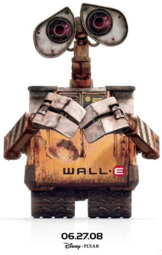
Discrete consensus in wireless networks

Anand D. Sarwate and Tara Javidi

Information Theory and Applications Center
University of California, San Diego

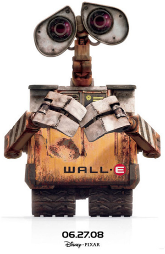
12 May 2010

Applications for consensus algorithms



http://www.flickr.com/photos/skreuzer/354316053/

Applications for consensus algorithms



Consensus is a fundamental task in distributed systems:

http://www.flickr.com/photos/skreuzer/354316053/

Applications for consensus algorithms

Consensus is a fundamental task in distributed systems:

- Coordination and control in robotic networks

http://www.flickr.com/photos/skreuzer/354316053/

Applications for consensus algorithms

Consensus is a fundamental task in distributed systems:

- Coordination and control in robotic networks
- Load balancing and calibration

http://www.flickr.com/photos/skreuzer/354316053/

Applications for consensus algorithms

Consensus is a fundamental task in distributed systems:

- Coordination and control in robotic networks
- Load balancing and calibration
- Primitive for distributed computation

`http://www.flickr.com/photos/skreuzer/354316053/`

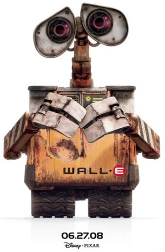
Consensus protocols : goals

http://www.flickr.com/photos/skreuzer/354316053/

Consensus protocols : goals

http://www.flickr.com/photos/skreuzer/354316053/

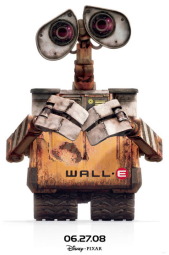
Consensus protocols : goals



- Discrete computation and communication : reality is digital.

http://www.flickr.com/photos/skreuzer/354316053/

Consensus protocols : goals



- Discrete computation and communication : reality is digital.
- Low complexity : can piggyback in existing packet structure.

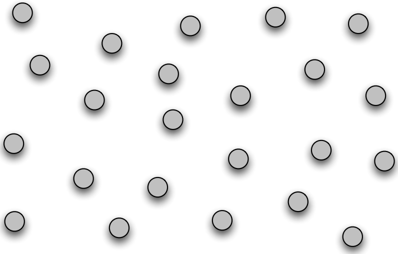
http://www.flickr.com/photos/skreuzer/354316053/

Consensus protocols : goals

- Discrete computation and communication : reality is digital.
- Low complexity : can piggyback in existing packet structure.
- Fast : should not pay “too much” for distributed implementation.

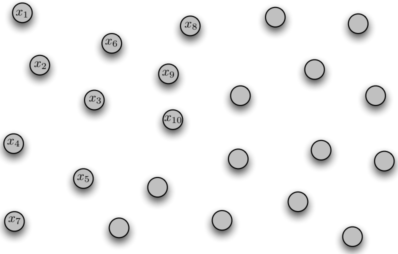
http://www.flickr.com/photos/skreuzer/354316053/

The consensus problem



At a high level, consensus is distributed averaging:

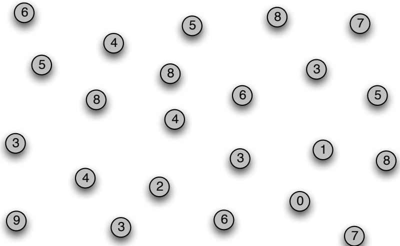
The consensus problem



At a high level, consensus is distributed averaging:

- n nodes in a network with initial values $x_i(0)$ for $i = 1, 2, \ldots n$

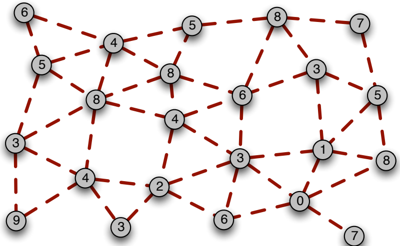
The consensus problem



At a high level, consensus is distributed averaging:

- n nodes in a network with initial values $x_i(0)$ for $i = 1, 2, \ldots n$

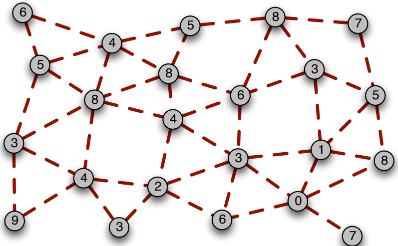
The consensus problem



At a high level, consensus is distributed averaging:

- n nodes in a network with initial values $x_i(0)$ for $i = 1, 2, \ldots n$

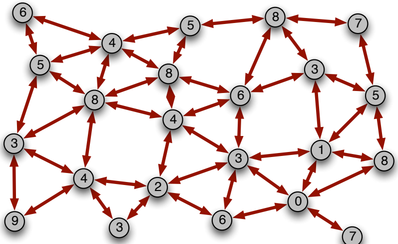
The consensus problem



At a high level, consensus is distributed averaging:

- n nodes in a network with initial values $x_i(0)$ for $i = 1, 2, \ldots n$
- Pass messages (either synchronously or asynchronously)

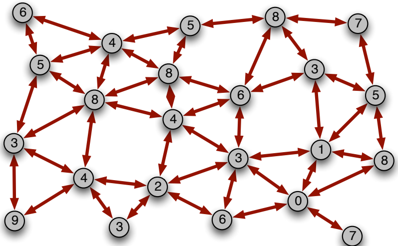
The consensus problem



At a high level, consensus is distributed averaging:

- n nodes in a network with initial values $x_i(0)$ for $i = 1, 2, \ldots n$
- Pass messages (either synchronously or asynchronously)

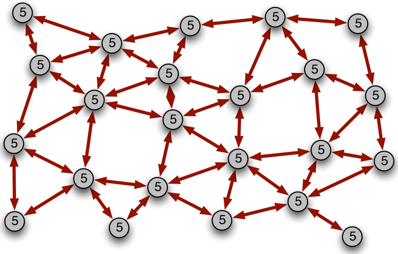
The consensus problem



At a high level, consensus is distributed averaging:

- n nodes in a network with initial values $x_i(0)$ for $i = 1, 2, \ldots n$
- Pass messages (either synchronously or asynchronously)
- Goal is to compute average $x_{\text{ave}} = \frac{1}{n} \sum x_i(0)$

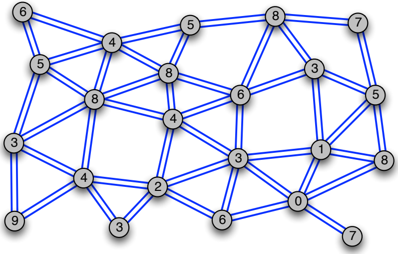
The consensus problem



At a high level, consensus is distributed averaging:

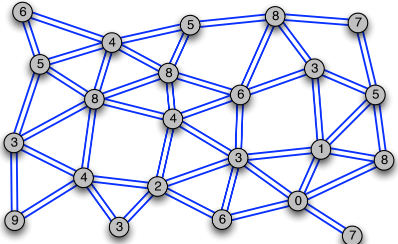
- n nodes in a network with initial values $x_i(0)$ for $i = 1, 2, \ldots n$
- Pass messages (either synchronously or asynchronously)
- Goal is to compute average $x_{\text{ave}} = \frac{1}{n} \sum x_i(0)$

Typical assumptions are unrealistic



Existing work doesn't "look practical":

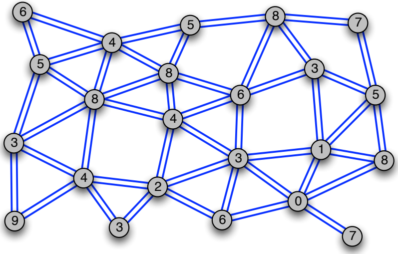
Typical assumptions are unrealistic



Existing work doesn't "look practical":

- Transmit and receive real numbers

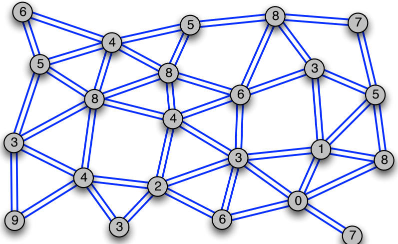
Typical assumptions are unrealistic



Existing work doesn't "look practical":

- Transmit and receive real numbers
- Consensus is the only goal of the network

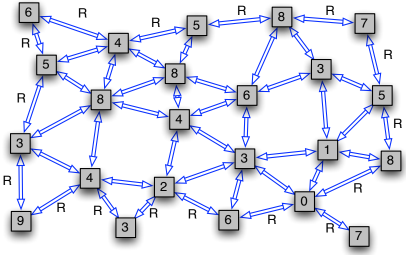
Typical assumptions are unrealistic



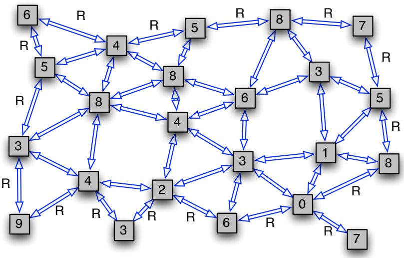
Existing work doesn't "look practical":

- Transmit and receive real numbers
- Consensus is the only goal of the network
- Asymptotics and universality

This talk : quantizing communication and computation

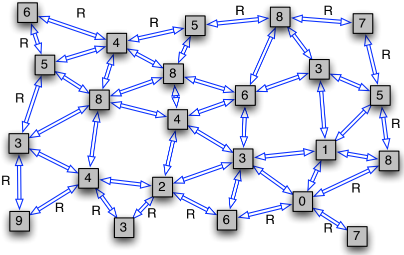


This talk : quantizing communication and computation



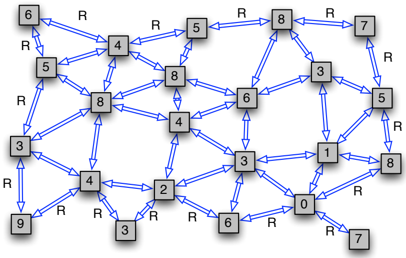
- Discrete computation *and* communication works fine.

This talk : quantizing communication and computation



- Discrete computation *and* communication works fine.
- Theoretical guarantees on performance.

This talk : quantizing communication and computation



- Discrete computation *and* communication works fine.
- Theoretical guarantees on performance.
- Possible metric for optimizing parameters.

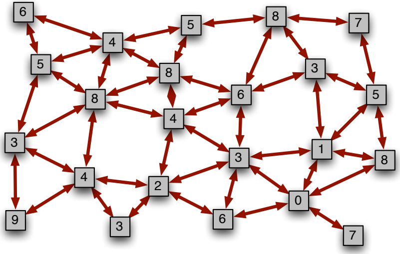
Quantizing communication and computation

Consensus algorithms : De Groot ('74), Chatterjee and Seneta ('77). Tsitsiklis ('84)

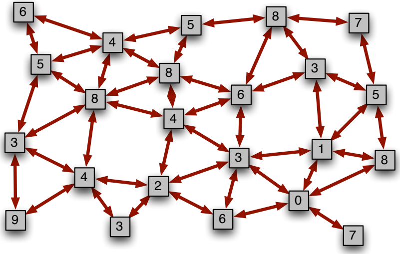
Lots of recent work on quantizing communication:

Xiao et al. '07	noisy messages
Yildiz and Scaglione '08	quantization noise
Aysal et al. '07	probabilistic quantization
Kar and Moura '10	dithered quantization
Nedić et al. '09	universal bounds
Carli et al. '10	adaptive quantization

The model

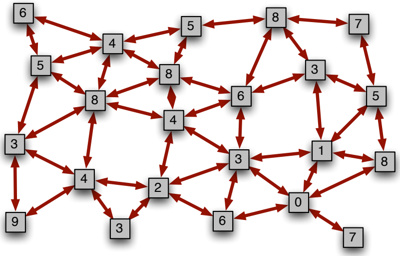


The model



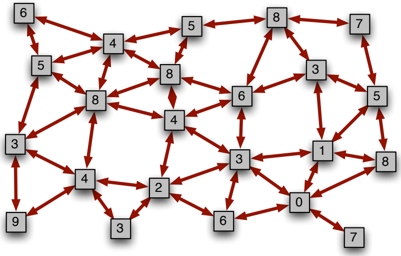
- n nodes in a fixed static graph

The model



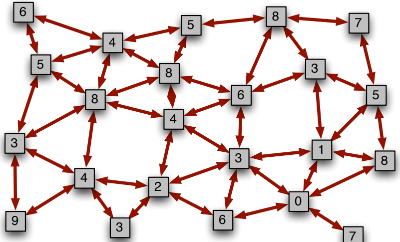
- n nodes in a fixed static graph
- Node i starts with initial values $x_i(0)$ for $i = 1, 2, \ldots n$

The model



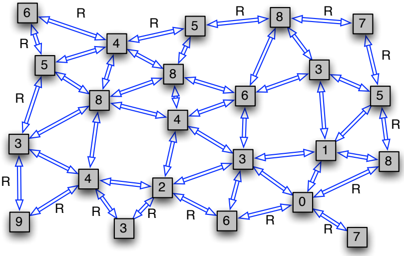
- n nodes in a fixed static graph
- Node i starts with initial values $x_i(0)$ for $i = 1, 2, \ldots n$
- Assume $x_i(0) \in [0, 1]$, uniformly quantized to $R + R_0$ bits

The model

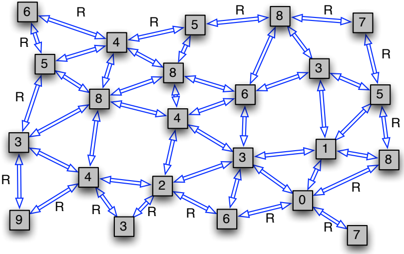


- n nodes in a fixed static graph
- Node i starts with initial values $x_i(0)$ for $i = 1, 2, \ldots n$
- Assume $x_i(0) \in [0, 1]$, uniformly quantized to $R + R_0$ bits
- Goal is to compute $x_{\text{ave}} = \frac{1}{n} \sum x_i(0)$

Quantized communication

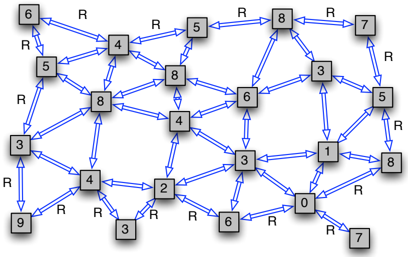


Quantized communication



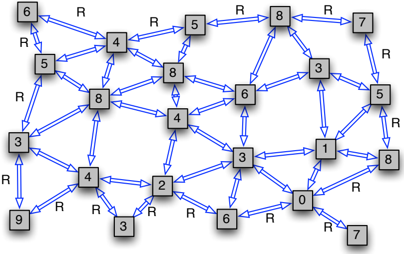
- At each time t all neighbors (i, j) exchange messages

Quantized communication



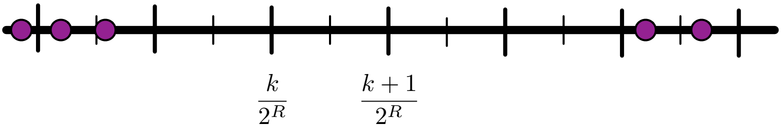
- At each time t all neighbors (i, j) exchange messages
- Messages $i \to j$ and $j \to i$ must take no more than R bits

Quantized communication



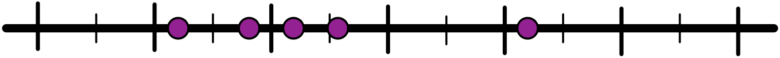
- At each time t all neighbors (i, j) exchange messages
- Messages $i \to j$ and $j \to i$ must take no more than R bits
- Update $x_i(t)$ as a function of $x_i(t-1)$ and messages $\{j \to i\}$

Defining consensus



Consensus is when all estimates end up in the same quantization bin.

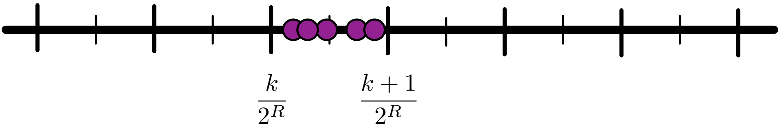
Defining consensus



$\frac{k}{2^R}$ $\frac{k+1}{2^R}$

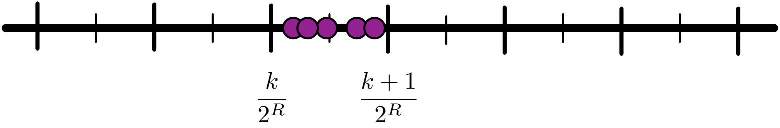
Consensus is when all estimates end up in the same quantization bin.

Defining consensus



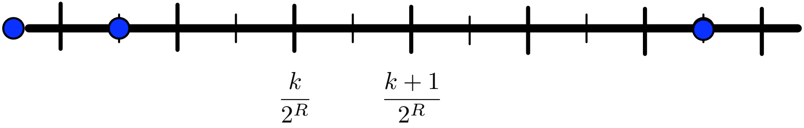
Consensus is when all estimates end up in the same quantization bin.

Defining consensus



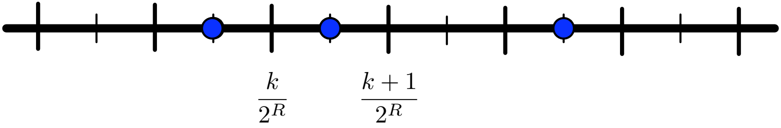
Consensus is when all estimates end up in the same quantization bin.
Consensus at R bit resolution.

Defining consensus



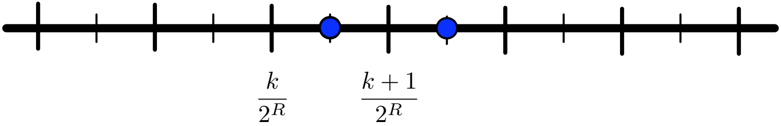
Consensus is when all estimates end up in the same quantization bin.
Consensus at R bit resolution.

Defining consensus



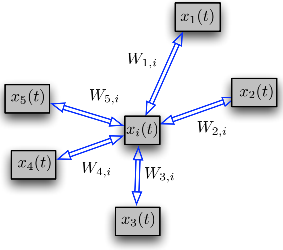
Consensus is when all estimates end up in the same quantization bin.
Consensus at R bit resolution.

Defining consensus



Consensus is when all estimates end up in the same quantization bin.
Consensus at R bit resolution.

A simple protocol



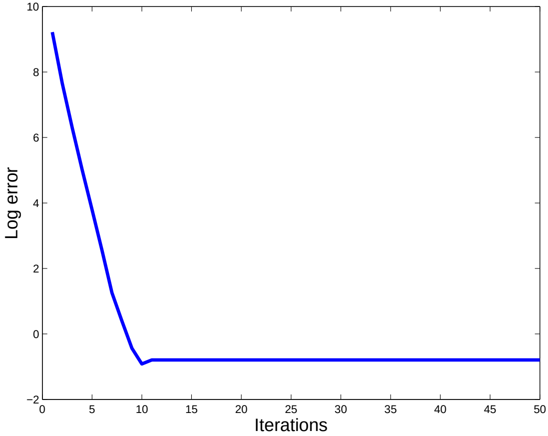
$$x_i(t+1) = x_i(t) + \sum_{j \in \mathcal{N}_i} W_{ij}(\hat{x}_j(t) - \hat{x}_i(t)).$$

- Assume entries of W are also quantized to R_0 bits.
- Iterations preserve sum $\sum_i x_i(t)$.

So how well does it work?

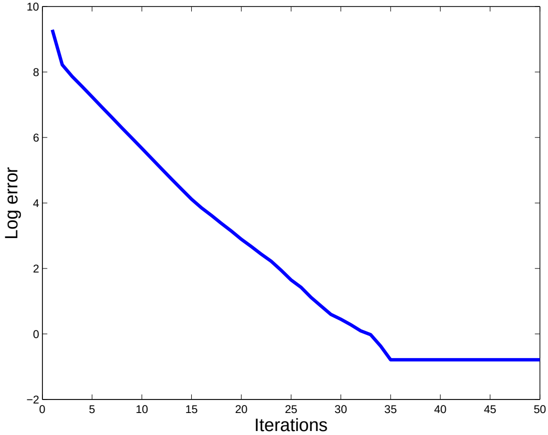
So how well does it work?

Random topology, 49 nodes, good connectivity

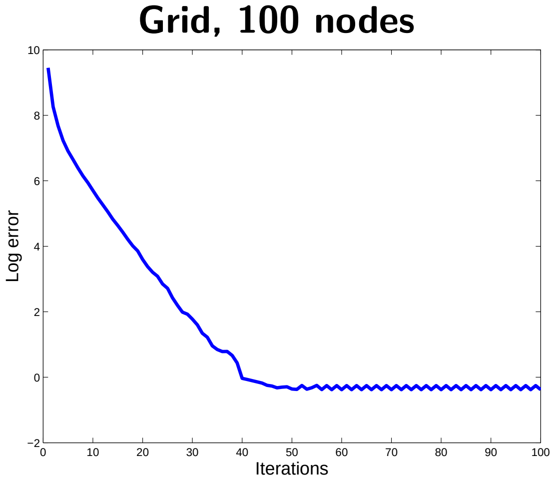


So how well does it work?

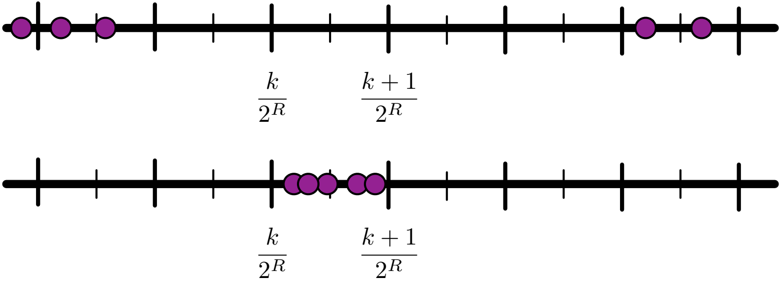
Random topology, 49 nodes, poor connectivity



So how well does it work?

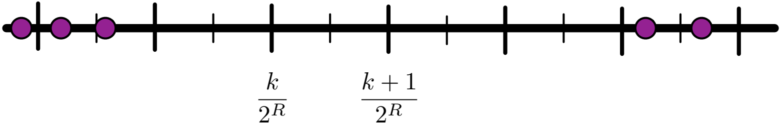


Convergence results



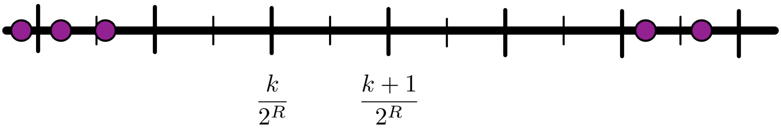
Main result : If W is doubly stochastic, symmetric, and $W_{ii} > 0$ then the algorithm converges to consensus.

Convergence ideas



$$\begin{aligned}\mathbf{x}(t+1) &= (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) + W\hat{\mathbf{x}}(t) \\ &= \mathbf{e}(t) + W\hat{\mathbf{x}}(t) \\ (\mathbf{x}(t+1) - x_{\mathrm{ave}}\mathbf{1}) &= (\mathbf{x}(t) - x_{\mathrm{ave}}\mathbf{1}) + (W - I)\mathbf{e}(t).\end{aligned}$$

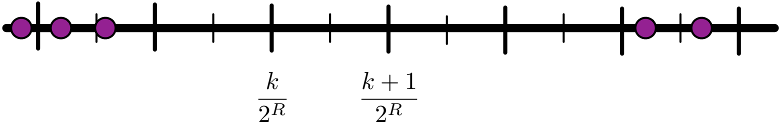
Convergence ideas



$$\begin{aligned} \mathbf{x}(t+1) &= (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) + W\hat{\mathbf{x}}(t) \\ &= \mathbf{e}(t) + W\hat{\mathbf{x}}(t) \\ (\mathbf{x}(t+1) - x_{\text{ave}}\mathbf{1}) &= (\mathbf{x}(t) - x_{\text{ave}}\mathbf{1}) + (W - I)\mathbf{e}(t). \end{aligned}$$

- W doubly stochastic, symmetric $\rightarrow$ reversible MC with π uniform

Convergence ideas

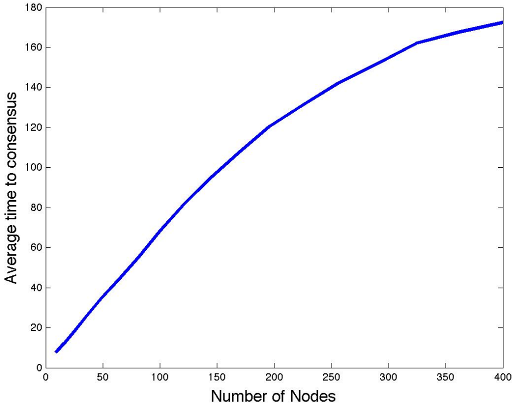


$$\begin{aligned}\mathbf{x}(t+1) &= (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) + W\hat{\mathbf{x}}(t) \\ &= \mathbf{e}(t) + W\hat{\mathbf{x}}(t) \\ (\mathbf{x}(t+1) - x_{\text{ave}}\mathbf{1}) &= (\mathbf{x}(t) - x_{\text{ave}}\mathbf{1}) + (W - I)\mathbf{e}(t).\end{aligned}$$

- W doubly stochastic, symmetric $\rightarrow$ reversible MC with π uniform
- Self-transitions ensure $\|(W-I)\mathbf{e}(t)\|_1$ decreases.

Convergence ideas

Time to converge vs. size of grid



The rate of convergence

$$\begin{aligned}\mathbf{x}(t) &= W\mathbf{x}(t-1) + (I - W)\mathbf{e}(t-1) \\ &= W^t\mathbf{x}(0) + \sum_{s=1}^{t}(W^{s-1} - W^s)\mathbf{e}(t-s).\end{aligned}$$

The rate of convergence

$$\begin{aligned}\mathbf{x}(t) &= W\mathbf{x}(t-1) + (I - W)\mathbf{e}(t-1) \\ &= W^t\mathbf{x}(0) + \sum_{s=1}^{t}(W^{s-1} - W^s)\mathbf{e}(t-s).\end{aligned}$$

- First term $W^t\mathbf{x}(0) \to x_{\text{ave}}\mathbf{1}$ by standard MC results.

The rate of convergence

$$\mathbf{x}(t) = W\mathbf{x}(t-1) + (I - W)\mathbf{e}(t-1)$$
$$= W^t \mathbf{x}(0) + \sum_{s=1}^{t} (W^{s-1} - W^s)\mathbf{e}(t-s).$$

- First term $W^t \mathbf{x}(0) \to x_{\text{ave}}\mathbf{1}$ by standard MC results.
- Second term is not quite a telescoping sum...

Limiting error term

$$\mathbf{E}(t) = \sum_{s=1}^{t} (W^{s-1} - W^{s}) \mathbf{e}(t-s).$$

Limiting error term

$$\mathbf{E}(t) = \sum_{s=1}^{t} (W^{s-1} - W^{s}) \mathbf{e}(t-s).$$

- $|E_i(t)|$ is a convex function of past errors.

Limiting error term

$$\mathbf{E}(t) = \sum_{s=1}^{t} (W^{s-1} - W^{s}) \mathbf{e}(t-s).$$

- $|E_i(t)|$ is a convex function of past errors.
- Upper bound by maximizing over $\{\mathbf{e}(t-s) \in \{-\frac{\Delta}{2}, \frac{\Delta}{2}\}\}$

Limiting error term

$$\mathbf{E}(t) = \sum_{s=1}^{t} (W^{s-1} - W^{s}) \mathbf{e}(t-s).$$

- $|E_i(t)|$ is a convex function of past errors.
- Upper bound by maximizing over $\{\mathbf{e}(t-s) \in \{-\frac{\Delta}{2}, \frac{\Delta}{2}\}\}$

$$\Delta \sum_{s=1}^{t} \left\| W^{s-1}(i,\cdot) - W^{s}(i,\cdot) \right\|_{TV}$$

Limiting error term

$$\mathbf{E}(t) = \sum_{s=1}^{t} (W^{s-1} - W^{s}) \mathbf{e}(t-s).$$

- $|E_i(t)|$ is a convex function of past errors.
- Upper bound by maximizing over $\{\mathbf{e}(t-s) \in \{-\frac{\Delta}{2}, \frac{\Delta}{2}\}\}$

$$\Delta \sum_{s=1}^{t} \left\| W^{s-1}(i,\cdot) - W^{s}(i,\cdot) \right\|_{TV}$$

Game is now to find good bounds on one-step change...

A generic upper bound for reversible chains

A generic upper bound for reversible chains

For reversible Markov chains,

$$\left\|\delta_i^T W^s - \pi^T\right\|_{TV}^2 \le \frac{W^2(i,i)}{\pi(i)} \lambda_2(W)^{2s-2},$$

A generic upper bound for reversible chains

For reversible Markov chains,

$$\left\|\delta_i^T W^s - \pi^T\right\|_{TV}^2 \leq \frac{W^2(i,i)}{\pi(i)} \lambda_2(W)^{2s-2},$$

Applying it gives

$$|E_i(t)| \leq c\sqrt{n}\Delta$$

A generic upper bound for reversible chains

For reversible Markov chains,

$$\left\| \delta_i^T W^s - \pi^T \right\|_{TV}^2 \leq \frac{W^2(i,i)}{\pi(i)} \lambda_2(W)^{2s-2},$$

Applying it gives

$$|E_i(t)| \leq c\sqrt{n}\Delta$$

Says error decay rate is exponential only to the level of $\sqrt{n}\Delta$.

A generic upper bound for reversible chains

For reversible Markov chains,

$$\left\| \delta_i^T W^s - \pi^T \right\|_{TV}^2 \le \frac{W^2(i,i)}{\pi(i)} \lambda_2(W)^{2s-2},$$

Applying it gives

$$|E_i(t)| \le c\sqrt{n}\Delta$$

Says error decay rate is exponential only to the level of $\sqrt{n}\Delta$. **!!!**

Well-connected networks

Well-connected networks

If graph is "well-connected", then:

$$\left\| \mathbf{a}W^k - \mathbf{b}W^k \right\|_{TV} \le \|\mathbf{a} - \mathbf{b}\|_{TV} \, d(W)^k.$$

where

$$d(W) = \sup_{i,j} \|W(i,\cdot) - W(j,\cdot)\|_{TV}$$

Well-connected networks

If graph is "well-connected", then:

$$\left\| \mathbf{a} W^k - \mathbf{b} W^k \right\|_{TV} \le \|\mathbf{a} - \mathbf{b}\|_{TV} \, d(W)^k.$$

where

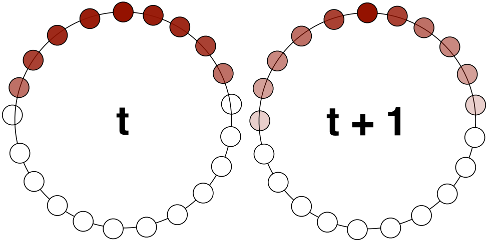
$$d(W) = \sup_{i,j} \|W(i, \cdot) - W(j, \cdot)\|_{TV}$$

Gives

$$|E_i(t)| = c\Delta.$$

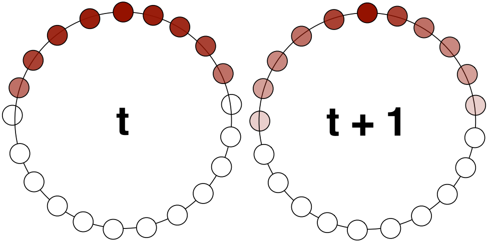
Rate is slower than $\lambda_2(W)$.

A closer look at the error term



$$\left\| W^{s-1}(i,\cdot) - W^{s}(i,\cdot) \right\|_{TV}$$

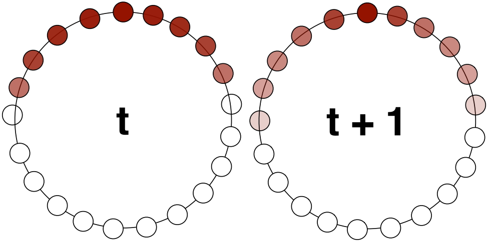
A closer look at the error term



$$\left\| W^{s-1}(i, \cdot) - W^{s}(i, \cdot) \right\|_{TV}$$

- If chain mixes slowly, each step is small.

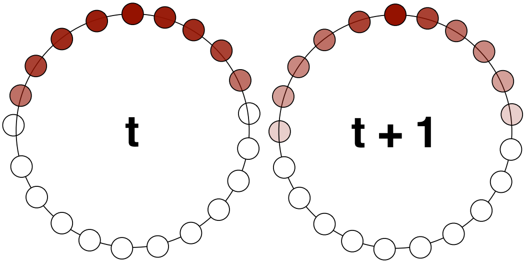
A closer look at the error term



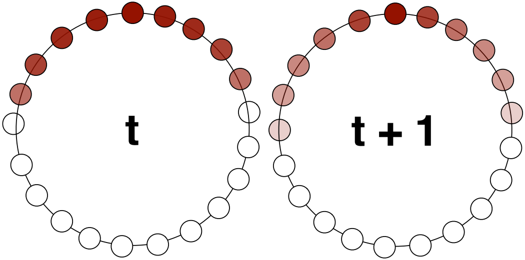
$$\left\| W^{s-1}(i, \cdot) - W^{s}(i, \cdot) \right\|_{TV}$$

- If chain mixes slowly, each step is small.
- If chain mixes quickly, sum of steps is small.

Approaches to refining the bound

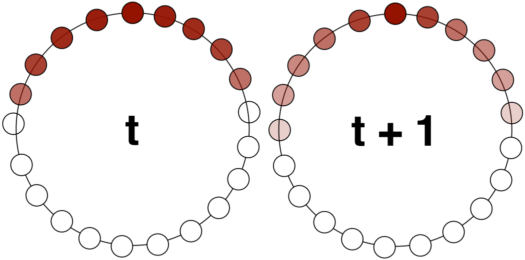


Approaches to refining the bound



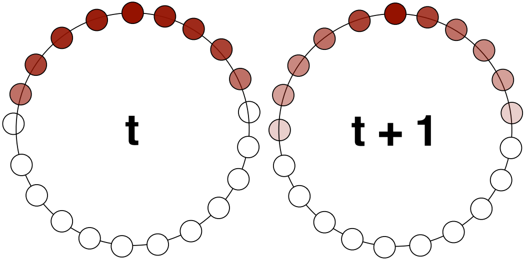
- We know that $\left\|W^{s-1}(i,\cdot) - W^{s}(i,\cdot)\right\|_{TV}$ is given by the best coupling time.

Approaches to refining the bound



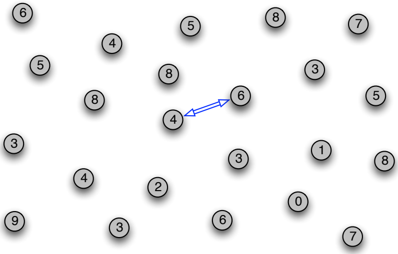
- We know that $\left\| W^{s-1}(i,\cdot) - W^{s}(i,\cdot) \right\|_{TV}$ is given by the best coupling time.
- Construct a coupling that better matches the observed behavior.

Approaches to refining the bound

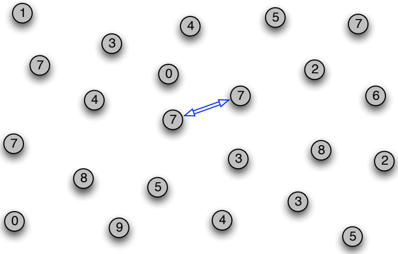


- We know that $\left\| W^{s-1}(i,\cdot) - W^{s}(i,\cdot) \right\|_{TV}$ is given by the best coupling time.
- Construct a coupling that better matches the observed behavior.
- Possible connections to other interesting criteria?

Extensions to gossip

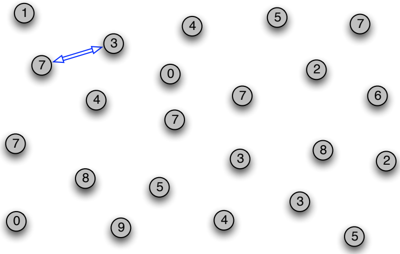


Extensions to gossip



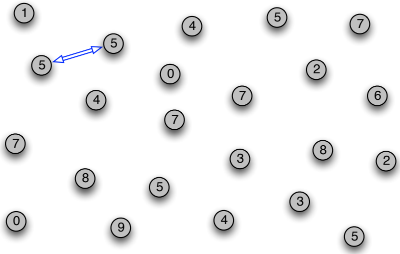
- Lots of work on *gossip* (asynchronous iterations)

Extensions to gossip



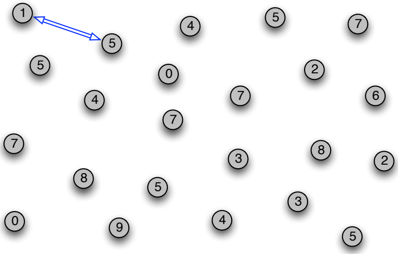
- Lots of work on *gossip* (asynchronous iterations)

Extensions to gossip



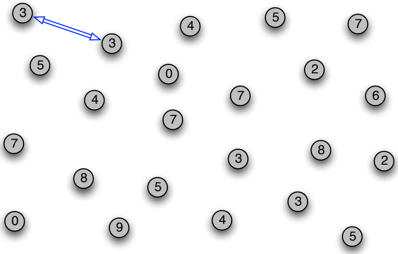
- Lots of work on *gossip* (asynchronous iterations)

Extensions to gossip



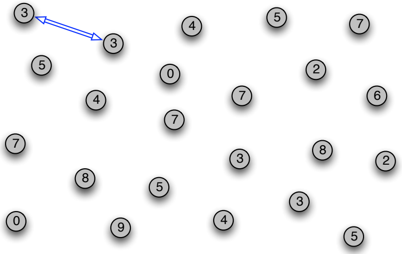
- Lots of work on *gossip* (asynchronous iterations)

Extensions to gossip



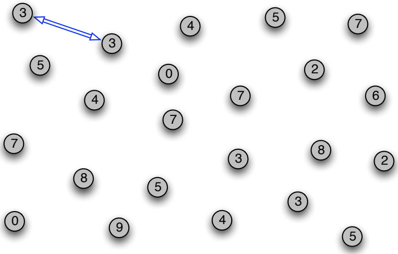
- Lots of work on *gossip* (asynchronous iterations)

Extensions to gossip



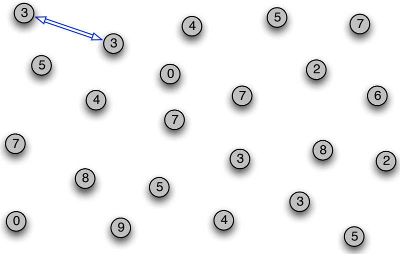
- Lots of work on *gossip* (asynchronous iterations)

Extensions to gossip



- Lots of work on *gossip* (asynchronous iterations)
- Kashyap et al. ('07), Carli et al. ('10), Zhu and Martínez ('10), Lavaei and Murray ('10)...

Extensions to gossip



- Lots of work on *gossip* (asynchronous iterations)
- Kashyap et al. ('07), Carli et al. ('10), Zhu and Martínez ('10), Lavaei and Murray ('10)...
- Should be able to prove similar results.

Conclusions and cbservations

- Quantization is important for practical applications.
- Average consensus to within reasonable resolution can be fast.
- Overhead can be reduced by piggybacking on existing traffic.

Next step: implement e.g. in a sensornet testbed..

Thanks!