Clock synchronization in multi-hop wireless networks

P.R. Kumar

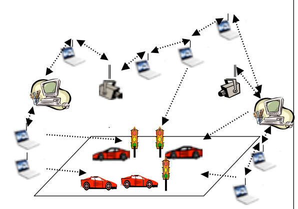
(with V. Borkar, N. Freris, S. Graham, A. Giridhar and K. Plarre, C. Robinson, H-J. Schuetz, R. Solis)

Dept. of Electrical and Computer Engineering, and
Coordinated Science Lab
University of Illinois, Urbana-Champaign

WISARD,
Bangalore, Jan 5-6, 2008

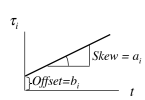
Clock synchronization in distributed systems

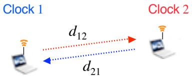
- Knowledge of time is important in Networks
  - Communication network protocols
    - Slotted protocols
  - Sensor network applications
    - Location by time of flight
  - Networked control
    - Coordination
- However no two clocks agree
- How to synchronize clocks in distributed systems?



Impossibility of synchronizing two clocks

- Linear clocks $\tau_i(t) = a_i t + b_i$



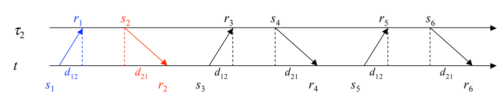


Reference Clock
$\tau_1(t) \equiv t$
$a_1=1, b_1=0$

- Need to determine a_2 and b_2
- Delays d_{12} and d_{21} in two directions
- **Theorem (Graham & K '04)**
  - It is impossible to determine all the four parameters $(d_{12}, d_{21}, a_2, b_2)$ through any packet exchanges

Proof of impossibility of synchronizing two clocks

$$r_1 = a_2(s_1 + d_{12}) + b_2$$



$$s_2 = a_2(r_2 - d_{21}) + b_2$$

$$\begin{bmatrix} r_1 \\ s_2 \\ r_3 \\ s_4 \\ \cdots \end{bmatrix} = \begin{bmatrix} s_1 & 1 & 0 & 1 \\ r_2 & 0 & -1 & 1 \\ s_3 & 1 & 0 & 1 \\ r_4 & 0 & -1 & 1 \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} a_2 \\ a_2 d_{12} \\ a_2 d_{21} \\ b_2 \end{bmatrix}$$

Rank 3:
Cannot estimate
4 parameters

Special case:
Parameters can be estimated if delays are assumed symmetric

General strongly connected network

- **Theorem (Freris & K '07)**

  - Every node can estimate the skews of all nodes
  - The offset vector $\hat{b} = (\hat{b}_i : 2 \le i \le n)$ is a free vector in R^{n-1} to choose.
  - Uncertainty set for delay vector $\hat{d} = (\hat{d}_{ij} : (i,j) \in E)$ is affine image of R^{n-1}

$$\hat{d}_{ij} = d^*_{ij} + \frac{1}{a_i}\hat{b}_i - \frac{1}{a_j}\hat{b}_j, \text{ where } d^*_{ij} := \frac{1}{a_i} r^{(k)}_{i,j} - \frac{1}{a_j} s^{(k)}_i$$

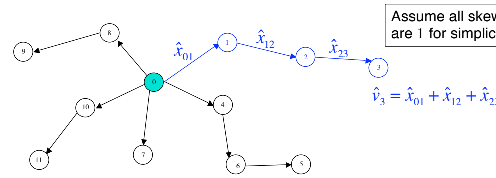
  - Under causality, uncertainty set for offset vector is compact polyhedron:

$$\frac{1}{a_i}\hat{b}_i - \frac{1}{a_j}\hat{b}_j \ge -d^*_{ij}, \text{ with } \hat{b}_1 := 0$$

  - Any sender node can predict the time at the receiver at which it receives packet

Gurewitz, Sidon and Sidi `06 have considered case of all $a_i \equiv 1$

Traditional approach to multi-hop clock synchronization

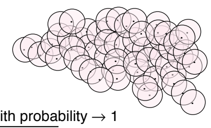


Assume all skews are 1 for simplicity

- Construct a rooted tree
  - Add up edge offsets to get clock-offset at a node
- How accurate is this?
- Std. Dev of Error $= \Theta\left(\sqrt{Diameter}\right)$

Performance analysis of traditional tree-based method: Random network

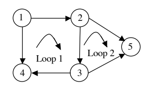
- Random multi-hop network with n nodes



- **Theorem (Gupta & K 1998)**
  - Geometric random graph is connected with probability $\to 1$ if and only if range r(n) satisfies $r(n) = \sqrt{\frac{\log n + \Gamma_n}{\pi n}}$ with $\Gamma_n \to \infty$
- All nodes choose a common range large enough for network connectivity
- Diameter of random graph at critical connectivity range $= O\left(\sqrt{\frac{n}{\log n}}\right)$
- Std. Dev of Error $= O\left(\left(\frac{n}{\log n}\right)^{1/4}\right)$
- Error grows *polynomially* as the number of nodes increases

Can we do better? Spatial smoothing (Solis, Borkar and K '05)

- How to improve the error?
  - Use constraints satisfied by *time*
- Sum of offsets along any loop is zero



$$\sum_{e \in Directed\ Cycle} x_e = 0$$

- Let A = incidence matrix of graph

$A =$

	(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

- Then $x = A^T v$
- Formulate as minimization problem: $\underset{\hat{v}}{\text{Min}} \left\| \hat{x} - A^T \hat{v} \right\|^2$
- Similarly for skew: $\sum_{e \in Directed\ Cycle} \log \hat{\alpha}_e = 0$

A fully distributed asynchronous multi-hop time synchronization protocol

- How to construct a distributed asynchronous algorithm which solves this optimization problem?
- Use coordinate descent

$$\frac{\partial}{\partial \hat{v}_j} \left\| \hat{x} - A^T \hat{v} \right\|^2 = 0$$

- Gives

$$\hat{v}_{j,\text{new}} = \frac{1}{|N_j|} \sum_{\text{edges } (i,j)} \left(\hat{v}_{i,\text{old}} + \hat{x}_{ij} \right)$$

- Fully asynchronous, distributed

$$\frac{\partial}{\partial \hat{v}_j} \left\| \hat{v} - \left(AA^T\right)^{-1} A\hat{x} \right\|^2 = 0$$

(Squares condition number)

$$\begin{cases} \hat{v}_{j,\text{new}} = \frac{1}{|N_j|+1} \cdot \frac{1}{|N_i|} \sum_{\text{edges } (i,j)} \left(e_j - e_i \right) \\ e_j \triangleq |N_j| \hat{v}_{j,\text{old}} - \sum_{k \in N_j} \left(\hat{v}_{k,\text{old}} + \hat{x}_{jk} \right) \end{cases}$$

Asymptotic error and Convergence rate (Giridhar & K `06)

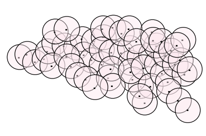
- **Theorem**
  - Asymptotic error variance is *Resistance Distance* of network
    - » Replace each link by a 1 Ω resistor
  - Time $T(n,\varepsilon,v(0))$ to converge to an ε-neighborhood of the optimal solution for a κ-connected network is:

$$\left(\log \frac{1}{\varepsilon \|v(0)\|} \right) \frac{\sum D_i}{D_0} \le T(n,\varepsilon,v(0)) \le \left(\log \frac{1}{\varepsilon \|v(0)\|} \right) \frac{\left(\sum D_i\right)^2}{\kappa^2}$$

- Prior work of Karp, Elson, Estrin, Shenker (2003)
  - Consider RBS, MVUE, and shows the connection to resistance distance
  - Ignores problem of delay
    - » So differences in receipt times of same broadcast gives estimates of offset

Performance analysis of spatial smoothing method: Random network

- Random multi-hop network with n nodes
  - All nodes choose a common range large enough for network connectivity



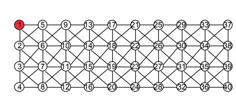
- **Theorem (Giridhar & K '05)**

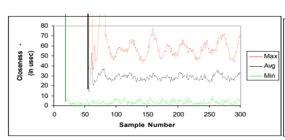
$$R_{\max} = \Theta(1) \text{ w.h.p.}$$

- Std. Dev of Error = $\Theta(1)$ w.h.p
- Synchronization error can be kept bounded in large wireless networks
  - Lends support for the feasibility of time-based computing in large distributed wireless networks

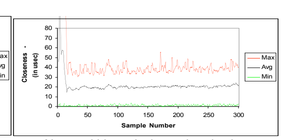
Experimental results

- Implementation on large sensor networks (Solis, Borkar and K '05)



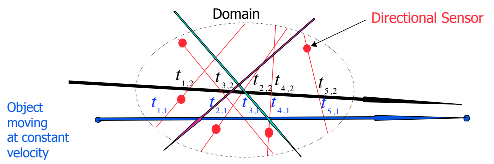


Leading algorithm (FTSP, Maroti et al '04)



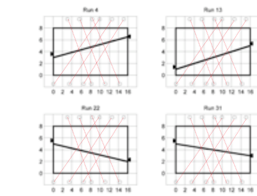
New multi-hop clock synchronization algorithm

Application: Object tracking by directional sensors (Plarre & K '05)



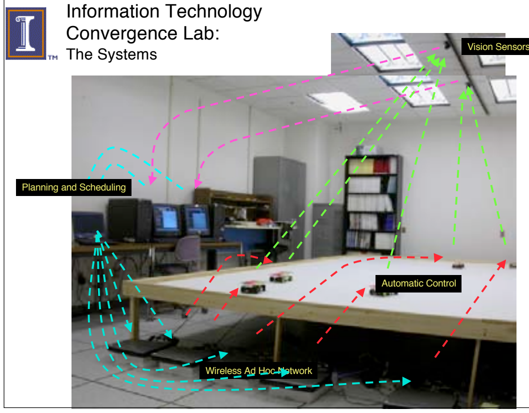
- Sensor locations and directions are *unknown*
- Object locations, tracks, and speeds are *unknown*
- Only times of crossings are *known*
- Goal: Estimate *trajectories of all objects* as well as *all sensor lines*
  - Optimization problem is highly nonconvex
- Key idea: Use an *adaptive basis* tuned to the motions of the first two objects

Implementation with laser pointers and Motes and a Lego car

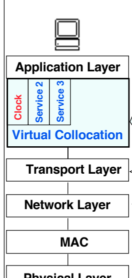


Clock synchronization in networked control

Information Technology Convergence Lab: The Systems

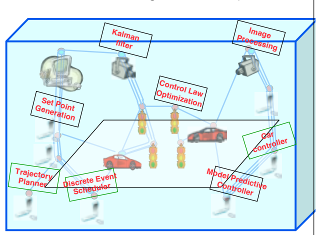


The Abstraction Layers



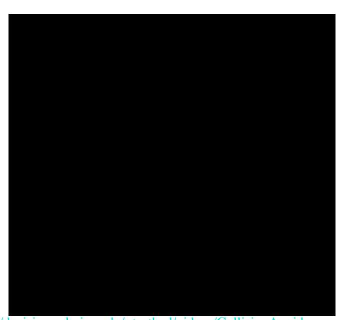
Etherware
- Location independence
- Semantic addressing of components
- System startup and upgrade during execution
- Automatic migration of components for performance

- Middleware manages the *Components*



- Minimal reconfiguration and reprogramming to build sensor-actuator systems

Collision avoidance (Schuetz, Robinson & K '05)



http://decision.csl.uiuc.edu/~testbed/videos/CollisionAvoidance.mpg

Thank you