Engineering Emergence through Gossip

Márk Jelasity

University of Bologna
Italy

Project funded by the Future and Emerging Technologies arm of the IST Programme

Abstract Components of Gossip (I)

- Communication network
  - The nodes of the network are people
  - The connections (edges) in the network are defined by relations such as neighbours, friends, relatives, etc. Communication takes place among connected people in this network
  - Network is restricted: e.g. in social networks often small diameter, high clustering, Zipf degree distribution
  - It can change, perhaps as a result of gossip

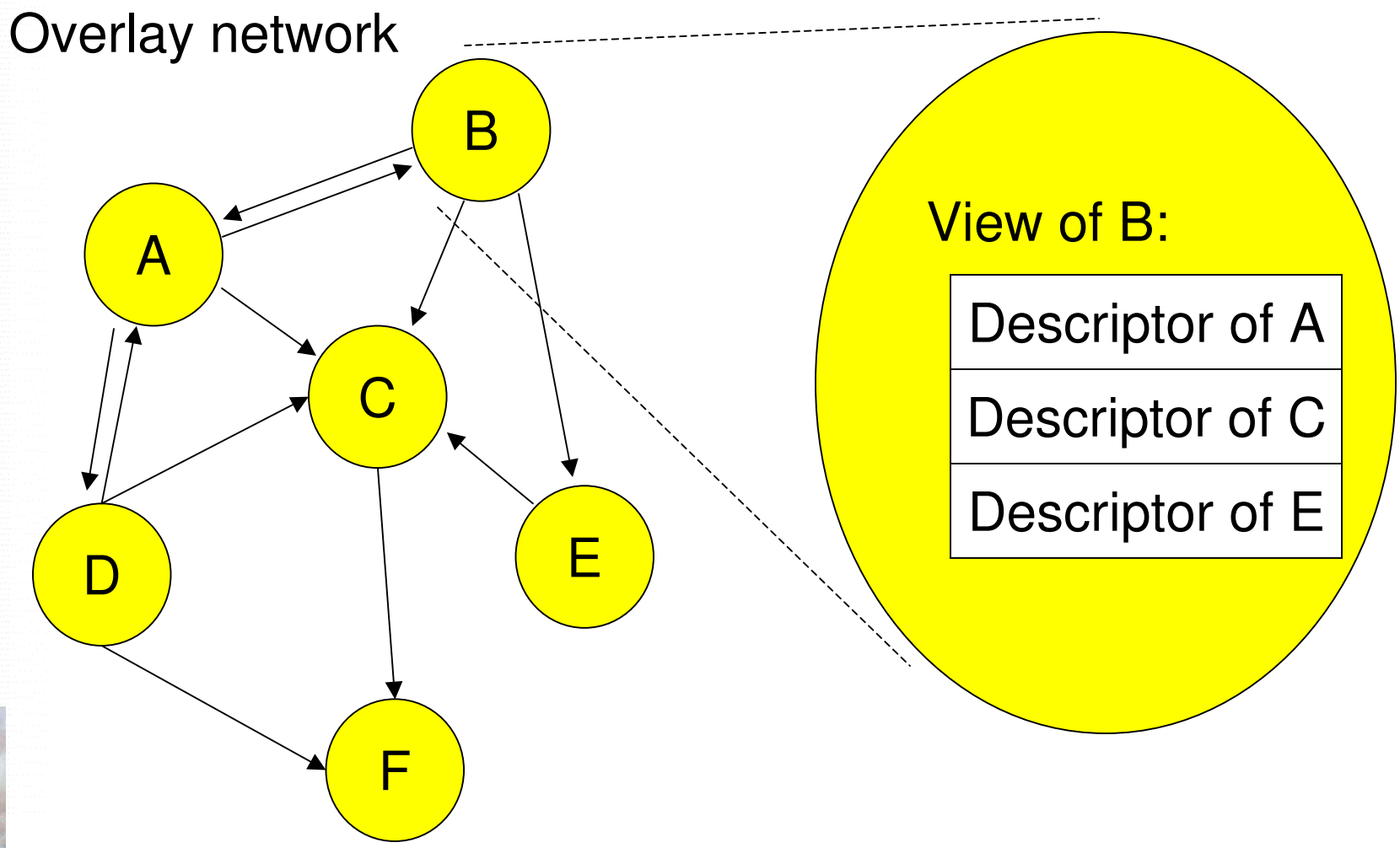
Abstract Components of Gossip (II)

- Communication Algorithm
  - People exchange information with their neighbours in the social network more or less regulary.
  - They might have a bias towards interacting with some people (famous, rich, understanding, close-by, funny, etc).
- Computation
  - People also process information: they reason about it, alter or combine it. They also percieve or forget information.

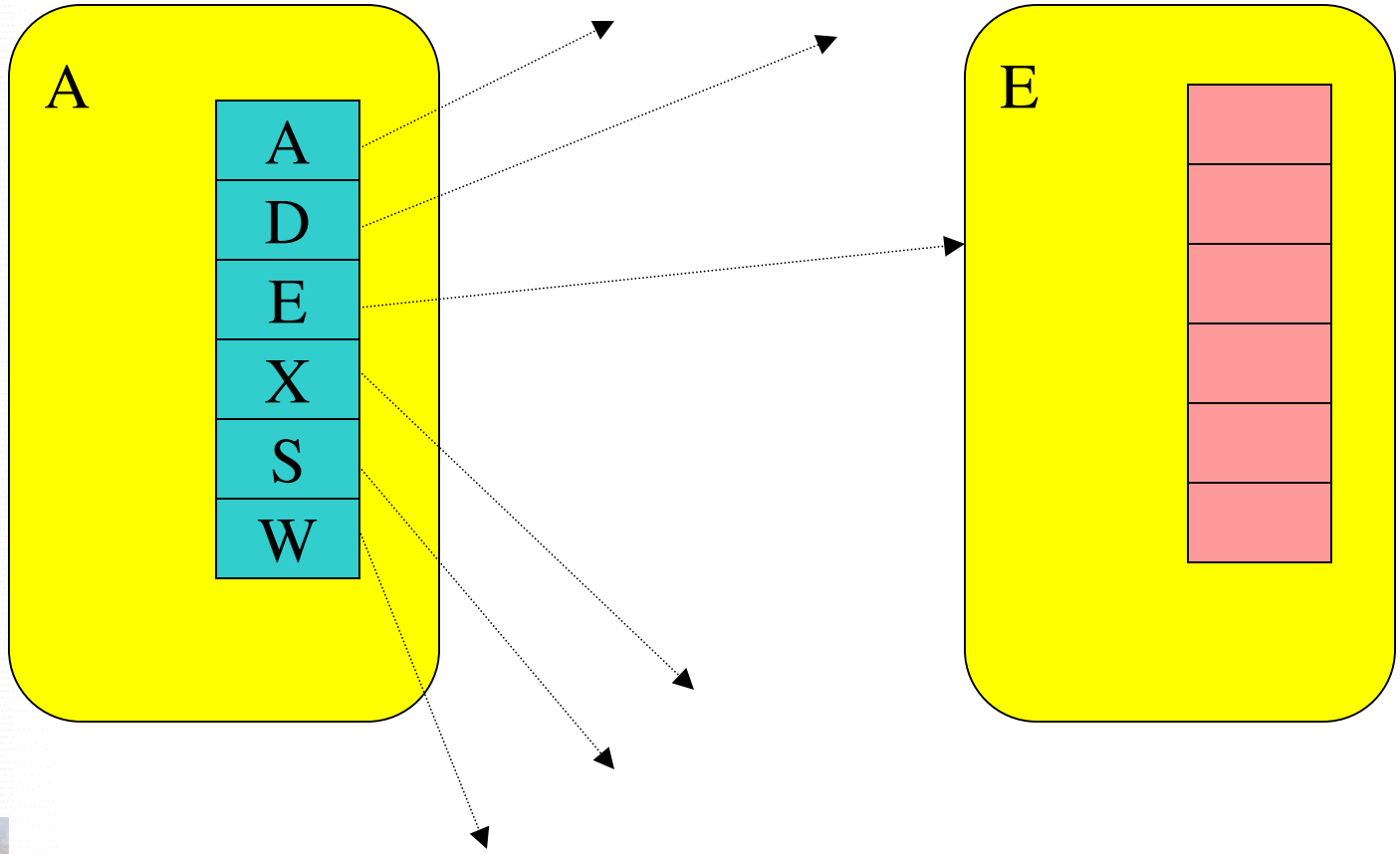
Overlay Networks

- Communication Network
  - Nodes are computing devices connected to a computer network
  - Neighbours are defined by the “knows-about” relation (NOT physical neighbors in the network). Eg WWW, file-sharing networks, Skype.
- Communication Algorithm
  - Each node regularly selects a neighbour to exchange state information with
- Computation
  - Can be arbitrary. It is a very powerful framework that covers information spreading but also other processes like diffusion, reaction-diffusion, random walks, etc.

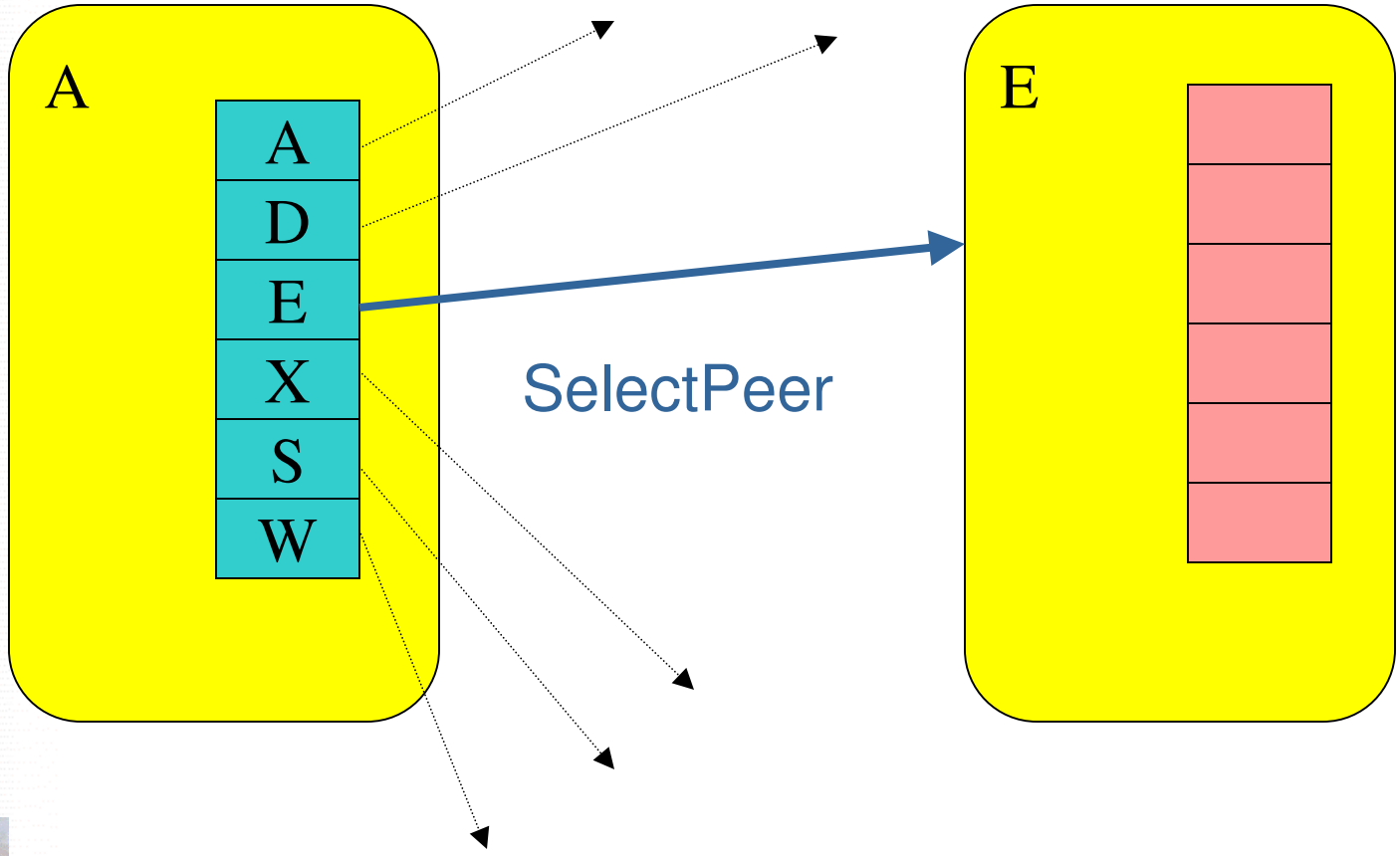
System Abstraction: basic concepts



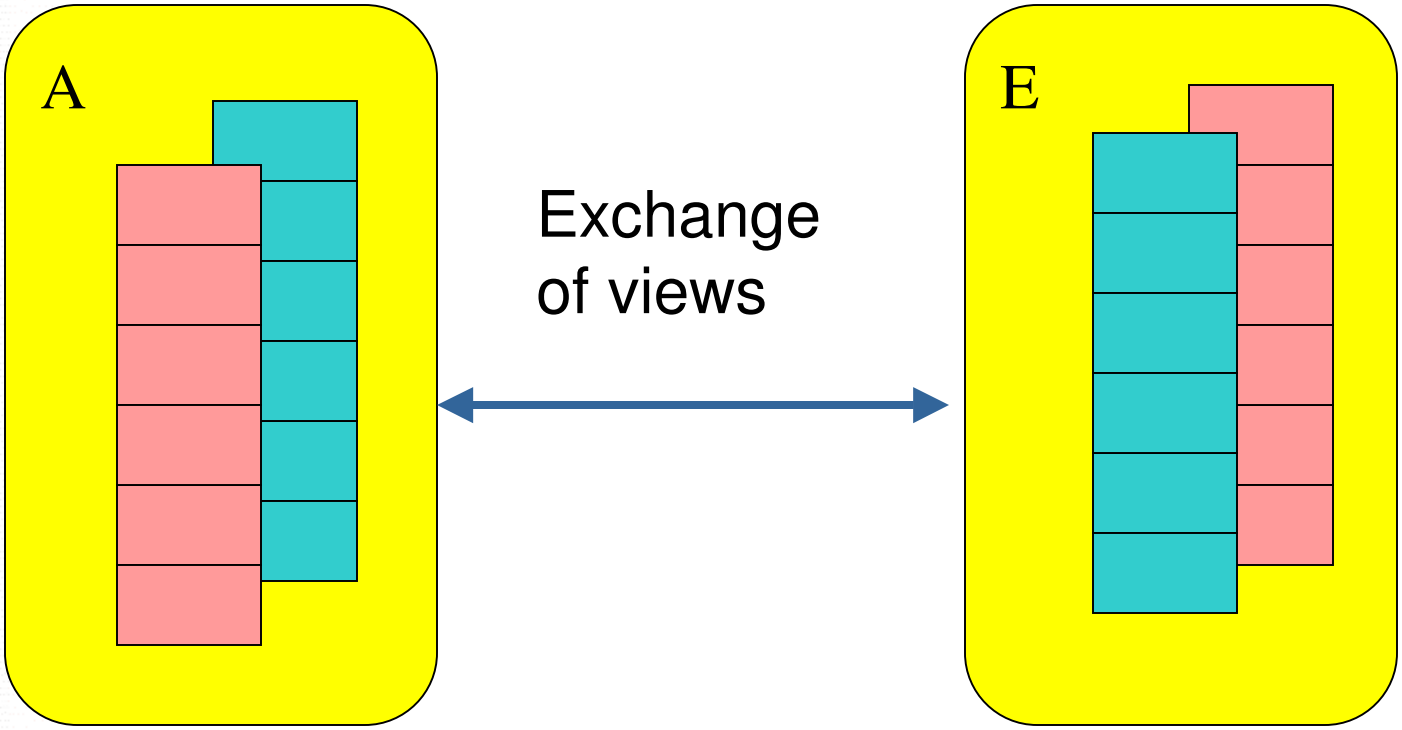
Gossip protocols for topology management



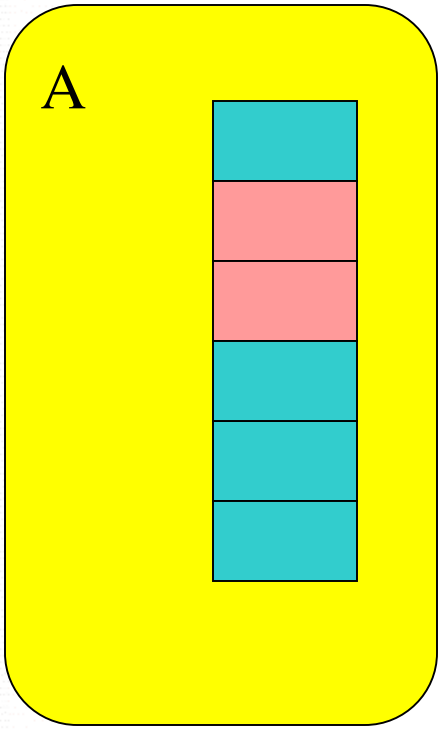
Gossip protocols for topology management



Gossip protocols for topology management

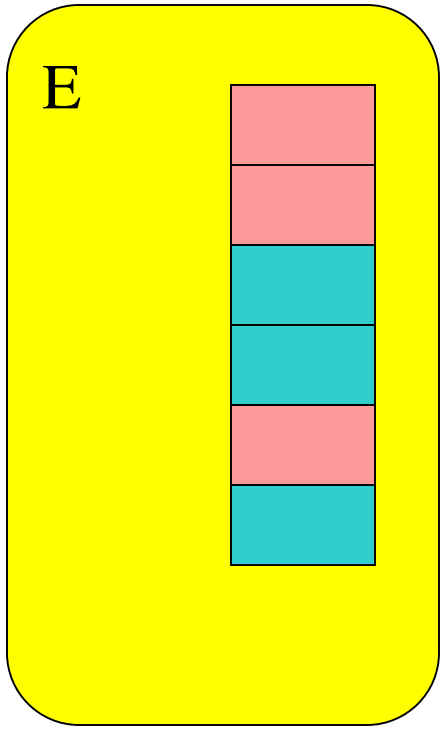


Gossip protocols for topology management



Both sides apply update

thereby redefining topology



Gossip protocols for topology management

- Fully symmetric and decentralized model
- Components of the framework
  - node descriptors: in the view we store not only the address but additional information as well about the nodes
  - selectPeer: uses the actual view to select a peer to contact
  - update($view_1$, $view_2$): based on information available on the peer nodes in the views (node descriptors) constructs the next view

Newscast: a gossip protocol for random topologies

- Goal: generate and maintain a
  - connected random topology
  - in the face of extreme dynamism
- node descriptors: contain timestamp of creating the descriptor
- selectPeer: randomly selects a neighbor from the view
- update: fills the view with the freshest descriptors. New information gradually replaces old information

Newscast: Summary

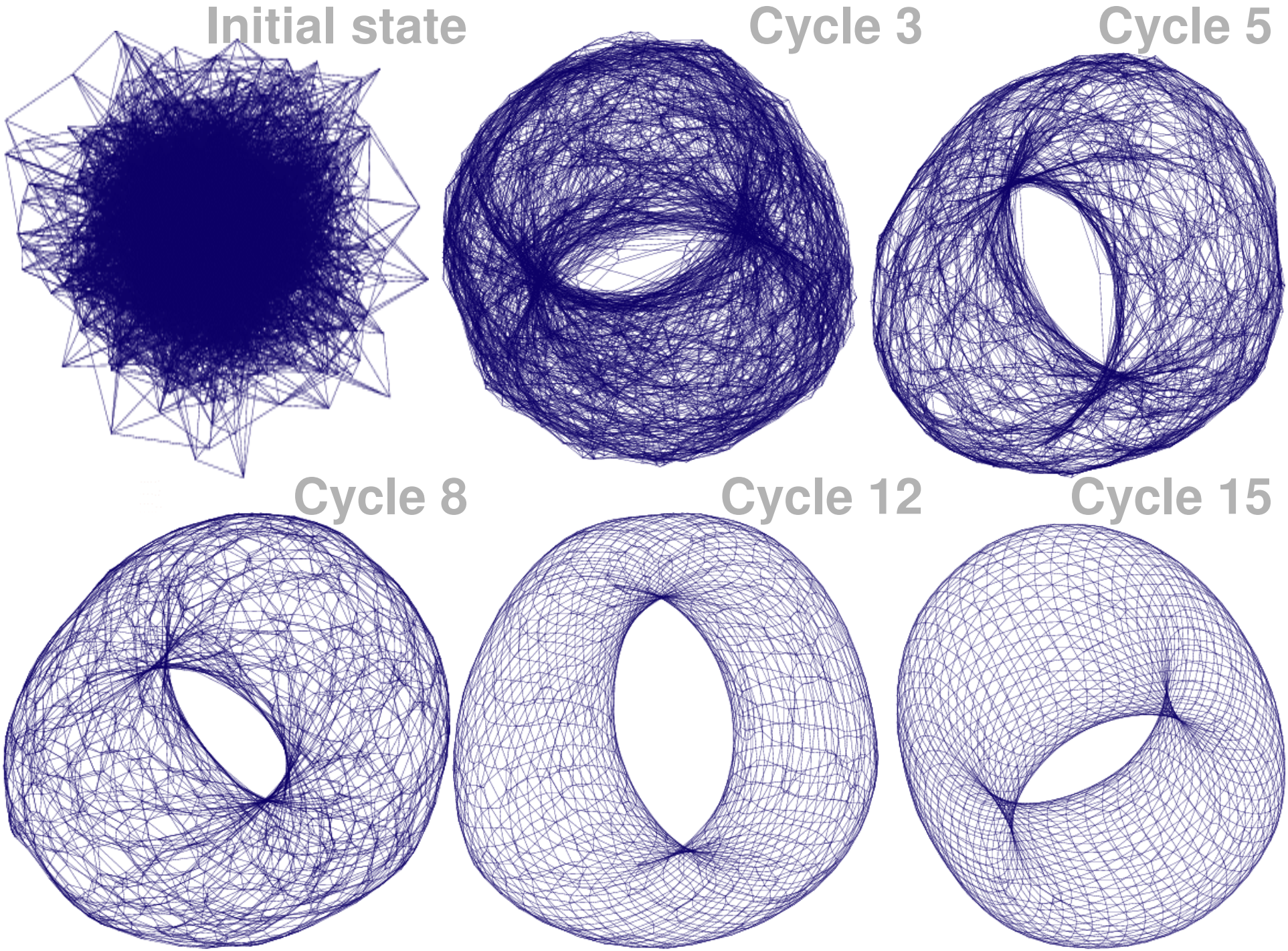
- extremely robust to node and link failure and node dynamism (churn)
- maintains a connected approximately random topology
- scalable
- useful as a source of a continuous stream of random samples from the set of nodes: peer sampling service

T-Man: a gossip protocol for structured topologies

- Goal: quickly generate and maintain a
  - A very wide range of pre-specified or even dynamically specified topologies
  - In the face of dynamism (churn, failures, etc)
- node descriptors: contain the profile of the node (real number, vector, etc)
- selectPeer: Ranks view using a ranking function that defines the target topology and selects the lowest rank neighbor
- update: fills the view with the lowest rank descriptors

Distance based ranking functions

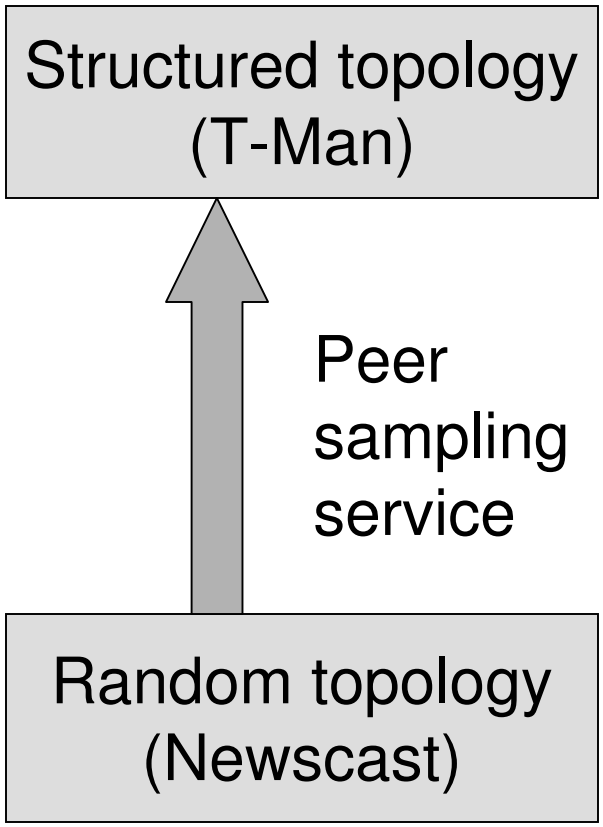
- Example 1 (ring and line): Let the nodes be real numbers. Let the ranking function be defined by the distance $d(a,b)=|a-b|$. For the ring, apply periodic boundary conditions, assuming nodes are from $[0,N]$.
- Example 2 (mesh and torus): Let the nodes be two dimensional real vectors. Similarly to the ring, let the Manhattan distance define the topology


Initial state
Cycle 3
Cycle 5
Cycle 8
Cycle 12
Cycle 15

Biological inspiration

- Result of collaboration with TU Dresden (Andreas Deutsch)
- Biological pattern formation and regeneration: an interesting theory is based on cell adhesion
  - different cell types ”like” or ”dislike” each other
  - any cell configuration has an energy
  - the cells try to improve their neighborhood through a stochastic process

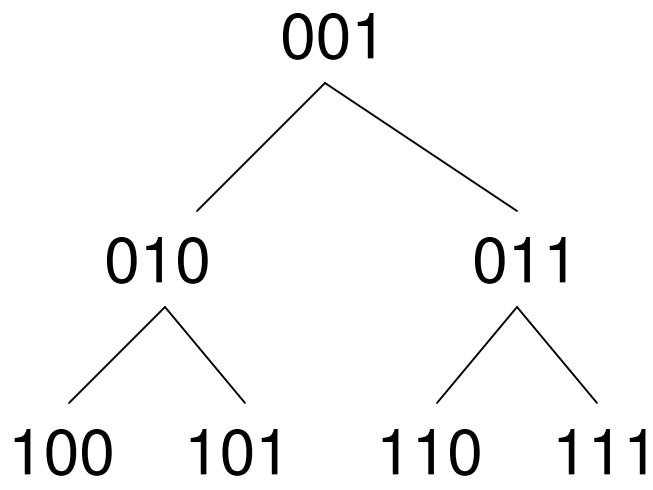
Layered structure



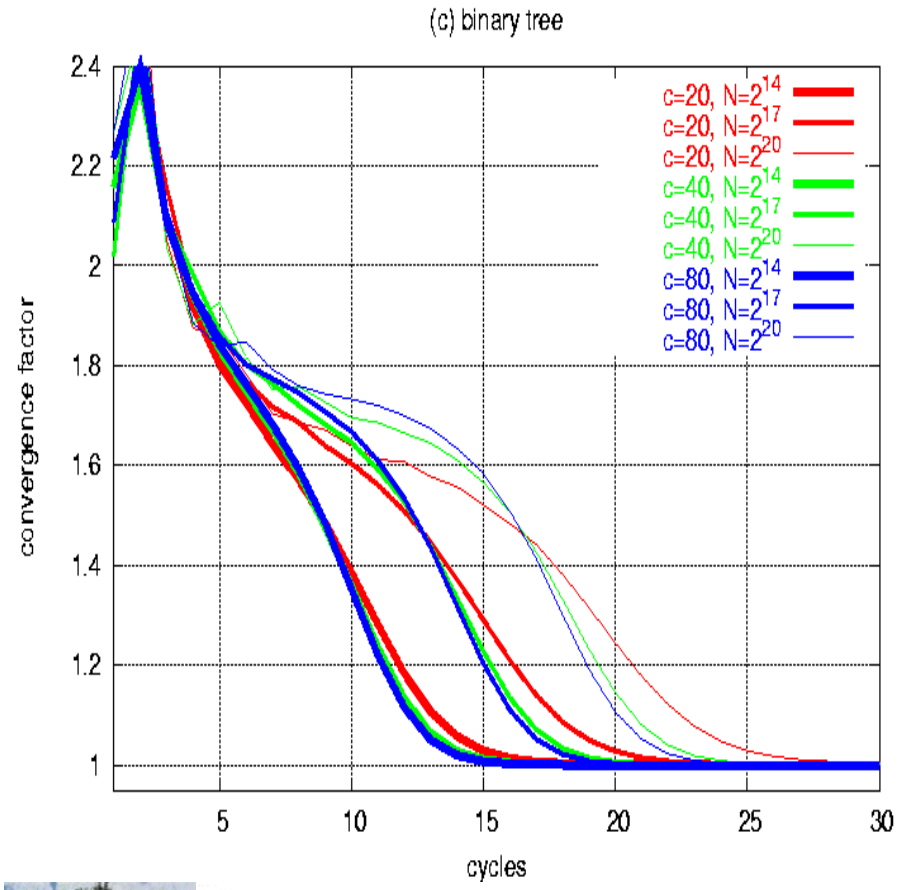
- T-Man views are initialized at random (join)
- T-Man sends random nodes too during information exchange, not only the structured (T-Man) view
  - this helps joining nodes
  - this makes it possible to adapt to changing ranking functions

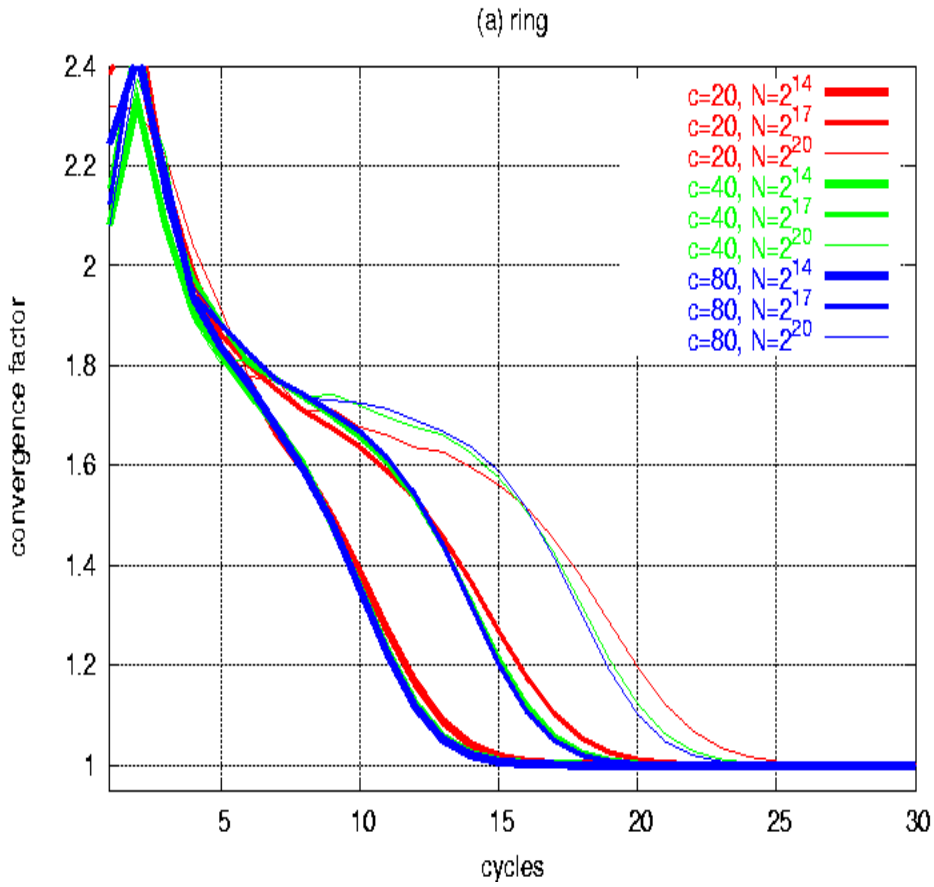
Distance based ranking functions

- Example 3 (binary tree): Let the nodes be binary strings of length m. Let the ranking function be defined by the distance given by the hop count in the binary undirected rooted tree as illustrated

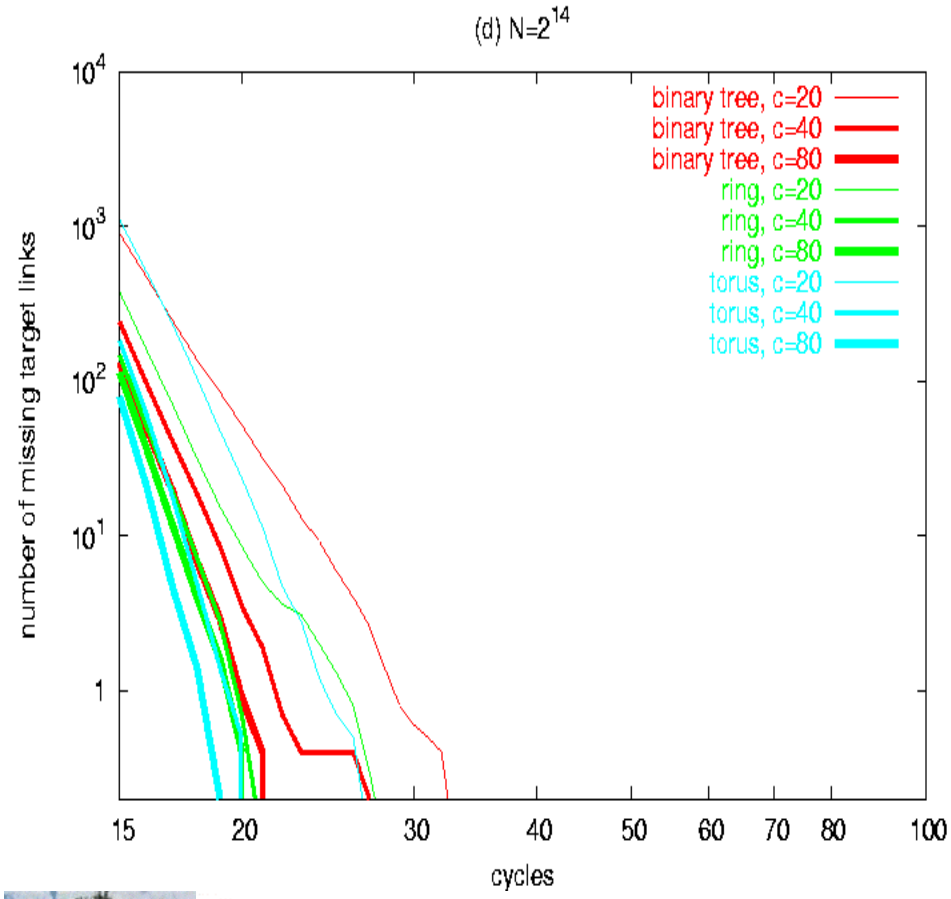


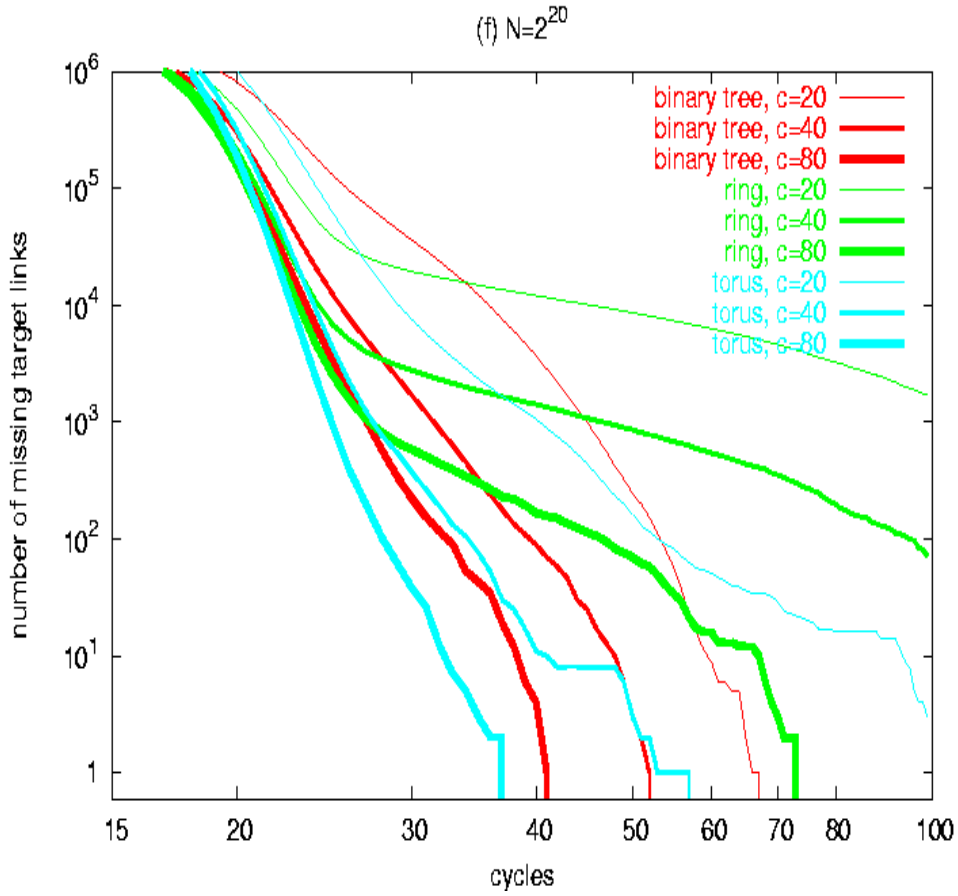
Convergence factor





Time to reach perfect topology

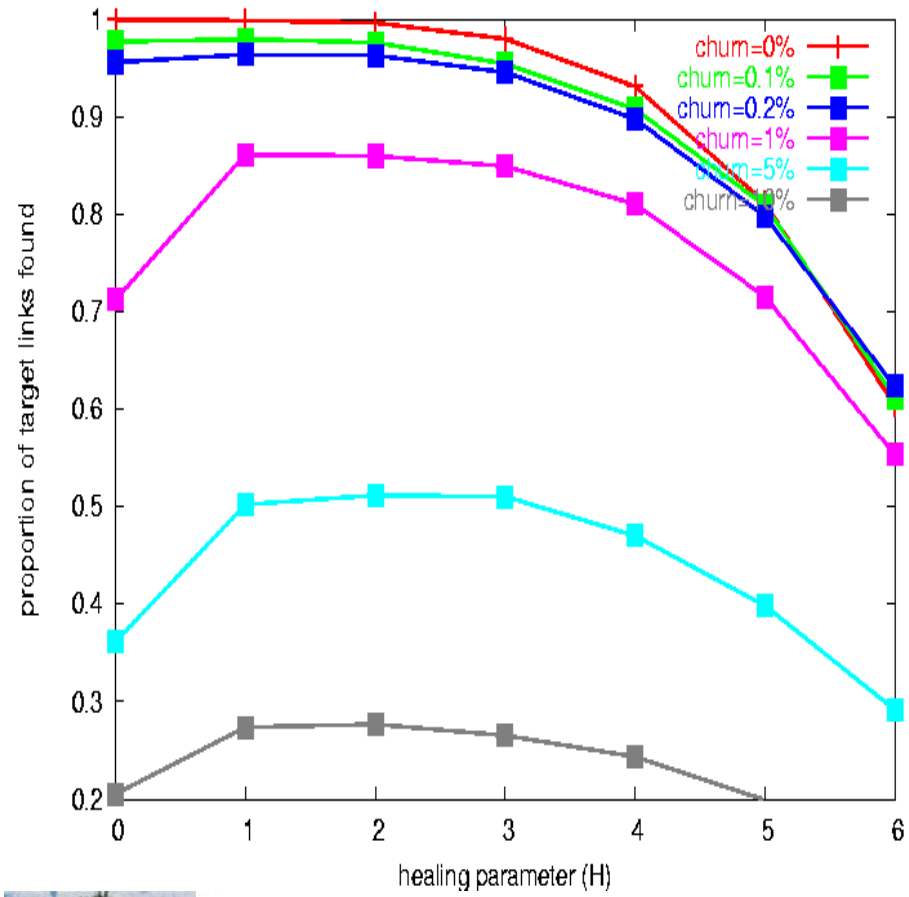




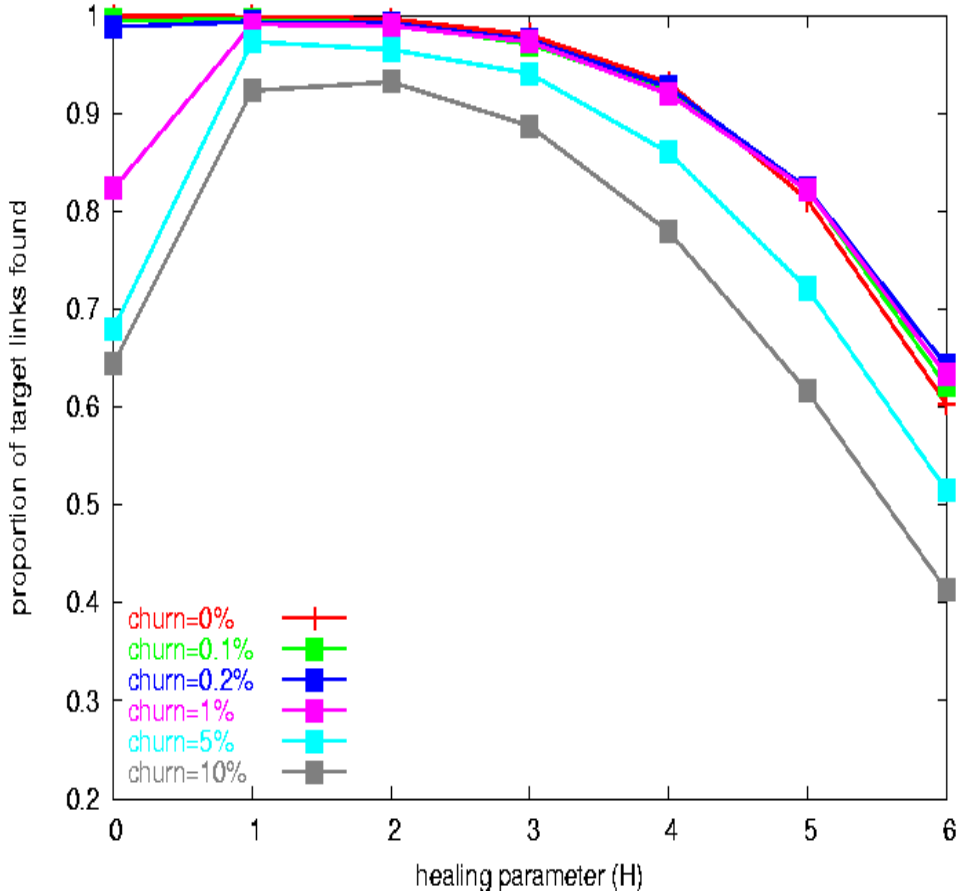
Self-healing

- Similarly to newscast, we add the creation timestamp to node descriptors
- Before exchanging views, the nodes remove the H oldest descriptors (H: self-healing parameter)
- Experiments with artificial, extremely high churn rates

Self-healing

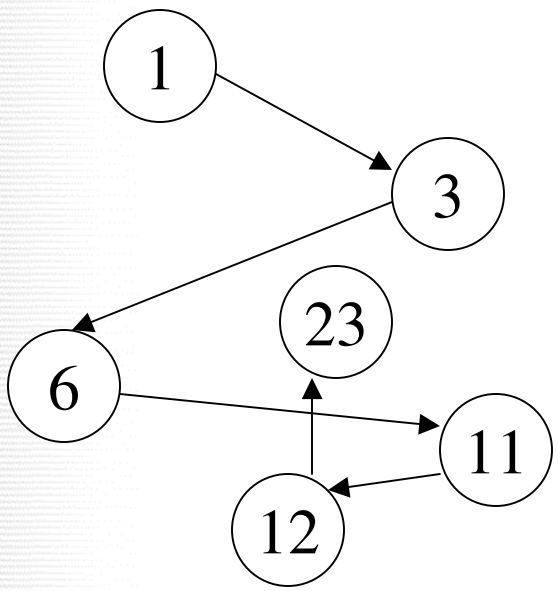


All nodes

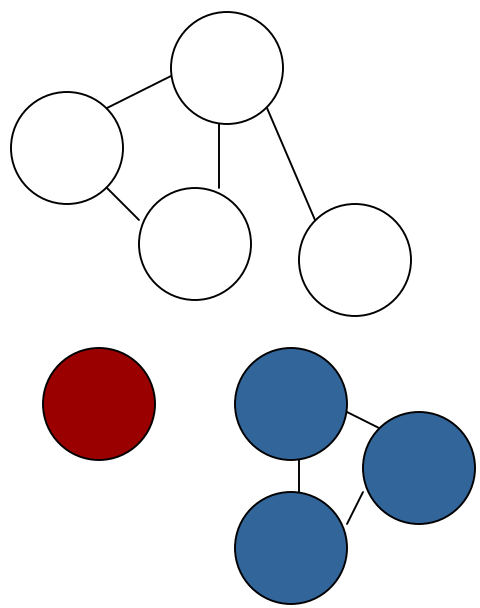


Nodes younger than 10 cycles

Applications of T-Man



Sorting



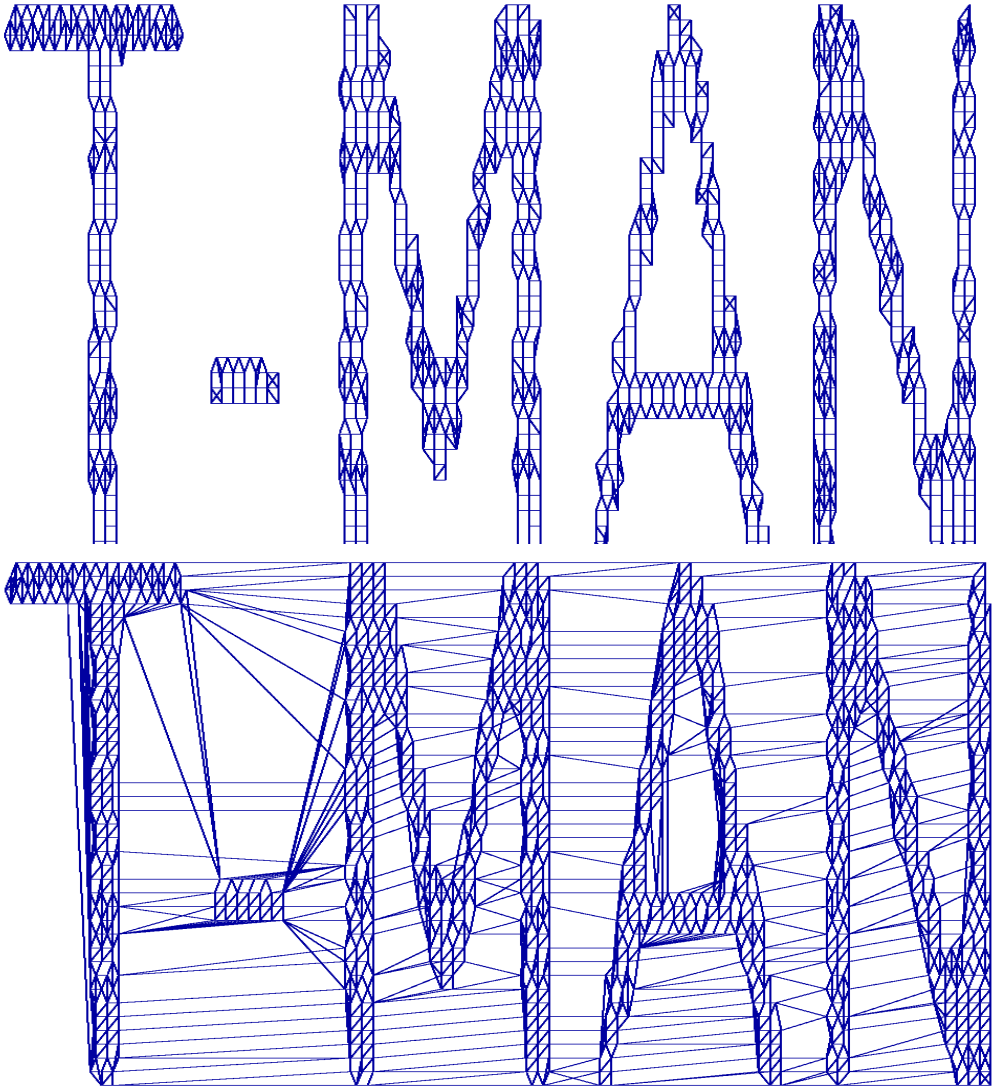
Clustering

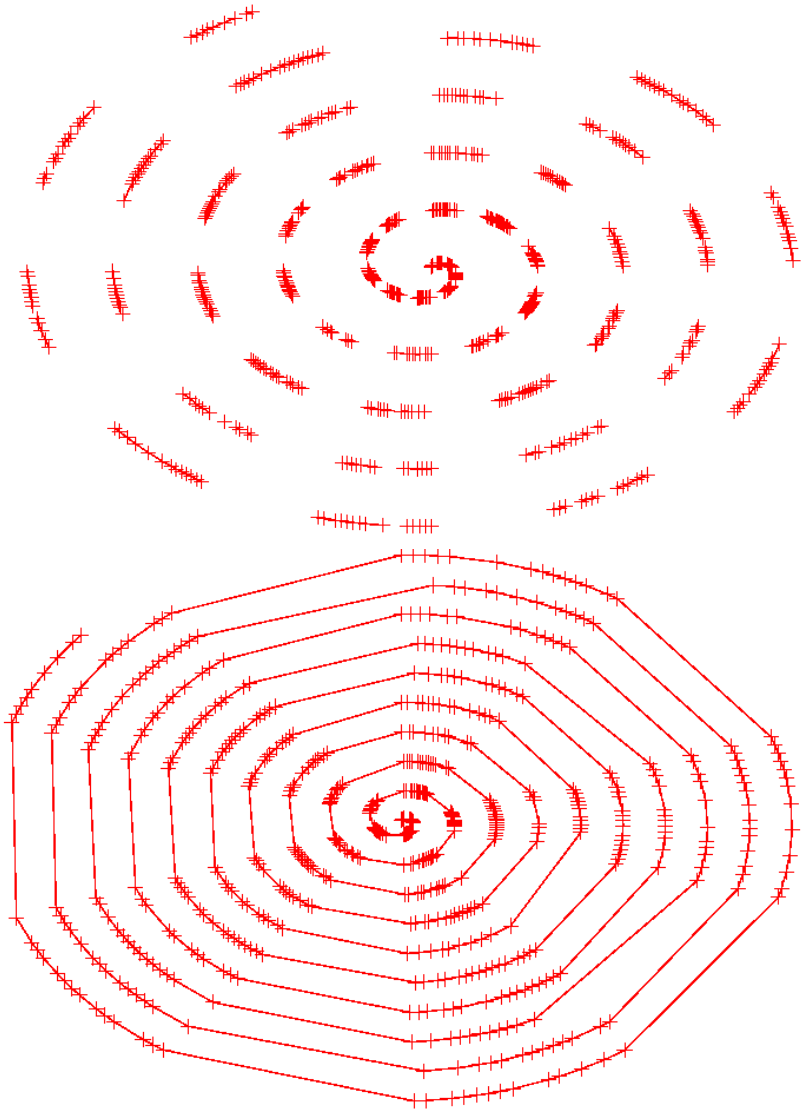
- (geographical, semantic, etc) proximity overlays
- Distributed Hashtables

Direction dependent ranking functions

- Example 4 (sorting): Let <= be a total ordering over the nodes. Let the ranking function apply a distance function consistent with <= separately to those < and > than the base node, and merge the ranked two subsets
  - For example R(10,{1,2,4,100, 300}) could return (4,100,2,300,1). No distance function over the set of nodes generates this ranking function!
- Example 5 (2d proximity): Similar to sorting, classifying nodes into four subsets, ordering them according to distance and merging them.

Illustration of clustering and sorting





T-Man Summary

- capable of generating a wide range of topologies (small and large diameter, clustered, sorted, etc)
- experimental results show that T-Man is scalable: converges with high accuracy in approximately logarithmic time
- many interesting open questions of both theorethical and experimental nature

Fully Distributed Data Aggregation (data mining)

- We assume that we have an overlay network (WWW, file-sharing, or even mobile phones, etc)
- The network is assumed to be large-scale and highly dynamic
- The task is to collect global information about the system (average, maximum, etc of some parameters, network size, data model fitting)

Implementation of Aggregation

- State: current approximation of aggregate
- selectPeer: uses newscast as a service to select a peer to contact
- updateState(s1,s2): elementary aggregation step, examples include
  - (s1+s2)/2 for average
  - $(s1s2)^{1/2}$ for geometric mean
  - max(s1,s2) for maximum
- combining elementary aggregations more complex functions can be computed such as sum, network size, variance, etc.

Illustration of Averaging

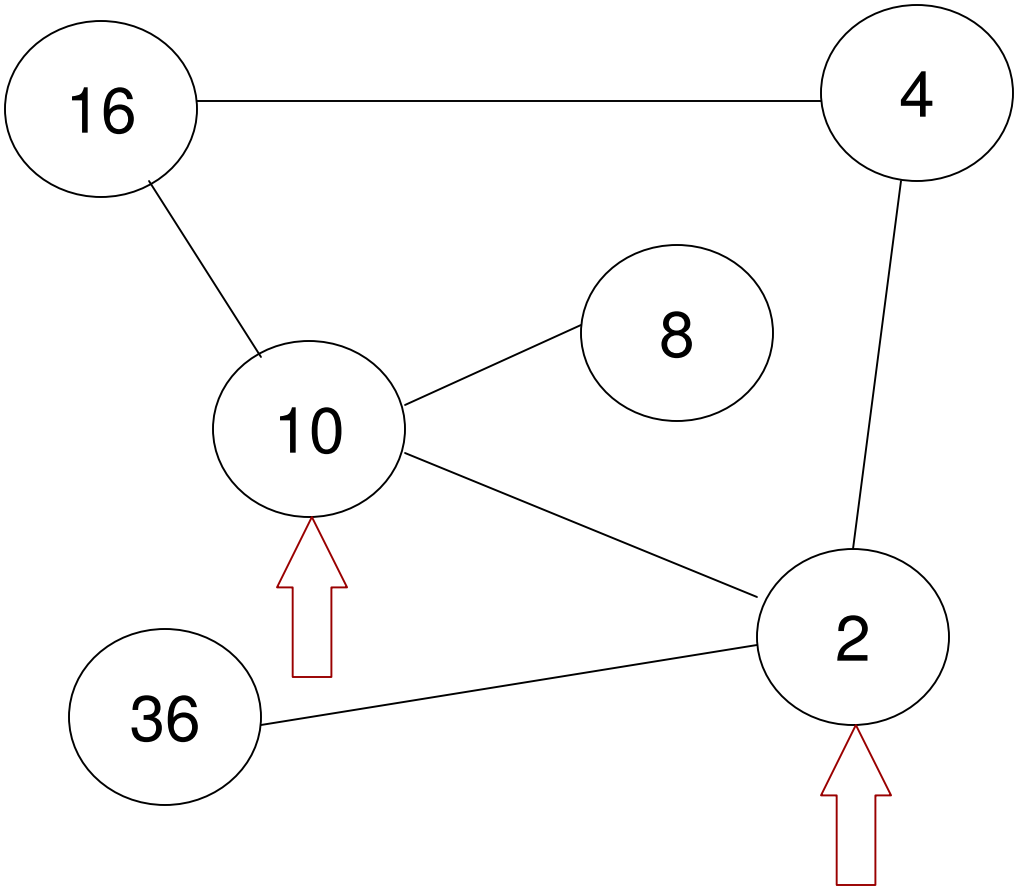


Illustration of Averaging

(10+2)/2=6

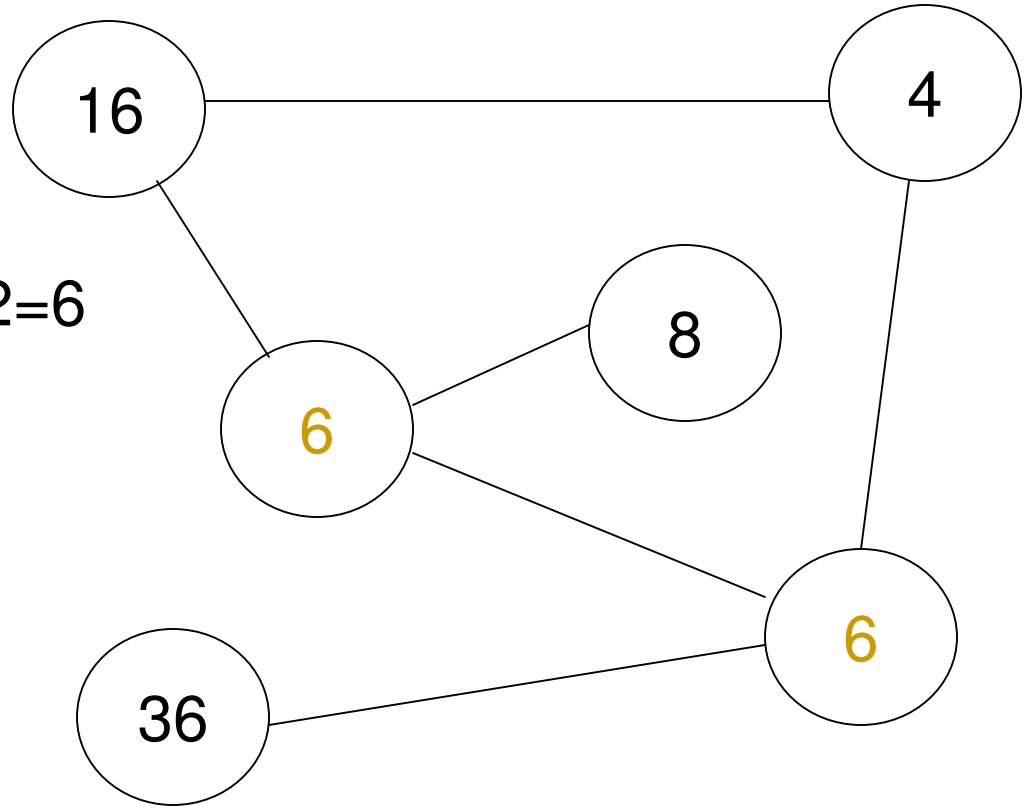


Illustration of Averaging

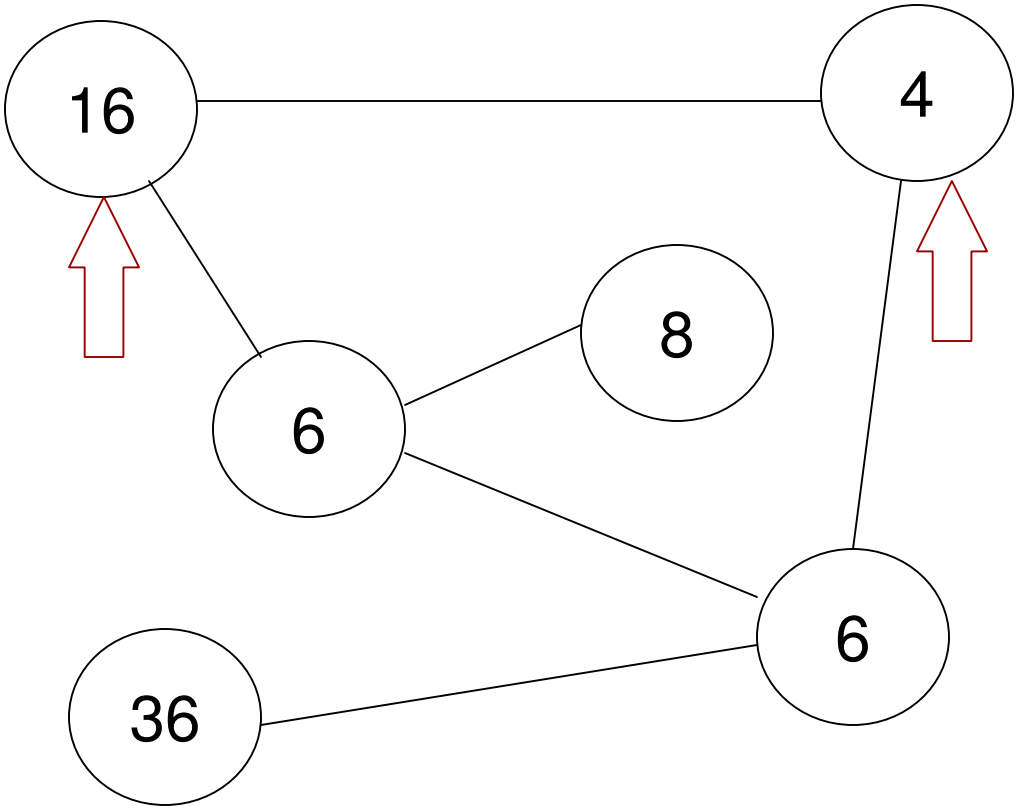


Illustration of Averaging

(16+4)/2=10

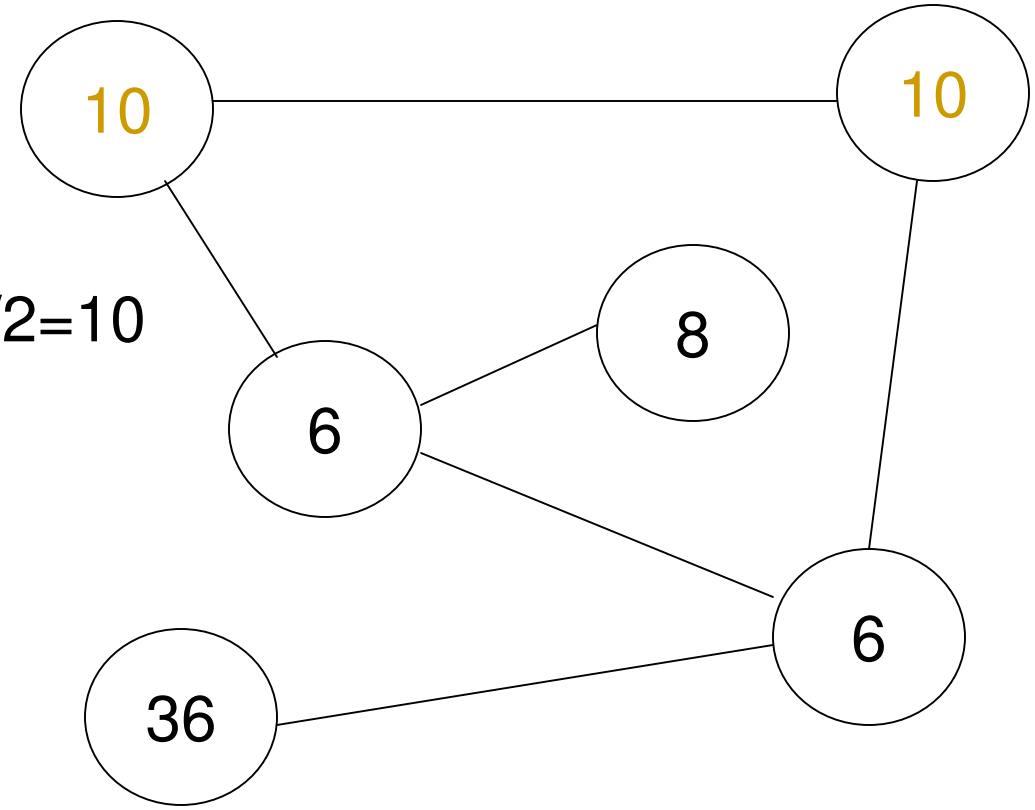
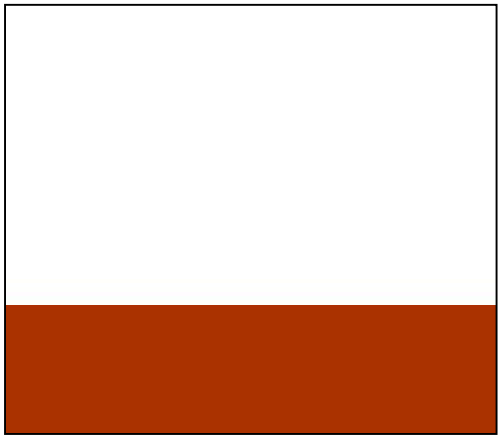
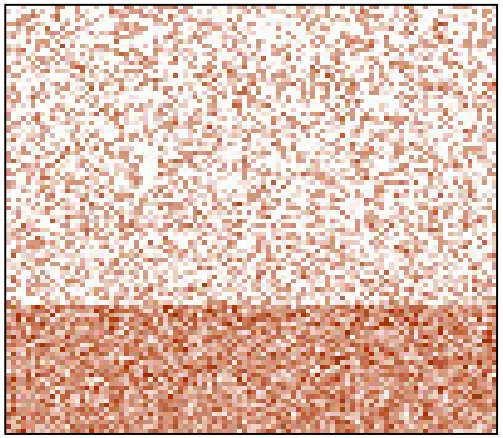


Illustration of average calculation

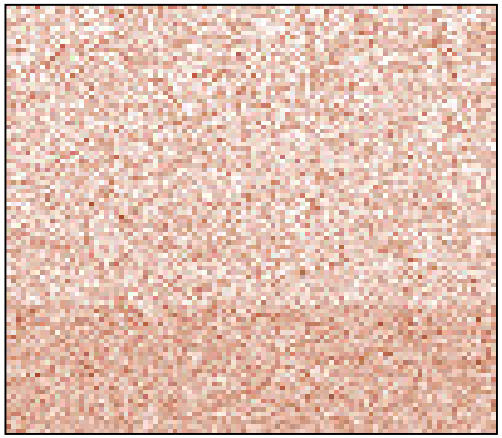
Initial state



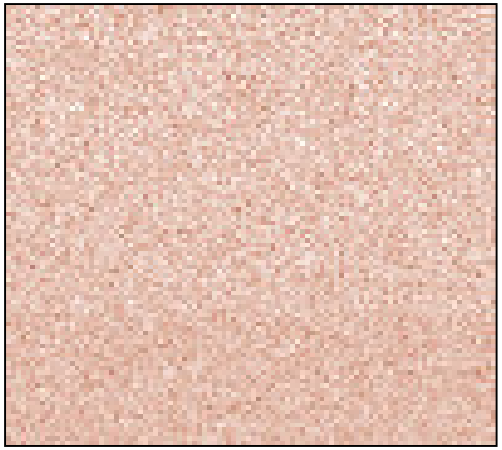
Cycle 1



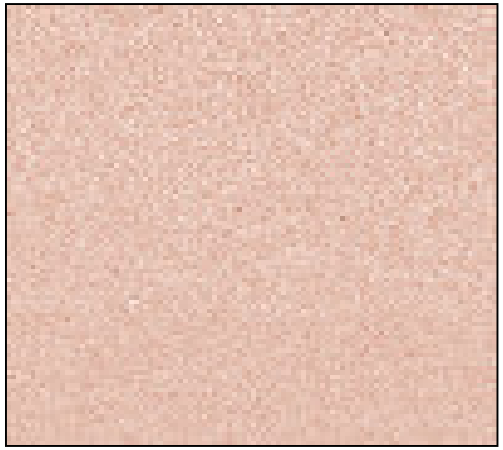
Cycle 2



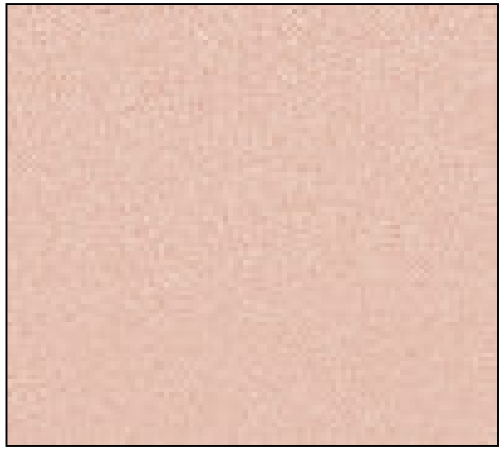
Cycle 3



Cycle 4



Cycle 5



The base theorem

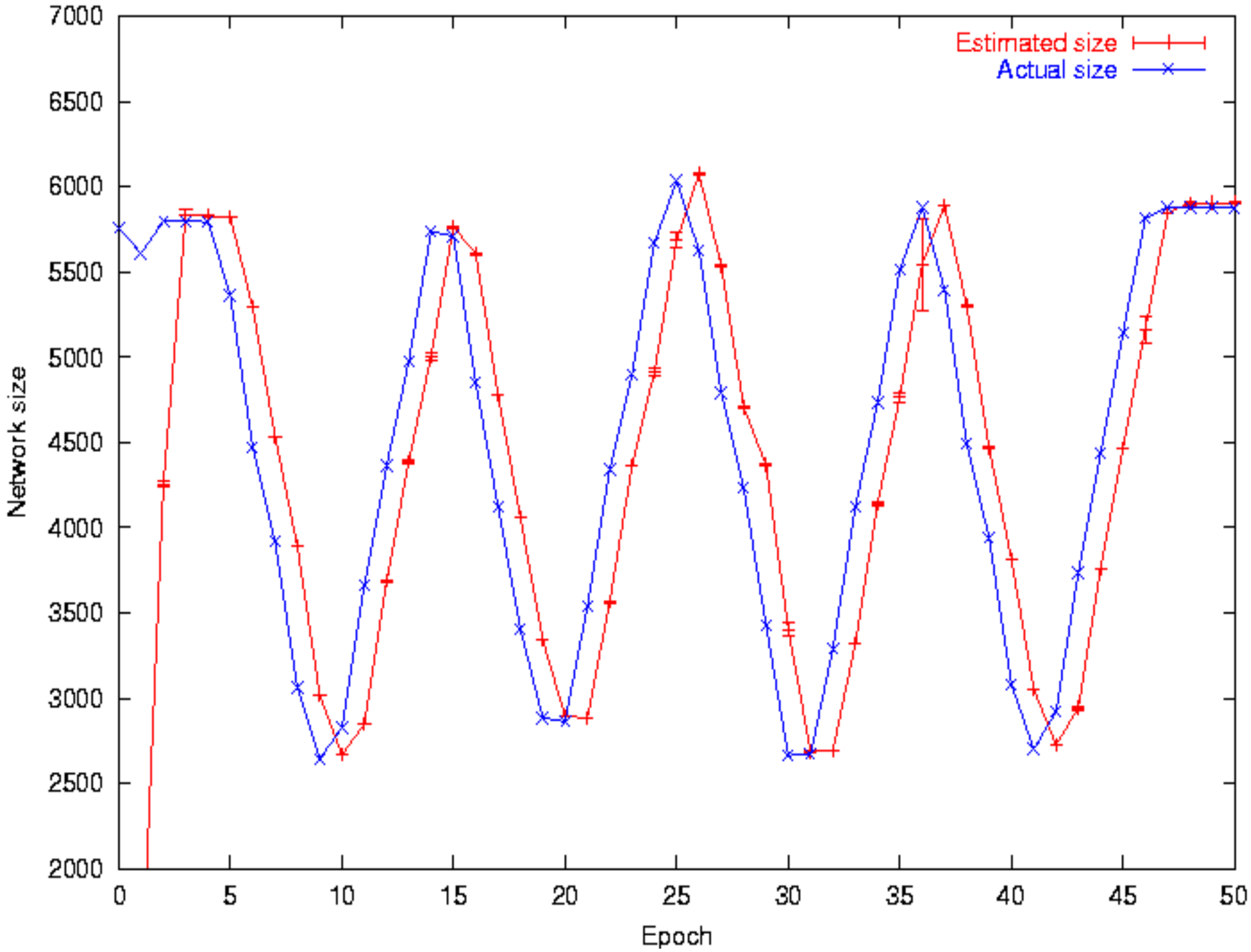
$$E(\sigma_{i+1}^2) \approx E(2^{-\varphi})E(\sigma_i^2)$$

Where φ Is the random variable that defines the number of times a random node participates in an information exchange during a cycle.

Convergence factor

It follows that if the underlying overlay network is random then

$$P(\varphi = j) = \frac{1}{(j-1)!} e^{-1} \rightarrow E(2^{-\varphi}) = \frac{1}{2\sqrt{e}}$$


Estimated size
Actual size
Network size
Epoch
7000
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
0
5
10
15
20
25
30
35
40
45
50

Aggregation: Summary

- In case of averaging, the variance of the set of approximations decreases exponentially
- Extreme robustness to node and link failure and node dynamism (churn)