Optimal and Global Time Synchronization in Sensornets

Richard Karp Jeremy Elson Deborah Estrin Scott Shenker

Abstract

Time synchronization is necessary in many distributed systems, but achieving synchronization in sensornets, which combine stringent precision requirements with severe resource constraints, is particularly challenging. This challenge has been met by the recent Reference-Broadcast Synchronization (RBS) proposal, which provides on-demand pairwise synchronization with low overhead and high precision. In this paper we introduce a model of the basic RBS synchronization paradigm that treats clock offset and clock skew on different time scales. Within the context of this model we derive the optimally precise and globally consistent clock synchronization. This approach can be used with any synchronization paradigm that produces pairwise synchronizations with independent errors.

1 Introduction

Many traditional distributed systems employ time synchronization to improve the consistency of data and the correctness of algorithms. Time synchronization plays an even more central role in sensornets, whose deeply distributed nature necessitates fine-grained coordination among nodes. Precise time synchronization is needed for a variety of sensornet tasks such as sensor data fusion, TDMA scheduling, localization, coordinated actuation, and power-saving duty cycling. Some of these tasks require synchronization precision measured in μsecs, which is far more stringent than the precision required in traditional distributed systems. Moreover, the severe power limitations endemic in sensornets constrain the resources they can devote to synchronization. Thus, sensornet time synchronization must be both more precise, and more energy-frugal, than traditional time synchronization methods.

The recent *Reference-Broadcast Synchronization* (RBS) design meets these two exacting objectives [5] by producing on-demand pairwise synchronization with low overhead and high precision. In this paper we introduce a simplified model of the RBS synchronization paradigm in which we treat clock skew and clock offset on different time scales (as opposed to RBS, which treats them simultaneously). We then ask, for offsets and skews separately, how to produce pairwise synchronizations that are both optimally precise and globally consistent.

To make our objectives and results clear, we first briefly review RBS. The RBS approach is based on several observations about the nature of radio communication in sensornets:

- *Sensornet communications are broadcast locally, rather than sent point-to-point as in traditional distributed systems.* This means that each transmission can reach several receivers.
- *Sensornet radio ranges are short compared to the product of the speed of light times the synchronization precision.* Thus, each broadcast is seen essentially simultaneously by the receivers within range.[1]
- *Delays between time-stamping and sending a packet are significantly more variable than the delays between receipt and time-stamping a packet.* Thus, estimates of when a packet is sent are far noisier than estimates of when it is received.

[1]Note that the extreme synchronization requirement of 1 μsec translates into about 1000ft, which is roughly the range of the MICA2 radios. There are some systems, such as seismographic sensornet arrays, where much longer radio ranges are envisioned, but there the synchronization requirements are correspondingly more relaxed. Moreover, if the locations of the nodes are known, such propagation delays can be taken into account. Therefore, in what follows we will assume that propagation delays are either negligible or explicitly compensated for.

Most traditional methods synchronize a receiver with a sender by transmitting current clock values, and are thus sensitive to transmission delay variability and asymmetry. To avoid these vulnerabilities, RBS instead synchronizes receivers with each other. Reference broadcast signals are periodically sent in each region, and sensornet nodes record the time-of-arrival of these packets. Nodes within range of the same reference broadcast can synchronize their clocks by comparing their respective recent time-of-arrival histories. Nodes at distant locations (not in range of the same reference broadcast) can synchronize their clocks by following a chain of pairwise synchronizations. RBS is quite accurate because it is completely insensitive to transmission delays and asymmetries. In fact, errors in RBS arise only from two sources:

- Differences in time-of-flight to different receivers: as discussed earlier, in many sensornet systems we can safely assume that such differences are either completely negligible compared to the synchronization precision or, when location information is available, these differences can be explicitly compensated for.[2]
- Delays in recording packet arrivals: see [5] for a much fuller discussion of this point, but measurements described therein suggest that the receiving delays can be reasonably modeled as a Gaussian centered around some mean, with the mean being the same for all nodes (assuming they share the same hardware/software).

Both of these errors are extremely small in typical sensornet systems, with the former dominated by the latter. Therefore, most of the errors in synchronization are due to essentially random delays in recording time-of-arrivals.

To penetrate this noise, RBS uses pairwise linear regressions of the time-of-arrival data from a shared broadcast source. While this seems like a very promising approach, and has been verified on real hardware, there are two aspects of RBS, and in fact of any similar synchronization algorithm, that we wish to understand better. This paper is devoted to a theoretical analysis of these two issues, which we now describe in turn.

First, the resulting synchronization is purely pairwise. That is, for any pair of nodes i, j, RBS can compute coefficients $a_{i,j}, b_{i,j}$ that translate readings on i's clock into readings on j's clock: $t_j \approx t_i a_{ij} + b_{ij}$. However, this set of pairwise translations is not necessarily globally consistent. Converting times from i to j, and then j to k can be different than directly converting from i to k; *i.e.,* the transitive properties $a_{ij}a_{jk} = a_{ik}$ and $b_{ij}a_{jk} + b_{jk} = b_{ik}$ need not hold. This collection of (possibly inconsistent) pairwise synchronizations is not ideal when several sensornet nodes must have a single shared clock in order to carry out some joint task. Thus, we would like to understand how one could ensure the global consistency of the synchronizations. Note that requiring the pairwise synchronizations to be globally consistent is equivalent to saying that there is some universal time standard to which all nodes are synchronized (*e.g.,* the time of one particular node could serve as this universal time, though we choose to adopt a more distributed approach).

Second, the pairwise synchronizations are not optimally precise (*i.e.,* they do not have minimal variance from the truth). The RBS synchronization of two sensornet nodes is based only their time-of-arrival information from a single broadcast source. No information from other broadcast sources is used, nor is time-of-arrival information from other receivers. Thus, much relevant data is being ignored in the synchronization process, resulting in suboptimal precision.[3] We would like to understand how to use all available information to compute the optimally precise pairwise synchronizations.

[2] In other words, when locations are known each can receiver estimate, based on the relative location of the source, when the signal was sent (rather than when it was received) by subtracting out the computed time-of-flight. Synchronization comparisons would then be based on these computed send-times, not the time-of-arrivals.

[3] Some of this is inherent in the RBS approach and some is an artifact of the particular design described in [5]. Using only a single synchronization source is an artifact; not incorporating time-of-arrival data from other receivers is inherent in the general pairwise-comparison approach adopted by RBS. When we compare the optimal method against RBS in Section 4, we only consider the inherent differences (*i.e.,* we consider information from all relevant synchronization sources but only from the receivers directly involved).

Our two goals of optimal pairwise synchronization and globally consistent synchronization are logically distinct. In Section 3 we discuss these two goals and show that they have the same technical answer. That is, the most precise set of pairwise synchronizations are, in fact, globally consistent. We then, in Section 4, compare the precision of the optimal synchronization to that achieved by RBS. One result of note is that in a 2-dimensional grid-like network (whose nature we explain more carefully in Section 4), the error (variance) in the optimal synchronization between nodes a distant L apart grows as $\log L$, as opposed to growing linearly in L for RBS.

While our discussion is focused entirely on RBS, our methods and results could be extended to any pairwise synchronization procedure whose errors were independent. In addition, our focus here is primarily theoretical in that we have tried to understand the limits of what could be accomplished without regard for feasibility. However, in Section 5 we briefly discuss how one might turn this theory into practice. This requires confronting the fact that synchronization involves both *offset* and *skew*, where offset is the difference in value and skew is the difference in rate between two clocks.[4] The discussion in Sections 3 and 4 deals only with offsets; we assume that all clocks progress at the same rate, but start with arbitrary initial settings. In Section 5, we discuss how to use the same computational framework to deal with clock skew. We also describe how the set of calculations needed for our optimal and global synchronization could be achieved in a practical manner.

However, before delving into any of the technical details, we first give a cursory overview of related work.

2 Related Work

This paper deals primarily with RBS, which we have already reviewed. There are, of course, many other relevant approaches to clock synchronization, and we now mention briefly a few. See [5, 6] and references therein for more thorough reviews of the literature.

The most straightforward approach is to use the Global Positioning System (GPS) as the source of a universal clock. While GPS is extremely accurate, with commercial GPS receivers able to achieve better than 200nsec accuracy relative to UTC, GPS requires sensornet nodes to be equipped with special receivers; while including GPS receivers may become standard in future sensornet node designs, it is absent in some of the current systems (*e.g.*, the Berkeley Motes [10]). Moreover, GPS requires a clear sky view, and thus does not work inside buildings, underwater, or beneath dense foliage.

There is a large literature on how to synchronize clocks in traditional networked systems (*e.g.*, [3, 8, 15, 12]); among these, the Network Time Protocol [12] is the most widely deployed time synchronization algorithm and is notable for being scalable, self-configuring, robust to failures, and thoroughly tested. NTP and the other traditional methods, despite their many differences, all achieve synchronization through the exchange of current clock values. As discussed previously, this approach is vulnerable to sending delays and asymmetries in paths and does not take advantage of the special properties of sensornet broadcasts. Moreover, it assumes the presence of synchronized global-clocks, such as GPS, at many points in the network, and so the main focus is reducing the variance along the paths to these time oracles.

There are several proposals for synchronizing clocks within a single broadcast domain [18, 17, 13]. These all exploit the special properties of broadcast media and achieve high precision. However, they cannot synchronize nodes that do not lie within the same broadcast domain. Since our focus here is on global clock synchronization, we don't discuss these more local approaches further.

Two global synchronization proposals of note are [11] and [16]. The microsecond precision achieved in [11] is similar to our goals here, but the approach assumes a fixed topology and guarantees on latency and determinism in packet delivery. A very energy-efficient time diffusion algorithm is presented in [16], but the precision analysis assumes deterministic transmission times. Our interest here is in synchronization algorithms that do not require

[4] In our previous notation where $t_j \approx t_i a_{ij} + b_{ij}$, a_{ij} represents the relative clock skew and b_{ij} represents the relative clock offset.

specific underlying networks to function.

Some synchronization designs, such as [9, 7], integrate the MAC with the time synchronization procedure. While our discussion does not make assumptions about the underlying hardware and MAC, the results would benefit from these MAC-specific features to the extent that they reduce the magnitude of the receive-time errors.

Another quite different approach is that taken in [14], which doesn't directly synchronize clocks but instead refers to events in terms of their age. When exchanging these timestamps, they are updated to reflect the passage of time. As such, this work is complementary to what we discuss here, and represents a very attractive way to keep track of event times without adjusting local clocks.

The problem of calibration [19] is related to that of synchronization, though it differs in some essential details. The discussion in [2] is especially relevant to our discussion here, as it considers how to use nonlocal information across multiple calibration paths in a consistent manner.

3 Optimal and Global Synchronization

In this section we consider a simple model where clocks all progress at the same rate (*i.e.*, no skew), but have arbitrary offsets; we later, in Section 5, extend our results to the case of general clock skew. After describing the model and notation, we consider the question of optimal pairwise synchronization and then that of the most likely globally consistent synchronization. We then show their equivalence and end this section by describing a simple iterative computation of the solution and its variance.

3.1 Model and Notation

We consider the case where there are n sensornet nodes, and let r_i denote the i'th such node. These nodes use synchronization signals to align their clocks; let s_k denote the k'th synchronization signal. Our treatment does not care from whence these signals come, only which nodes hear them, so we don't identify the source of these signals. We let E be the set of pairs (r_i, s_k) such that node r_i receives signal s_k; in what follows, we will use the terms "node" and "receiver" interchangeably. In order to explain our theory, we make reference to a perfect universal time standard or clock; of course, no such clock exists and our results do not depend on such a clock, but it is a useful pedagogical fiction. In fact, the approximation of such a universal time standard is one of the goals of our approach.

We assume, in this section, that all clocks progress at the same rate and that propagation times are insignificant (or have been explicitly compensated for). We represent the offset of a node, or receiver, by the variable T_i. This offset is the difference between the local time on i's clock and the universal absolute time standard. Of course, there is a degree of freedom in choosing these T_i, as they could all be increased by the same constant without changing any of the pairwise conversions; the addition of such a constant term would reflect changing the setting of the global clock. We represent by U_k the time when synchronization signal s_k is sent (or, equivalently, received) according to the absolute time standard. The U_k's are not known, but are estimated as part of the synchronization process; thus, they are outputs, not inputs, of our theory.

Each node records the times-of-arrival of all synchronization messages they receive (*i.e.*, all those that they are in range of). We let y_{ik} denote the measured time on r_i's clock when it receives signal s_k. The quantity y_{ik} is defined if and only if $(r_i, s_k) \in E$. The basic assumption we make about measurement errors is that:

$$y_{ik} = U_k + T_i + e_{ik} \tag{1}$$

where e_{ik} is a random variable with mean zero and variance V_{ik}.[5] We further assume that all these random variables are independent.

[5] In reality, of course, the e_{ik} terms will have nonzero mean, but this mean is shared by all nodes i and all signals k, and so becomes a constant adjustment to all terms U_k.

This notion of independent errors is crucial. In fact, our treatment could be applied to other (non-RBS) pairwise synchronization methods as long as the intrinsic errors were independent. We focus exclusively on RBS, because the nature of the underlying errors are well understood (see [5]) and they appear to be independent; however, we hope to later extend our model to other approaches for which this independence assumption also holds.[6]

To convert times from node i to node j, one merely adds the difference $T_j - T_i$. In our previous notation, $b_{ij} = T_j - T_i$, and our assumption of uniform clock speed sets all $a_{ij} = 1$. To find the optimal (*e.g.*, the minimum-variance) pairwise synchronization between nodes i and j, we must produce the minimum-variance estimate of the difference $T_j - T_i$. Below we produce such an estimator that uses a network flow formulation related to the concept of the effective resistance of a resistor network. This approach estimates the difference $T_j - T_i$ directly, rather than estimating each quantity separately. Thus, there is no obvious *a priori* guarantee that these estimates will be consistent with each other (that is, there is no *a priori* guarantee that the optimal estimate of $T_j - T_i$ plus the optimal estimate of $T_k - T_j$ will equal the optimal estimate of $T_k - T_i$). Moreover, even if they are consistent, it is not clear *a priori* that they are the most likely set of offset assignments.

In contrast, to produce a globally consistent synchronization, we must estimate all the T_i independently and seek a maximum-likelihood joint choice of all the offsets T_i. When we assume the measurement errors e_{ik} are Gaussian we are able to reduce this maximum-likelihood problem to a linear system of least-squares equations. Surprisingly, the solution to this system of equations also solves the flow problem used to produce minimum-variance estimators.

3.2 Minimum-Variance Pairwise Synchronization

Given two nodes r_1 and r_2 an unbiased estimator of $T_1 - T_2$ can be obtained from any appropriate path between r_1 and r_2. In general such a path is of the alternating form $r_{i_1}, s_{k_1}, r_{i_2}, s_{k_2}, \cdots, s_{k_t}, r_{i_{t+1}}$ where $r_{i_1} = r_1$ and $r_{i_{t+1}} = r_2$ and each adjacent pair is in E. The corresponding estimator is $y_{i_1,k_1} - y_{i_2,k_1} + y_{i_2,k_2} - \cdots - y_{i_{t+1},k_t}$. which, in view of the equation $y_{ik} = U_k + T_i + e_{ik}$, is equal to $T_1 - T_2 + e_{i_1,k_1} - e_{i_2,k_1} + e_{i_2,k_2} - \cdots - e_{i_{t+1},k_t}$. This estimator is unbiased because each e_{ik} has zero mean.[7] By the independence of the e_{ik} its variance is $V_{i_1,k_1} + V_{i_2,k_1} + V_{i_2,k_2} + \cdots + V_{i_{t+1},k_t}$.

By considering appropriate weighted combinations of alternating paths we can obtain an estimator of much lower variance than any single path can provide, thus providing a more accurate synchronization of the two nodes. Such a weighted combination of paths is a flow from r_1 and r_2, satisfying the *flow conservation requirement* that the net flow into any node except r_1 and r_2 is zero. In this subsection we characterize the minimum-variance estimator of $T_1 - T_2$ in terms of flows.

Consider an undirected flow network with edge set E. We will use the following convention regarding summations: $\sum_{ik}$ will denote a summation over all pairs (i, k) such that $\{r_i, s_k\} \in E$; when k is understood from context, $\sum_i$ will denote a summation over all i such that $\{r_i, s_k\} \in E$; and when i is understood from context, $\sum_k$ will denote a summation over all k such that $\{r_i, s_k\} \in E$.

We first state, without proof, a basic but straightforward fact about unbiased estimators:

Theorem 1. *The unbiased estimators of $T_1 - T_2$ are precisely the linear expressions $\sum_{ik} f_{ik} y_{ik}$ such that $\{f_{ik}\}$ is a flow of value 1 from r_1 to r_2. Here f_{ik} is positive if the flow on edge $\{r_i, s_k\}$ is directed from r_i to s_k, and negative if the flow is directed from s_k to r_i. The variance of the unbiased estimator $\{f_{ik}\}$ is $\sum f_{ik}^2 V_{ik}$. A similar statement holds for the unbiased estimators of $T_j - T_i$, for any i and j.*

The problem of finding a minimum-variance unbiased estimator of $T_1 - T_2$ is related to the problem of determining the effective resistance between two nodes of a resistor network. In order to sketch this connection we review some basic facts about resistive electric networks.

[6] For instance, the traditional methods of pairwise synchronization, in which current clock values are exchanged, the errors in clock offset estimations are not independent between trials, as they are sensitive to asymmetries in the path.

[7] In fact, the conclusion follows if we only assume that they all have the same mean, which is why we are able to ignore the constant shared mean of the receive delays.

Let G be a connected undirected graph with vertex set V and edge set A, such that there is a resistance $R(u,v)$ associated with each edge $\{u,v\}$. An *applied current vector* is a vector e with a component $e(u)$ for each vertex, such that $\sum_{u\in V} e(u) = 0$; $e(u)$ represents the (steady-state) current (positive, negative or zero) injected into the network at vertex u. Associated with every applied current vector e is an assignment to each ordered pair $[u,v]$ of adjacent vertices of a current $c(u,v)$ and to each vertex u a potential $p(u)$ satisfying Kirchhoff's law (net current into a vertex = 0) and Ohm's law $p(v) - p(u) = c(u,v)R(u,v)$. The current is unique and the potential is unique up to an additive constant. When we want to identify the particular applied current vector e we write $c_e(u,v)$ and $p_e(v)$. A key property is the *superposition principle*:

$$c_{e_1+e_2}(u,v) = c_{e_1}(u,v) + c_{e_2}(u,v)$$

and

$$p_{e_1+e_2}(v) - p_{e_1+e_2}(u) = (p_{e_1}(v) - p_{e_1}(u)) + (p_{e_2}(v) - p_{e_2}(u))$$

The *effective resistance* between u and v is the potential difference $p(v) - p(u)$ when the applied current vector is as follows: $e(u) = 1$, $e(v) = -1$ and all other components of e are zero; *i.e.*, when one unit of current is injected at u and extracted at v. More generally, the effective resistance is the ratio of the potential difference to the current flow (which is merely the amount of current flowing out of the source).

The effective resistance between u and v can be characterized in terms of a minimum-cost flow problem with quadratic costs. It is the minimum, over all currents $c(u,v)$ satisfying Kirchhoff's law (with external current 1 at u and -1 at v) of $\sum_{(u,v)\in E} c(u,v)^2 R(u,v)$. This quadratic objective function represents the power dissipation in the network.

Now consider the undirected bipartite graph of signals $\{s_k\}$ and receivers $\{r_i\}$ as a resistor network, with the variance V_{ik} as the resistance of the edge $\{s_k, r_i\}$. Combining Theorem 1 with the minimum- cost-flow characterization of effective resistance we obtain the following theorem.

Theorem 2. *The minimum variance of an unbiased estimator of $T_1 - T_2$ is the effective resistance between r_1 and r_2, and the corresponding estimator is $\sum_{ik} f_{ik} y_{ik}$ where f_{ik} is the current along the edge from r_i to s_k when one unit of current is injected at r_1 and extracted at r_2.*

The following theorem establishes the mutual consistency of the minimum-variance estimators of the differences between offsets. Let $A(i,j)$ be the minimum-variance estimator of $T_i - T_j$.

Theorem 3. *For any three indices i, m and j, we have $A(i,m) + A(m,j) = A(i,j)$.*

Proof: We give the proof for the case where i, m and j are distinct, the other cases being trivial. For any two indices p and q let $e(p,q)$ be the applied current current vector with a 1 in position p, $a-1$ in position q and 0 elsewhere, and let $c(e(p,q))$ be the corresponding vector of edge currents. Then $A(i,m) = y \cdot c(e(i,m))$, $A(m,j) = y \cdot c(e(m,j))$ and $A(i,j) = y \cdot c(e(i,j))$, where y is the vector of measured values y_{ik}. Since $e(i,j) = e(i,m) + e(m,j)$ it follows from the superposition principle that $c(e(i,j)) = c(e(i,m)) + c(e(m,j))$. Thus

$$A(i,j) = y \cdot c(e(i,j)) = y \cdot c(e(i,m)) + c(e(m,j)) = A(i,m) + A(m,j)$$

QED

It follows from Theorem 2 that we can compute $A(i,j)$ for all i and j by computing $A(i,m)$ for all i and a fixed m and using the identity $A(i,j) = A(i,m) - A(j,m)$. This shows that the set of minimum-variance pairwise synchronizations are globally consistent. The question remains whether they are the maximally likely set of offset assignments.

3.3 Maximum-Likelihood Offset Assignments

We now seek the maximally likely set of offset assignments T_i. This approach is guaranteed to produce a globally consistent set of pairwise synchronizations, but it is not clear *a priori* that they are minimum-variance pairwise synchronizations. In this formulation we assume that the y_{ik} are independent Gaussian random variables such that y_{ik} has mean $U_k + T_i$ and variance V_{ik}. Then the joint probability density $\mathcal{P}$ of the y_{ik} given values T_i for the offsets of the receivers and U_k for the absolute transmission times of the signals is given by:

$$\mathcal{P} = \prod_{ik} \frac{1}{\sqrt{2\pi V_{ik}}} e^{-\frac{(y_{ik}-U_k-T_i)^2}{2V_{ik}}}$$

We shall derive a system of linear equations for the T_i and U_k that maximize this joint probability density.

Let C_{ik} denote the reciprocal of V_{ik}. We refer to C_{ik} as the *conductance* between s_k and r_i. Let D_k denote $\sum_i C_{ik}$.

Differentiating the logarithm of the joint probability density with respect to each of the U_k and T_i we find that the choice of $\{U_k\}$ and $\{T_i\}$ that maximizes the joint probability density is a solution to the following system of equations:

For each k,

$$\sum_i C_{ik}(U_k + T_i) = \sum_i C_{ik} y_{ik} \tag{2}$$

For each i,

$$\sum_k C_{ik}(U_k + T_i) = \sum_k C_{ik} y_{ik} \tag{3}$$

Eliminating the variables U_k we arrive at the following equations interrelating the offsets T_i:

For each r_i,

$$\sum_k \frac{1}{D_k} \sum_j C_{ik} C_{jk} (T_j - T_i) = \sum_k \frac{1}{D_k} \sum_j C_{ik} C_{jk} (y_{jk} - y_{ik}) \tag{4}$$

where the double summations are over all pairs (k, j) such that $(r_i, s_k) \in E$, $j \neq i$ and $(r_j, s_k) \in E$.

3.4 Equivalence of the Two Formulations

The following theorem shows that, even though the minimum-variance pairwise synchronization and the maximum-likelihood offset assignment appear to be based on different principles, they determine the same values of $T_j - T_i$, for all i and j.

Theorem 4. *For any fixed index m we obtain a solution to the system of equations 4 by setting $T(i) = A(i, m)$ for each i.*

Proof: We need to show that, for each i

$$\sum_k \frac{1}{D_k} \sum_j C_{ik} C_{jk} (A(j, m) - A(i.m)) = \sum_k \frac{1}{D_k} \sum_j C_{ik} C_{jk} (y_{jk} - y_{ik})$$

Since $A(j, i) = A(j, m) - A(i, m)$ it suffices to prove that

$$\sum_k \frac{1}{D_k} \sum_j C_{ik} C_{jk} A(j, i) = \sum_k \frac{1}{D_k} \sum_j C_{ik} C_{jk} (y_{jk} - y_{ik})$$

where the double summations are over all pairs (k, j) such that $(r_i, s_k) \in E$, $j \neq i$ and $(r_j, s_k) \in E$.

Let y be the vector with generic element y_{ik}. By the superposition principle the left-hand side is the inner product of y with the current vector when $\sum_k \frac{C_{ik}}{D_k} \sum_j C_{ik} C_{jk}$ units of current are extracted at i and for each $j \neq i$ $\sum_k \frac{C_{ik} C_{jk}}{D_k}$ units of flow are inserted at j. One can check that Kirchhoff's Law and Ohm's Law are satisfied when the potentials and currents are as follows: at all r_j, where $j \neq i, p(r_j) = 0$; $p(r_i) = 1$; for each s_k incident with r_i, $p(s_k) = \frac{C_{ik}}{D_k}$; for each s_k not incident with r_i, $p(s_k) = 0$; for each edge $[r_j, s_k]$ such that r_i is incident with $s_k, c(r_j, s_k) = \frac{C_{ik} C_{jk}}{D_k}$; for each edge $[s_k, r_i], c(s_k, r_i) = \frac{C_{ik}}{D_k} \sum_{j \neq i} C_{jk}$, where s_k is adjacent to r_i and the sum is over all j unequal to i such that r_j is incident with s_k. Except as stated all currents are zero. One can now check that for every component y_{pq} the current in edge $[r_p, s_q]$ is equal to the coefficient of y_{pq} in the right-hand side, thus proving that the two sides of the equation are equal. QED

3.5 Solving the Equations

The solution to the system of equations 2 and 3 can be found through a simple two-step iterative process. In the first step, the y_{ik} and T_i are used to estimate the U_k:

For each k,

$$U_k \leftarrow \frac{\sum_i C_{ik}(y_{ik} - T_i)}{\sum_i C_{ik}}$$

In the second step, the y_{ik} and U_k are used to estimate the T_i:

For each i,

$$T_i \leftarrow \frac{\sum_k C_{ik}(y_{ik} - U_k)}{\sum_k C_{ik}}$$

Each iteration reduces $\sum_i \sum_k C_{ik}(y_{ik} - U_k - T_i)^2$. It follows that the iterative process converges to a solution of the system. Convergence can be accelerated by over-relaxation techniques which are standard in numerical analysis [1].

The solution of the system maximizes the joint probability density of the measurements $\{y_{ik}\}$ under the assumption that the quantities $y_{ik} - U_k - T_i$ are independent Gaussian random variables with means 0 and respective variances V_{ik}. The solution provides a minimum-variance estimator of each quantity $T_i - T_j$, the difference of offsets between receivers r_i and r_j.

While this theory produces optimal (in two senses of optimality, maximum-likelihood and minimum-variance) estimators, it does not directly reveal the quality of the estimated values. In the next section we derive expressions for the variances of the estimators.

3.6 Computing the Variance

We now compute the variance of our estimator of the offset $T_2 - T_1$ between receivers 1 and 2 (or, similarly, between any two receivers). The approach is based on Theorem 2, which expresses this variance as the effective resistance between two nodes of a resistive network. We later, in Section 4, use these error estimates to compare the precision of the optimal method to that of RBS in a few simple scenarios. Note that because the optimal method and RBS are both unbiased, the variance is the relevant measure of precision.

Consider a resistor network with a node r_i for receiver i, a node s_k for each signal k and a resistance V_{ik} between r_i and s_k. Assume that the potential difference between r_1 and r_2 is held at 1. Then the effective resistance R is the reciprocal of the amount of current flowing from r_1 to r_2 when current is conserved at all other nodes. Let P_i denote the potential at receiver r_i and Q_k, the potential at signal node s_k.

Since C_{ik} is the conductance of the edge (r_i, s_k) it follows from Ohm's Law that the current along the edge from r_i to s_k is $C_{ik}(Q_k - P_i)$. Kirchhoff's Law of conservation of current thus states that:

For all i except 1 and 2,

$$\sum_k C_{ik}(Q_k - P_i) = 0$$

For all k,

$$\sum_i C_{ik}(Q_k - P_i) = 0$$

This system of equations, together with the equations $P_1 = 0$ and $P_2 = 1$ can be solved by an iterative process that alternates between updating the Q_k and updating the P_i in a manner analogous to the iterative solution of Equations 2 and 3. Once we have the quantities P_i and Q_k, the variance, or the effective resistance R, is given by the ratio of the potential difference and the current flow:

$$R = \frac{1}{\sum_k f_{1k}} \tag{5}$$

Where the sum in the denominator is over all k such that $(r_1, s_k) \in E$.

4 Precision Results

We now compute the variance of this optimal estimator in a few simple scenarios. We first consider cases where we can calculate the variance analytically and then consider the more realistic scenarios where we are forced to calculate the variance numerically. In what follows, we set $V_{ik} = 1$ for all i, k; thus, the absolute values of the variances are not of interest since they scale linearly in the V_{ik}. To give a point of reference, we compare the resulting variances to those achieved by RBS.

4.1 Analytical Calculations

Let r_1 and r_2 be receivers. We compare the variances of the estimators of $T_1 - T_2$ in some special cases Each particular problem instance is specified by listing, for each signal, the set of receivers that receive it. Alternatively, the instance can be described by a bipartite graph, with the receivers constituting one part of the vertex set, and the signals constituting the other part.

Recall that, with no clock skew, the RBS estimation of $T_1 - T_2$ comes from the shortest path of the form

$$r_{i_1}, s_{k_1}, r_{i_2}, s_{k_2}, \cdots, s_{k_t}, r_{i_{t+1}}$$

where $r_{i_1} = r_1$ and $r_{i_{t+1}} = r_2$ and each adjacent pair is in E.[8] The resulting variance is merely the length of the path (since we've set all $V_{ik} = 1$).

In general, we expect that the estimator of $T_1 - T_2$ provided by the optimal method will have much lower variance than the estimator provided by RBS when there are many alternate (and mostly disjoint) paths of near-minimum hop count between r_1 and r_2. The following examples are consistent with this expectation.

[8] In practice, RBS picks the path with the lowest observed variance, but for our comparison we pick the shortest path in terms of absolute conversion hops.

Example 1: Every signal is received by every receiver. In this case, RBS and the optimal method provide the same estimate of $T_1 - T_2$, and its variance is $2/S$, where S is the number of signals.

Example 2: There are n receivers and $\binom{n}{2}$ signals; for every pair i, j of receivers there is a signal that is received only by receivers i and j. In this case optimal method yields a variance of $2/n$ whereas RBS produces a variance of 2.

Example 3: The bipartite graph is the d-dimensional unit hypercube. The effective resistance between any pair of nodes (and hence the variance provided by the optimal method), is less than or equal to 1 when $d \leq 2$ and less than or equal to $\frac{8}{3d}$ for all d, whereas RBS provides a variance equal to the distance between r_1 and r_2.

Example 4: The bipartite graph is an infinite d-dimensional grid with unit distance between neighboring nodes. Let L be the Manhattan distance between r_1 and r_2. It is known [4] that the effective resistance between r_1 and r_2 is $O(\log L)$ when $d = 2$ and is bounded above by a constant when $d \geq 3$. Thus the variance of the estimate provided by optimal method is $O(\log L)$ when $d = 2$ and bounded by a constant when $d \geq 3$. RBS provides a variance of L in both cases. However, the grid considered here is unnatural, because it does not accurately reflect real radio ranges; instead, it is assumed that synchronization sources are perfectly interspersed with receivers on a grid, and only the nearest $2d$ receivers can hear a given source.

These artificialities aside, the infinite 2-dimensional grid is, to some extent, the prototype of situations where many sensor nodes are scattered more or less uniformly over a region of the Earth's surface. Thus one expects that, in such cases, the estimator provided by the optimal method for the difference in clock offsets between two nodes will have a variance proportional to the logarithm of their distance, whereas the estimator provided by RBS will have a variance proportional to their distance. Below we numerically investigate the 2-dimensional case with more realistic configurations.

4.2 Numerical Calculations

We now consider a slightly more realistic grid, one where we need to use numerical calculations to determine the results. We consider a large rectangular grid (42 by 42 nodes), and each node is in range of the nearest eight neighbors. All nodes send a single synchronization message.

We measure the distance between any two nodes in terms of the number of pairwise comparisons needed to compute their offsets in RBS. We call this the RBS path length; for points with the same x or y coordinate, the RBS path length is half of the number of grid hops between the nodes.

We first consider, in Figure 1, how the optimal and RBS variances grow with increasing path length. The left-hand graph compares the variance of RBS and the optimal method, and the right-hand graph shows the optimal method by itself so its log-like growth is more apparent. These results confirm the analytical results for how variance grows with path length on a grid-like sensornet configuration. Note that even with short path lengths, the optimal method has significantly lower variance than RBS through its use of global information.

There are situations where one need only synchronize a pair of nodes, rather than the whole sensor network. The optimal method uses information from all nodes, which would incur significant energy expenditures. We now ask how many nodes need be involved in order to achieve high precision. Figure 2 illustrates what happens when we consider increasing rings of nodes around paths of RBS path length 2, 10, and 20. We start by involving only those nodes within one hop of the path, and then consider those within two hops of the path, and so on. As is clear from the graphs, there is a very rapidly diminishing return in precision once the depth of the surrounding nodes is on the order of the path length. This suggests that one could greatly reduce the energy required to produce precise pairwise synchronizations by limiting the depth of the surrounding grid.

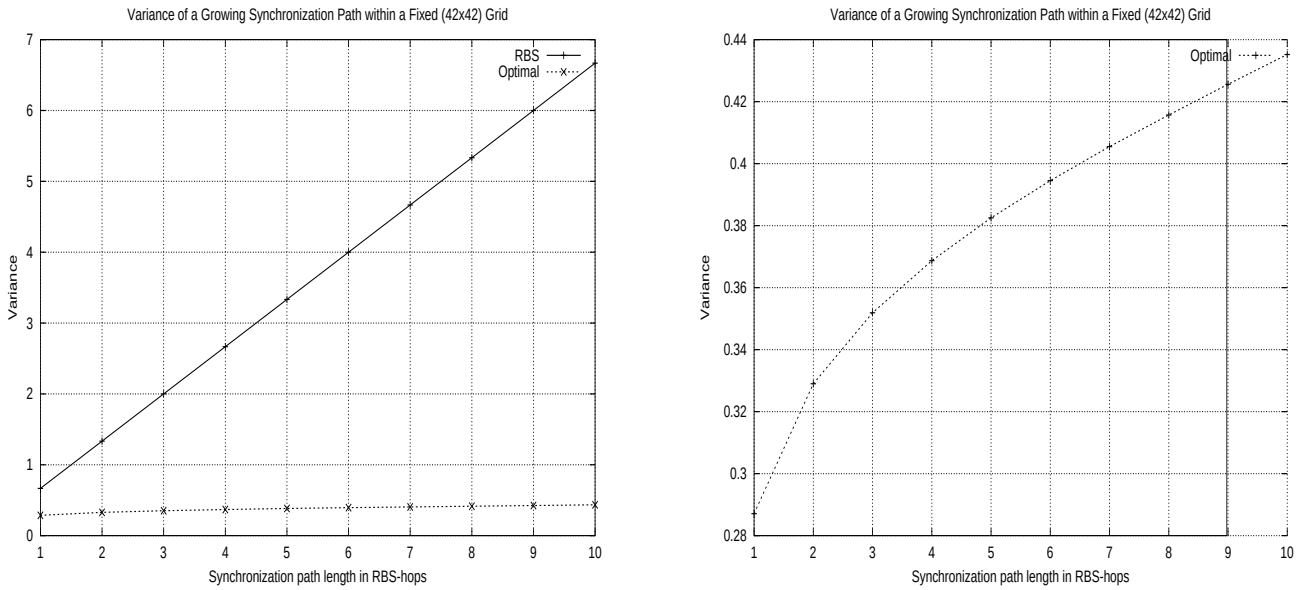

Figure 1: The variance in the pairwise synchronization resulting from the optimal method and RBS as the RBS path length between the two nodes is increased from one to ten. The grid is 42 by 42 and all $V_{ik} = 1$. The left-hand graph shows both the optimal method and RBS, while the right-hand graph shows only the optimal method.

5 From Theory to Protocol

We have described abstractly how one could optimally compute the appropriate clock offsets T_i from the measurement data y_{ik}. In this section we briefly discuss how one might transform this theory into a practical protocol. This discussion is by no means complete or definitive, and is completely untested; instead, we offer it only as providing some glimmer that the ideas of presented here could be successfully applied to real systems with their skewed clocks and energy constraints.

The two issues we address are:

- Generalizing the theory to compensate for clock skew
- Turning the abstract calculation into a series of practical message exchanges

5.1 Clock Skew

The theoretical treatment assumed that all clocks progressed at the same rate. We now relax this assumption and describe how one can estimate the relative rates of clocks. In particular, we wish to estimate parameters α_i that describe the rate of the local clocks relative to the standard clock: if a time δ has elapsed on the universal standard clock then each local clock shows that time $\alpha_i\delta$ has elapsed (so large α_i reflect fast clocks). As with the offsets T_i, there is a degree of freedom in choosing these α_i; each could be multiplied by the same constant (which would only change the speed of the absolute clock).

Given the pair (α_i, T_i) for some node i, we can translate local times t_i into *standard* times τ: $\tau = \frac{t_i}{\alpha_i} - T_i$. Moreover, if one had the constants α_i, then one can estimate the T_i's as in the previous section by first dividing all local clocks by α_i. Thus, we must now describe how to obtain estimates of these skew values α_i, and do so without knowledge of the offsets T_i (since the computation of the T_i requires knowledge of the α_i).

To estimate clock rates, we use the same set of synchronization signals, but now select pairs of them originating from the same source spaced at sizable intervals (*i.e.*, large compared to the variances V_{ik} of the individual measurements).

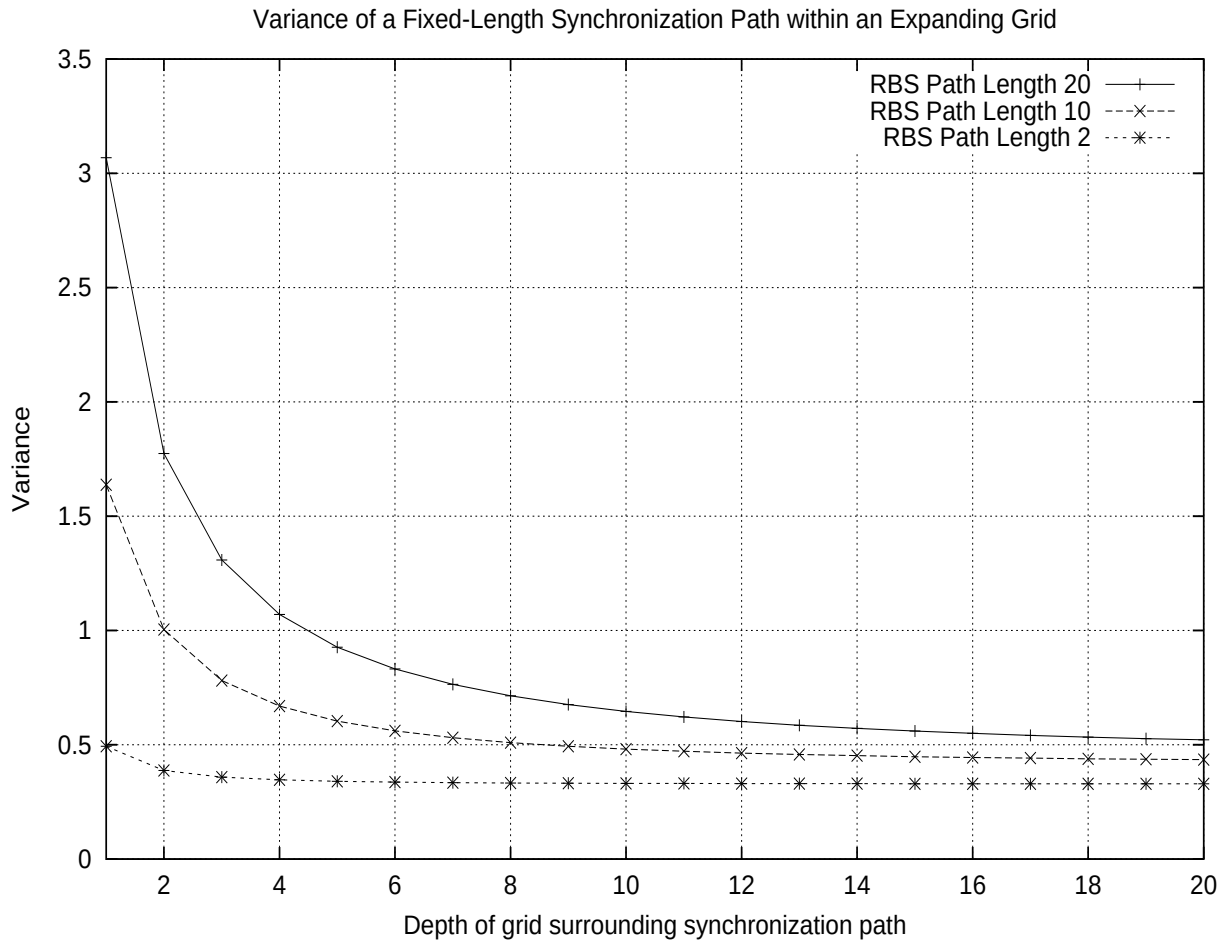

Figure 2: The variance in the pairwise synchronization resulting from the optimal method as the depth of the grid surrounding the path is increased. The path lengths are 2, 10, and 20, and all $V_{ik} = 1$.

We label the k'th signal pair by p_k. We let W_k and w_{ik} represent the time elapsed between their transmission as measured by, respectively, the standard clock and i's local clock. In the notation of Section 3, W_k is the difference between the pair of signals of the U values; w_{ik} is the corresponding difference in the y values. We assume that the measurement errors, as expressed by the e_{ik}, are negligible compared to the magnitude of the W_k. If all clocks progressed at perfectly constant rates, then $w_{ik} = W_k \alpha_i$ for each i, k and we could estimate the variables α_i based on a single measurement for each i.

However, clock rates drift and wander over time in random and unpredictable ways. The skew variable α_i represent the long-time averages of the skew, and instantaneous estimates of the skew are affected by drifts in the clock rate. More specifically, we assume that clock rates vary in such a way that $w_{ik} = \alpha_i e^{\delta_{ik}} W_k$ where δ_{ik} is a random variable with mean zero and variance X_{ik}.

Note that, when taking the logs, the equation becomes:

$$\log w_{ik} = \log W_k + \log \alpha_i + \delta_{ik}$$

Note that when changing variables, this is exactly the form of Equation 1 with the following substitutions:

- $y_{ik} \rightarrow \log w_{ik}$

- $U_k \rightarrow \log W_k$
- $T_i \rightarrow \log \alpha_i$
- $e_{ik} \rightarrow \delta_{ik}$
- $V_{ik} \rightarrow X_{ik}$

Thus, we can apply all of the previous theory to the estimate of clock skew. The difference is that the basic measurements now are the locally measured *intervals* between two synchronization signals (and thus are unaffected by the offsets), and the magnitude of these intervals is much larger than the measurement errors (*i.e.*, $W_k \gg V_{ik}$) so the only significant errors arise from clock frequency drift. The same set of equations, and the same iterative procedure, will produce the optimal and globally consistent estimates of skews through the set of parameters α_i.

We can treat skew and offsets on different time scales. That is, we can adjust the parameters α_i roughly every τ_s time units, whereas we adjust the parameters T_i roughly every τ_o time units, with $\tau_s \gg \tau_o$; the absolute values of these quantities will depend on the nature of the clocks and the setting. When computing the offsets we treat the skew as constant (and known), so we can apply the theory we presented earlier. On longer time scales, we adjust the skew using the same iterative procedure (with different variables).

The result is that we can treat general clocks with both offsets and skews. Experiments with real clocks will be needed before we can fine tune the time constants and verify that this two-time-scale approach is valid.

5.2 Outline of a Synchronization Protocol

The calculations in Section 3 seem, at first glance, far too complex for implementation in actual sensornets. This may well be true, but here we sketch out how one might achieve the desired results in an actual sensornet protocol. None of the various parameters are specified; we only sketch out the structure of what a protocol might look like.

The synchronization process can use any message as a synchronizing signal. We will assume that all messages have unique identifiers, so different nodes can know that they are referring to the receipt of the same message. Also, in what follows pairs of nodes are considered to be in range of another node if and only if they can exchange messages; pairs where one node can hear another, but not vice versa, are not considered to be in range. We first describe the approach for estimating clock offsets, and then later describe how to use this for estimating clock skew.

Each node broadcasts a synchronization status message every τ_o (with some randomness), which contains data for the last τ_w seconds; τ_w represents a time window after which data is discarded. Each status message contains:

- Their current estimate of T_i.
- Their current estimates of U_k for all previous status messages sent within the last τ_w seconds.
- Their time-of-arrival data y_{ik} for all status messages received in the last τ_w seconds.

Upon receipt of a status message, node i uses the data to update their estimate of T_i and U_k as described in the iterative equations 2 and 3. Thus, each round of synchronization messages invokes another round in the iterative computation.

At longer intervals, τ_s, nodes send skew status messages that additionally contain the data on α_i, W_k, and w_{ik}. This data can be used to update the skew variables in the same way as for the offset variables.

The main open question is what rate of message passing is needed to achieve reasonable degrees of convergence and whether this entails too much energy consumption. The answer will depend greatly on the nature of clock drifts and measurement errors in real systems. If the rates of change are slow, then once the system is reasonably well synchronized only a slow rate of iterations will be required to stay converged. If the rates of change are high, then a

much faster rate of iterations will be required to stay within the desired precision bounds. Because we don't know what the relevant rates of change will be, we don't offer any conjectures about the feasibility of this approach. Instead, we hope to investigate the issue empirically by deploying this approach in an experimental setting.

While perhaps too energy-expensive, our approach does have the advantage of compactly representing the synchronization data. Approaches that produce pairwise synchronizations must retain at node i the conversion coefficients (a_{ij}, b_{ij}) for *all* nodes j to which i might need to synchronize. In cases where many nodes need to coordinate, this might be unwieldy.

6 Discussion

Clock synchronization is important for sensornets because they often require close coordination between nodes in tasks as varied as data fusion, TDMA scheduling, and coordinated actuation. RBS is a promising approach to sensornet clock synchronization, and this paper investigated how one might extend this approach to yield optimally precise and globally consistent clock synchronization. Our main finding is that there is a simple iterative clock adjustment algorithm that achieves both of these aims.

It is an open question whether this result will lead to a practical synchronization method. The key issue is whether the rate of iterations needed to meet the desired precision bounds is too energy intensive. However, as we described in Section 4, if one is only interested in synchronizing a pair of nodes, then one can greatly reduce the scope of nodes involved in the synchronization process.

Even if our results do not lead to a feasible synchronization algorithm, they can provide a yardstick against which to compare methods of computing offsets and skews from reference-broadcast time-of-arrival data. Designers now know the optimal results that can be achieved, and can make their own energy-precision tradeoffs with that in mind.

Acknowledgments

We would like to thank David Karger for suggesting the maximum likelihood formulation and for many other useful comments during the early stages of this work.

References

[1] BERTSEKAS, D. P., AND TSITSIKLIS, J. N. *Parallel and Distributed Computation – Numerical Methods*. Prentice Hall, 1989. ISBN 0-13-648759-9.

[2] BYCHKOVSKIY, V., MEGERIAN, S., ESTRIN, D., AND POTKONJAK, M. Colibration: A Collaborative Approach to In-Place Sensor Calibration. In *2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)* (Palo Alto, CA, USA, April 2003).

[3] CRISTIAN, F. Probabilistic clock synchronization. *Distributed Computing 3* (1989), 146–158.

[4] DOYLE, P. G., AND SNELL, J. L. *Random Walks and Electrical Networks*. Mathematical Association of America, 1984.

[5] ELSON, J., GIROD, L., AND ESTRIN, D. Fine–grained network time synchronization using reference broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)* (Boston, MA, December 2002), pp. 147–163.

[6] ELSON, J., AND RÖMER, K. Wireless sensor networks: A new regime for time synchronization. In *Proceedings of the First Workshop on Hot Topics In Networks (HotNets-I)* (Princeton, New Jersey, October 2002).

[7] GANERIWAL, S., KUMAR, R., ADLAKHA, S., AND SRIVASTAVA, M. B. Network-wide Time Synchronization in Sensor Networks. Tech. rep., University of California, Dept. of Electrical Engineering, 2002.

[8] GUSELL, R., AND ZATTI, S. The accuracy of clock synchronization achieved by TEMPO in Berkeley UNIX 4.3 BSD. *IEEE Transactions on Software Engineering 15* (1989), 847–853.

[9] HILL, J., AND CULLER, D. A wireless embedded sensor architecture for system-level optimization. Tech. rep., U.C. Berkeley, 2001.

[10] HILL, J., SZEWCZYK, R., WOO, A., HOLLAR, S., CULLER, D., AND PISTER, K. System architecture directions for networked sensors. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)* (Cambridge, MA, USA, November 2000), ACM., pp. 93–104.

[11] LIAO, C., MARTONOSI, M., AND CLARK, D. W. Experience with an adaptive globally-synchronizing clock algorithm. In *Eleventh Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA '99)* (New York, June 1999), pp. 106–114.

[12] MILLS, D. L. Internet time synchronization: The network time protocol. In *Zhonghua Yang and T. Anthony Marsland (Eds.), Global States and Time in Distributed Systems*. IEEE Computer Society Press, 1994.

[13] MOCK, M., FRINGS, R., NETT, E., AND TRIKALIOTIS, S. Continuous clock synchronization in wireless real-time applications. In *The 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00)* (Washington - Brussels - Tokyo, Oct. 2000), IEEE, pp. 125–133.

[14] RÖMER, K. Time Synchronization in Ad Hoc Networks. In *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 01)* (Long Beach, CA, Oct. 2001). www.inf.ethz.ch/vs/publ/papers/mobihoc01-time-sync.pdf.

[15] SRIKANTH, T. K., AND TOUEG, S. Optimal clock synchronization. *J-ACM 34*, 3 (July 1987), 626–645.

[16] SU, W., AND AKYILDIZ, I. Time-Diffusion Synchronization Protocol for Sensor Networks. Tech. rep., Georgia Institute of Technology, 2002.

[17] VERÍSSIMO, P., AND RODRIGUES, L. A posteriori agreement for fault-tolerant clock synchronization on broadcast networks. In *Proceedings of the 22nd Annual International Symposium on Fault-Tolerant Computing (FTCS '92)* (Boston, MA, July 1992), D. K. Pradhan, Ed., IEEE Computer Society Press, pp. 85–85.

[18] VERISSIMO, P., RODRIGUES, L., AND CASIMIRO, A. Cesiumspray: a precise and accurate global time service for large-scale systems. Technical Report NAV-TR-97-0001, Universidade de Lisboa, 1997.

[19] WHITEHOUSE, K., AND CULLER, D. Calibration as parameter estimation in sensor networks. In *Proceedings of the First ACM International Workshop on Sensor Networks and Applications (WSNA 2002)* (Atlanta, Georgia, USA, September 2002).