. When more stable drift rates are assumed, e.g., \( R_{max} = 10^{-6}, \) a synchronization period of up to 110 seconds is possible. These numbers should be compared to the 3.1 and 18 seconds quoted by Lamport and Melliar-Smith for two of their CFs.

The situation changes radically when larger drift rates are assumed. With \( \rho_{max} = 0.5 \cdot 10^{-4}, \) the protocol described in this paper still achieves a synchronization period of 15 seconds with the same parameters and with \( R_{max} = 5 \cdot 10^{-8}. \) The protocol of Lamport and Melliar-Smith then requires resynchronizations every 31 ms resp. 180 ms.

Many CS protocols that tolerate Byzantine behavior of the clocks require that \( N \geq 3k + 1, \) although usually authenticated versions of the protocols exist for which \( N \geq 2k + 1 \) is sufficient. However, the protocol described here always requires \( N \geq 3k + 1 \) since clocks may be removed triple-wise by the CF from the set of clocks that defines average time as described in section 4.

If the drifts are stable, very long synchronization periods can be obtained. In the unrealistic case where the drifts are exactly constant, synchronization periods of unrestricted length are possible. The reason for this is that the estimates of the relative drift become more and more precise as the synchronization period increases.

A hierarchical organization of the protocol is a subject for further investigation because, in practice, many distributed systems are too large for a completely distributed clock synchronization protocol. The overheads of a completely distributed protocol increase rapidly with the number of processors in the system. Therefore, a hierarchically organized protocol was foreseen even though this has led to less elegant results than with a non-hierarchical organization of the system. However, in a hierarchical system, clock reading errors increase with each level. The error in the estimate of the average clock at the lower level is a multiple of the estimate of the clocks at the lower level. Therefore, synchronization between different parts of the system will be less precise. In general, this should not be a problem since usually a less precise synchronization is required between different parts of
the system. Also, fewer messages should be exchanged between these parts. Therefore, the resulting inefficiencies in a statically scheduled system should not be significant.

Acknowledgements

The authors thank all staff members of the DEDOS group for their support and comments. We especially thank R.T. Gerth for his suggestions for the notation and his critical comments and H. van de Wetering for his assistance with Bresenham's algorithm.

References


[Cris89] Cristian, F., Probabilistic Clock Synchronization, IBM Almaden research center, Distributed Computing 1989-3, pp 146-158.


[LaMe85] Lamport, L and Melliar-Smith, P.M., Synchronizing clocks in the presence of faults, J. A.C.M., Vol. 32, No. 1, January 1985, pp 52-78.


In this series appeared:

91/01 D. Alstein

91/02 R.P. Nederpelt
Implication. A survey of the different logical analyses "if..., then...", p. 26.
H.C.M. de Swart

91/03 J.P. Katoen
Parallel Programs for the Recognition of P-invariant Segments, p. 16.
L.A.M. Schoenmakers

91/04 E. v.d. Sluis
Performance Analysis of VLSI Programs, p. 31.
A.F. v.d. Stappen

91/05 D. de Reus
An Implementation Model for GOOD, p. 18.

91/06 K.M. van Hee
SPECIFICATIEMETHODEN, een overzicht, p. 20.

91/07 E. Poll
CPO-models for second order lambda calculus with recursive types and subtyping, p. 49.

91/08 H. Schepers
Terminology and Paradigms for Fault Tolerance, p. 25.

91/09 W.M.P.v.d.Aalst
Interval Timed Petri Nets and their analysis, p. 53.

91/10 R.C.Backhouse
POLYNOMIAL RELATORS, p. 52.
P.J. de Bruin

91/11 R.C. Backhouse
Relational Catamorphism, p. 31.
P.J. de Bruin
G. Malcolm
E. Voermans
J. v.d. Woude

91/12 E. van der Stuijs
A note on Extensionality, p. 21.

91/13 F. Rietman
The PDB Hypermedia Package. Why and how it was built, p. 63.

91/14 P. Lemmens

91/15 A.T.M. Aerts
An example of proving attribute grammars correct: the representation of arithmetical expressions by DAGs, p. 25.
K.M. van Hee

91/16 A.J.J.M. Marcelis
Transforming Functional Database Schemes to Relational Representations, p. 21.
P.M.E. de Bra
K.M. van Hee
<table>
<thead>
<tr>
<th>ID</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>91/18</td>
<td>Rik van Geldrop</td>
<td>Transformational Query Solving, p. 35.</td>
</tr>
<tr>
<td>91/19</td>
<td>Erik Poll</td>
<td>Some categorical properties for a model for second order lambda calculus with subtyping, p. 21.</td>
</tr>
<tr>
<td>91/23</td>
<td>K.M. van Hee, L.J. Somers, M. Voorhoeve</td>
<td>Z and high level Petri nets, p. 16.</td>
</tr>
<tr>
<td>91/24</td>
<td>A.T.M. Aerts, D. de Reus</td>
<td>Formal semantics for BRM with examples, p. 25.</td>
</tr>
<tr>
<td>91/25</td>
<td>P. Zhou, J. Hooman, R. Kuiper</td>
<td>A compositional proof system for real-time systems based on explicit clock temporal logic: soundness and completeness, p. 52.</td>
</tr>
<tr>
<td>91/27</td>
<td>F. de Boer, C. Palamidessi</td>
<td>Embedding as a tool for language comparison: On the CSP hierarchy, p. 17.</td>
</tr>
<tr>
<td>91/28</td>
<td>F. de Boer</td>
<td>A compositional proof system for dynamic process creation, p. 24.</td>
</tr>
<tr>
<td>91/30</td>
<td>J.C.M. Baeten, F.W. Vaandrager</td>
<td>An Algebra for Process Creation, p. 29.</td>
</tr>
<tr>
<td>91/31</td>
<td>H. ten Eikelder</td>
<td>Some algorithms to decide the equivalence of recursive types, p. 26.</td>
</tr>
<tr>
<td>91/33</td>
<td>W. v.d. Aalst</td>
<td>The modelling and analysis of queueing systems with QNM-ExSpect, p. 23.</td>
</tr>
<tr>
<td>91/34</td>
<td>J. Coenen</td>
<td>Specifying fault tolerant programs in deontic logic, p. 15.</td>
</tr>
<tr>
<td>Year</td>
<td>Authors</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>92/01</td>
<td>J. Coenen, J. Zwiers, W.-P. de Roever</td>
<td>A note on compositional refinement, p. 27.</td>
</tr>
<tr>
<td>92/02</td>
<td>J. Coenen, J. Hooman</td>
<td>A compositional semantics for fault tolerant real-time systems, p. 18.</td>
</tr>
<tr>
<td>92/03</td>
<td>J.C.M. Baeten, J.A. Bergstra</td>
<td>Real space process algebra, p. 42.</td>
</tr>
<tr>
<td>92/05</td>
<td>J.P.H.W.v.d.Eijnde</td>
<td>Conservative fixpoint functions on a graph, p. 25.</td>
</tr>
<tr>
<td>92/06</td>
<td>J.C.M. Baeten, J.A. Bergstra</td>
<td>Discrete time process algebra, p. 45.</td>
</tr>
<tr>
<td>92/07</td>
<td>R.P. Nederpelt</td>
<td>The fine-structure of lambda calculus, p. 110.</td>
</tr>
<tr>
<td>92/10</td>
<td>P.M.P. Rambags</td>
<td>Composition and decomposition in a CPN model, p. 55.</td>
</tr>
<tr>
<td>92/13</td>
<td>F. Kamareddine</td>
<td>Set theory and nominalisation, Part II, p. 22.</td>
</tr>
<tr>
<td>92/14</td>
<td>J.C.M. Baeten</td>
<td>The total order assumption, p. 10.</td>
</tr>
<tr>
<td>92/15</td>
<td>F. Kamareddine</td>
<td>A system at the cross-roads of functional and logic programming, p. 36.</td>
</tr>
<tr>
<td>92/16</td>
<td>R.R. Seljée</td>
<td>Integrity checking in deductive databases; an exposition, p. 32.</td>
</tr>
<tr>
<td>92/17</td>
<td>W.M.P. van der Aalst</td>
<td>Interval timed coloured Petri nets and their analysis, p. 20.</td>
</tr>
<tr>
<td>92/18</td>
<td>R. Nederpelt, F. Kamareddine</td>
<td>A unified approach to Type Theory through a refined lambda-calculus, p. 30.</td>
</tr>
<tr>
<td>92/20</td>
<td>F. Kamareddine</td>
<td>Are Types for Natural Language? p. 32.</td>
</tr>
<tr>
<td>92/21</td>
<td>F. Kamareddine</td>
<td>Non well-foundedness and type freeness can unify the interpretation of functional application, p. 16.</td>
</tr>
<tr>
<td>Year</td>
<td>Authors</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>92/22</td>
<td>R. Nederpelt, F.Kamareddine, E.Klein</td>
<td>A useful lambda notation, p. 17.</td>
</tr>
<tr>
<td>92/23</td>
<td>F.Kamareddine, E.Klein</td>
<td>Nominalization, Predication and Type Containment, p. 40.</td>
</tr>
<tr>
<td>92/24</td>
<td>M.Codish, D.Dams, Eyal Yardeni</td>
<td>Bottom-up Abstract Interpretation of Logic Programs, p. 33.</td>
</tr>
<tr>
<td>92/25</td>
<td>E.Poll</td>
<td>A Programming Logic for Fo, p. 15.</td>
</tr>
<tr>
<td>93/01</td>
<td>R. van Geldrop</td>
<td>Deriving the Aho-Corasick algorithms: a case study into the synergy of programming methods, p. 36.</td>
</tr>
<tr>
<td>93/02</td>
<td>T. Verhoeff</td>
<td>A continuous version of the Prisoner's Dilemma, p. 17</td>
</tr>
<tr>
<td>93/03</td>
<td>T. Verhoeff</td>
<td>Quicksort for linked lists, p. 8.</td>
</tr>
<tr>
<td>93/04</td>
<td>E.H.L. Aarts, J.H.M. Korst, P.J. Zwietering</td>
<td>Deterministic and randomized local search, p. 78.</td>
</tr>
<tr>
<td>93/05</td>
<td>J.C.M. Baeten, C. Verhoef</td>
<td>A congruence theorem for structured operational semantics with predicates, p. 18.</td>
</tr>
<tr>
<td>93/06</td>
<td>J.P. Veltkamp</td>
<td>On the unavoidability of metastable behaviour, p. 29</td>
</tr>
<tr>
<td>93/07</td>
<td>P.D. Moerland</td>
<td>Exercises in Multiprogramming, p. 97</td>
</tr>
<tr>
<td>93/08</td>
<td>J. Verhoosel</td>
<td>A Formal Deterministic Scheduling Model for Hard Real-Time Executions in DEDOS, p. 32.</td>
</tr>
<tr>
<td>93/10</td>
<td>K.M. van Hee</td>
<td>Systems Engineering: a Formal Approach Part II: Frameworks, p. 44.</td>
</tr>
</tbody>
</table>
| 93/15 | J.C.M. Baeten  
|       | J.A. Bergstra  
|       | R.N. Bol       |
| 93/16 | H. Schepers    
|       | J. Hooman      |
| 93/17 | D. Alstein     
|       | P. van der Stok|
| 93/18 | C. Verhoef     |
| 93/19 | G-J. Houben    |
| 93/20 | F.S. de Boer   |
| 93/21 | M. Codish      
|       | D. Dams        
|       | G. Filé        
|       | M. Bruynooghe  |
| 93/22 | E. Poll        |
| 93/23 | E. de Kogel    |
| 93/24 | E. Poll and Paula Severi |
| 93/25 | H. Schepers and R. Gerth |
| 93/26 | W.M.P. van der Aalst |
| 93/27 | T. Kloks and D. Kratsch |
| 93/28 | F. Kamareddine and R. Nederpelt |
| 93/29 | R. Post and P. De Bra |
| 93/30 | J. Deogun      
|       | T. Kloks       
|       | D. Kratsch     
|       | H. Müller      |
| 93/31 | W. Körver      |
| 93/32 | H. ten Eikelder and H. van Geldrop |
| 93/33 | L. Loyens and J. Moonen |

- A Trace-Based Compositional Proof Theory for Fault Tolerant Distributed Systems, p. 27.
- Hard Real-Time Reliable Multicast in the DEDOS system, p. 19.
- A congruence theorem for structured operational semantics with predicates and negative premises, p. 22.
- The Design of an Online Help Facility for ExSpect, p. 21.
- A Typechecker for Bijective Pure Type Systems, p. 28.
- Relational Algebra and Equational Proofs, p. 23.
- Pure Type Systems with Definitions, p. 38.
- Multi-dimensional Petri nets, p. 25.
- Finding all minimal separators of a graph, p. 11.
- A Semantics for a fine λ-calculus with de Bruijn indices, p. 49.
- GOLD, a Graph Oriented Language for Databases, p. 42.
- On Vertex Ranking for Permutation and Other Graphs, p. 11.
- Derivation of delay insensitive and speed independent CMOS circuits, using directed commands and production rule sets, p. 40.
- ILIAS, a sequential language for parallel matrix computations, p. 20.
<table>
<thead>
<tr>
<th>Page</th>
<th>Authors</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>93/34</td>
<td>J.C.M. Baeten and J.A. Bergstra</td>
<td>Real Time Process Algebra with Infinitesimals</td>
<td>39</td>
</tr>
<tr>
<td>93/35</td>
<td>W. Ferrer and P. Severi</td>
<td>Abstract Reduction and Topology</td>
<td>28</td>
</tr>
<tr>
<td>93/36</td>
<td>J.C.M. Baeten and J.A. Bergstra</td>
<td>Non Interleaving Process Algebra</td>
<td>17</td>
</tr>
<tr>
<td>93/37</td>
<td>J. Brunekreef J-P. Katoen R. Koymans S. Mauw</td>
<td>Design and Analysis of Dynamic Leader Election Protocols in Broadcast Networks</td>
<td>73</td>
</tr>
<tr>
<td>93/38</td>
<td>C. Verhoef</td>
<td>A general conservative extension theorem in process algebra</td>
<td>17</td>
</tr>
<tr>
<td>93/39</td>
<td>W.P.M. Nuijten E.H.L. Aarts D.A.A. van Erp Taalman Kip K.M. van Hee</td>
<td>Job Shop Scheduling by Constraint Satisfaction</td>
<td>22</td>
</tr>
<tr>
<td>93/40</td>
<td>P.D.V. van der Stok M.M.M.P.J. Claessen D. Alstein</td>
<td>A Hierarchical Membership Protocol for Synchronous Distributed Systems</td>
<td>43</td>
</tr>
<tr>
<td>93/41</td>
<td>A. Bijlsma</td>
<td>Temporal operators viewed as predicate transformers</td>
<td>11</td>
</tr>
<tr>
<td>93/42</td>
<td>P.M.P. Rambags</td>
<td>Automatic Verification of Regular Protocols in P/T Nets</td>
<td>23</td>
</tr>
<tr>
<td>93/43</td>
<td>B.W. Watson</td>
<td>A taxonomy of finite automata construction algorithms</td>
<td>87</td>
</tr>
<tr>
<td>93/44</td>
<td>B.W. Watson</td>
<td>A taxonomy of finite automata minimization algorithms</td>
<td>23</td>
</tr>
</tbody>
</table>