# Network Classless Time Protocol Based on Clock Offset Optimization

Omer Gurewitz, *Member, IEEE*, Israel Cidon, *Senior Member, IEEE*, and Moshe Sidi, *Senior Member, IEEE*

***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, prove its convergence to the optimal clock offsets and show its properties. For compatibility, CTP may use the packet format and number of measurements 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 outperforms hierarchical schemes such as NTP in the sense of clock accuracy with respect to a universal clock.**

***Index Terms*—Classless time protocol (CTP), estimation, one-way delay, measurements, network management, time synchronization, UTC.**

## I. 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, and allow the sharing of information and resources over a wide geographic spread. 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 time synchronization is that timing information tends to deteriorate over time and distance. Even if 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 packets exchanged, unless packet transmission time is known precisely.

The applications of time synchronization in distributed systems are diverse. Server log files are used in firewalls, 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 in understanding and tracking 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 extensive 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 networks for velocity estimations [3], source localization, or to suppress redundant packets by recognizing that they describe duplicate detections of the same event by different sensors. Clock synchronization algorithms for sensor networks that address specific sensor network issues like power efficiency are presented in [4] and [5].

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 [6]–[8]. 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 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 *CTP—the Classless Time Protocol* that reduces offset errors using a novel non-hierarchical approach that is based on a peer to peer protocol in which each node exchanges probe packets with its neighbors to conduct measurements and adjust its clock accordingly. The approach exploits convex optimization theory

Manuscript received March 9, 2005; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Crovella. This work was supported in part by a grant from the Cisco University Research Program Fund at Community Foundation Silicon Valley.

The authors are with the Electrical Engineering Department, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: gurewitz@rice.edu; cidon@ee.technion.ac.il; moshe@ee.technion.ac.il).

Digital Object Identifier 10.1109/TNET.2006.880181

1063-6692/$20.00 © 2006 IEEE

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 extensive numerical examples we show that CTP outperforms the hierarchical schemes in terms of clock accuracy. To the best of our knowledge, CTP is the first global clock synchronization scheme that explicitly tunes the clock offsets in order to minimize a global network-wide cost function [9]. Recently, two other papers developed global clock synchronization schemes for sensor networks [10], [11].

This paper is organized as follows. In Section II, we present the model used throughout the paper. Section III discusses the underlying methodology and introduces the underlying optimization problem. Section IV contains the analysis and presents the optimal clock assembly. We then propose in Section V the CTP and show that its distributed version converges to the optimal solution. Several important properties of the CTP are given in Section VI. Finally, numerical results are given in Section VII which demonstrate the performance of the CTP, compare it with other schemes, and show its advantages. The paper is conluded with a discussion in Section VIII.

## II. The Model, Assumptions, and Background

The goal of this paper is to introduce a novel approach for time synchronization between each clock in the network with a UTC which is the local time in a group of nodes which will be called the reference time nodes.

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.

### A. 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. 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 $i$ and $j$ 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 $i$'s neighbors in the underlying network, i.e., one link away from node $i$, and let $|G_i|$ be the number of such neighbors.

We start with a model in which only one out of the $N$ nodes is a UTC (generalization for several reference time nodes is discussed in Section VI); this UTC will be denoted by 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.

### B. The Delay

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.

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 that is usually associated with the propagation delay, and the variable component is usually related to the queueing delay.

### C. The Measurements

Our goal is to synchronize the nodes in the network with the reference node. The synchronization is based on measurements taken by each node. This is carried out in the manner suggested by NTP [6]–[8]: 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 $i$ upon transmission to node $j$ ($T_{ij}^{[k]}$). The receiver $j$ stamps its local time both upon receiving a packet ($R_{ij}^{[k]}$), and upon retransmitting the packet back to the source ($T_{ji}^{[k]}$). The source $i$ stamps its local time upon receiving the packet back ($R_{ji}^{[k]}$). Each packet $[k]$ will eventually have four time stamps on it: $T_{ij}^{[k]}$, $R_{ij}^{[k]}$, $T_{ji}^{[k]}$ and $R_{ji}^{[k]}$. Such time stamps are part of standard NTP packets.[1] We intend to estimate the clock offset by looking at the $n$ most recent packets.

Special care should be given to network environments where clock drifts are present. The problem of frequency synchronization is an important issue that is not in the scope of this paper. Similar to NTP, CTP can run properly by removing skew influences from measurements conducted between neighboring nodes, a topic that has been studied in the literature. Therefore, throughout this section we will assume that skew errors were removed from all measurements conducted between neighboring nodes, using any one of the techniques suggested in [12]–[14].

For each link $e_{ij} \in \mathcal{E}$ connecting the two nodes $i$ and $j$, let $x_{ij}^{[k]}$ be the one-way link delay experienced by probe packet $[k]$ while travelling from node $i$ to node $j$. The round-trip delay of probe packet $[k]$ between nodes $i$ and $j$, which is the sum of the two one-way link delays will be denoted by $RTT_{ij}^{[k]}$ ($RTT_{ij}^{[k]} =$

[1] Note that it is sufficient to have only two time stamps on each packet, $T_{ij}^{[k]}$ and $R_{ij}^{[k]}$, which eliminates the need for sending the packet back by node $j$. Obviously, node $j$ will send its own probe packets which will provide the two other entries $T_{ji}^{[k]}$ and $R_{ji}^{[k]}$. We suggest to use four time stamps for compliance with the NTP packet format.

$x_{ij}^{[k]} + x_{ji}^{[k]}$). The local time at node $i$ when the time according to the UTC is $t_0$, shall be denoted by $Time_i(t_0)$; obviously $Time_0(t_0) = t_0$. The actual clock offsets from the UTC, which are unknown, will be denoted by $\hat{\tau}_i$ for each $i \in \mathcal{N}$. 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 $\hat{\tau}_0 = 0$. Note that $\hat{\tau}_i$ can be positive or negative and if node $i$ moves its clock by $\hat{\tau}_i$, then its clock coincides with the clock of the UTC.

Let us also denote by $\Delta T_{ij}^{[k]}$ the time difference between the transmission of probe packet $[k]$ by node $i$, according to node $i$ clock, and the arriving time of the packet at node $j$ according to its own clock, i.e., $\Delta T_{ij}^{[k]} = R_{ij}^{[k]} - T_{ij}^{[k]}$. The different times are taken according to different clocks which are not necessarily synchronized, hence the computed time $\Delta T_{ij}^{[k]}$ is not the one-way delay but rather the sum of the one-way delay experienced by probe packet $[k]$ while travelling from node $i$ to node $j$ and the difference between the two clock offsets

$$\Delta T_{ij}^{[k]} = x_{ij}^{[k]} + \hat{\tau}_i - \hat{\tau}_j. \tag{1}$$

Note that $\Delta T_{ij}^{[k]}$ can take positive and negative values.

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

### D. Network Time Protocol (NTP) Background

The Network Time Protocol (NTP) is the widely accepted standard for synchronizing clocks in the Internet [6]–[8]. NTP suggests a complete and robust solution for clock synchronization with respect to the UTC. Besides the narrow task of synchronizing time with respect to UTC based on trusted measurements, NTP deals with additional issues that relate to the design and the implementation of the protocol. For example, NTP suggests filters and algorithms to discard outliers and filter malicious users. It also addresses authentication and clock design issues. These issues are outside the scope of this study. Since CTP follows NTP in terms of local setup and packet formats we assume the use of the same schemes suggested by NTP. In this subsection we briefly review a few aspects of NTP which are directly dealt with in this study, i.e., the task of estimating clock offsets based on the filtered measurements.

According to NTP, each node $i$ 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 $RTT_{ij}^{[k]} = (R_{ij}^{[k]} - T_{ij}^{[k]}) + (R_{ji}^{[k]} - T_{ji}^{[k]})$. Node $i$ estimates its own clock offset relative to node $j$'s clock as $(1/2)\left[(R_{ij}^{[k]} - T_{ij}^{[k]}) - (R_{ji}^{[k]} - T_{ji}^{[k]})\right]$. 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 $j$'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 UTC are preferred—giving NTP its hierarchical nature. Averaging of these offsets results in the clock offset relative to the UTC.

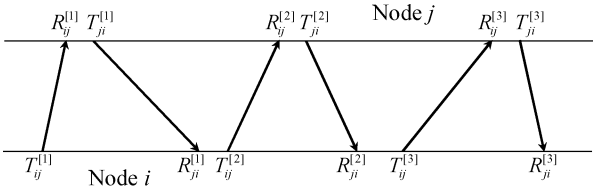


Fig. 1. Exchange of three NTP massages between nodes $i$ and $j$. The minimum $\Delta T_{ij}$ is obtained by packet 1 ($= R_{ij}^{[1]} - T_{ij}^{[1]}$). The minimum $\Delta T_{ji}$ is obtained by packet 3 ($= R_{ji}^{[3]} - T_{ji}^{[3]}$), while the minimum $RTT_{ij}$ is obtained by packet 2 ($= (R_{ij}^{[2]} - T_{ij}^{[2]}) + (R_{ji}^{[2]} - T_{ji}^{[2]})$). 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 obtained based on the packet which experienced the minimum round-trip delay. $R_{ij}^{[1]} - T_{ij}^{[1]} \le R_{ij}^{[2]} - T_{ij}^{[2]},\ R_{ji}^{[3]} - T_{ji}^{[3]} \le R_{ji}^{[2]} - T_{ji}^{[2]}$, hence $(R_{ij}^{[1]} - T_{ij}^{[1]}) + (R_{ji}^{[3]} - T_{ji}^{[3]}) \le (R_{ij}^{[2]} - T_{ij}^{[2]}) + (R_{ji}^{[2]} - T_{ji}^{[2]})$.

## III. Optimization Methodology

### A. The 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 incurs a very small queueing delay. Since propagation delay is fixed, these packets are those which incur the least measurement noise in the form of queueing delay, hence we rely on them in our estimation. The issue is then how to identify these packets.

Since clocks are not synchronized, it is impossible to determine how much of $R_{ij}^{[k]} - T_{ij}^{[k]}\ \forall k$ is due to delay and how much is due to clock offset. NTP suggests to find the packet that incurs the shortest *round-trip* delay, and treat it as if it incurred no queueing delay. Note that for packet round-trip times clock offset influences are eliminated, $\Delta T_{ij}^{[k]} - \Delta T_{ji}^{[k]} = (x_{ij}^{[k]} + \hat{\tau}_i - \hat{\tau}_j) + (x_{ji}^{[k]} + \hat{\tau}_j - \hat{\tau}_i) = x_{ij}^{[k]} + x_{ji}^{[k]}$.

Following [15], one can easily observe that in a sequence of packet exchanges between two neighbors the probability of the same packet sent back and forth over a link to incur small queueing delay in both directions is much smaller than the probability of finding two opposite direction packets (not necessarily the same packet) that both incur small queueing delay. Note that even when dealing with two different packets sent at opposite directions, clock offset is eliminated, $\Delta T_{ij}^{[k]} - \Delta T_{ji}^{[l]} = (x_{ij}^{[k]} + \hat{\tau}_i - \hat{\tau}_j) + (x_{ji}^{[l]} + \hat{\tau}_j - \hat{\tau}_i) = x_{ij}^{[k]} + x_{ji}^{[l]}$. Fig. 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.

Formally, for any sequence of $2n$ real numbers $d_1, d_2, \ldots, d_{2n}$ where the first $n$ numbers represent one-way delay in one direction and the other $n$ represent one-way delay in the opposite direction, it is clear that $\min_{1 \le i \le n}(d_i + d_{i+n}) \ge \min_{1 \le i \le n,\ 1 \le j \le n}(d_i + d_{j+n})$.

By measuring the delay on each directed link separately one increases the probability of hitting or getting closer to the

TABLE I
EXAMPLE OF A LOG OF EIGHT NTP PACKET EXCHANGES BETWEEN NODES $i$ AND $j$

| $k$ | $T_{ij}^{[k]}$ | $R_{ij}^{[k]}$ | $T_{ji}^{[k]}$ | $R_{ji}^{[k]}$ |
|---|---|---|---|---|
| 1 | 8 | 11 | 12 | 16 |
| 2 | 18 | 24 | 25 | 26 |
| 3 | 28 | 31 | 32 | 33 |
| 4 | 38 | 40 | 41 | 42 |
| 5 | 48 | 51 | 52 | 54 |
| 6 | 58 | 61 | 62 | 65 |
| 7 | 68 | 75 | 76 | 76 |
| 8 | 78 | 81 | 82 | 87 |

one-way propagation delay which will lead to better clock synchronization. Table I provides an example of a log of eight NTP packet exchanges between nodes $i$ and $j$. The original NTP measurement filter will pick packet number four as the packet that experienced minimum round-trip delay of 3 time units with offset of 0.5 (see Table I). The modified measurement filter [15] will choose packet number four as the packet that experienced minimum delay on the path from node $i$ to node $j$ and packet number seven as the packet that experienced minimum delay on the path from node $j$ to node $i$; the total round-trip delay is now only 2 time units and the clock offset is 1 (see Table I).

Consequently in the rest of the paper we rely on the modified measurement filter that provides $\Delta T_{ij}$ (the minimum measurement) for nodes $i$ and $j$.

### *B. 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 a good 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 $\tau_i$ denotes the clock adjustment of node $i$. 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. 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.

At this point it seems that a natural choice for a global objective function should be to minimize the accumulated error of all clock adjustments with respect to real clock offsets over all nodes, i.e., $\min_{\vec{\tau}} \left(\sum_{i \in \mathcal{N}} |\tau_i - \hat{\tau}_i|\right)$. Alternatively, for easier analysis, one can take the square instead of the absolute value:

$$\min_{\vec{\tau}} \left(\sum_{i \in \mathcal{N}} (\tau_i - \hat{\tau}_i)^2\right). \tag{2}$$

The problem with any objective function that depends on $\hat{\tau}_i, \ \forall i \in \mathcal{N}$ is that in order to evaluate it or find its optimum we have to know the set of actual clock offsets $\hat{\tau}_i, \ \forall i \in \mathcal{N}$ that are exactly the unknown values we are trying to estimate in the first place. Note that any clock offset is unknown at that node or any other node in the network. In other words, if we knew (locally or at some other place) each clock offset with respect to the UTC, we would know how to perfectly synchronize the clocks without any measurement phase. Consequently, any objective function cannot include explicitly any of the real clock offsets $\hat{\tau}_i, \ \forall i \in \mathcal{N}$.

Additional desirable properties of the objective function are that it is well defined for all clock adjustments $\vec{\tau} \in \mathbb{R}^{N-1}$ (since any clock adjustment 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.

Estimation of the clock offset between a node and the UTC will have to rely on packet exchange between the two nodes. Since queueing delay is accumulated with each additional hop along the path it seems reasonable to try and break the estimation hop chain into smaller units. Our suggestion is that the objective function would be a function of the clock offset differences between neighboring nodes which are the smallest possible units.

The only data available when adjusting the clocks is data collected through the NTP measurements. This data is comprised of entries such as $\Delta T_{ij}^{[k]}$ 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)). Using the modified measurement filter described in the previous subsection we obtain $\Delta T_{ij}$ from these entries.

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 $\Delta T_{ij}$ on the link from $i$ to $j$. This entry is influenced by two clocks only, node's $i$ clock and node's $j$ clock, which are at the two ends of the link $e_{ij}$. If we move the clocks at the two nodes, $i$ and $j$ by $\tau_i$ and $\tau_j$, respectively, the adjusted measurements, $\Delta T'_{ij}$ and $\Delta T'_{ji}$ will be

$$\Delta T'_{ij} = \Delta T_{ij} - \tau_i + \tau_j, \quad \Delta T'_{ji} = \Delta T_{ji} + \tau_i - \tau_j. \tag{3}$$

It is important to note that the sum $\Delta T'_{ij} + \Delta T'_{ji}$, which is the round-trip delay between $i$ and $j$, 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 [16]. Alternatively, one 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 [17].

We suggest the objective function to be

$$F(\vec{\tau}) = \sum_{\forall e_{ij} \in \mathcal{E}} (\Delta T'_{ij} - \Delta T'_{ji})^2 = \sum_{\forall e_{ij} \in \mathcal{E}} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2. \quad (4)$$

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

## IV. 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 (4). The feasible domain of the offset vector is $\mathbb{R}^{N-1}$ since all values of clock adjustments are allowed. In order to determine the optimal $\tau_i$'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 (4) has a unique global minimum within the feasible domain.

The proof of the Proposition is given in Appendix A.

The optimal value of $\vec{\tau}$ which minimizes (4) can now be obtained by partially differentiating (4) with respect to each variable, $\tau_i\ \forall i \in \{\mathcal{N} \setminus 0\}$ ($\tau_0 = 0$ by definition) and equate it to zero.

$$\begin{aligned}\frac{\partial F(\vec{\tau})}{\partial \tau^i} &= \frac{\partial}{\partial \tau^i}\left(\sum_{\forall e_{hl} \in \mathcal{E}} (\Delta T_{hl} - \Delta T_{lh} - 2 \cdot (\tau_h - \tau_l))^2\right) \\ &= -4 \cdot \left(\sum_{\{l \mid e_{il} \in \mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2 \cdot (\tau_i - \tau_l)) \right. \\ &\quad \left. - \sum_{\{l \mid e_{li} \in \mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2 \cdot (\tau_l - \tau_i))\right) \\ &= -8 \sum_{\{l \mid e_{il} \in \mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2 \cdot (\tau_i - \tau_l)) = 0. \quad (5)\end{aligned}$$

For all $i \neq 0$ such that $i \in \mathcal{N}$, the equation set described in (5) can be written as

$$2|G_i| \cdot \tau_i - \sum_{\{l \mid e_{il} \in \mathcal{E}\}} 2\tau_l = \sum_{\{l \mid e_{il} \in \mathcal{E}\}} (\Delta T_{il} - \Delta T_{li}). \quad (6)$$

The set of (6) can be written in a matrix form as

$$\vec{\tau} \cdot \mathbf{A} = \vec{\Delta} \quad (7)$$

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 row vectors' $\vec{\tau}$ and $\vec{\Delta}$ elements are simply $\tau(i) = \tau_i$ and $\Delta(i) = \sum_{\{l \mid e_{il} \in \mathcal{E}\}} (\Delta T_{il} - \Delta T_{li})$ for $i = 1,\ 2, \ldots, N-1$.

*Corollary 1:* In the optimal solution each node satisfies the relation: $\sum_{\{l \mid e_{il} \in \mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2 \cdot (\tau_i - \tau_l)) = 0\ \forall i \in \mathcal{N} \setminus \{0\}$.

*Proof:* In Proposition 1 we show that (5) has a unique solution which is the optimal one. Since (5) is equivalent to (7), there is a unique solution to $\sum_{\{l \mid e_{il} \in \mathcal{E}\}} (\Delta T_{il} - \Delta T_{li} - 2 \cdot (\tau_i - \tau_l)) = 0\ \forall i \in \mathcal{N} \setminus \{0\}$, which is the optimal one. *Q.E.D.*

## V. The Classless Time Protocol (CTP)

In the previous section we introduced the optimal values of the offsets $\tau_i$'s that minimize the objective function (4). Obviously, the most straightforward method to solve the optimization problem is to use a centralized protocol. Each node transmits its minimum measurements ($\Delta T_{ij}$) to a centralized entity (e.g., a network management station) which collects all the measurements and computes the clock adjustments that should be made by each node according to $\vec{\tau} = \vec{\Delta} \cdot \mathbf{A}^{-1}$. The centralized entity transmits to each node the clock adjustment it should perform, as well as the new $\Delta T_{ij}$ according to $\tau_i$ and $\tau_j$. Each node updates its measurements, and keeps tracking of the link delays (via probe packets). Whenever a lower value for $\Delta T_{ij}$ 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. Since this protocol is not hierarchical and is based on peer-to-peer measurements we call it *CTP—Classless Time Protocol*.

A more challenging approach is to synchronize the clocks in a distributed fashion. Fortunately, the CTP can be transformed into a distributed protocol that converges to the optimal offset values as we describe in the sequel. The basic structure of the distributed CTP is that each node $i$, besides node 0, maintains a record in which it holds the entries $\Delta T_{ij}$, $\Delta T_{ji}$ and $\Delta_{ij} = \Delta T_{ij} - \Delta T_{ji}$ for each neighbor $j \in G_i$. In order to maintain the record, each node periodically transmits a probe packet over each of its outgoing links, attains a $\min \Delta T_{ij}$ and $\min \Delta T_{ji}$ and changes its record accordingly.

The suggested distributed optimization is iterative. There are many iterative methods that can be used [18], [19]. In the distributed CTP in each iteration, a subset of nodes, which can include any number of nodes between one node to all nodes beside 0, performs a *clock adjustment procedure*. According to this procedure, the node adjusts its clock by $\tau_i = (1/2|G_i|) \sum_{j \in G_i} \Delta_{ij}$, 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 $i$ modifies all its records, $\Delta T_{ij}^{new} = \Delta T_{ij}^{old} - \tau_i$, $\Delta T_{ji}^{new} = \Delta T_{ji}^{old} + \tau_i$ and $\Delta_{ij}^{new} = \Delta_{ij}^{old} - 2\tau_i$. In addition, it transmits its clock change to all its neighbors. When node $i$ receives a notification regarding a clock change performed by one of its neighbors, say node $j$, it modifies the record entries related to this node, $\Delta T_{ij}^{new} = \Delta T_{ij}^{old} + \tau_j$, $\Delta T_{ji}^{new} = \Delta T_{ji}^{old} - \tau_j$ and $\Delta_{ij}^{new} = \Delta_{ij}^{old} + 2\tau_j$ and performs the "Clock Adjustment Procedure". Note that the total record changes performed after each iteration due to the clocks adjustments in both node $i$ and $j$ 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 $\Delta_{ij}^{new} = \Delta_{ij}^{old} - 2\tau_i + 2\tau_j$. A pseudocode of the distributed CTP is given in [20].

Next we show that by performing the distributed CTP, the clock offsets will converge to the optimal values, and each clock in the network will converge eventually to the clock that would have been 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 larger than prior to the adjustment.

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

*Proposition 3:* If a set of arbitrary nodes, denoted by $\Psi$, move their clock by $\tau_i^{[h]} = (1/2|G_i|)\sum_{j\in G_i}\Delta_{ij}^{[h-1]}$, the new sum $\sum_{\forall e_{kl}\in\mathcal{E}}(\Delta_{kl}^{[h]})^2$ is not larger than the sum prior to the adjustment.

The proof appears in [20].

Finally, we state the proposition regarding the convergence of the distributed CTP.

*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 (4) i.e., the set of clocks that would have been obtained by performing the centralized protocol.

The proof appears in [20].

## VI. CTP Properties

In this section we provide additional properties of CTP that further illustrate its advantages for synchronizing clocks in networks with respect to a UTC and compare some of its properties with NTP.

We begin by showing the effect of having a number of UTCs.

*1) Property 1:* When using the CTP there is no restriction on the number of UTCs, and there can be as many UTCs as one wishes.

*Proof:* Assume we have an $N$ nodes $E$ links network with $m$ UTCs. The objective function is the one suggested in (4), i.e., $F(\vec{\tau}) = \sum_{\forall e_{ij}\in\mathcal{E}}(\Delta T_{ij}-\Delta T_{ji}-2\tau_i+2\tau_j)^2$. The goal is to minimize $F(\vec{\tau})$ over $\vec{\tau}\in\mathbb{R}^{N-m}$ where $\tau_i = 0\ \forall i\in\{\text{set of } m \text{ UTCs}\}$. Let us look at a corresponding $N-m+1$ nodes, $E$ links network. In this network there is only one UTC. The set of links is similar to the set of links in the first network, where any link connecting a node to any UTC is replaced by a corresponding link which connects the node to the single UTC in the corresponding network. The goal now is to optimize $\sum_{\forall e_{ij}\in\mathcal{E}}(\Delta T_{ij}-\Delta T_{ji}-2\tau_i+2\tau_j)^2$ over $\vec{\tau}\in\mathbb{R}^{N-m}$ where $\tau_0 = 0$. The set of equations is exactly the same which means that the same set of measurements $\Delta T_{ij}$, results in the same optimization point with the same set of clock offsets. Furthermore, given a set of measurements where all the UTCs are indistinguishable (all called 0) there is no way of telling to which out of the two networks these measurements belong. Therefore, running the CTP on a network with one or more UTCs without differentiating between the UTCs will yield the right clock offsets which optimize (4). *Q.E.D.*

Next, we show how nodes influence each other when the CTP is performed. To this end, we first define the term *influence*. We say that node $k$ clock is *influenced* by node $j$ if the clock offset obtained by node $k$ after performing the CTP depends on node $j$'s clock offset and the measurements taken by it, i.e., if the value obtained for $\tau_k$ by solving (6), depends on the value obtained for $\tau_j$ and the entries $\Delta T_{ji}\ \forall i\in G_j$. We say that node $j$ *influences* node $k$ clock, if node $k$ clock is influenced by node $j$. We can now state the following property:

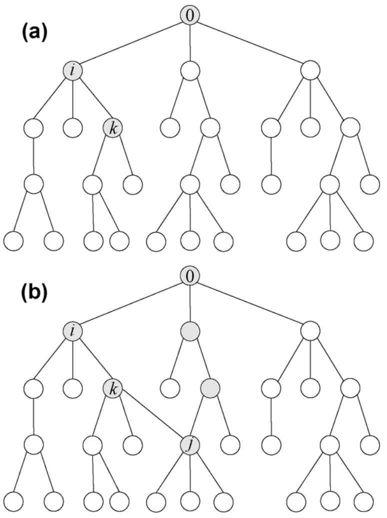


Fig. 2. The nodes that influence node $k$'s clock in tree topology and non-tree topology networks.

*2) Property 2:* Using the CTP node $k$ clock is influenced by another node $i$ only if there exists a simple path from node $k$ to UTC which passes through node $i$.

The proof of this property is given in [20].

The importance of Property 2 is both practical and intuitive. The practical importance is that by knowing the network topology we can compute the clock adjustments separately for the different groups. This aspect is particularly important for the distributed algorithm suggested in Section V since each node should base its clock adjustment only on neighboring nodes that participate in a simple path from it to the UTC.

Property 2 also provides very good insight to the improved results which are presented in Section VII. It also clarifies one of the reasons that makes CTP better than other schemes for most network topologies, and makes CTP comparable to the hierarchical schemes for network topologies which are "tailored" for hierarchical schemes, such as tree topology. For example, consider the tree topology network depicted in Fig. 2(a). The only nodes that influence node $k$ are nodes $i$ and 0 which are the nodes along the path between nodes $k$ and 0. On the other hand, by adding a new link between nodes $k$ and $j$ [see Fig. 2(b)], we add a new simple path between nodes $k$ and 0. Hence, by using the CTP node $k$'s clock will be also influenced by node $j$ and the rest of the nodes along the path.

We now turn to provide some comparison between CTP and NTP. Let us rewrite the clock adjustment performed by node $i$ according to CTP, (6):

$$\tau_i = \frac{1}{|G_i|}\sum_{l\in G_l}\left(\frac{\Delta T_{il}-\Delta T_{li}+2\tau_l}{2}\right). \tag{8}$$

As can be seen, node $i$'s clock adjustment is an average of its estimated clock offset with respect to *all* its neighbors. Note that the term in the parenthesis is similar to the clock offset

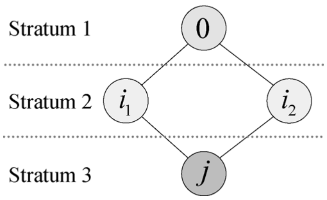


Fig. 3. A simple four-node network topology.

estimation suggested by NTP in the case where node $l$ is one stratum above node $i$, after node $l$ adjusted its clock by $\tau_l$. One may (wrongly) expect that an NTP version which adjusts a clock based on a set of selected parent clocks denoted by $P_i$ for node $i$'s set of selected parents, can be formalized in the same manner as CTP as follows:

$$F(\vec{\tau}) = \sum_{\{e_{ij} | i \in \mathcal{N}, j \in P_i\}} (\Delta T'_{ij} - \Delta T'_{ji})^2. \tag{9}$$

As we will show in the sequel, NTP cannot be formalized in this way. Therefore CTP is not a simple generalization of multiparent NTP.

The minimum attained by (9) or a similar objective function must satisfy that the directional derivative in each direction will equal zero, i.e., $\partial F(\vec{\tau})/\partial \tau_i = 0 \ \forall i \in \mathcal{N}$. Differentiating (9) with respect to $\tau_i$ results in terms originating from both links $e_{i,l}, \ l \in P_i$ and links $e_{k,i}, \ i \in P_k$. To reach any optimum, local or global, $\tau_i$ must depend on both sets $\tau_l, \ l \in P_i$ and $\tau_k, \ i \in P_k$, which means that node $i$ must be influenced both by its set of parent nodes as well as by the set of the *child* nodes, if they exist. The last observation contradicts the hierarchical approach.

To better clarify this point let us use an example that compares the classless and the hierarchical schemes. The topology of the example is depicted in Fig. 3. In the hierarchical scheme even if each node is synchronized based on all its parents, node $i_1$ and node $i_2$ will synchronize solely with respect to node 0. Only node $j$ will synchronize with respect to the two nodes $i_1$ and $i_2$. On the other hand, according to CTP each of the three nodes will be synchronized based on the other nodes. Looking at a numerical example, let us assume that the measured differences are $(\Delta T_{i_1,0} - \Delta T_{0,i_1}) = (\Delta T_{j,i_1} - \Delta T_{j,i_1}) = (\Delta T_{j,i_2} - \Delta T_{i_2,j}) = 4$ and $(\Delta T_{i_2,0} - \Delta T_{0,i_2}) = 8$. The clock offsets computed according to multi parent NTP will be $\tau_{i_1}^{NTP} = 2$, $\tau_{i_2}^{NTP} = 4$, $\tau_j^{NTP} = \frac{1}{2}\left(\frac{4+4}{2} + \frac{4+8}{2}\right) = 5$. According to CTP $\tau_{i_1}^{CTP} = 2.5$, $\tau_{i_2}^{CTP} = 3.5$, $\tau_j^{CTP} = 5$. Note that since $\tau_j$ in both schemes is the average of the cumulative difference along the two paths between $j$ and 0 the result is the same. However, $\tau_{i_1}$ and $\tau_{i_2}$ in NTP is based on a single path to 0 while CTP looks at two disjoint paths to 0, hence the result is not the same.

Note that according to property 2, in the special case of a tree topology CTP coincides with NTP. In the topology described in Fig. 2(a), for example, the outcome from both schemes will be exactly the same. On the other hand, in the topology described in Fig. 2(b), only the set of nodes which are connected to the right branch of node 0 will result in the same clock offset in both schemes; the rest of the nodes including the leftmost leaf will yield different results.

## VII. Numerical Results

In order to evaluate the accuracy of clock synchronization and convergence rate achieved using CTP, we apply it to a variety of network topologies and delays and compare CTP to several versions of NTP.

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

### A. Measurement Filter

We start by investigating the modified measurement filter [15]. As explained in Section III-A, 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.

Since the modified measurement filter is relevant on a per link basis, we examine it on one thousand node pairs connected by a single link, where in each pair only one node is initiating probe packets and estimating the round-trip propagation delay while the other node only replies. Mukherjee and Paxson [24], [25] showed that packet delay along an Internet path is well modeled using shifted gamma distribution. Based on this observation we model all delays as shifted Erlang distribution which is a special case of the gamma distribution in the case that one of the parameters takes only integer values. The shift parameter of each link is chosen based on uniform distribution ($\sim U[0, 10]$). The Erlang parameters $\alpha$ and $\theta$ were randomly selected between 1 to 10 and between 0.1 to 1, respectively. On each link, eight packets are transmitted as suggested by NTP and $\Delta T_{ij}$ are measured based on these packets.

In Fig. 4 we compare the upper bound of the round-trip propagation delay obtained by two different methods: 1) original measurement filter selecting the round-trip packet pair that experiences the minimum round-trip delay out of the $n$ recent packet pairs and 2) modified measurement filter selecting the two one-way packets that experienced the minimum delay in each direction separately. We examine the results based on the same $n$ packet pairs and use window size $n = 8$ as suggested in [6].

Fig. 4 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 is evident that the modified measurement filter suggested in Section III-A provides a 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 5 time units is 0.41 for the "two packets", while it is only 0.26 for the

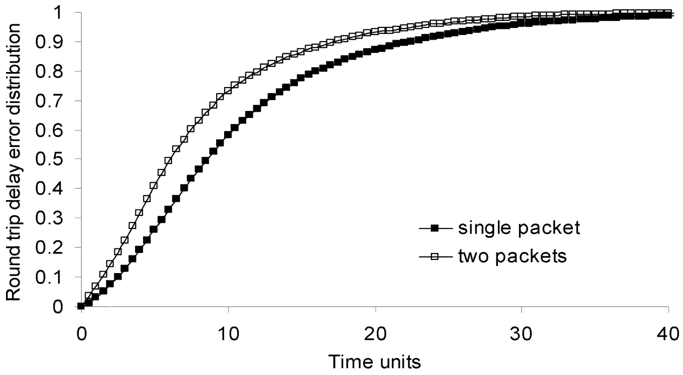


Fig. 4. Distribution of delay errors.

"single packet". Note that due to the nature of the modified measurement filter of picking the minimum sum of opposite directions delay based on two separate measurements, all links, with no exception, attain a value which cannot be worse than the one attained using the original filter.

### *B. CTP Numerical Results*

Next we numerically examine the behavior of the clock adjustments ($\vec{\tau}$) that minimize the objective function suggested in Section III-B. These clock adjustments are calculated by the protocol suggested in Section V.

In order to thoroughly examine CTP we derive the results for a rich variety of network topologies and scenarios. We apply CTP over a random network topology as well as over a specific network example suggested in [29] as explained below.

In order to evaluate CTP, we first need to construct the network topology setup. In particular, we need to model and characterize the overlay network of NTP entities (clients and servers) within private and public networks that are likely to need improved time synchronization as was described in the introduction. While extensive literature about modeling various network topologies (Internet autonomous systems, router topologies, WWW-based topologies etc.) exists, no previous work models the specific overlay network of NTP servers and clients. In the global Internet (which by itself is not necessarily the most important application for CTP), overlay connectivity between any NTP entities (servers and clients) is usually based on administrative configurations. Such administrative selections do not necessarily correlate to any underlying or physical network proximity. Most times, the client may not be even aware of the location of the server it exchanges timing information with. Consequently, when evaluating CTP it is highly reasonable to base it on a random overlay network in which each entity arbitrarily (in the sense of physical location) chooses a list of entities it communicates with (its set of neighbors). However, the topology also needs to take into account the limited depth of NTP hierarchies as reported by the literature. Therefore, the network topology we choose to implement is based on the random model of [27]. Other parameters besides connectivity such as the NTP hierarchy depth, delay parameters, clock offsets etc. were chosen based on the literature, mainly based on an NTP survey conducted at MIT by Minar [23]. Finally, in order to account for other cases like using CTP within routers of wireless base-station backbones we also included a typical ISP backbone topology suggested by Keralapura *et al.* in [29].

The random network construction is based on a Breadth First Search (BFS) principle. We start with a single UTC, restrict the hop distance of each node to the UTC to be at most a certain number of hops. The depth (maximum hop distance from UTC) is selected to be 6 according to [23]. The links between the nodes (within the same hierarchy or between adjacent layers) are randomly selected. As before, the delays are assumed to be distributed according to shifted Erlang distribution which best represents Internet one-way path delay [24]–[26]. The shifted Erlang parameters were selected according to the IEEE 802.20 Working Group [28], and found to be compliant with the NTP survey [23]. The shift associated with the propagation delay of each link is chosen based on uniform distribution ($\sim U[0, 10]$), the number of exponentials ($\alpha$) and the mean time between events ($\theta$), are randomly selected between 1 and 5 and between 0.1 and 3, respectively. The parameters are sampled once for each directed link. As can be seen in the analytic part of the paper, CTP as well as NTP are invariant (in terms of the final clock values) to the initial clocks' offset with respect to the UTC. However for model completeness, based on [23] we chose the offsets to randomly vary with a uniform distribution between $-10$ and 10 ($\sim U[-10, 10]$).

Based on NTP specification [8], eight round-trip packets are sent over each link and $\Delta T_{ij}$ are measured based on these packets.

Note that both CTP and NTP operate better when the *path* between neighboring nodes in the underlying network is symmetric (symmetric path does not necessarily result in symmetric delay measurements). Both schemes adjust clocks based on the difference between the estimated propagation delays at each direction expecting them to be the same after synchronization. We start with symmetric paths, hence in the first set of results the shift in the Erlang distribution (which relates to the propagation delay but **not** the queueing delay), is chosen once for both directions of any existing link. Subsequently we repeat the evaluation for paths with asymmetric propagation delays.

In order to evaluate our results we compare them with three NTP-based hierarchical schemes. In the first scheme, denoted by NTP-1, each node arbitrarily selects a single neighbor which is one hop closer to the UTC than itself. The clock offset is computed as: $\tau_i = (\Delta T_{ij}^{[k]} - \Delta T_{ji}^{[k]})/2$. Node $i$ clock is adjusted by $\tau_i$. We start with nodes that are one hop away from the UTC, move to nodes that are two hops away from the UTC, etc. Note that NTP-1 is the most common implementation of NTP [8], [30]. The second scheme, denoted by NTP-2, is similar to the NTP-1 scheme, but this time $\Delta T_{ij}$ and $\Delta T_{ji}$ are selected based on the modified measurement filter suggested in Section III-A. The third scheme, denoted by NTP-3 is a multi-parent scheme. Here, each node computes its clock offsets, $(\Delta T_{ij} - \Delta T_{ji})/2$, with respect to all its neighbors which are one hop closer to the UTC than itself. The node moves its clock by the **average** clock offset. Again, $\Delta T_{ij}$ and $\Delta T_{ji}$ are selected separately. The protocol is hierarchical starting with the nodes that are one hop away from the UTC and advancing till it reaches the nodes that

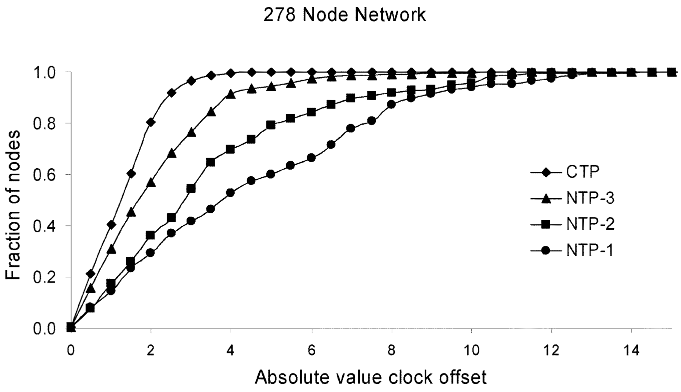


Fig. 5. The fraction of nodes with clock offset with respect to the UTC that is not greater than $x$, on a 278 node network.

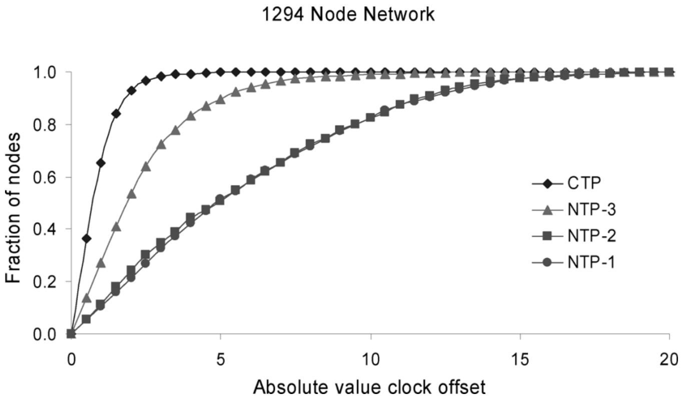


Fig. 6. The fraction of nodes with clock offset with respect to the UTC that is not greater than $x$, on a 1294 node network.

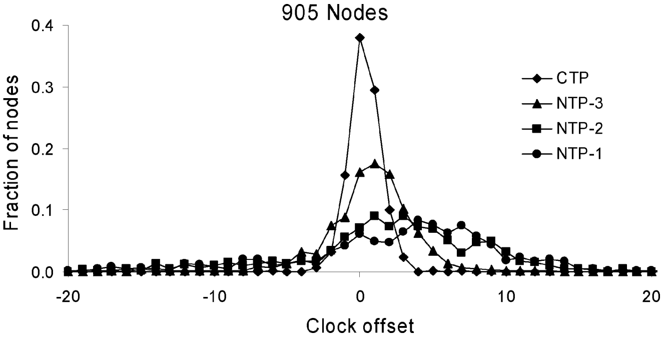
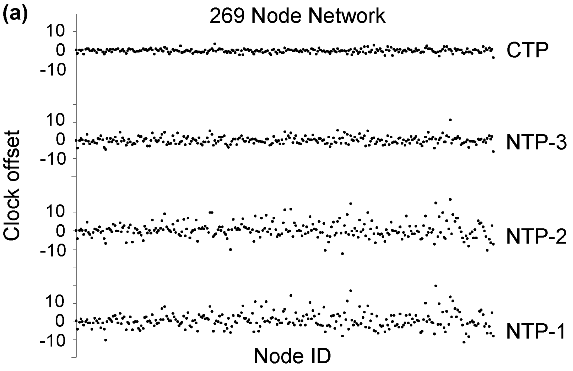


Fig. 7. The fraction of nodes with clock offset with respect to the UTC that is between $x - 1$ and $x$ (PDF), on a 905 node network.

are the furthest from the UTC. NTP-3 is based on the multi-parent NTP suggested in [30].

We operated CTP and the three hierarchical schemes in three networks and adjusted the clocks accordingly. Figs. 5 and 6 show the results on 278 and 1294 node networks, respectively. The $y$ axis on each graph represents the fraction of nodes with absolute value clock offset, with respect to the UTC, not greater than the clock offset depicted by the $x$ value. Fig. 7 depicts the results in a 905 node network. The $y$ axis represents the probability density function (fraction of nodes out of the 905 nodes) with the clock offsets described by the $x$ axis.

Figs. 5–7 clearly demonstrate the significant improvement in terms of clock accuracy of CTP over all hierarchical schemes.



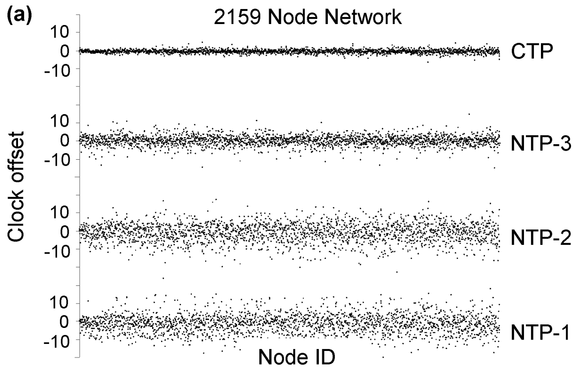
(b)

| | S1 | S2 | S3 | S4 | S5 | S6 | Av |
|---|---|---|---|---|---|---|---|
| CTP | 0.91 | 0.67 | 0.79 | 0.92 | 0.87 | 1.05 | 0.91 |
| NTP-3 | 1.36 | 1.54 | 1.68 | 1.41 | 1.55 | 1.59 | 1.55 |
| NTP-2 | 1.36 | 2.07 | 2.28 | 2.75 | 3.43 | 3.73 | 3.06 |
| NTP-1 | 1.51 | 2.15 | 2.22 | 2.97 | 3.31 | 3.93 | 3.15 |

Fig. 8. (a) The clock offset dispersion on a 269 node network. (b) The distribution of absolute clock difference after synchronization, i.e., $|\tau_i - \hat{\tau}_i|$, between the hierarchies in the 269 node network.

For example, it is evident in the graphs that about 40% of all nodes in the 278 node network and about two thirds of all nodes in the 1294 node network have clock offset with respect to the UTC not greater than one time unit after performing CTP. In NTP-1, -2, and -3, only 14%, 18%, and 31% for the 278 node network, and 10%, 11%, and 27% for the 1294 node network get the same result, respectively. In Fig. 7 it can be seen that after performing CTP 99% of the nodes will have clock offset less than three time units from the UTC and all the nodes will have clock offset less than ten time units from the UTC. Looking at the three hierarchical schemes it can be seen that between $-3$ to 3 time units from the UTC lie only 32%, 44% and 80% of the nodes for the NTP-1, $-2$ and $-3$, respectively. The error bounds for the hierarchical schemes are $[-21.4, 34.9]$, $[-18.3, 29.1]$ and $[-16.4, 13.7]$, respectively.

In order to demonstrate the clock offset dispersion around the UTC clock, we draw graphs 8 and 9. In these graphs, the $x$ axis is the node ID. The $y$ axis is the clock offset with respect to the UTC after performing each scheme. Fig. 8(a) depicts the clock offset dispersion on a 269 node network while Fig. 9(a) relates to a 2159 node network. In both graphs it can be seen, as expected, that CTP which is a global scheme keeps all offsets in a very narrow region which means small errors in the adjusted clocks. The other schemes are characterized by a much wider clock offset domain. Furthermore, CTP keeps the region about the same regardless of the distance from the UTC while in the hierarchical scheme the farther one gets from the UTC (higher node ID), the wider the region is. In order to highlight this we distribute the absolute clock offset between the different distance layers in tables. In Figs. 8(b) and 9(b) each column depicts the average absolute clock offset of all nodes which are a single hop from UTC (S1), two hops from UTC (S2) etc., i.e., $Sk = Avg|\hat{\tau}_i - \tau_i|$, $i \in \{\text{set of nodes which are } k \text{ hops from the UTC}\}$. The last column depicts the average absolute clock offset of all



(b)

| | S1 | S2 | S3 | S4 | S5 | S6 | Av |
|---|---|---|---|---|---|---|---|
| CTP | 0.71 | 0.61 | 0.60 | 0.80 | 0.90 | 1.03 | 0.94 |
| NTP-3 | 1.08 | 2.25 | 2.10 | 2.10 | 2.13 | 2.00 | 2.07 |
| NTP-2 | 1.08 | 2.09 | 2.71 | 3.57 | 4.23 | 4.37 | 4.16 |
| NTP-1 | 0.63 | 1.82 | 2.70 | 3.65 | 4.24 | 4.55 | 4.25 |

Fig. 9. (a) The clock offset dispersion on a 2159 node network. (b) The distribution of absolute clock difference after synchronization, i.e., $|\tau_i - \hat{\tau}_i|$, between the hierarchies in the 2159 node network.

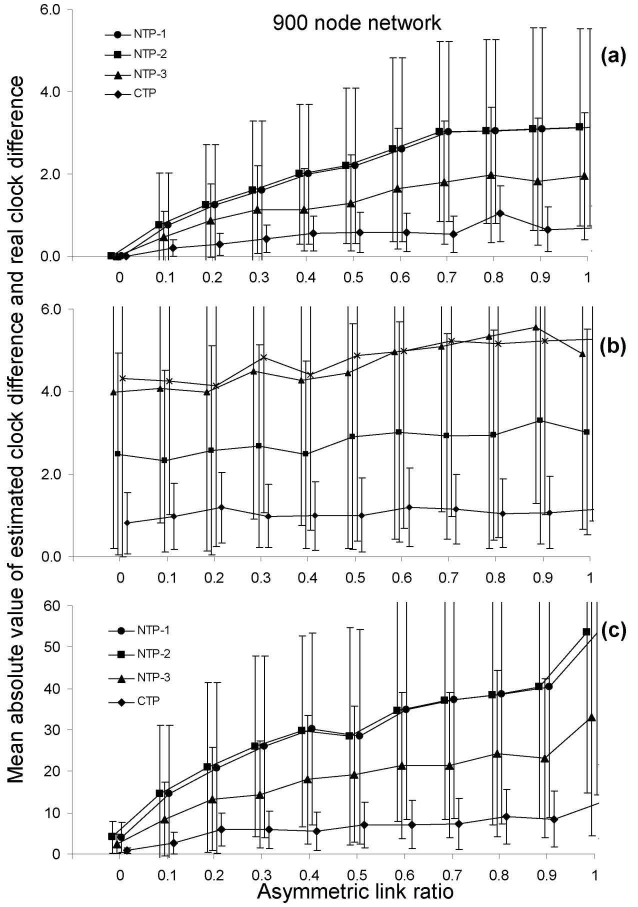


Fig. 10. The mean ± standard deviation of the absolute value of the estimated clock difference and the real clock difference ($|\tau_i - \hat{\tau}_i|$) in the 900 node network with asymmetric paths run. (a) The delay is uniformly distributed. On the asymmetric paths the delay was randomly selected in each direction separately over the same interval. (b) The delays are distributed according to shifted Erlang distribution. The shift on the asymmetric paths was selected separately over the same interval. (c) The delays are distributed according to shifted Erlang distribution. The interval which depicts the shift was dramatically different on the two directions of any asymmetric path.

nodes in the network, i.e., $Av = Avg|\hat{\tau}_i - \tau_i|,\ i \in \mathcal{N}$. The two tables emphasize that CTP is hardly influenced from hop distance from the UTC whereas in hierarchical schemes accuracy deteriorates the farther one gets from the UTC.

Next we address asymmetric networks. We ran the four schemes in a 900 node random network that was constructed in the same manner described before. The network parameters were chosen as before, only this time, we randomly selected some links to be asymmetric. For such an asymmetric link, the Erlang's shift which is associated with the propagation delay was randomly selected for each direction separately. We examined two asymmetric scenarios. In both scenarios the Erlang's shift for both directions over the asymmetric link was uniformly distributed. In the first scenario the uniform interval was the same in both directions ($\sim U[0, 10]$); see Fig. 10(a) and (b). In the second scenario, the uniform interval in the two directions was dramatically different: $\sim U[0, 10]$ in one direction, and $\sim U[0,100]$ in the other direction; see Fig. 10(c).

In order to better clarify the influence of the asymmetric path, we ran the first scenario twice. In addition to the setting described above, we also ran it on a setting where there was no queueing delay, i.e., only propagation delay. In this case the measured delay is exactly the propagation delay shifted by the clock offset and the one-way link delay on symmetric links is exactly $(\Delta_{ij} + \Delta_{ji})/2$, see Fig. 10(a).

We ran the schemes eleven times in each scenario changing the proportion of asymmetric links. Since we could not find any strong evidence for the amount of asymmetry in practical networks we decided to treat this amount as a study parameter. We started with 0% asymmetry which means that all links are symmetric continued with 10% of the links asymmetric and so on up to 100% asymmetry.

Fig. 10 depicts the asymmetric network results. The $x$ axis describes the ratio of asymmetric links, e.g., 0.3 relates to a network with 30% asymmetric links. The $y$ axis describes the average of the absolute value difference between the synchronized clocks and the UTC, $Avg_{i\in\mathcal{N}}|\tau_i - \hat{\tau}_i|$. We also added to the graph the standard deviation based on the 900 node network.

As expected, when there is no queueing delay (the delay is constant), the modified measurement filter is redundant and NTP-1 and NTP-2 coincide, as observed in Fig. 10(a). Notice that in Fig. 10(b) which examines the schemes in the presence of queueing delays, the average hardly changes even as the portion of asymmetric paths increases. This emphasizes that in the model we chose to run our simulations, the queueing delay (Erlang distribution) dominates the results over the propagation delay. It can be seen in Fig. 10(c) that in the unrealistic scenario when the propagation delay is so dramatically different at the opposite link directions, all four schemes do not perform well. However, CTP still outperforms the hierarchical schemes in all three cases.

Next we ran the four schemes on a typical ISP network suggested in [29] depicted in Fig. 11. Both the shift and the Erlang

TABLE II
ABSOLUTE VALUE OF THE ESTIMATED CLOCK DIFFERENCE AND THE REAL CLOCK DIFFERENCE, I.E., $|\tau_i - \hat{\tau}_i|$, ON THE 20 NODE NETWORK DESCRIBED IN FIG. 11. PERCENTAGE OF SYMMETRIC LINKS: (A) 100%; (B) 75%; (C) 50%

(A)

| Node | CTP | NTP-1 | NTP-2 | NTP-3 |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 2.97 | 0.29 | 0.05 | 0.05 |
| 2 | 1.07 | 5.90 | 4.83 | 4.83 |
| 3 | 0.34 | 0.51 | 0.03 | 0.03 |
| 4 | 0.42 | 0.04 | 0.11 | 0.11 |
| 5 | 0.19 | 0.04 | 0.08 | 0.08 |
| 6 | 0.02 | 0.22 | 0.31 | 0.31 |
| 7 | 1.87 | 0.58 | 0.14 | 2.03 |
| 8 | 1.74 | 0.65 | 0.94 | 2.41 |
| 9 | 0.25 | 0.71 | 1.29 | 0.46 |
| 10 | 0.30 | 1.28 | 0.55 | 0.55 |
| 11 | 0.26 | 1.53 | 0.42 | 0.25 |
| 12 | 0.04 | 3.78 | 3.31 | 3.31 |
| 13 | 0.17 | 0.55 | 1.21 | 0.92 |
| 14 | 0.27 | 2.53 | 0.33 | 0.33 |
| 15 | 0.86 | 0.43 | 0.95 | 1.03 |
| 16 | 0.64 | 0.61 | 0.68 | 0.68 |
| 17 | 1.11 | 2.57 | 0.84 | 0.95 |
| 18 | 0.30 | 0.81 | 0.15 | 0.15 |
| 19 | 0.45 | 0.38 | 0.74 | 1.66 |
| | | | | |
| $E$ | 0.66 | 1.17 | 0.85 | 1.01 |
| $\sigma$ | 0.76 | 1.49 | 1.20 | 1.27 |

(B)

| Node | CTP | NTP-1 | NTP-2 | NTP-3 |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 2.34 | 1.08 | 1.07 | 1.07 |
| 2 | 0.20 | 2.30 | 1.93 | 1.93 |
| 3 | 1.01 | 0.00 | 0.01 | 0.01 |
| 4 | 0.41 | 0.56 | 1.42 | 1.42 |
| 5 | 1.41 | 4.23 | 3.01 | 3.01 |
| 6 | 2.19 | 1.94 | 3.07 | 3.07 |
| 7 | 1.49 | 1.32 | 1.44 | 0.80 |
| 8 | 2.42 | 3.88 | 4.90 | 1.10 |
| 9 | 0.76 | 0.97 | 0.17 | 0.64 |
| 10 | 0.74 | 1.55 | 1.60 | 1.60 |
| 11 | 1.68 | 4.70 | 3.72 | 1.98 |
| 12 | 0.61 | 2.97 | 3.62 | 3.62 |
| 13 | 1.02 | 0.19 | 0.04 | 1.94 |
| 14 | 2.17 | 0.68 | 0.54 | 0.54 |
| 15 | 0.72 | 1.04 | 1.02 | 1.48 |
| 16 | 1.28 | 4.85 | 3.64 | 3.64 |
| 17 | 2.00 | 2.27 | 0.60 | 1.24 |
| 18 | 0.22 | 4.15 | 2.85 | 2.85 |
| 19 | 1.15 | 3.52 | 3.08 | 1.16 |
| $E$ | 1.19 | 2.11 | 1.89 | 1.65 |
| $\sigma$ | 0.75 | 1.63 | 1.49 | 1.10 |

(C)

| Node | CTP | NTP-1 | NTP-2 | NTP-3 |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1 | 0.18 | 0.02 | 0.16 | 0.16 |
| 2 | 1.30 | 0.72 | 0.78 | 0.78 |
| 3 | 0.40 | 6.37 | 5.34 | 5.34 |
| 4 | 2.15 | 1.11 | 2.56 | 2.56 |
| 5 | 1.83 | 5.44 | 4.05 | 4.05 |
| 6 | 0.95 | 1.36 | 1.87 | 1.87 |
| 7 | 0.27 | 0.18 | 0.30 | 1.23 |
| 8 | 0.51 | 0.73 | 0.59 | 0.13 |
| 9 | 1.36 | 4.05 | 2.87 | 3.87 |
| 10 | 2.35 | 5.66 | 5.91 | 5.91 |
| 11 | 2.15 | 7.96 | 6.68 | 0.79 |
| 12 | 1.52 | 6.52 | 5.43 | 5.43 |
| 13 | 2.15 | 0.99 | 2.18 | 3.78 |
| 14 | 0.68 | 4.44 | 5.00 | 5.00 |
| 15 | 0.52 | 3.75 | 2.71 | 0.49 |
| 16 | 0.98 | 5.72 | 4.33 | 4.33 |
| 17 | 0.77 | 3.99 | 2.56 | 3.38 |
| 18 | 1.08 | 6.55 | 5.10 | 5.10 |
| 19 | 0.40 | 7.05 | 5.61 | 0.42 |
| $E$ | 1.08 | 3.63 | 3.20 | 2.73 |
| $\sigma$ | 0.74 | 2.72 | 2.15 | 2.10 |

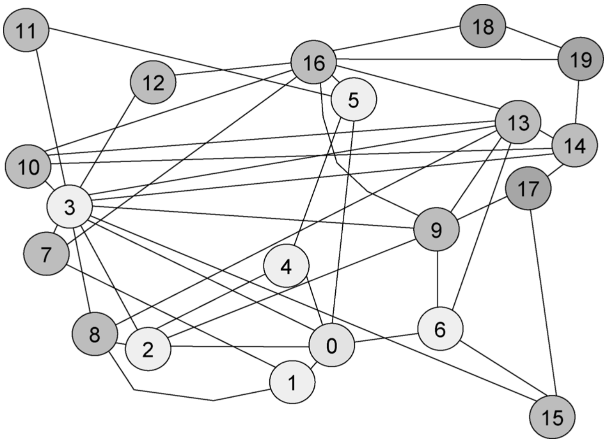


Fig. 11. 20 node typical IP network suggested by [27].

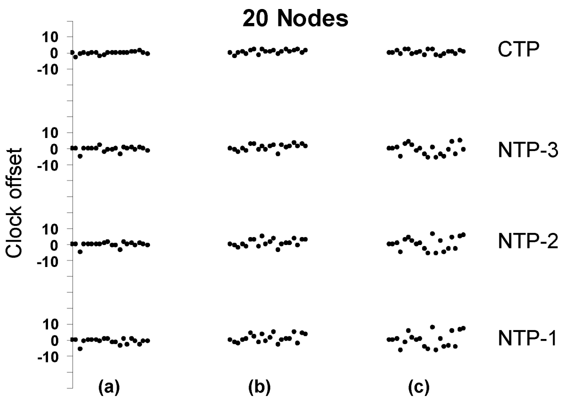


Fig. 12. The clock offset dispersion on the 20 node network described in Fig. 11. Percentage of asymmetric links: (a) 0%; (b) 25%; (c) 50%.

parameters for each directed link are calculated based on the length of the link.

We ran the four schemes three times changing the percentage of the asymmetric links. The three runs consider 0% of the links are asymmetric, 25% asymmetric links and 50% asymmetric links. The results are presented in Fig. 12(a)–(c), respectively.

It can be seen in Fig. 12 that as expected the more symmetric the network is the better the results are for all four schemes. However, it can also be seen in the figure that CTP outperforms the hierarchical schemes for all percentages of link asymmetry.

The values in Table II are the absolute value of the estimated clock difference and the real clock difference, i.e., $|\tau_i - \hat{\tau}_i|$. We also show in the table the average differences of all the twenty nodes and the standard deviation based on the 20 node results.

Even though some of the nodes have better synchronization with one of the hierarchical schemes, generally, CTP outperforms all hierarchical schemes.

### C. Distributed CTP

The third part of our numerical analysis is dedicated to the convergence rate of the distributed CTP. We examined the clock offset after 0, 1, 3, 5, and 10 iterations with respect to the optimal solution as given in (7). Fig. 13 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 10 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%, and 99% are within half a time unit from the optimal solution after the first, third, fifth, and tenth iteration, respectively.

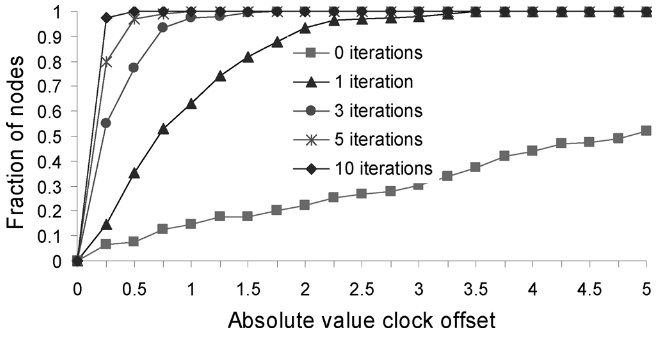


Fig. 13. The fraction of nodes in a 169 node network with clock offset with respect to the set of optimal clock offsets (optimal solution) not greater than $t$, during the implementation of the suggested distributed protocol.

## VIII. Discussion

In this paper, we introduced a new methodology for time synchronization by utilizing an objective function that evaluates the impact of local clock offsets on the overall objective. The suggested objective function is optimized to the case where the capacity and propagation delays of all links is symmetrical (similar to the rationale used by NTP round-trip delay halving). However, it can also be applied to cases where links are asymmetric.

We suggest a protocol for clock adjustments that minimizes the objective function. The suggested solution borrows techniques known in solving optimization problems. Obviously, any additional knowledge regarding the links or clocks in the network can be incorporated as a set of constraints with the proper modifications of solving constrained optimization problems. Our distributed network protocol, CTP, converges to the set of clock adjustments that minimizes the objective function. While there are additional protocols that can be used, we chose a protocol which is easy to implement and requires only minor modifications to the format and number of packets used by NTP. Numerical results illustrate that our approach works well in various randomly chosen networks, and substantially outperforms the hierarchical schemes.

## Appendix

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

*Lemma 1:* The objective function $F(\vec{\tau})$ given in (4) can be expressed in a quadratic form.

*Proof:* The objective function (4) given by $F(\vec{\tau}) = \sum_{\forall e_{ij}\in\mathcal{E}}(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2$ can be written in quadratic form as follows:

$$F(\vec{\tau}) = \vec{\tau}\mathbf{P}\vec{\tau}^T + \vec{q}\vec{\tau}^T + r \qquad (10)$$

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

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

and

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

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

*Proof:* The matrix $\mathbf{P}$ is a symmetric matrix since $P_{ij} = P_{j,i} = -\delta_{ij}$. 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 $\vec{\tau}=\vec{0}$.

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

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 $\tau_i$ should be equal zero for all $i\in G_0$, and as a consequence all nodes $j$ which are neighbors of node 0's neighbors ($j\in\{G_i|i\in G_0\}$), etc. Since the network is connected we will have that $\vec{\tau}\mathbf{P}\vec{\tau}^T = 0$ if and only if $\tau_i = 0 \quad \forall i\in\mathcal{N}$ ($\vec{\tau}=\vec{0}$). 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 [21], [22].

Since we are adjusting the original measurements ($\Delta T_{ij}$) 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 $\vec{\tau}_{opt} = -(1/2)\vec{q}\mathbf{P}^{-1}$ and $F(\vec{\tau}_{opt}) = r - (1/4)\vec{q}\mathbf{P}^{-1}\vec{q}^T$ [21], [22].

This concludes the proof of Proposition 1.

## Acknowledgment

The authors would like to thank Prof. H. Levy for helpful discussions.

## References

[1] "Task 4–Using Syslog, NTP and modem call records to isolate and troubleshoot faults," in *Basic Dial NMS Implementation Guide*. San Jose, CA: Cisco, 2000 [Online]. Available: 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," *Int. J. Network Management*, vol. 11, no. 1, pp. 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," in *Proc. Workshop on Data Communication in Latin America and the Caribbean*, 2001, pp. 20–41.

[4] K. Romer, "Time synchronization in ad hoc networks," presented at the ACM MobiHoc, Long Beach, CA, Oct. 2001.

[5] J. Elson, L. Girod, and D. Estrin, "Fine-grained netowrk time synchronization using reference broadcasts," presented at the ACM OSDI 2002, Boston, MA, Dec. 2001.

[6] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. COM-39, no. 10, pp. 1482–1493, Oct. 1991.

[7] ——, "Improved algorithms for synchronizing computer network clocks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 3, pp. 245–254, Jun. 1995.

[8] ——, Network Time Protocol (Version 3) Specification, Implementation and Analysis. Network Working Group Report. Univ. Delaware, RFC-1305, 1992, p. 113.

[9] O. Gurewitz, I. Cidon, and M. Sidi, "Network time synchronization using clock offset optimization," in *Proc. IEEE ICNP*, Atlanta, GA, Nov. 2003, pp. 212–221.

[10] J. Elson, R. Karp, C. Papadimitriou, and S. Shenker, "Global synchronization in sensornets," in *Proc. 6th Latin American Symposium on Theoretical Informatics (LATIN'04)*, Buenos Aires, 2004, pp. 609–624.

[11] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 564–574.

[12] V. Paxson, "On calibrating measurements of packet transit times," presented at the ACM SIGMETRICS, Madison, WI, 1998.

[13] S. Moon, P. Skelley, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proc. IEEE INFOCOM*, New York, 1999, pp. 227–234.

[14] L. Zhang, Z. Liu, and C. H. Xia, "Clock synchronization algorithms for network measurements," in *Proc. IEEE INFOCOM*, New York, 2002, pp. 160–169.

[15] B. Patt-Shamir, "A theory of clock synchronization," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 1994.

[16] O. Gurewitz and M. Sidi, "Estimating one-way delays from cyclic-path delay measurements," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 1038–1044.

[17] M. Tsuru, T. Takine, and Y. Oie, "Estimation of clock offset from one-way delay measurement on asymmetric path," presented at the Symp. Applications and the Internet (SAINT) Workshops, Nara, Japan, Jan. 2002.

[18] J. Stoer and R. Bulirsch*, Introduction to Numerical Analysis*, 3rd ed. New York: Springer-Verlag, 2002.

[19] C. T. Kelley*, Iterative Methods for Linear and Nonlinear Equations*. Philadelphia, PA: SIAM, 1995.

[20] O. Gurewitz, I. Cidon, and M. Sidi, Network classless time protocol based on clock offset optimization, CCIT, Tech. Rep. 430, Jun. 2003 [Online]. Available: http://www.ee.technion.ac.il/CCIT/info/Publications/Articles/430.pdf

[21] R. T. Rockafellar*, Convex Analysis*. Princeton, NJ: Princeton Univ. Press, 1972.

[22] S. Boyd and L. Vandenberghe, "Convex optimization," Stanford Univ. and UCLA, 2002 [Online]. Available: http://www.stanford.edu/class/ee364 and http://www.ee.ucla.edu/ee236b

[23] N. Minar, "A survey of the NTP network," MIT, Dec. 1999 [Online]. Available: http://www.media.mit.edu/nelson/research/ntp-survey99/

[24] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," *Internetworking: Research and Experience*, vol. 5, no. 4, pp. 163–205, 1994.

[25] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Trans. Netw.*, vol. 7, no. 3, pp. 277–292, Jun. 1999.

[26] A. Corlett, D. I. Pullin, and S. Sargood, "Statistics of one-way Internet packet delays," presented at the 53rd IETF, Minneapolis, MN, Mar. 2002.

[27] E. W. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. IEEE INFOCOM*, San Francisco, CA, 1996, pp. 594–602.

[28] 802.20 Evaluation Criteria—Ver. 05. Draft Permanent Document of IEEE Working Group 802.20. Sep. 2003.

[29] R. Keralapura, C. N. Chuah, G. Iannaccone, and S. Bhattacharrya, "Service availability: A new approach to characterizing network topologies," in *Proc. IEEE IWQoS*, Jun. 2004, pp. 232–241.

[30] M. A. Lombardi, NIST Time and Frequency Services NIST, Special Publication 432, 2002, p. 59.

**Omer Gurewitz** (S'00–M'05) received the B.Sc. degree in physics from Ben Gurion University, Beer Sheva, Israel, in 1991, and the M.Sc. and Ph.D. degrees from the Technion–Israel Institute of Technology, Haifa, in 2000 and 2005, respectively.

He is a Postdoctoral Fellow in the Department of Electrical and Computer Engineering at Rice University, Houston, TX. His research interests include design optimization and performance evaluation of computer networks.

**Israel Cidon** (M'85–SM'90) received the B.Sc. and D.Sc. degrees in electrical engineering from the Technion–Israel Institute of Technology, Haifa, in 1980 and 1984, respectively.

He is a Tark Professor of Electrical Engineering and the Dean of the Electrical Engineering Department at the Technion. His research interests include converged wireline and wireless networks and Network on Chip. During 1985–1994, he was a Research Member and the Manager of the Network Architecture and Algorithms group at IBM T. J. Watson Research Center. In 1994–1995, he founded and managed high-speed networking at Sun Microsystems Labs. He co-founded Micronet Ltd. (1981), Viola Networks (1998) and Actona Technologies (2000–acquired by Cisco in 2004).

Dr. Cidon received IBM Outstanding Innovation Awards for his work on the PARIS project (1989) and topology update algorithms (1993). He was a founding editor for the IEEE/ACM TRANSACTIONS ON NETWORKING and an editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.

**Moshe Sidi** (S'77–M'78–SM'87) received the B.Sc., M.Sc., and D.Sc. degrees from the Technion–Israel Institute of Technology, Haifa, Israel, in 1975, 1979 and 1982, respectively, all in electrical engineering.

In 1982, he joined the faculty of the Electrical Engineering Department at the Technion, where he is currently a Chaired Professor. During the academic year 1983–1984, he was a Post-Doctoral Associate at the Massachusetts Institute of Technology, Cambridge. During 1986–1987, he was a visiting scientist at IBM, Thomas J. Watson Research Center. He coauthored the book *Multiple Access Protocols: Performance and Analysis* (Springer Verlag, 1990).

Dr. Sidi served as the Editor for Communication Networks for the IEEE TRANSACTIONS ON COMMUNICATIONS from 1989 to 1993, as the Associate Editor for Communication Networks and Computer Networks for the IEEE TRANSACTIONS ON INFORMATION THEORY from 1991 to 1994, as a founding Editor in the IEEE/ACM TRANSACTIONS ON NETWORKING from 1993 to 1997, and as an Editor for the *Wireless Journal* from 1993 to 2001. He also served as the General Chair for IEEE INFOCOM 2000.