Electrical Engineering
Electronics
Computers
Communications

Network Time Synchronization using Decentralized Kalman Filtering

Maxime Cohen

Supervised by
Prof. Nahum Shimkin

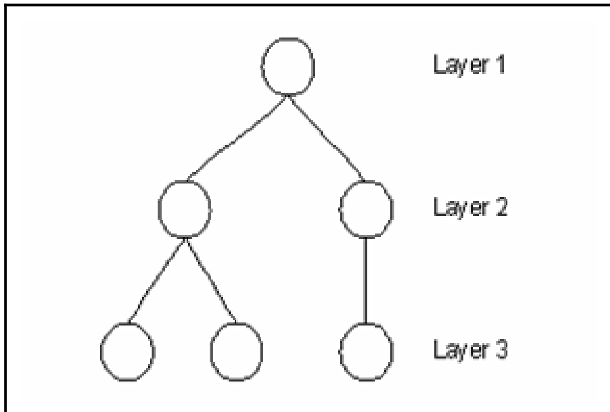
October 11, 2009

Outline

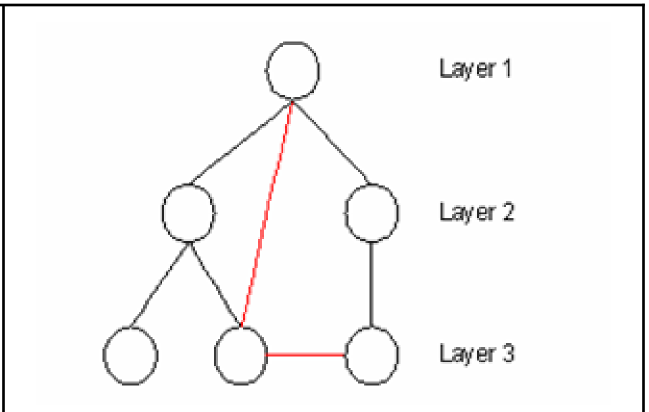
- **Introduction and Motivation**
- Related Work
- Problem Formulation
- Scientific Background
  - Least-Squares Estimation
  - Kalman Filtering
- Algorithms
  - LS algorithm and KF Framework
  - Optimal Centralized Algorithm
  - Decentralized Algorithm
- Convergence Analysis
- Extensions
- Clock Skew Estimation
- Numerical Results
- Conclusions and Future Work

Introduction and Motivation

- Accurate clock synchronization is important in many computer networks applications (e.g., sleep scheduling in the case of low duty cycle and tracking in wireless sensor networks).
- The accuracy of clock synchronization was improved by exploiting global network-wide constraints (loops), taking account of a-priori knowledge and then considering recursive algorithms for multiple measurement sets.



Tree Network



General Network

Introduction and Motivation (Cont.)

- The **Kalman Filter framework** allows exploiting some a-priori knowledge and providing different weights to the measurements according to their accuracy.

- **Decentralized** estimation: requiring only local communication with one-hop neighbors.

- Equivalence with the **sensor localization problem**. Indeed, our algorithms can solve any problem in which we want to estimate some quantities given relative measurements.

Outline

- Introduction and Motivation
- **Related Work**
- Problem Formulation
- Scientific Background
    - Least-Squares Estimation
    - Kalman Filtering
- Algorithms
    - LS algorithm and KF Framework
    - Optimal Centralized Algorithm
    - Decentralized Algorithm
- Convergence Analysis
- Extensions
- Clock Skew Estimation
- Numerical Results
- Conclusions and Future Work

Related Work

- NTP (*Network Time Protocol*), D. L. Mills, 1991, 1992 and 1995 (version 3) – The most widely accepted standard for synchronizing clocks over the internet, hierarchical procedure.
- O. Gurewitz, I. Cidon, M. Sidi, *Network time synchronization using clock offset optimization,* 2003.
- R. Solis, V. Borkar, P. R. Kumar, A *new distributed time synchronization protocol for multihop wireless networks,* 2005.
- Other methods were proposed, like RBS (*Reference Broadcast Synchronization*) and an interesting extension of D. Estrin et. Al, 2003, *Optimal and Global time synchronization in sensornets.*

Related Work (Cont.)

- **Decentralized Kalman Filter (DKF)** was extensively treated in the literature and many approaches (both optimal and heuristics) were proposed.
  First, the algorithms were applied to **fully connected** networks: e.g., [Hasemipour, Roy and Laub, 1988] and [Rao and Durant-Whyte, 1991].

- **Consensus algorithms:** [Xiao and Boyd, 2004], [Olfati-Saber and Shamma, 2005] and [R. Carli et al., 2008]. In fact, numerous of these methods are related to data fusion in networks and particularly distributed Kalman Filtering using weighted averaging [Alriksson and Rantzer, 2006].

- Later on, several methods for **locally (or sparsely) connected** networks are considered, like: [Barooah and Hespanha, 2005] and [Khan and Moura, 2008].

Outline

- Introduction and Motivation
- Related Work
- **Problem Formulation**
- Scientific Background
  - Least-Squares Estimation
  - Kalman Filtering
- Algorithms
  - LS algorithm and KF Framework
  - Optimal Centralized Algorithm
  - Decentralized Algorithm
- Convergence Analysis
- Extensions
- Clock Skew Estimation
- Numerical Results
- Conclusions and Future Work

Problem Formulation

- **System Model:** $$T_i = \alpha_i t + \tau_i$$

α_i - Skew (rate deviation) parameter

τ_i - Offset parameter

t - Real time (or reference time)

T_i - Local clock (node number i)

Our goal is to synchronize each node in the network with the reference time. This is equivalent to estimate α_i and τ_i for each network's node. The algorithm has to be decentralized (local broadcasts) and to converge to the optimal centralized solution.

Assumptions:

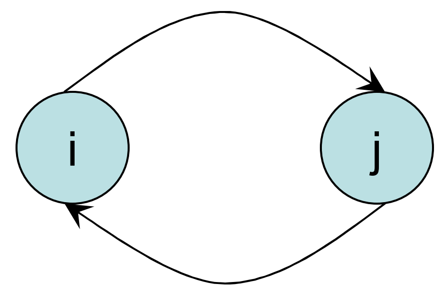
1. Time-invariant offsets.
2. All the clocks run exactly at the same speed (i.e., no skew: $\alpha_i = \alpha_j = 1$).

Problem Formulation (Cont.)

- **<u>The measurements:</u>**

Transmission delay   Noise (queuing delay)

$$\Delta T_{ij}(k_m) = R_j(k_m) - T_i(k_m) = x_{ij}(k_m) - \tau_i + \tau_j + \varepsilon_{ij}$$



Received time at j   Estimated Offset at j

Transmission time at i

Assuming symmetric transmission delay leads to the following relative measurements:

$$\hat{O}_{ij} = \frac{1}{2}\left(\Delta T_{ij}(k_m) - \Delta T_{ji}(k_m)\right) = \tau_j - \tau_i + \varepsilon_{ij}$$

Problem Formulation (Cont.)

- In order to apply the Kalman Filter, we need to find the state space model.
- Define the vector: $\underline{x}(n) \doteq (\tau_1 = 0, \tau_2, \ldots \tau_N)^T$
- **State space model:**

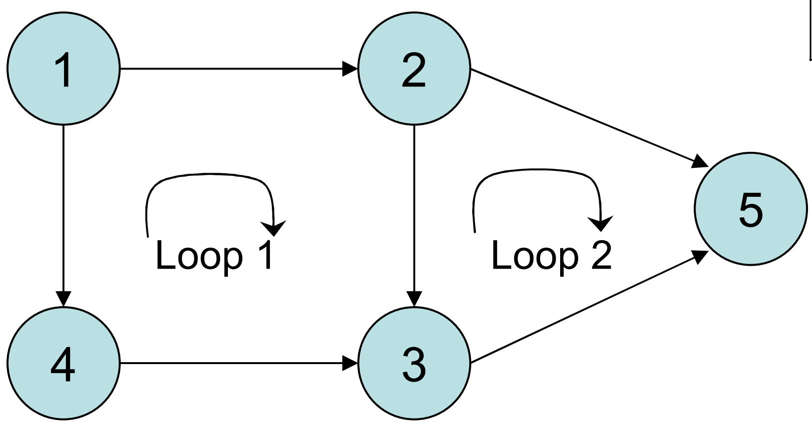
$$\begin{cases} \underline{x}(n+1) = \underline{x}(n) + \underline{w}(n) \\ y(n) = A^T \underline{x}(n) + \underline{v}(n) \end{cases}$$

$A =$

	(1,2)	(2,3)	(3,4)	(1,4)	(2,5)	(3,5)
1	+1	0	0	+1	0	0
2	−1	+1	0	0	+1	0
3	0	−1	+1	0	0	+1
4	0	0	−1	−1	0	0
5	0	0	0	0	−1	−1

Here, A is the reduced incidence matrix.

<u>Example:</u>



Outline

- Introduction and Motivation
- Related Work
- Problem Formulation
- **Scientific Background**
  - Least-Squares Estimation
  - Kalman Filtering
- Algorithms
  - LS algorithm and KF Framework
  - Optimal Centralized Algorithm
  - Decentralized Algorithm
- Convergence Analysis
- Extensions
- Clock Skew Estimation
- Numerical Results
- Conclusions and Future Work

Scientific Background

Least-Squares Estimation

- Minimizing a quadratic function: $\Pi = \sum_{i=1}^{n} \left(d_i\right)^2 = \left\|\underline{d}\right\|^2 = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2 \to \min$.
- The basic technique for computing the regression coefficients and very common in several domains: convex optimization, signal-processing, control, statistics…
- In the non-deterministic case, it gives an estimator that is equivalent to the LMMSE and to the Maximum-Likelihood estimator (for the Gaussian case).
- It is possible to assign a weight to each measurement according to its accuracy: the Weighted Least-Squares (**WLS**) case.
- The algorithm can be recursive (**RLS**) and decentralized (**DLS**).

Scientific Background (Cont.)

$$\begin{cases} \underline{x}(k+1) = \Phi(k+1,k)\underline{x}(k) + \Gamma(k+1,k)\underline{w}(k) + \Psi(k+1,k)\underline{u}(k) \\ y(k+1) = H(k+1)\underline{x}(k+1) + \underline{v}(k+1) \end{cases}$$

Kalman Filtering

- Given the above state space model.
- Assuming that $\{\underline{w}(k)\}$ and $\{\underline{v}(k)\}$ are independent white Gaussian noises with zero mean and covariances $Q(k) \geq 0$ and $R(k) > 0$ respectively.
- The initial state of the system $\underline{x}(0)$ is uncorrelated with the noises and verifies:

$$E\left[\underline{x}(0)\right] = \underline{m}_x(0) \; ; \; \mathrm{cov}\left[\underline{x}(0)\right] = E\left[\left(\underline{x}(0) - \underline{m}_x(0)\right)\left(\underline{x}(0) - \underline{m}_x(0)\right)^T\right] = P_x(0)$$

- The state estimation cycle is divided into two steps:

Time update (prediction)	Measurement update
$\begin{cases} \hat{\underline{x}}(k+1 \mid k) = \Phi(k+1,k)\hat{\underline{x}}(k \mid k) + \Psi(k+1,k)\underline{u}(k) \\ P(k+1 \mid k) = \Phi(k+1,k)P(k \mid k)\Phi^T(k+1,k) + \Gamma(k+1,k)Q(k)\Gamma^T(k+1,k) \end{cases}$	$\begin{cases} \hat{\underline{x}}(k+1 \mid k+1) = \hat{\underline{x}}(k+1 \mid k) + K(k+1)\left[\underline{z}(k+1) - H(k+1)\hat{\underline{x}}(k+1 \mid k)\right] \\ K(k+1) = P(k+1 \mid k)H^T(k+1)\left[H(k+1)P(k+1 \mid k)H^T(k+1) + R(k+1)\right]^{-1} \\ P(k+1 \mid k+1) = \left[I - K(k+1)H(k+1)\right]P(k+1 \mid k) \end{cases}$

Initialization: $\hat{\underline{x}}(0 \mid 0) = \underline{m}_x(0) \; ; \; P(0 \mid 0) = P_x(0)$

Scientific Background (Cont.)

- The Kalman Filter is the LMMSE estimator and MMSE for the Gaussian case.

- **<u>The KF as a LS problem:</u>**

  The minimizing solution of the following constrained deterministic optimization problem is equivalent to the MMSE solution (and to the MAP estimator) under the Gaussian assumption.

$$\min_{\{\underline{x}_n\},\{\underline{w}_n\}} \left\{ \begin{aligned} & J_k = \frac{1}{2}\left(\underline{x}(0)-\overline{x}_0\right)^T P_0^{-1}\left(\underline{x}(0)-\overline{x}_0\right)+\frac{1}{2}\sum_{n=0}^{k-1}(\underline{w}_n)^T Q_n^{-1}(\underline{w}_n)+ \\ & +\frac{1}{2}\sum_{n=0}^{k}(\underline{y}_n-H_n\underline{x}_n)^T R_n^{-1}(\underline{y}_n-H_n\underline{x}_n) \end{aligned} \right\}$$

$$\boxed{\underline{x}_{LS}=\underline{x}_{MMSE}=\underline{x}_{MAP}}$$

$$s.t.\ \underline{x}_{n+1}=\Phi_n\underline{x}_n+\Gamma_n\underline{w}_n\,,\qquad n=0,\ldots,k-1$$

Outline

Algorithms

CTP or LS Algorithm [Gurewitz, Cidon, Sidi, 2003 and Solis, Borkar, Kumar, 2005]

$$\tau_i = \frac{1}{|N_i|} \sum_{j \in N_i} \left(\hat{O}_{ji} + \tau_j \right)$$

- A decentralized algorithm that outperforms NTP without increasing complexity.
- Each node computes its estimated offsets by the average on its neighbors.
- Minimize the following quadratic function:

$$J = (y - A^T \underline{x})^T (y - A^T \underline{x}) = \sum_{\substack{i,j \\ j \in N_i}} \left(\hat{O}_{ji} - \tau_i + \tau_j \right)^2$$

- We will extend this result using a Kalman Filter framework.

Algorithms (Cont.)

Kalman Filter Framework

$$\begin{cases} \underline{x}(n+1) = \underline{x}(n) \\ y(n) = A^T \underline{x}(n) + \underline{v}(n) \end{cases}$$

- State space model of the system:
- Start with the pair of parameters $\overline{x}_0, P_0$ the goal is to find $\hat{\underline{\tau}}_{opt}$ by using the KF in a centralized fashion, or equivalently to minimize J:

$$J = (\underline{x} - \overline{x}_0)^T P_0^{-1} (\underline{x} - \overline{x}_0) + (y - A^T \underline{x})^T R^{-1} (y - A^T \underline{x}) \xrightarrow{\min} \hat{x}(0)$$

- <u>Several cases:</u>

1) P_0^{-1}=0, R^{-1}=I, i.e., the regular Least-Squares problem.
2) P_0^{-1}=0, R^{-1} is a diagonal matrix: the Weighted Least Squares problem.
3) P_0^{-1} can be a diagonal or a non-diagonal matrix.

Optimal Centralized Algorithm

- Compute the gradient and set it to zero:

$$\nabla_{\underline{x}} J = \left(AR^{-1}A^{T} + P_0^{-1} \right)\underline{x} - \left(AR^{-1} \right) y - P_0^{-1}\overline{x}_0 = 0$$

$$\boxed{\underline{x} = \left(AR^{-1}A^{T} + P_0^{-1} \right)^{-1} \left(AR^{-1}y + P_0^{-1}\overline{x}_0 \right)}$$

- The corresponding error covariance matrix:

$$E\left[\left(\underline{x} - \hat{\underline{x}} \right)\left(\underline{x} - \hat{\underline{x}} \right)^{T} \right] = \left(AR^{-1}A^{T} + P_0^{-1} \right)^{-1}$$

- Prohibitively expensive in terms of energy consumption, bandwidth and communication time.

- We need to develop a decentralized version for the general case.

Decentralized Algorithm

Let us develop the decentralized algorithm for the case where R^{-1} and P_0^{-1} are diagonal matrices.

$$J = (\underline{x} - \overline{x}_0)^T P_0^{-1} (\underline{x} - \overline{x}_0) + (y - A^T \underline{x})^T R^{-1} (y - A^T \underline{x})$$

$$\frac{\partial J}{\partial \tau_i} = \left(A R^{-1} A^T \right)_i \underline{x} - \left(A R^{-1} \right)_i y + \left(P_0^{-1} \right)_{i*} \underline{x} - \left(P_0^{-1} \right)_{i*} \overline{x}_0 = 0$$

Here, $\left(P_0^{-1} \right)_{i*}$ is the i-th row of the matrix P_0^{-1}.

This implies: $$\tau_i \left(\sum_{j \in N_i} \frac{1}{r_{ji}} + \frac{1}{p_i} \right) = \sum_{j \in N_i} \frac{1}{r_{ji}} \left(\hat{O}_{ji} + \tau_j \right) + \frac{1}{p_i} \cdot \tau_i(0)$$

(Weighting Average) $$\boxed{\tau_i = \frac{1}{\left(\sum_{j \in N_i} \frac{1}{r_{ji}} + \frac{1}{p_i} \right)} \cdot \left[\sum_{j \in N_i} \frac{1}{r_{ji}} \left(\hat{O}_{ji} + \tau_j \right) + \frac{\tau_i(0)}{p_i} \right]}$$

Decentralized Algorithm (Cont.)

The results are presented in the following table.

LS $P_0^{-1} = 0 \; ; \; R = I$	$\tau_i = \frac{1}{\lvert N_i \rvert} \sum_{j \in N_i} \left(\hat{O}_{ji} + \tau_j \right)$
WLS R^{-1} is diagonal and PSD.	$\tau_i = \frac{1}{\sum_{j \in N_i} \frac{1}{r_{ji}}} \cdot \sum_{j \in N_i} \frac{1}{r_{ji}} \left(\hat{O}_{ji} + \tau_j \right)$
DKF-1 P_0^{-1} is diagonal.	$\tau_i = \frac{1}{\left(\sum_{j \in N_i} \frac{1}{r_{ji}} + \frac{1}{p_i} \right)} \cdot \left[\sum_{j \in N_i} \frac{1}{r_{ji}} \left(\hat{O}_{ji} + \tau_j \right) + \frac{\tau_i(0)}{p_i} \right]$
DKF-2 P_0^{-1} is non-diagonal.	$\tau_i = \frac{1}{\left(\sum_{j \in N_i} \frac{1}{r_{ji}} + \left(P_0^{-1} \right)_{ii} \right)} \left[\sum_{j \in N_i} \frac{1}{r_{ji}} \left(\hat{O}_{ji} + \tau_j \right) + \left(P_0^{-1} \right)_{ii} \tau_i(0) - \sum_{\substack{k=1 \\ k \neq i}}^{N} \left(P_0^{-1} \right)_{ik} \left(\tau_k - \tau_k(0) \right) \right]$

Decentralized Algorithm (Cont.)

Remarks

- We showed that the WLS and the KF algorithms are equivalent to the Maximum-Likelihood and the Maximum-A-Posteriori estimators, respectively under the appropriate assumptions.
- We also explained that the WLS case can be written in a decentralized version of the Jacobi algorithm.
- One can implement the above equations through a synchronous algorithm of the form (for DKF-1):

$$\hat{\tau}_i^{(k+1)}(1) = \frac{1}{\left(\sum_{j \in N_i} \frac{1}{r_{ji}} + \frac{1}{p_i}\right)} \cdot \left[\sum_{j \in N_i} \frac{1}{r_{ji}} \left(\hat{O}_{ji}^{(1)} + \hat{\tau}_j^{(k)}(1)\right) + \frac{\tau_i(0)}{p_i}\right] \quad i = 2, 3, \ldots N$$

Problems:

1. After the first step, the covariance matrix of the KF is not diagonal anymore.
2. Convergence Analysis of the decentralized algorithms (4 cases).

Outline

- Introduction and Motivation
- Related Work
- Problem Formulation
- Scientific Background
    - Least-Squares Estimation
    - Kalman Filtering
- Algorithms
    - LS algorithm and KF Framework
    - Optimal Centralized Algorithm
    - Decentralized Algorithm
- **Convergence Analysis**
- Extensions
- Clock Skew Estimation
- Numerical Results
- Conclusions and Future Work

Convergence Analysis

Theorem 1

Suppose that:

- A single set of measurements is available.
- *The matrix* R is diagonal and PSD, that is: $\infty > \left(r_{ji}\right)^{-1} \geq 0 \;\forall i, j$.
- *The offsets are time-invariant.*
- *The initial state vector* $\overline{x}_0$ *is* known.
- *The initial covariance matrix* P_0 is known and *verifies:* $\begin{cases} \sum_j \left(P_0^{-1}\right)_{ij} \geq 0 \\ \left(P_0^{-1}\right)_{ii} \geq 0 \\ \left(P_0^{-1}\right)_{ij} \leq 0 \;\; (i \neq j) \end{cases}$

Then:

If the previous clock adjustment operation is applied (in a synchronous way) by all nodes ($i = 2, 3, \ldots N$) in all iterations, the set of estimated offsets $\hat{\tau}_i$ converges to the set of offsets that minimize the objective function:

$$J = (\underline{x} - \overline{x}_0)^T P_0^{-1} (\underline{x} - \overline{x}_0) + (y - A^T \underline{x})^T R^{-1} (y - A^T \underline{x})$$

namely, the set of offsets that would have been obtained by performing the centralized optimal protocol.

Convergence Analysis (Cont.)

- The proof for the general case can be found in the thesis. Here, for our convenience, we present the proof for the DKF-1 case.
- The convergence analysis for the multiple measurement case can be found in the thesis.
- **Proof** for the DKF-1 case:

  The synchronous iteration can be written in vector form as the following:

$$\underline{\tau}^{(k+1)} = \underline{\tau}^{(k)} - \left(\tilde{D} + P_0^{-1}\right)^{-1}\left(AR^{-1}A^T\underline{\tau}^{(k)} - AR^{-1}y - P_0^{-1}\overline{x}_0 + P^{-1}\underline{\tau}^{(k)}\right) \quad \left(\tilde{D}^{-1}\right)_{ij} = \begin{cases} \dfrac{1}{\sum_{j \in N_i} \dfrac{1}{r_{ji}}} & i = j \\ 0 & otherwise \end{cases}$$

The optimal solution (equivalent to perform the centralized protocol) is:

$$\underline{\tau}^* = \left(AR^{-1}A^T + P_0^{-1}\right)^{-1}\left(AR^{-1}y + P_0^{-1}\overline{x}_0\right)$$

Convergence Analysis (Cont.)

Let us define: $\overline{\tau}^{(k)} \doteq \underline{\tau}^{(k)} - \underline{\tau}^{*}$

Then we obtain:

$$\overline{\tau}^{(k+1)} \doteq \underline{\tau}^{(k+1)} - \underline{\tau}^{*} = \underline{\tau}^{(k)} - \left(\tilde{D} + P_0^{-1}\right)^{-1} \left(AR^{-1}A^T\underline{\tau}^{(k)} - AR^{-1}y - P_0^{-1}\overline{x}_0 + P_0^{-1}\underline{\tau}^{(k)}\right) - \left(AR^{-1}A^T\right)^{-1} AR^{-1}y$$

$$\overline{\tau}^{(k+1)} = \left[I - \left(\tilde{D} + P_0^{-1}\right)^{-1} \left(AR^{-1}A^T + P_0^{-1}\right)\right]\underline{\tau}^{(k)} +$$

$$+\left(\tilde{D} + P_0^{-1}\right)^{-1} \left(AR^{-1}A^T + P_0^{-1}\right)\underbrace{\left(AR^{-1}A^T + P_0^{-1}\right)^{-1} AR^{-1}y}_{\underline{\tau}^{*}} - \underbrace{\left(AR^{-1}A^T\right)^{-1} AR^{-1}y}_{\underline{\tau}^{*}}$$

$$\overline{\tau}^{(k+1)} = \left[I - \left(\tilde{D} + P_0^{-1}\right)^{-1} \left(AR^{-1}A^T + P_0^{-1}\right)\right]\left(\underline{\tau}^{(k)} - \underline{\tau}^{*}\right)$$

Defining: $M \doteq I - \left(\tilde{D} + P_0^{-1}\right)^{-1} \left(AR^{-1}A^T + P_0^{-1}\right)$ we have the following iteration equation:

$$\overline{\tau}^{(k+1)} = M\overline{\tau}^{(k)}$$

The necessary and sufficient condition for convergence is that the spectral radius of M is strictly smaller than 1.

Convergence Analysis (Cont.)

- The structure of the matrix M can be determined by inspection as the following:

$$M_{ij} = \begin{cases} 0 & i = j \\ \dfrac{\frac{1}{r_{ji}}}{\sum_{j \in N_i} \frac{1}{r_{ji}} + \frac{1}{p_i}} & i \neq j \; and \; i, j \; are \; neighbors \\ 0 & otherwise \end{cases}$$

- We can easily check that M is both non-negative and sub-stochastic if:

$$\infty > \frac{1}{r_{ji}} \geq 0; \; \forall i, j \quad and \quad \frac{1}{p_i} \geq 0 \; \forall i \; \Rightarrow \boxed{\rho(M) < 1} \qquad Q.E.D$$

- In the most general case the convergence conditions are given by (diagonal dominance):

$$\infty > \frac{1}{r_{ji}} \geq 0; \; \forall i, j \qquad \sum_{\forall j} \left(P_0^{-1}\right)_{ij} \geq 0; \quad \begin{cases} \left(P_0^{-1}\right)_{ii} \geq 0 \\ \left(P_0^{-1}\right)_{ij} \leq 0 \end{cases} \Rightarrow \boxed{\left(P_0^{-1}\right)_{ii} \geq -\sum_{\forall j \neq i} \left(P_0^{-1}\right)_{ij} \geq 0}$$

- Moreover, we show that if the a-priori inverse covariance matrix verifies the convergence conditions, the a-posteriori inverse covariance matrix will verify them too.

Outline

- Introduction and Motivation
- Related Work
- Problem Formulation
- Scientific Background
    - Least-Squares Estimation
    - Kalman Filtering
- Algorithms
    - LS algorithm and KF Framework
    - Optimal Centralized Algorithm
    - Decentralized Algorithm
- Convergence Analysis
- **Extensions**
- Clock Skew Estimation
- Numerical Results
- Conclusions and Future Work

Recursive Version: Multiple Measurement Update

- The main problem is that if the matrix P^{-1} is not a diagonal matrix, each node needs to communicate with every other node and not only with its neighbors. Thus, in this case each node has to know the global topology of the entire network. The solution is to look for a recursive algorithm.
- Assume that the objective function is given by:

$$J = (\underline{x} - \overline{x}_0)^T P^{-1} (\underline{x} - \overline{x}_0) + \sum_{k=1}^{n} (y(k) - A^T \underline{x})^T R^{-1} (y(k) - A^T \underline{x})$$

- By repeating the one-measurement derivation for the multiple-measurement case, we can obtain:

$$\hat{\tau}_i^{(k+1)}(n) = \hat{\tau}_i(n-1) + \frac{1}{n\sum_{j\in N_i}\frac{1}{r_{ji}} + \frac{1}{p_i}} \cdot \left\{ \sum_{j\in N_i} \frac{1}{r_{ji}} \left[\hat{O}_{ji}^{(n)} - \underbrace{\left(\hat{\tau}_i(n-1) - \hat{\tau}_j(n-1)\right)}_{\text{the estimated measurement}} + n\left(\hat{\tau}_j^{(k)}(n) - \hat{\tau}_j(n-1)\right) \right] \right\}$$

Recursive Version (Cont.)

The recursive, iterative and decentralized (optimal) algorithm is given in its final form by:

$$\hat{\tau}_i^{(k+1)}(n) = \hat{\tau}_i(n-1) + \frac{1}{\left[I_i(n)\right]^{-1}} \cdot \left\{ \sum_{j \in N_i} \frac{1}{r_{ji}} \cdot \left[\hat{O}_{ji}^{(n)} - \left(\hat{\tau}_i(n-1) - \hat{\tau}_j^{(k)}(n) \right) \right] + \left(n-1\right) \cdot \left[\sum_{j \in N_i} \frac{1}{r_{ji}} \cdot \left(\hat{\tau}_j^{(k)}(n) - \hat{\tau}_j(n-1) \right) \right] \right\}$$

$$\begin{cases} \left[I_i(n)\right]^{-1} = \left[I_i(n-1)\right]^{-1} + \sum_{j \in N_i} \frac{1}{r_{ji}} = \frac{1}{p_i} + n \cdot \sum_{j \in N_i} \frac{1}{r_{ji}} \\ \left[I_i(0)\right]^{-1} = \left[P_i(0)\right]^{-1} = \frac{1}{p_i} \end{cases} \qquad i = 2, \ldots, N$$

We showed that the elements of I are the diagonal elements of P. Hence, the variances of the estimates at each step are obtained.

Recursive Version (Cont.)

Remarks

- We obtained a decentralized, synchronous recursive algorithm that converges to the optimal centralized KF solution. The main advantage of this algorithm is its local nature; each network's node needs to communicate only with its neighbors.
- The equations are not similar to the KF equations.
- We note that the same equations may be obtained from the information form of the KF after appropriate manipulations.
- An alternative approach is to consider the sub-optimal algorithm that neglects the off-diagonal terms of the inverse covariance matrix. It reduces the complexity but looses the optimality.

Incorporating a Discount Factor

The objective function to be minimized is ($0 < \gamma < 1$):

$$J = \gamma^n (\underline{x} - \overline{x}_0)^T P_0^{-1} (\underline{x} - \overline{x}_0) + \sum_{k=1}^{n} \left[\gamma^{n-k} (y(k) - A^T \underline{x})^T R^{-1} (y(k) - A^T \underline{x}) \right] \xrightarrow{\min} \hat{\underline{x}}_{opt}(n)$$

The recursive algorithm in this case:

$$\hat{\tau}_i^{(k+1)}(n) = \hat{\tau}_i(n-1) + \frac{1}{\left[I_i(n)\right]^{-1}} \cdot \left\{ \sum_{j \in N_i} \frac{1}{r_{ji}} \cdot \left[\hat{O}_{ji}^{(n)} - \left(\hat{\tau}_i(n-1) - \hat{\tau}_j^{(k)}(n) \right) \right] + \left(\sum_{k=1}^{n-1} \gamma^{n-k} \right) \cdot \left[\sum_{j \in N_i} \frac{1}{r_{ji}} \cdot \left(\hat{\tau}_j^{(k)}(n) - \hat{\tau}_j(n-1) \right) \right] \right\}$$

$$\begin{cases} \left[I_i(n)\right]^{-1} = \gamma \cdot \left[I_i(n-1)\right]^{-1} + \sum_{j \in N_i} \frac{1}{r_{ji}} = \left(\sum_{k=1}^{n-1} \gamma^{n-k} \right) \cdot \sum_{j \in N_i} \frac{1}{r_{ji}} + \gamma^n \frac{1}{p_i} \\ \left[I_i(0)\right]^{-1} = \left[P_i(0)\right]^{-1} = \frac{1}{p_i} \end{cases}$$

This is a recursive synchronous and decentralized algorithm that computes the optimal offsets (in the MMSE sense) hence, equivalent to the Kalman Filter solution.

Additional Extensions

- We investigated the case where a process noise is incorporated in the dynamical state space equation.

$$\begin{cases} \underline{x}(n+1) = \underline{x}(n) + \underline{w}(n) \\ y(n) = A^T\, \underline{x}(n) + \underline{v}(n) \end{cases}$$

  In this latter, only a centralized algorithm was obtained.

- We improved the algorithms to handle with dynamic changes in the communication topology by considering temporary link failures (following the work of Barooah and Hespanha, 2005).

Outline

- Introduction and Motivation
- Related Work
- Problem Formulation
- Scientific Background
  - Least-Squares Estimation
  - Kalman Filtering
- Algorithms
  - LS algorithm and KF Framework
  - Optimal Centralized Algorithm
  - Decentralized Algorithm
- Convergence Analysis
- Extensions
- **Clock Skew Estimation**
- Numerical Results
- Conclusions and Future Work

Clock Skew Estimation

Combined Skew and Offset Estimation

- State space model that includes the skew: $\begin{cases} \underline{x}(n+1) = \underline{x}(n) + T_S \cdot \underline{b} \\ y(n) = A^T\, \underline{x}(n) + \underline{v}(n) \end{cases}$ $\quad \underline{b} \sim N(0, B)$

- Define the augmented state vector: $X(n) = \begin{pmatrix} \underline{x}(n) \\ \underline{b} \end{pmatrix}$ for which we have the following model:

$$\begin{cases} X(n+1) = \begin{pmatrix} \underline{x}(n+1) \\ \underline{b} \end{pmatrix} = \begin{pmatrix} I & T_S I \\ 0 & I \end{pmatrix} \begin{pmatrix} \underline{x}(n) \\ \underline{b} \end{pmatrix} = \Phi X(n) \\ y(n) = \begin{pmatrix} A^T & 0 \end{pmatrix} X(n) + \underline{v}(n) \end{cases}$$

- Centralized Kalman Filter algorithm: LS approach or KF equations.

- Decentralized implementation: decentralized Jacobi-like iterations generally diverge (spectral radius of the iteration matrix is bigger than 1). Future direction: relaxed Jacobi algorithm (gradient method) with small step size.

Clock Skew Estimation (Cont.)

Separate Skew and Offset Estimation

- Offset estimation using one of the previous methods. 3 methods are developed in order to estimate the skew parameters.
- The logarithmic method, introduced by (Karp et. al. 2003 and Solis et. Al. 2006).
- Two additional original methods: The multiplicative method and a state space based solution.
- Only the third method employs the same measurement format than the offset estimation problem.

Clock Skew Estimation (Cont.)

<u>The Logarithmic Method:</u>

If we perform the following substitution: $\begin{cases} \hat{O}_{ji} \to z_{ji} \triangleq \log\left(\dfrac{\Delta T_i}{\Delta R_j}\right) \\ \tau_i \to \beta_i = \log(\alpha_i) \end{cases}$

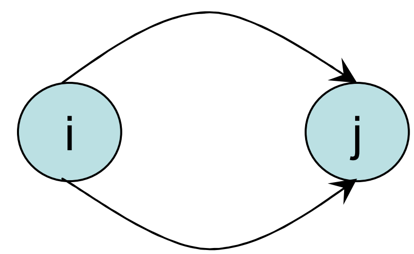
Then, one can easily show that we will obtain the same problem as the previous offset optimization problem.

The solution for the basic LS case is given by: $\boxed{\widehat{\beta}_i^{(k+1)} = \dfrac{1}{|N_i|} \sum_{j \in N_i} \left(z_{ji} + \widehat{\beta}_j^{(k)} \right)}$

The remaining question is how we obtain the measurements: z_{ji}

$$\frac{\Delta T_i}{\Delta R_j} = \frac{T_i(t_{k_2}) - T_i(t_{k_1})}{R_j(t_{k_2}) - R_j(t_{k_1})} \cong \frac{(\alpha_i t_2 + \cancel{\tau_i}) - (\alpha_i t_1 + \cancel{\tau_i})}{(\alpha_j t_2 + \cancel{\tau_j}) - (\alpha_j t_1 + \cancel{\tau_j})} = \frac{\alpha_i}{\alpha_j}$$

$$\Rightarrow z_{ji} \triangleq \log \frac{\Delta T_i}{\Delta R_j} \cong \log(\alpha_i) - \log(\alpha_j)$$



In practice, we will treat skew synchronization and offset synchronization on different time scales. That is, we will adjust the parameters α_i every τ_{skew} time units, whereas we will adjust the parameters τ_i every τ_{offset} time units, with: $\boxed{\tau_{skew} \gg \tau_{offset}}$

Clock Skew Estimation (Cont.)

- We have developed an additional method without requiring the application of the logarithm function on the sets of measurements. The corresponding objective function is given by:

$$J(\alpha) = \sum_{i,j \in N_i} \left(\frac{\Delta T_i}{\Delta R_j} - \frac{\alpha_i}{\alpha_j} \right)^2$$

- The main drawback of these two methods is that the measurements for the offset estimation and for the skew estimation are not similar.

- We proposed an additional method based on the state space model. First, we showed that adding a constant random noise (bias) to the dynamical equation is equivalent to relax the skew assumption.

$$\begin{cases} \underline{x}(n+1) = \underline{x}(n) + T_s \underline{b} = \underline{x}(0) + (n+1) T_s \cdot \underline{b} \\ y(n) = A^T \underline{x}(n) + \underline{v}(n) = A^T \left[\underline{x}(0) + n T_s \cdot \underline{b} \right] + \underline{v}(n) \end{cases} \Rightarrow \begin{cases} X(n+1) = \begin{pmatrix} \underline{x}(n+1) \\ \underline{b} \end{pmatrix} = \begin{pmatrix} I & T_s \cdot I \\ 0 & I \end{pmatrix} \begin{pmatrix} \underline{x}(n) \\ \underline{b} \end{pmatrix} = \Phi X(n) \\ y(n) = \begin{pmatrix} A^T & 0 \end{pmatrix} X(n) + \underline{v}(n) \end{cases}$$

Clock Skew Estimation (Cont.)

State Space based Solution

We consider the following decentralized sub-optimal algorithm: estimate the offsets after each set of measurements in a recursive way, and the skew parameters after T_b sets of measurements only (according to the pair of farthest measurements).

By operate on the state space equations, we obtain the same mathematical problem: $y(T_b) - y(0) = A^T T_b \cdot \underline{b} + \underline{v}(T_b) - \underline{v}(0)$

Just now, the noise is: $\underline{\tilde{v}} = \dfrac{\underline{v}(T_b) - \underline{v}(0)}{T_b} \sim N\left(0, \dfrac{2}{(T_b)^2} R\right)$

$$\begin{cases} x \to \underline{b} \\ y \to \dfrac{y(T_b) - y(0)}{T_b} \\ v \to \tilde{v} \end{cases}$$

The skew estimation algorithms are given by:

LS case	**KF** case
$\hat{b}_i^{(k+1)} = \dfrac{1}{T_b \cdot \lvert N_i \rvert} \sum_{j \in N_i} \left(\hat{O}_{ji}^{(T_b)} - \hat{O}_{ji}^{(0)} + T_b \cdot \hat{b}_j^{(k)} \right)$	$\hat{b}_i^{(k+1)} = \dfrac{1}{T_b \cdot \left(\sum_{j \in N_i} \dfrac{1}{2r_{ji}} + \dfrac{1}{B_i} \right)} \left[\sum_{j \in N_i} \dfrac{1}{2r_{ji}} \left(\hat{O}_{ji}^{(T_b)} - \hat{O}_{ji}^{(0)} + T_b \cdot \hat{b}_j^{(k)} \right) + T_b \cdot \dfrac{\overline{b}_i(0)}{B_i} \right]$

Clock Skew Estimation (Cont.)

- In brief, we have obtained several decentralized algorithms for estimating the skew parameters $\underline{b}$ with no dependence on the offsets $\underline{\tau}$ ($\alpha_i = b_i + 1$).

- In practice, we will estimate $\hat{\tau}_i \left(i = 2, \ldots, N \right)$ after each set of measurements, whereas $\hat{b}_i \left(i = 2, \ldots, N \right)$ is estimated according to $y\left(T_b\right)$ and $y\left(0\right)$ at the first cycle, $y\left(2T_b\right)$ and $y\left(T_b + 1\right)$ at the second cycle, etc. In each time interval, we will assume that the skew parameter remains constant.

- In summary, we have developed 3 different methods to comply with general clocks that including frequency and time offsets. The state space based solution gives the advantage that the algorithm uses the same measurements as in the offset estimation procedure, but we need to know the parameter T_b .

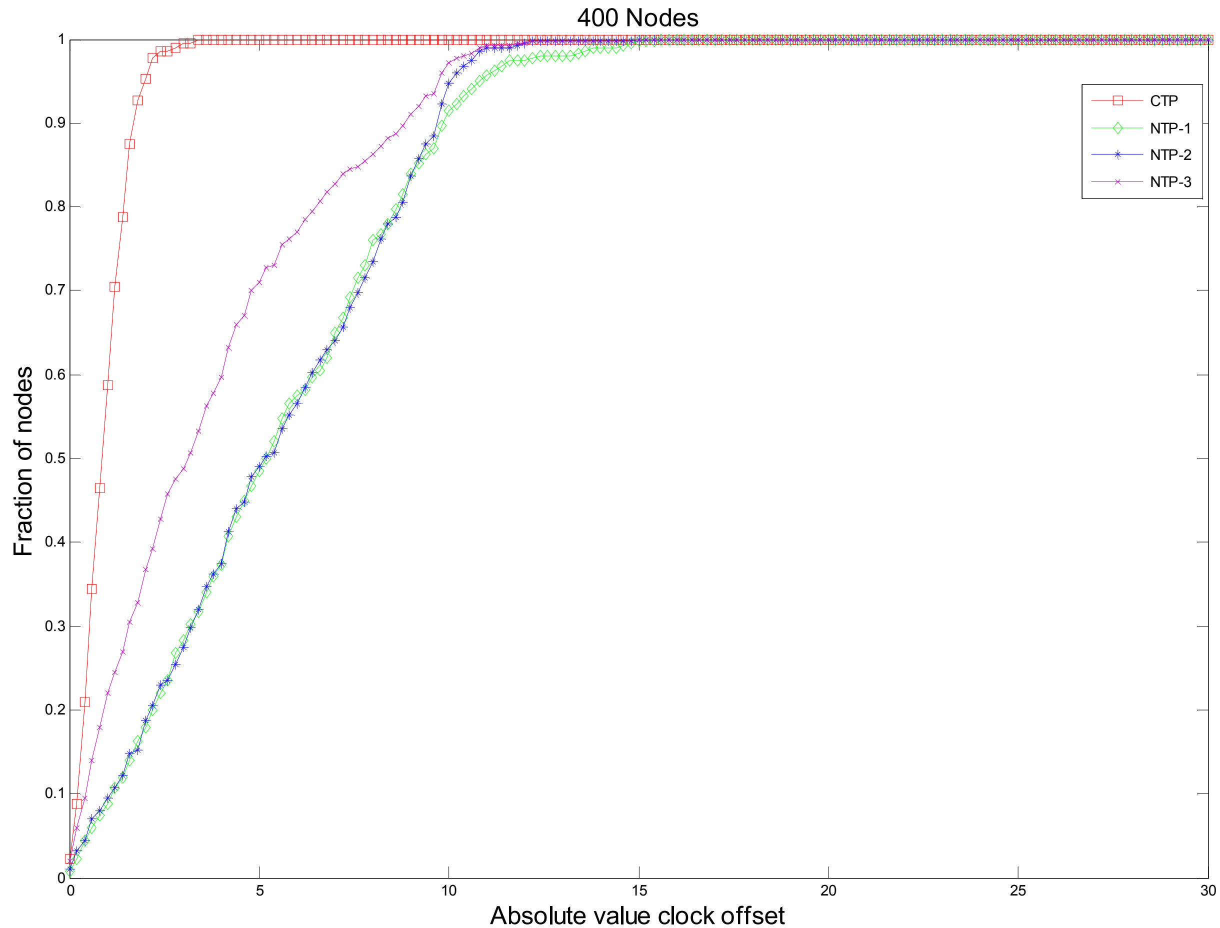
Outline

- Introduction and Motivation
- Related Work
- Problem Formulation
- Scientific Background
  - Least-Squares Estimation
  - Kalman Filtering
- Algorithms
  - LS algorithm and KF Framework
  - Optimal Centralized Algorithm
  - Decentralized Algorithm
- Convergence Analysis
- Extensions
- Clock Skew Estimation
- **Numerical Results**
- Conclusions and Future Work

Numerical Results

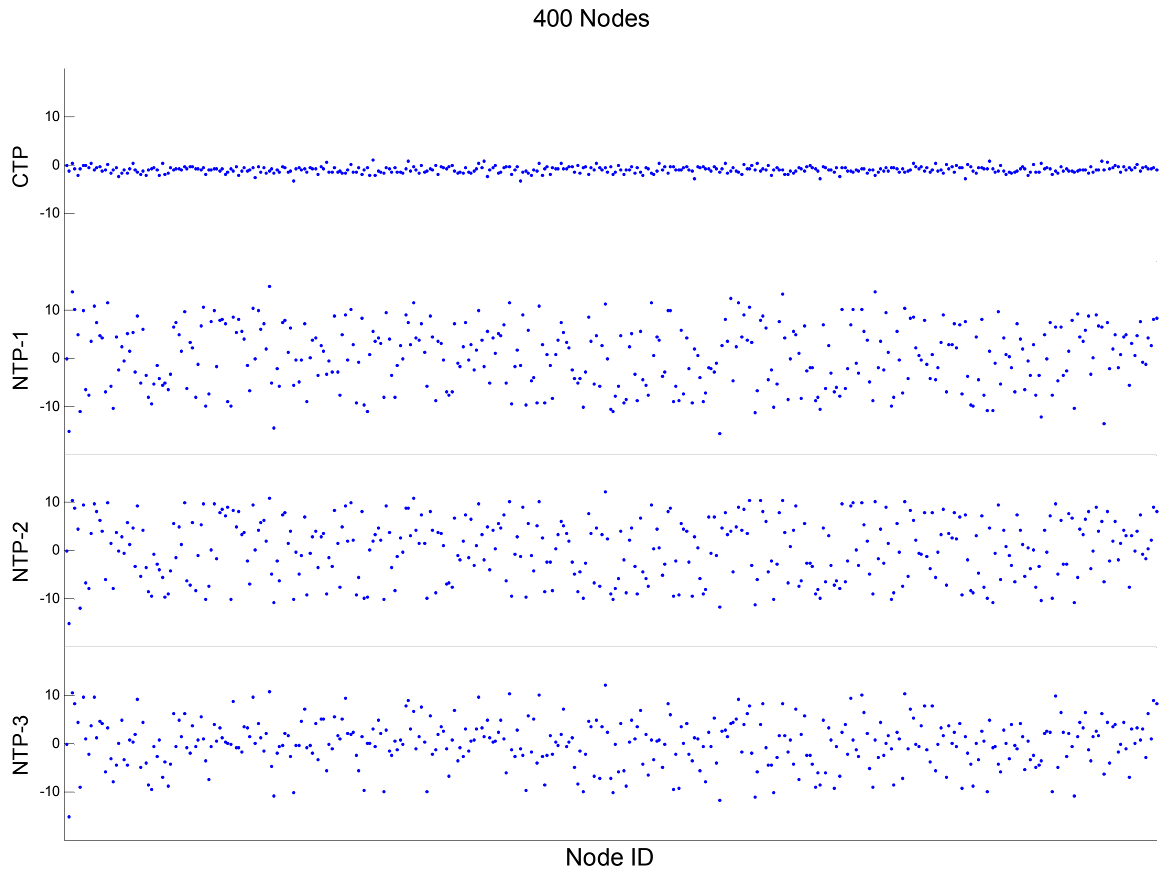
- The setup and the network topologies are based on the literature, and essentially on: Gurewitz, Cidon and Sidi, 2003.
- Compare the CTP (LS) with three different hierarchical versions of the Network Time Protocol (NTP).
- Analyze the convergence rate of the decentralized algorithm.
- Compare LS with WLS in several cases.
- Compare LS with DKF.
- Recursive Algorithm.

Numerical Results (Cont.)



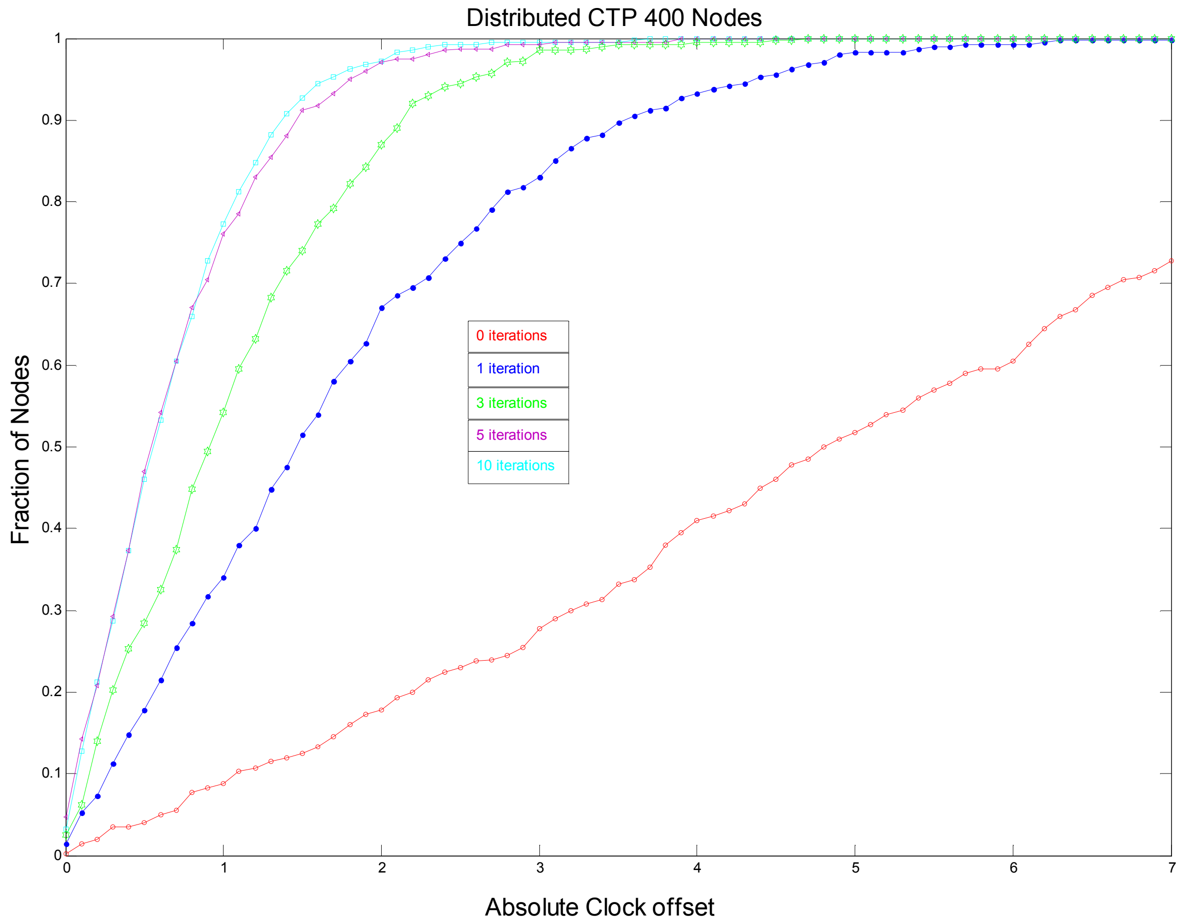
Distribution of the clock offsets for the different algorithms in a 400 node network.

Numerical Results (Cont.)



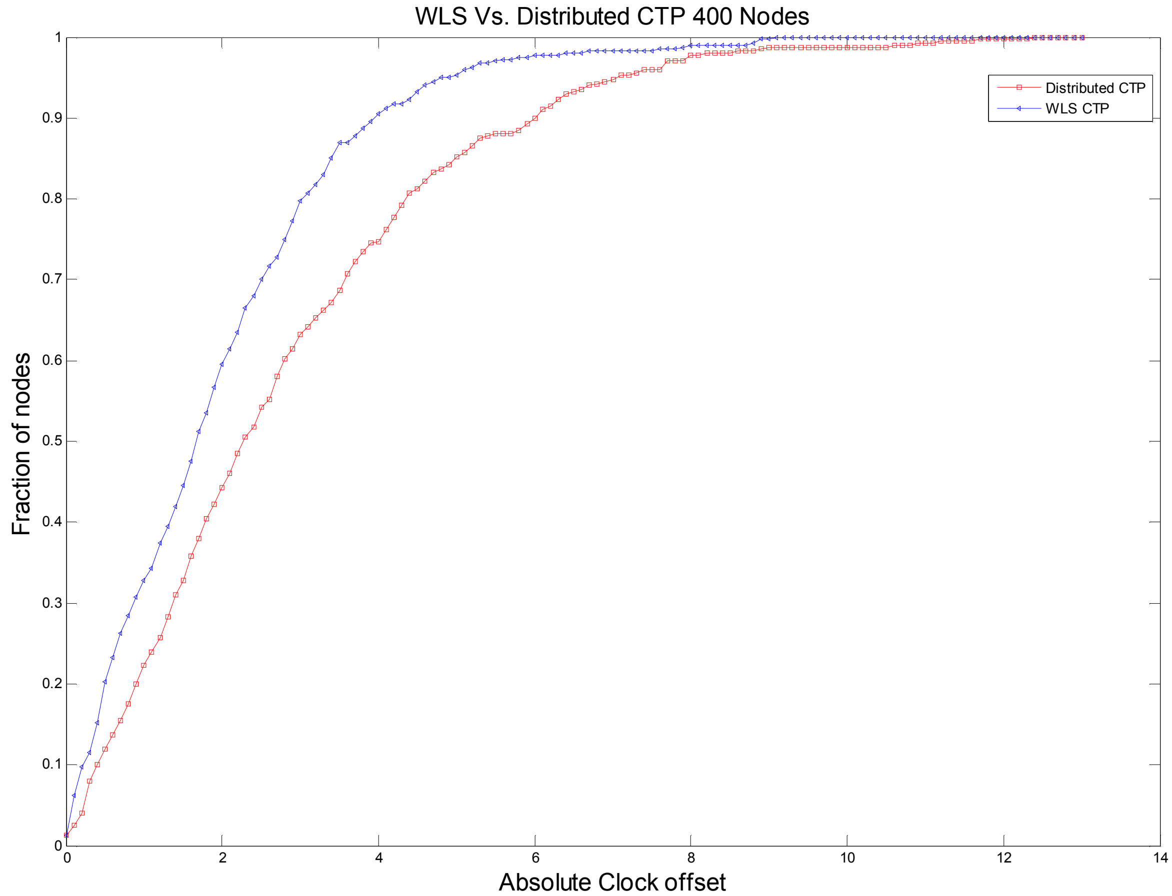
The clock offsets dispersion on a 400 node network.

Numerical Results (Cont.)



Rate convergence analysis of the decentralized CTP algorithm in a 400 node network.

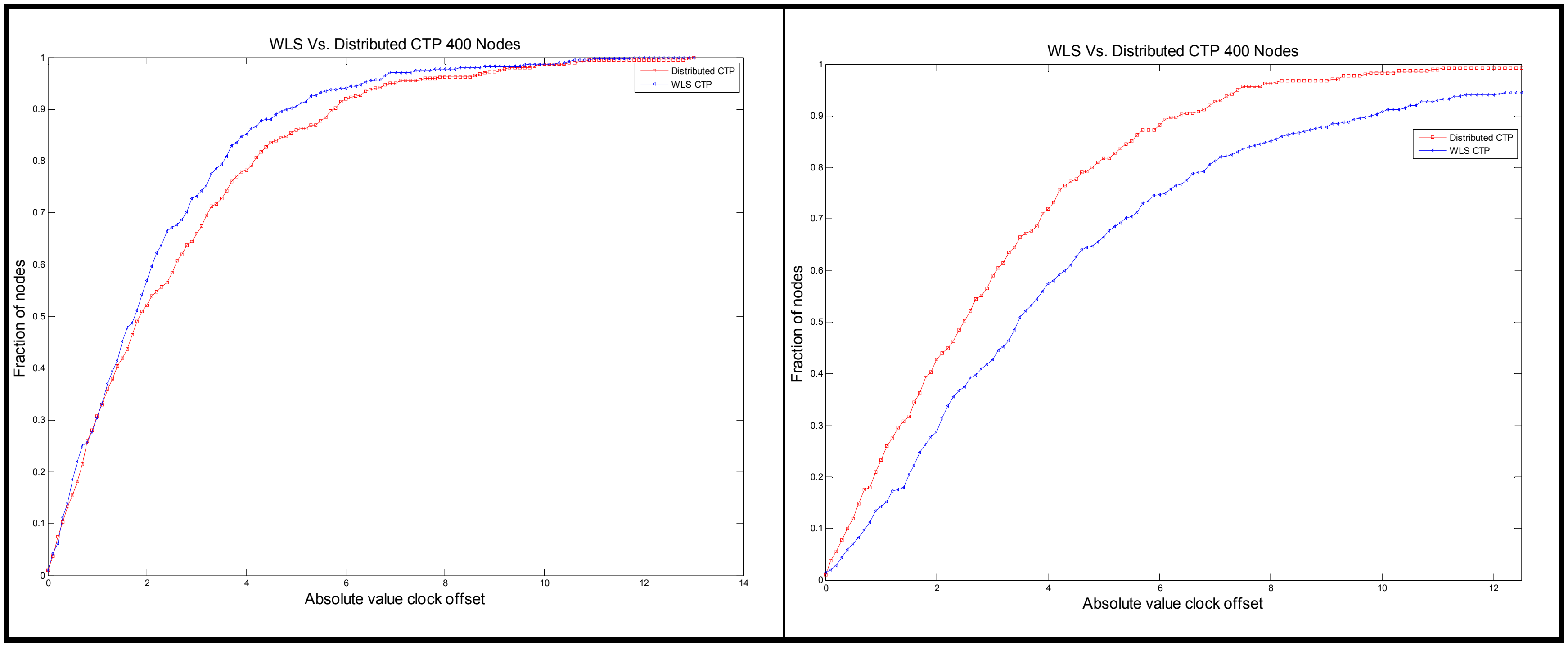
Numerical Results (Cont.)



Comparison between the decentralized CTP and WLS algorithms in a 400 node network.

Numerical Results (Cont.)

Analysis of the robustness of the matrix R using noise with two different variances.

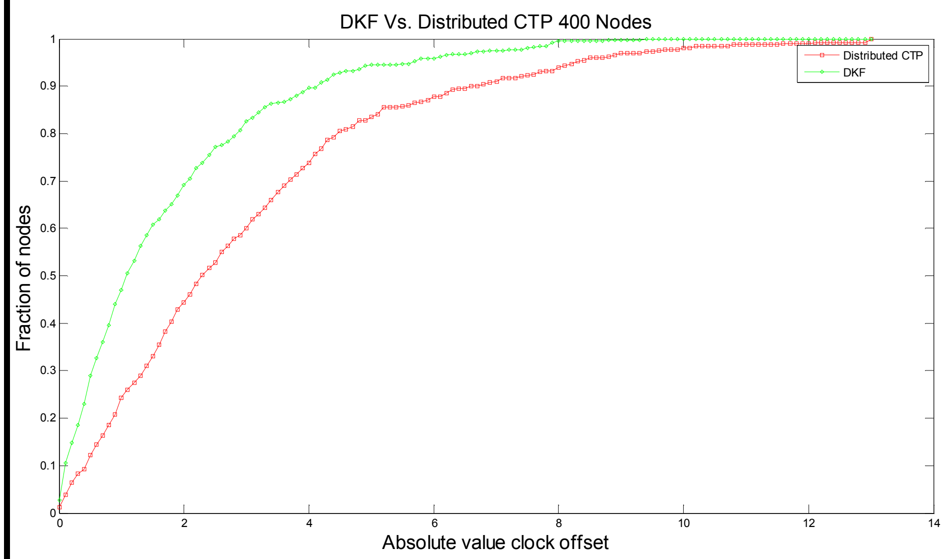


Comparison between the decentralized CTP and WLS algorithms (with additive Gaussian noises on R) in a 400 node network.

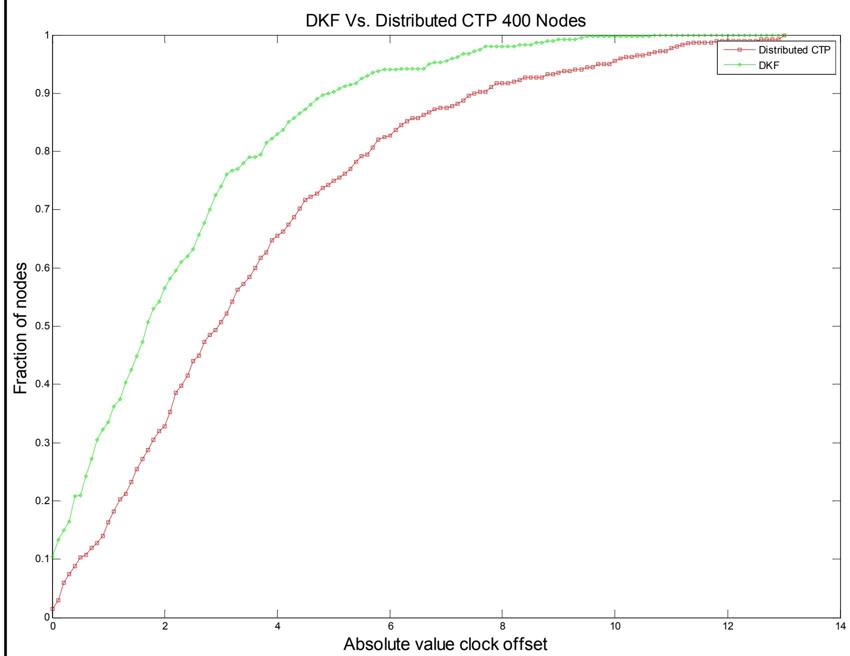
Numerical Results (Cont.)

Analysis of the DKF algorithm for different a-priori covariance matrix.

$$(P_0)_{ii} = \begin{cases} \sim U(0.01, 0.19)\ ; & \textit{half of the nodes} \\ \sim U(5, 10)\ ; & \textit{the remainder} \end{cases}$$



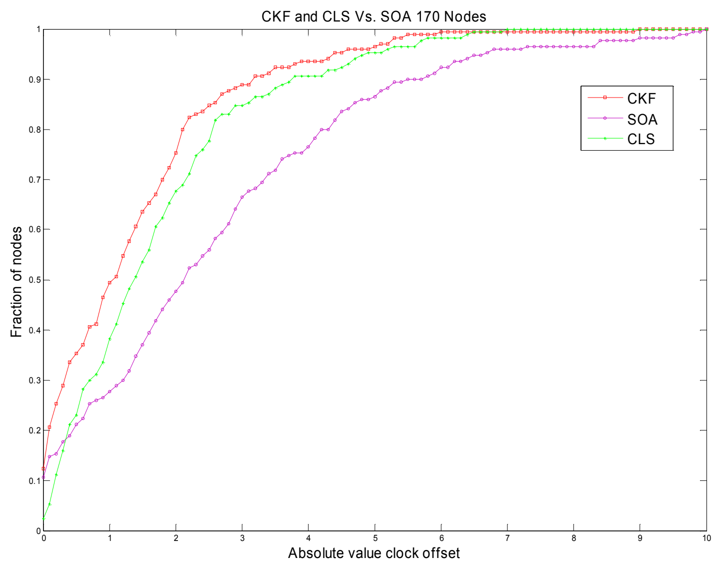
$$(P_0)_{ii} = \begin{cases} 0.01 & ; 10\%\ \textit{of the nodes} \\ \rightarrow \infty\ ; & \textit{the remainder} \end{cases}$$



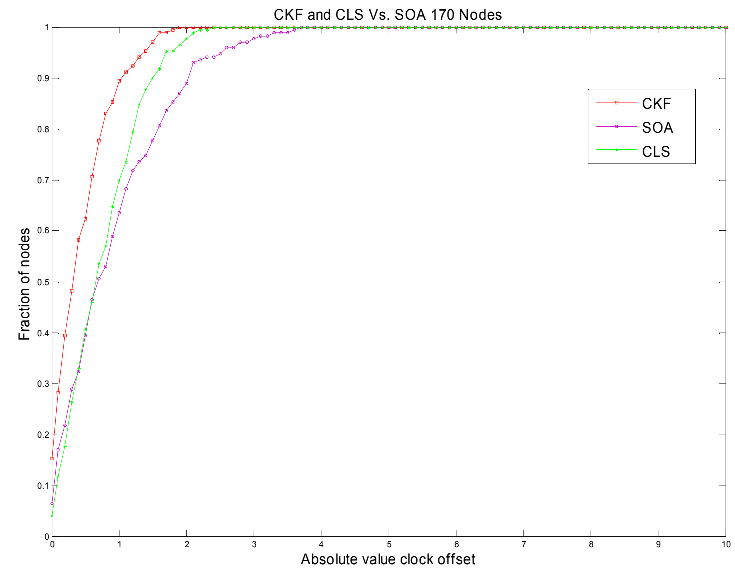
Comparison between the decentralized CTP and DKF algorithms in a 400 node network.

Recursive Algorithm

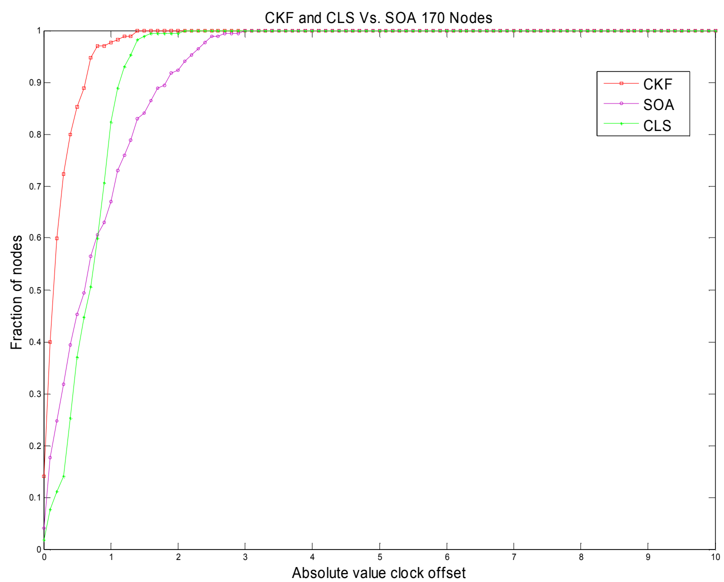
n=1



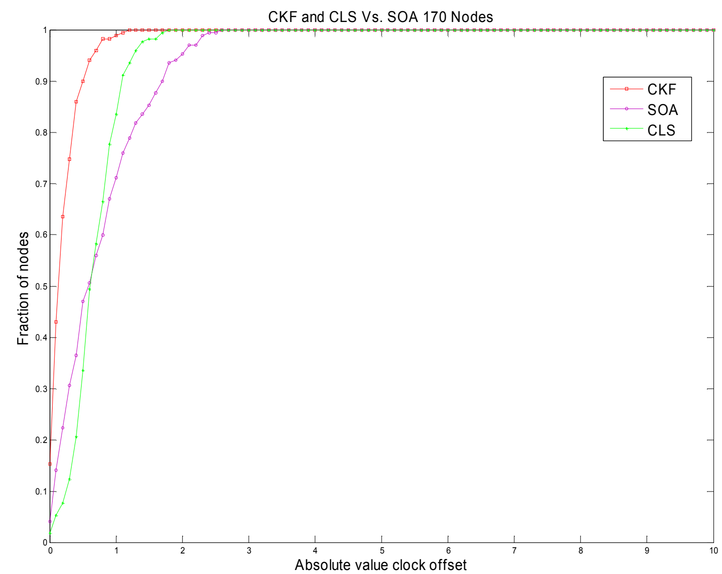
n=10



n=30



n=50



Outline

- Introduction and Motivation
- Related Work
- Problem Formulation
- Scientific Background
  - Least-Squares Estimation
  - Kalman Filtering
- Algorithms
  - LS algorithm and KF Framework
  - Optimal Centralized Algorithm
  - Decentralized Algorithm
- Convergence Analysis
- Extensions
- Clock Skew Estimation
- Numerical Results
- **Conclusions and Future Work**

Conclusions and Future Work

- Decentralized algorithms for estimating the offset at each network's node using a Kalman Filter framework were obtained.
- The main algorithm is both decentralized and recursive and converges to the optimal solution.
- Several extensions to the basic algorithm were considered.
- We treat the case of general clocks with both offsets and skews.
- The different algorithms were tested on typical networks. In most of the cases, the proposed algorithms outperform the NTP schemes and the LS method.

Conclusions and Future Work (Cont.)

We mention the following directions for future research:

- Solve optimally (in a decentralized way) the combined problem including both skews and offsets, and the case for which a process noise is incorporated.
- Extend to dynamic network topologies with time-varying offsets.
- Simulations of the case with non-uniform skews.
- Non-linear estimation of the distances between several agents.

Thank you for your attention!