CCIT Report # 430 June 2003

Network Classless Time Protocol Based on 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, prove its convergence to the optimal clock offsets and show its properties. 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.

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 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 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 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 similar protocol complexity.

The 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 section.

II. The Model, assumptions and background

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.

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 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 start with a model in which only one out of the N nodes is a "reference time node" (generalization for several reference time nodes is given in Section VI); 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.

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.

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, neither we assume any knowledge regarding their 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).

C. 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_i^k). The receiver Λ_j stamps its local time both upon receiving a packet (R_j^k), and upon retransmitting the packet back to the source (T_j^k). The source Λ_i stamps its local time upon receiving the packet back (R_i^k). Each packet k will eventually have four time stamps on it: T_i^k, R_j^k, T_j^k and R_i^k. 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 $x_{i,j}^k$ 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_{ij}^k ($RTT_{ij}^k = x_{i,j}^k + x_{j,i}^k$). 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_{ij}^k 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., $\Delta T_{ij}^k = R_j^k - T_i^k$. Note that the different times are taken according to different clocks which are not necessarily synchronized, hence the computed time ΔT_{ij}^k, 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

[1]Note that it is sufficient to have only two time stamps on each packet, T_i^k and R_j^k, 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_j^k and R_i^k. We suggest to use four time stamps for compliance with the NTP message format.

between the two clock offsets,

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

Note that ΔT_{ij}^k can take negative values.

We will give a special significance to the packet that experiences the minimum delay over each 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 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, $\Delta T_{ij} = \Delta T_{ij}^{P^{ij}}$.

D. Network Time Protocol (NTP) Background

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: $RTT_{ij}^k = (R_j^k - T_i^k) + (R_i^k - T_j^k)$. The node also estimates the clock offset of node Λ_i's clock relative to node Λ_j's clock as: $\frac{1}{2}\left[(R_j^k - T_i^k) - (R_i^k - T_j^k)\right]$. 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.

III. Methodology

A. 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:

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

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, 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 [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:

$$\begin{aligned} F(\vec{\tau}) &= \sum_{\forall e_{i,j} \in \mathcal{E}} (\Delta T_{ij}^{'} - \Delta T_{ji}^{'})^2 \\ &= \sum_{\forall e_{i,j} \in \mathcal{E}} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2 \end{aligned} \tag{3}$$

The goal is to minimize $F(\vec{\tau})$ over $\vec{\tau} \in \mathbb{R}^{N-1}$ 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 $\Delta T_{ij} = \min_k[\Delta T_{ij}^k]$.

B. 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.

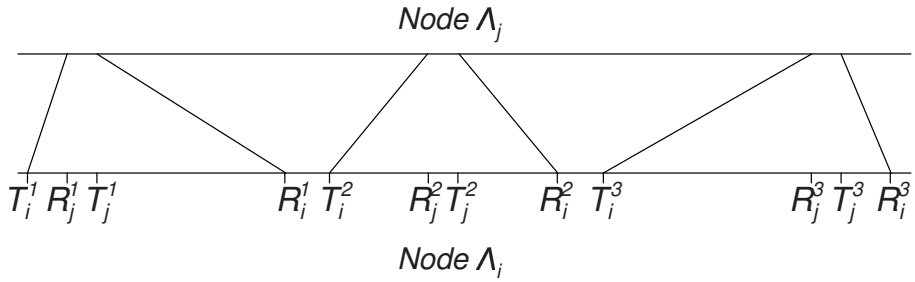

Fig. 1. Exchange of three NTP massages between nodes Λ_i and Λ_j. The minimum ΔT_{ij} is obtained by packet 1, $\min_k \Delta T_{ij} = R_j^1 - T_i^1$. The minimum ΔT_{ji} is obtained by packet 3, $\min_k \Delta T_{ji} = R_i^3 - T_j^3$, while the minimum RTT_{ij} is obtained by packet 2, $\min_k RTT_{ij} = (R_j^2 - T_i^2) + (R_i^2 - T_j^2)$. 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 obtained based on the packet which experienced the minimum round trip delay. $R_j^1 - T_i^1 \leq R_j^2 - T_i^2$, $R_i^3 - T_j^3 \leq R_i^2 - T_j^2$, hence $(R_j^1 - T_i^1) + (R_i^3 - T_j^3) \leq (R_j^2 - T_i^2) + (R_i^2 - T_j^2)$

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 (3). 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 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 Appendix A.

The optimal value of $\vec{\tau}$ which minimizes (3) can now be obtained by partially differentiating (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.

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

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

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

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

$$\vec{\tau} \cdot \mathbf{A} = \vec{\mathbf{\Delta}} \tag{6}$$

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 $i = 1,\ 2,\ \ldots,\ N-1$.

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

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. *Q.E.D.*

V. The Classless Time Protocol (CTP)

In the previous section we introduced the optimal values of the offsets τ_i's that minimize the objective function (3). Obviously, the most straightforward method to solve the optimization problem is to use a centralized protocol. Each node transmits its minimum measurements (ΔT_{ij}) to a centralized entity 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 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. 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 Λ_i, besides node Λ_0, maintains a record in which it holds the entries ΔT_{ij}, ΔT_{ji} and $\Delta_{ij} = \Delta T_{ij} - \Delta T_{ji}$ for each neighbor $\Lambda_j \in G_i$. 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 [9], [10]. 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 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 $\Delta_{ij}^{new} = \Delta_{ij}^{old} - 2\tau_i + 2\tau_j$. A pseudocode of the distributed CTP is given in Appendix B.

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 bigger 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 Λ_i in the h-th iteration, $\Delta_{ij}^{[h]}$ denotes the value of Δ_{ij} after the h-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 appears in Appendix C.

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 (3) i.e., the set of clocks that would have

been obtained by performing the centralized protocol.

Proof: Proposition 3 suggests that the series $S^{[h]} = \sum_{e_{ij}\in\mathcal{E}}(\Delta_{ij}^{[h]})^2$ obtained in each iteration, is a non increasing series. Since the series is bounded from below by zero, it must converge. We will show that it converges to the optimal value which results by solving (6).

Let us assume that the series $S^{[h]}$ converges to Θ, and prove that:

$$\Theta = \min_{\vec{\tau}\in\mathbb{R}^{N-1}} \left\{ \sum_{\forall e_{i,j}\in\mathcal{E}} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2 \right\} \tag{7}$$

Based on Proposition 3 and according to (12) in Appendix C:

$$\begin{aligned} S^{[h]} &= S^{[h-1]} - 4\sum_{e_{ij}\in\mathcal{E}} \left(\tau_i^{[h]} + \tau_j^{[h]}\right)^2 \overset{h\to\infty}{\longrightarrow} \Theta \\ &\Rightarrow \quad \sum_{e_{ij}\in\mathcal{E}} \left(\tau_i^{[h]} + \tau_j^{[h]}\right)^2 \overset{h\to\infty}{\longrightarrow} 0 \\ &\Rightarrow \quad \tau_i^{[h]} + \tau_j^{[h]} \overset{h\to\infty}{\longrightarrow} 0 \qquad \forall e_{ij}\in\mathcal{E} \end{aligned}$$

Since $\tau_0 = 0$ we have:

$$\begin{aligned} \tau_0 = 0 \quad &\Rightarrow \quad \tau_i^{[h]} \overset{h\to\infty}{\longrightarrow} 0 \qquad \forall \Lambda_i \in G_0 \\ &\Rightarrow \quad \tau_j^{[h]} \overset{h\to\infty}{\longrightarrow} 0 \qquad \forall \Lambda_j \in G_{\Lambda_i\in G_0} \\ &\Rightarrow \quad \cdots \quad \Rightarrow \quad \tau_i^{[h]} \overset{h\to\infty}{\longrightarrow} 0 \qquad \forall \Lambda_i \in N \\ &\Rightarrow \quad \tau_i^{[h]} = \frac{1}{2|G_i|}\sum_{j\in G_i} \Delta_{ij}^{[h]} = 0 \qquad \forall \Lambda_i \in \mathcal{N}\setminus\Lambda_0 \\ &\Rightarrow \quad \sum_{j\in G_i} \Delta_{ij}^{[h]} = 0 \qquad \forall \Lambda_i \in \mathcal{N}\setminus\Lambda_0 \end{aligned}$$

According to Corollary 1 the consequence of all nodes adjusting their clocks according to the optimal solution which is unique, is that for each node $\sum_{j\in G_i}\Delta_{ij} = 0$, consequently Θ must be the optimal solution. *Q.E.D.*

VI. CTP Properties

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

We begin by showing the CTP performance in an ideal case.

Property 1: In the ideal case, where on each unidirectional link in the network at least one packet experienced no queueing delay, the CTP ensures that all clocks in the network will be perfectly synchronized with respect to the UTC.

Proof: Since in the ideal case there is at least one packet on each link that has experienced no queuing delay and since the propagation delay on all bidirectional links is symmetrical, there exists a solution in which $\Delta T_{i,j} = \Delta T_{j,i} \quad \forall \Lambda_i, \Lambda_j$ including Λ_0. Hence, there is a solution in which $F(\vec{\tau})$ defined in (3) gets the value 0. Since $F(\vec{\tau})$ is always greater than or equal to zero, this solution must be the optimum. This solution gives the true clock offsets, otherwise some of the $\Delta T_{i,j}$ must be negative, which is not possible. *Q.E.D.*

Next we show the effect of having a number of UTCs.

Property 2: When using the CTP there is no restriction on the number of reference time nodes (UTC), and there can be as many UTCs as one wishes.

Proof: Assume we have an N nodes E links network with m reference time nodes. The objective function is the one suggested in (3), i.e. $F(\vec{\tau}) = \sum_{\forall e_{i,j}\in\mathcal{E}}(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2$. The goal is to minimize $F(\vec{\tau})$

over $\vec{\tau} \in \mathbb{R}^{N-m}$ where $\tau_i = 0 \quad \forall \Lambda_i \in \{set\ of\ m\ reference\ time\ nodes\}$. Let us look at a corresponding $N-m+1$ nodes, E links network. In this network there is only one reference time node. The set of links is similar to the set of links in the first network, where any link connecting a node to any reference time node is replaced by a corresponding link which connects the node to the single reference time node in the corresponding network. The goal now is to optimize $\sum_{\forall e_{i,j} \in \mathcal{E}} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2$ 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 ΔT_{ij}, results in the same optimization point with the same set of clock offsets. Furthermore given a set of measurements where all the reference time nodes are indistinguishable (all called Λ_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 reference time nodes without differentiating between the reference time nodes will yield the right clock offsets which optimize (3). *Q.E.D.*

Finally, 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 Λ_k clock is *influenced* by node Λ_j if the clock offset obtained by node Λ_k after performing the CTP depends on node Λ_j's clock offset and the measurements taken by it, i.e. if the value obtained for τ_k by solving (5), depends on the value obtained for τ_j and the entries $\Delta T_{ji}\ \forall \Lambda_i \in G_j$. We say that node Λ_j *influences* node Λ_k clock, if node Λ_k clock is influenced by node Λ_j. We can now state the following property:

Property 3: Using the CTP node Λ_k clock is influenced by another node Λ_i only if there exists a simple path from node Λ_k to UTC which passes through node Λ_i.

The proof of this property is given in Appendix D.

The importance of Proposition 3 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 3 also provides very good insight to the excellent 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 not less significant makes CTP not worse than hierarchical schemes such as NTP for network topologies which are "tailored" for hierarchical schemes, such as tree topology. For example let us look at the tree topology network suggested in Figure 2.a. The only nodes that influence node Λ_k are the nodes Λ_a and Λ_0 which are the nodes along the path between nodes Λ_k and Λ_0. On the other hand by adding a new link between nodes Λ_k and Λ_b we add a new simple path between nodes Λ_k and Λ_0, hence by using the CTP node Λ_k's clock will be also influenced by node Λ_b and the rest of the nodes along the path.

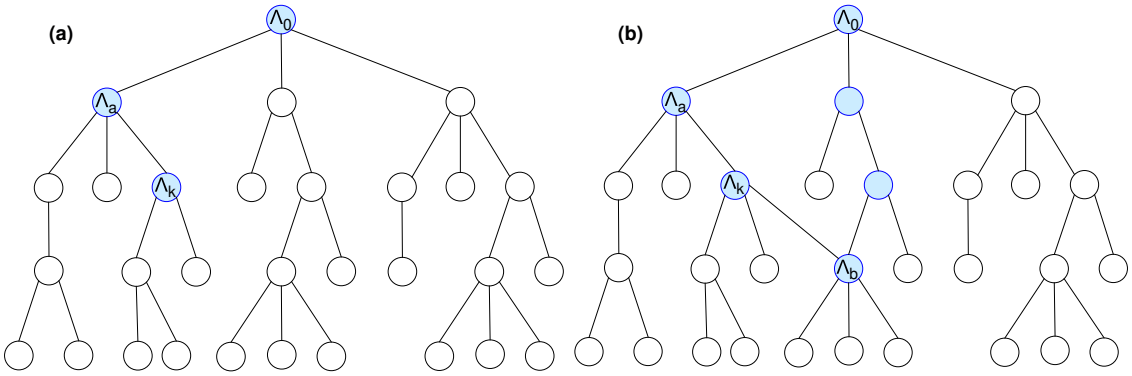

Fig. 2. The nodes that influence node Λ_k's clock in tree topology and non tree topology networks.

VII. Numerical Results

A. 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 con-

struction 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 initial clocks' offsets 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 packets are transmitted as suggested by NTP and ΔT_{ij} are measured based on these packets.

B. 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 III-B. 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.

We start by investigating the measurement filter. As explained in Section III-B, 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 3 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 3 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 III-B 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 5 unit is 0.41 for the "one way method", while it is only 0.26 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.

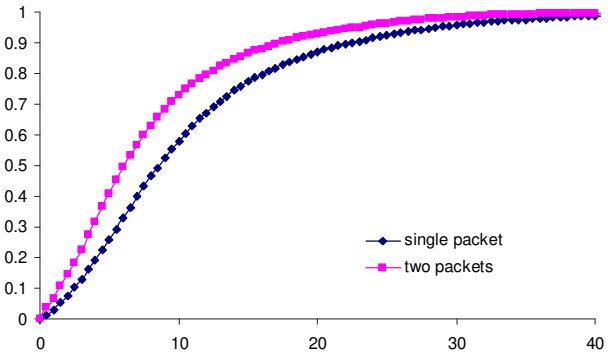

Fig. 3. Distribution of delay errors

Next we examine the clock adjustments ($\vec{\tau}$) that minimize the objective function suggested in Section III-

A. The clock adjustments are determined by applying the centralized protocol suggested in V. 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 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 III-B. This modification not only changes the quantity of the offset as demonstrated in Figure 3, 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 three networks and adjusted the clocks accordingly. Figures 4 and 5 show the results on 490 and 1092 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 clock offset depicted by the x value. Figure 6 depicts the results in a 1292 node network. The y axis presents the probability density function (fraction of nodes out of the 1292 nodes) with the clock offsets described by the x axis.

Figures 4, 5 and 6 clearly demonstrate the significant improvement in terms of clock accuracy of the CTP over all hierarchical schemes. For example, it can be seen in the graphs that about one third of all nodes in the 490 node network and about 38% of the nodes in the 1092 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 8%, 10% and 11% for the 490 node network, and 9%, 11% and 13% for the 1092 node network get the same result, respectively. In Figure 6 it can be seen that after performing the CTP 95% of the nodes will have their clocks less than 5 time units from the UTC and all the nodes will have their clocks less than 10 time units from the UTC. Looking at the three hierarchical schemes it can be seen that between -5 to 5 time units from the UTC lie only 38%, 40% and 50% of the nodes for the hierarchical-1, -2 and -3 respectively. The error bounds for the hierarchical scheme are $[-29.63, 50.55]$, $[-23.28, 50.79]$ and $[-20.27, 33.06]$ respectively.

In order to demonstrate the clock offset dispersion around the UTC clock, we draw graphs 7 and 8. 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 one of the schemes. Figure 7 depicts the clock offset dispersion on a 263 node network while Figure 8 relates to a 1014 node network. In both graphs it can be seen as expected that the 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, the CTP keeps the region about the same regardless of the distance from the UTC while in the hierarchical scheme the farther you get from the UTC (higher node ID) the wider the region is.

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 (6). Figure 9 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.

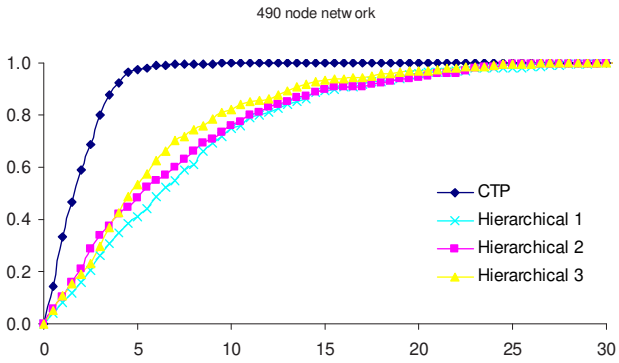

Fig. 4. The fraction of nodes with clock offset with respect to the reference time node that is not greater than t, on a 490 node network.

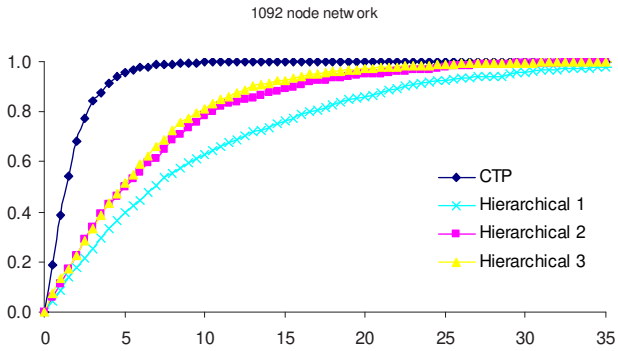

Fig. 5. The fraction of nodes with clock offset with respect to the reference time node that is not greater than t, on a 1092 node network.

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 not symmetric.

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

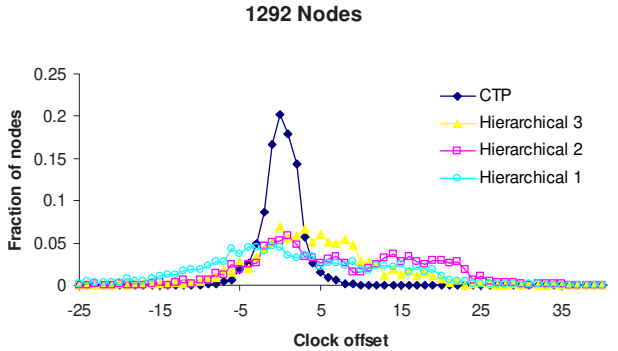

Fig. 6. The fraction of nodes with clock offset with respect to the reference time node that is between $x-1$ and x (PDF), on a 1292 node network.

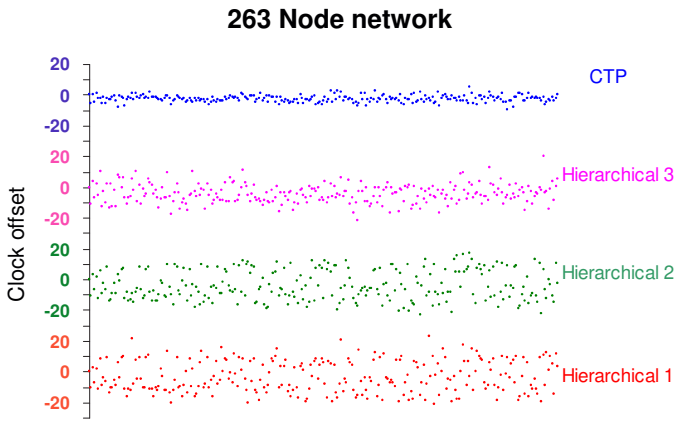

Fig. 7. The clock offset dispersion on a 263 node network

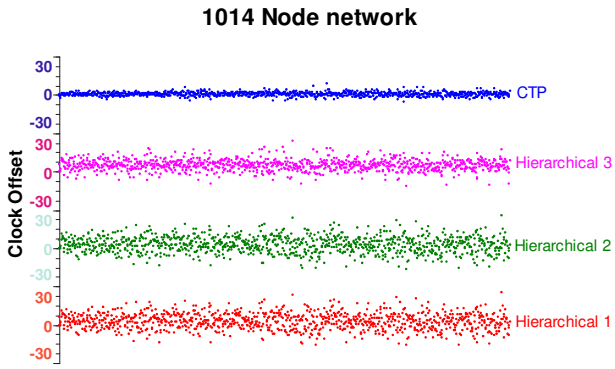

Fig. 8. The clock offset dispersion on a 1014 node network

used, we chose a protocol which is easy to implement and requires only minor modifications to the format and number of messages used by NTP. Numerical results illustrate that our approach works well in various randomly chosen networks, and substantially outperforms hierarchical schemes such as NTP.

Acknowledgments

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

Appendix A

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.

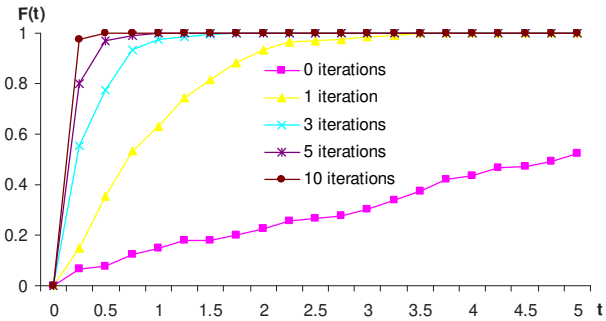

Fig. 9. The fraction of nodes in a 169 node network with clock offset with respect to the set of optimal clock offsets (optimal solution) not greater than t, during the implementation of the suggested distributed protocol.

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:

$$F(\vec{\tau}) = \vec{\tau}\mathbf{P}\vec{\tau}^{\mathbf{T}} + \vec{\mathbf{q}}\vec{\tau}^{\mathbf{T}} + \mathbf{r} \tag{8}$$

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 $\vec{\tau}\mathbf{P}\vec{\tau}^{\mathbf{T}} > \mathbf{0} \quad \forall \vec{\tau} \in \mathbb{R}^{\mathbf{N-1}}$ except $\vec{\tau} = \vec{0}$.

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

Hence $\vec{\tau}\mathbf{P}\vec{\tau}^{\mathbf{T}} \geq \mathbf{0} \quad \forall \vec{\tau} \in \mathbb{R}^{\mathbf{N-1}}$. In order for $\vec{\tau}\mathbf{P}\vec{\tau}^{\mathbf{T}}$ 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 $\vec{\tau}\mathbf{P}\vec{\tau}^{\mathbf{T}} = \mathbf{0}$ 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 $F(\vec{\tau}) = \vec{\tau}\mathbf{P}\vec{\tau}^{\mathbf{T}} + \vec{\mathbf{q}}\vec{\tau}^{\mathbf{T}} + \mathbf{r}$ 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.

Appendix B

In this appendix we describe simple, conceptual version of the distributed CTP using pseudo-code formulation.

We consider a connected network $(\mathcal{N}, \mathcal{E})$ and assume that there is a Data-Link control protocol associated with each link, that ensures data reliability, where reliability implies that all packets produced by a source node

are delivered to the destination node in finite time, and in the same order as they were transmitted (FIFO assumption). We also separate the algorithm into two phases. The first phase is responsible for conducting the measurements, and pass them through the measurement filter explained in Section III-B. The second phase takes care of the clock adjustment based on the measurements, and is in charge of the convergence to the optimal point discussed in Section IV. In the protocol presented here the two phases are separated. We restrict our algorithm to a single time execution, which implements the ideas presented in the paper. The second phase is operated in a loop with no termination conditions.

Protocol:
Messages:
$MSG1(T_i)$ - The message sent by node Λ_i which carries the time stamp of its transmission time
$MSG2(T_i, R_j, T_j)$ - The message sent by node Λ_i which carries the time stamps of the transmission time, the receiving time by the node on the other side of the transmitted link, and the transmission time by the same node on the same link on the direction, T_i, R_j, T_j respectively.
$MSG3(\Delta T^i_{j,i})$ - The message sent by node Λ_i to notify its neighbor Λ_j its updated $\Delta T_{j,i}$).
$MSG4(\tau_i)$ - The message sent by node Λ_i to notify its neighbors regarding the time it adjusted its clock, in order for them to modify their ΔT_{ij}, and ΔT_{ji}.
Variables:
G_i - set of Λ_i neighbor nodes.
$|G_i|$ - degree of Λ_i.
ΔT_{ij} - the minimum time difference obtained over all packets, between the reception time of a packet by node Λ_j according to its own clock, and the transmission time of the packet according to node Λ_i clock.
ΔT_{ji} - the minimum time difference obtained over all packets, between the reception time of a packet by node Λ_i and the transmission time of the packet by node Λ_i clock, each according to the local time.
τ_i - the clock adjustment conducted by node Λ_i.
$clock_i$ - the clock at node Λ_i.
$N_i(j)$ - level of last message received from neighbor Λ_j .
L - the number of messages exchanged by a node before performing the clock adjustment procedure.

INITIALIZATION()
1 $\Delta T_{ij} \leftarrow \infty$
2 $\Delta T_{ji} \leftarrow \infty$
3 $\tau_i \leftarrow 0$
4 $l \leftarrow 0$

FILTER ALGORITHM()
1 **for** $Start1$
2 **do** $N_i(j) \leftarrow 0 \quad \forall \Lambda_j \in G_i$
3 **if** $l < L$
4 **then** $T_i \leftarrow Clock$
5 Send $MSG1(T_i)$ to all $\Lambda_j \in G_i$
6 **if** $l = L$
7 **then** Send $MSG3(\Delta T^i_{j,i})$ to all $\Lambda_j \in G_i$
8 $l \leftarrow (l + 1)$
9 **for** $MSG1(T_j)$ from Λ_j
10 **do** $R_i \leftarrow Clock$
11 **if** $l = 0$
12 **then** $Start1$
13 **if** $\Delta T_{ji} > R_i - T_j$
14 **then** $\Delta T_{ji} \leftarrow R_i - T_j$
15 $T_i \leftarrow Clock$
16 Send $MSG2(T_j, R_i, T_i)$ to Λ_j

for $MSG2(T_i, R_j, T_j)$ from Λ_j
  do $R_i \leftarrow Clock$
    $N_i(j) \leftarrow 1$
    if $\Delta T_{ij} > R_j - T_i$
      then $\Delta T_{ij} \leftarrow R_j - T_i$
    if $\Delta T_{ji} > R_i - T_j$
      then $\Delta T_{ji} \leftarrow R_i - T_j$
    if $N_i(k) = 1 \ \ \forall \Lambda_k \in G_i$
      then $Start1$
for $MSG3(\Delta T^j_{ji})$ from Λ_j
  do $N_i(j) \leftarrow 1$
    if $\Delta T_{ij} > \Delta T^j_{ji}$
      then $\Delta T_{ij} \leftarrow \Delta T^j_{ji}$
    if $N_i(k) = 1 \ \ \forall \Lambda_k \in G_i$
      then $Start2$

Clock Adjustment Algorithm()

for $Start2$
  do $N_i(j) \leftarrow 0 \quad \forall \Lambda_j \in G_i$
    $\tau_i \leftarrow \frac{1}{2|G_i|} \sum_{l \in G_i} (\Delta T_{il} - \Delta T_{li})$
    $Clock \leftarrow Clock + \tau_i$
    for $\forall \Lambda_j \in G_i$
      do $\Delta T_{ij} \leftarrow \Delta T_{ij} - \tau_i$
        $\Delta T_{ji} \leftarrow \Delta T_{ji} + \tau_i$
     Send $MSG4(\tau_i)$ to all $\Lambda_j \in G_i$
for $MSG4(\tau_j)$ from Λ_j
  do $N_i(j) \leftarrow 1$
    $\Delta T_{ij} \leftarrow \Delta T_{ij} + \tau_j$
    $\Delta T_{ji} \leftarrow \Delta T_{ji} - \tau_j$
    if $N_i(k) = 1 \ \ \forall \Lambda_k \in G_i$
      then $Start2$
        $l \leftarrow (l + 1)$

Appendix C

Proof of Proposition 3: Since nodes belong to set Ψ adjust their clock by $\frac{1}{2|G_i|} \sum_{j \in G_i} \Delta_{ij}$ and nodes which are not in the set do not adjust their clock at all let us denote:

$$\tau_i^{[h]} = \begin{cases} \frac{1}{2|G_i|} \sum_{j \in G_i} \Delta_{ij}^{[h-1]} & \text{if } \Lambda_i \in \Psi \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$\begin{aligned} S^{[h]} &= \sum_{e_{ij} \in \mathcal{E}} \left(\Delta_{ij}^{[h]}\right)^2 = \sum_{e_{ij} \in \mathcal{E}} \left(\Delta_{ij}^{[h-1]} - 2\tau_i^{[h]} + 2\tau_j^{[h]}\right)^2 \\ &= \sum_{e_{ij} \in \mathcal{E}} \Big((\Delta_{ij}^{[h-1]})^2 - 4\Delta_{ij}^{[h-1]}\tau_i^{[h]} + 4\Delta_{ij}^{[h-1]}\tau_j^{[h]} + \\ &\qquad +4(\tau_i^{[h]})^2 + 4(\tau_j^{[h]})^2 - 8\tau_i^{[h]}\tau_j^{[h]}\Big) \\ &= \sum_{e_{ij} \in \mathcal{E}} \left(\Delta_{ij}^{[h-1]}\right)^2 - 4 \sum_{e_{ij} \in \mathcal{E}} \left(\tau_i^{[h]} + \tau_j^{[h]}\right)^2 + \end{aligned}$$

$$+\sum_{e_{ij}\in\mathcal{E}}\left(8(\tau_i^{[h]})^2+8(\tau_j^{[h]})^2-4\Delta_{ij}^{[h-1]}\tau_i^{[h]}+4\Delta_{ij}^{[h-1]}\tau_j^{[h]}\right) \tag{10}$$

Let us concentrate on the third part:

$$\begin{aligned}
&\sum_{e_{ij}\in\mathcal{E}}\left(8(\tau_i^{[h]})^2+8(\tau_j^{[h]})^2-4\Delta_{ij}^{[h-1]}\tau_i^{[h]}+4\Delta_{ij}^{[h-1]}\tau_j^{[h]}\right)\\
&\quad=4\sum_{e_{ij}\in\mathcal{E}}\left(2(\tau_i^{[h]})^2+2(\tau_j^{[h]})^2-\Delta_{ij}^{[h-1]}\tau_i^{[h]}-\Delta_{ji}^{[h-1]}\tau_j^{[h]}\right)\\
&\quad=4\sum_{\Lambda_i\in\mathcal{N}}\sum_{\Lambda_j\in G_i}\left(2(\tau_i^{[h]})^2-\Delta_{ij}^{[h-1]}\tau_i\right)\\
&\quad=4\left\{\sum_{\Lambda_i\in\mathcal{N}}\left(2(\tau_i^{[h]})^2|G_i|\right)-\sum_{\Lambda_i\in\mathcal{N}}\tau_i^{[h]}\sum_{\Lambda_j\in G_i}\Delta_{ij}^{[h-1]}\right\}
\end{aligned} \tag{11}$$

When passing from the second to the third row we sum over the nodes instead of summing over the links. Based on (9) for $\tau_i^{[h]}\neq 0$ we have $\sum_{j\in G_i}\Delta_{ij}^{[h-1]}=2|G_i|\tau_i^{[h]}$, we can write 11:

$$\begin{aligned}
&4\left\{\sum_{\Lambda_i\in\mathcal{N}}\left(2(\tau_i^{[h]})^2|G_i|\right)-\sum_{\Lambda_i\in\mathcal{N}}\tau_i^{[h]}\sum_{\Lambda_j\in G_i}\Delta_{ij}^{[h-1]}\right\}\\
&\quad=4\left\{\sum_{\Lambda_i\in\Psi}\left(2(\tau_i^{[h]})^2|G_i|\right)-\sum_{\Lambda_i\in\Psi}\tau_i^{[h]}\cdot 2|G_i|\tau_i^{[h]}\right\}=0
\end{aligned}$$

Putting it back to (10) we have:

$$\begin{aligned}
S^{[h]} &= \sum_{e_{ij}\in\mathcal{E}}\left(\Delta_{ij}^{[h-1]}\right)^2-4\sum_{e_{ij}\in\mathcal{E}}\left(\tau_i^{[h]}+\tau_j^{[h]}\right)^2\\
&= S^{[h-1]}-4\sum_{e_{ij}\in\mathcal{E}}\left(\tau_i^{[h]}+\tau_j^{[h]}\right)^2
\end{aligned} \tag{12}$$

This means that unless all $\tau_i^{[h]}$'s are zero the sum after the next iteration is smaller than prior to the iteration.

Appendix D

Proof of Property 3: Let node Λ_k be an arbitrary node in $\mathcal{N}$. We divide the nodes in the network with respect to node Λ_k into two groups. The first group denoted by Θ_k includes all nodes that participate as intermediate nodes in at least one simple (cycle free) path between Λ_k and Λ_0. The second group denoted by Ψ_k includes the rest of the nodes ($\Psi_k\equiv\mathcal{N}\setminus\Theta_k$). Figure 10 illustrates the two groups. Throughout the proof we will use the notation $\Lambda_x\rightsquigarrow\Lambda_y$ to denote a simple path from node Λ_x to Λ_y.

The proof is a direct consequence of the five Lemmas below.

Lemma 1: Any simple path from Λ_k to Λ_0 passes only through nodes in Θ_k.

Proof: Immediate consequence of Θ_k definition. *Q.E.D.*

Lemma 2: Θ_k is not an empty set, specifically nodes Λ_0 and Λ_k are in Θ_k.

Proof: The network is connected, hence there is at least one simple path between any two nodes, specifically between Λ_k and Λ_0. This path passes through both nodes which makes them belong to Θ_k. *Q.E.D.*

Lemma 3: Any two nodes in Θ_k can be arranged such that there is a simple path from Λ_k to one of them, and another simple path from the other to Λ_0, where the two paths do not pass through any common node.

Proof: Let us look at the two nodes $\Lambda_a,\Lambda_b\in\Theta_k$. Since both nodes are in Θ_k each one of them must participate as an intermediate node in at least one simple path between nodes Λ_k and Λ_0. Let us denote the

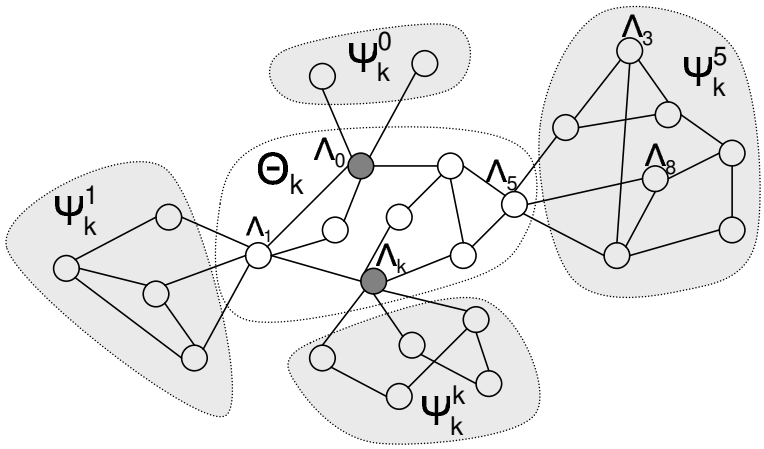

Fig. 10. Network partition into groups with respect to Λ_k.

simple path $\Lambda_k \rightsquigarrow \Lambda_a \rightsquigarrow \Lambda_0$ by p_a, and the simple path $\Lambda_k \rightsquigarrow \Lambda_b \rightsquigarrow \Lambda_0$ by p_b. When referring to only part of a path we will denote which path the segment was extracted from, e.g. $\Lambda_x \overset{p_a}{\rightsquigarrow} \Lambda_y$ will refer to the path between nodes Λ_x and Λ_y which is a fragment of the path p_a.

Let Λ_x be the last node which is mutual to both paths $\Lambda_k \overset{p_a}{\rightsquigarrow} \Lambda_0$ and $\Lambda_k \overset{p_b}{\rightsquigarrow} \Lambda_b$, i.e. Λ_x is an intermediate node on both paths, p_a and $\Lambda_k \overset{p_b}{\rightsquigarrow} \Lambda_b$ and the path $\Lambda_x \overset{p_b}{\rightsquigarrow} \Lambda_b$ does not pass through any other node which is in p_a. Note that Λ_x can be Λ_k, Λ_a or Λ_b in the cases where there are no mutual nodes in the two paths p_a and $\Lambda_k \overset{p_b}{\rightsquigarrow} \Lambda_b$, if Λ_a is an intermediate node in $\Lambda_k \overset{p_b}{\rightsquigarrow} \Lambda_b$ and there are no mutual nodes in the paths $\Lambda_a \overset{p_b}{\rightsquigarrow} \Lambda_b$ and $\Lambda_a \overset{p_a}{\rightsquigarrow} \Lambda_0$, or Λ_b is an intermediate node in p_a, respectively.

If Λ_x is reached before Λ_a on the path $\Lambda_k \overset{p_a}{\rightsquigarrow} \Lambda_0$, then the two paths $\Lambda_k \overset{p_a}{\rightsquigarrow} \Lambda_x \overset{p_b}{\rightsquigarrow} \Lambda_b$ and $\Lambda_a \overset{p_a}{\rightsquigarrow} \Lambda_0$ comply with the Lemma (Figure 11(a)). Otherwise, if Λ_x is reached after Λ_a on the path $\Lambda_k \overset{p_a}{\rightsquigarrow} \Lambda_0$ the two simple paths that satisfy the Lemma are $\Lambda_k \overset{p_a}{\rightsquigarrow} \Lambda_a$ and $\Lambda_b \overset{p_b}{\rightsquigarrow} \Lambda_x \overset{p_a}{\rightsquigarrow} \Lambda_0$ (Figure 11(b)). Note that for the second case we use the bidirectional nature of the links to reverse the path $\Lambda_x \overset{p_b}{\rightsquigarrow} \Lambda_b$. *Q.E.D.*

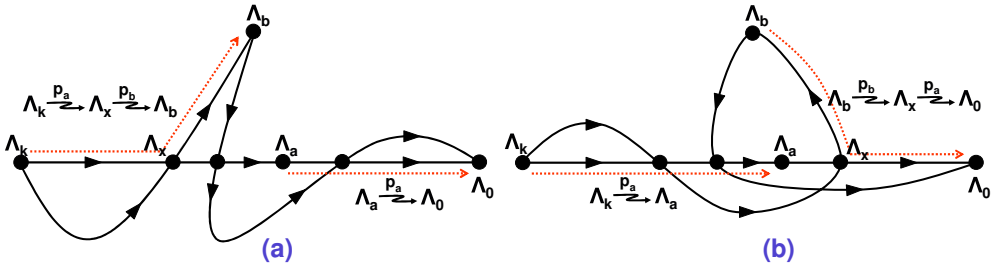

Fig. 11. Illustration of the two scenarios in the proof to Lemma 3. The two disjoint simple paths that can be extracted from the two paths $\Lambda_k \rightsquigarrow \Lambda_a \rightsquigarrow \Lambda_0$ and $\Lambda_k \rightsquigarrow \Lambda_b \rightsquigarrow \Lambda_0$.

Lemma 4: Each node in Ψ_k has a corresponding node in Θ_k, which will be called its root node with respect to Λ_k, such that all paths from it to node Λ_0 pass through the root node without passing any other node in Θ_k before reaching the root node. In Figure 10 for example node Λ_5 is the root node of nodes Λ_3, Λ_8 and all the rest of the nodes in the area which is marked as Ψ_k^5.

Proof: We will prove the Lemma by contradiction. We will assume that there is a node in group Ψ_k that does not have a root node and we will show that this contradicts the definition of Ψ_k.

Let us assume that node $\Lambda_\xi \in \Psi_k$ does not have a root node. Since the network is connected there must be at least one simple path from Λ_ξ to Λ_0. This path must traverse at least one node in group Θ_k ($\Lambda_0 \in \Theta_k$ according to Lemma 2), hence if Λ_ξ does not have a root node there must be at least two simple paths from node Λ_ξ to Λ_0 which differ in the first node along the path which belongs to group Θ_k. Let us denote these two nodes by $\Lambda_a, \Lambda_b \in \Theta_k$ and the two paths $\Lambda_\xi \rightsquigarrow \Lambda_a$ and $\Lambda_\xi \rightsquigarrow \Lambda_b$ by p_a and p_b, respectively. Let Λ_w be the last node in the path $\Lambda_\xi \rightsquigarrow \Lambda_a$ which is also in $\Lambda_\xi \rightsquigarrow \Lambda_b$ (Figure 12). Note that node Λ_w is not in Θ_k

since Λ_a and Λ_b are the first two nodes along the two paths p_a and p_b which are in group Θ_k. The two paths $\Lambda_w \overset{p_a}{\rightsquigarrow} \Lambda_a$ and $\Lambda_w \overset{p_b}{\rightsquigarrow} \Lambda_b$ do not share any mutual node beside node Λ_w itself (Figure 12).

According to Lemma 3 since Λ_a and Λ_b are in Θ_k, we can find two simple paths such that one is from Λ_k to one of the nodes and the other is from the second node to Λ_0 that do not share any mutual node. Let us assume without loss of generality that node Λ_a is the first node, and Λ_b is the second node, hence there are two simple paths $\Lambda_k \rightsquigarrow \Lambda_a$ and $\Lambda_b \rightsquigarrow \Lambda_0$ which do not share any common node.

Now let us look at the path $\Lambda_k \rightsquigarrow \Lambda_a \overset{p_a}{\rightsquigarrow} \Lambda_w \overset{p_b}{\rightsquigarrow} \Lambda_b \rightsquigarrow \Lambda_0$. This path is a simple path from node Λ_k to Λ_0 which passes through node Λ_w, hence node Λ_w must be in Θ_k by definition, but this is a contradiction to the assumption that Λ_a and Λ_b are the first two nodes in paths p_a and p_b which belong to Θ_k, which means that any node that is not in Θ_k (participates as an intermediate node in at least one simple path between nodes Λ_k and Λ_0) must have a root node. *Q.E.D.*

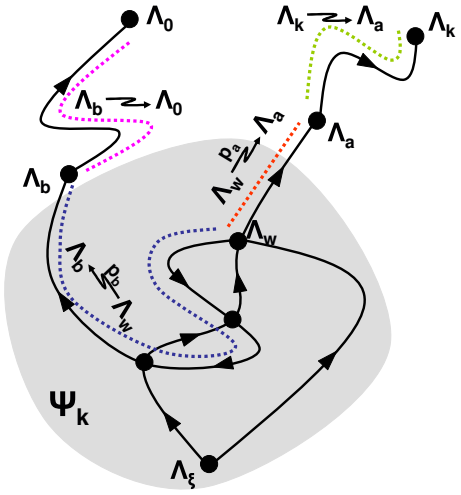

Fig. 12. Illustration of the proof of Lemma 4. If $\Lambda_\xi \in \Psi_k$ does not have a root node there must be a simple path from node Λ_k to Λ_0 traversing node $\Lambda_w \in \Psi_k$.

As a consequence of Lemma 4 we can partition group Ψ_k into smaller groups in which each group comprises all nodes that share a common root node in group Θ_k. Let us denote by Ψ_k^i the subgroup which relates to root node $\Lambda_i \in \Theta_k$ (Figure 10). Let us also denote by Φ_k^i the set of all links which connect the nodes in group Ψ_k^i, including the links connecting the nodes in Ψ_k^i to their root node Λ_i; and by Φ_k (without the superscript) the set of links connecting the nodes in Θ_k among themselves.

Lemma 5: The union of Θ_k and all the sets Ψ_k^i, $\Lambda_i \in \Theta_k$ span all the nodes in the network, i.e. $\bigcup_{\Lambda_i \in \Theta_k} \Psi_k^i \bigcup \Theta_k \equiv \mathcal{N}$ and the intersection of any two such groups is an empty set, i.e. a node belongs to exactly one group. The union of Φ_k and all Φ_k^i, $\Lambda_i \in \Theta_k$ span the entire links in the network, i.e. $\bigcup_{\Lambda_i \in \Theta_k} \Phi_k^i \bigcup \Phi_k \equiv \mathcal{E}$. The intersection of any two such groups is an empty set.

Proof: The network is connected hence all nodes have at least one simple path to Λ_0. Therefore each node either belongs to Θ_k or to one of the Ψ_k^i. According to Lemma 4 each node not in Θ_k has a single root node, hence each node belongs to exactly one group. The second part of the Lemma is a direct consequence of the first part. Each link in the network either connects two nodes that belong to the same group (Θ_k or one of the Ψ_k^i) or connects a node in Ψ_k^i to its root node in Θ_k, which makes it belong to exactly one group. *Q.E.D.*

Now we turn to prove Property 3. We will show that the clock offset obtained by node Λ_k is the same whether all nodes were running the CTP or only the nodes in group Θ_k independently from the rest, i.e. node Λ_k's clock is not influenced by nodes which do not participate in a simple path between nodes Λ_k and Λ_0.

We will start from the objective function $F(\vec{\tau})$ suggested in (3) over all links. We will partition the sum to the smaller groups Φ_k and all Φ_k^i, $\Lambda_i \in \Theta_k$ which according to Lemma 5 span all $\mathcal{E}$ (note that by attaching subgroup Φ_k^i to each node $\Lambda_i \in \Theta_k$ some of the subgroups might be empty).

$$\min_{\vec{\tau}} \left\{ \sum_{\forall e_{i,j} \in \mathcal{E}} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j\right)^2 \right\} = \tag{13.a}$$

$$\min_{\vec{\tau}} \left\{ \sum_{\forall e_{i,j} \in \Phi_k} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j\right)^2 + \right. \tag{13.b}$$

$$\left. \sum_{\forall \Lambda_l \in \Theta_k} \left(\sum_{\forall e_{i,j} \in \Phi_k^l} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j'\right)^2 \right) \right\}$$

Let us separate each sum over all links in Φ_k^l into two partial sums. The first is over all links which are connecting nodes in Ψ_k^l to the root node Λ_l, $S_1^l = \{e_{l,j} | j \in \Psi_k^l,\ j \in G_l\}$, and the second sum is over the rest of the links which connect nodes in Ψ_k^l among themselves, $S_2^l = \Phi_k^l \setminus S_1^l$. Let us also change variables and let:

$$\tau_i' = \begin{cases} \tau_i & \text{if } \Lambda_i \in \Theta_k \\ \tau_i - \tau_l & \text{if } \Lambda_i \in \Psi_k^l \end{cases}$$

(13.b) will take the form:

$$\min_{\vec{\tau}} \left\{ \sum_{\forall e_{i,j} \in \Phi_k} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j\right)^2 + \right. \tag{13.c}$$

$$\left. + \sum_{\forall \Lambda_l \in \Theta_k} \left(\sum_{\forall e_{l,j} \in S_1^l} (\Delta T_{l,j} - \Delta T_{j,l} - 2\tau_l + 2\tau_j)^2 + \right. \right.$$

$$\left. \left. \sum_{\forall e_{i,j} \in S_2^l} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i + 2\tau_j)^2 \right) \right\} =$$

$$\min_{\vec{\tau'}} \left\{ \sum_{\forall e_{i,j} \in \Phi_k} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j'\right)^2 + \right. \tag{13.d}$$

$$\left. + \sum_{\forall \Lambda_l \in \Theta_k} \left(\sum_{\forall e_{l,j} \in S_1^l} (\Delta T_{l,j} - \Delta T_{j,l} - 2\tau_l + 2\tau_j' + 2\tau_l)^2 \right. \right.$$

$$\left. \left. + \sum_{\forall e_{i,j} \in S_2^l} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' - 2\tau_l + 2\tau_j' + 2\tau_l)^2 \right) \right\} =$$

$$\min_{\vec{\tau'}} \left\{ \sum_{\forall e_{i,j} \in \Phi_k} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j'\right)^2 + \right. \tag{13.e}$$

$$\left. + \sum_{\forall \Lambda_l \in \Theta_k} \left(\sum_{\forall e_{l,j} \in S_1^l} (\Delta T_{l,j} - \Delta T_{j,l} + 2\tau_j')^2 \right. \right.$$

$$\left. \left. + \sum_{\forall e_{i,j} \in S_2^l} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j')^2 \right) \right\} =$$

As can be seen from (13.e), the first sum $\sum_{\forall e_{i,j} \in \Phi_k} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j'\right)^2$ depends only on τ_i''s which belong to nodes in Θ_k, and each component of the second sum,
$\left(\sum_{\forall e_{l,j} \in S_1^l} (\Delta T_{l,j} - \Delta T_{j,l} + 2\tau_j')^2 + \sum_{\forall e_{i,j} \in S_2^l} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j')^2 \right)$, depends only on τ_i''s which belong to nodes in Ψ_k^l. According to Lemma 5 the groups Θ_k and Ψ_k^l $\forall \Lambda_l \in \Theta_k$ are disjoint (have no common nodes), hence each term depends on a different set of τ_i's and the minimum of the sum equals the sum of the minimums, i.e. (13.e) takes the form:

$$\min_{\{\tau_h' | \Lambda_h \in \Theta_k\}} \sum_{\forall e_{i,j} \in \Phi_k} \left(\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j'\right)^2 + \tag{13.f}$$

$$+ \sum_{\forall \Lambda_l \in \Theta_k} \left\{ \min_{\{\tau_h' | \Lambda_h \in \Psi_k^l\}} \left[\sum_{\forall e_{l,j} \in S_1^l} (\Delta T_{l,j} - \Delta T_{j,l} + 2\tau_j')^2 \right. \right.$$

$$\left. \left. + \sum_{\forall e_{i,j} \in S_2^l} (\Delta T_{ij} - \Delta T_{ji} - 2\tau_i' + 2\tau_j')^2 \right] \right\}$$

(13.f) completes the proof of the property since according to (13.f) the minimum over group Φ_k could be computed separately where the clock movements $\tau'_i \quad \Lambda_i \in \Theta_k$ are the exact clock movements, $\tau'_i = \tau_i$. Note that the rest of the clock adjustments can also be computed separately based on the minimum obtained for each group Φ^l_k,
$\min\left(\sum_{\forall e_{l,j} \in S^l_1} (\Delta T_{l,j} - \Delta T_{j,l} + 2\tau'_j)^2 + \sum_{\forall e_{i,j} \in S^l_2} (\Delta T_{ij} - \Delta T_{ji} - 2\tau'_i + 2\tau'_j)^2\right)$ and then based on the computed $\tau_i \quad \Lambda_i \in \Theta_k$ we can compute $\tau_j \ \Lambda_j \in \Psi^l_k$ as $\tau_j = \tau'_j - \tau_l$ *Q.E.D.*

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