Distributed consensus protocols for clock synchronization in sensor networks

Luca Schenato

joint work with:
A. Basso, G. Gamba

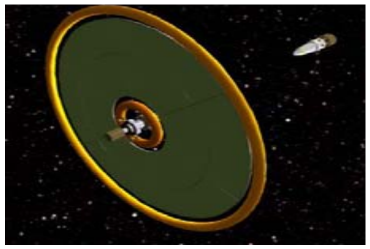
DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Networked Control Systems

Drive-by-wire systems

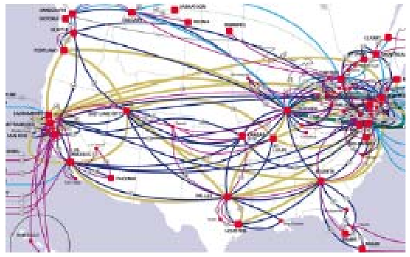
Swarm robotics

Smart structures:
adaptive space telescope

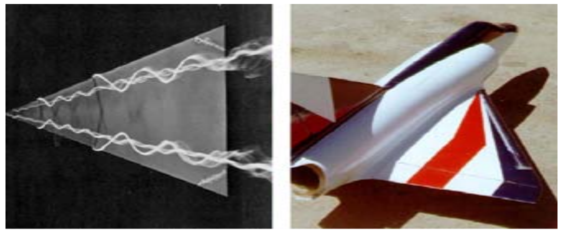


Wireless Sensor
Networks

Traffic Control:
Internet and transportation



Smart materials:
sheets of sensors and actuators

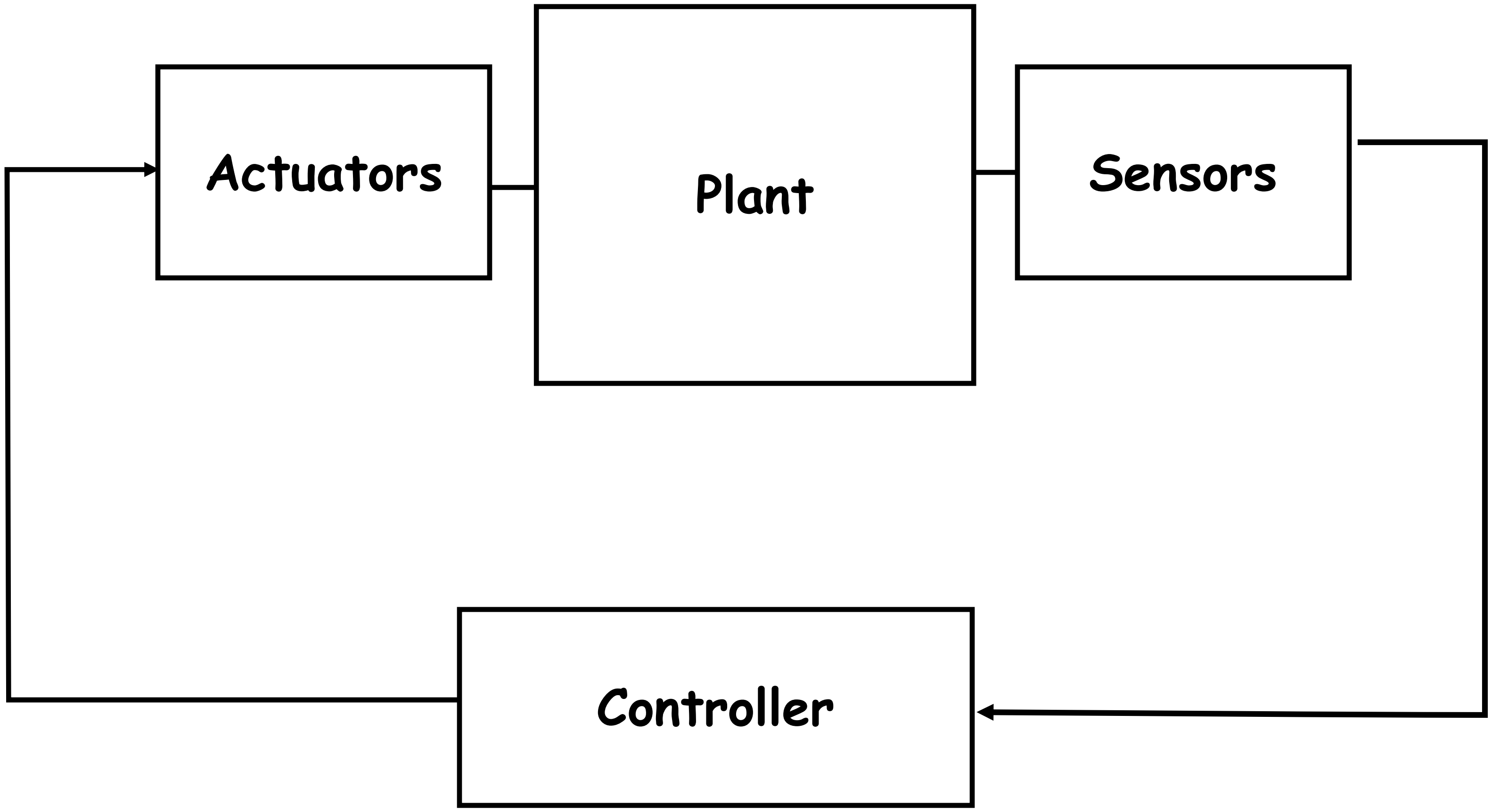


NCSs: physically distributed dynamical systems interconnected by a communication network

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

NCSs: what’s new for control?

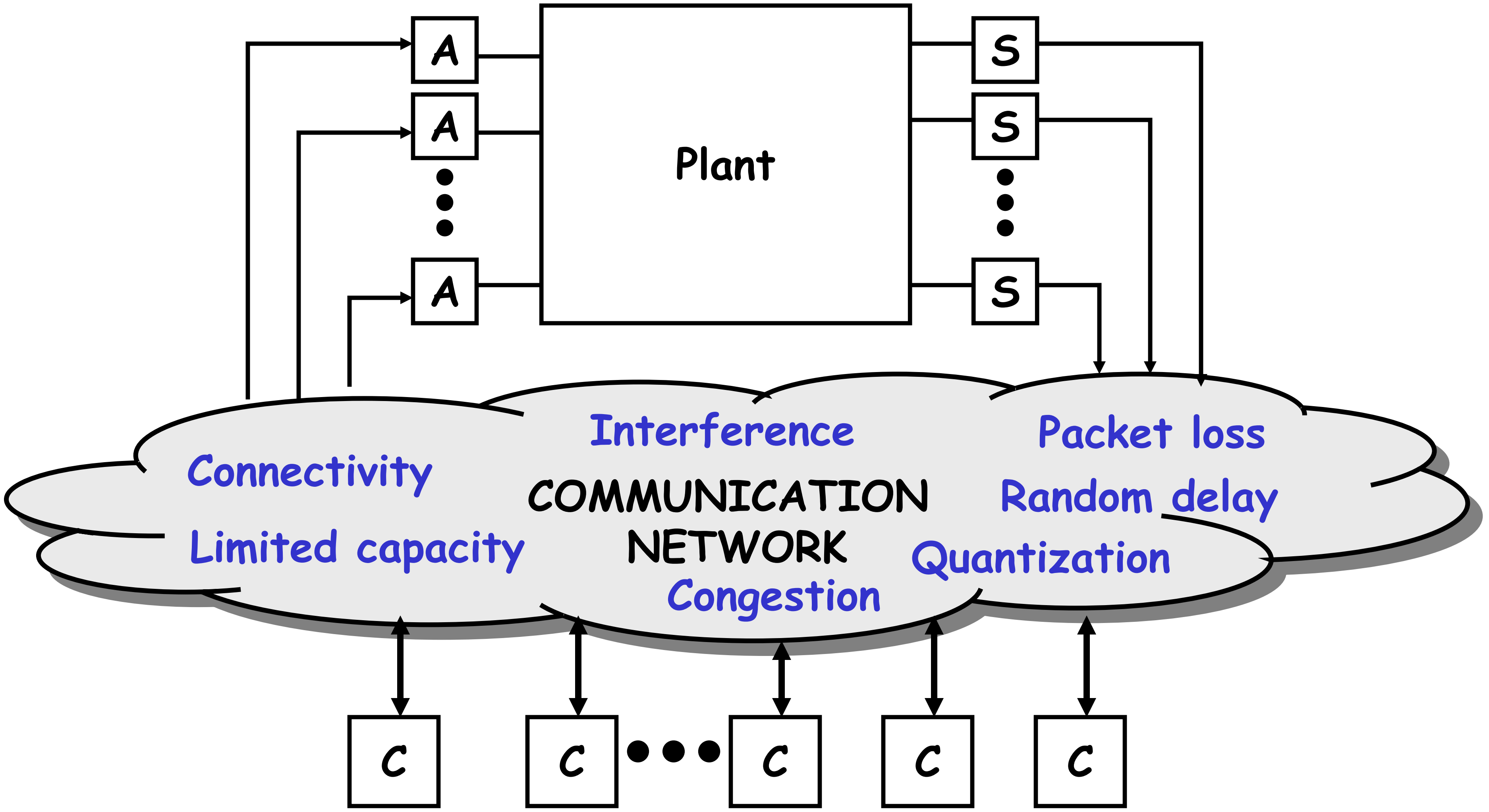
Classical architecture: Centralized structure



DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

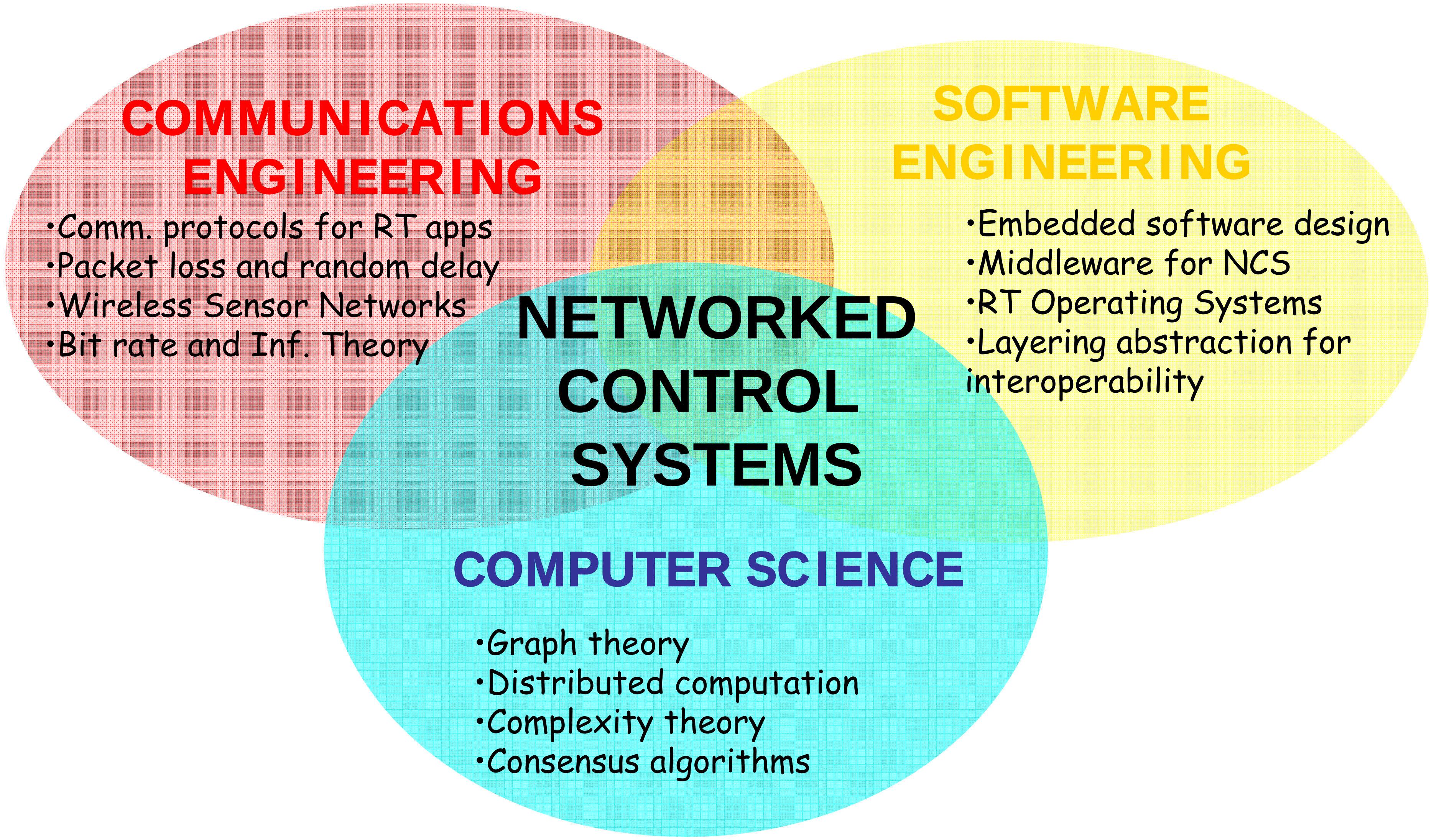
NCSs: what's new for control?

NCSs: Large scale distributed structure



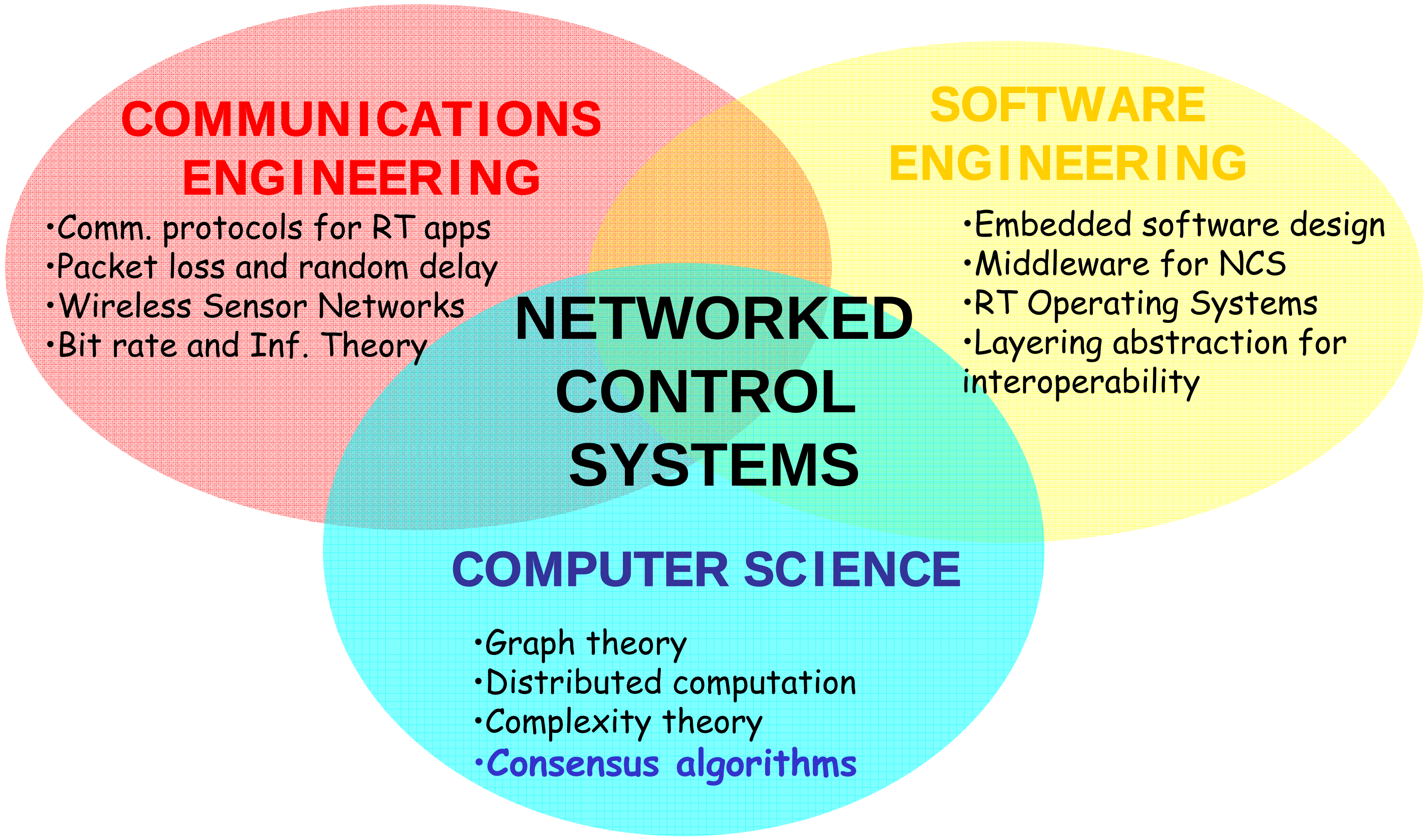
DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Interdisciplinary research needed


COMMUNICATIONS ENGINEERING
•Comm. protocols for RT apps
•Packet loss and random delay
•Wireless Sensor Networks
•Bit rate and Inf. Theory
SOFTWARE ENGINEERING
•Embedded software design
•Middleware for NCS
•RT Operating Systems
•Layering abstraction for interoperability
NETWORKED CONTROL SYSTEMS
COMPUTER SCIENCE
•Graph theory
•Distributed computation
•Complexity theory
•Consensus algorithms

DEPARTMENT OF INFORMATION ENGINEERING UNIVERSITY OF PADOVA

Interdisciplinary research needed


COMMUNICATIONS ENGINEERING
•Comm. protocols for RT apps
•Packet loss and random delay
•Wireless Sensor Networks
•Bit rate and Inf. Theory
SOFTWARE ENGINEERING
•Embedded software design
•Middleware for NCS
•RT Operating Systems
•Layering abstraction for interoperability
NETWORKED CONTROL SYSTEMS
COMPUTER SCIENCE
•Graph theory
•Distributed computation
•Complexity theory
•Consensus algorithms

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: 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” ?

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

A robotics example: the rendezvous problem

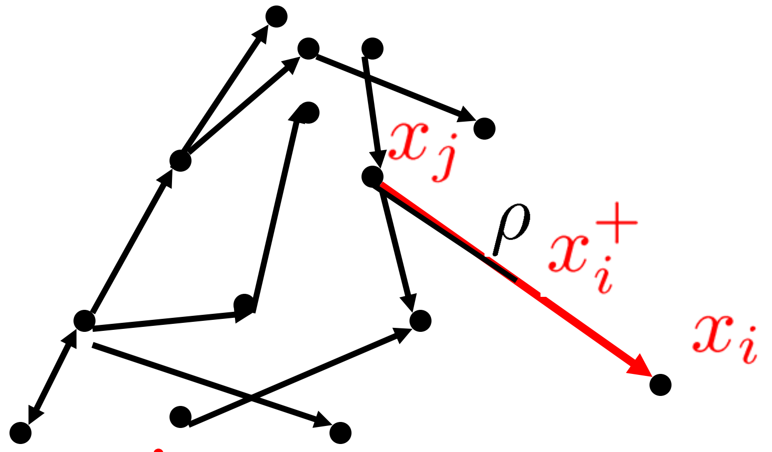
GOAL: a set of N vehicles should converge to a common location using only local communication

Vehicle dynamics

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

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

$$x^+ = x + u$$



Control law

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

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

Closed loop dynamics

$$x^+ = Px$$

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

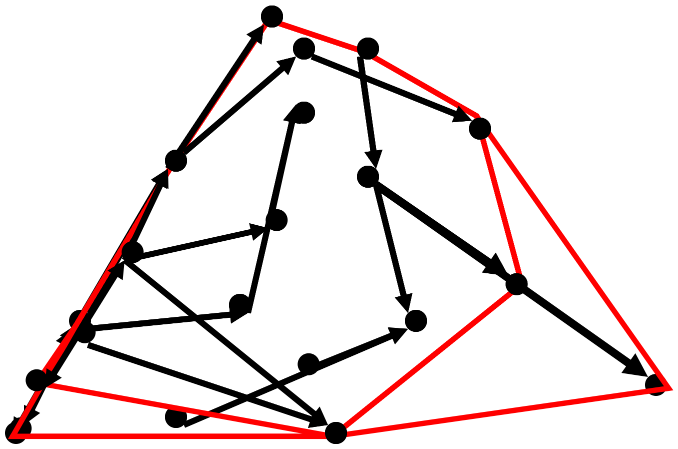
$$P = \begin{array}{c} \\ \\ \end{array}\begin{array}{cccccc|c} & & & j & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 1-\rho & 0 & \rho & 0 & 0 & \leftarrow i \\ 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & \rho & 1-\rho & 0 & \\ 0 & 0 & 0 & \rho & 0 & 1-\rho & \end{array}$$

$$P_{ij} \geq 0$$

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

(ROW) STOCHASTIC MATRIX

A robotics example: the rendezvous problem



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

$$P_{ij}\geq 0$$

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

$$P=\begin{array}{cccccccccc} & & & & & j & & & & \\ \left(\right. & 1-\square & 0 & \cdots & & \square\ 0 & 0 & 0 & \left.\right) & \\ & & \ddots & & & & & & & \\ & 0\ 0\ 0 & 1-\square & 0 & & \square & 0 & 0 & & \leftarrow i \\ & & & & \ddots & & & & & \end{array}$$

(ROW) STOCHASTIC MATRIX

$$x^+=Px$$

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

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

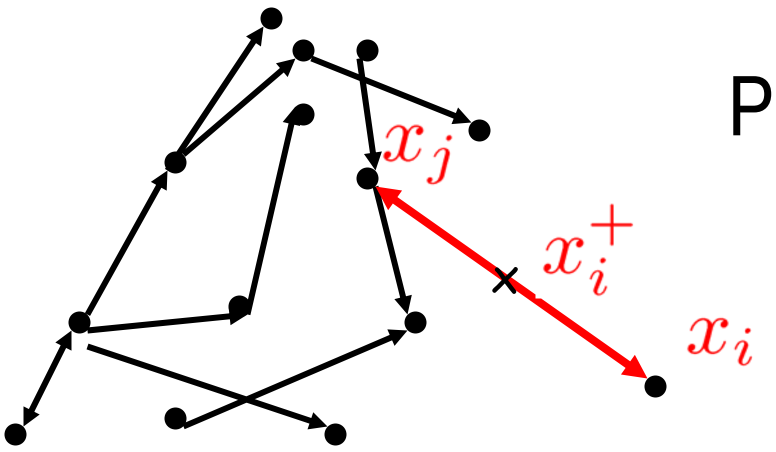
$$\alpha\in\mathrm{convHull}(x_1(0),\ldots,x_N(0))$$

A robotics example: the rendezvous problem

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

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

Center of mass is constant



$$P = \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\begin{array}{cccccccccc} i & & j & & & & & & \\ \downarrow & & \downarrow & & & & & & \\ 1-□ & 0 & \cdot\,□\,\cdot & 0 & 0 & 0 & 0 & & \leftarrow i \\ & \cdot & & & & & & & \\ & & \cdot & & & & & & \\ □ & 0 & 0 & 1-□ & 0 & 0 & 0 & 0 & \leftarrow j \\ & & & \cdot & & & & & \\ & & & & \cdot & & & & \end{array}$$

$$P_{ij} \geq 0$$

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

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

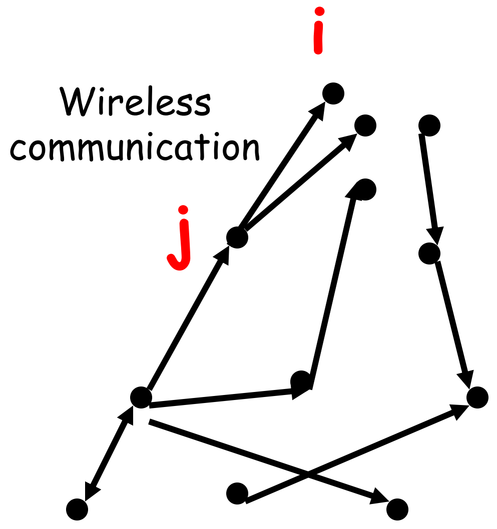
$$\boxed{\text{DOUBLY STOCHASTIC MATRIX}}$$

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

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

A signal processing example: the average consensus

GOAL: Compute best estimate of random variable



$y_i = x + v_i$, measurament of node i

$v_i \sim \mathcal{N}(0,\sigma)$ gaussian noise

$x \in \mathbb{R}$ variable to be estimated

$$\hat{x}^{global} = \frac{1}{N}\sum y_i$$

$\hat{x}_i^{local}(0) = y_i(0)$, initialization

$$x_i^{local}(t+1) = (1-\rho)\hat{x}_i^{local}(t) + \rho\hat{x}_j^{local}(t)$$

$P_{ij} \geq 0$

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

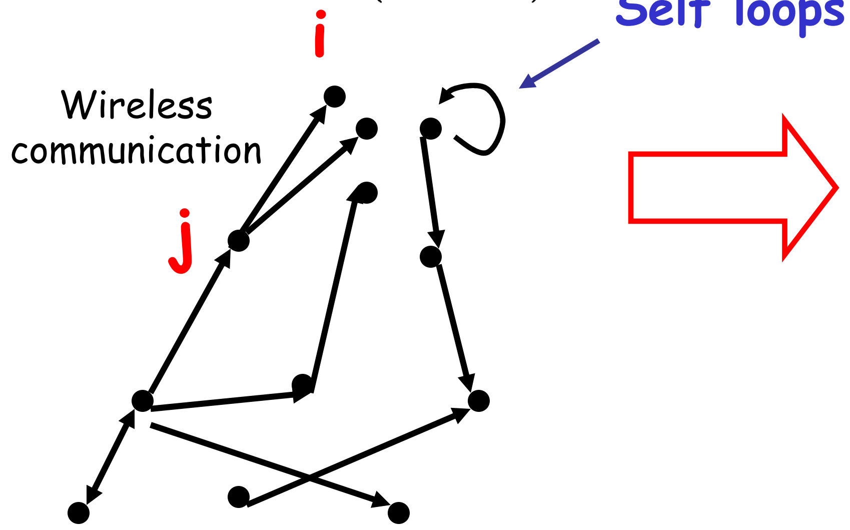
$\mathbf{1}^T P = \mathbf{1}^T$

graph well connected

$$\Rightarrow \boxed{\hat{x}_i^{local}(t) \longrightarrow \hat{x}^{global}, \ \forall i = 1, \ldots, N}$$

Graphs & stochastic matrices

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$



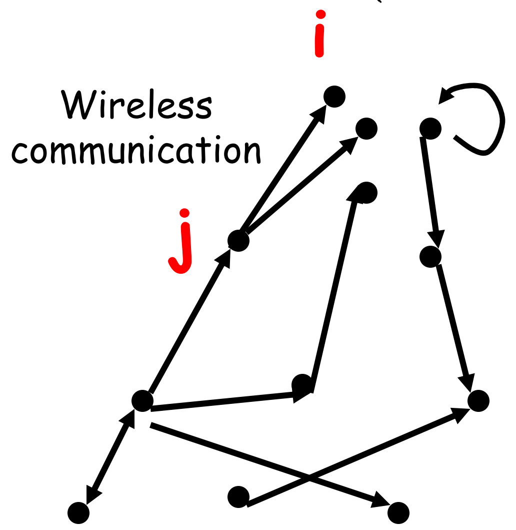
$$P = \begin{pmatrix} * & 0 & * & 0 & 0 & 0 & * & 0 & 0 \\ 0 & * & 0 & * & 0 & 0 & * & * & 0 \\ 0 & 0 & * & * & 0 & 0 & * & * & 0 \\ & & & \ddots & & & & & \\ 0 & 0 & * & 0 & * & 0 & * & 0 & * \end{pmatrix}$$

$$P_{ij} \geq 0$$
$$P\mathbf{1} = \mathbf{1}$$
$$P \sim \mathcal{G}$$

- Given *G*
  - When ∃P that achieves consensus ?
  - When ∃P that achieves average consensus ?
  - How to design P for fastest convergence ?
  - How to compute optimal P_{ij} using local communication (distributed) ?
  - How does performance scale with # nodes ?
  - What about time-varying or state-dependent graph & matrices, i.e. P=P(t,x) ?

When ∃P that achieves consensus ?

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$



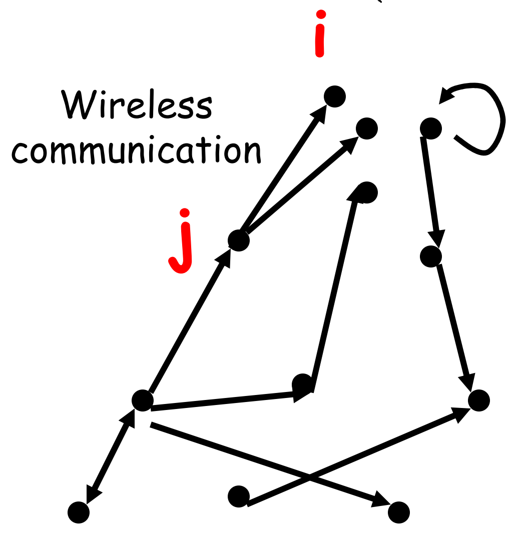
$$P = \begin{pmatrix} 1/3 & 0 & 1/3 & 0 & 0 & 0 & 1/3 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 \\ & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \end{pmatrix}$$

- Iff graph is connected, i.e. path i→j or j→i & and the graph formed by maximally strongly connected subgraphs has only one sink
- (suboptimal) P is $P_{ij} = \frac{1}{\text{in}-\text{degree}(i)}$ where in-degree=sum of non-zero entry in the row, i.e. incoming links
- Can be computed in distributed fashion
- If graph not sufficiently connected, agents converge to convex hull of some anchor points

"Analysis of coordination in multiple agents formations through partial difference equations",
G Ferrari-Trecate, A Buffa, M Gati, submitted for pub.

When ∃P that achieves average consensus ?

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$



- Iff graph of strongly connected, i.e. there is path i→j and j→i
- Not easy to find P, in fact $P_{ij} = \frac{1}{\text{in}-\text{degree}(i)}$ does not work
- If graph is undirected, then ∃ $P=P^T$, can be computed in distributed fashion (SUBOPTIMAL)

"Consensus and Cooperation in Networked Multi-Agent Systems", R Olfati-Saber, JA Fax, RM Murray, PIEEE Jan 07

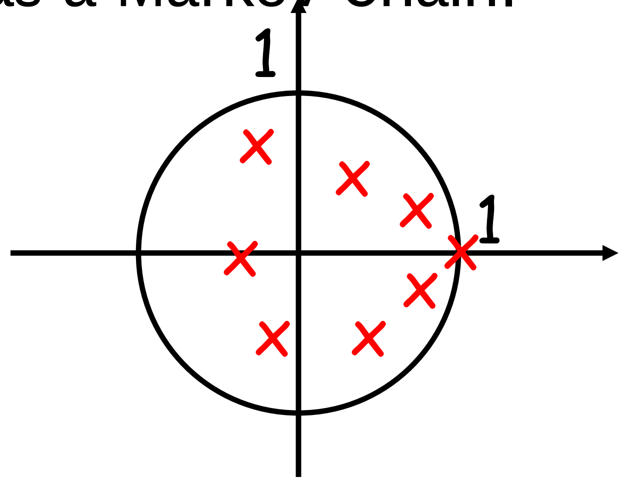
DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

How to design P for fastest convergence ?

- Stochastic matrix P can be seen as a Markov Chain.

$$\lambda_i(P) \le 1, \sigma = |\lambda_2|$$

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



- 1-σ = spectral gap

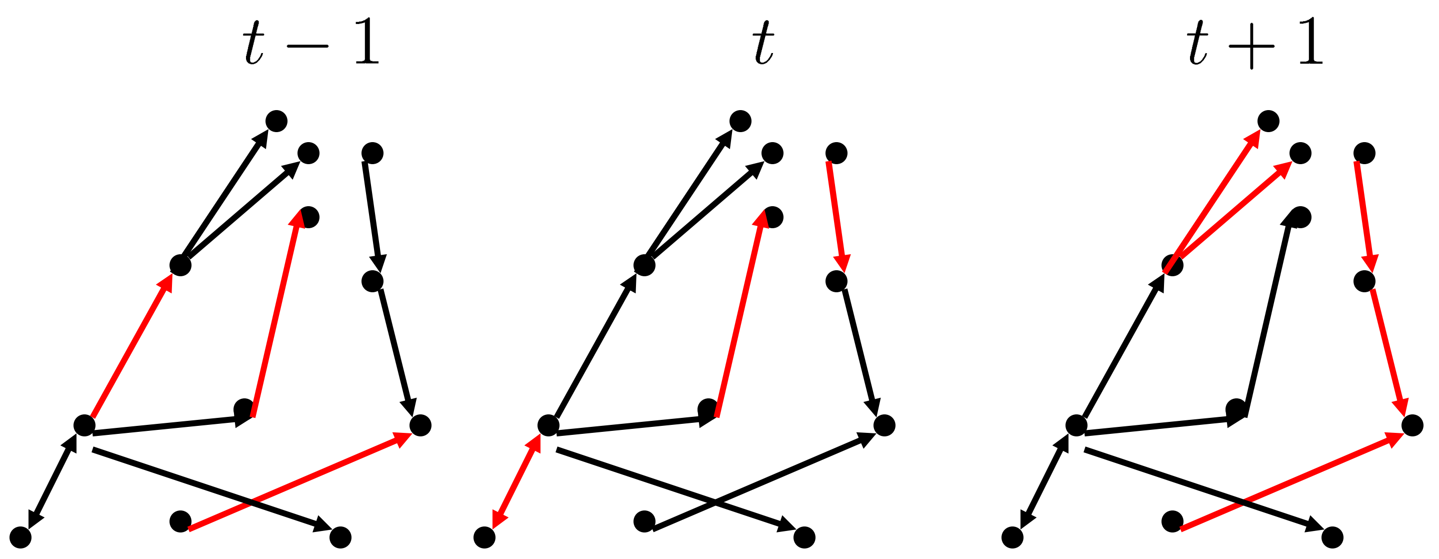
$$\boxed{\begin{array}{ll} \min_P & \sigma(P) \\ s.t. & P \sim \mathcal{G} \\ & P \text{ stochastic} \end{array}}$$

- Very hard problem (centralized) in general. Some fast convex algorithm if
  - G undirected
  - G has symmetries (Cayley graphs & circulant matrices)

"Fastest mixing Markov chain on a graph", S. Boyd, P Diaconis, L. Xiao, SIAM Review 2004

"Communication constraints in the average consensus problem", R.Carli F. Fagnani, A. Speranzon, S. Zampieri, to apper Automatica

Time-varying communication algorithms



$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$
$$P_{ij}(t) \geq 0$$
$$P(t)\mathbf{1} = \mathbf{1}$$
$$P(t) \sim \mathcal{G}$$

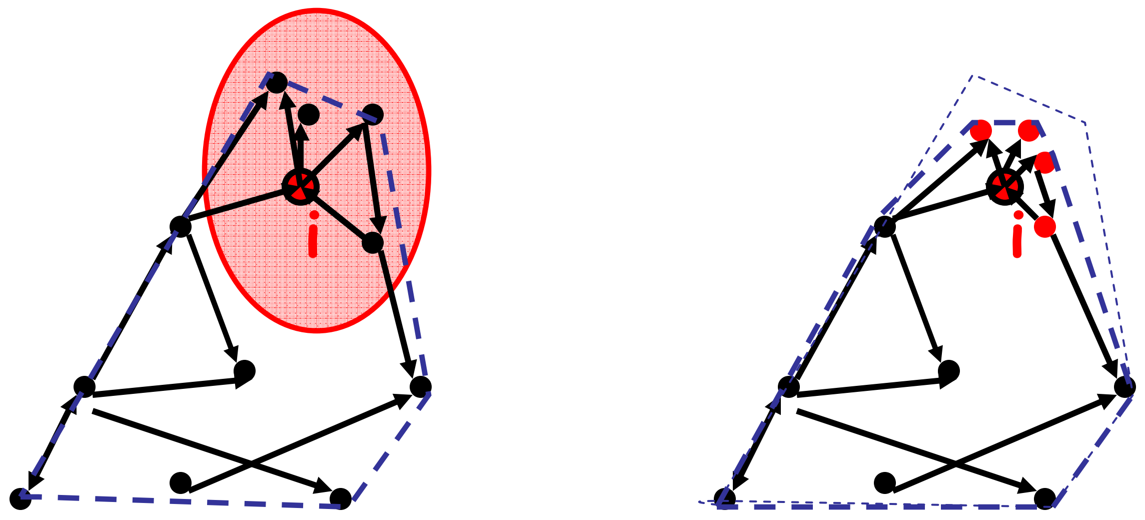
$$x(t) = \left(\prod_{k=1}^{t} P(k)\right) x(0) \overset{?}{\to} \alpha\mathbf{1}$$

- If union of sub-graphs within a sufficiently long time-window, are strongly connected, then ∃ P(t) that guarantee convergence

"Coordination of groups of mobile autonomous agents using nearest neighbor rules" A. Jadbabaie, J. Lin, and A. S. Morse, TAC '03

- If pairwise update guarantees average consensus, $P_{ij}(t)=P_{ji}(t)$

Randomized communication algorithms



$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$
$$P_{ij}(t) \geq 0$$
$$P(t)\mathbf{1} = \mathbf{1}$$
$$P(t) \sim \mathcal{G}$$

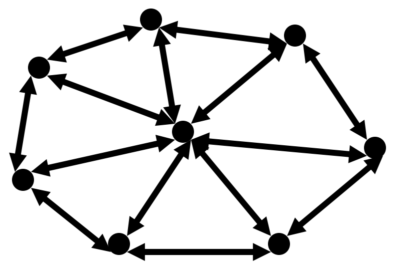
- i→j, j∈ Neighbors(i), at random with probability $p_{i\to j}$
- Do averaging when link established, $x_j^+ = \frac{x_j + x_i}{2}$
- p_{ij} can be determined by sensor network (packet loss prob.)
- p_{ij} can be designed (comm. protocol) to increase convergence speed
- For geometric random graphs, random walk is close to optimal choice

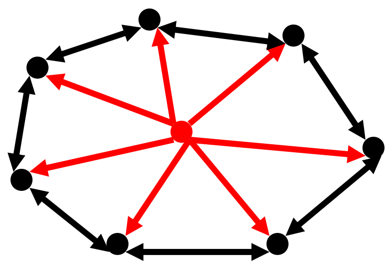
$$p_{i\to j} = \frac{1}{\text{out}-\text{degree}(i)}$$

"Randomized Gossip Algorithms", S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, IEE TIF '03

Optimal Randomized communication algorithms

Underlying communication graph



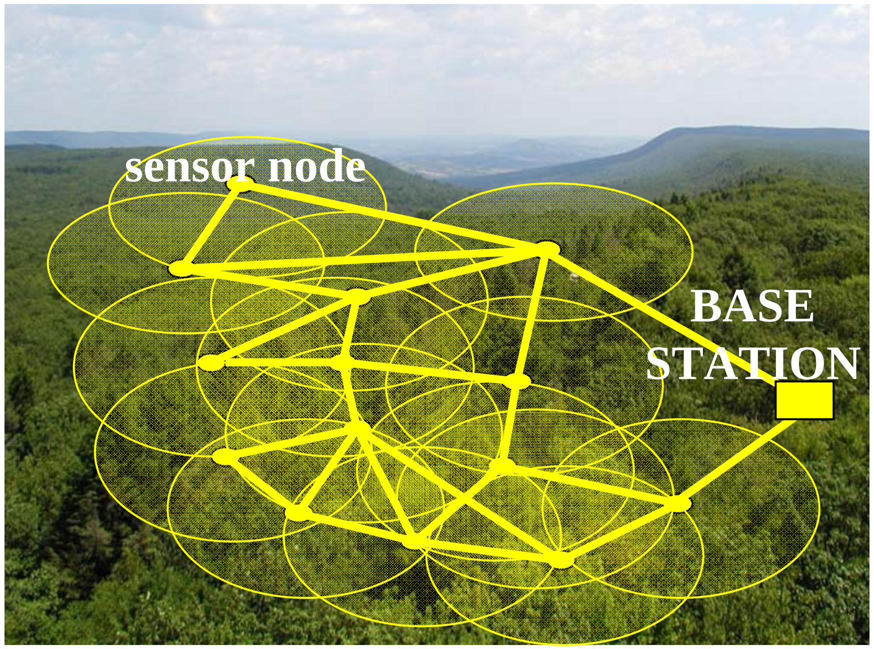


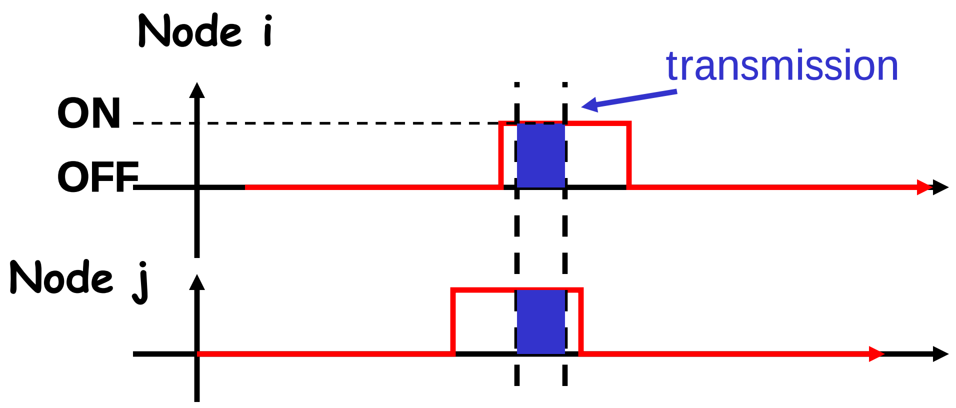
$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$
$$P_{ij}(t) \geq 0$$
$$P(t)\mathbf{1} = \mathbf{1}$$
$$P(t) \sim \mathcal{G}$$

- Given underlying communication graph (with possibly lossy links)
- Average update equation $x_j^+ = \frac{x_j + x_i}{2}$

- How should I select a randomized scheduling policy for node broadcast selection ?

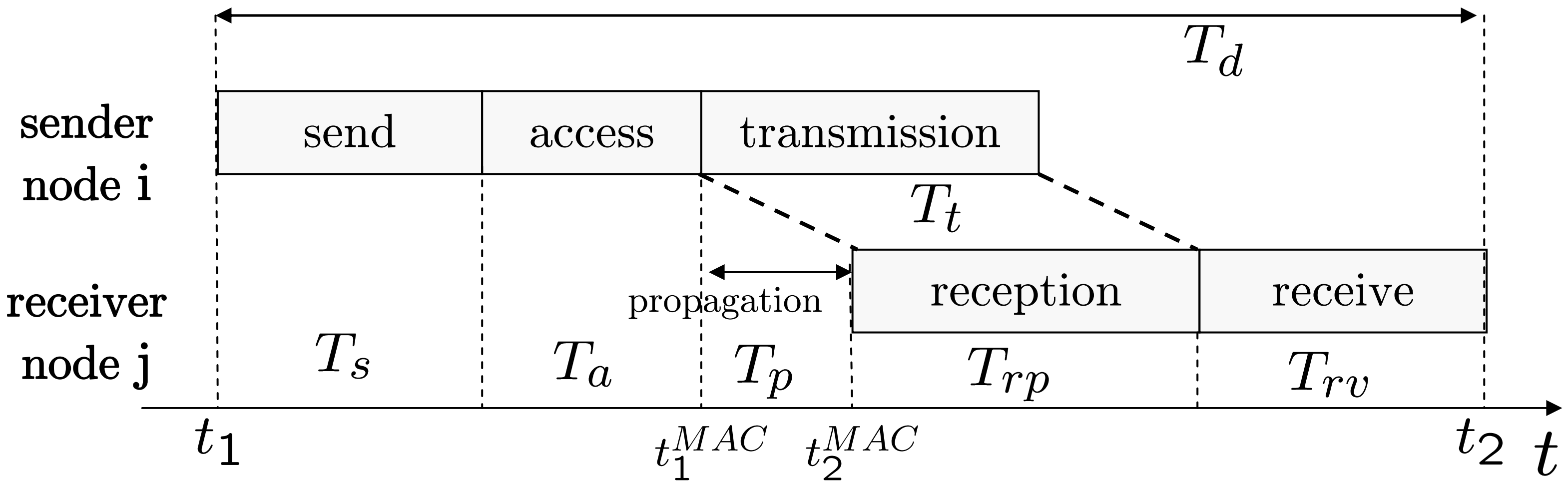
Time synchronization in sensor networks





- 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

Communication 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

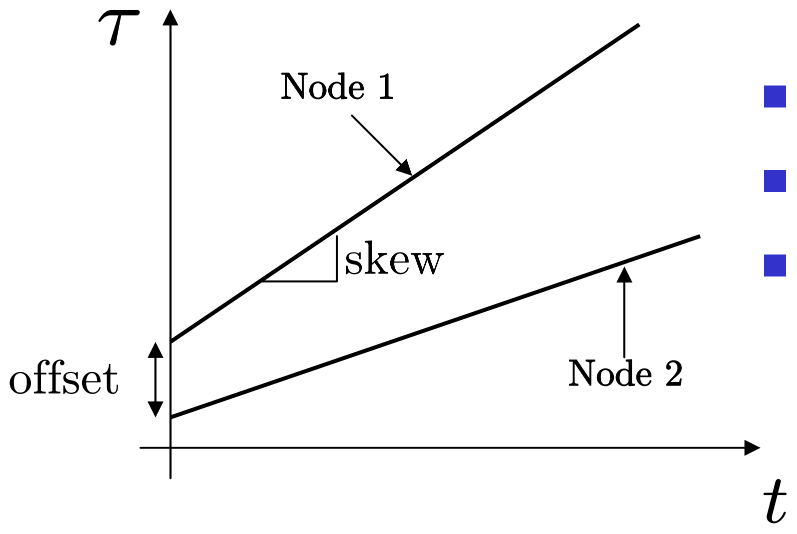
- **MAC layer time-stamping**
  - Read local clock t_1^{MAC} at node *i* when start sending first bit
  - Write t_1^{MAC} on leaving packet
  - Read and store local clock t_2^{MAC} at node *j* when start receiving first bit

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

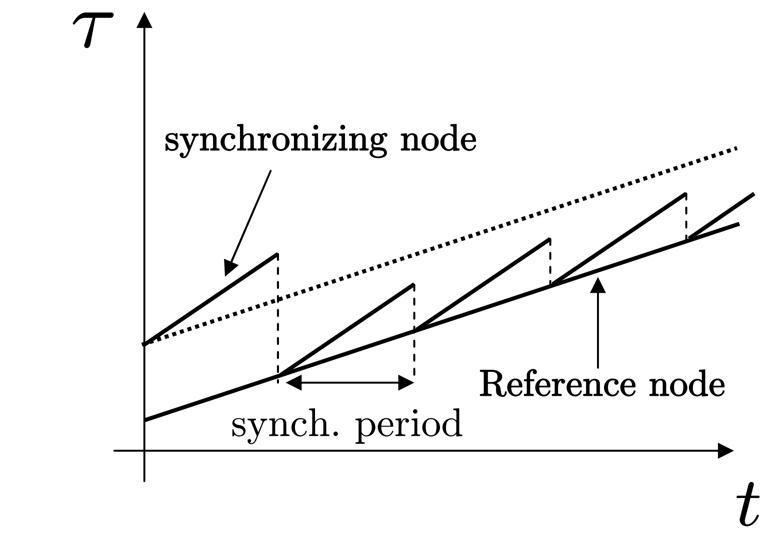
t_1^{MAC}	data	header

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Clock characteristics & standard clock pair sych



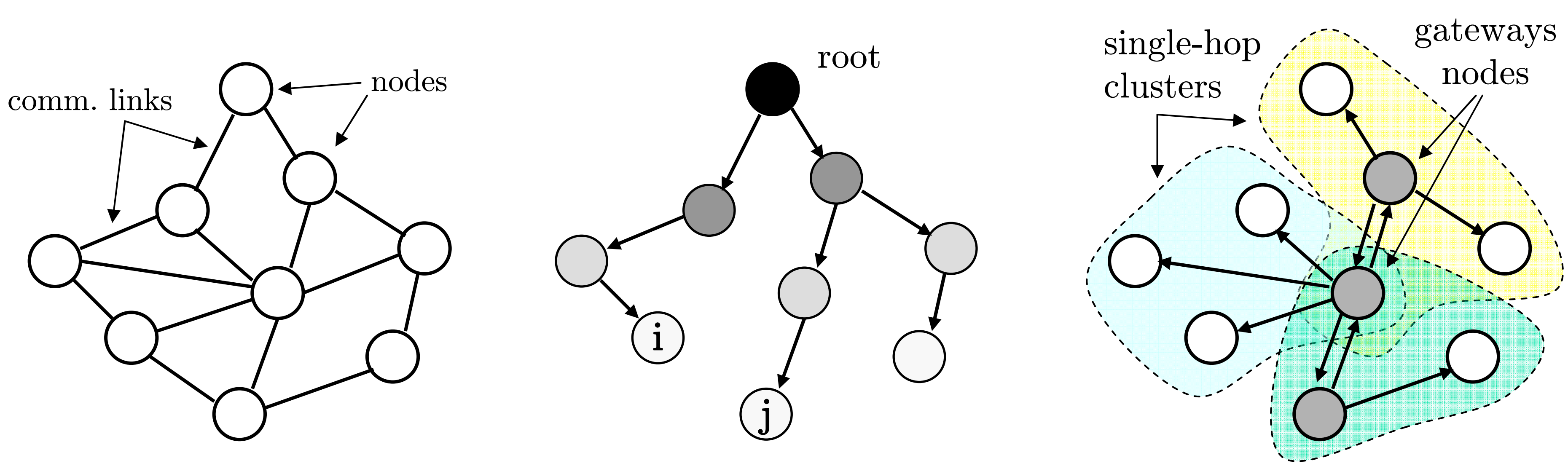
- Offset: instantaneous time difference
- Skew: clock speed
- Drift: derivative of clock speed



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

Sych topologies for sensor networks

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



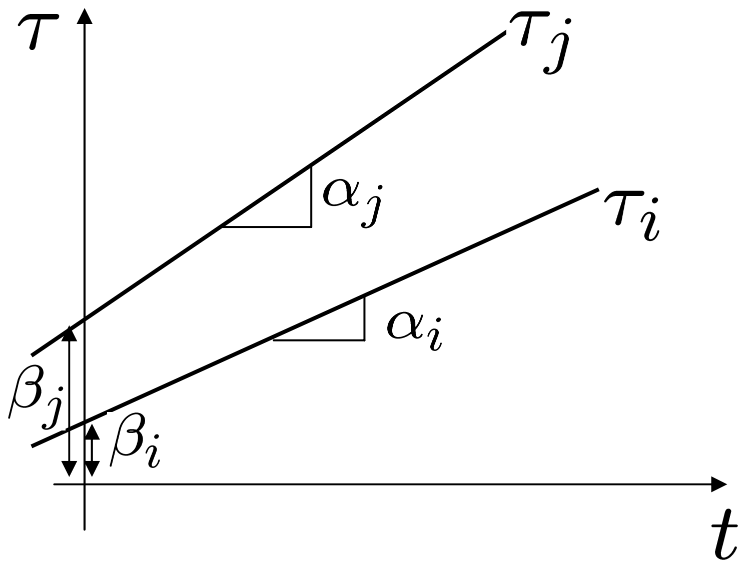
- PROS
  - Straightforward extension of pair synch
- CONS
  - Links may disappear
  - Root or gateways might temporarily disappear or die
  - New nodes might appear
  - Can be made adaptive but high protocol overhead

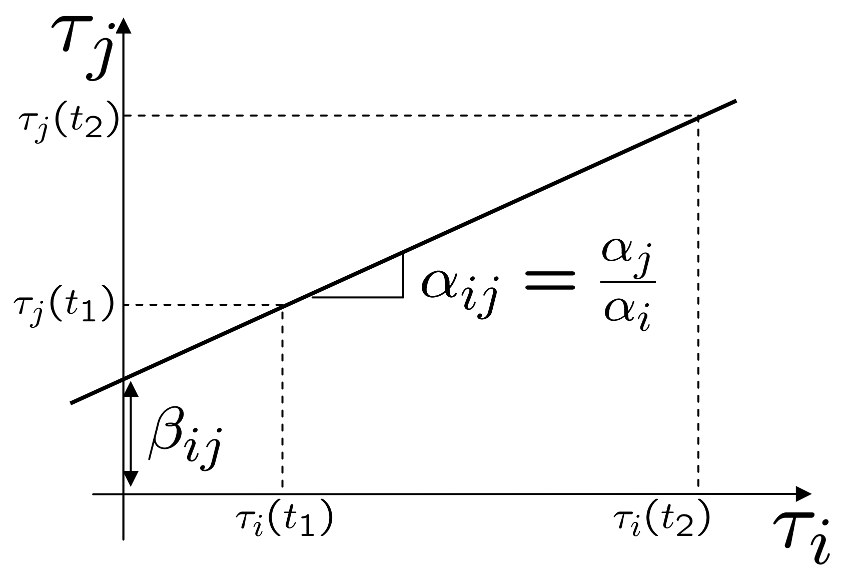
Ideal protocol features

- 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

Modeling (1)





Local clocks

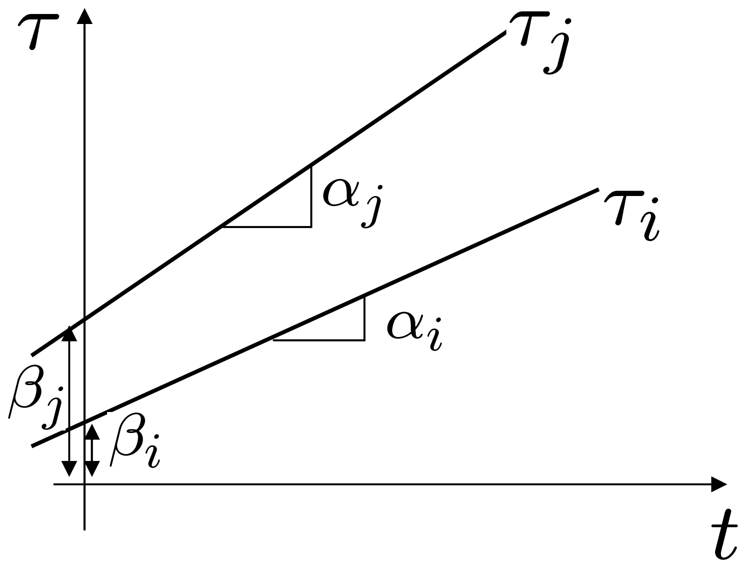
$$\tau_i(t) = \alpha_i t + \beta_i$$
$$\tau_j(t) = \alpha_j t + \beta_j$$

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

(α_j, β_j, t) cannot be measured directly

Relative skew CAN be measured

Modeling (2)



Local clocks

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

Virtual reference clock

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

Local clock estimate

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

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

GOAL: find $(\hat{\alpha}_j, \hat{o}_j)$ such that
$\lim_{t\to\infty} \hat{\tau}_i(t) = \tau_v(t), \forall i = 1, .., N$

$\Longleftrightarrow$

GOAL: find $(\hat{\alpha}_j, \hat{o}_j)$ such that
$\hat{\alpha}_i(t) \rightarrow \frac{\alpha_v}{\alpha_i}$
$\hat{o}_i(t) \rightarrow \beta_v - \frac{\alpha_v}{\alpha_i}\beta_i$
$\forall i = 1, .., N$

Averaging for skew compensation

find $\hat{\alpha}_j$ such that

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

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

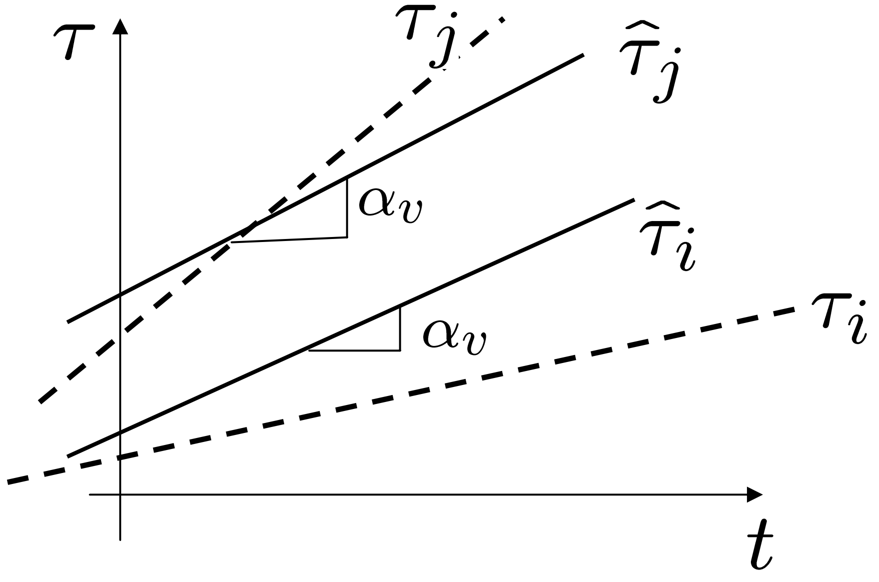
$$x_i^+ = (1-\rho)x_i + \rho x_j$$

&

Graph sufficiently connected

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

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



$$\boxed{\begin{array}{l} \hat{\alpha}(0) = 1 \\ \hat{\alpha}_i^+ = (1-\rho)\hat{\alpha}_i + \rho\frac{\alpha_j}{\alpha_i}\hat{\alpha}_j \\ \alpha_v \in \text{ConvexHull}[\alpha_1(0), .., \alpha_N(0)] \end{array}}$$

Averaging for offset compensation

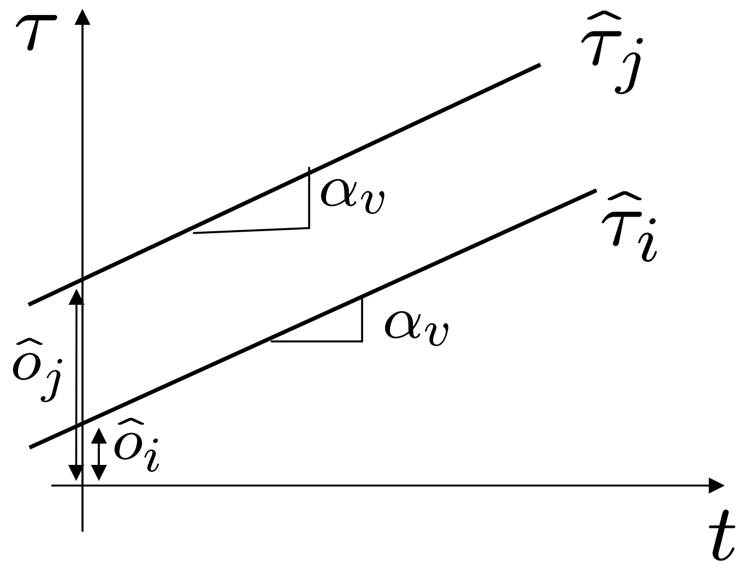
After skew compensation:

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

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

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)

- Relative Skew Estimation

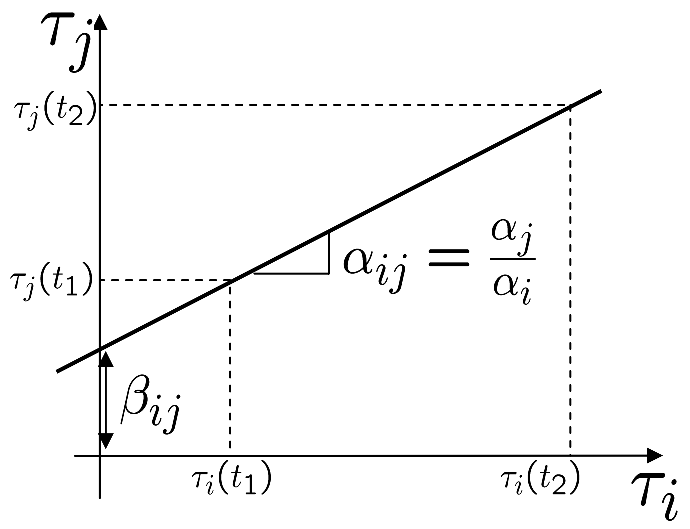
$$\eta_{ij}(0) = 1$$
$$\eta_{ij}^{+} = \rho_\eta \eta_{ij} + (1-\rho_\eta)\frac{\tau_j(t_2)-\tau_j(t_1)}{\tau_i(t_2)-\tau_i(t_1)}$$
$$\boxed{\eta_{ij}(t) \rightarrow \alpha_{ij}}$$

- Skew Compensation

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

- Offset Compensation

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



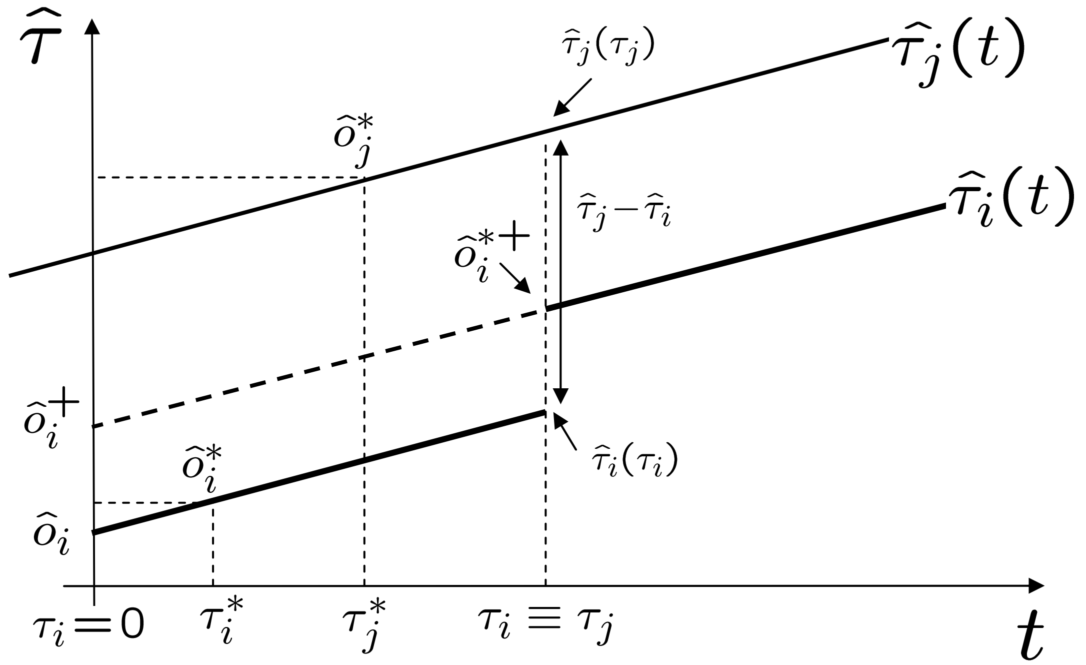
$$\boxed{t \rightarrow \infty,\ \hat{\tau}_i(t) = \hat{\tau}_j(t),\ \forall (i,j)}$$

Numerical considerations

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

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

$$\boxed{\hat{\tau}_j(t) = \hat{\alpha}_j(\tau_i - \tau_i^*) + \hat{o}_j^*}$$

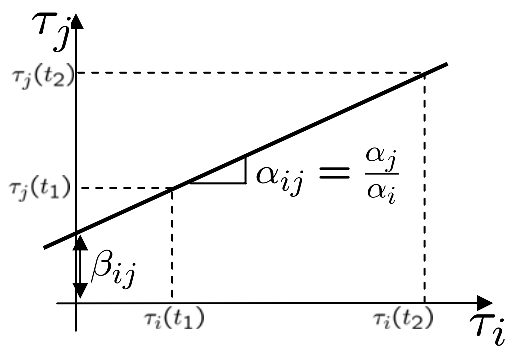


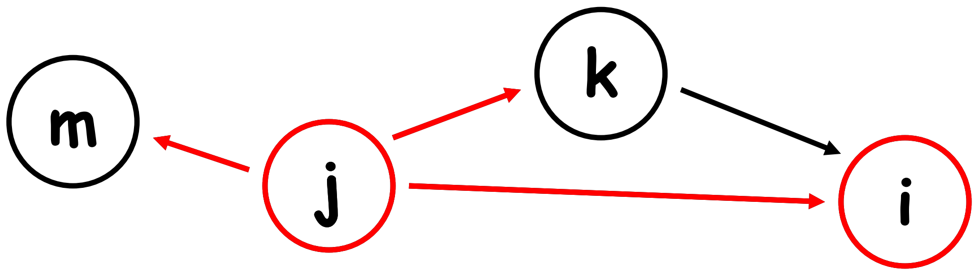
$$\begin{aligned}\hat{o}_i^{*+}(\tau_i) = &\quad \hat{\tau}_i + (1-\rho_o)(\hat{\tau}_j - \hat{\tau}_i) = \rho_o \hat{\tau}_i + (1-\rho_o)\hat{\tau}_j\\ &= \rho_o\big(\hat{\alpha}_i(\tau_i - \tau_i^*) + \hat{o}_i^*\big) + (1-\rho_o)\big(\hat{\alpha}_j(\tau_j - \tau_j^*) + \hat{o}_j^*\big)\end{aligned}$$

Implementation (1)

```
Algorithm 1 Node $i$: Parameter update
Input: synch packet with data $(\tau_j, \tau_j^*, \hat{o}_j^*, \hat{\alpha}_j)$ from node $j$
 1: $\tau_i \leftarrow$ read_local_clock()
 2: if j is a new node then
 3:     $\eta_{ij} \leftarrow 1$
 4: else
 5:     $\eta_{ij} \leftarrow \rho_\eta \eta_{ij} + (1-\rho_\eta)\frac{\tau_j - \tau_{jj}^{old}}{\tau_i - \tau_{ij}^{old}}$
 6:     $\hat{\alpha}_i \leftarrow \rho_\alpha \hat{\alpha}_i + (1-\rho_\alpha)\eta_{ij}\hat{\alpha}_j$
 7:     $\hat{o}_i^* \leftarrow \rho_o\big(\hat{\alpha}_i(\tau_i - \tau_i^*) + \hat{o}_i^*\big) + (1-\rho_o)\big(\hat{\alpha}_j(\tau_j - \tau_j^*) + \hat{o}_j^*\big)$
 8:     $\tau_i^* \leftarrow \tau_i$
 9: end if
10: $\tau_{jj}^{old} \leftarrow \tau_j$
11: $\tau_{ij}^{old} \leftarrow \tau_i$
```

$$\hat{\tau}_j(t) = \hat{\alpha}_j(\tau_i(t) - \tau_i^*) + \hat{o}_j^*$$





Send packet

τ_j	τ_j^*	$\hat{o}_j^*$	$\hat{\alpha}_j$	j

NOTE: do NOT send $\hat{\tau}_j$

Local variables of node i

in-node	h_i		
j	η_{ij}	τ_{ij}^{old}	τ_{jj}^{old}
k	η_{ik}	τ_{ik}^{old}	τ_{kk}^{old}
⋮			

τ_i^*	$\hat{o}_i^*$	$\hat{\alpha}_i$

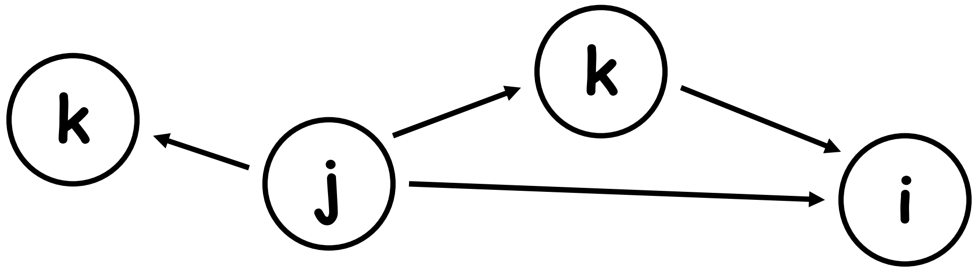
τ_i

Implementation (2)

Algorithm 1 Node i: Parameter update

```
Input: synch packet with data $(\tau_j, \tau_j^*, \hat{o}_j^*, \hat{\alpha}_j)$ from node $j$
 1: $\tau_i \leftarrow$ read_local_clock()
 2: if j is a new node then
 3:     $\eta_{ij} \leftarrow 1$
 4: else
 5:     $\eta_{ij} \leftarrow \rho_\eta \eta_{ij} + (1-\rho_\eta)\frac{\tau_j - \tau_{jj}^{old}}{\tau_i - \tau_{ij}^{old}}$
 6:     $\hat{\alpha}_i \leftarrow \rho_\alpha \hat{\alpha}_i + (1-\rho_\alpha)\eta_{ij}\hat{\alpha}_j$
 7:     $\hat{o}_i^* \leftarrow \rho_o\big(\hat{\alpha}_i(\tau_i - \tau_i^*) + \hat{o}_i^*\big) + (1-\rho_o)\big(\hat{\alpha}_j(\tau_j - \tau_j^*) + \hat{o}_j^*\big)$
 8:     $\tau_i^* \leftarrow \tau_i$
 9: end if
10: $\tau_{jj}^{old} \leftarrow \tau_j$
11: $\tau_{ij}^{old} \leftarrow \tau_i$
```

$$\hat{\tau}_j(t) = \hat{\alpha}_j(\tau_i - \tau_i^*) + \hat{o}_j^*$$



Send packet

τ_j	τ_j^*	$\hat{o}_j^*$	$\hat{\alpha}_j$	j

Local variables of node i

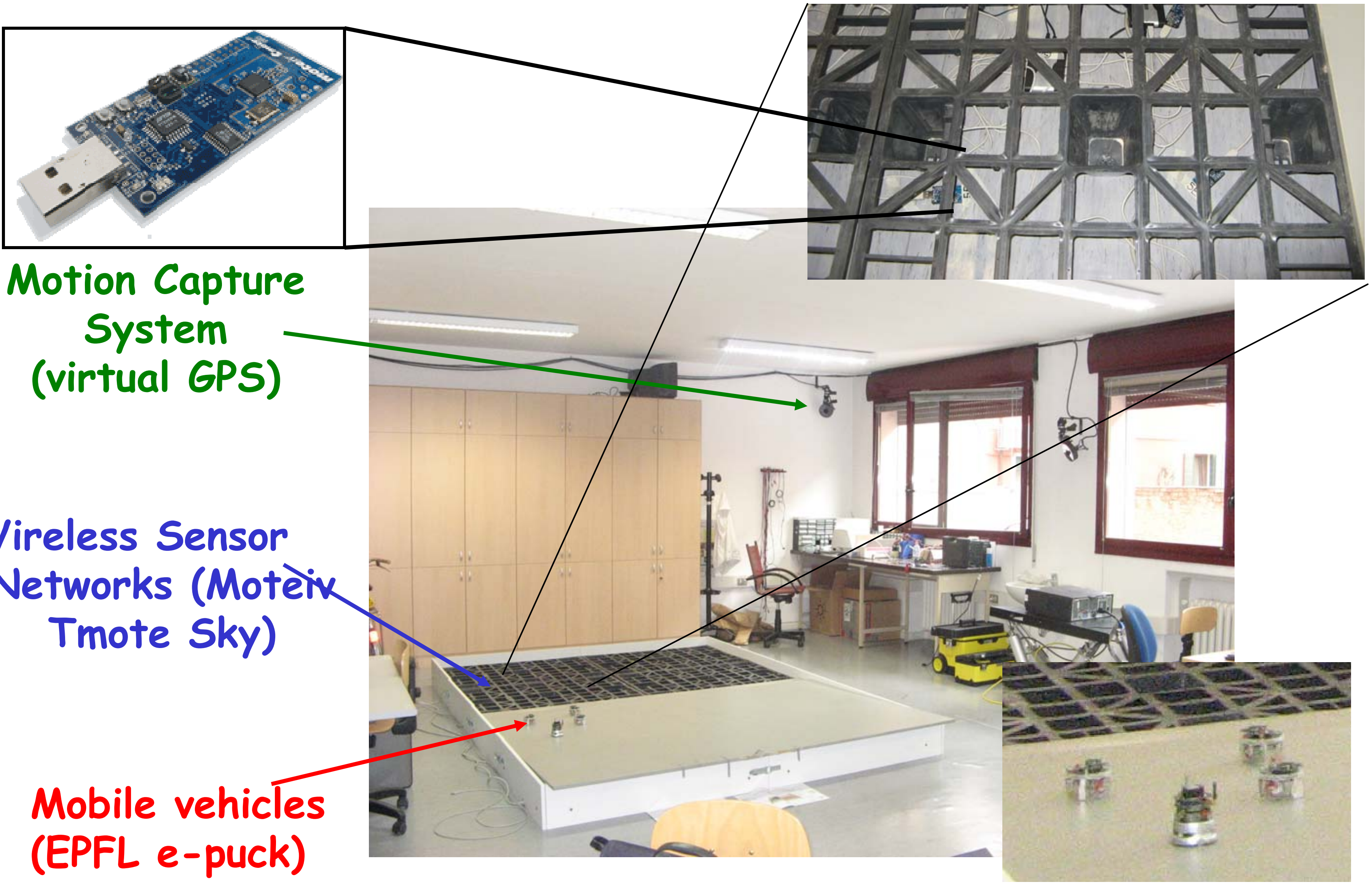
in-node	h_i		
j	η_{ij}	τ_{ij}^{old}	τ_{jj}^{old}
k	η_{ik}	τ_{ik}^{old}	τ_{kk}^{old}
⋮			

τ_i^*	$\hat{\alpha}_i$	$\hat{o}_i^*$

τ_i

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

The testbed

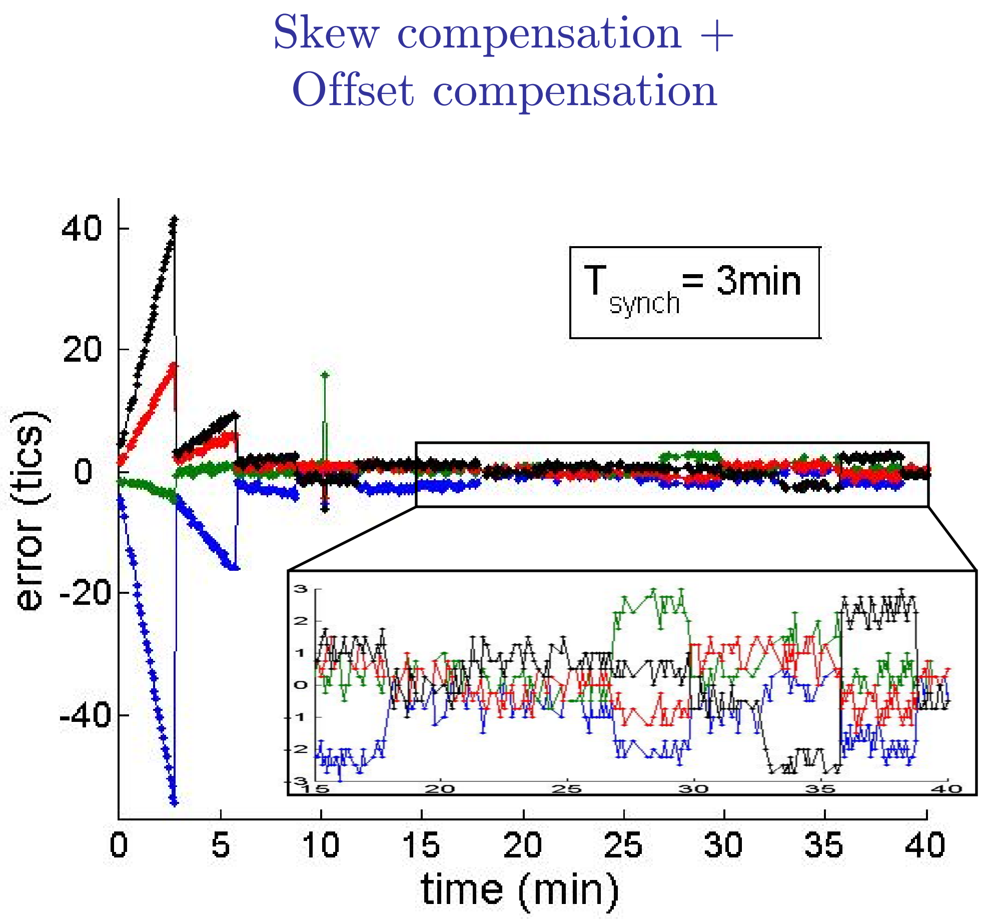


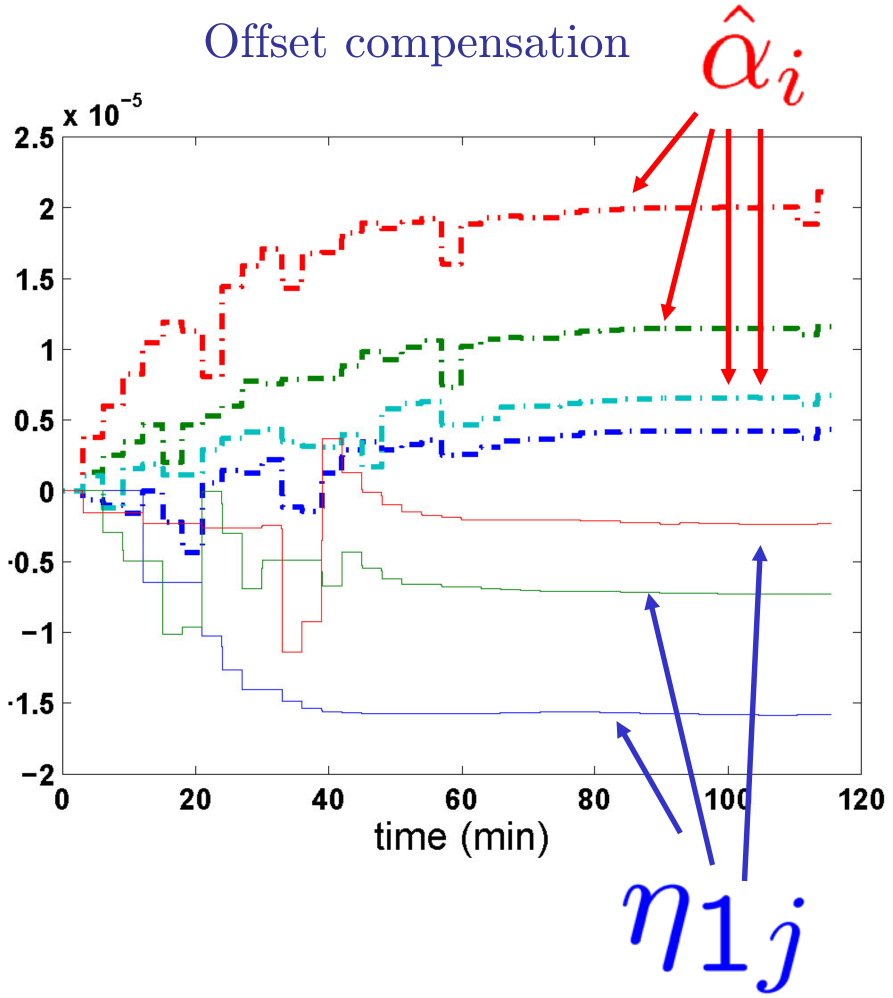
Motion Capture System (virtual GPS)

Wireless Sensor Networks (Moteiv Tmote Sky)

Mobile vehicles (EPFL e-puck)

Experimental results (1)

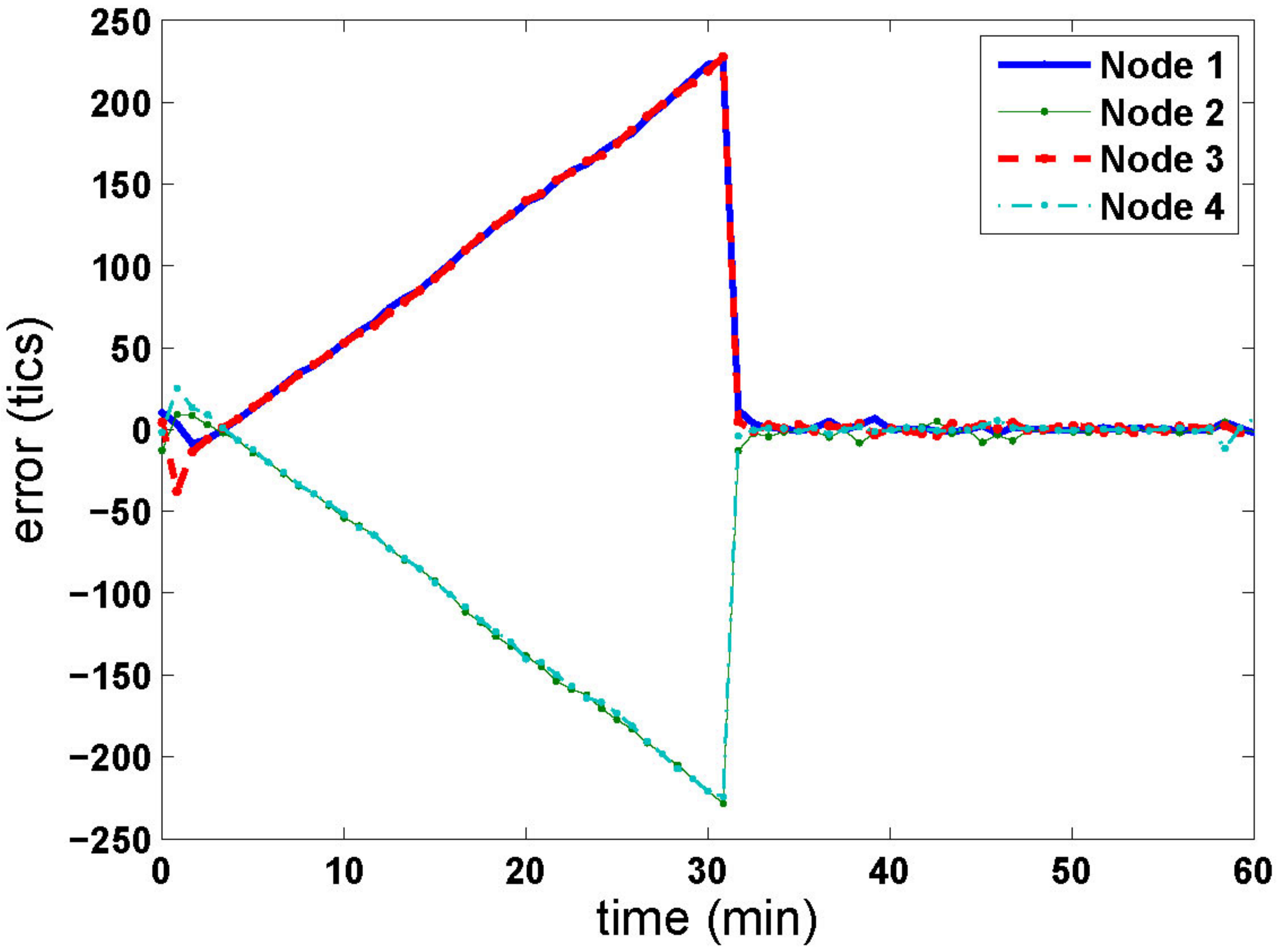




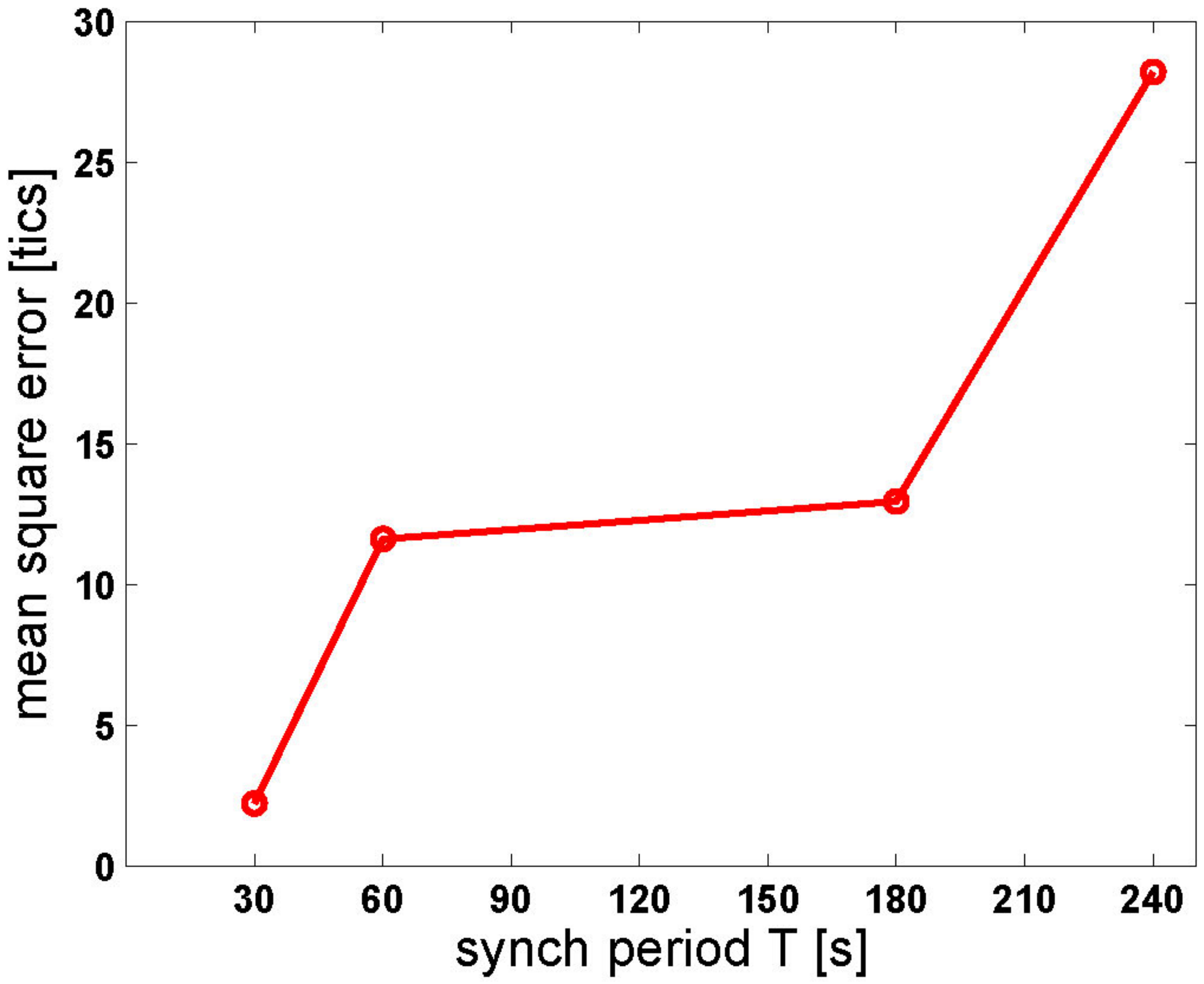
4 Nodes
Synch. period = 3min
1 tic = 30μs (32kHz clock)

DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

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

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