# Observable Clock Synchronization

**Danny Dolev***
Hebrew University

**Rüdiger Reischuk**†
Med. Universität zu Lübeck

**Ray Strong**‡
IBM Almaden Research Center

August 1994

**Abstract**

While the synchronization of time-of-day clocks ordinarily requires information flow in both directions between the clocks, this information need not flow directly via messages. However, to take advantage of indirect information flow, we have to make a number of complex assumptions about the behavior of the communication medium. To facilitate the verification of such assumptions, we develop a relativistic theory of observable clock synchronization that does not use or depend on a Newtonian framework or real time. Within the context of this theory, we focus on the problem of estimating the time on a remote clock. We generalize the concept of **rapport** to capture the situation when such an estimate is sufficient for clock synchronization purposes. With a single property, called the Observable Drift Property, we characterize the information flow required for obtaining rapport. We compare our relativized and observable concepts with analogs based on the notion of real time in order to show that we are studying the right quantities. We then give several clock synchronization algorithms that replace round trip synchronization algorithms but use only passive message receipt in a broadcast medium. These algorithms make the message cost of synchronization independent of the number of participants. Each algorithm measures and can report its own precision and does not depend on any assumed bound on message transmission and processing time. These properties have in the past only been associated with variants of round trip

---

*Computer Science Department, Jerusalem, Israel
This research was supported in part by GIF I-207-199.6/91
†Institut für Theoretische Informatik, 23560 Lübeck, Germany
This research was supported in part by GIF I-207-199.6/91 Part of this research was done while the author visited ICSI.
‡San Jose, California 95120-6099

synchronization where the message cost is proportional to the number of participants. Verifying that a particular network can support one of these algorithms is reduced (by our results) to verifying four straightforward axioms and the Observable Drift Property.

# 1 Introduction

We consider the problem of synchronizing the clocks in a network of processors. For simplicity, we assume that the processors of the network communicate using messages on a single broadcast medium (e.g. ethernet or token ring). One standard measure of goodness of a clock synchronization scheme is the maximum simultaneous difference allowed between any two clocks. We call this difference the *precision* of the clock synchronization scheme. Note that precision is only defined in a context in which simultaneity is meaningful. The traditional approach to the analysis of clock synchronization protocols assumes what is called a *real time* model of events in the network. This model presumes a Newtonian time frame of reference that is not directly observable from within the system. Further assumptions on the system model bind observable events to bounds on real time. Since the idea of *simultaneous* events is well defined in such a system, so is precision. In this paper, we provide an alternative relativistic approach to the analysis of clock synchronization algorithms. We introduce the concept of *observable distance* to play the role of precision within our framework.

We characterize the causal relationships between events in our model with three straightforward axioms that describe the behavior of clocks, messages, and processors. Then we give a fourth *observable drift* axiom that is, in a sense, both necessary and sufficient to achieve the kind of synchronization that we discuss in this paper. Adding standard assumptions about real time, we show how to convert between observable distance and real time precision, tying our results to those in the literature [LM85, DHSS84].

A second measure of goodness of a clock synchronization protocol is *message efficiency*, the number of message transmissions per unit time required by the algorithm (to achieve a given precision). Our experience with clock synchronization protocols indicates that (among those protocols that provide a dynamic estimate of current precision) the message efficiency for a given precision generally varies at least linearly with the number of participants [DSW94]. This is because they usually involve periodic round trip communication (either one-to-all or all-to-all). In this paper, we provide new clock synchronization protocols for broadcast media for which message efficiency does not depend on the number of participants.

We use a modification of the observable drift axiom to characterize relationships between events that can be used in place of direct messages for purposes of clock synchronization. Using this characterization we prove the correctness of new message efficient clock synchronization protocols that achieve good precision by eavesdropping on communication between other processors. The new clock synchronization protocols depend on many assumptions about the nature of the broadcast medium. Viewing our correctness proof another way, we show that these assumptions can all be checked by verifying the Observable Drift Property. We have succeeded in reducing the problem of verifying whether the model assumptions are realistic for a particular network to the problem of verifying one *observable* property of the network. Thus our methods have application far beyond the proof of correctness of any particular algorithm.

Although a lot of work has been done already in the area of clock synchronization [SWL90, YM93], the kinds of questions addressed in this paper are new. Lower and upper bounds for synchronizing by broadcasting given in [HS91] do not apply here because that paper assumes that broadcast messages may have arbitrary delay, but all messages of each single broadcast arrive at the recipients within a time window of bounded size. Clock synchronization in shared memory systems, which have a different kind of broadcasting capability, has recently been studied in [DW93].

Depending on much previous work on how to synchronize clocks given that one can read remote clocks and estimate the accuracy of the reading (v. [CS90, DHSS84, LM85, M84]), we will focus in this paper on the problem of reading a remote clock. We generalize a concept of *rapport* introduced in [C89] to capture the goal of remote clock reading and depend on these well known techniques to convert a method for achieving rapport into a complete method for clock synchronization. In focussing on the problem of reading remote clocks, our approach is similar to that of [PR94]. However, there are significant differences between our approaches and goal and those of [PR94] that make the results almost orthogonal: Patt-Shamir and Rajsbaum have developed a theory within the context of real time and simultaneity and with the goal of obtaining and characterizing optimal clock synchronization as a function of an observed pattern of communication. Our theory is developed outside this context with the goal of characterizing patterns of communication that make clock synchronization possible.

## 2 A Relativistic Model for Synchronization

We want to talk about synchronization of distributed systems without referring to real time or assuming simultaneity because these concepts refer to things that are not observable. Our purpose is to describe real systems using observable concepts so that our assumptions about how they behave can be checked.

A processor (more accurately, a singly threaded software process) can only observe its local clock reading and the sequentially ordered events happening at its site. It cannot read the clocks of other processors directly. Whether events at different processors happen simultaneously, or one happens a little before or a little after the other, could only be observed from outside the system by something that could observe both sites directly.

We use the following notation. Single lower case letters $\boldsymbol{a, b, c, d, \ldots}$ represent events. Events are partitioned into four different types:

$$\mathbf{Send,\ \ Rcv,\ \ Buf,\ \ Clock,}$$

which describe the subset of *send, receive, buffer empty,* resp. *clock events.* The events *send* and *receive* have the usual meaning, where *send* may initiate a point-to-point message transmission or a broadcast to all participants. *buffer empty* means that a processor checks the contents of its incoming message buffer, and notices that it is empty. Finally, *Clock* means that a processor reads its local hardware clock. The list of event types is by no means exhaustive. We simply limit the scope of this paper to these four types.

The processors of a distributed system are represented as upper case letters $\boldsymbol{P}, \boldsymbol{Q}, \ldots$. They also partition the set of events according to their location. For example,

$$\boldsymbol{a} \in \boldsymbol{P} \quad := \quad \text{event } a \text{ happens at processor } P.$$

We assume that each event occurs at a unique location (processors do not share events), and use abbreviations like

$$\boldsymbol{Clock_P} \quad := \quad Clock \ \cap \ P \ .$$

The events at each processor are totally ordered by the relation $\xrightarrow{\textbf{Seq}}$.

Since we are focussing only on the problem of reading a remote clock, we make the simplifying assumption that there is only one clock at each location. For each location $P$ there is a function

$$\boldsymbol{T_P} \ : \ Clock_P \rightarrow \mathbb{R}$$

from the clock events at $P$ to real numbers representing the reading of the clock at $P$. The union of all such functions is called $\boldsymbol{T}$ and is itself a function from $Clock$ to real numbers.

In addition to $\xrightarrow{\text{Seq}}$ we have a primitive binary relation $\xrightarrow{\textbf{Bcst}}$ :

$$\begin{array}{lll} a \xrightarrow{\text{Bcst}} b & := & a \in Send, \quad \text{and} \\ & & b \in Rcv \quad \text{is a corresponding receive event at another location.} \end{array}$$

Note that for a *send*-event $a$ there may be many *receive*-events that relate to $a$ by $\xrightarrow{\text{Bcst}}$ depending on whether $a$ was a point-to-point message transmission or a broadcast that addresses several processors. We do not require that message transmission be reliable. In particular, there may be *send*-events without any *receive*-events.

Define $\xrightarrow{\textbf{Caus}}$ as the transitive closure of the union of $\xrightarrow{\text{Seq}}$ and $\xrightarrow{\text{Bcst}}$. This relation is identical to Lamport's causality relation [L78].

**Definition 1** *A* **run** $\boldsymbol{\Phi}$ *of a given distributed system is a set of events with relations* $\xrightarrow{\text{Seq}}$ *and* $\xrightarrow{\text{Bcst}}$ *and a local timing function* $T$ *that satisfies the following axioms.*

**Axiom 1 Clock Advance Axiom:**
*If* $a \xrightarrow{\text{Seq}} b$ *then* $T(b) > T(a)$.

**Axiom 2 Broadcast Axiom:**
*If* $b \in Rcv$ *then there is a unique* $a \in Send$ *such that* $a \xrightarrow{\text{Bcst}} b$.

**Axiom 3 Causality Axiom:**
$\xrightarrow{\text{Caus}}$ *imposes a strict partial ordering of events (without cycles).*

**Axiom 4 Observable Drift Axiom:**
*(see Figure 1) There exists a global real constant* $\rho_0 > 0$ *(the maximum drift rate) such that if* $a, b, c,$ *and* $d$ *are clock events and if* $a \xrightarrow{\text{Caus}} b$, $b \xrightarrow{\text{Seq}} c$, $a \xrightarrow{\text{Seq}} d$, *and* $c \xrightarrow{\text{Caus}} d$ *then*

$$T(c) - T(b) \;<\; (1 + \rho_0)\,[T(d) - T(a)]\ .$$

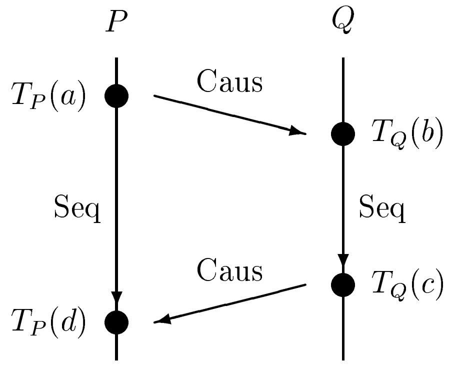


Figure 1: Observable Drift Axiom: $T_Q(c) - T_Q(b) \;<\; (1 + \rho_0)\,[T_P(d) - T_P(a)]$

The first three axioms should be fairly obvious and natural. Observable Drift is the primary focus of our research. The real number $\rho_0$ is a bound on the rate at which clocks may drift apart (cf. [DHSS84]).

**Definition 2** *Let* $\Phi$ *be a run and* $E$ *a subset of events of* $\Phi$ *. Then* $\texttt{cut}_{\boldsymbol{\Phi}}\boldsymbol{(E)}$ *is the partial run obtained from* $\Phi$ *by taking* $E$ *and those events that causally precede an event in* $E$ *.*

When no confusion will result, we will omit explicit mention of the run.

We are now in a position to define what it means to obtain observable clock synchronization. Intuitively, clocks are synchronized to within a given precision if no experiment using messages can demonstrate that they are farther apart. Instead of trying somehow to encompass all possible experiments, the following will suffice.

**Definition 3** *(see Figure 2)*
*The clocks at locations* $P$ *and* $Q$ *are said to be within* **observable distance** $\boldsymbol{\gamma}$ *of each other at the time of clock event* $a$ *at* $P$ *if, for any clock events* $b$ *and* $c$ *at* $Q$ *with* $b \xrightarrow{\text{Caus}} a \xrightarrow{\text{Caus}} c$ *,*

$$T_Q(b) - \gamma \;<\; T_P(a) \;<\; T_Q(c) + \gamma\ .$$

*Formally, in this situation, we will write*

$$OD_a(T_P, T_Q) \leq \gamma.$$

However, a bound on observable distance at the time of one clock event is not enough information to synchronize clocks. The bound could have been accidentally satisfied

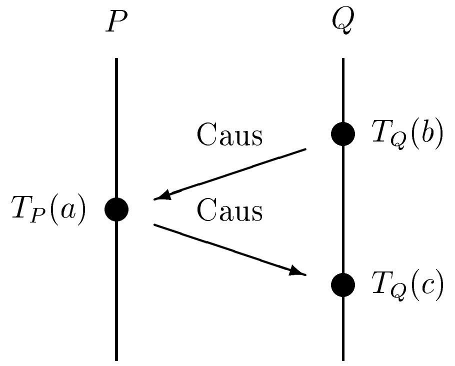


Figure 2:
Observable Distance: $T_Q(b) - \gamma \ < \ T_P(a) \ < \ T_Q(c) + \gamma$

because there were no nearby events that were causally related to the event in question. Instead, in order to be sure that the remote clock has been accurately read, we use a notion similar to rapport in [C89].

Let $\boldsymbol{\rho}$ be an arbitrary real constant at least as large as the bound $\rho_0$ on the drift given by the Observable Drift Axiom.

**Definition 4** *Clocks $T_P$ and $T_Q$ are said to be within* **observable rapport $\boldsymbol{\pi}$** *with respect to drift $\rho$ in run $\Phi$ after some event $a$ at $P$ if, in any run that contains $\mathtt{cut}_\Phi(\{a\})$ and for every event $d$ with $a \xrightarrow{\text{Seq}} d$,*

$$OD_d(T_P, T_Q) \ \leq \ \pi + \ \rho\,(T_P(d) - T_P(a)) \ .$$

*This situation will be denoted by*

$$OR_{a,\,\rho}\,(T_P, T_Q) \ \leq \ \pi \ .$$

The techniques for converting the ability to produce rapport into a clock synchronization algorithm are well known (cf. [C89, CS90, DHSS84, LM85]); so we will assume that we can achieve **observable clock synchronization** if we can achieve observable rapport.

Since simultaneity is not observable, we need not restrict attention only to the Lamport causality relation. In fact it will be very helpful to define relations that satisfy much weaker properties but still manage to order events sufficiently for clock synchronization purposes. We have discovered that the Observable Drift Property below exactly characterizes such relations.

**Definition 5** *(see Figure 3) Let $\xrightarrow{\text{Rel}}$ be an arbitrary relation between events and let $\rho$ be a positive real $\geq \rho_0$ . We say that $\xrightarrow{\text{Rel}}$ has the* **Observable Drift Property** *with respect to $\rho$ if, in any run, for all clock events $a, b, c, d$ , $a \xrightarrow{\text{Rel}} b$ , $b \xrightarrow{\text{Seq}} c$ , $c \xrightarrow{\text{Caus}} d$ , and $a \xrightarrow{\text{Seq}} d$ implies*

$$(1 + \ \rho\,)\ [T(d) - T(a)] \ > \ T(c) - T(b) \ .$$

*In this situation we write*

$$ODrift(\xrightarrow{\text{Rel}}) \leq \rho\,.$$

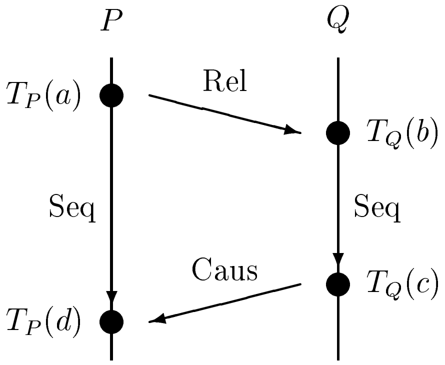


Figure 3: Observable Drift Property: $T_Q(c) - T_Q(b) \;<\; (1+\rho)\,[T_P(d) - T_P(a)]$

**Definition 6** *(see Figure 4)*
*Clock events $a, b, c$ form a* **round trip with respect to a relation** $\xrightarrow{\textbf{Rel}}$ *if* $a \xrightarrow{\text{Seq}} c$, $a \xrightarrow{\text{Rel}} b$, *and* $b \xrightarrow{\text{Caus}} c$.

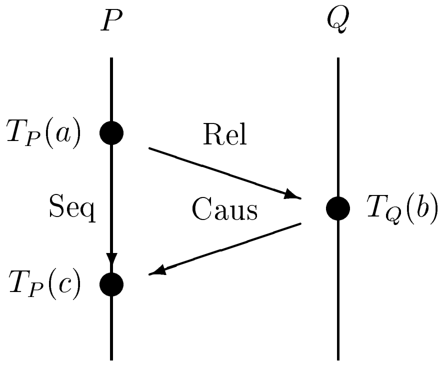


Figure 4: Round Trip with respect to relation $\xrightarrow{\text{Rel}}$

## 3 Observable Clock Synchronization Realized

In this section we will show that a synchronization method based on the Observable Drift Property will actually achieve the goal (observable rapport). Its properties are similar to other clock synchronization schemes based on round trips (e.g. [M84, C89]). But it does not require a physical round trip. Thus it can be faster or more message efficient depending on the relation $\xrightarrow{\text{Rel}}$ that is chosen.

**Theorem 1** *Let the Observable Drift Property hold for* $\xrightarrow{\text{Rel}}$ *with* $ODrift(\xrightarrow{\text{Rel}}) \leq \rho$. *Let events $a, b, c$ form a round trip with respect to* $\xrightarrow{\text{Rel}}$, *where $a$ and $c$ happen at* *processor $P$ and $b$ at $Q$. Let $\pi = max(T_P(c) - T_Q(b), T_Q(b) - T_P(a)) + \rho\,(T_P(c) - T_P(a))$. Then*

$$OR_{c,\rho}\,(T_P, T_Q) \leq \pi \ .$$

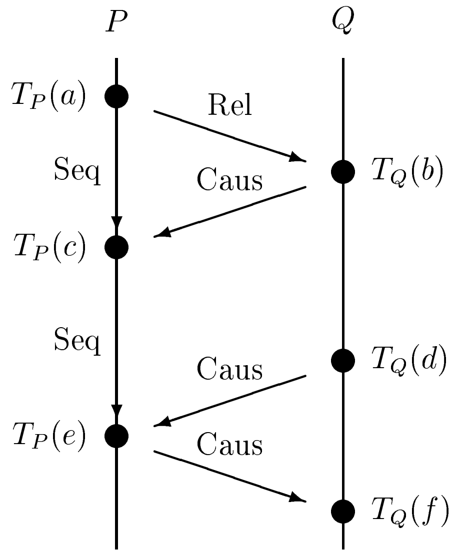


Figure 5:
Synchronization within observable distance based on the Observable Drift Property:
at event $e$ the observable distance is bounded by the observable rapport and the drift.

*Proof:* Let $e$ be any clock event with $c \xrightarrow{\text{Seq}} e$. Let $d$ and $f$ be any clock events at $Q$ with $d \xrightarrow{\text{Caus}} e \xrightarrow{\text{Caus}} f$. Let $\gamma = \pi + \rho\,(T(e) - T(c))$. Note that

$$\gamma = max[T(c) - T(b), T(b) - T(a)] + \rho\,[T(e) - T(a)].$$

We must show that $T(d) - \gamma < T(e) < T(f) + \gamma$. By the Clock Advance Axiom $T(b) - \gamma < T(c) < T(e)$ so it suffices to consider events $d$ with $b \xrightarrow{\text{Seq}} d$. Now by the Observable Drift Property for events $a, b, c, e$,

$$\begin{aligned} T(d) &< (1+\rho)\,[T(e) - T(a)] + T(b) \\ &= T(e) + \rho\,[T(e) - T(a)] + T(b) - T(a) \\ &\leq T(e) + \gamma \ . \end{aligned}$$

Note that by the Causality Axiom there cannot be events $f$ that follow $e$ with respect to $\xrightarrow{\text{Caus}}$ and are at or precede $b$. To prove the second half of the inequality, we distinguish two cases.

1.) $T(e) - T(a) \leq T(f) - T(b)$: Then

$$T(e) \ \leq \ T(f) + T(a) - T(b) \ < \ T(f) + T(c) - T(b) \ < \ T(f) + \gamma \ .$$

2.) $T(f) - T(b) < T(e) - T(a)$:
Then, using the Observable Drift Axiom for the events $b, c, e, f$ and the property $\rho_0 \le \rho$,

$$\begin{array}{rcl} T(e) & < & (1+\rho_0)\,[T(f)-T(b)] + T(c) \;=\; T(f) + \rho_0\,[T(f)-T(b)] + T(c) - T(b) \\ & < & T(f) + \rho_0\,[T(e)-T(a)] + T(c) - T(b) \;\le\; T(f) + \gamma\,. \end{array}$$

In either case, $T(e) < T(f) + \gamma$, which completes the proof. ∎

The usual clock synchronization technique adjusts one of the clocks to minimize rapport after the hypothesized round trip of our theorem. This minimum rapport occurs if $T_Q(b) = \frac{T_P(a)+T_P(c)}{2}$. Thus we have $T_P(c) - T_Q(b) = T_Q(b) - T_P(a) = \frac{T_P(c)-T_P(a)}{2}$ and $\pi = (\frac{1}{2} + \rho)(T_P(c) - T_P(a))$.

Observable distance is almost symmetric with respect to the two processors involved, although it is defined with respect to an event happening at one of the processors. Observable rapport is even more symmetric as demonstrated by the following theorem, which says that the same rapport bound (actually, a slightly tighter one) holds from the point of view of $Q$.

**Theorem 2** *Under the hypotheses of Theorem 1,*

$$OR_{b,\rho}(T_P, T_Q) \le max(T_P(c) - T_Q(b), T_Q(b) - T_P(a)).$$

*Proof:* Let $e$ be any clock event with $b \xrightarrow{\text{Seq}} e$. Let $d$ and $f$ be any events at $P$ with $d \xrightarrow{\text{Caus}} e \xrightarrow{\text{Caus}} f$. Let $\gamma = max[T(c)-T(b), T(b)-T(a)] + \rho\,[T(e)-T(b)]$. We must show $T(d) - \gamma < T(e) < T(f) + \gamma$. By the Clock Advance Axiom $T(c) - \gamma < T(b) < T(e)$ so it suffices to consider events $d$ with $c \xrightarrow{\text{Seq}} d$. Applying the Observable Drift Axiom and the assumption that $\rho_0 \le \rho$, we have $T(d) - T(c) < (1 + \rho_0)[T(e) - T(b)]$, so that $T(d) - \gamma < T(e)$.

By the Causality Axiom we must have $a \xrightarrow{\text{Seq}} f$. To prove the second half of the inequality we distinguish two cases.

1.) $T(e) - T(b) \le T(f) - T(a)$: Then $T(e) \le T(f) + T(b) - T(a) < T(f) + \gamma$.

2.) $T(e) - T(b) > T(f) - T(a)$: Then by the Observable Drift Property,

$$\begin{array}{rcl} T(e) & < & T(b) + (1+\rho)[T(f) - T(a)] \\ & < & T(f) + T(b) - T(a) + \rho[T(e) - T(b)] \;\le\; T(f) + \gamma \end{array}$$

In either case $T(e) < T(f) + \gamma$, which completes the proof. ∎

## 4 Observable versus Realtime Synchronization

Assuming that observable distance is the correct measure for the closeness of clocks, we have shown how to read a remote clock and thus how to achieve clock synchronization.

However, observable distance is not an exact analog to the notions of precision or error that have been defined previously within the real time framework. Notice that the observable distance between two clocks can be negative. In this section, we explore the relationship between our observable framework and the real time framework and show that observable distance is the appropriate measure for clock synchronization purposes. In particular we will show how to convert between real time precision and observable distance.

For this purpose we need to make explicit assumptions about real time in addition to our original four axioms.

We postulate a new universal clock $T_R$ that measures *real time* and is defined at every event in every run. We assume that the Observable Drift Axiom extends to $T_R$ in the sense that, if $a$ and $b$ are clock events with $a \xrightarrow{\text{Seq}} b$, then $T(b) - T(a) < (1 + \rho_0)[T_R(b) - T_R(a)]$ and $T_R(b) - T_R(a) < (1 + \rho_0)[T(b) - T(a)]$. Moreover, real time acts like any other clock, so that, if $a$, $b$, $c$, and $d$ are clock events and $a \xrightarrow{\text{Caus}} b$, $b \xrightarrow{\text{Seq}} c$, $c \xrightarrow{\text{Caus}} d$, and $a \xrightarrow{\text{Seq}} d$, then $T(c) - T(b) < (1 + \rho_0)[T_R(d) - T_R(a)]$ and $T_R(c) - T_R(b) < (1 + \rho_0)[T(d) - T(a)]$.

We extend our local timing functions to provide an easy way to express simultaneity: for any pair of locations $P, Q$, for each clock event $p \in P$, we assume a *shadow* clock event $p_Q$ in the domain of $T_Q$ with $T_R(p) = T_R(p_Q)$.

In this section we assume that there is no minimum granularity to the range of $T$. This assumption that the range of $T$ is dense in the reals keeps us from having to introduce additive terms to the extended Drift Axiom to deal with the smallest observable clock difference. We only make this simplifying assumption when we compare with theories based on real time. It is not needed in the previous section.

We assume that there is a minimum possible real time duration $\tau_0 > 0$ between any two clock events at different locations that are related by $\xrightarrow{\text{Caus}}$, that is, if $a \xrightarrow{\text{Caus}} b$ and $a$ and $b$ are clock events at different locations, then $|T_R(a) - T_R(b)| \geq \tau_0$.

Finally, we assume a system reaction time bound $\lambda$ that is a sufficient real time duration after any event to allow for the possibility of beginning a minimum real time duration causal relationship. More formally, we assume that for any run $\Phi$ and for any $\text{cut}_\Phi(\{e\})$, there is another possible run $\Psi$ that contains all the events and shadow events of $\text{cut}_\Phi(\{e\})$ with the same local and real timings and contains clock events $p$ and $q$ with $T_R(p) \leq T_R(e) + \lambda$, $p \xrightarrow{\text{Caus}} q$, and $T_R(q) = T_R(p) + \tau_0$. Note that we only assume the possibility of a minimum real time duration causal relationship, not that we can force one to happen.

**Definition 7** *The* **real time precision** *between clocks* $T_P, T_Q$ *at the time of clock event* $p \in P$ *is defined by*

$$RD_p(T_P, T_Q) \ := \ |T_P(p) - T_Q(p_Q)| \ .$$

Using the concept of real time precision, we can define a real time analog to observable rapport, which we call real time rapport:

**Definition 8** *We say clocks* $T_P, T_Q$ *have* **real time rapport** *bound* $\pi$ *with drift* $\rho$ *as of the time of event* $p \in P$ *(denoted* $RR_{p,\rho}(T_P, T_Q) \leq \pi$ *) if, for any event* $e$ *with* $p \xrightarrow{\text{Seq}} e$ *,*

$$RD_e(T_P, T_Q) \;\leq\; \pi + \rho\ (T(e) - T(p))\ .$$

In this paper we offer the following observations relating these real time concepts with their observable analogs.

**Theorem 3** *If* $RR_{p,\rho}(T_P, T_Q) \leq \epsilon$ *, then* $OR_{p,\rho}(T_P, T_Q) \leq \epsilon - \frac{\tau_0}{1+\rho_0} < \epsilon - \tau_0 + \rho_0\, \tau_0$ *.*

*Proof:* Let $e$ be a clock event with $p \xrightarrow{\text{Seq}} e$ and let $d$ and $f$ be clock events at $Q$ with $d \xrightarrow{\text{Caus}} e \xrightarrow{\text{Caus}} f$. By hypothesis we have $T(e_Q) \leq T(e) + \epsilon + \rho\,(T(e) - T(p))$. Also, $(1+\rho_0)(T(e_Q) - T(d)) > T_R(e_Q) - T_R(d) \geq \tau_0$. Thus $T(d) < T(e) + \epsilon - \frac{\tau_0}{1+\rho_0} + \rho\,(T(e) - T(p))$. Similarly, $T(e) \leq T(e_Q) + \epsilon + \rho\,(T(e) - T(p))$ and $(1+\rho_0)(T(f) - T(e_Q)) > T_R(f) - T_R(e_q) \geq \tau_0$. Thus $T(e) < T(f) + \epsilon - \frac{\tau_0}{1+\rho_0} + \rho\,(T(e) - T(p))$. The final inequality follows from the fact that $\rho_0 > 0$ so that $\frac{1}{1+\rho_0} > 1 - \rho_0$ . ∎

**Theorem 4** *If* $OR_{a,\rho}(T_P, T_Q) \leq \pi$ *, then* $RR_{a,\rho}(T_P, T_Q) \leq \pi + \tau_0 + \rho_0\, \tau_0 + 2\,\rho_0\,\lambda + \rho\,(1+\rho_0)(\lambda + \tau_0)$ *.*

*Proof:* Let $e$ be a clock event with $a \xrightarrow{\text{Seq}} e$. Let $\gamma = \pi + \tau_0 + \rho_0\,\tau_0 + 2\,\rho_0\,\lambda + \rho\,(\lambda + \tau_0) + \rho\,[T(e) - T(a)]$. We must show that $T(e) \leq T(e_Q) + \gamma$ and $T(e_Q) \leq T(e) + \gamma$.

First we will establish $T(e) < T(e_Q) + \gamma$. By our real time assumptions there is a run containing events $a$, $e$, and $e_Q$ with the same local and real timings in which we still have $OR_{a,\rho}(T_P, T_Q) \leq \pi$ and there is a minimum duration causal relationship beginning at $p$ and ending at $q$ with $e \xrightarrow{\text{Seq}} p$ and $T_R(p) - T_R(e) \leq \lambda$. Let $\nu = T_R(p) - T_R(e) = T_R(p_Q) - T_R(e_Q)$. By our real time Drift assumption, $T(e) < T(p) - \frac{\nu}{1+\rho_0}$. By the assumed observable rapport, $T(p) \leq T(q) + \pi + \rho\,[T(p) - T(a)]$. Since $T_R(q) - T_R(p) = T_R(q) - T_R(p_Q) = \tau_0$, our real time Drift assumption implies that $T(q) < T(p_Q) + (1+\rho_0)\,\tau_0$. Finally, by the real time Drift assumption, $T(p_Q) < T(e_Q) + (1+\rho_0)\nu$. Resolving these inequalities yields $T(e) < T(e_Q) - \frac{\nu}{1+\rho_0} + \pi + \rho\,[T(p) - T(a)] + (1+\rho_0)\,\tau_0 + (1+\rho_0)\nu$. Using $1 + \rho_0 - \frac{1}{1+\rho_0} < 2\,\rho_0$ and $\nu \leq \lambda$, we have $T(e) < T(e_Q) + \pi + \tau_0 + \rho_0\,\tau_0 + 2\,\rho_0\,\lambda + \rho\,[T(p) - T(e)] + \rho\,[T(e) - T(a)]$ In this case the term $\rho\,[T(p) - T(e)] \leq \rho\,(1+\rho_0)\lambda$. Thus $T(e) < T(e_Q) + \gamma$.

To prove the other half of the inequality, we use a symmetric construction where the minimum real time duration causal relation begins at $q$ and ends at $p$. The analysis is symmetric, except that $\rho\,[T(p) - T(e)] \leq \rho\,(1+\rho_0)(\lambda + \tau_0)$. Thus $T(e_Q) < T(e) + \gamma$, which completes the proof. ∎

Ignoring drift terms (since $\rho_0$ and $\rho$ are presumably very small) we have shown that the real time rapport is approximately the observable rapport plus $\tau_0$ .

**Corollary 1**

$$RR \approx OR + \tau_0 \ .$$

## 5 Examples of Indirect Causality and Synchronization Protocols Based on such Relations

A basic protocol for synchronizing a slave clock to a master clock and maintaining an observable distance of approximately $\theta$ (when possible) given a method for producing a round trip using a $\xrightarrow{\text{Rel}}$ that satisfies the Observable Drift Property with drift rate $\rho$ is as follows:

Slave Protocol:
$D \leftarrow \infty$
$X \leftarrow -\infty$
Do Forever
  Obtain Round Trip Times $T(a), T(b), T(c)$
  $R \leftarrow (\frac{1}{2} + \rho)(T(c) - T(a))$
  If $R < D + \rho(T(c) - X)$ then begin
    Adjust local clock by $T(b) - \frac{T(a)+T(c)}{2}$
    $D \leftarrow R$
    $X \leftarrow T(c)$
  End
  Wait Timeout$(X, D, \theta, \rho)$
End

Here the Timeout can be set so that, if $D$ is sufficiently less than $\theta$ then the $T =$ Timeout$(X, D, \theta, \rho)$ will satisfy $D + \rho(T - X) < \theta$ but with sufficient time allowed to obtain a new round trip; and otherwise, it can be set to some arbitrary period. The details of such settings, methods for amortizing the time adjustment of the local clock to maintain a smooth time service, and methods for tolerating message or clock failures are beyond the scope of this paper but have been studied in [CS90], for example.

In this section we will explore some appropriate candidate relations $\xrightarrow{\text{Rel}}$ with the property that the round trip involves no communication from the slave. In this way we make the message overhead independent of the number of slaves that are being synchronized to the same master.

Our first example is illustrated in figure 6. This simplest kind of indirect causality is obtained by allowing a sufficiently long delay between messages sent by the master.

Formally we say $a \xrightarrow{\text{IndDelay}} d$ holds between events $a \in S$ and $d \in M$ if there exist events $b \in Buf_S$ , $c \in Clock_M$ , and $e \in Clock_S$ such that $a \xrightarrow{\text{Seq}} b \xrightarrow{\text{Seq}} e$ , $c \xrightarrow{\text{Seq}} d$ ,

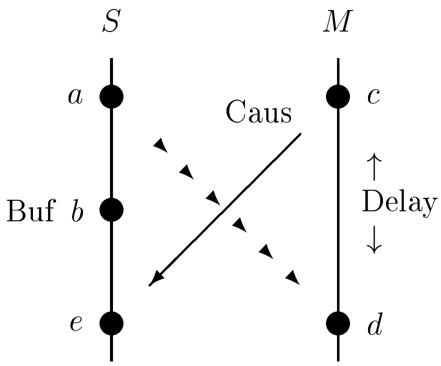


Figure 6: $a \overset{\text{IndDelay}}{\longrightarrow} d$

$c \overset{\text{Caus}}{\longrightarrow} e$, and $T(d) - T(c) \geq Delay$. Thus $Delay$ parameterizes a whole family of candidates $\overset{\text{IndDelay}}{\longrightarrow}$.

Recall that a $Buf$ event means that the message buffers at $S$ are empty. The intuitive idea is that if $a$ occurs before such a buffer empty event and then S receives a message from M that was sent at $c$, then $d$ occurs "after" $a$, provided we allow sufficient time between $c$ and $d$. Note that we are not counting on any minimum message transmission delay greater than $0$ or on any maximum message transmission and processing delay. We are simply counting on the impossibility of both messages (the one sent after $c$ and another one sent after $d$ ) being in transit at the time of event $b$. A sufficient $Delay$ to guarantee that only one message can be in transit is clearly a property of the network. However, it is not clear how to verify this property directly, since we cannot observe the two locations simultaneously. What we have shown in this paper is that it is sufficient to verify the Observable Drift Property for $\overset{\text{IndDelay}}{\longrightarrow}$ with a given $Delay$. Relative to this verification (and that of the axioms) we can guarantee that a clock synchronization protocol based on a round trip of the form $a \overset{\text{IndDelay}}{\longrightarrow} d \overset{\text{Caus}}{\longrightarrow} f$ will synchronize the slave to the master as outlined above. Moreover, this round trip involves no active transmission of messages by $S$. Thus the message overhead of such a protocol is independent of the number of slaves that are being synchronized to a given master. (Here we are ignoring questions of message loss and other fault tolerance issues.)

The family of $\overset{\text{IndDelay}}{\longrightarrow}$ based protocols provides a wide range of choices; but each choice is not very robust with respect to changes in the characteristics of the network. If the load changes, we may have to change $Delay$ in ways that are not easily predictable from the observations necessary to verify Observable Drift. A more robust example of a suitable relation $\overset{\text{Rel}}{\longrightarrow}$ is obtained by eavesdropping of a slave $S$ on round trip message exchanges between the master $M$ and one other location $A$. Figure 7 illustrates the requirements to relate two events $a \in S$ and $e \in M$ by an indirect causality relation $\overset{\text{IndEaves}}{\longrightarrow}$.

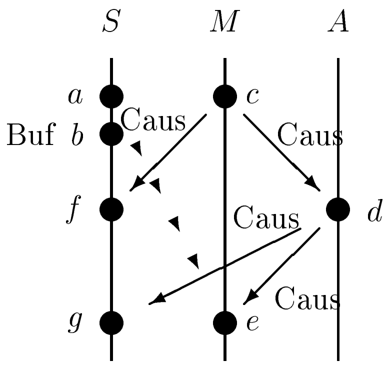


Figure 7: $a \xrightarrow{\text{IndEaves}} e$

Formally we say $a \xrightarrow{\text{IndEaves}} e$ for $a \in Clock_S$ and $e \in Clock_M$ if there exist $b \in Buf_S$, $f, g \in Clock_S$, $c \in Clock_M$, and $d \in Clock_A$ such that $a \xrightarrow{\text{Seq}} b \xrightarrow{\text{Seq}} f \xrightarrow{\text{Seq}} g$, $c \xrightarrow{\text{Seq}} e$, $c \xrightarrow{\text{Caus}} f$, $c \xrightarrow{\text{Caus}} d \xrightarrow{\text{Caus}} e$, and $d \xrightarrow{\text{Caus}} g$. Intuitively, the fact that the buffers of $S$ are empty at $b$ means that the delay of round trip $c \xrightarrow{\text{Caus}} d \xrightarrow{\text{Caus}} e$ is sufficient to insure that $e$ occurs "after" $a$ in some networks. Thus we can use $a \xrightarrow{\text{IndEaves}} e$ as the first leg of a round trip $a \xrightarrow{\text{IndEaves}} e \xrightarrow{\text{Caus}} h$ with $h \in Clock_S$ without requiring $S$ to send any message. We need only check that $\xrightarrow{\text{IndEaves}}$ satisfies the Observable Drift Property.

Again, we can produce a whole family of candidate relations based on $\xrightarrow{\text{IndEaves}}$ by increasing the number of round trips between $M$ and $A$.

## 6 Conclusion

We have provided a relativistic theory of clock synchronization, which depends on neither real time nor simultaneity. In the context of this theory we have characterized a sufficient precursor to clock synchronization called rapport. Rapport is achieved by reading a remote clock in such a way that the reading provides a bound on the rate at which it becomes less accurate. We have shown that rapport can be achieved even though some of the participants passively receive and do not send messages.

In order to relate the concepts of our theory (observable rapport and observable distance) to previous work, we introduced concepts of simultaneity and real time into our framework and showed that the analog real time rapport is approximately the observable rapport plus the minimum real time transmission and processing delay, the approximation depending on ignoring drift terms.

Finally, we have provided two families of candidate protocols for achieving rapport between a slave and a master without requiring the slave to send any messages. The main contribution of our work is that we have reduced the set of assumptions that need to be checked on a given network (to see whether one of the corresponding family of protocols will correctly synchronize clocks) to verifying the axioms and the Observable Drift Property.

## 7 Acknowledgements

The authors would like to thank Joe Halpern for suggesting the term Observable Drift.

## References

[AHR93] H. Attiya, A. Herzberg, S. Rajsbaum, *Optimal Clock Synchronization under Different Delay Assumptions,* Proc. 12. PODC, 1993, 109-120.

[C89] F. Christian, *Probabilistic Clock Synchronization,* Distr. Computing 3, 1989, 146-158.

[CS90] F. Cristian and F. Shmuck, *Continuous Clock Amortization Need Not Affect the Precision of a Clock Synchronization Algorithm,* Proc. 9 PODC, 133-143, 1990.

[DHS86] D. Dolev, J. Halpern, R. Strong, *On the Possibility and Impossibility of Achieving Clock Synchronization,* J. CSS 32, 1986, 230-250.

[DHSS84] D. Dolev, J. Halpern, B. Simons, and R. Strong, *Dynamic Fault-Tolerant Clock Synchronization,* JACM, in print; a previous version appeared in Proc. 3. PoDC, 1984, 89-102.

[DSW94] D. Dolev, R. Strong, and E. Wimmers, *Experience with RAPID Prototypes*, IBM Research Report RJ9692, 1994.

[DW93] S. Dolev, J. Welch, *Wait-Free Clock Synchronization,* Proc. 12. PODC, 1993, 97-108.

[HMM85] J. Halpern, N. Megiddo, A. Munshi, *Optimal Precision in the Presence of Uncertainty,* J. Complexity 1, 1985, 170-196.

[HS91] J. Halpern, I. Suzuki, *Clock Synchronization and the Power of Broadcasting,* Distr. Computing 5, 1991, 73-82.

[L78] L. Lamport, *Time, Clocks and the Ordering of Events in Distributed Systems,* C. ACM 21, 1978, 558-565.

[LM85] L. Lamport and M. Melliar-Smith, *Synchronizing Clocks in the Presence of Faults,* Journal of the ACM, 32:1, 52-78, 1985.

[LL84] J. Lundelius, N. Lynch, *An Upper and Lower Bound for Clock Synchronization,* Inf. & Control 62, 1984, 190-204.

[M84] K. Marzullo, *Maintaining time in a distributed system: An example of a loosely coupled distributed service,* Ph.D. dissertation, Stanford University, 1984.

[PR94] B. Patt-Shamir and S. Rajsbaum, *A Theory of Clock Synchronization,* to appear in Proc. 26th Symp. on Theory of Computation, 1994.

[SWL90] B. Simons, J. Welch, and N. Lynch, *An Overview of Clock Synchronization,* in B. Simons and A. Spector eds., Fault-Tolerant Distributed Computing, LNCS 448, Springer-Verlag, 84-96, 1990.

[YM93] Z. Yang and A. Marsland eds., *Global States and Time in Distributed Systems*, IEEE Computer Society Press, 1993.