Network Science: Decentralized network formation

Ozalp Babaoglu
Dipartimento di Informatica — Scienza e Ingegneria
Università di Bologna
www.cs.unibo.it/babaoglu/

Decentralized network formation

- Network formation models such as Erdős-Rényi and Watts-Strogatz are "full information" models in that they build the network from a "bird's eye" point of view
  - The set of all nodes is known
  - It is static
- In practical settings, this is unrealistic because
  - The set of nodes is not known
  - The set of nodes is huge (think Internet)
  - The set of nodes is dynamic and continuously changing

Decentralized network formation

- In these settings, we want to impose algorithmic constraints similar to those of Kleinberg for the navigation problem
  - The degree of nodes is a constant c independent of the total number of nodes n
  - At any given time, nodes know only their immediate neighbors
  - The set of all nodes may change due to new nodes entering the system and existing nodes leaving the system either voluntarily or because of failures
- The dynamics leading to changes in the set of nodes is called *churn*

Decentralized network formation

- Many Internet-scale applications (file sharing, content delivery, collaborative work, network storage, etc.) rely on an overlay network that needs to be built and maintained under these circumstances
- How to construct an overlay network with properties similar to an Erdős-Rényi network in a totally decentralized manner (based on local information only)?

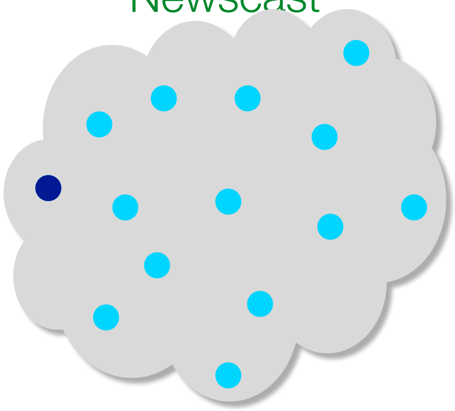
Newscast

- Decentralized protocol that creates and maintains a *random* overlay
- Highly resilient to churn
- Extremely simple design based on information ***gossip***:
  - Each node only knows the set of its immediate neighbors called its ***view***
  - Each node periodically picks a node at random from its view
  - The two nodes exchange their views and update them
- The random view exchange makes the algorithm very robust to failures and changes in the overlay

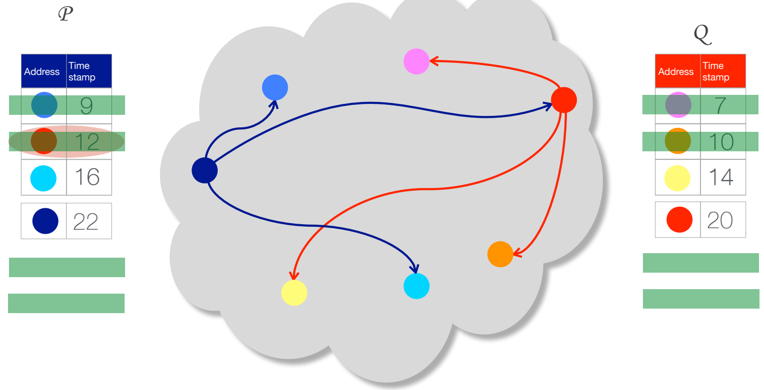
Newscast

- The view of a node consists of c *entries*, where c is a parameter and *entry*={*node address*, *timestamp*}
- Protocol proceeds in fixed-length time intervals called *rounds*
- At each round, node $\mathcal{P}$ executes the following steps:
  - select random entry, say Q, from local view
  - send to node Q the local view plus a fresh entry for self
  - receive view of node Q and merge it with local view
  - keep the c most recent entries as the new local view
- The view of a node changes at each round

Newscast

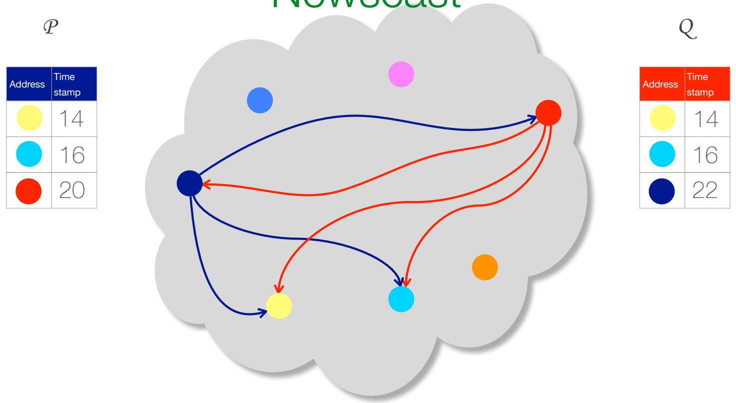


Newscast



1. Pick random peer Q from current view
2. Send each other current views plus fresh link to self
3. Keep c freshest links (remove own info, duplicates)

Newscast

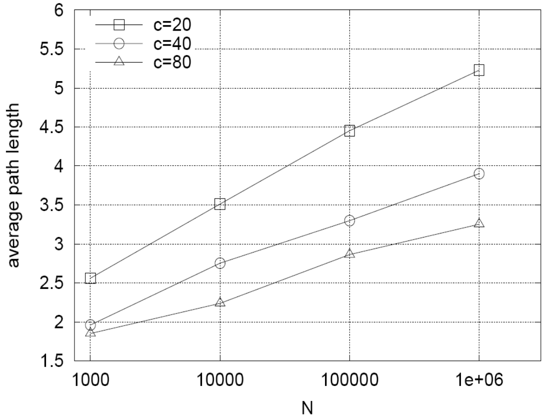


1. Pick random peer Q from current view
2. Send each other current views plus fresh link to self
3. Keep c freshest links (remove own info, duplicates)

Newscast

- Extremely robust to node and link failures and node dynamism (churn)
- Builds and maintains a connected, approximately random topology
- Has low overhead
- Is scalable

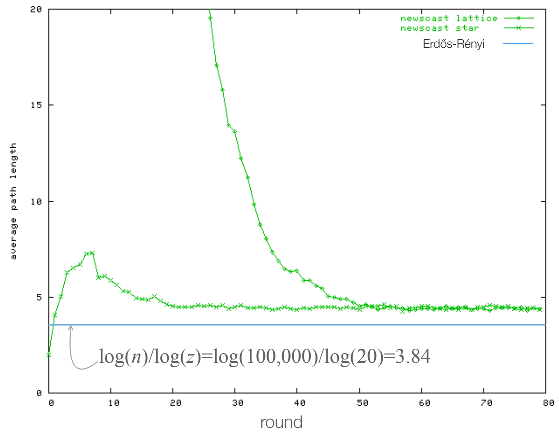
Newscast
Average path length: Network size



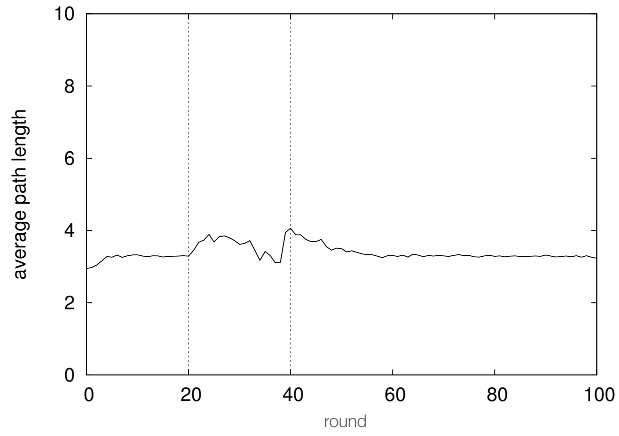
Newscast Experiments

- Simulation studies: 100,000 nodes, view size c=20
- Dynamic scenarios:
  - *Star*: start with no nodes, add 5000 nodes each round (connecting them to first node only)
  - *Lattice*: start with regular linear lattice (each node connected to k nearest nodes)
  - *Churn*: between cycles 20 and 40 replace 10% of nodes at each round

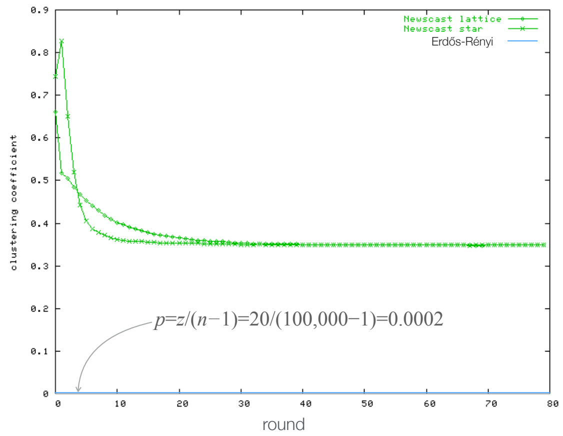
Newscast
Average path length: Convergence



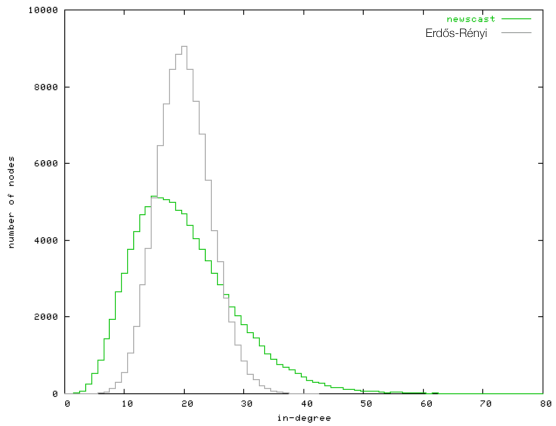
Newscast
Average path length: Churn



Newscast
Clustering Coefficient



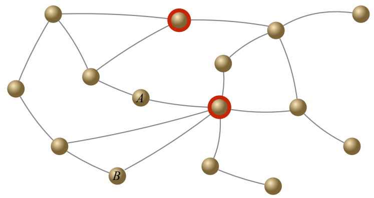
Newscast
In-Degree Distribution



Robustness of networks

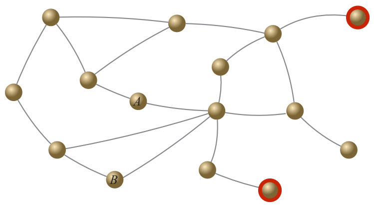
- Network components (nodes, links) can either fail or can be attacked to cause maximum harm
- Model *failures* by removing nodes or links selected at random — included in churn
- Model *attacks* by removing nodes that have the largest degree and links that are bridges
- Measures of robustness based on
  - how removal of nodes/links affects connectivity
  - how removal of nodes/links affects average path length
- Depending on the severity of these effects, label networks as being more or less *robust* or *resilient*

Robustness of networks



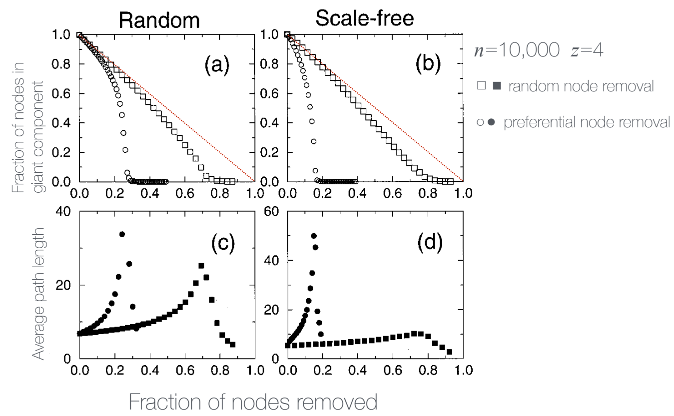
- Suppose two nodes fail
- Distance between A and B increases from 2 to 5
- Fraction of nodes in the giant component decreases from 100% to 40%

Robustness of networks

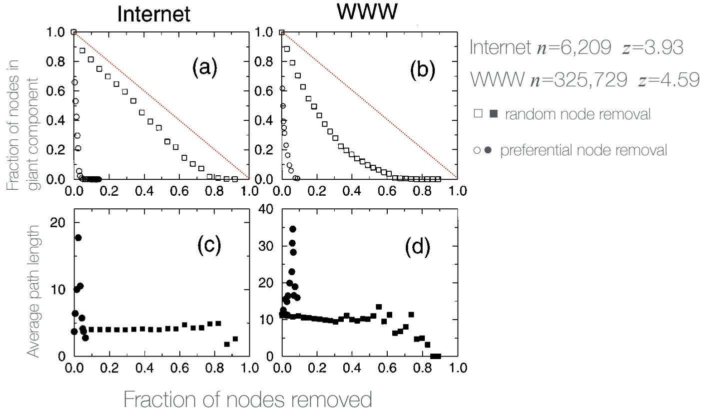


- Situation would have been very different if two other nodes were to fail
- Distinguish two different failure models:
  - *Random node removals* (model normal failure scenarios)
  - *Preferential node removals* (model "attack" scenarios)

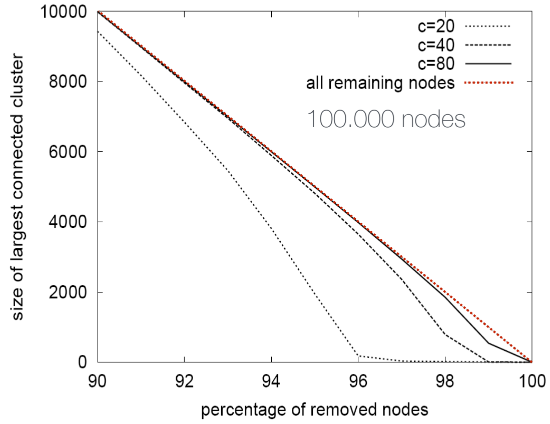
Robustness of networks



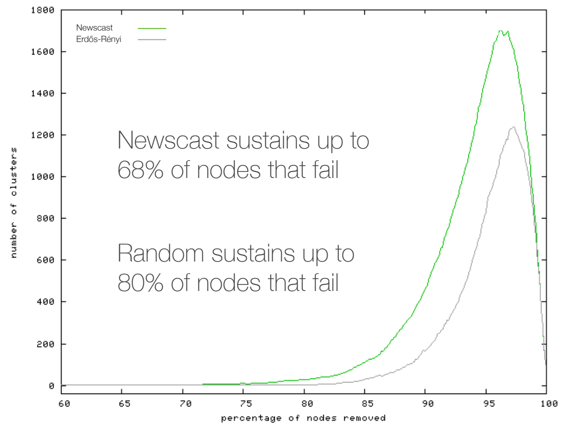
Robustness of networks



Newscast Robustness



Newscast Robustness



Newscast sustains up to 68% of nodes that fail

Random sustains up to 80% of nodes that fail

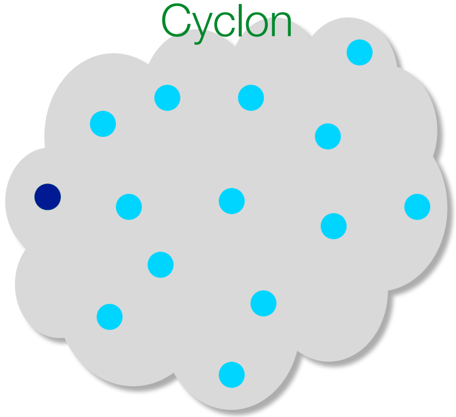
Newscast summary

- Newscast generates networks that are
  - Resilient to churn
  - Fairly robust
  - In-degree distributions close to binomial
  - High clustering coefficient
  - Small average path length (diameter) — small world
- High clustering coefficient is bad for:
  - Flooding algorithms: result in many redundant messages
  - Robustness: large clusters weakly connected to the rest of the network

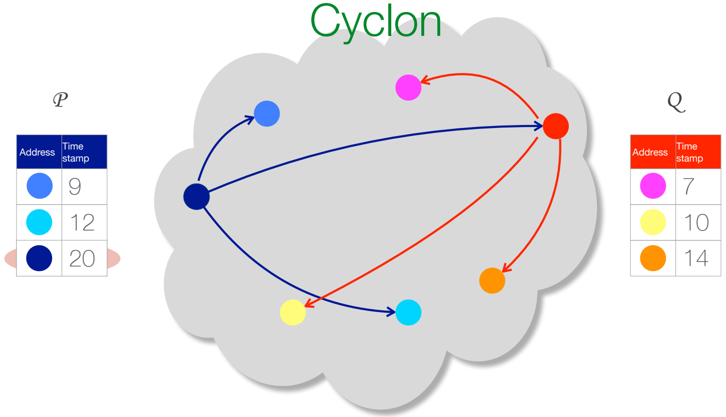
Cyclon

- How to modify Newscast to generate networks with small diameter and small clustering coefficients?
- Idea: instead of node $\mathcal{P}$ picking a random node from its current view, it picks the oldest node, say $\mathcal{Q}$
- Replace $\mathcal{Q}$'s entry at $\mathcal{P}$ with self and current time
- Exchange up to ℓ random entries from view with $\mathcal{Q}$
- Discard self entries, keep all others, first by filling empty slots, then replacing entries among the ones sent to $\mathcal{Q}$

Cyclon

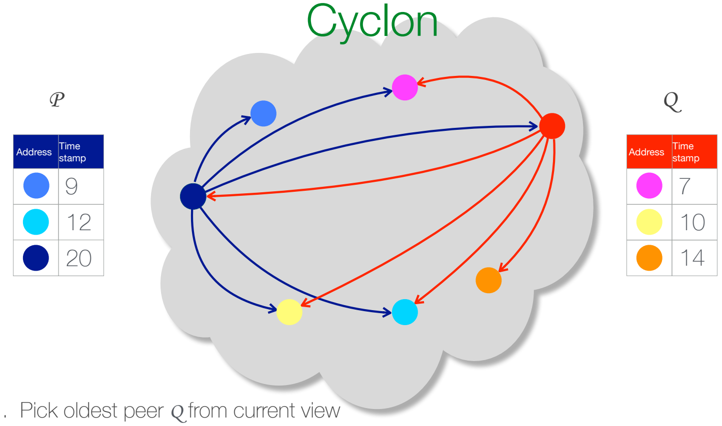


Cyclon



1. Pick oldest peer $\mathcal{Q}$ from current view
2. Replace $\mathcal{Q}$'s entry at $\mathcal{P}$ with self and current time

Cyclon

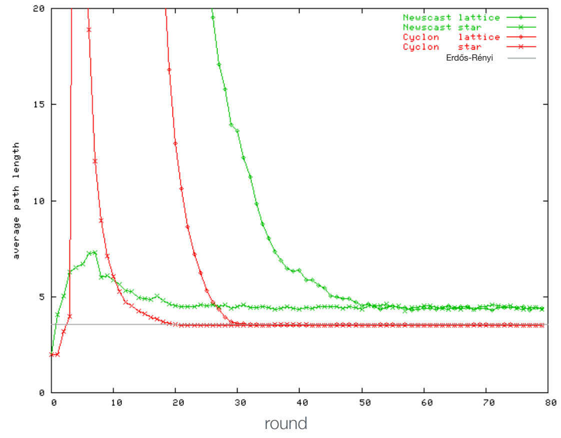


1. Pick oldest peer $\mathcal{Q}$ from current view
2. Replace $\mathcal{Q}$'s entry at $\mathcal{P}$ with self and current time
3. Exchange up to ℓ random entries with $\mathcal{Q}$
4. Update current view by filling empty slots and replacing entries among the ones sent to $\mathcal{Q}$

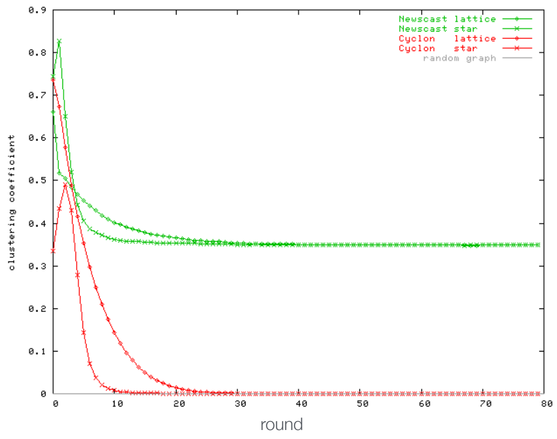
Cyclon
Advantages

- Connectivity is always guaranteed
- Uses less bandwidth — Only small part of the view is exchanged

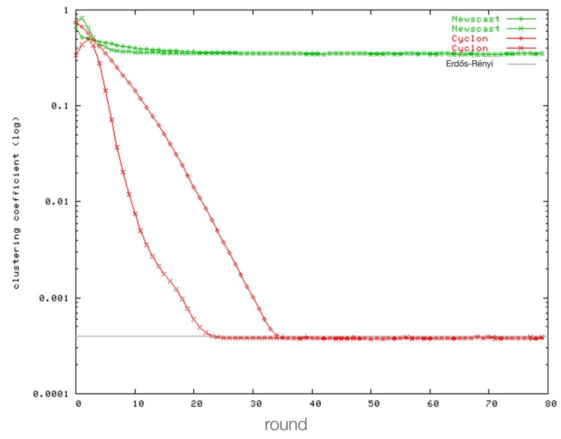
Cyclon
Average path length



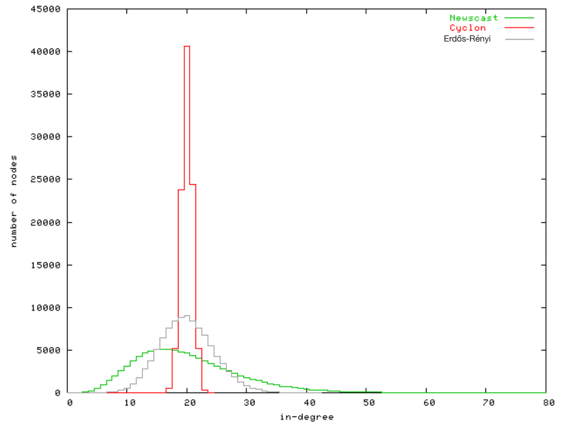
Cyclon
Clustering Coefficient



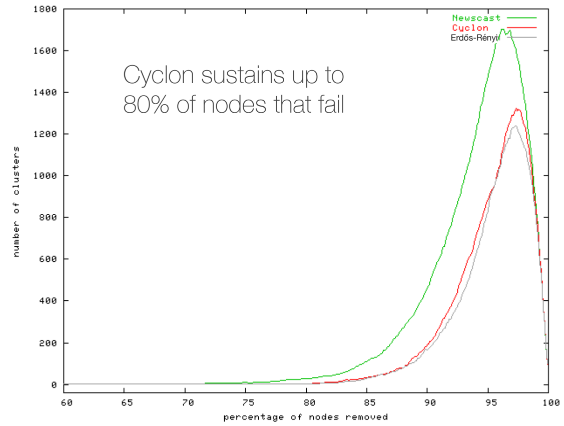
Cyclon
Clustering Coefficient (log scale)



Cyclon
In-Degree distribution



Cyclon
Robustness



Cyclon summary

- Small clustering coefficient
- Small average path length
- Cyclon generated graphs are closer to *random* (rather than *small world*)

Topology management

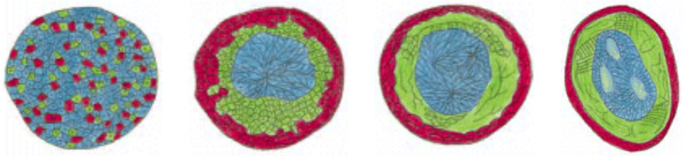
- Certain applications run more efficiently on a structured overlay network topology such as a mesh, ring, hypercube, etc.
- How to construct and maintain overlay networks that have desired topological properties in a manner that is
  - Decentralized
  - Self-organizing (insensitive to initial state)
  - Scalable (insensitive to network size)
  - Robust (insensitive to churn)
- If this *topology management* problem can be solved efficiently and rapidly, it can be used to satisfy application topological needs on-demand

Developmental biology

- *Morphogenesis* attempts to understand the processes that control the organized spatial distribution of cells during embryonic development and that give rise to the characteristic forms of tissues, organs, and overall body anatomy
- An interesting theory based on "differential cell adhesion"
  - different cell types "sort out" based on "likes" and "dislikes" for each other
  - any cell configuration has an energy level
  - cells try to minimize the free energy in the system by a stochastic movement process

Developmental biology

"Cells from different parts of an early amphibian embryo sort out according to their origins" (Townes & Holtfreter 1955)



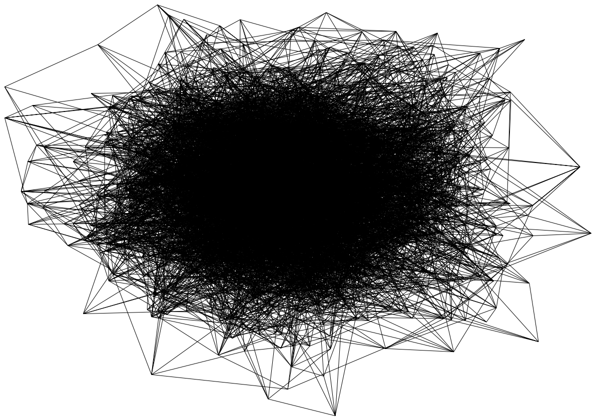
Back to topology management

- In biological systems, adhesion limited by physical constraints
- In overlay networks, we have the freedom to define peer relationships as we wish, resulting in a vast range of potential target topologies
- Notion of "like" and "dislike" captured by a *ranking function*
- Each ranking function defines a particular target topology
- Target topology can be changed on-the-fly by informing all nodes to start using the appropriate ranking function (after having distributed it)

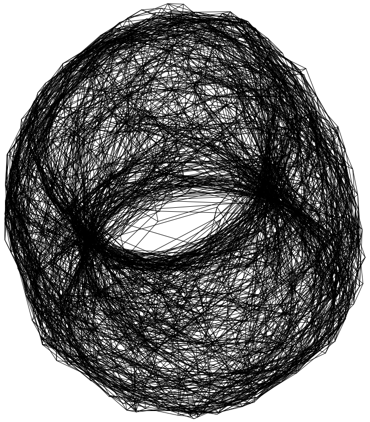
T-Man

- Each node maintains a local view of neighbors
- Periodically, each node exchanges its view with a peer selected at random among its current view
- Each node updates its local view by applying the ranking function to the union of the two views

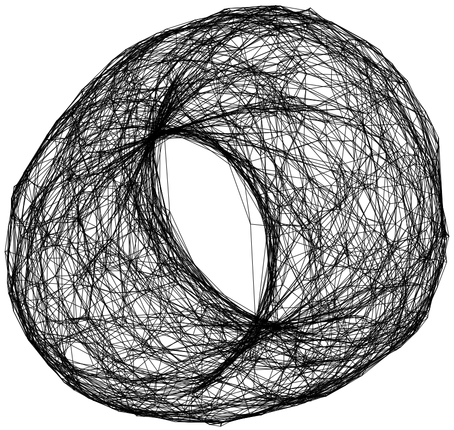
T-Man Torus example: initial state



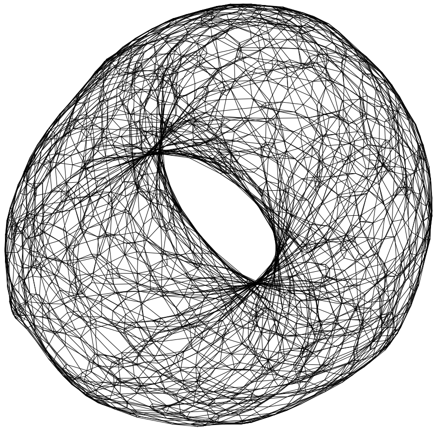
T-Man Torus example: after 3 rounds



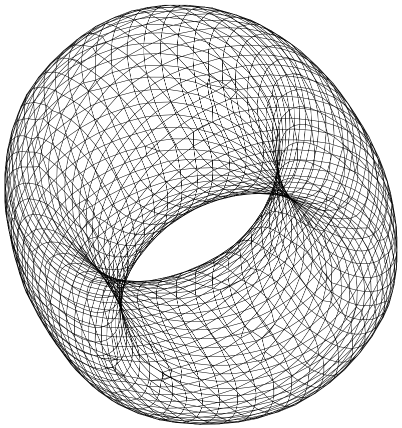
T-Man Torus example: after 5 rounds



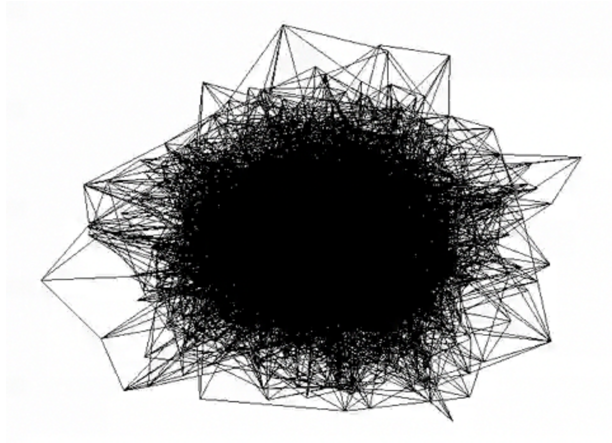
T-Man Torus example: after 8 rounds



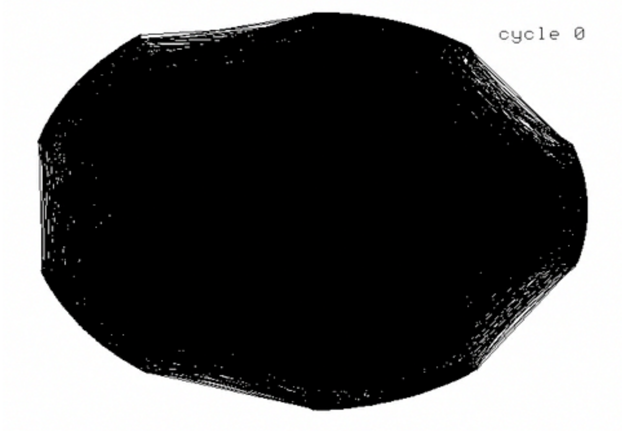
T-Man Torus example: after 15 rounds



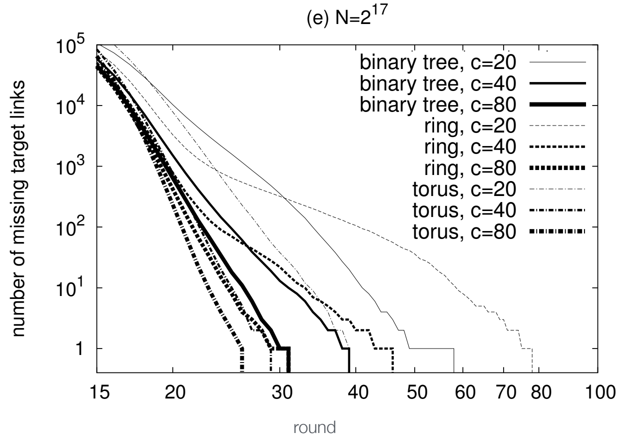
T-Man Torus example: The movie



T-Man
Sorting example



T-Man
Exponential convergence — time



T-Man

Exponential convergence — network size