Distributed consensus protocols for clock synchronization in sensor networks

Luca Schenato

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Workshop on cooperative multi agent systems

Pisa, 6/12/2007

Outline

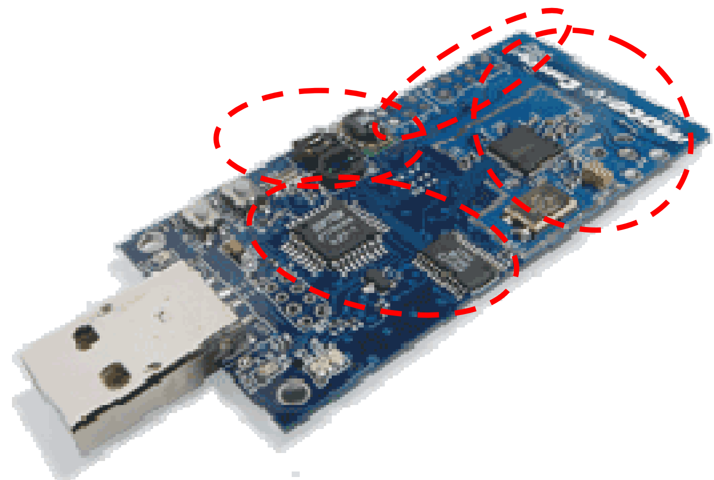
- Motivations
- Intro to consensus algorithms
- Average Time Synch Protocol
  - Algorithm
  - Experimental results
- Conclusions

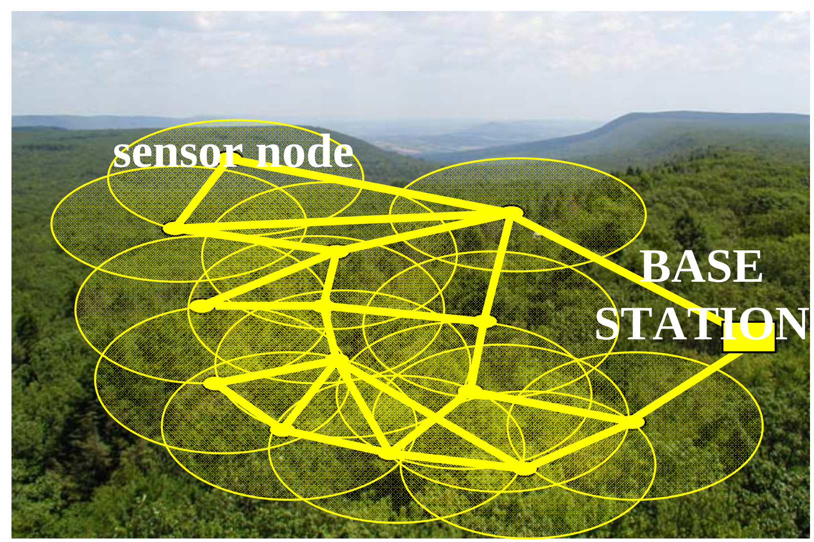
Outline

- Motivations
- Intro to consensus algorithms
- Average Time Synch Protocol
  - Algorithm
  - Experimental results
- Conclusions

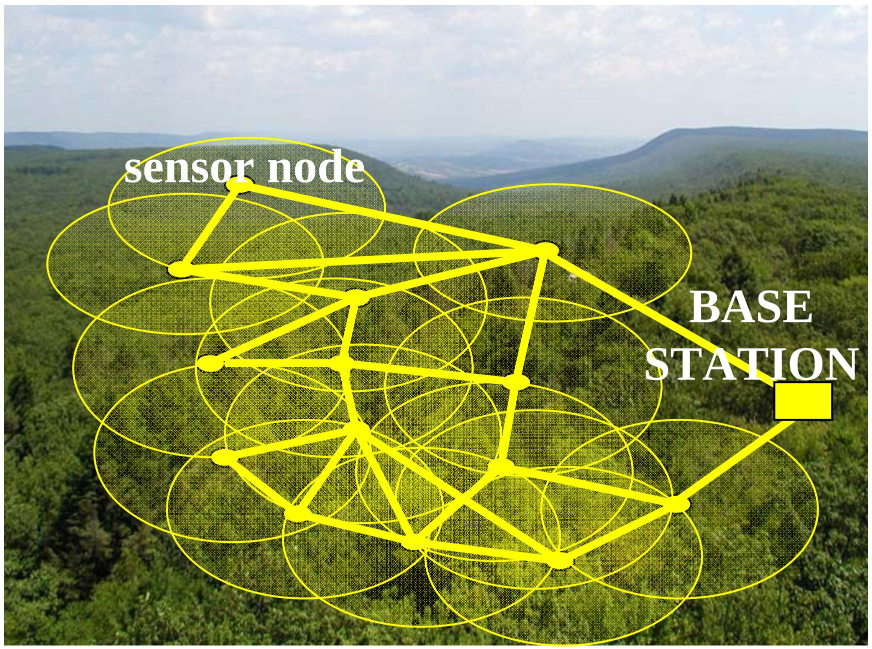
Wireless Sensor Actuator Networks (WSANs)

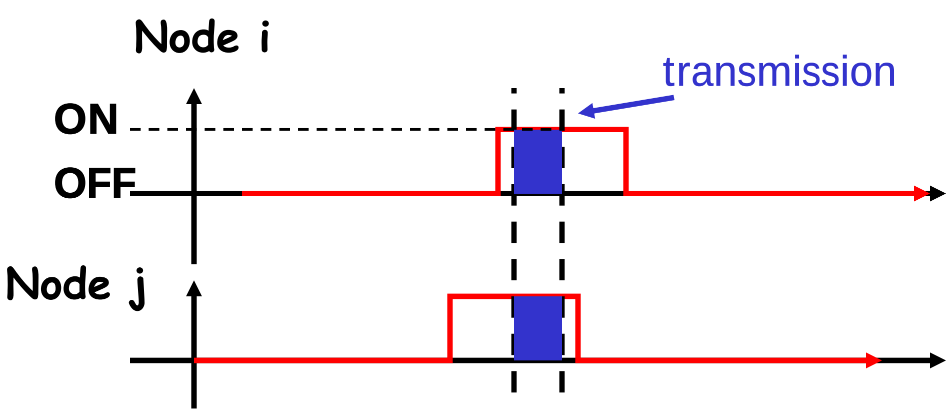
- Small devices
  - μController, Memory
  - Wireless radio
  - Sensors & Actuators
  - Batteries
- Inexpensive
- Multi-hop communication
- Programmable (micro-PC)





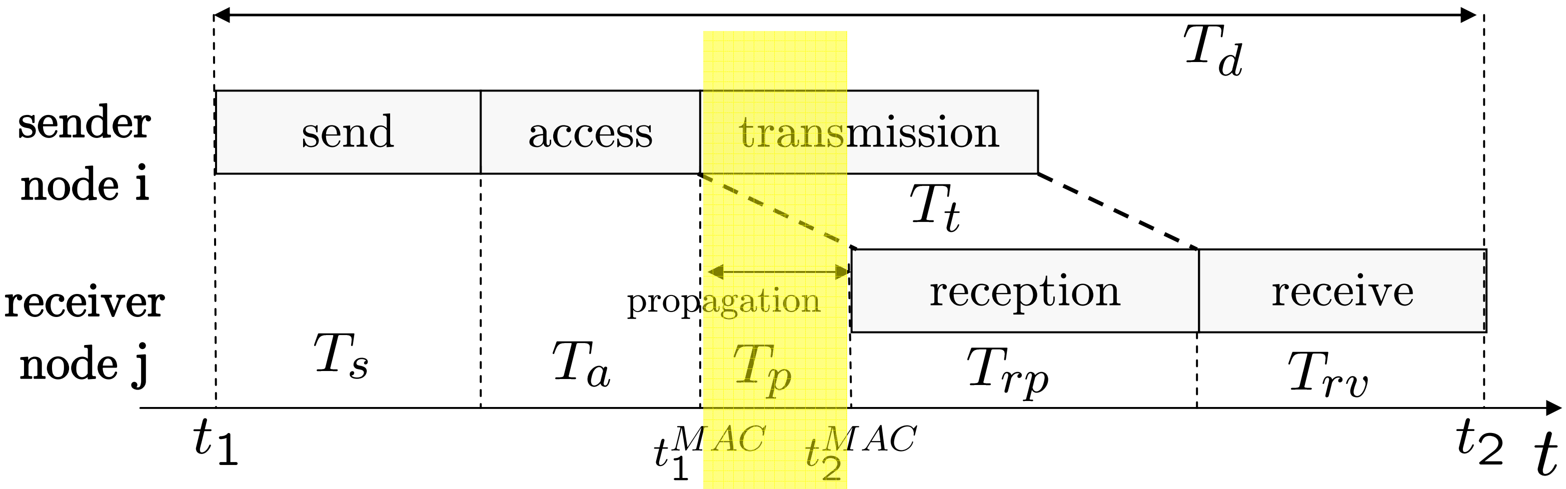
Time synchronization in sensor networks





- Why time-synch ?
  - Spatio-temporal correlation of events such as tracking
  - Communication scheduling TDMA to reduce interference
  - Power management
- Problems:
  - Every node has own clock
  - Different offsets
  - Different speeds (skew)
  - Random transmission delay

Measurement delivery delay



$T_s, T_{rv} \sim 100ms$ random, depends on OS

$T_s \sim 0.1 - 1s$, VERY random, depends on traffic and radio

$T_t = T_{rp} \sim 10 - 500ms$, deterministic, depends on packet size

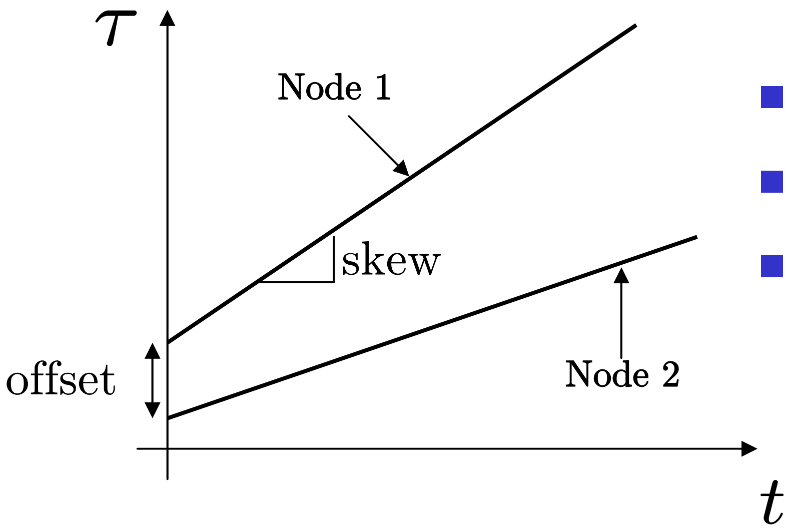
$T_t = T_{rp} \sim 100ns$, deterministic, depends on packet size

- **MAC layer time-stamping**
  - Read local clock t_1^{MAC} at node *i* when start sending first bit
  - Write t_1^{MAC} on leaving packet
  - Read and store local clock t_2^{MAC} at node *j* when start receiving first bit

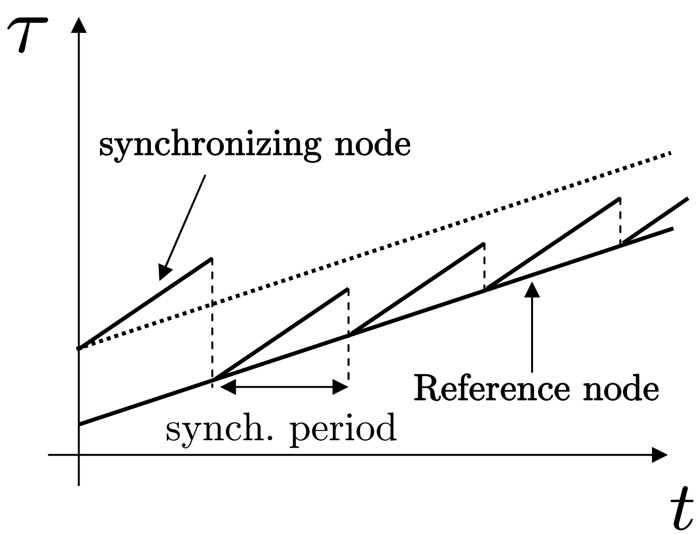
$\Rightarrow T_{delay} \sim 100ns$

t_1^{MAC}	data	header

Clock characteristics & standard clock pair sych



- Offset: instantaneous time difference
- Skew: clock speed
- Drift: derivative of clock speed

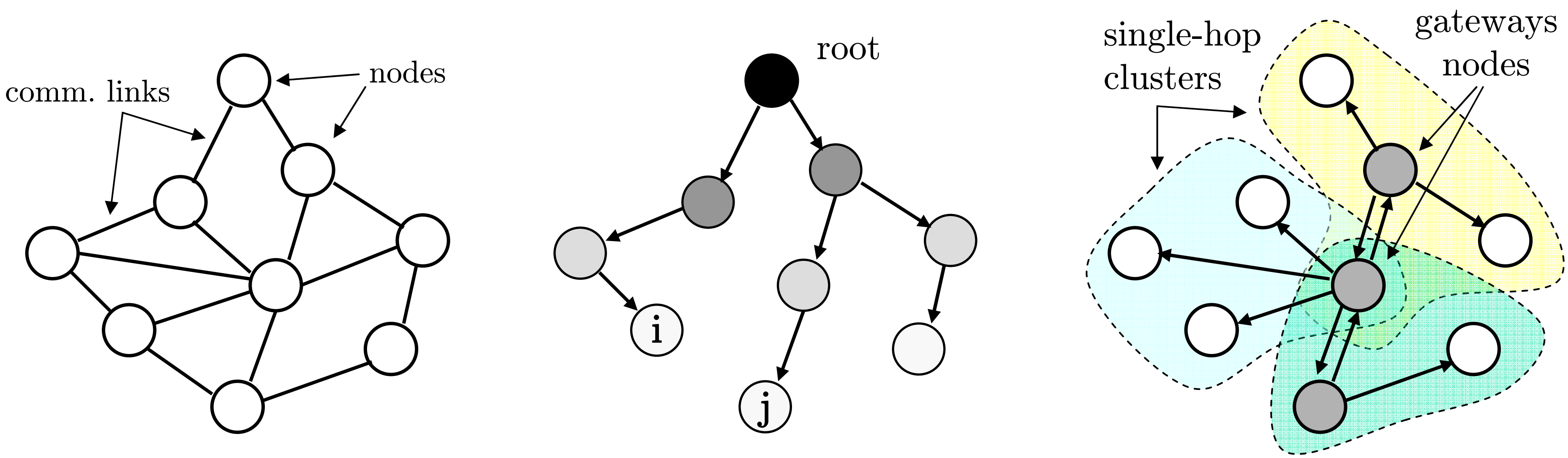


- Offset synch: periodically remove offset with respect to reference clock
- Skew compensation: estimate relative speed with respect to reference clock

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Synch topologies for sensor networks

Tree-based sync **Cluster-based sync**



- PROS
  - Straightforward extension of pair synch
- CONS
  - Links may disappear
  - Root or gateways might temporarily disappear or die
  - New nodes might appear
  - Can be made adaptive but high protocol overhead

Ideal protocol features

- Distributed:
  - each sensor runs the same code
- Asynchronous:
  - Non-uniform updating period
- Adaptive:
  - should handle dying nodes, appearing nodes, moving nodes
- Simple to implement
- Robust to packet loss
- Long synch periods

	distrib.	skew comp.	MAC timestamp
Time-synch Prot. for Sensor Networks	no	no	no
Lightweight Time Synch.	no	no	no
Flooding Time Synch Prot.	no	yes	yes
Reference Broadcast Synchronization	no	yes	yes
Reachback Firefly Algorithm	yes	no	yes
Distributed Time Synch Prot.	yes	yes	yes
Average Time Synch Prot.	yes	yes	yes

Outline

- Motivations
- Intro to consensus algorithms
- Average Time Synch Protocol
  - Algorithm
  - Experimental results
- Conclusions

The consensus problem

- Main idea
  - Having a set of agents to agree upon a certain value using only local information exchange (local interaction)
- Also known as:
  - Agreement algorithms (economics, signal processing)
  - Gossip algorithms (CS & communications)
  - Synchronization (statistical mechanics)
  - Rendezvous (robotics)
- Suitable for (noisy) sensor networks
  - Clock synchronization: all clocks gives the same time
  - Signal Processing/Estimation: mean temperature in a room
  - Target detection: do we agree there is target ?
  - Fault detection: is that sensor properly functioning ?
  - Attack detection: is that sensor being “tampered” ?

A robotics example: the rendezvous problem

GOAL: a set of N vehicles should converge to a common location using only local communication

Vehicle dynamics

$$x_i^+ = x_i + u_i, \quad x_i, u_i \in \mathbb{C}$$

$$x = [x_1 \; x_2 \ldots x_N]^T \in \mathbb{C}^N$$

$$x^+ = x + u$$

Control law

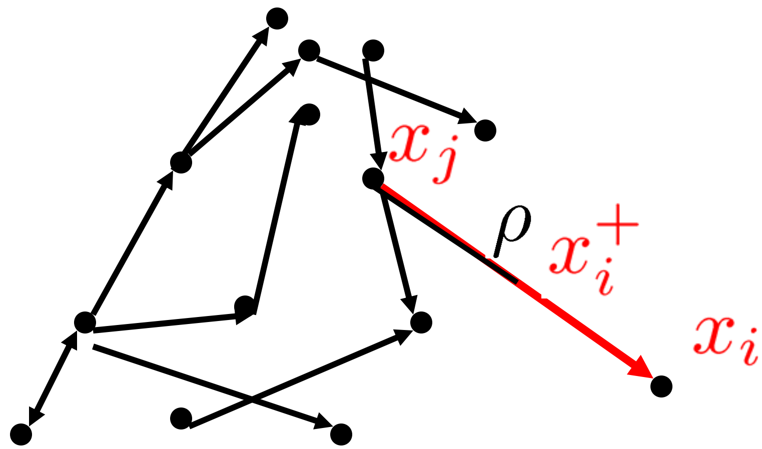
$$u_i^+ = \rho(x_j - x_i)$$

$$x_i^+ = (1-\rho)x_i + \rho x_j$$

Closed loop dynamics

$$x^+ = Px$$

$$\mathbf{1} = [1 \; 1 \ldots 1]^T$$



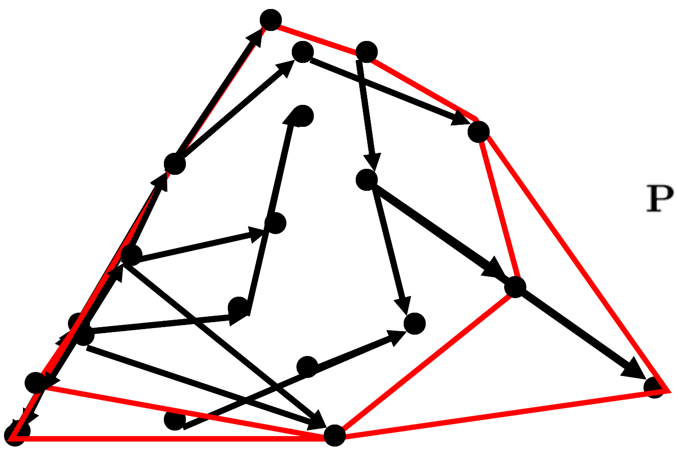
$$P = \begin{array}{cl} \begin{array}{cccccc} & & & j \downarrow & & \end{array} & \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\rho & 0 & \rho & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \rho & 1-\rho & 0 \\ 0 & 0 & 0 & \rho & 0 & 1-\rho \end{bmatrix} & \begin{array}{l} \\ \leftarrow i \\ \\ \\ \\ \\ \end{array} \end{array}$$

$$P_{ij} \geq 0$$

$$P\mathbf{1} = \mathbf{1}$$

(ROW) STOCHASTIC MATRIX

A robotics example: the rendezvous problem



transmitting agent i

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\rho & 0 & \rho & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \rho & 1-\rho & 0 \\ 0 & 0 & 0 & \rho & 0 & 1-\rho \end{bmatrix}$$

listening agents j's

$$\mathbf{1} = [1\ 1 \ldots 1]^T$$

$$P_{ij} \geq 0$$

$$P\mathbf{1} = \mathbf{1}$$

(ROW) STOCHASTIC MATRIX

$$x^+ = Px$$

Convex hull always shrinks.
If communication graph sufficiently connected, then shrinks to a point

$$x(t) = P^t x(0) \rightarrow \alpha \mathbf{1}$$

$$\alpha \in \mathrm{convHull}(x_1(0), \ldots, x_N(0))$$

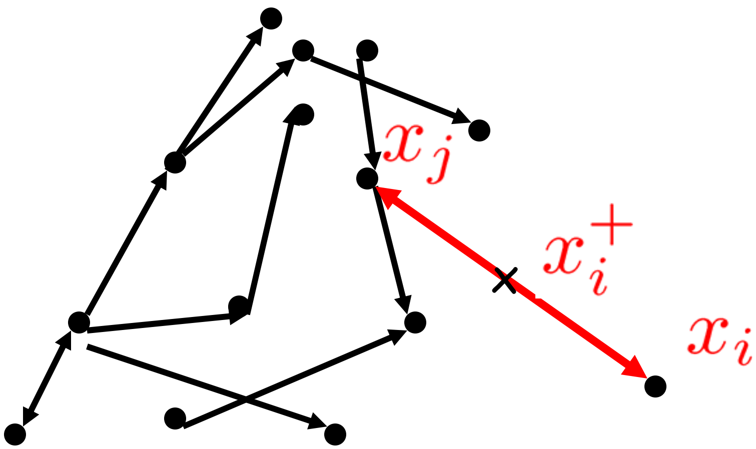
A robotics example: the rendezvous problem

$$x_j^+ = (1-\rho)x_j + \rho x_i$$

&

$$\boxed{x_i^+ = (1-\rho)x_i + \rho x_j}$$

$$\frac{x_j^+ + x_i^+}{2} = \frac{x_j + x_i}{2}$$



$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-\rho & 0 & 0 & \rho & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \rho & 0 & 0 & 1-\rho & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_{ij} \geq 0$$

$$P\mathbf{1} = \mathbf{1}$$

$$\mathbf{1}^T P = \mathbf{1}^T$$

DOUBLY STOCHASTIC MATRIX

$$x(t) = P^t x(0) \to \alpha \mathbf{1}$$

$$\alpha = \frac{1}{N}\sum_i x_i(0)$$

Convergence to mean of initial conditions, i.e. center of mass

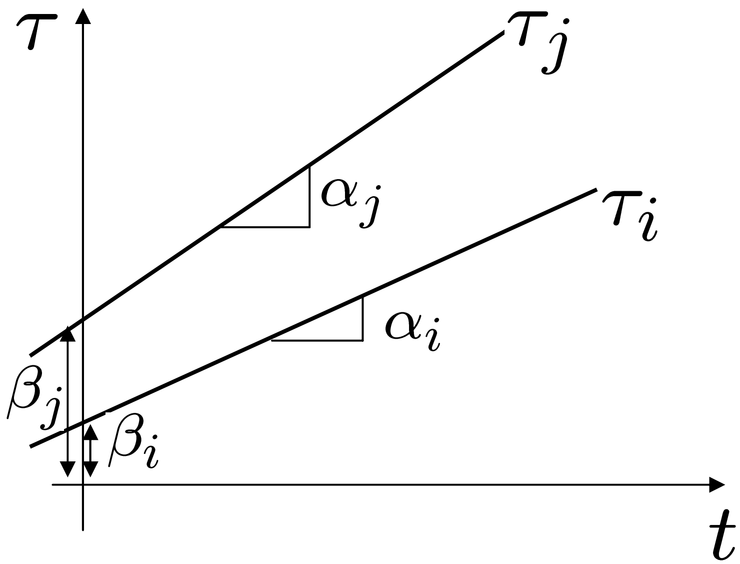
Convergence conditions

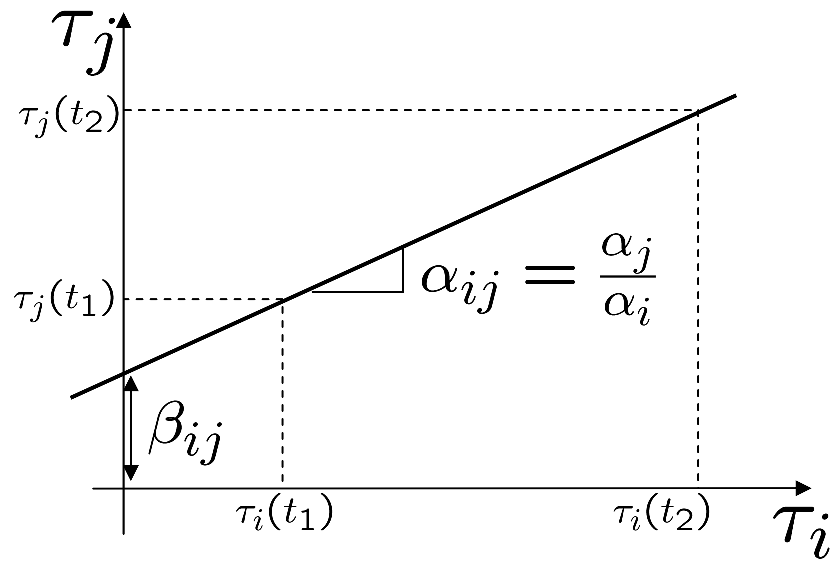
- Static: x(t+1)=Px(t)
  - Markov chains (60's), (Perron-Frobenious theorem)
- Time-varying (worst case): x(t+1)=P(t)x(t)
  - Tsitsiklis's Ph.D. thesis
  - M. Cao, A. Morse, and B. Anderson, "Reaching a consensus in a dynamically changing environment: a graphical approach'' *submitted to SIAM Journal on Control and Optimization*.
- Non-linear: x(t+1)=P(x,t)x(t)
  - L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169– 182, Feb. 2005.
- Stochastic: x(t+1)=P(t)x(t), P(t)~pdf(P)
  - Gossip algorithms: CS community, Xiao-Boyd
  - F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *to appear in IEEE Trans. on Selected areas in communication*, 2007
- Survey of its applications to control problems:
  - R. Olfati Saber and J.A. Fax and R.M. Murray, "Consensus and Cooperation in Multi-Agent Networked Systems," vol. 95, no.1, pp. 215-233 *Proceedings of IEEE*, 2007

Outline

- Motivations
- Intro to consensus algorithms
- Average Time Synch Protocol
  - Algorithm
  - Experimental results
- Conclusions

Modeling (1)





Local clocks

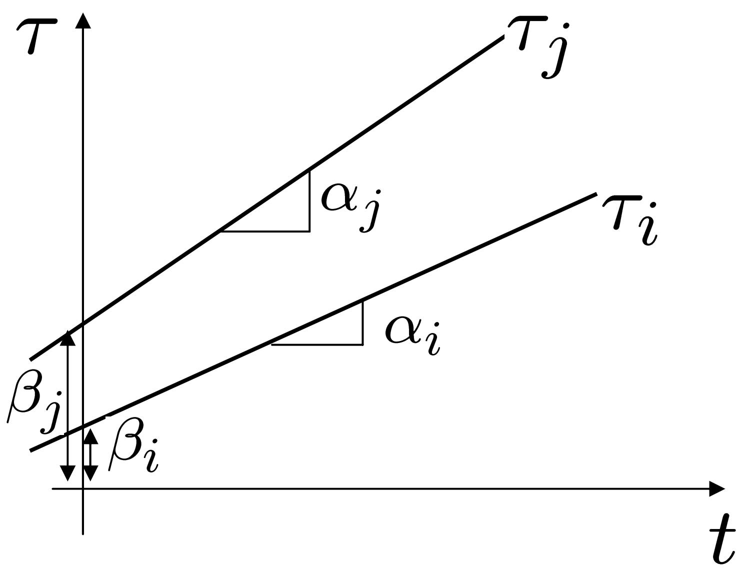
$$\tau_i(t) = \alpha_i t + \beta_i$$
$$\tau_j(t) = \alpha_j t + \beta_j$$

(α_j, β_j, t) cannot be measured directly

$$\begin{aligned} \tau_j &= \frac{\alpha_j}{\alpha_i}\tau_i + (\beta_j - \frac{\alpha_j}{\alpha_i}\beta_i) \\ &= \alpha_{ij}\,\tau_i + \beta_{ij} \end{aligned}$$

Relative skew CAN be measured

Modeling (2)



Local clocks

$$\tau_i(t) = \alpha_i t + \beta_i \qquad i = 1, \dots, N$$

Virtual reference clock

$$\tau_v(t) = \alpha_v t + \beta_v, \alpha_v \simeq 1$$

Local clock estimate

$$\hat{\tau}_i(t) = \hat{\alpha}_j \tau_i + \hat{o}_i \qquad i = 1, \dots, N$$

$$\hat{\tau}_i(t) = \hat{\alpha}_i \alpha_i t + \hat{\alpha}_i \beta_i + \hat{o}_i$$

GOAL: find $(\hat{\alpha}_j, \hat{o}_j)$ such that

$\hat{\alpha}_i(t)\alpha_i \to \alpha_v$

$\hat{\alpha}_i(t)\beta_i + \hat{o}_i(t) \to \beta_v$

$\forall i = 1, .., N$

Averaging for skew compensation

$$x_i(t) \triangleq \hat{\alpha}_i(t)\alpha_i \to \alpha_v$$

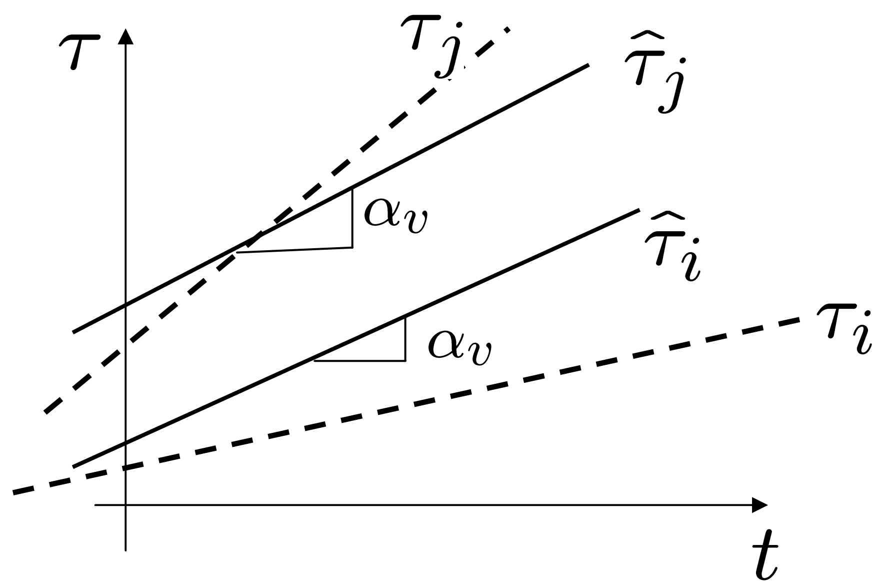
$$x_i^+ = (1-\rho)x_i + \rho x_j$$

&

Graph sufficiently connected

$$\hat{\alpha}_i^+\alpha_i = (1-\rho)\hat{\alpha}_i\alpha_i + \rho\hat{\alpha}_j\alpha_j$$

$x_i(t) \to \alpha_v \in$ ConvexHull$[x_1(0), .., x_N(0)]$



$$\hat{\alpha}(0) = 1$$

$$\hat{\alpha}_i^+ = (1-\rho)\hat{\alpha}_i + \rho\frac{\alpha_j}{\alpha_i}\hat{\alpha}_j$$

$\alpha_v \in$ ConvexHull$[\alpha_1(0), .., \alpha_N(0)]$

Averaging for offset compensation

After skew compensation:

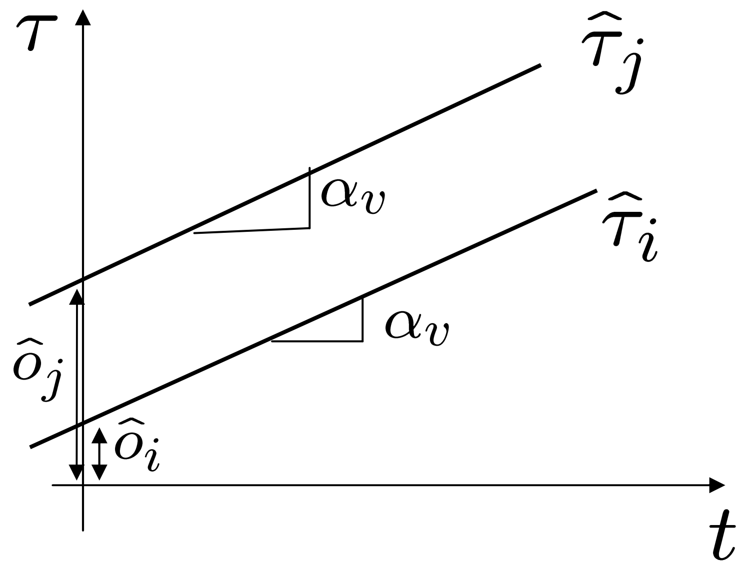
$$\hat{\tau}_i(t) = \alpha_v t + \hat{o}_i$$

$$\hat{\tau}_j(t) = \alpha_v t + \hat{o}_j$$

we want

$$\hat{o}_i(t) \to \beta_v, \ \forall i = 1, .., N$$

$$\begin{aligned} \hat{o}_i^+ &= (1-\rho)\hat{o}_i + \rho\hat{o}_j \\ &= \hat{o}_i + \rho(\hat{o}_j - \hat{o}_i) \\ &= \hat{o}_i + \rho(\hat{\tau}_j - \hat{\tau}_i) \end{aligned}$$



Average Time Synchronization Protocol (ATSP)

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

- **Relative Skew Estimation**

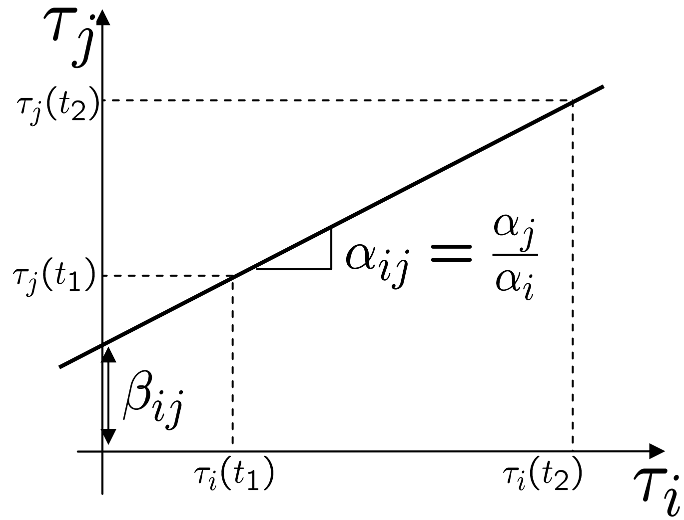
$$\eta_{ij}(0) = 1$$
$$\eta_{ij}^{+} = \rho_\eta \eta_{ij} + (1-\rho_\eta)\frac{\tau_j(t_2)-\tau_j(t_1)}{\tau_i(t_2)-\tau_i(t_1)}$$
$$\boxed{\eta_{ij}(t) \longrightarrow \alpha_{ij}}$$

- **Skew Compensation**

$$\widehat{\alpha}_i(0) = 1$$
$$\widehat{\alpha}_i^{+} = (1-\rho_\alpha)\widehat{\alpha}_i + \rho_\alpha \eta_{ij}\,\widehat{\alpha}_j$$
$$\boxed{\hat{\alpha}_i(t) \longrightarrow \alpha_v}$$

- **Offset Compensation**

$$\widehat{o}_i(0) = 0$$
$$\begin{aligned}\widehat{o}_i^{+} &= \widehat{o}_i + \rho_o(\widehat{\tau}_j - \widehat{\tau}_i) \\ &= \widehat{o}_i + \rho_o(\widehat{\alpha}_j\tau_j + \widehat{o}_j - \widehat{\alpha}_i\tau_i - \widehat{o}_j)\end{aligned}$$
$$\boxed{\hat{o}_i(t) \longrightarrow \beta_v}$$



$$\boxed{t \to \infty,\ \hat{\tau}_i(t) = \hat{\tau}_j(t),\ \forall (i,j)}$$

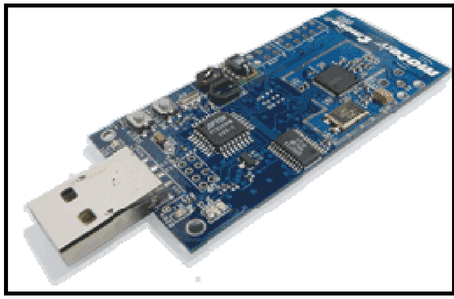
Implementation

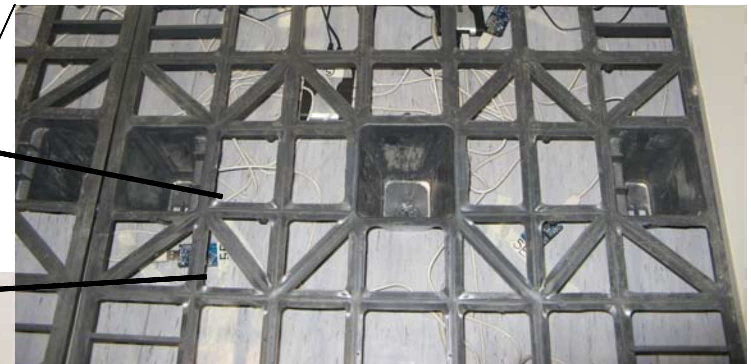
Algorithm 1 Node i: Parameter update

Input: synch packet with data $(\tau_j, \tau_j^*, \hat{o}_j^*, \hat{\alpha}_j)$ from node j

$\tau_i \leftarrow$ read_local_clock()
if j is a new node **then**
    $\eta_{ij} \leftarrow 1$
else
    $\eta_{ij} \leftarrow \rho_\eta \eta_{ij} + (1 - \rho_\eta) \frac{\tau_j - \tau_{jj}^{old}}{\tau_i - \tau_{ij}^{old}}$
    $\hat{\alpha}_i \leftarrow \rho_\alpha \hat{\alpha}_i + (1 - \rho_\alpha) \eta_{ij} \hat{\alpha}_j$
    $\hat{o}_i^* \leftarrow \rho_o \big(\hat{\alpha}_i(\tau_i - \tau_i^*) + \hat{o}_i^*\big) + (1 - \rho_o)\big(\hat{\alpha}_j(\tau_j - \tau_j^*) + \hat{o}_j^*\big)$
    $\tau_i^* \leftarrow \tau_i$
end if
$\tau_{jj}^{old} \leftarrow \tau_j$
$\tau_{ij}^{old} \leftarrow \tau_i$

The testbed





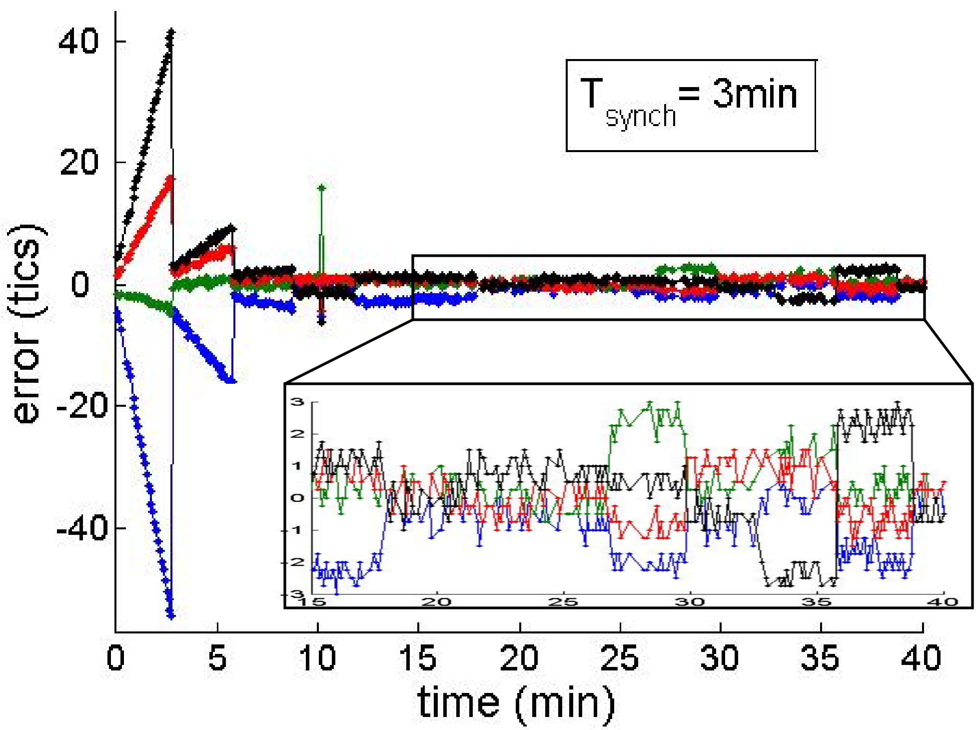
Motion Capture System (virtual GPS)

Wireless Sensor Networks (Moteiv Tmote Sky)

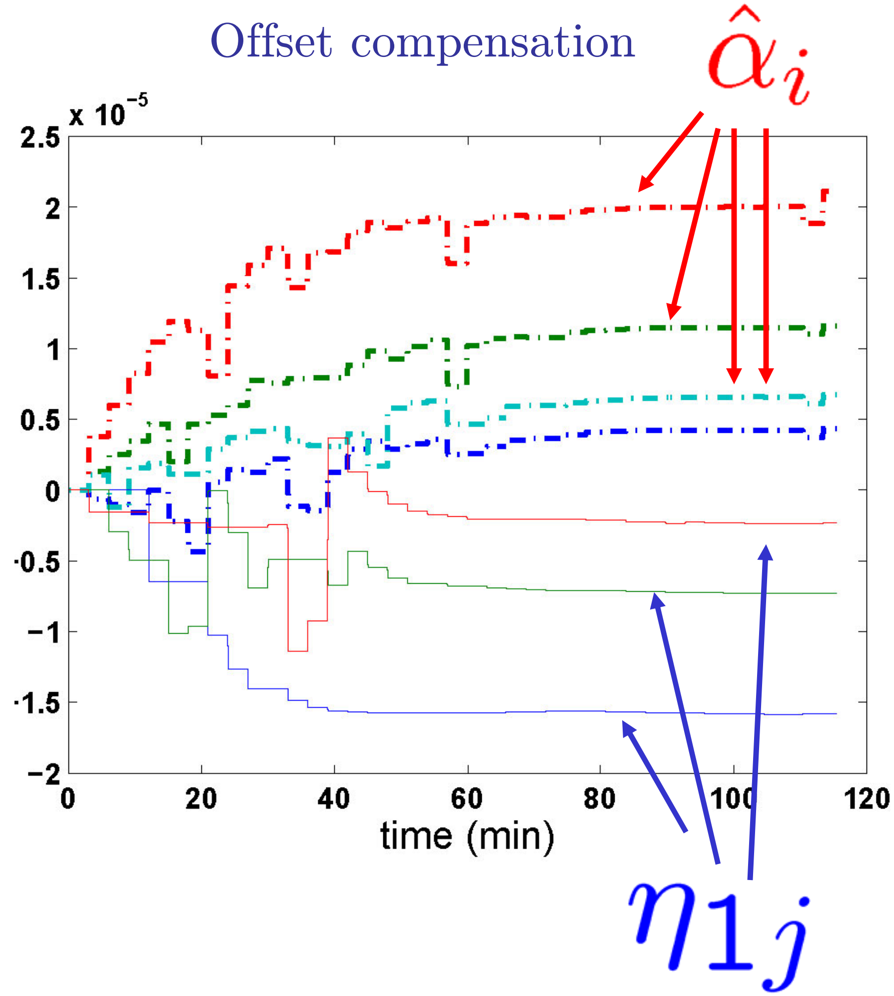
Mobile vehicles (EPFL e-puck)

Experimental results (1)

Skew compensation +
Offset compensation

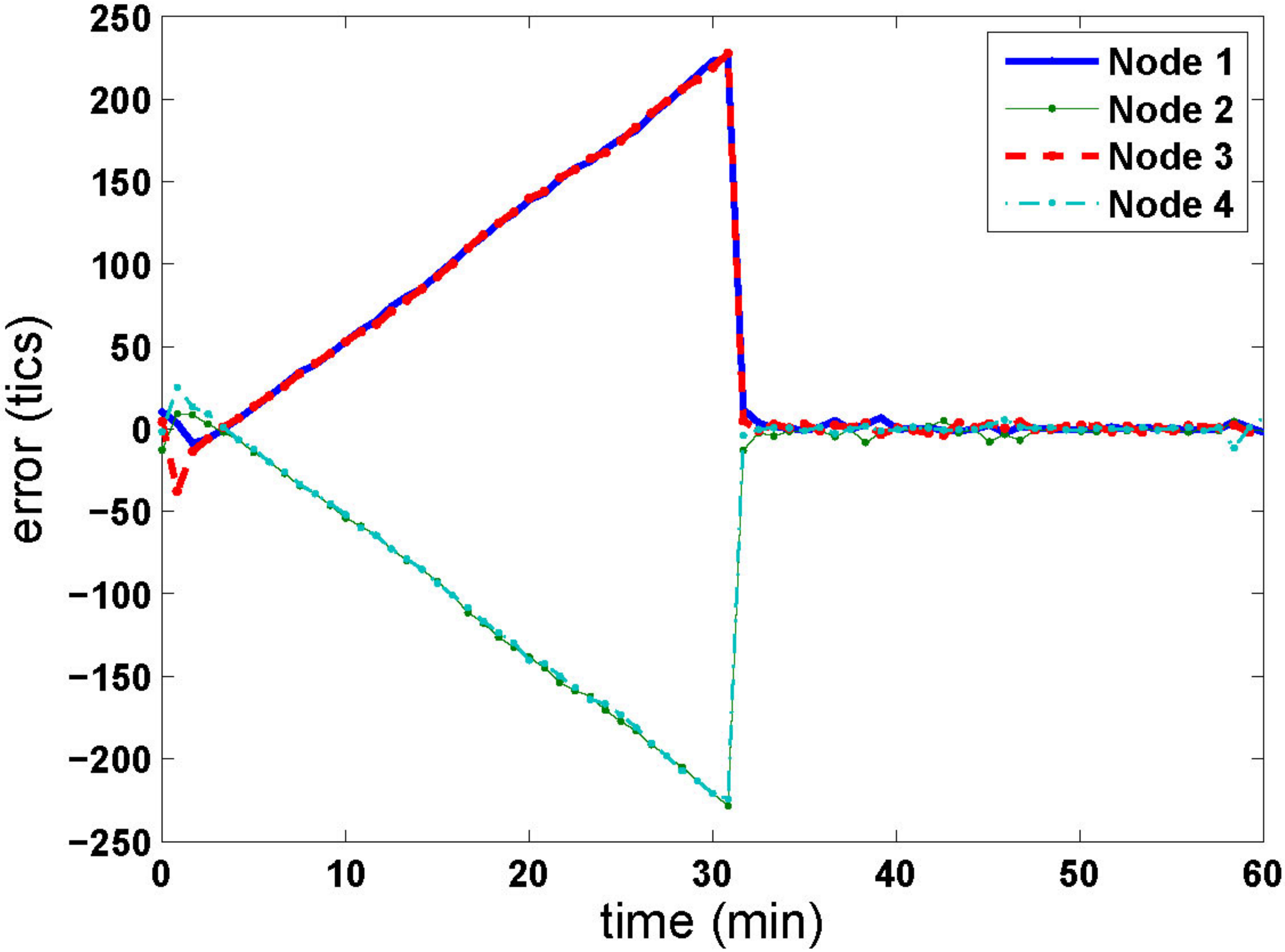


Offset compensation

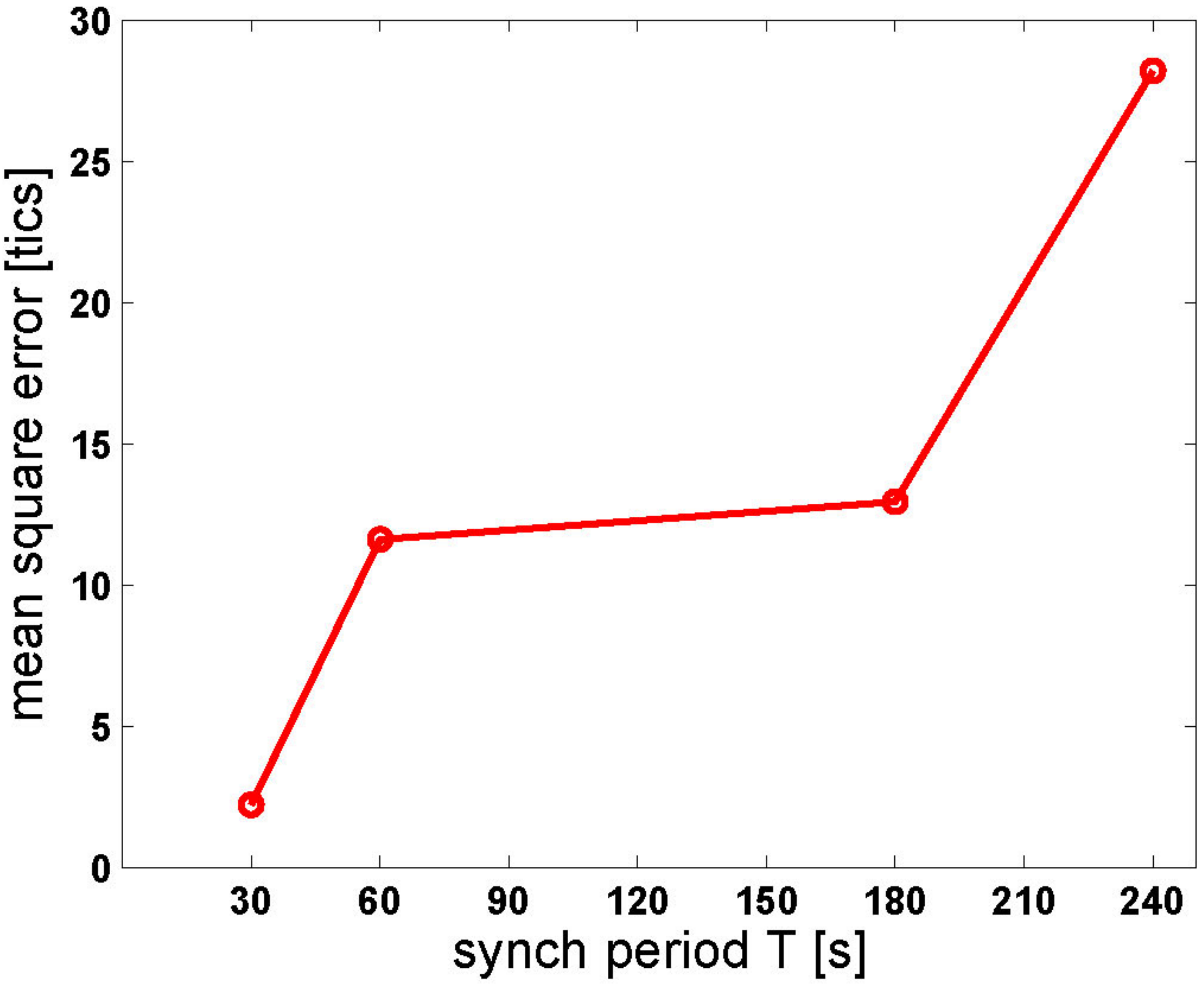


4 Nodes
Synch. period = 3min
1 tic = 30μs (32kHz clock)

Experimental results (2)



Experimental results (3)



Conclusions

- Time-synch in sensor network is natural example of consensus algorithms
- Average Time Sych Protocol
  - Purely distributed
  - Robust to packet loss, time-varying network topology
  - Asynchronous
  - Minimal memory and computational requirements
- Preliminary results are promising
- Still software issues with MAC layer time-stamping
- Details: L. Schenato, G. Gamba, *"A distributed consensus protocol for clock synchronization in wireless sensor network",* to appear in CDC'07

Future work

- How to compute optimal weights ρ?
- Can estimate mean error as function of network size, i.e. #nodes & #links/node, and noise?
- Test on a 8x8 network grid and compare with state-of-art time-synch protocols
- Use it for TDMA scheduling and power saving
- Even simpler consensus algorithms ?
  - R. Carli, A. Chiuso, L. Schenato, S. Zampieri: *"A PI Consensus Controller for Networked Clocks Synchronization"*, submitted to IFAC'08