Global Synchronization in Sensornets

Richard M. Karp Jeremy Elson Christos H. Papadimitriou 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. Within the context of this model we characterize the optimally precise clock synchronization algorithm and establish its global consistency. In the course of this analysis we point out unexpected connections between optimal clock synchronization, random walks, and resistive networks, and present a polynomial-time approximation scheme for the problem of calculating the effective resistance in a network based on min-cost flow. We also sketch a polynomial-time algorithm for finding a schedule of data acquisition giving the optimal trade-off between energy consumption and precision of clock synchronization. We end with a discussion of more practical issues, including synchronizing in the presence of clock skews, defining the actual message exchanges, and possible deployment in a network of seismic sensors that requires global clock consistency.

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 by producing on-demand pairwise synchronization with low overhead and high precision [5]. RBS is specifically designed for sensornet contexts in which (1) communications are locally broadcast, (2) the maximum speed-of-light delay between sender and receiver is small compared to the desired synchronization precision, and (3) the delays between time-stamping and sending a packet are significantly more variable than the delays between receipt and time-stamping a packet (so estimates of when a packet is sent are far noisier than estimates of when it is received). See [5] for a much fuller discussion of this last 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).

There is a vast literature on clock synchronization in the theory and distributed systems literature, which we don't have space to review (*e.g.*, [3, 7, 16, 13]; see [4] and references therein for a comprehensive review). We note, however, that most traditional methods synchronize a receiver with a sender by transmitting current clock values, and are thus sensitive to transmission delay variability and asymmetry. In contrast, RBS avoids these vulnerabilities by synchronizing receivers with each other, leveraging the special properties of sensornet communications. Reference broadcast signals are periodically sent in each region, and sensornet nodes record the times-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 therefore completely insensitive to transmission delays and asymmetries. In fact, errors in RBS arise only from differences in time-of-flight to different receivers and delays in recording packet arrivals. In the contexts for which RBS is intended, both of these errors are quite small and the latter dominates the former. Therefore, most of the errors in synchronization are due to essentially random delays in recording times-of-arrival (which, as observed earlier, are reasonably modeled as Gaussian).

To penetrate this randomness, 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 improve upon. First, the resulting synchronization is purely pairwise, in that 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 via $t_j \approx t_i a_{ij} + b_{ij}$, but these pairwise translations are 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.[1] Second, the pairwise synchronizations are not optimally precise in that 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.[2]

We address these limitations in a simplified model of synchronization in which changes in clock skew (differences in the *rates* of clocks) occur at much longer time scales than changes in clock offset (differences in the current clock *values*);[3] that is, we assume that over short time scales the clock skews are known (and compensated for) so synchronization is used only to adjust for clock offsets; estimates of clock skew are taken on much longer time scales. Thus, in what follows we will assume that all clocks advance at the same rate (because any differences in rate are explicitly compensated for); later, in Section 5, we will relax this assumption.

Our focus in this paper is primarily theoretical and we do not evaluate the feasibility (in terms of energy consumption) of this approach. However, we are planning to adopt this approach in a seismographic sensornet array. The requirements of optimally precise and globally consistent time synchronization are particularly acute in this context. We expect that there are ways to increase the energy efficiency of the approach described here without sacrificing significant precision or consistency.

RBS is, of course, not the only approach to sensornet clock synchronization. In some contexts, Global Positioning System (GPS) can provide a universal clock signal, but GPS requires a clear sky view, and thus does not work inside buildings, underwater, or beneath dense foliage. Moreover, many current sensornet nodes (*e.g.,* the Berkeley Motes [9]) are not equipped with GPS. There are several proposals for synchronizing clocks within a single broadcast domain [19, 18, 14], but they do not generalize to global synchronization which is what we address here.

Two global synchronization proposals of note are [12] and [17]. The microsecond precision achieved in [12] 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 [17], but the precision analysis assumes deterministic transmission times. Our interest here is in synchronization

[1]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).

[2]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.

[3]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.

algorithms that do not require specific underlying networks to function.

Some synchronization designs, such as [8, 6], 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. We note that while the discussion of our approach builds on RBS, our methods and results could be extended to any pairwise synchronization procedure whose errors were independent.

Another quite different approach is that taken in [15], which doesn't directly synchronize clocks but instead refers to events in terms of their age, not time. The problem of calibration [20] 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.

2 Summary of Results

The core problem of the paper is the following. We are given a set of *receivers*. Each receiver r_i has a clock that is offset from a (fictitious) universal time standard by a constant amount T_i. We are also given a set of synchronization signals. Each synchronization signal s_k is transmitted at an unknown time U_k and is received by a set of receivers. the time-of-flight of s_k is negligible. If r_i is a receiver of s_k then r_i measures the arrival time of s_k on its local clock. Let this measured time be y_{ik}. We assume that $y_{ik} = U_k + T_i + e_{ik}$ where the error e_{ik} is a random variable with zero mean and variance V_{ik}. We also assume that the errors e_{ik} are independent.

The main results are as follows:

1. We define a resistive network with a node for each receiver and each signal, such that the minimum-variance estimator of $T_i - T_j$ is derivable from the distribution of current when one unit of current is inserted at node T_i and extracted at node T_j.The variance of the estimator is the effective resistance between T_i and T_j. The variances V_{ik} appear as resistances in this network.
2. The minimum-variance estimator is globally consistent, in the sense that for any triple (i, j, m) the estimates of $T_i - T_j$, $T_j - T_m$ and $T_m - T_j$ sum to zero.
3. The effective resistance between two nodes of a network can be approximated with relative error ϵ by performing $\sqrt{\frac{V}{\epsilon R}}$ flow augmentations on the network, where V is the sum of the resistances and R is the effective resistance.
4. The effective resistance of the regular infinite d-dimensional grid is given explicitly, and indicates the advantage of the proposed synchronization scheme over its predecessor RBS in this case. We believe that similar advantages will typically be realized in large-scale sensornets in which the sensors have a homogeneous spatial distribution.
5. Under the additional assumption that the errors e_{jk} are Gaussian, the maximum-likelihood joint choice of the T_i and U_k, subject to the convention that $T_1 = 0$, is the unique solution to a linear system of least-squares equations. This sparse system of equations can be solved iteratively by a distributed sensornet algorithm in which each receiver or generator of a signal is responsible for updating the corresponding variable T_i or U_k.
6. The maximum-likelihood joint choice of the T_i and U_k agrees exactly with the minimum-variance pairwise estimates of $T_i - T_j$. Therefore the minimum-variance estimator is what is known in statistics as an efficient estimator.
7. The maximum-likelihood estimate of T_i is exactly the hitting time of a random walk from r_i to r_0 on a weighted directed graph with 'delay' y_{ik} on edge $[r_i, s_k]$ and $-y_{ik}$ on edge $[s_k, r_i]$, where r_i is a receiver of signal s_k, and the transition probabilities out of each vertex are inversely proportional to the variances V_{ik}.

8. A polynomial-time algorithm is presented which, given a set of receivers and a set of potential signals, determines the optimal repetition rate of each signal to minimize energy consumption while keeping the variance of the estimate of each offset $T_i - T_j$ below a specified value.
9. A method is given for estimating clock skews under the assumption that the clock of each receiver r_i advances at a fixed rate α_i per unit time. The method is based on measurements of the time elapsed on the clock of each signal transmitter and the time elapsed on the clock of each receiver between two transmissions of the same signal widely separated in time. The method is based on an isomorphism between this version of the clock skew problem and the clock offset problem described above.

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 r_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 it receives (*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}. We further assume that all these random variables are independent.

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 $T_j - T_i$. 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.

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.

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.

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. Its proof is a simple application of the superposition principle. 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 $A(i,m) + A(m,j) = A(i,j)$.*

It follows from Theorem 3 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_k 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 *conductivity* between s_k and r_i.

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,

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

¿From these equations we can derive an interpretation of each T_i as the hitting time of a random walk from r_i to r_0 on a directed graph with a positive or negative 'delay' on each edge. Assume that the set of indices i associated with the receivers is disjoint from the set of indices k associated with the signals. Under this assumption there is no ambiguity in defining, for each signal index k, a new variable T_k equal to $-U_k$. The system of equations becomes:

For each k,

$$T_k = \frac{\sum_i C_{ik}(-y_{ik} + T_i)}{\sum_i C_{ik}} \tag{4}$$

For each $i \neq 0$,

$$T_i = \frac{\sum_k C_{ik}(y_{ik} + T_k)}{\sum_k C_{ik}} \tag{5}$$

Fixing T_0 at 0, it is clear by inspection that these equations support the following interpretation: for each receiver i, T_i is the expected total delay of a random walk starting at r_i and ending at the first visit to r_0, where the transition probability from r_i to s_k is $\frac{C_{ik}}{\sum_k C_{ik}}$, the transition probability from s_k to r_i is $\frac{C_{ik}}{\sum_i C_{ik}}$, the delay on a transition from r_i to s_k is y_{ik} and the (negative) delay on a transition from s_k to r_i is $-y_{ik}$.

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. Our proof is based on the superposition principle, but the theorem can also be seen as a consequence of the Cramer-Rao inequality [11], a general tool for eastablishing that the variance of an estimator is best possible.

Theorem 4. *For any fixed index m we obtain a solution to the system of equations 4 and 5 by setting* $T(i) = A(i, m)$ *for each* i.

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].

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. The variance of each the estimators can be obtained by computing the effective resistance between the corresponding receivers. This can be done exactly by solving a system of linear equations or approximately by a new approximation algorithm based on minimum-cost flow. these two approaches are described in the following subsections.

3.6 Computing Optimal and Near-Optimal Estimators and Their Variances

As we have seen, finding an optimal unbiased estimator of $T_2 - T_1$ and determining its variance is an equivalent problem to computing the distribution of currents, and the corresponding effective resistance, when a unit of current is injected at node s and extracted at node t of a resistive network. A standard approach is to set up and solve a system of linear equations for the currents and potentials using Ohm's Law and Kirchhoff's Law.

In certain special cases the effective resistance can be determined analytically. For example, in the infinite d-dimensional grid with unit resistors and unit distance between neighboring nodes, the effective resistance between two nodes at Manhattan distance L is $O(\log L)$. Thus, in the clock synchronization problem corresponding to this network, our approach would yield an estimator with variance $O(\log L)$ whereas RBS, which bases its estimator on a single path, would yield an estimator with variance L.

3.7 A PTAS for Effective Resistance

An alternate approach is to use the formulation of effective resistance as a flow problem in which each edge has unbounded capacity and cost quadratic in the flow. The quadratic edge costs can then be approximated by piecewise-linear functions, yielding a flow problem with finite capacities but linear costs [10]. Pursuing this idea, we have shown that the effective resistance can be approximated within relative error ϵ by performing $\sqrt{\frac{V}{\epsilon R}}$ flow augmentations in the linear-cost network, where V is the sum of the resistances and R is the effective resistance. Moreover, this bound can be achieved without knowing R in advance.

Given a resistive network $G = ([n], S, V)$, we denote by R the sought by V the sum of the resistances V_{ij} of all edges of G.

Theorem 5. *R can be approximated within $\epsilon > 0$ by $\sqrt{\frac{V}{\epsilon R}}$ flow augmentations.*

Proof: For any positive real F, let $Q(F)$ be the problem of finding a flow of value F that minimizes the quadratic objective function $\sum_{i,j} f_{ik}^2 V_{ij}$ over all flows $\{f_{ij}\}$ of value F from s to t, and let $C^*(F)$ be the cost of a minimum-cost solution to $Q(F)$. Notice that the effective resistance is $C^*(1)$, and $C^*(F) = F^2 C^*(1)$.

For each F we shall define a linear-cost network flow problem $L(F)$ in which the quadratic objective function of $Q(F)$ is replaced by a piecewise-linear approximation which becomes very tight when F is sufficiently large.

The piecewise-linear function $G(x)$ is defined as follows: $G(0) = 0$; for any odd positive integer $2t + 1$, $G(2t+1) = 4t^2$; over the interval [0,1] and each interval $[2t+1, 2t+3]$ G is linear. Then, for all nonnegative x, $G(x) \leq x^2 \leq G(x) + 1$.

For any positive real F let $L(F)$ be the problem of minimizing $\sum_{ij} G(f_{ij}) V_{ij}$ over all flows from r_1 to r_2 of value F. Let $D(F)$ denote the cost of an optimal solution of $L(F)$. Then $D(F) \leq C^*(F) \leq C^*(F) + V$, since a minimum-cost flow in $L(F)$ will have cost less than or equal to $D(F) + V$ with respect to the quadratic cost function of $Q(F)$.

Our goal is to compute a solution to $Q(F)$ of cost less than or equal to $(1 + \epsilon)C^*(F)$. By the above inequalities, it suffices to take an optimal flow for $L(F)$, for any F greater than or equal to $\sqrt{\frac{V}{\epsilon R}}$. Since R is initially unknown, we will solve the sequence of problems $L(1), L(2), L(3), \cdots$ until a solution is found that can be verified to solve some $Q(F)$ within the approximation ratio $1 + \epsilon$. This solution, scaled down by the factor F, provides the required approximate solution to the original Problem $Q(1)$.

To solve this sequence of linear-cost flow problems we construct a network in which, between any pair (r_i, r_j) of adjacent vertices there are (in principle) infinitely many parallel edges. The first of these has capacity 1, and each each subsequent edge has capacity 2. The cost coefficient of the first edge is 0, and the cost coefficient of the tth subsequent edge is $4tV_{ij}$. The cost of a flow of f along an edge is f times the cost coefficient of the edge. In an optimal solution to this linear network flow problem for a specified flow value F, the flow through this set of parallel edges will exhaust the capacities of these edges in increasing order of their cost coefficients. It is easy to check that this linear network flow problem is equivalent to Problem $L(F)$.

If we start with the zero flow and repeatedly augment the flow by sending one unit of flow along a minimum-cost flow-augmenting path from r_1 to r_2 then, after F augmentations, we will have a minimum-cost flow for $L(F)$. The computational cost of each augmentation is $O(m)$, where m is the number of edges in the resistor network for the original problem $Q(1)$.

For each successive F, an optimal flow for $L(F)$ is computed, and its cost is computed both in $Q(F)$ and in $L(F)$. When for some F, the ratio of these costs is less than or equal to $1 + \epsilon$, the current flow (scaled down by the factor F) achieves the desired approximation ratio, and the algorithm halts. This will happen after at most $\sqrt{\frac{V}{\epsilon R}}$ flow augmentations. In the case where all V_ij are equal to 1, $V = m$ and $R \geq 1/m$, so the number of flow augmentations is at most $m\epsilon^{-1/2}$, and the execution time of the polynomial-time approximation scheme is at most O($m^2\epsilon^{-1/2}$).

4 Optimal Synchronization Design

An interesting problem raised by the results above is to select a subset of the set of available signals, and their rate of synchronization messages, so as to minimize the energy consumption required to achieve a specified precision in the estimates of all offsets $T_i - T_j$. We formulate this problem as a continuous nonlinear optimization problem, and present a polynomial-time algorithm, based on the ellipsoid method, for approximating the solution to any desired accuracy.

We associate with each signal s_k a real variable x_k giving the frequency with which the signal is repeated. We assume that successive repetitions are independent, so that the composite signal obtained by averaging x_k repetitions of s_k reduces the variance of each measured value y_{ik} by the factor x_k, yielding a variance of $\frac{V_{ik}}{x_k}$. We also assume that the rate of power consumption for the network is proportional to the sum of the x_k.

We wish to minimize $\sum_k x_k$ subject to the requirement that, for the corresponding set of variances $\frac{V_{ik}}{x_k}$, the effective resistance between each pair of receivers is at most a specified value α. The joint choice of all the variables x_k yields a point in a euclidean space of dimension equal to the number of signals s_k. For a given pair r_i, r_j of receivers, let K_{ij} be the set of points in this space for which the effective resistance between r_i and r_j is less than or equal to α. Let K be the intersection of all the sets K_{ij}. Thus our *synchronization design problem* is:

$$\min \sum_k x_k$$

$$\text{subject to } \; x \in K$$

.

Note that, because dividing all the resistances in a network by a factor t reduces the effective resistances by that same factor, the optimal choice of x for a bound β on the effective resistances is obtained from the optimal choice for α simply by multiplying each x_k by $\frac{\alpha}{\beta}$.

The following can be shown: each set K_{ij} is convex and possesses a polynomial-time *separation oracle*; *i.e.*, a polynomial-time algorithm which, given any point p not in K_{ij}, returns a hyperplane separating p from K_{ij}. It follows at once that K is convex and possesses a polynomial-time separation oracle. Given these facts, the following theorem is a consequence of general results on separation vs. optimization due to [GLS]. Full details will be given in the final paper.

Theorem 6. *The optimum solution to the synchronization design problem can be approximated to any desired accuracy in polynomial time.*

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: (1) generalizing the theory to compensate for clock skew and (2) 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). 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 this is exactly the form of Equation 1 with the following substitutions:

- $y_{ik} \to \log w_{ik}$
- $U_k \to \log W_k$
- $T_i \to \log \alpha_i$
- $e_{ik} \to \delta_{ik}$
- $V_{ik} \to X_{ik}$

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 (which can be achieved by applying RBS initially without our global modifications) 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.

Our first planned real-world deployment is for an ad-hoc deployable distributed array for detecting seismic activity. Seismologists often perform source localization through coherent beam-forming, requiring time consistency within the array of order 10 microseconds. Traditionally, all nodes in a seismic array are time synchronized using satellites in the Global Positioning System, which provides the international UTC timescale to sub-microsecond precision. Interest in network time synchronization has grown because it allows instrumentation of areas that are seismically interesting but inaccessible to GPS (*e.g.*, within structures, canyons, or tunnels). As an array grows in network diameter, existing RBS implementations may prove insufficient because RBS does not optimize for global coherence that, unlike some other sensor network applications, is required for seismic arrays. This makes it an ideal test application for our scheme.

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. We would also like to thank Deborah Estrin for stimulating conversations on this topic.

References

[1] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation – Numerical Methods*. Prentice Hall, 1989. ISBN 0-13-648759-9.

[2] Vladimir Bychkovskiy, Seapahn Megerian, Deborah Estrin, and Miodrag Potkonjak. 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] Flaviu Cristian. Probabilistic clock synchronization. *Distributed Computing*, 3:146–158, 1989.

[4] Jeremy Elson. *Time Synchronization in Wireless Sensor Networks*. PhD thesis, University of California, Los Angeles, 2003.

[5] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine–grained network time synchronization using reference broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, pages 147–163, Boston, MA, December 2002.

[6] Saurabh Ganeriwal, Ram Kumar, Sachin Adlakha, and Mani B. Srivastava. Network-wide Time Synchronization in Sensor Networks. Technical report, University of California, Dept. of Electrical Engineering, 2002.

[7] R. Gusell and S. Zatti. The accuracy of clock synchronization achieved by TEMPO in Berkeley UNIX 4.3 BSD. *IEEE Transactions on Software Engineering*, 15:847–853, 1989.

[8] Jason Hill and David Culler. A wireless embedded sensor architecture for system-level optimization. Technical report, U.C. Berkeley, 2001.

[9] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, pages 93–104, Cambridge, MA, USA, November 2000. ACM.

[10] T.C. Hu. Minimum cost flows in convex cost networks. *Naval Research Logistics Quarterly*, 13(1):1–9, 1966.

[11] E.L. Lehmann. *Theory of Point Estimation*. Chapman and Hall, New York, 1991.

[12] Cheng Liao, Margaret Martonosi, and Douglas W. Clark. Experience with an adaptive globally-synchronizing clock algorithm. In *Eleventh Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA '99)*, pages 106–114, New York, June 1999.

[13] David L. Mills. 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.

[14] M. Mock, R. Frings, E. Nett, and S. Trikaliotis. Continuous clock synchronization in wireless real-time applications. In *The 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00)*, pages 125–133, Washington - Brussels - Tokyo, October 2000. IEEE.

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

[16] T. K. Srikanth and Sam Toueg. Optimal clock synchronization. *J-ACM*, 34(3):626–645, July 1987.

[17] W. Su and I. Akyildiz. Time-Diffusion Synchronization Protocol for Sensor Networks. Technical report, Georgia Institute of Technology, 2002.

[18] P. Veríssimo and L. Rodrigues. A posteriori agreement for fault-tolerant clock synchronization on broadcast networks. In Dhiraj K. Pradhan, editor, *Proceedings of the 22nd Annual International Symposium on Fault-Tolerant Computing (FTCS '92)*, pages 85–85, Boston, MA, July 1992. IEEE Computer Society Press.

[19] Paulo Verissimo, Luis Rodrigues, and Antonio Casimiro. Cesiumspray: a precise and accurate global time service for large-scale systems. Technical Report NAV-TR-97-0001, Universidade de Lisboa, 1997.

[20] Kamin Whitehouse and David Culler. 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.