Fundamental issues in networked control systems – 2:
Latencies and Time

P. R. Kumar

Dept. of Electrical and Computer Engineering,
and Coordinated Science Lab
University of Illinois, Urbana-Champaign
Email: prkumar@illinois.edu
Web: http://decision.csl.illinois.edu/~prkumar
Latencies: A Theory of QoS for Wireless
QoS for Wireless

- Increasing use of wireless networks for serving traffic with QoS constraints:
  - VoIP
  - Real-time Monitoring
  - Interactive Video
  - Networked Control

- “Best-effort” services are not adequate for QoS support
  - Not really “best”
Challenges

- How to formulate a mathematical framework for QoS?

- Relevant
  - Jointly deal with three QoS criteria/constraints:
    - Deadlines
    - Delivery ratios
    - Channel unreliabilities

- Tractable
  - Solutions needed for providing QoS support:
    - Admission control policies for flows
    - Packet Scheduling policies

(Hou, Borkar & K ’09)
Client-Server Model

- A wireless system with an Access Point serving \( N \) clients

- Time is slotted

- One slot = One packet

- AP indicates which client should transmit in each time slot
Model of Unreliable Channel

- **Unreliable channels**

- **One packet transmission in each slot**
  - Successful with probability $p_n$
  - Fails with probability $1-p_n$
  - $\text{Prob}(\text{Packet transmission on Channel } n \text{ is successful}) = p_n$
  - So packet delivery time is a geometrically distributed random variable $\gamma_n$ with mean $1/p_n$

- **Non-homogeneous link qualities**
  - $p_1, p_2, \ldots, p_N$ can be different
QoS Model

- Clients generate packets with fixed period $\tau$
- Packets expire and are dropped if not delivered in the period
- Delay of successfully delivered packet is therefore at most $\tau$
- Delivery ratio of client $n$ should be at least $q_n$

$$\liminf_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} 1(\text{Packet delivered to Client } n \text{ in } k\text{-th period}) \geq q_n \quad a.s.$$
Multiple-Time Scale QoS requirements

- Unreliable channels
  - Short time scale: Slots

- Arrivals and Deadlines
  - Medium time scale:
    - Period $\tau$ arrivals
    - Relative Deadline $\tau$

- Delivery ratio requirements
  - Long time scale:

$$\liminf_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} 1\text{(Packet of client } n \text{ delivered in } k\text{-th period)} \geq q_n \text{ a.s.}$$
Protocol for Operation

- AP indicates which client should transmit in each time slot

- Uplink
  - Poll (ex. CF-POLL in 802.11 PCF)
Protocol for Operation

- AP indicates which client should transmit in each time slot

- Uplink
  - Poll (ex. CF-POLL in 802.11 PCF)
  - Data

\[ p_n = \text{Prob( both Poll/Data are delivered)} \]

- Downlink
  - Data
  - ACK

\[ p_n = \text{Prob( both Data/ACK are delivered)} \]
Protocol for Operation

- AP indicates which client should transmit in each time slot

- Uplink
  - Poll (ex. CF-POLL in 802.11 PCF)
  - Data
  - No need for ACK
  - $p_n = \text{Prob( both Poll/Data are delivered)}$
Protocol for Operation

- AP indicates which client should transmit in each time slot

- Uplink
  - Poll (ex. CF-POLL in 802.11 PCF)
  - Data
  - No need for ACK
  - $p_n = \text{Prob( both Poll/Data are delivered)}$

- Downlink
  - Data
Protocol for Operation

- AP indicates which client should transmit in each time slot

- **Uplink**
  - Poll (ex. CF-POLL in 802.11 PCF)
  - Data
  - No need for ACK
  - $p_n = \text{Prob( both Poll/Data are delivered)}$

- **Downlink**
  - Data
  - ACK
  - $p_n = \text{Prob( both Data/ACK are delivered)}$
Implied Work Load

- The proportion of time slots needed for client $n$ is

$$\omega_n = \frac{1}{p_n} \times \frac{q_n}{\tau}$$

expected number of time slots needed for a successful transmission
Implied Work Load

- The proportion of time slots needed for client $n$ is

$$ w_n = \frac{1}{p_n} \times \frac{q_n}{\tau} $$

number of required successful transmissions in a period
Implied Work Load

- The proportion of time slots needed for client $n$ is

$$\omega_n = \frac{1}{p_n} \times \frac{q_n}{\tau}$$

normalize by period length
The proportion of time slots needed for client $n$ is

$$w_n = \frac{1}{p_n} \times \frac{q_n}{\tau}$$

So $w_n$ is the “implied work load”
Problem Formulation

- Admission control
  - Decide whether a set of clients is feasible

- Scheduling policy
  - Design a policy that fulfills every set of clients that is feasible
Necessary Condition for Feasibility of QoS Requirements

- Necessary condition from classical queuing theory:
  \[ \sum_{n=1}^{N} w_n \leq 1 \]

- But not sufficient

- Reason: Additional idleness is caused by packet drops due to deadline expiry
  - Only one packet and no queueing
Stronger Necessary Condition

- Let $I = \text{Expected proportion of idle time}$
  
  $$I := \frac{1}{\tau} E \left[ \left( \tau - \sum_{n=1}^{N} \gamma_n \right)^+ \right] \text{ where } \gamma_n \sim \text{Geom}(p_n)$$

- Stronger necessary condition
  
  $$\sum_{n=1}^{N} w_n + I \leq 1$$

- Sufficient?

- Still not sufficient!
Even Stronger Necessary Condition

- If the set of clients \( \{1, 2, \ldots, N\} \) is feasible, then any \textit{smaller} set of clients \( S \subseteq \{1, 2, \ldots, N\} \) should also be feasible.

- Let \( I_S = \frac{1}{\tau} E \left[ \left( \tau - \sum_{n \in S} \gamma_n \right)^+ \right] \) = Expected proportion of idle time for \( S \).

- Stronger necessary condition: \( \sum_{n \in S} w_n + I_S \leq 1, \forall S \subseteq \{1, 2, \ldots, N\} \).

- Not enough to just consider \( \{1, 2, \ldots, N\} \).

- \textbf{Theorem (Hou, Borkar & K ’09)}
  Condition is necessary and sufficient for a set of clients to be fulfillable.
Counterexample

- Two clients
  - Period $\tau = 3$
  - $p_1 = p_2 = 0.5$
  - $q_1 = 0.876$, $q_2 = 0$
  - $w_1 = 1.752/3$, $w_2 = 0$
  - $I_{\{1\}} = I_{\{2\}} = 1.25/3$
  - $I_{\{1,2\}} = 0.25/3$

- Now $w_1 + w_2 + I_{\{1,2\}} = 2.002/3 < 1$

- However, $w_1 + I_{\{1\}} = 3.002/3 > 1$
Delivery Debt Scheduling Policies

- Compute “debt” owed to each client at beginning of period
- A client with higher debt gets a higher priority on that period
Two Definitions of Debt

- **The time debt of client** $n$
  - $w_n$ – actual proportion of transmission time given to client $n$

- **The weighted delivery debt of client** $n$
  - Weighted delivery debt $= (q_n - \text{actual delivery ratio})/p_n$

- **Theorem**: (Hou, Borkar & K -09):
  Both largest debt first policies fulfill every set of clients that can be fulfilled
Computationally Tractable Policy for Admission Control

- Admission control consists of determining feasibility

- Apparently, we need to check:

\[ \sum_{n \in S} w_n + I_S \leq 1, \quad \forall S \subseteq \{1, 2, \ldots, N\} \]

- \(2^N\) tests, so computationally complex
- However:

- **Theorem (Hou, Borkar & K ’09)**
  - Can order the clients according to \(q_n\) in decreasing order
  - Can be reduced to \(N\) tests
  - Polynomial time \(O(N \tau \log \tau)\) algorithm for admission control
Simulation testing on ns-2

- Implement on IEEE 802.11 Point Coordination Function (PCF)
  - PCF: a Point Coordinator (PC) assigns transmission opportunities to clients
  - Packets should be sent by broadcasting to avoid ACKs
  - Compatible with Distributed Coordination Function (DCF)

- Application: VoIP standard

<table>
<thead>
<tr>
<th>64 kbps data rate</th>
<th>20 ms period</th>
</tr>
</thead>
<tbody>
<tr>
<td>160 Byte packet</td>
<td>11 Mb/s transmission rate</td>
</tr>
<tr>
<td>610 µs time slot</td>
<td>32 time slots in a period</td>
</tr>
</tbody>
</table>
Evaluated Four Policies

- DCF
- PCF with randomly assigned priorities
- Time-debt first policy
- Weighted-delivery debt first policy
Traffic requirements: Test at Edge of Feasibility

- Two groups of clients:
  - Group A requires 99% delivery ratio
  - Group B requires 80% delivery ratio
  - The $n^{th}$ client in each group has $(60+n)\%$ channel reliability

- Feasible set: 11 group A clients and 12 group B clients
- Infeasible set: 12 group A clients and 12 group B clients

- Evaluation Measure
  - $DMR(n) := (q_n - \text{percentage of actual delivered packets})^+$

- $DMR$ of system $= \sum_{n=1}^{N} DMR(n)$
Results for a Feasible Set

(Hou, Borkar & K '09)
Results for an Infeasible Set

(Hou, Borkar & K ’09)
Summary

- Formulate a framework for QoS that deals with deadlines, delivery ratios, and channel unreliabilities
- Characterize when QoS is feasible
- Provide efficient scheduling policy
- Provide efficient admission control policy
- Address implementation issues
Time:

How accurately can we synchronize clocks over wireless networks?

And how do we synchronize them?
Need for clock synchronization in distributed systems

- Knowledge of time is important in Networks
  - Communication network protocols
  - Sensor network applications
  - Networked control

- However no two clocks agree

- How to synchronize clocks in distributed systems?
Impossibility of synchronizing two clocks
Impossibility of synchronizing two clocks

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

- Delays $d_{12}$ and $d_{21}$ in two directions

- Clock 1 is the Reference Clock: $a_1 = 1, b_1 = 0$

**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

\[ r_1 = a_2(s_1 + d_{12}) + b_2 \]

\[ s_2 = a_2(r_2 - d_{21}) + b_2 \]

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

Rank 3: Cannot estimate 4 parameters
What is determinable and what is not

**Theorem (Graham & K '04)**

i. The skew $a_2$ can be estimated correctly.

ii. The round-trip delay $(d_{1j} + d_{ji})$ can be estimated precisely.

iii. The sender can predict the receiver's time at which receiver receives a packet.

iv. The offset is unknown. It represents one undeterminable degree of freedom.

v. Delays are affine functions of the unknown offset.

vi. By invoking causality, we can determine an interval in which the offset lies

Proof

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

\[
a_2^* := \frac{r_{1,2}^{(k)} - r_{1,2}^{(l)}}{s_1^{(k)} - s_1^{(l)}}
\]

\[
d_{12}^* := \frac{r_{1,2}^{(k)} - a_2^* s_1^{(k)}}{a_2^*} \\
\]

\[
d_{21}^* := \frac{a_2^* r_{2,1}^{(l)} - s_2^{(l)}}{a_2^*}
\]

\[
\hat{d}_{12} \geq 0 \text{ and } \hat{d}_{21} \geq 0 \Rightarrow b_2 \in [-a_2^* d_{21}^*, a_2^* d_{12}^*]
\]

\[
r_{1,2}^{(k)} = a_2 s_1^{(k)} + a_2 d_{12} + b_2 = a_2^* s_1^{(k)} + a_2^* d_{12}^*
\]
Estimating skew, latency and offset for symmetric delays

Solution: Assume \( d_1 = d_2 \), i.e., delays in both directions are symmetric

Skew \( \alpha = \frac{t_3^A - t_1^A}{t_3^B - t_1^B} \)

Latency \( d_A = \frac{1}{2} \left[ (t_2^A - t_1^A) - \alpha (t_2^B - t_1^B) \right] \)

Offset \( = (t_1^B - t_1^A) - d_A \)

Use RLS for parameter updating

Issues
- Ill conditioning
- Reparametrization
  - Instead of \((\alpha, \text{offset at time } 0)\)
    use \((\alpha, \text{offset now})\)
- Combination of windowing and exponential forgetting
Limitations for network clock synchronization
Model of problem

- A network of \( n \) clocks
  - Links can be unidirectional or bidirectional

- Linear clocks: \( \tau_i(t) = a_i t + b_i \)
- Clock 1 is the reference clock: \( \tau_1(t) \equiv t \)
- Delay \( d_{ij} \) when clock \( i \) sends a packet to clock \( j \)
Special case: Star graph

- There are \((n-1)\) independent links
- So \((n-1)\) degrees of freedom
- The offset of each node is an unknown parameter

- Uncertainty set of all unknown parameters = Translate of an \((n-1)\)-dimensional subspace

(Freris and Kumar 2007).
Estimation of skew

**Theorem** (Freris and Kumar 2007)

i. Every node can estimate *its own skew*

There is a directed spanning tree rooted at the reference clock

ii. Every node can estimate the skew of all nodes

There is a directed path from every node to every other node

\[
a_2 = \frac{r_{1,2}^{(k)} - r_{1,2}^{(l)}}{s_1^{(k)} - s_1^{(l)}}
\]

\[
a_3 = \frac{r_{2,3}^{(j)} - r_{2,3}^{(m)}}{s_2^{(j)} - s_2^{(m)}}
\]
Estimates of offsets and delays

**Theorem** (Freris and Kumar 2007)

Let edge set of network be $E$, and suppose network is bi-connected

i. The offset vector $\hat{b} = (\hat{b}_i : 2 \leq i \leq n)$ is free to choose. So there are $(n-1)$ degrees of freedom.

ii. Uncertainty set for delay vector $\hat{d} = (\hat{d}_{ij} : (i,j) \in E)$ is image under affine map of $\mathbb{R}^{n-1}$

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

iii. Under causality, the uncertainty set for offset vector is a compact polyhedron:

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

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

- More detailed model for link delay

\[ d_{ij} = \alpha_i + \tau_{ij} + \beta_j \]

- Unknown transmitter specific transmit delay
- Known distance dependent propagation delay
- Unknown receiver specific reception delay

- Does not help

(Freris and Kumar 2007).
Analyzing accuracy clock synchronization over multi-hop networks in stochastic case
Estimating skew, latency and offset for symmetric delays

- Solution: Assume $d_1 = d_2$, i.e., delays in both directions are symmetric

\[
\text{Skew } \alpha = \frac{t_3^A - t_1^A}{t_3^B - t_1^B} \\
\text{Latency } d_A = \frac{1}{2} \left[ (t_2^A - t_1^A) - \alpha(t_2^B - t_1^B) \right] \\
\text{Offset } = (t_1^B - t_1^A) - d_A
\]

- Use RLS for parameter updating
- Issues
  - Ill conditioning
  - Reparametrization
    - Instead of $(\alpha, \text{offset at time 0})$ use $(\alpha, \text{offset now})$
    - Combination of windowing and exponential forgetting

(Solis, Borkar & K 2006).
Traditional approach to multi-hop clock synchronization

- Construct a rooted tree
  - Add up edge offsets to get clock-offset at a node
- How accurate is this?

\[ \hat{v}_3 = \hat{x}_{01} + \hat{x}_{12} + \hat{x}_{23} \]

\[ \text{Std. Dev of Error} = \Theta(\sqrt{\text{Diameter}}) \]
Performance analysis of traditional tree-based method: Collocated network

- Collocated network with $n$ nodes

- Diameter = 1 hop

- Std. Dev of Error = $\Theta(\sqrt{Diameter})$

- Error does not go to zero even as number of nodes increases

(Giridhar & K 2006)
Performance analysis of traditional tree-based method: Lattice

- Regular Lattice with $n$ nodes

\[
\text{Diameter} = \Theta(\sqrt{n})
\]

- Std. Dev of Error $= \Theta\left(n^{1/4}\right)$

- Error grows polynomially as the number of nodes increases

(Giridhar & K 2006)
Performance analysis of traditional tree-based method: Random network

- Random multi-hop network with \( n \) nodes
  - Geometric random graph is connected with probability \( \to 1 \)
    
    \[
    r(n) = \sqrt{\frac{\log n + \Gamma_n}{\pi n}} \text{ with } \Gamma_n \to \infty
    \]
  - 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 \( = O\left(\left(\frac{n}{\log n}\right)^{1/4}\right)\)
  
  (Giridhar & K 2006)
- Again error grows polynomially as the number of nodes increases
Can we do better?

Spatial smoothing to improve accuracy
Constraint satisfied by time

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

So \( \sum_{e \in \text{Directed Cycle}} x_e = 0 \)
Matrix formulation: Kirchoff’s voltage law

- Let $A = \text{incidence matrix of graph}$

<table>
<thead>
<tr>
<th></th>
<th>(1,2)</th>
<th>(2,3)</th>
<th>(3,4)</th>
<th>(2,5)</th>
<th>(3,5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

- Let $x = \text{edge offsets}$, and $\nu = \text{node offsets from the root node}$

- Then $x = A^T \nu$

- Analogy with KVL: $x_e = \text{edge potential difference}$, and $\nu = \text{node potential}$
Spatial smoothing (Solis, Borkar & K ‘05)

- Let $\hat{x}_{ij} = \text{estimate of } x_{ij}$
  - Obtained by exchanging packets between neighbors

- Since “true” offsets satisfy $x = A^T \nu$

- Formulate estimation of $\{\hat{\nu}_i\}$ as minimization problem: $\min_{\hat{\nu}} \|\hat{x} - A^T \hat{\nu}\|^2$

- Goal: To improve the estimate of time at node $i$, we make use of all packet exchanges between all remote nodes $j$ and $k$

- For skew: $\sum_{e \in \text{Directed Cycle}} \log \hat{\alpha}_e = 0$
How to do this in a distributed asynchronous manner?
Distributed asynchronous multi-hop time synchronization protocol

- How to construct a distributed asynchronous algorithm which solves this optimization problem \( \min_{\hat{v}} \| \hat{x} - A^T \hat{v} \|^2 \)?

- Use coordinate descent \( \frac{\partial}{\partial \hat{v}_j} \| \hat{x} - A^T \hat{v} \|^2 = 0 \)

- Gives \( \hat{v}_{j,\text{new}} = \frac{1}{|N_j|} \sum_{\text{edges } (i,j)} \left( \hat{v}_{i,\text{old}} + \hat{x}_{ij} \right) \) (Solis, Borkar & K 2006)

- Distributed Asynchronous algorithm
  - Select a non-root node \( j \)
  - Average current estimates of node offsets of all the neighbors of \( j \)
  - Update the node estimate of node \( j \)

- Algorithm does not require any knowledge of topology of network
  - Can even be employed for mobile networks
  - Can also be used to change reference nodes midstream
    - Root node is the one which simply broadcasts zero offset all the time
What kind of accuracy can spatial smoothing algorithm achieve?
Asymptotic error covariance of spatial smoothing

- The true edge offsets satisfy \( x = A^T v \)

- Suppose the estimates are noisy: \( \hat{x}_e = x_e + w_e \) where \( w_e \sim N(0,1) \), iid.

- Hence \( \hat{x} = x + w = A^T v + w \), where \( w \sim N(0,I) \)

- The solution of the minimization problem \( \min_\hat{v} \| \hat{x} - A^T \hat{v} \|^2 \) is:
  \[
  \hat{v} = \left( AA^T \right)^{-1} A \hat{x} \\
  = \left( AA^T \right)^{-1} A (A^T v + w) \\
  = v + \left( AA^T \right)^{-1} A w
  \]

- So synchronization error is \( \tilde{v} = \left( AA^T \right)^{-1} A w \)

- So covariance of synchronization error is \( \Sigma \triangleq \text{Cov}(\tilde{v}) = \left( AA^T \right)^{-1} \)
Characterization of error covariance

- What are the diagonal terms of \((AA^T)^{-1}\)?
  - \(A\) is the reduced incidence matrix
  - \(AA^T\) is the principal submatrix of Laplacian

- **Theorem: Circuit theory (See Chua, Desoer & Kuh book)**
  - Consider electrical network with edges replaced by unit resistors
  - \((AA^T)^{-1}_{ii}\) is the resistance between node \(i\) and the root node

- Prior work of Karp, Elson, Estrin, Shenker (2003)
  - Consider Reference Broadcast Scheme where different nodes receive same broadcast
  - Ignores problem of delay
  - So differences in receipt times of same broadcast gives estimates of offset
  - Considers the minimum variance unbiased estimate of node offsets from such receipt times and shows the connection to resistance distance
  - Nodes estimate broadcast time as well as offset
Performance analysis of spatial smoothing method: Collocated network

- Collocated network with $n$ nodes

Complete graph on $n$ vertices:

$$R_{\text{max}} = \frac{2}{n}$$

- Std. Dev of Error $= \Theta\left(\frac{1}{\sqrt{n}}\right)$

- Error goes to zero as number of nodes increases

(Giridhar & K 2006)
Performance analysis of spatial smoothing method: Lattice

- Regular Lattice with $n$ nodes

- Std. Dev of Error $= \Theta\left(\sqrt{\log n}\right)$

- Error grows logarithmically as the number of nodes increases

Based on formula derived by F.Y. Wu (2002):

$$R_{\text{max}} = \Theta\left(\log n\right)$$

(Giridhar & K 2006)
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 ‘06)**
  $$R_{\text{max}} = \Theta(1) \text{ w.h.p.}$$

  Proof: Involves constructing $\log n$ grid graphs in parallel

- Std. Dev of Error = $O(1)$ 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
How fast does convergence take place?
Convergence of Algorithms

- **THEOREM (Giridhar & K ‘06)**

  - On a connected graph, both synchronous and asynchronous algorithms converge to least-squares optimal solution.

  - The convergence time of the synchronous algorithm to within error of \( \varepsilon \) can be bounded as

\[
\frac{\sum d_i}{d_0} \log \left( \frac{1}{\varepsilon \| v^{(0)} \|} \right) < T_\varepsilon (v^{(0)}) < \frac{\left( \sum d_i \right)^2}{\kappa^2} \log \left( \frac{1}{\varepsilon \| v^{(0)} \|} \right)
\]

  \( \kappa \): Edge connectivity of graph
Convergence times for some wireless networks

- Lattice graph: $O(n^2)$
  - Numerical experiments “indicate” $n \log^2 n$

- Random planar graph: $O(n^2)$

- Translates to $O(n^3)$ messages sent over network

(Giridhar & K 2006)
Experimental results

- Solis, Borkar and K ‘05: Implementation on large sensor networks

![Graph showing experimental results]

- Leading algorithm (FTSP, Maroti et al ‘04)
- New multi-hop clock synchronization algorithm
An application of clock synchronization to sensor networks:

Object tracking by scattered directional sensors
(Plarre & K ‘05)
Tracking Problem

- 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

Plarre & K ’05)
Implementation with laser pointers and Motes and a Lego car

Object Tracking
with
Directional Sensors

Plarre & K ‘05)
References-1


References-2


References-3

References-4


- Jerry Chiang, Jason Haas, Yih-Chun Hu, P. R. Kumar and Jihyuk Choi, “Fundamental Limits on Secure Clock Synchronization and Man-In-The-Middle Detection in Fixed Wireless Networks.” *Proceedings of Infocom 2009*, April 19-25, 2009, Rio de Janeiro, Brazil.


Thank you
Attribution-Noncommercial-No Derivative Works 3.0 Unported

You are free:

- to Share — to copy, distribute and transmit the work.

Under the following conditions:

- Attribution — You must attribute this work to P. R. Kumar (with link).

  ![Attribution](http://creativecommons.org/nl/3.0/acl/about/)

  [http://decision.csl.illinois.edu/~prkumar/talks.html](http://decision.csl.illinois.edu/~prkumar/talks.html)

- Noncommercial — You may not use this work for commercial purposes.

- No Derivative Works — You may not alter, transform, or build upon this work.

With the understanding that:

- Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.

- Other Rights — In no way are any of the following rights affected by the license:
  - Your fair dealing or fair use rights;
  - The author's moral rights;
  - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.