Network Time Synchronization Using Clock Offset Optimization

Omer Gurewitz, Israel Cidon and Moshe Sidi
Electrical Engineering Department
Technion, Haifa 32000
Israel

Abstract

Time synchronization is critical in distributed environments. A variety of network protocols, middleware and business applications rely on proper time synchronization across the computational infrastructure and depend on the clock accuracy. The "Network Time Protocol" (NTP) is the current widely accepted standard for synchronizing clocks over the internet. NTP uses a hierarchical scheme in order to synchronize the clocks in the network. In this paper we present a novel non-hierarchical peer-to-peer approach for time synchronization termed CTP - Classless Time Protocol. This approach exploits convex optimization theory in order to evaluate the impact of each clock offset on the overall objective function. We define the clock offset problem as an optimization problem and derive its optimal solution. Based on the solution we develop a distributed protocol that can be implemented over a communication network and prove its convergence to the optimal clock offsets. For compatibility, the CTP may use the exact format and number of messages used by NTP. We also present methodology and numerical results for evaluating and comparing the accuracy of time synchronization schemes. We show that the CTP substantially outperforms hierarchical schemes such as NTP in the sense of clock accuracy with respect to a universal clock, without increasing complexity.

1. Introduction

Common distributed computation systems consist of a collection of autonomous entities linked via an underlying network and do not share a common memory or a common clock. They are equipped with distributed system software that enables the collection to operate as an integrated facility. They allow the sharing of information and resources over a wide geographic spread and they are many times superior to traditional centralized systems in terms of sharing, cost and growth. Clock synchronization is a critical piece of the infrastructure for any such distributed system.

The notion "clock synchronization" relates to at least two different aspects of coordinating distant clocks. The first aspect is "frequency synchronization" which relates to the task of adjusting the clocks in the network to run with the same frequency. The second is "time synchronization" which relates to the task of setting the clocks in the network so that they all agree upon a particular epoch with respect to a Universal Time-Coordinated (UTC).

The basic difficulty in clock synchronization is that timing information tends to deteriorate over time and distance. Particularly when the frequencies of two clocks are not identical and are not known in advance. Even if the two clocks were initially time synchronized, over time they are drifting apart, hence they need to be time-synchronized from time to time. Moreover, when two remote computers are exchanging timing information, there is cumulative loss of accuracy along the path traversed by the messages exchanged, unless message transmission time is known precisely.

The application of time synchronizing in distributed systems is diverse. Server log files are used in firewall, VPN security-related activity, bandwidth usage and various logging, management, authentication, authorization and accounting functions. Since they are a collection of information from different hosts, it is essential that the time stamps be correct in order to coordinate the time of network events, which helps understand and track the time sequence of network events. For example, Cisco routers use clock synchronization in order to compare time logs from different networks for tracking security incidents, analyzing faults and troubleshooting [1].

Wireless ad-hoc networks make particularly exten-

Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)
1092-1648/03 $17.00 © 2003 **IEEE**

sive use of synchronized time. In addition to the basic requirements of traditional distributed systems, ad-hoc networks also use time synchronization for mobility prediction [2] or in sensor network for velocity estimations [3], source localization, or to suppress redundant messages by recognizing that they describe duplicate detections of the same event by different sensors.

Global Positioning Systems (GPS) provide accurate time synchronization but are scarce in computer networks. Moreover, an embedded GPS requires continuous reception of multiple satellites which is hard to accomplish indoors or at secured data centers.

Network Time Protocol (NTP) is the current standard for synchronizing clocks on the Internet [4], [5], [6]. NTP is designed to distribute accurate and reliable time information to systems operating in diverse and widely distributed internetworked environment. The architecture, protocols and algorithms establish a distributed subnet of time servers, operating in a self organizing, hierarchical configuration where clocks are synchronized to Universal Time-Coordinated (UTC). NTP suggests data filtering and peer selection algorithms in order to reduce the offset which is the time difference between the clock and the "Universal Time".

The main contribution of our paper is the introduction of the *CTP - the Classless Time Protocol* that reduces offset errors using a novel non-hierarchical approach that uses a peer to peer protocol in which each node sends and receives probe packets only to and from its neighbors to conduct measurements and adjust its clock accordingly. The approach exploits convex optimization theory to evaluate the impact of each clock offset on the overall objective function. We present a set of clock adjustments which provide the optimal solution of a related optimization problem and suggest a methodology in order to evaluate the global accuracy of the synchronization. Using numerical analysis we show that the CTP substantially outperforms the hierarchical schemes in terms of clock accuracy while preserving s similar protocol complexity.

The paper is organized as follows: In Section 2 we present the model used throughout the paper. Section 3 discusses the underlying methodology and introduces the underlying optimization problem. Section 4 contains the analysis and presents the optimal clock assembly. We then propose in Section 5 the CTP - a distributed protocol that converges to the optimal solution. Finally, numerical results are given in Section 6 which demonstrate the performance of the CTP, compare it with other schemes and show its advantages. The paper is conluded with a discussion section.

2. The Model

The goal of this paper is to introduce a novel distributed approach for time synchronization between each clock in the network with a "Universal Time-Coordinated" (UTC) which is the local time in a group of nodes which will be called the reference time nodes. For our analysis, we assume that the errors accumulated because of skew between the clocks is negligible while the synchronization is taking place, hence throughout this work clock synchronization means time synchronization with the UTC. However, CTP is still applicable to networks where clock drifts are presented as long as CTP is operated frequently enough.

We split the model description into three aspects: the network, the delay and the measurements. We begin by introducing the network model that is used. We end the section with a brief description of NTP.

2.1. The Network Model

A communication network is composed of a set of entities which are connected by physical links. Naturally not all entities are interested in synchronizing their clocks, while others may not be capable of participating in the protocol. We will focus throughout this paper on an underlying network which consists of the entities that do participate in the clock synchronization protocol. The participating entities will be called nodes and denoted by Λ_i for node i. Let $\mathcal{N}$ denote this set of nodes and let $N = |\mathcal{N}|$ be the number of nodes. We define a directed link between two nodes as a directed path between the two nodes that does not contain any other node in $\mathcal{N}$. The directed link connecting nodes Λ_i and Λ_j will be denoted by e_{ij} and the collection of all links by $\mathcal{E}$. Note that each link can be composed of several physical segments. We will assume throughout the paper that all links are bidirectional, namely if $e_{ij} \in \mathcal{E}$, then $e_{ji} \in \mathcal{E}$ (if e_{ij} exists so does e_{ji}). Let us also denote by G_i the set of nodes which are node Λ_i's neighbors in the underlying network, i.e., one link away from node Λ_i, and let $|G_i|$ be the number of such neighbors.

We consider a model in which only one out of the N nodes is a "reference time node" (generalization for several reference time nodes is straightforward); this "reference time node" will be denoted by Λ_0.

Since clock synchronization is based on measurements taken by each node using probe packets, it is highly dependent on the delay experienced by these probe packets. In the next subsection we will concentrate on the delay characteristics of the model.

2.2. The Delay Model

The problem of synchronizing clocks is highly related to the problem of measuring one way link delays. If the clocks of the two nodes at both ends of a link are synchronized, the task of measuring one way link delay is simple: one end node sends a probe packet with its time stamp on it; the difference between the arriving time and the transmission time is the one way link delay. Similarly, if the exact one-way link delay on a specific link is known, the task of synchronizing the clocks at the two nodes on both ends of the link is simple: one end node sends a probe packet with its time stamp on it; the difference between the arriving time and the transmission time minus the link delay is the two clocks' offset. In this subsection we concentrate on the one-way link delay model and its measurement.

Due to the nature of delay, link delays cannot be negative. They may however have a minimum value greater than zero. A common approach is to divide the delay into two basic components: The constant component is the minimum delay and is usually associated with the propagation delay; the variable component is usually related to the queueing delay.

For our analysis, we assume that the two directions of a link connecting any two nodes in $\mathcal{N}$ are symmetric in the sense of capacity and distance. Therefore, the constant component of the delay in the two directions is the same (the propagation delay on the physical links comprising the logical link is the same in both directions). We will not assume, though, that the traffic load (queueing delay) in the two directions is identical and we have no knowledge regarding any dependence. Consequently, in our model the total delay (propagation + queueing) in the two directions is asymmetric, where the minimum that can be obtained in the two directions is the same. Note that CTP (like NTP) also works in situations where the propagation delays in both directions are asymmetric (but its objective function may need to be changed).

2.3. The Measurements

Our goal is to synchronize the nodes in the network with the reference node Λ_0. The synchronization is based on measurements taken by each node. This is carried out in the manner suggested by NTP [4], [5], [6]: Each node is continuously sending probe packets (NTP packets) every so often to each one of its neighbors (other nodes or reference time nodes). Time is stamped on packet k by the sender Λ_i upon transmission (T^k_i). The receiver Λ_j stamps its local time both upon receiving a packet (R^k_j), and upon retransmitting the packet back to the source (T^k_j). The source Λ_i stamps its local time upon receiving the packet back (R^k_i). Each packet k will eventually have four time stamps on it: T^k_i, R^k_j, T^k_j and R^k_i. Such time stamps are part of standard NTP messages[1]. We intend to estimate the clock offset by looking at the n most recent packets. We assume that all packets transmitted by a node are delivered to its neighbors, and in the same order as they were transmitted.

For each link $e_{ij} \in \mathcal{E}$ connecting the two nodes Λ_i and Λ_j, let $x^k_{i,j}$ be the one-way link delay experienced by probe packet k while traveling from node Λ_i to Λ_j. The round trip delay of probe packet k between the nodes Λ_i and Λ_j, which is the sum of the two one way link delays will be denoted by RTT^k_{ij} ($RTT^k_{ij} = x^k_{i,j} + x^k_{j,i}$). The local time at node Λ_i when the time according to the "Universal Time" is t_0 shall be denoted by $Time_i(t_0)$; obviously $Time_0(t_0) = t_0$. The clock offsets from the "Universal Time" which are the quantities we are after will be denoted by $\hat{\tau}_i$ for each $\Lambda_i \in \mathcal{N}$. Note that $\hat{\tau}_i = Time_0(t_0) - Time_i(t_0)$ $\forall t_0$ (for all t_0 since we assume there is no skew), and $\hat{\tau}_0 = 0$. Let us also denote by ΔT^k_{ij} the time difference between the transmission of probe packet k by node Λ_i, according to node Λ_i clock, and the arriving time of the packet at node Λ_j according to its own clock i.e., $\Delta T^k_{ij} = R^k_j - T^k_i$. Note that the different times are taken according to different clocks which are not necessarily synchronized, hence the computed time ΔT^k_{ij}, is not the delay but rather the one way link delay experienced by probe packet k while traveling between node Λ_i to Λ_j, plus the difference between the two clock offsets,

$$\Delta T^k_{ij} = x^k_{ij} - \hat{\tau}_i + \hat{\tau}_j \qquad (1)$$

Note that ΔT^k_{ij} can take negative values.

We will give a special significance to the packet that experience the minimum delay over each of the directed links ($\forall e_{ij} \in \mathcal{E}$). Therefore, we will give special notation to this packet and all the quantities related to it. Let us denote by P^{ij} the index of the packet which experienced the minimum delay among all transmitted packets over the directed link e_{ij} and by ΔT_{ij} the minimum obtained by it, $\Delta T_{ij} = \Delta T^{P^{ij}}_{ij}$.

[1] Note that it is sufficient to have only two time stamps on each packet, T^k_i and R^k_j, which eliminates the need for sending the packet back by node Λ_j. Obviously, node Λ_j will send its own probe packets which will provide the two other entries T^k_j and R^k_i. We suggest to use four time stamps in order to be compliant with the NTP message format.

2.4. Network Time Protocol (NTP)

When discussing synchronizing clocks in a network, one usually refers to the "Network Time Protocol" (NTP), which is the widely accepted standard for synchronizing clocks in the internet [4],[5],[6]. NTP suggests a complete scheme for synchronizing clocks with respect to the UTC. In this subsection we briefly review a few aspects of NTP which are relevant to this study.

According to NTP, each node Λ_i computes the round trip delay for each probe packet that traverses link e_{ij} based on the four timing fields recorded on the packet. The computed round trip delay for packet k is: $RTT_{ij}^k = (T_i^k - R_j^k) + (T_j^k - R_i^k)$. The node also estimates the clock offset of node Λ_i's clock relative to node Λ_j's clock as: $\frac{1}{2}\left[(T_i^k - R_j^k) - (T_j^k - R_i^k)\right]$. NTP suggests the "minimum filter", which selects from the n most recent samples the sample with the lowest round trip delay; the offset which relates to this sample is the estimated clock offset relative to node Λ_j's clock. This method is based on the observation that the probability that an NTP packet will find a busy queue in one direction is relatively low, and the probability of a packet to find a busy queue in both directions is even lower. Each node estimates its relative clock offset with respect to a selected group of its neighbors clocks, where neighbors which are closer to a reference time node are preferred - giving NTP its hierarchical nature. Averaging on these offsets results in the clock offset relative to the UTC.

3. Methodology

3.1. The Objective Function

The goal of synchronizing clocks in a network is simple. The clocks of all nodes in the network should match the Universal Time-Coordinated (UTC). However, since there is no scheme that can ensure a perfect synchronization, a formalism is needed in order to evaluate how similar clocks are under a suggested synchronization scheme. Such a formalism is also important for comparing the performance of different synchronization schemes. In this subsection we will discuss the methodology we use for synchronizing clocks. We mainly focus on deriving an objective function that should be optimized in order to achieve the best clock synchronization (an evaluation function for assessing the quality of the synchronization).

We formulate the clock synchronization problem as an optimization problem. The variables are the set of clock adjustments, which will be denoted by $\vec{\tau} = \{\tau_1, \tau_2, \ldots, \tau_{N-1}\}$, where τ_i denotes the clock adjustment of node Λ_i. The input for the problem includes all the delay measurements.

The first issue under consideration when choosing an objective function is whether it should be local or global. Our goal is to synchronize all clocks in the network with the universal time; the assessment on how good the protocol is should be based on how close all the clocks are with respect to the universal time. Even if we are only interested in synchronizing a single clock in the network, it is clear that the accuracy of that clock depends on the accuracy of the clocks it is synchronized with, which are most probably its neighbors. The accuracy of these clocks depends upon the accuracy of the clocks they are synchronized with, etc. Hence the accuracy of a single clock with respect to the UTC relies on the accuracy of many clocks in the network. Therefore it does not matter whether we synchronize a single clock or many clocks; the accuracy of the synchronization is a function of the accuracy of many clocks in the network. Consequently, the objective function which evaluates the synchronization scheme should be a global function that takes into account the accuracy of all the clocks that participate in the procedure.

Additional desirable properties of the objective function are that it is well defined for all clock movements $\vec{\tau}$ (since any clock movement is legal), that it is a function of the conducted delay measurements (the only data available) and that it will be easy to compute and implement in a distributed environment.

The only data available when adjusting the clocks is data collected through the NTP measurements. This data is comprised of entries such as ΔT_{ij}^k for each link $e_{ij} \in \mathcal{E}$ and for each probe packet k. In a synchronized network these entries are simply the one way link delays (see 1)). In an ideal network where the only delay experienced by any packet is the propagation delay, the one way link delay in one direction is equal to the delay in the other direction, hence in such an ideal network which is also synchronized, we expect $\Delta T_{ij}^k = \Delta T_{ji}^k = \Delta T_{ij} \quad \forall e_{ij} \in \mathcal{E}\ , k$. Clearly, any clock adjustment influences all the measurements obtained while using this clock, hence when adjusting a clock we should discard or modify previous measurements obtained using this clock. Let us denote by $\Delta T_{ij}^{'}$ the modified entry ΔT_{ij} on the link Λ_i to Λ_j. This entry is influenced by two clocks only, node's Λ_i clock and node's Λ_j clock, which are at the two ends of the link e_{ij}. If we move the clocks at the two nodes, Λ_i and Λ_j by τ_i and τ_j, respectively, the adjusted measurements, $\Delta T_{ij}^{'}$ and $\Delta T_{ji}^{'}$ will be:

$$\Delta T_{ij}^{'} = \Delta T_{ij} - \tau_i + \tau_j \quad ; \quad \Delta T_{ji}^{'} = \Delta T_{ji} + \tau_i - \tau_j \quad (2)$$

It is important to note that the sum $\Delta T'_{ij} + \Delta T'_{ji}$ which in the ideal network is the round trip delay, does not change.

There are some functions that can comply with the properties described. For example, one can choose a function that yields the average clock movement over all possible clock movements [7]. Alternatively, we can take a function that minimizes the maximum link delay in the network, and then the second maximum link delay, etc (Min-Max). Other approaches which are used in similar problems can be used as well [8].

Our proposal is a function that emphasizes the symmetric nature of the propagation delay, and exploit the idea that once in a while there is a probe packet that suffers negligible or even no queueing delay, i.e., we expect that on each link there will be a probe packet in a sequence of trials that after synchronizing the clock, its entries will satisfy $\Delta T'_{ij} \approx \Delta T'_{ji}$ or $\Delta T'_{ij} - \Delta T'_{ji} \approx 0$.

Based on this observation we suggest the objective function to be:

$$\begin{aligned} F(\vec{\tau}) &= \sum_{\forall e_{i,j} \in \mathcal{E}} (\Delta T'_{ij} - \Delta T'_{ji})^2 \\ &= \sum_{\forall e_{i,j} \in \mathcal{E}} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2 \quad (3) \end{aligned}$$

The goal is to minimize $F(\vec{\tau})$ over $\vec{\tau} \in \mathrm{I\!R}^{N-1}$ since all clock adjustments are allowed.

In the next subsection we will further explain why the objective function depends only on the packet that experienced the minimum delay on each link $\Delta T_{ij} = \min_k[\Delta T^k_{ij}]$.

3.2. Measurements Filter

In any network which is not permanently overloaded one expects that once in a while each link will have a probe packet which suffers no queueing delay at all or nearly no queueing delay. The issue is hence how to identify these events. NTP suggests to find the packet pair that suffers the shortest round trip delay, and relate to it as a packet that suffered no queueing delay. Clearly, in a sequence of packet exchanges between two neighbors the probability of a packet pair to suffer no queueing delay in both directions is much smaller than the probability of arbitrary two counter directions packets (not necessarily a pair) to suffer no queueing delay in a different direction. Figure 1 demonstrates that the propagation delay bound obtained by taking minimum delays on each direction of a link separately is better (tighter) than the one obtained by taking the minimum round trip delay obtained by a single packet pair.

By measuring the delay on each directed link separately we increase the probability of hitting or getting closer to the one way propagation delay which will lead to better clock synchronization.

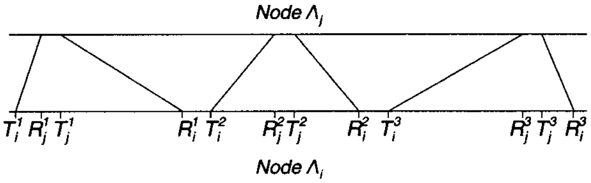

Figure 1. Exchange of three NTP massages between nodes Λ_i and Λ_j. The minimum ΔT_{ij} is obtained by packet 1, $\min_k \Delta T_{ij} = R_j^1 - T_i^1$. The minimum ΔT_{ji} is obtained by packet 3, $\min_k \Delta T_{ji} = R_i^3 - T_j^3$, while the minimum RTT_{ij} is obtained by packet 2, $\min_k RTT_{ij} = (R_j^2 - T_i^2) + (R_i^2 - T_j^2)$. Hence the lower bound on the round trip propagation delay based on the two separate packets that obtained the minimum one way trip delay is lower than that obtain based on the packet which experienced the minimum round trip delay. $R_j^1 - T_i^1 \le R_j^2 - T_i^2$, $R_i^3 - T_j^3 \le R_i^2 - T_j^2$, hence $(R_j^1 - T_i^1) + (R_i^3 - T_j^3) \le (R_j^2 - T_i^2) + (R_i^2 - T_j^2)$

4. Analysis

Recall that our goal is to find the (row) offset vector $\vec{\tau} = (\tau_1, \tau_2, \ldots, \tau_{N-1})$ that minimizes the objective function defined in (3). The feasible domain of the offset vector is $\mathrm{I\!R}^{N-1}$ since all values of clock adjustments are allowed. In order to determine the optimal τ_i's we first prove that there is a unique minimum for the objective function over the feasible domain.

Proposition 1: The objective function given in (3) has a unique global minimum within the feasible domain. The proof of the Proposition is given in the Appendix.

The optimal value of $\vec{\tau}$ which Minimizes (3) can now be obtained by partially differentiate (3) with respect to each variable, $\tau_i \; \forall i \in \{\mathcal{N} \backslash \Lambda_0\}$ ($\tau_0 = 0$ by definition) and equate it to zero.

$$
\begin{aligned}
\frac{\partial F(\vec{\tau})}{\partial \tau^i} &= \frac{\partial}{\partial \tau^i}\left(\sum_{\forall e_{hl}\in\mathcal{E}} (\Delta T_{hl} - \Delta T_{lh} - 2\cdot(\tau_h - \tau_l))^2\right)\\
&= -2\cdot\left(\sum_{\{l|e_{il}\in\mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2\cdot(\tau_i - \tau_l))\right.\\
&\qquad\left. - \sum_{\{l|e_{li}\in\mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2\cdot(\tau_l - \tau_i))\right)\\
&= -4 \sum_{\{l|e_{il}\in\mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2\cdot(\tau_i - \tau_l)) = 0 \quad (4)
\end{aligned}
$$

For all $i \neq 0$ such that $\Lambda_i \in \mathcal{N}$, the equation set described in (4) can be written as:

$$
2|G_i|\cdot\tau_i - \sum_{\{l|e_{il}\in\mathcal{E}\}} 2\tau_l = \sum_{\{l|e_{il}\in\mathcal{E}\}} (\Delta T_{il} - \Delta T_{li}) \quad (5)
$$

The set of equations (5) can be written in a matrix form as:

$$
\vec{\tau}\cdot\mathbf{A} = \vec{\Delta} \quad (6)
$$

where the $(N-1)\times(N-1)$ matrix elements of $\mathbf{A}$ are:

$$
a_{ij} = \begin{cases} 2|G_i| & \text{if } i = j \\ -2\delta_{ij} & \text{otherwise} \end{cases}
$$

with $\delta_{ij} = 1$ if link $e_{ij} \in \mathcal{E}$, and zero otherwise. The raw vectors' $\vec{\tau}$ and $\vec{\Delta}$ elements are simply $\tau(i) = \tau_i$ and $\Delta(i) = \sum_{\{l|e_{il}\in\mathcal{E}\}}(\Delta T_{il} - \Delta T_{li})$ for $i = 1,\ 2,\ \ldots,\ N-1$).

Proposition 2: The solution to (6) exists if and only if $\sum_{\{l|e_{il}\in\mathcal{E}\}}(\Delta T_{il} - \Delta T_{li} - 2\cdot(\tau_i - \tau_l)) = 0\ \forall i \in \mathcal{N}\backslash\{0\}$.
Proof: In Proposition 1 we show that (4) has a unique solution which is the optimal one. Since (4) is equivalent to (6), there is a unique solution to $\sum_{\{l|e_{il}\in\mathcal{E}\}}(\Delta T_{il} - \Delta T_{li} - 2\cdot(\tau_i - \tau_l)) = 0\ \forall i \in \mathcal{N}\backslash\{0\}$, which is the optimal one.

5. The Classless Time Protocol (CTP)

In the previous section we introduced the optimal solution.

Obviously, the most simple solution for optimizing the objective function is to using a centralized protocol. Each node transmits its minimum measurements (ΔT_{ij}) to a centralized entity which collects all the measurements and computes the clock adjustment that should be made by each node according to $\vec{\tau}^T = \mathbf{A}^{-1}\cdot\vec{\Delta}$. The centralized entity transmits to each node the clock adjustment it should perform, as well as the new ΔT_{ij} according to τ_i and τ_j. Each node updates its measurements, and keeps tracking of the link delays (via probe packets). Whenever a lower value for ΔT_{ij} is obtained on one of the links, the entry is modified. Once in a while the nodes update the centralized entity with the modified measurements.

We turn to present a distributed version of the optimization that synchronizes the clocks of the network with respect to a single "reference time node" Since this protocol is not hierarchical and is based on peer-to-peer measurements we call it *CTP - Classless Time Protocol.* The basic structure of CTP is that each node Λ_i, besides node Λ_0, maintains a record in which it holds the entries ΔT_{ij}, ΔT_{ji} and $\Delta_{ij} = T_{ij} - T_{ji}$ for each neighbor $\Lambda_j \in \mathcal{E}$. In order to maintain the record, each node will periodically transmit a probe packet over each of its outgoing links, attain a min ΔT_{ij} and min ΔT_{ji} and change its record accordingly.

The suggested distributed optimization is iterative. There are many iterative methods that can be used [9], [10]. In CTP in each iteration, a subset of nodes, which can include any number of nodes between one node to all nodes beside Λ_0, performs a "Clock Adjustment Procedure". According to this procedure, the node adjusts its clock by $\tau_i = \frac{1}{2|G_i|}\sum_{j\in G_i}\Delta_{ij}$, where $\tau_i > 0$ indicates that the clock should be moved forward and $\tau_i < 0$ indicates clock movement backward. After each clock adjustment, node Λ_i modifies all its record, $\Delta T_{ij}^{new} = \Delta T_{ij}^{old} - \tau_i$, $\Delta T_{ji}^{new} = \Delta T_{ji}^{old} + \tau_i$ and $\Delta_{ij}^{new} = \Delta_{ij}^{old} - 2\tau_i$. In addition, it transmits its clock change to all its neighbors. When node Λ_j receives a notification regarding a clock change performed by one of its neighbors, it modifies the record entries related to this node, $\Delta T_{ji}^{new} = \Delta T_{ji}^{old} + \tau_i$, $\Delta T_{ij}^{new} = \Delta T_{ij}^{old} - \tau_i$ and $\Delta_{ji}^{new} = \Delta_{ji}^{old} + 2\tau_i$ and performs the "Clock Adjustment Procedure". Note that the total record changes performed after each iteration due to the clocks adjustments in both node Λ_i and Λ_j clocks are $\Delta T_{ij}^{new} = \Delta T_{ij}^{old} - \tau_i + \tau_j$, $\Delta T_{ji}^{new} = \Delta T_{ji}^{old} - \tau_j + \tau_i$ and $\Delta_{ij}^{new} = \Delta_{ij}^{old} - 2\tau_i + 2\tau_j$.

Next we show that by performing CTP, the clock offsets will converge to the optimal values, and each clock in the network will converge eventually to the clock that was obtained by executing the centralized protocol. We start by showing that no matter how many nodes adjust their clocks during a single iteration, the objective function $\sum_{e_{ij}\in\mathcal{E}}\left(\Delta T_{ij}^{old} - \Delta T_{ji}^{old} - 2\tau_i + 2\tau_j\right)^2 = \sum_{e_{ij}\in\mathcal{E}}(\Delta_{ij})^2$ is not bigger than prior to the adjustment.

Let us denote by $^{[h]}$ all values that relate to the k-th iteration. For instance, $\tau_i^{[h]}$ denotes the clock adjustment performed by node Λ_i in the k-th iteration, $\Delta_{ij}^{[h]}$ denotes the value of Δ_{ij} after the k-th iteration, etc.

Proposition 3: If a set of arbitrary nodes, denoted by Ψ, move their clock by $\tau_i^{[h]} = \frac{1}{2|G_i|}\sum_{j \in G_i} \Delta_{ij}^{[h-1]}$, the new sum $\sum_{\forall e_{kl} \in \mathcal{E}} (\Delta_{kl}^{[h]})^2$ is not bigger than the sum prior to the adjustment.
The proof is omitted due to space limitations and can be found in [13].

Proposition 4: When the clock adjustment operation is applied by all nodes in all iterations, the set of clocks converges to the set of clocks which minimizes the objective function suggested by (3) i.e., the set of clocks that was obtained by performing the centralized protocol.
The proof is omitted due to space limitations and can be found in [13].

6. Numerical Results

6.1. The Underlying Network

In order to evaluate the accuracy of clock synchronization and convergence rate achieved using CTP, we applied it on a random network topology and compare CTP to several versions of NTP. The network construction is based on a Breadth First Search (BFS) principle. We start with a single "Reference Time Node", restrict the hop distance of each node to the "Reference Time Node" to be at most a certain number of hops. The connectivity between the nodes is randomly selected. The propagation delay of each link is chosen once for both directions of any existing link based on uniform distribution ($\sim U[0, 10]$). The queuing delay of each directed link is chosen as Erlang distribution where the number of exponentials (α) and the mean time between events (θ) are randomly selected between 1 to 10 and between 0.1 to 1, respectively. The parameters are sampled once for each directed link. The clocks' offset with respect to the "Reference Time Node" are randomly chosen with a uniform distribution between -10 to 10 ($\sim U[-10, 10]$).

On each link, eight NTP packets are transmitted as suggested by NTP and ΔT_{ij} are measured based on these packets.

6.2. The Results

We separate the numerical results into three different parts. In the first part we examine the measurement filter based on one way measurements as suggested in Section 3.2. In the second part we evaluate the performance of our scheme by implementing the centralized protocol suggested in Section 5. The third part examines the CTP suggested in Section 5.

We start by investigating the measurement filter. As explained in Section 3.2, by measuring delay separately on each link direction, we increase the probability of finding a packet that experiences no queuing delay or nearly no queuing delay which leads to better clock synchronization.

In Figure 2 we compare the upper bound of the round trip propagation delay obtained by two different methods: 1) Selecting the packet that experiences the minimum round trip delay out of the n recent packets; 2) Based on the same n packets but selecting the two packets that experienced the minimum delay in each direction separately. We examine the results for window size $n = 8$ as suggested in [4]. Since the measurement filter is relevant on a per link basis, we examine it on a thousand nodes network, where over each link only one node is initiating probe packets and estimating the round trip propagation delay while the other node only replies.

Figure 2 shows the distribution of the round trip propagation delay error based on the two methods, i.e., the distribution of the minimum round trip delay experienced by a single packet minus the actual round trip propagation delay, and the distribution of the minimum round trip delay obtained by two packets minus the actual round trip propagation delay. We denote in the graph the two schemes "single packet" and "two packets", respectively.

As expected it can be seen that the measurement filter suggested in Section 3.2 provides a much better (tighter) bound to the propagation delay, which means that the clock adjustment based on it is more accurate. For instance, we observe from the figure that the probability that the error will be less than 1 unit is 0.63 for the "one way method", while it is only 0.36 for the "round trip method". Note that due to the nature of the measurement filter of picking the minimum round trip delay based on two separate measurements, all links, with no exception, attain a bound which cannot be worse than the one attained using the other filter.

Next we examine the clock adjustments ($\vec{\tau}$) that minimize the objective function suggested in Section 3.1. The clock adjustments are determined by applying the centralized protocol suggested in 5. In order to evaluate our results we compare them with three hierarchical schemes.

In the first scheme, denoted by Hierarchical-1, each node selects among its neighbors which are one hop closer to the "reference time node" than itself, the one with the smallest RTT_{ij}, i.e., the neighbor with the lowest round trip delay bound as suggested by NTP. The clock offset is computed as: $\tau_i = \frac{\Delta T_{ij}^k - \Delta T_{ji}^k}{2}$. Node

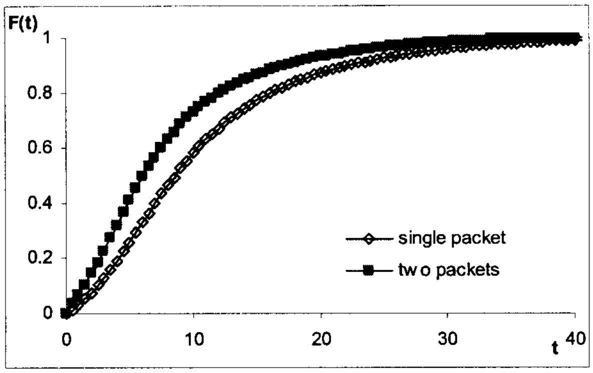

Figure 2. Distribution of delay errors

Λ_i clock is adjusted by τ_i. We start with nodes that are one hop away from the "reference time node", move to nodes that are two hops away from the "reference time node", etc. The second scheme, denoted by Hierarchical-2, is similar to the Hierarchical-1 scheme, but this time ΔT_{ij} and ΔT_{ji} are selected based on the measurement filter suggested in Section 3.2. This modification not only changes the quantity of the offset as demonstrated in Figure 2, but may also change the neighbor for which node Λ_i chooses to adjust its clock in respect with. In the third scheme, denoted by Hierarchical-3, each node computes its clock offsets, $\frac{\Delta T_{ij}-\Delta T_{ji}}{2}$, with respect to all its neighbors which are one hop closer to the "reference time node" than itself. The node moves its clock by the average clock offset. Again ΔT_{ij} and ΔT_{ji} are selected separately. The protocol is hierarchical starting with the nodes that are one hop away from the "reference time node" and advancing till it reaches the nodes that are the furthest from the "reference time node".

We operated the CTP and the three hierarchical schemes in two networks and adjusted the clocks accordingly. Figures 3 and 4 show the results on a 220 and 1317 node networks, respectively. The y axis on each graph presents the fraction of nodes with clock offset, with respect to the UTC, not greater than the value described by the x value.

Figures 3 and 4 clearly demonstrate the significant improvement in terms of clock accuracy of the CTP over all other hierarchical schemes. For example, it can be seen in the graphs that about one third of all nodes in the 220 node network and about 40% of the nodes in the 1317 node network have their clock offset with respect to the UTC not greater than one time unit after performing the CTP. In Hierarchical schemes 1, 2 and 3, only 7%, 15% and 16% for the 220 node network, and 12%, 15% and 17% for the 1317 node network get the same result, respectively.

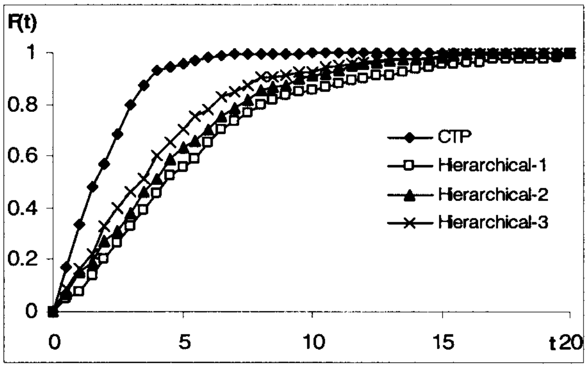

Figure 3. The fraction of nodes with clock offset with respect to the reference time node that is not greater than t, on a 220 node network.

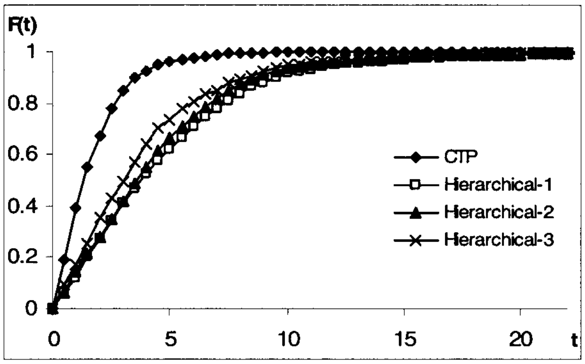

Figure 4. The fraction of nodes with clock offset with respect to the reference time node that is not greater than t, on a 1317 node network.

The third part of our numerical analysis is dedicated to the CTP convergence rate of CTP (in its distributed implementation). We examined the clock offset after 0, 1, 3, 5 and 10 iterations with respect to the optimal solution as given in (6). Figure 5 describes the fraction of nodes with clock offset with respect to the optimal clock offset not greater than t in a 169 node network. We start with a clock offset which is uniformly distributed, hence the offset from the optimal solution varies between 0 to 12 time units (0 iterations). It can be seen in the graph that before we start there are only 8% within half a time unit from the optimal solution. However 35%, 77%, 97%, 99% are within half a time unit from the optimal solution after the first, third, fifth and tenth iteration, respectively.

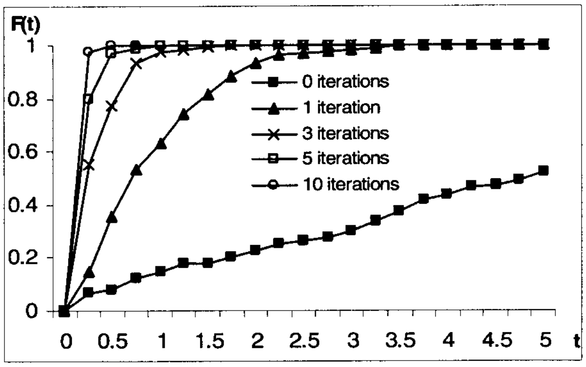

Figure 5. The fraction of nodes in a 169 node network with clock offset with respect to the set of optimal clock offsets (optimal solution) not greater than t, during the implementation of the suggested distributed protocol.

Acknowledgments

The authors would like to thank Prof. Hanoch Levy for suggesting the separation of the round-trip delays to one way components.

Appendix

To prove Proposition 1 we will first prove two simple Lemmas.

Lemma 1: The objective function $F(\vec{\tau})$ given in (3) can be expressed in a quadratic form.

Proof: The objective function (3) given by $F(\vec{\tau}) = \sum_{\forall e_{i,j} \in \mathcal{E}} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2$ can be written in quadratic form as follows:

$$F(\vec{\tau}) = \vec{\tau}\mathbf{P}\vec{\tau}^T + \vec{q}\vec{\tau}^T + r \tag{7}$$

where the $(N-1) \times (N-1)$ matrix elements of $\mathbf{P}$ are:

$$\frac{1}{4}P_{ij} = \begin{cases} |G_i| & \text{if } i = j \\ -\delta_{ij} & \text{otherwise} \end{cases}$$

with $\delta_{ij} = 1$ if link $e_{ij} \in \mathcal{E}$, and zero otherwise. The $(N-1)$ row vector elements of $\vec{q}$ are:

$$\frac{1}{4}q_i = \sum_{\Lambda_j \in G_i} (\Delta T_{ji} - \Delta T_{ij})$$

and

$$r = \sum_{e_{i,j} \in \mathcal{E}} (\Delta T_{ij} - \Delta T_{ji})^2$$

Lemma 2: The matrix $\mathbf{P}$ is a positive definite matrix.

Proof: The matrix $\mathbf{P}$ is a symmetric matrix since $P_{i,j} = P_{j,i} = -\delta_{ij}$. In order to show that it is positive definite we will show that $\vec{\tau}\mathbf{P}\vec{\tau}^T > 0 \quad \forall \vec{\tau} \in \mathbb{R}^{N-1}$ except $\vec{\tau} = \vec{0}$.

$$\begin{aligned} \vec{\tau}\mathbf{P}\vec{\tau}^T &= \sum_{i=1}^{N-1} \left(|G_i| \cdot \tau_i^2 - \sum_{j=1}^{N-1} \delta_{ij}\tau_i\tau_j \right) \\ &= \sum_{e_{i,j} \in \mathcal{E} \setminus \Lambda_0} \left(\tau_i^2 - 2\tau_i\tau_j + \tau_j^2\right) + \sum_{\{e_{i,0} | \Lambda_i \in G_0\}} \tau_i^2 \\ &= \sum_{e_{i,j} \in \mathcal{E} \setminus \Lambda_0} (\tau_i - \tau_j)^2 + \sum_{\{e_{i,0} | \Lambda_i \in G_0\}} \tau_i^2 \end{aligned}$$

Hence $\vec{\tau}\mathbf{P}\vec{\tau}^T \geq 0 \quad \forall \vec{\tau} \in \mathbb{R}^{N-1}$. In order for $\vec{\tau}\mathbf{P}\vec{\tau}^T$ to equal zero τ_i should be equal zero for all $\Lambda_i \in G_0$, and as a consequence all nodes Λ_j which are neighbors of node Λ_0's neighbors ($\Lambda_j \in \{G_i | \Lambda_i \in G_0\}$), etc. Since the network is connected we will have that $\vec{\tau}\mathbf{P}\vec{\tau}^T = 0$ if and only if $\tau_i = 0 \quad \forall \Lambda_i \in \mathcal{N}$ ($\vec{\tau} = \vec{0}$). Hence we conclude that the matrix $\mathbf{P}$ is positive definite.

Proof of Proposition 1: From Lemma 1 that proves that the objective function $F(\vec{\tau})$ has a quadratic form we conclude that $F(\vec{\tau})$ is a convex function. Furthermore, Lemma 2 proves that $\mathbf{P}$ is a positive definite matrix. Consequently, $F(\vec{\tau})$ is a strictly convex function [11], [12].

Since we are adjusting the original measurements (ΔT_{ij}) according to the clock movements, any clock adjustment $\vec{\tau}$ is a round trip delay conserving ($\Delta T'_{ij} + \Delta T'_{ji} = \Delta T_{ij} + \Delta T_{ji}$), hence any $\vec{\tau} = (\tau_1, \tau_2 \ldots, \tau_{N-1}) \in \mathbb{R}^{N-1}$ is feasible. $\mathbb{R}^{N-1}$ is clearly a convex set. Since the objective function is a strictly convex function there exists at most one global minimum of F. Since the objective function is quadratic, the optimal value is attained within the feasible domain.

It is interesting to note that for unconstrained quadratic optimization of the form $F(\vec{\tau}) = \vec{\tau}\mathbf{P}\vec{\tau}^T + \vec{q}\vec{\tau}^T + r$ for the special case in which $\mathbf{P}$ is a positive definite matrix, the unique optimal point is $\vec{\tau}_{opt} = -(\frac{1}{2})\vec{q}\mathbf{P}^{-1}$ and $F(\vec{\tau}_{opt}) = r - (\frac{1}{4})\vec{q}\mathbf{P}^{-1}\vec{q}^T$ [11], [12].

This concludes the proof of Proposition 1.

References

[1] Task 4-Using Syslog, NTP and modem call records to isolate and troubleshoot faults. *Basic Dial NMS Implementation Guide*, Cisco, 2000

http://www.cisco.com/univercd/cc/td/doc/cisintwk /intsolns/dialsol/nmssol/syslog.htm

[2] W. Su, S.-J. Lee and M. Gerla, Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11(1):1099-1190,2001.

[3] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, D. Estrin and L. Girod, Habitat monitoring: application driver for wireless communications technology *Workshop on Data communication in Latin America and the Caribbean*, pp. 20-41, San Jose, Costa Rica, 2001.

[4] D.L. Mills, Internet time synchronization: the Network Time Protocol. *IEEE Trans. Communications*, COM-39, 10:1482-1493, 1991.

[5] D.L. Mills, Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Trans. Networks*, 3(3):245-254,1995.

[6] D.L. Mills, Network Time Protocol (Version 3) specification, implementation and analysis. *Network Working Group Report*, RFC-1305, University of Delaware, 1992, p. 113.

[7] O. Gurewitz and M. Sidi, Estimating One-way Delays from Cyclic-Path Delay Measurements. *IEEE Infocom* 2001, Anchorage, AK, April 2001.

[8] M. Tsuru, T. Takine, Y. Oie, Estimation of clock offset from one-way delay measurement on asymmetric path. *Symposium on Applications and the Internet (SAINT) Workshops*, Narar City, Nara, Japan, January 2002.

[9] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, 3rd Ed. Springer-Verlag, New York, 2002

[10] C.T. Kelley, *Iterative methods for linear and nonlinear equations*, SIAM, Philadelphia, PA, 1995.

[11] R. T. Rockafellar, *Convex analysis*, Princeton University Press, 1972

[12] S. Boyd, L. Vandenberghe, *Convex optimization*, 2002 www.stanford.edu/class/ee364 and www.ee.ucla.edu/ee236b

[13] O. Gurewitz, I. Cidon and M. Sidi, Network Time Synchronization Using Clock Offset Optimization, CCIT Report 430, June 2003.