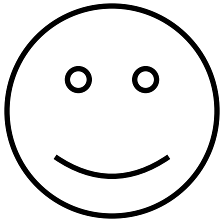
LiMoSense –
Live Monitoring in Dynamic Sensor Networks

Ittay Eyal, Idit Keidar, Raphi Rom



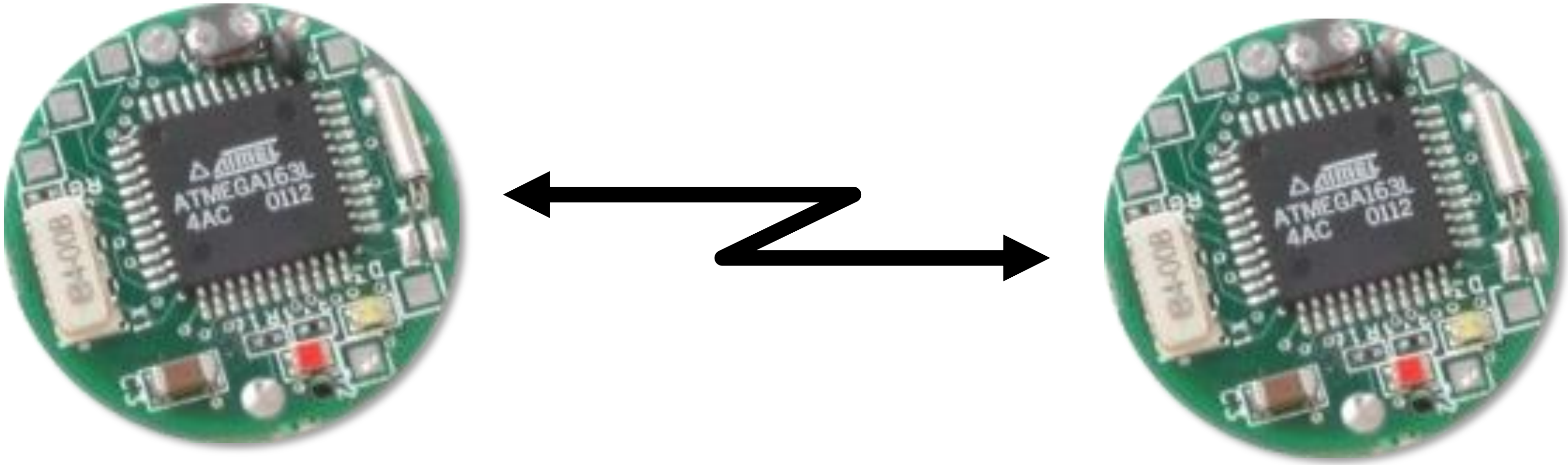
Technion, Israel

Israeli Networking Day. March 2011

Outline

- Background – static average aggregation in sensor networks.
- LiMoSense – Live and robust.
- Correctness.
- Convergence.
- Dynamic behavior.

Sensors



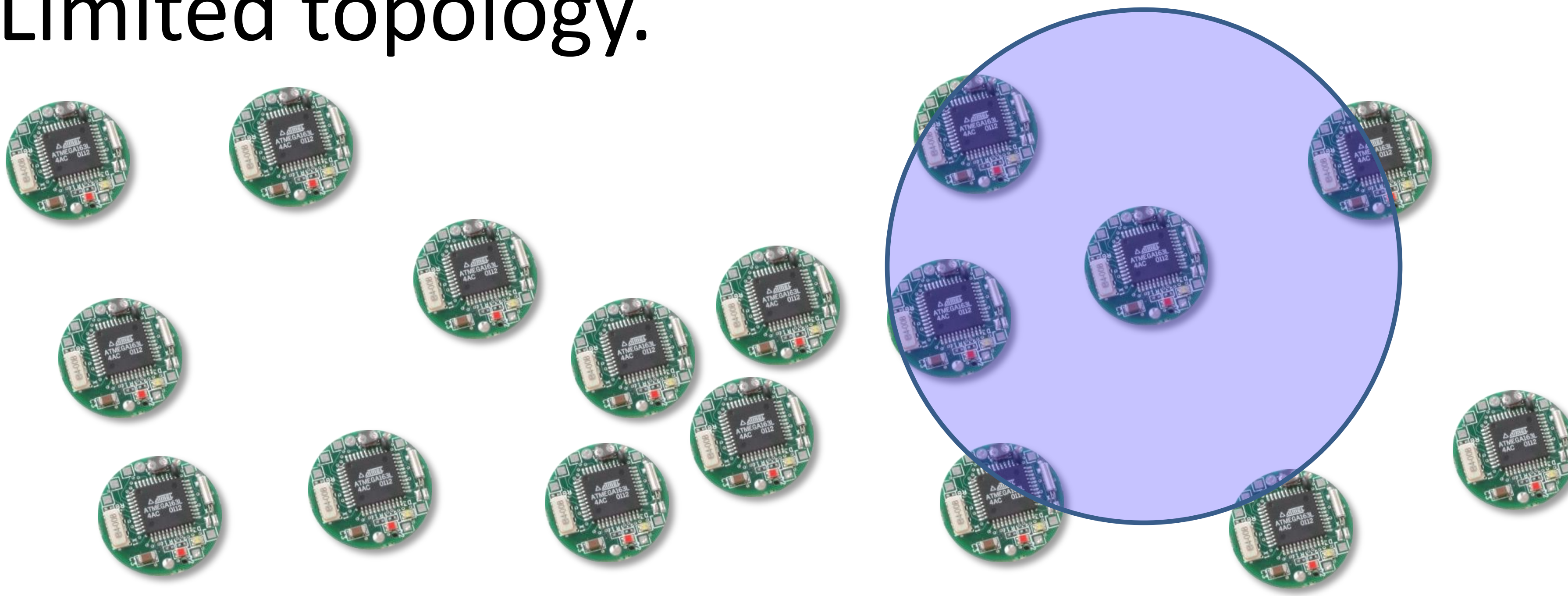
Read values and communicate.

Light-weight, little energy, error prone.

Sensor Network

Many sensors (at least thousands).

Limited topology.



Similar scenario in cloud monitoring

Average Aggregation

Target: Average of read values.

Reason:

- Environmental monitoring.
- Cloud computing load monitoring.

Challenge: Cannot collect the values.

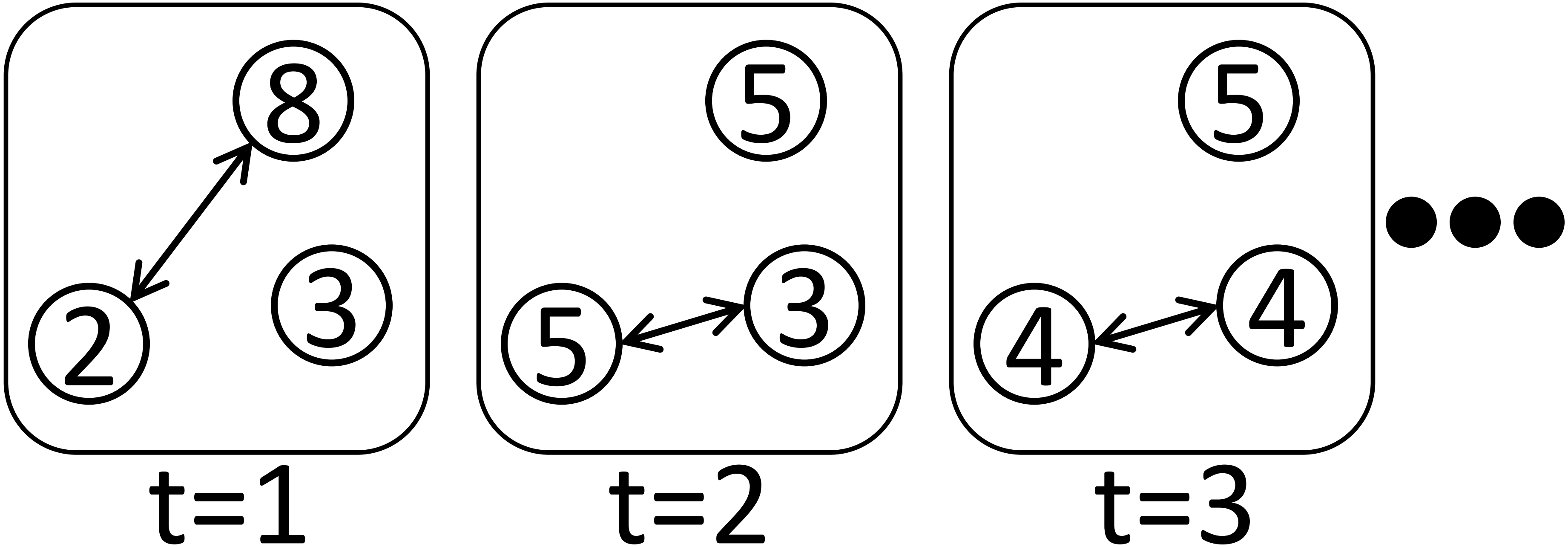
Solution: In-network aggregation.

Hierarchical solution?

Not robust.

Static error-free aggregation [1,2]

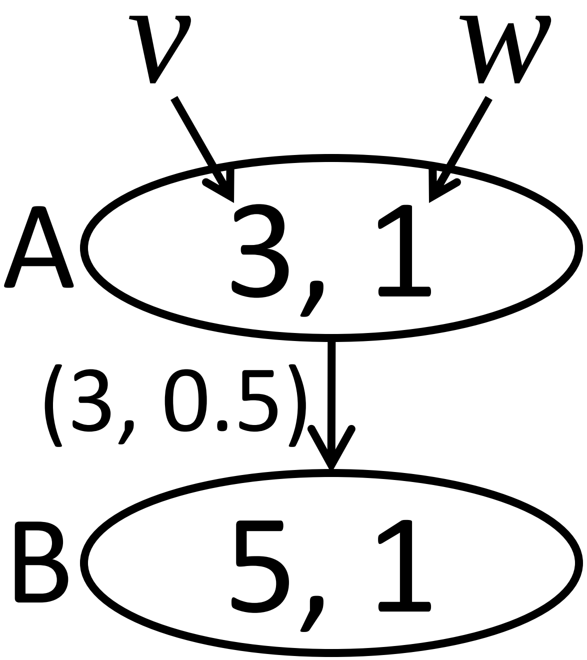
Bidirectional communication gossip



1. D. Kempe, A. Dobra, and J. Gehrke. *Gossip-based computation of aggregate information*. In FOCS, 2003.
2. S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. *Synopsis diffusion for robust aggregation in sensor networks*. In SenSys, 2004.

Static error-free aggregation [1,2]

Unidirectional communication gossip



Send half:

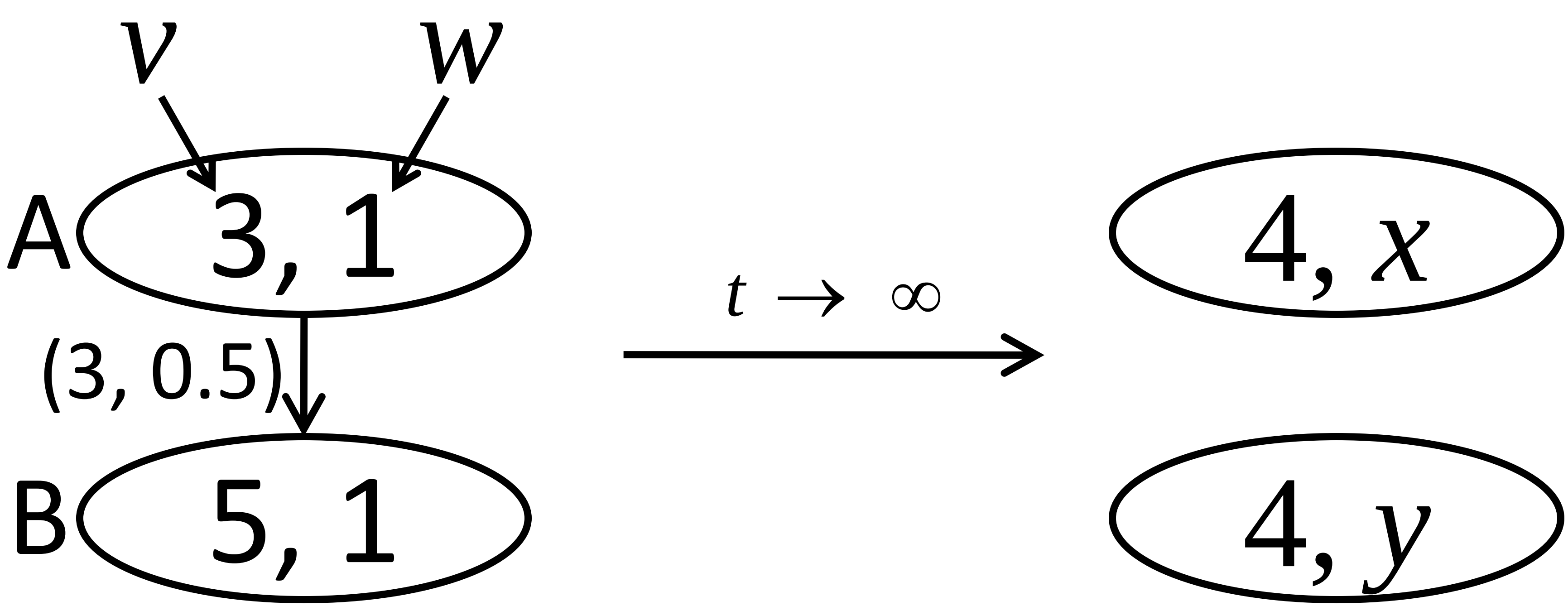
$$\left(v_A, \tfrac{1}{2} w_A \right)$$

$$v_B = \frac{w_B v_B + w_{\text{in}} v_{\text{in}}}{w_B + w_{\text{in}}}$$

1. D. Kempe, A. Dobra, and J. Gehrke. *Gossip-based computation of aggregate information*. In FOCS, 2003.
2. S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. *Synopsis diffusion for robust aggregation in sensor networks*. In SenSys, 2004.

Static error-free aggregation [1,2]

Unidirectional communication gossip



1. D. Kempe, A. Dobra, and J. Gehrke. *Gossip-based computation of aggregate information*. In FOCS, 2003.
2. S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson. *Synopsis diffusion for robust aggregation in sensor networks*. In SenSys, 2004.

Static error-free aggregation

Realistic?

- Monitoring → Dynamic input! (restart?)
- Cheap sensors → Crashes.
- Limited battery →

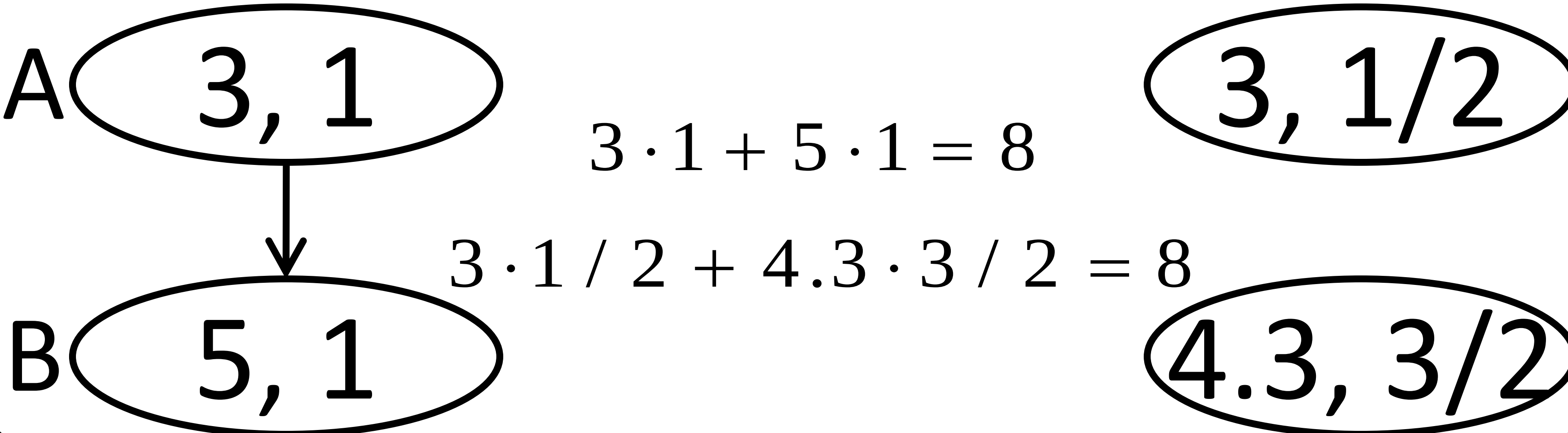
Link failure and message loss

LiMoSense –
Live Monitoring in Dynamic Sensor Networks

Live error-free aggregation

Observation: **Invariant**:

read sum = Weighted sum



$$3 \cdot 1 + 5 \cdot 1 = 8$$

$$3 \cdot 1 / 2 + 4.3 \cdot 3 / 2 = 8$$

Live error-free aggregation

Observation: **Invariant**:

read sum = Weighted sum

Goal: Maintain the invariant after input changes.

Live error-free aggregation

Each node stores:

1. Current weighted estimation.
2. Previously read value.

On read change: update weighted estimation to fix invariant.

$$est_i \leftarrow est_i + \frac{1}{w_i}\left(\mathrm{newRead}_i - \mathrm{prevRead}_i\right)$$

Weight unchanged.

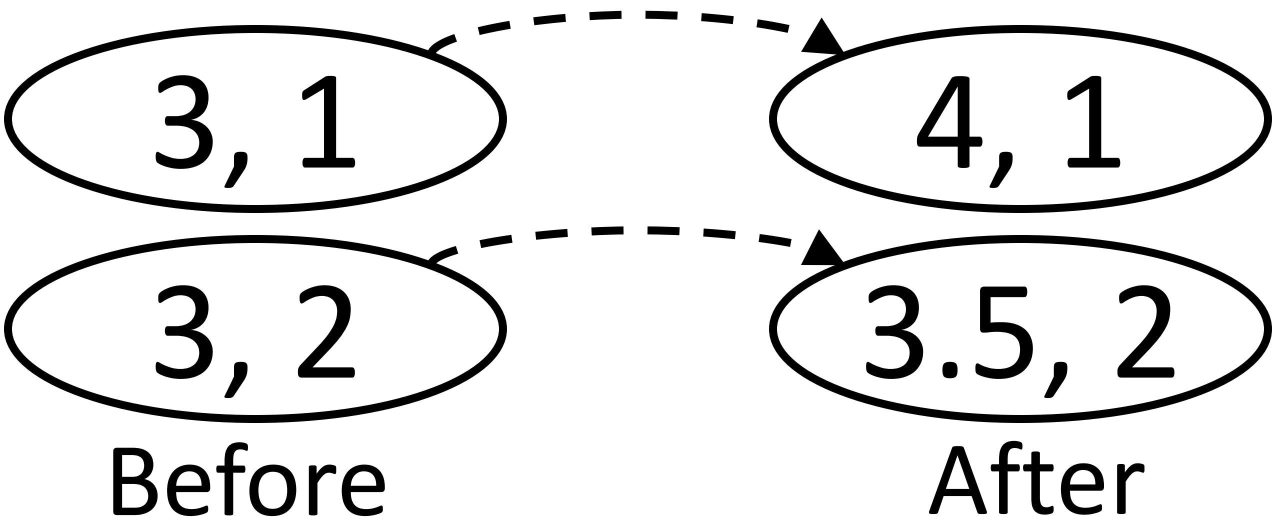
Live error-free aggregation

$$est_i \leftarrow est_i + \frac{1}{w_i}\left(\mathrm{newRead}_i - \mathrm{prevRead}_i\right)$$

Example: read value 0 → 1

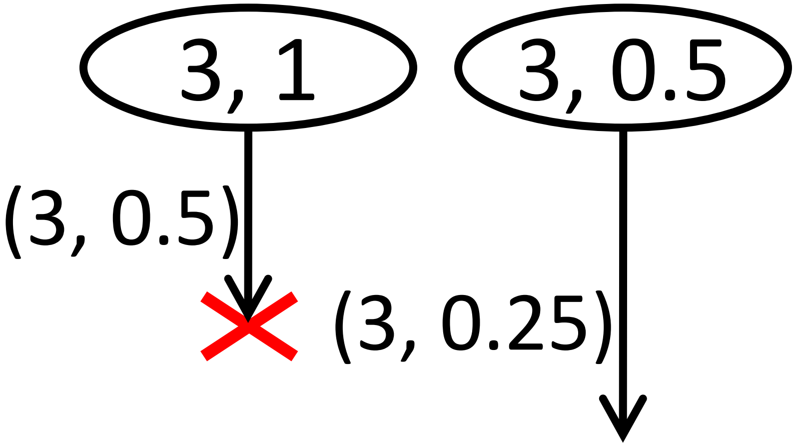
Case 1:

Case 2:



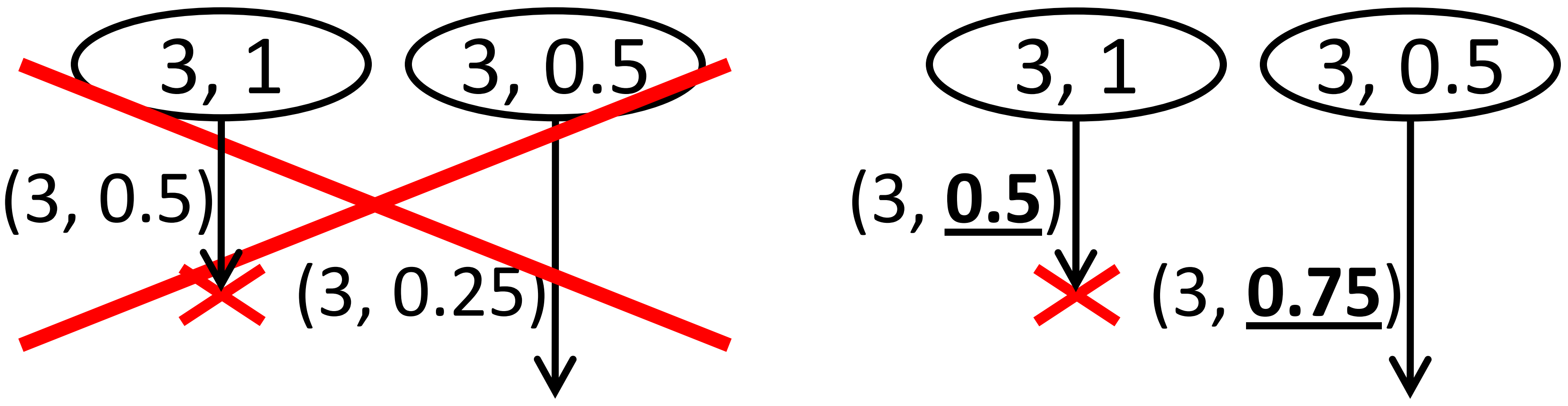
Live robust aggregation

Lost messages → lost weight → broken Invariant:



Live robust aggregation

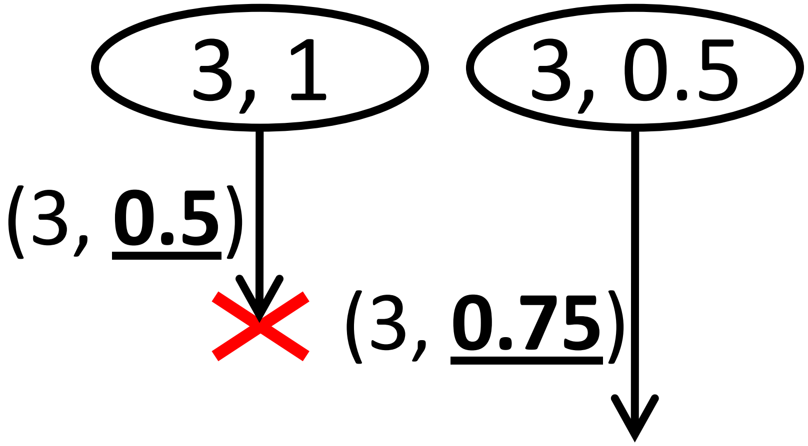
Solution: Send summary, not diff:



Live robust aggregation

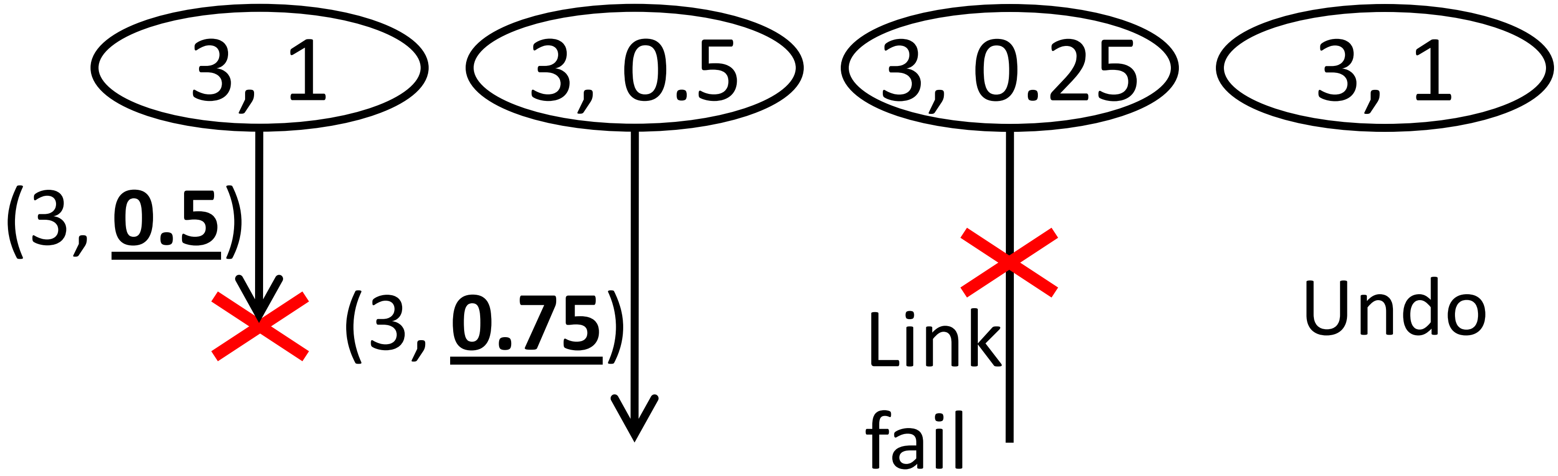
Solution: Send summary, not diff:

- Lost message:
  Fix on next one.
- Failed link:
  Transfer undo.



Live robust aggregation

Solution: Send summary, not diff:



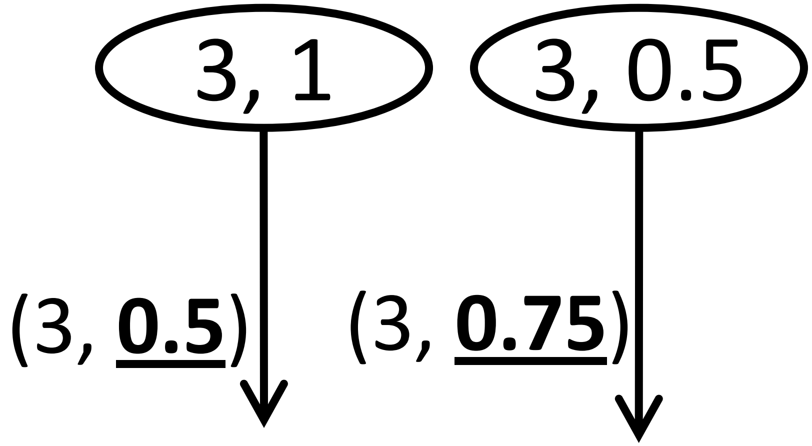
Live robust aggregation

Challenge: weight → infinity

Solution:

Hybrid push-pull:

Ask neighbor to send back inverse.



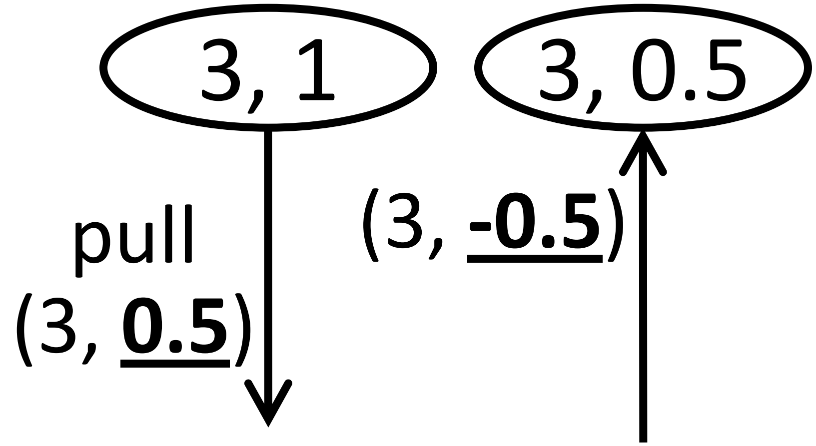
Live robust aggregation

Challenge: weight → infinity

Solution:

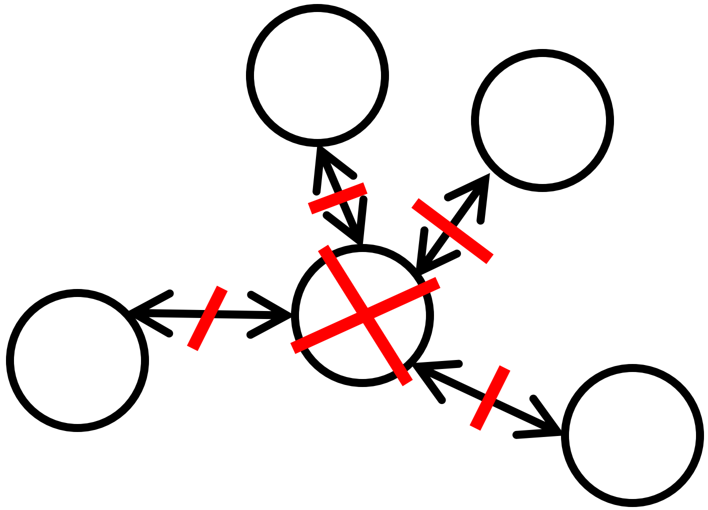
Hybrid push-pull:

Ask neighbor to send back inverse.



Live robust aggregation

Crashed node → lost links.



Correctness

Correctness - Safety

Theorem 1: The invariant always holds.

$$\langle R^t, n\rangle = \bigoplus_{i=1}^{n} \langle r_i^t, 1\rangle =$$

$$= \bigoplus_{i=1}^{n} \left(\langle est_i^t, w_i^t\rangle \oplus \bigoplus_{j \in neighbors_i^t} \left(sent_i^t(j) \ominus received_i^t(j)\right) \right) = \langle E^t, n\rangle$$

1. On message send/receive.
2. After value change.
3. After add/remove neighbor.
4. After node removal/addition.

Correctness - Liveness

Theorem 2: After GST, all estimations converge to the average.

1. Quantization constant and fairness.
2. Value propagation.
3. Convergence.

Correctness - Liveness

1. Quantization constant and fairness.

Weight is transferred in multiples of q.
Note – This does not effect accuracy.

Each node eventually succeeds to send a *push* message to all of its neighbors.

Correctness - Liveness

2. Value propagation.

Lemma:
For any time $t >= \mathrm{GST}$ and node i, there exists a time $t' > t$ after which every node j has a component of i with a weight larger than some bound (the bound is dependent on n and q)

Correctness - Liveness

3. Convergence.

Define a series $\mathrm{GST} = t_0, t_1, t_2, \ldots$
Where at t_i each node has a bounded-from-below portion from each node at $t_{i\text{-}1}$.
At each t_i, the largest error is smaller than in $t_{i\text{-}1}$, because it's mixed with other values.

Convergence Rate

Convergence

Static convergence rate (exchange gossip):

- Static input (after GST).
- Dense topology.
- Synchronous uniform runs.

Convergence

Static convergence rate (exchange gossip):

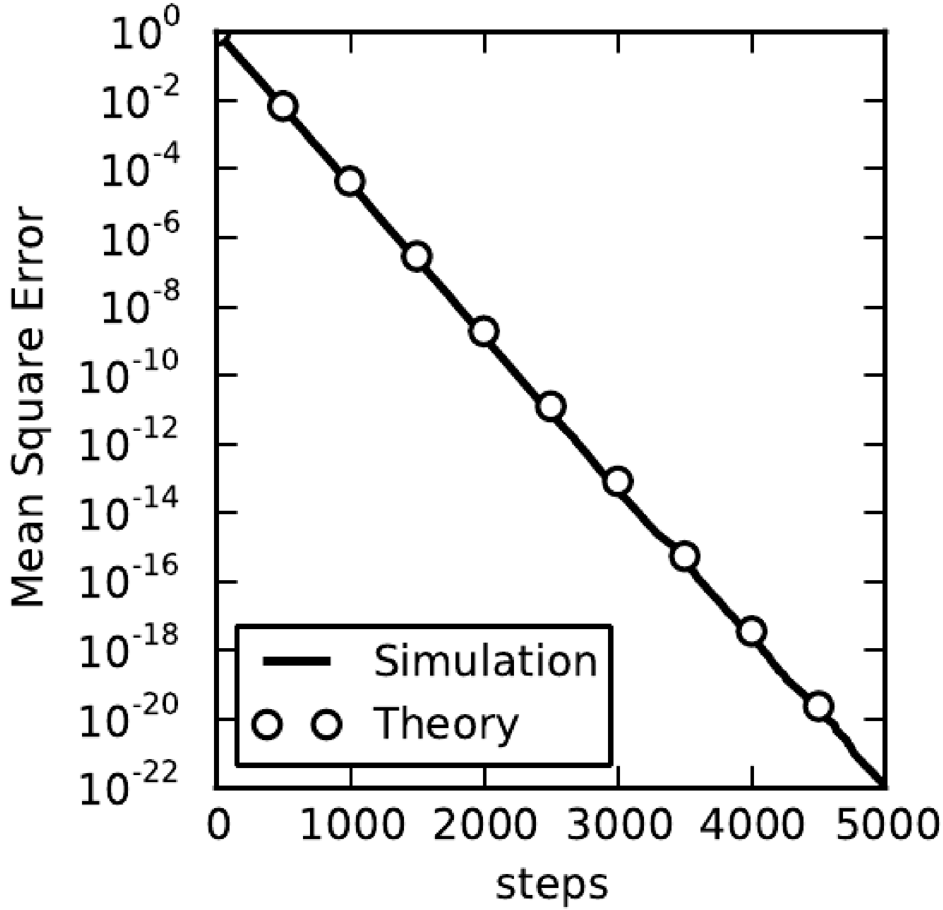
Assumption:

normal distribution of estimations.

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \underbrace{\frac{1}{2\pi} e^{\frac{-x^2-y^2}{2}}}_{\substack{\text{Probability density}\\ \text{of choosing } x \text{ and}\\ y}} \underbrace{\left(1 - \frac{1}{n}x^2 - \frac{1}{n}y^2 + \frac{2}{n}\left(\frac{x+y}{2}\right)^2\right)}_{\substack{\text{Variance in next step for the given } x,\\ y}} = 1 - \frac{1}{n}$$

Convergence

Static convergence rate (exchange gossip):

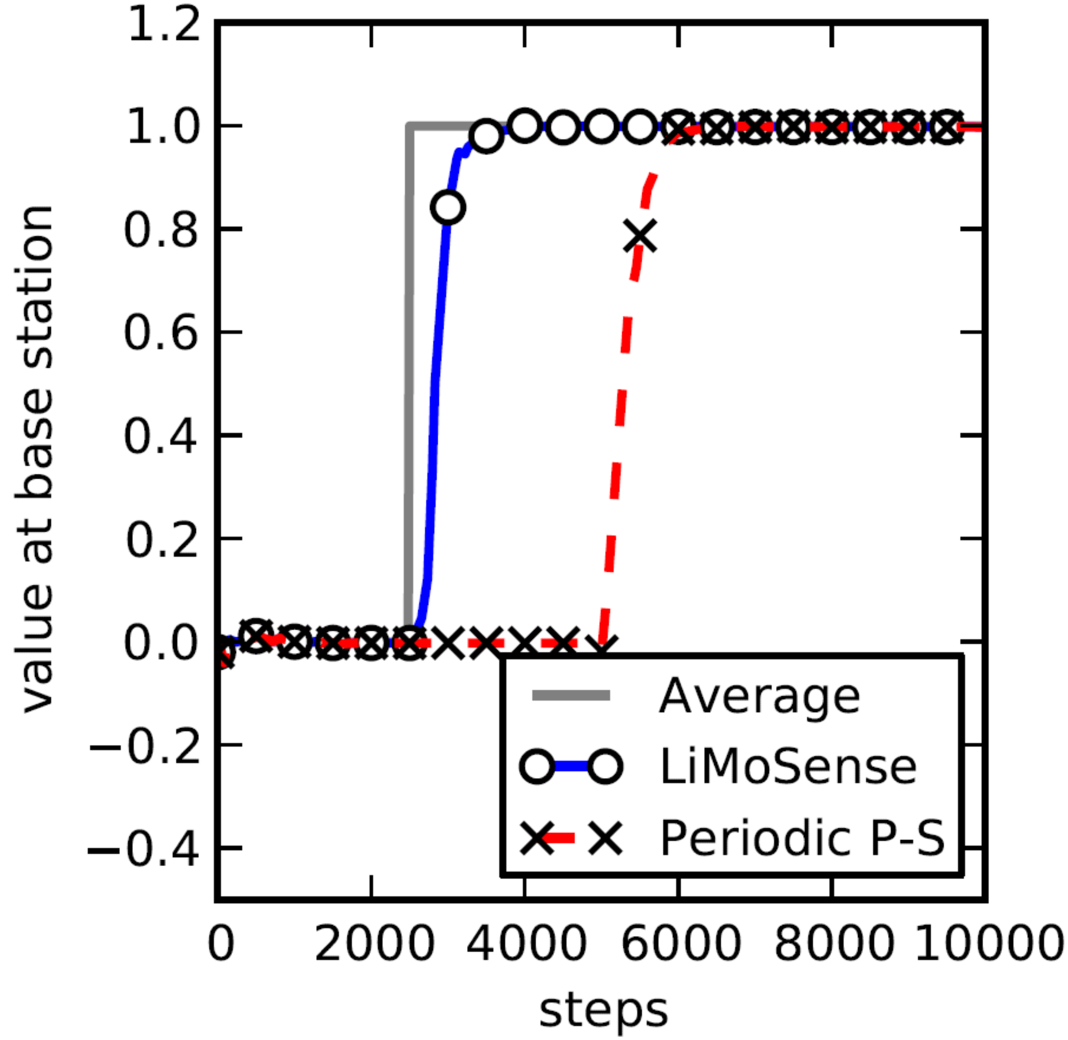


Dynamic Behavior

Step Function

100 nodes.
Standard Normal distribution

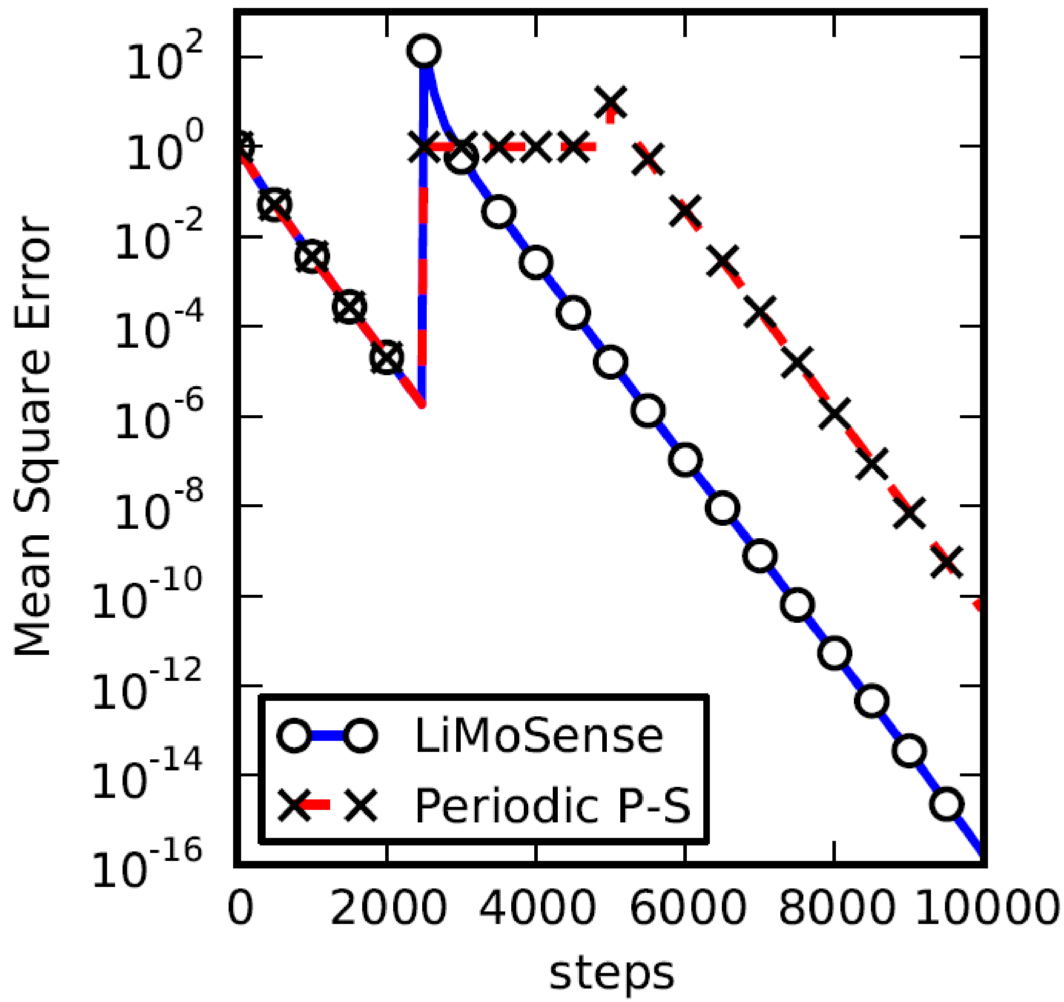
10 nodes change Values (+10)



Step Function

100 nodes.
Standard Normal distribution

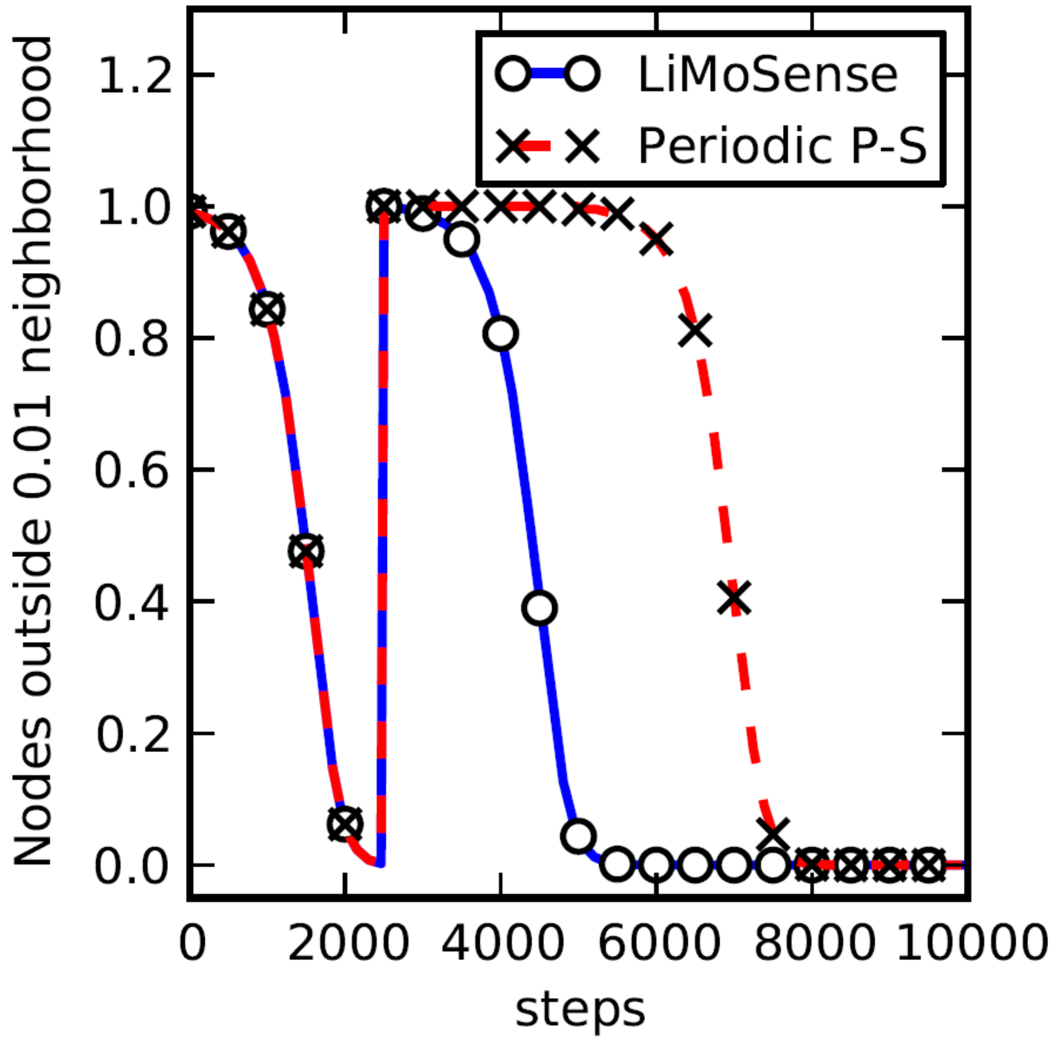
10 nodes change Values (+10)



Step Function

100 nodes.
Standard Normal distribution

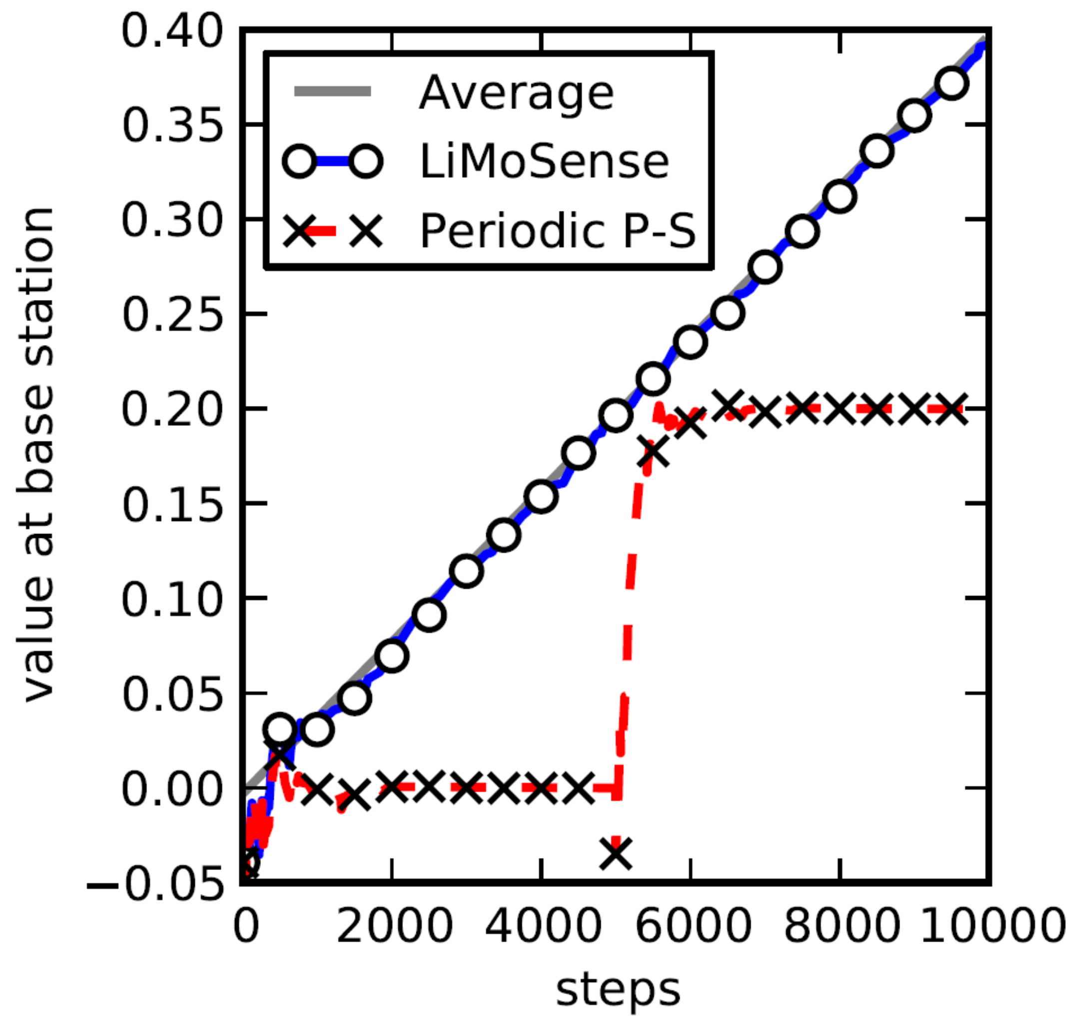
10 nodes change Values (+10)



Creeping Value Change

100 nodes.
Standard Normal distribution

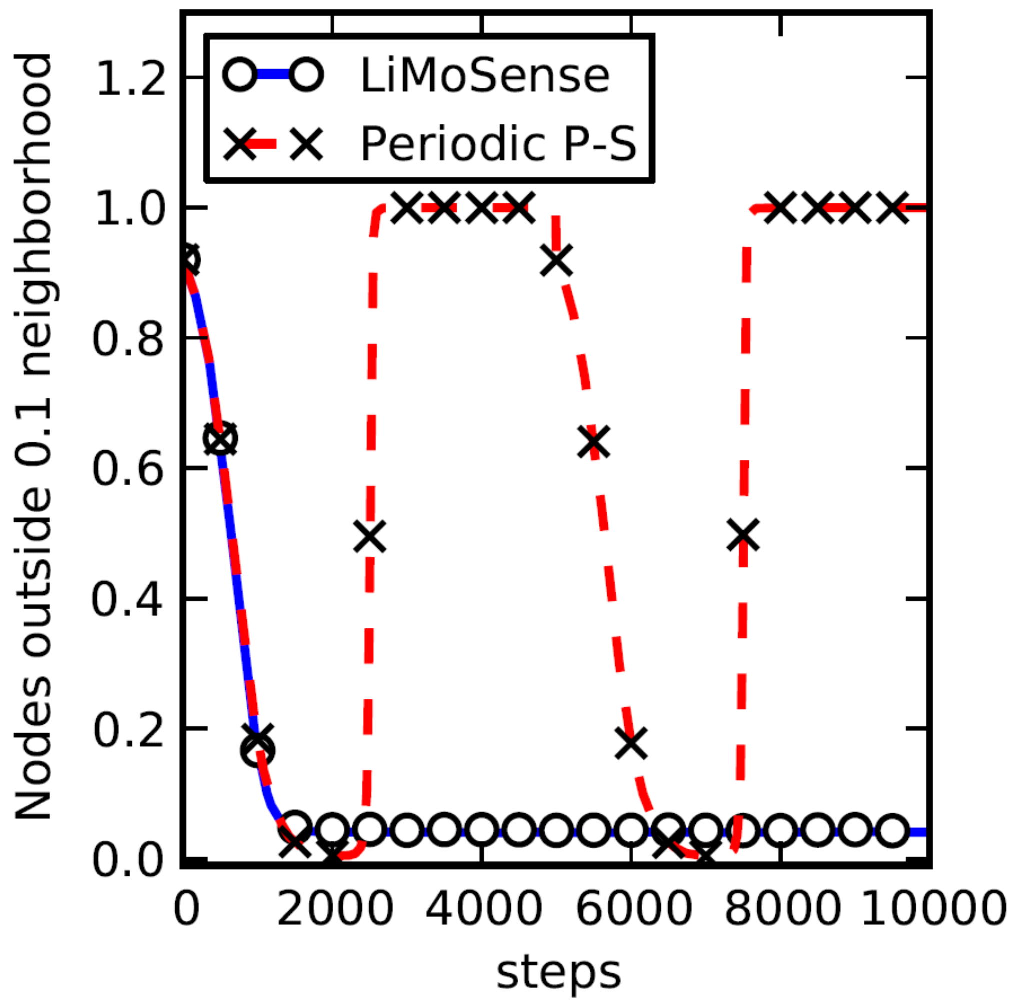
Every 10 steps,
10 nodes change Values (+0.01)



Creeping Value Change

100 nodes.
Standard Normal distribution

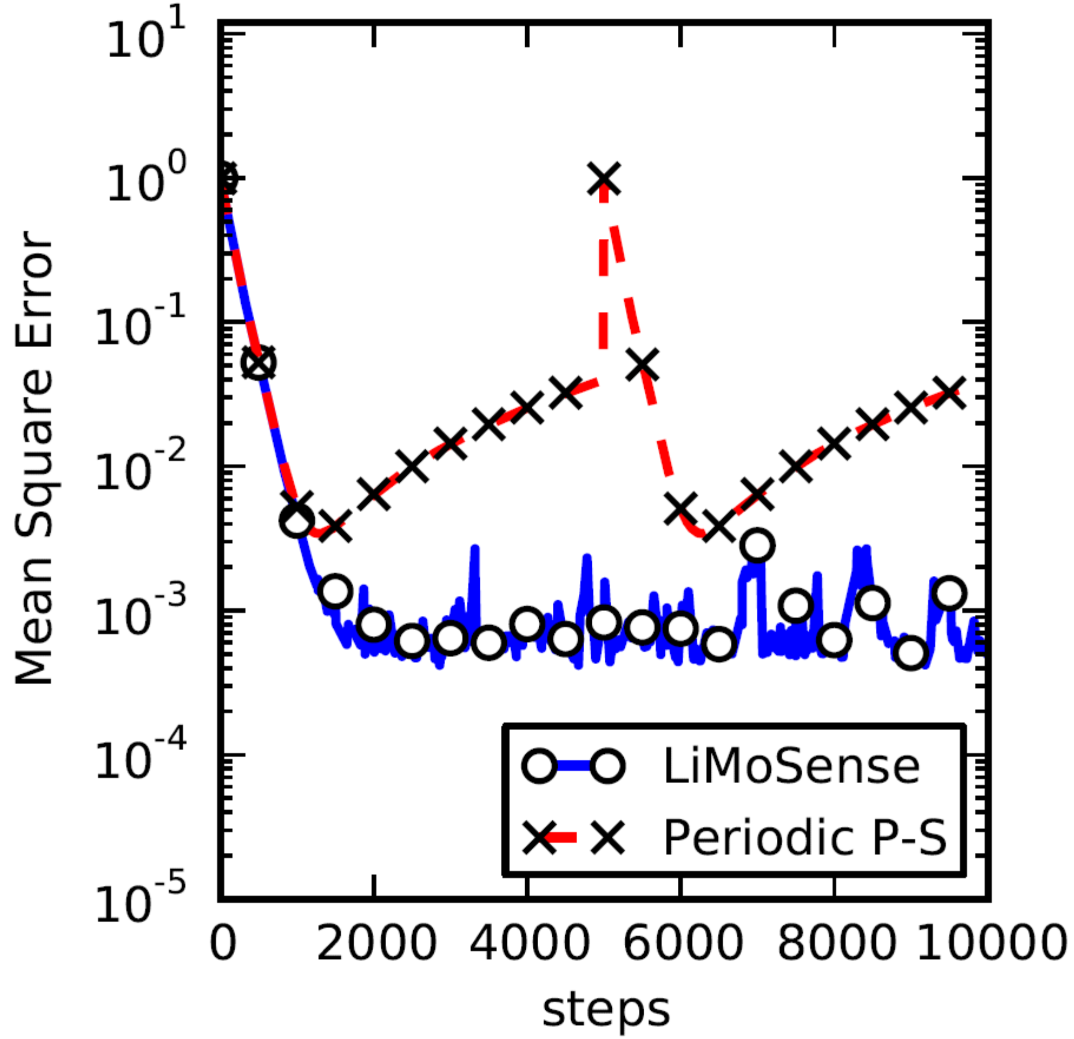
Every 10 steps,
10 nodes change
Values (+0.01)



Creeping Value Change

100 nodes.
Standard Normal distribution

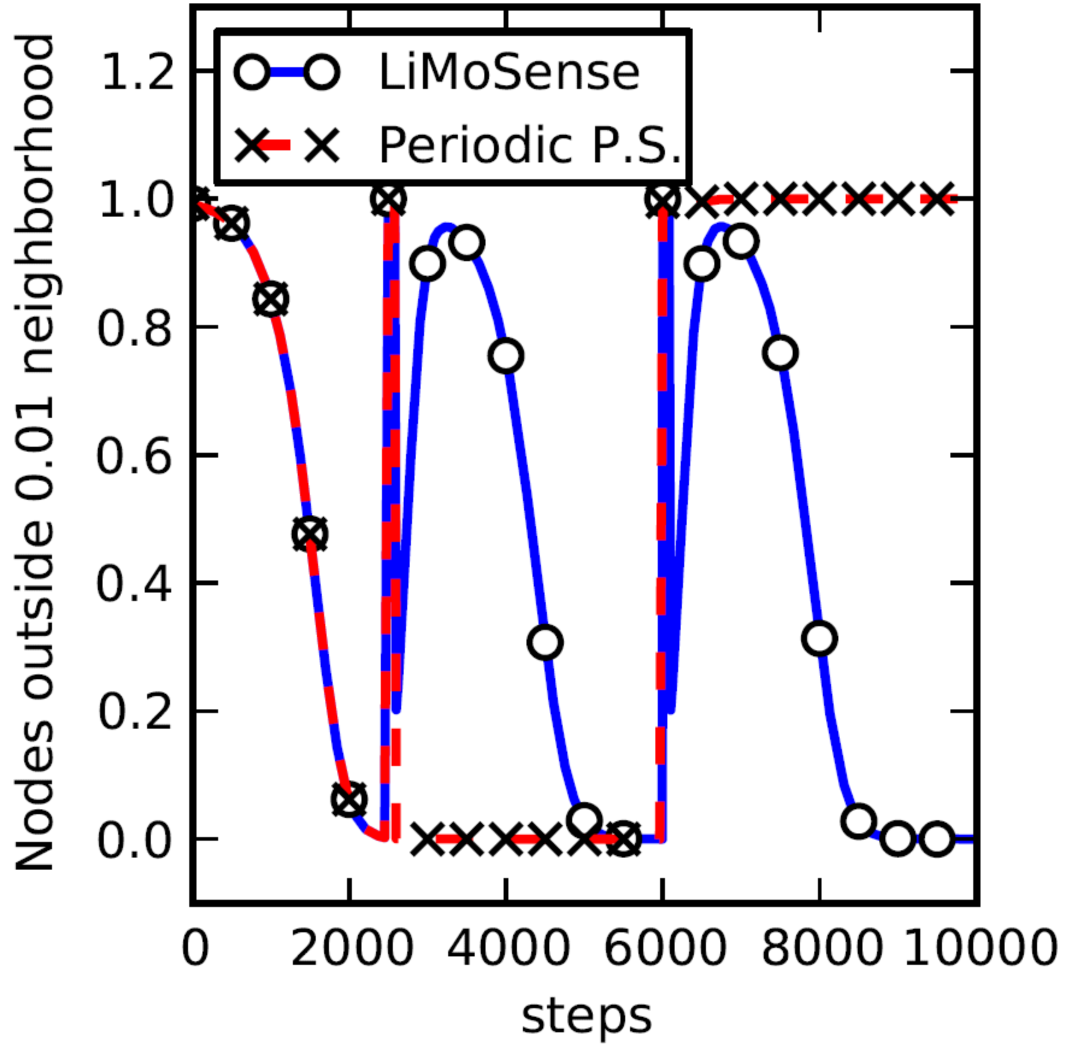
Every 10 steps,
10 nodes change
Values (+0.01)



Response to Step Function

100 nodes.
Standard Normal
distribution

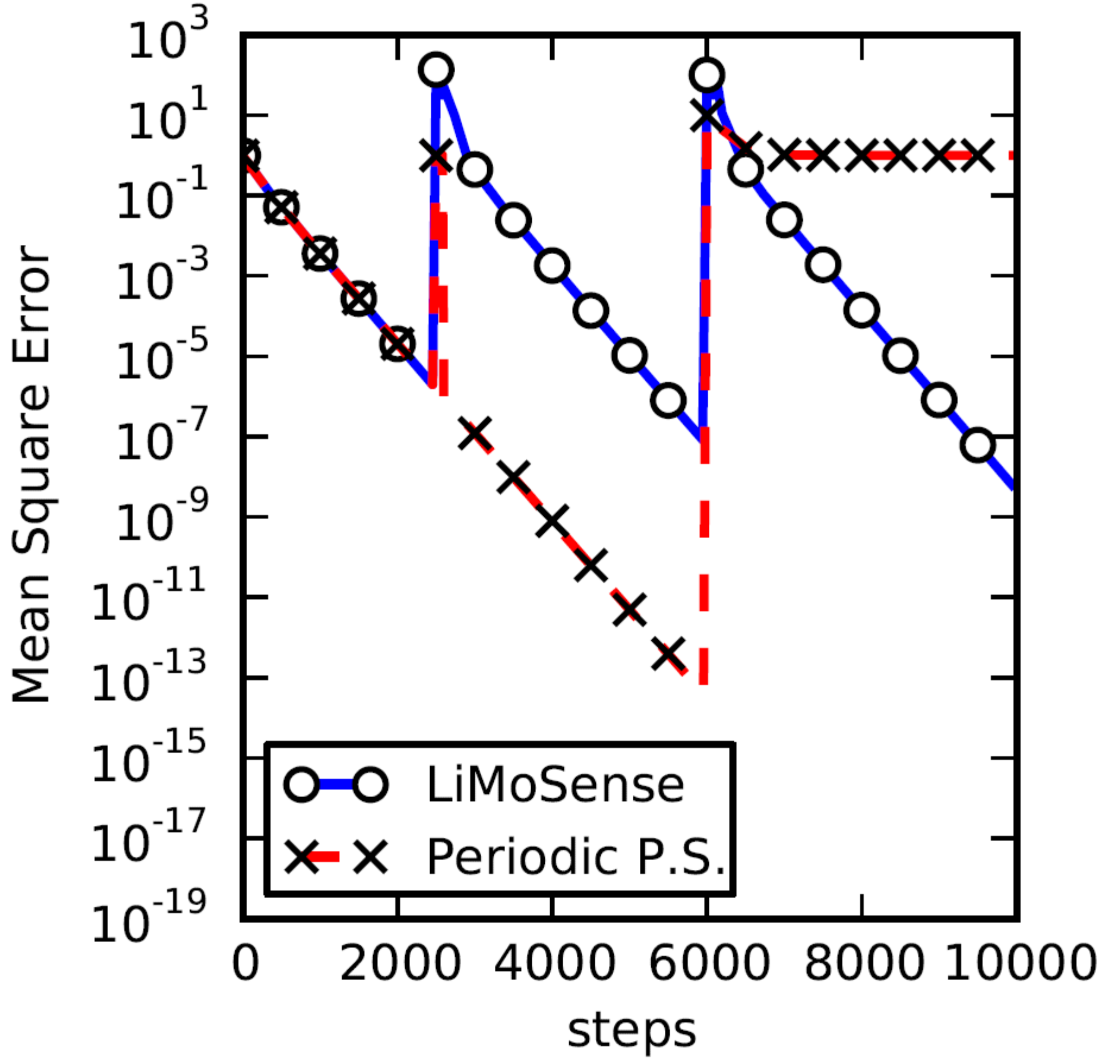
10 nodes change
Values (+10) for
100 steps



Response to Step Function

100 nodes.
Standard Normal distribution

10 nodes change Values (+10) for 100 steps

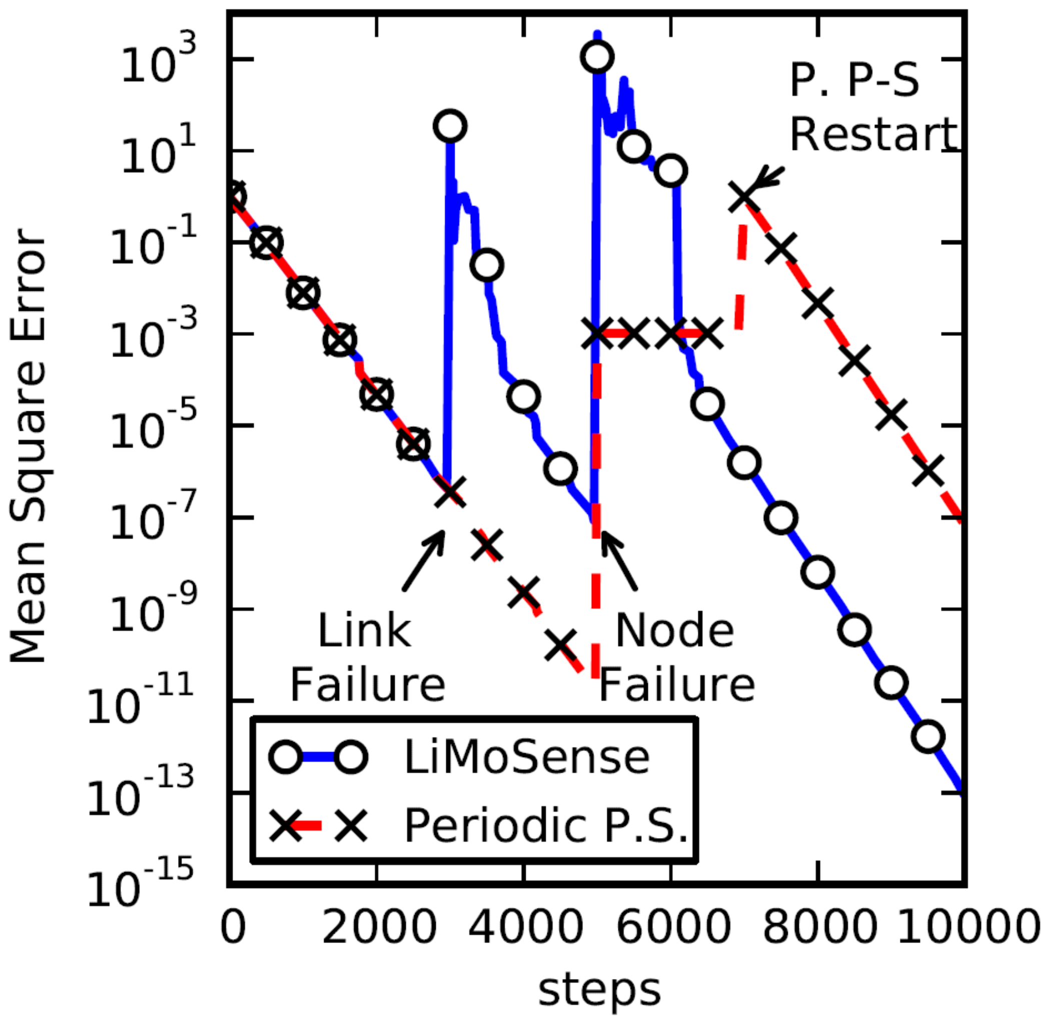


Dynamic Network

Disc Graph

t=2500: 10 nodes range decay, 7 lost links

t=5000: Node crash.



Summary

- LiMoSense – Live Average Monitoring in error prone dynamic sensor networks.
- Live: aggregate dynamic data reads.
- Fault tolerant: Message loss, link failure and node crash.
- Correctness in dynamic asynchronous settings.
- Exponential convergence after GST.
- Quick dynamic behavior.

Ittay Eyal, Idit Keidar, Raphael Rom. *LiMoSense - Live Monitoring in Dynamic Sensor Networks*, Technion technical report CCIT #786 March 2011EE.

Good Questions

Convergence

Static convergence rate (push gossip):

$$\int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \left(\underbrace{\frac{1}{2\pi} e^{\frac{-x^2-y^2}{2}}}_{\substack{\text{Probability density}\\ \text{of choosing } x \text{ and}\\ y}} \cdot \underbrace{\left(1 - \frac{1}{n}x^2 - \frac{1}{n}y^2 + \frac{1}{2n}x^2 + \frac{3}{2n}\left(\frac{\frac{1}{2}x + y}{3/2}\right)^2\right)}_{\substack{\text{Variance in next step for the given } x,\\ y}} \right) = 1 - \frac{2}{3n}$$