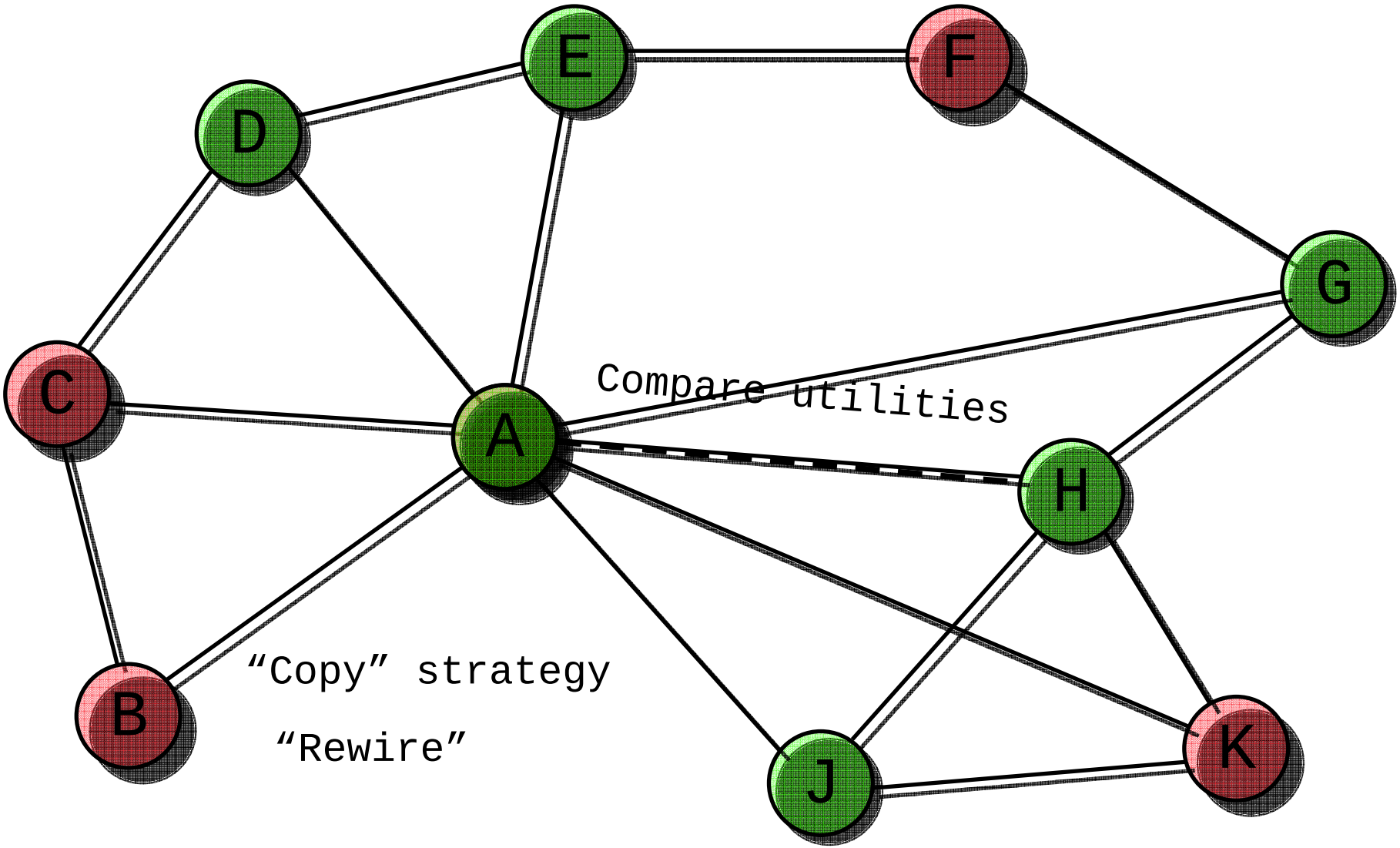
Greedy Cheating Liars and the Fools Who Believe Them

Stefano Arteconi, David Hales, Ozalp Babaoglu
University of Bologna, Italy

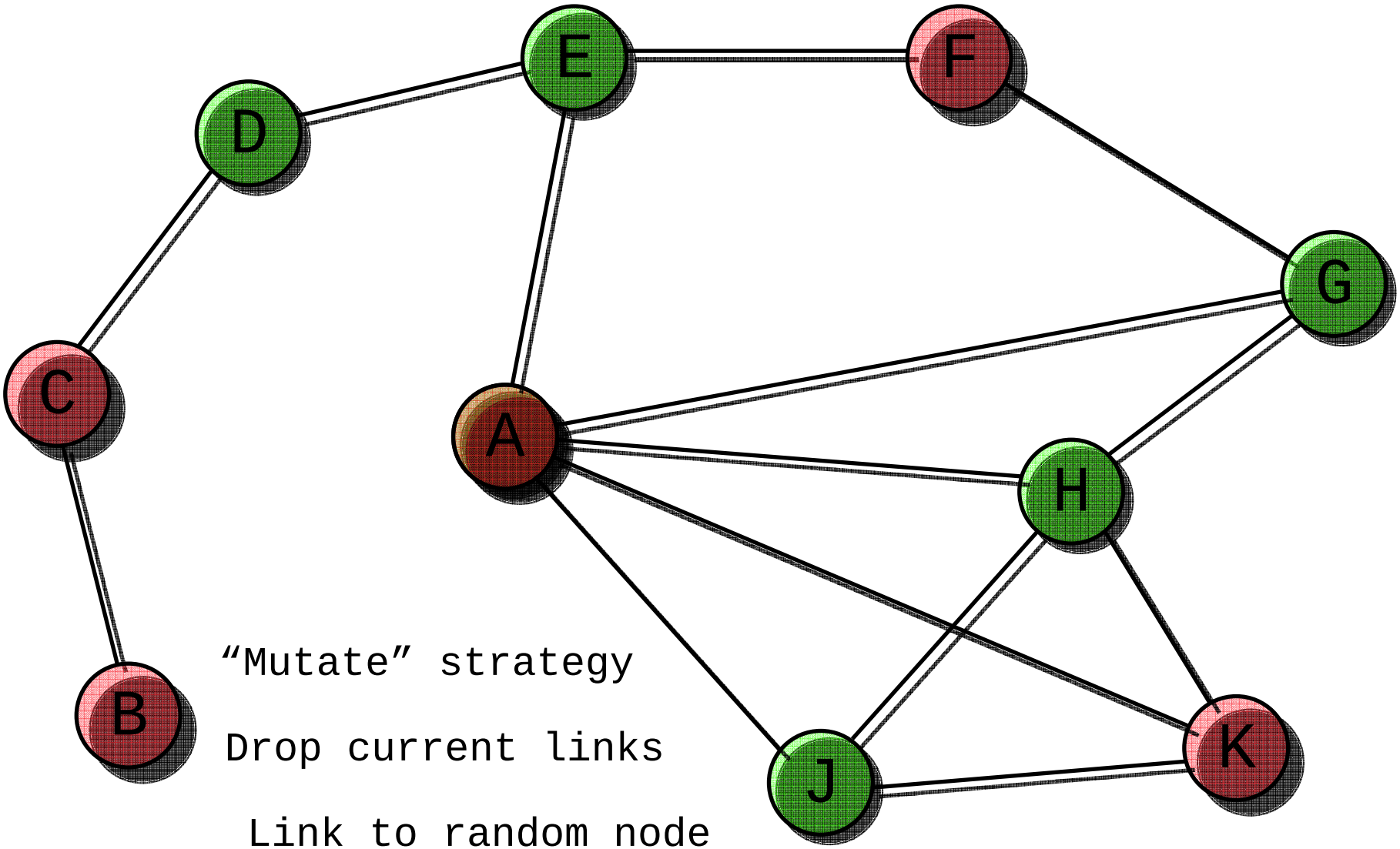
CSS-TW1, Bertinoro, 28/05/06

- P2P networks are usually open systems
  - Possibility to free-ride
  - High levels of free-riding can seriously degrade global performance
- SLAC algorithm sustains high levels of cooperation despite selfish nodes
- We show that certain types of cheating and lying behavior do not necessarily destroy cooperation (on the contrary, may even improve it!)

SLAC Algorithm: “Copy and Rewire”



SLAC Algorithm: “Mutate”



“Mutate” strategy

Drop current links

Link to random node

Prisoner's Dilemma

- We test SLAC with Prisoner's Dilemma (PD)
  - Captures the conflict between "individual rationality" and "common good"
  - Defection (*D*) leads to higher *individual* utility
  - Cooperation (*C*) leads to higher *global* utility
  - *DC* > *CC* > *DD* > *CD*
- Prisoner's Dilemma in SLAC
  - Nodes play PD with neighbors chosen randomly
  - Only pure strategies (always *C* or always *D*)
  - Strategy mutation: flip strategy
  - Utility: average payoff achieved

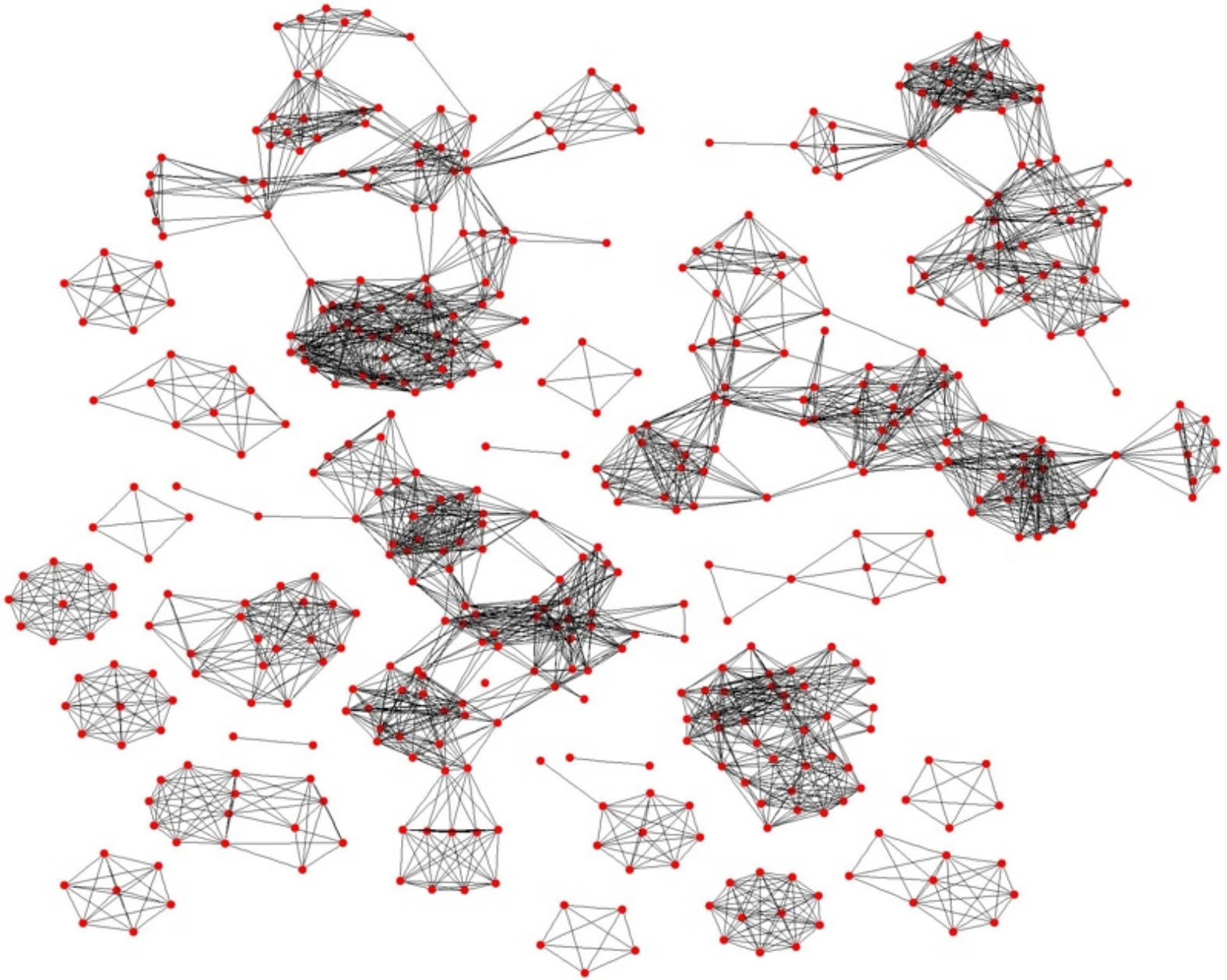
Why Does SLAC Work?

- SLAC produces very high levels of cooperation
- Nodes “move” throughout the network to find better neighborhoods
- This results in an evolution of the (interaction) network
- Group-like selection between clusters
  - Clusters of cooperating nodes grow and persist
  - Defecting nodes tend to become isolated

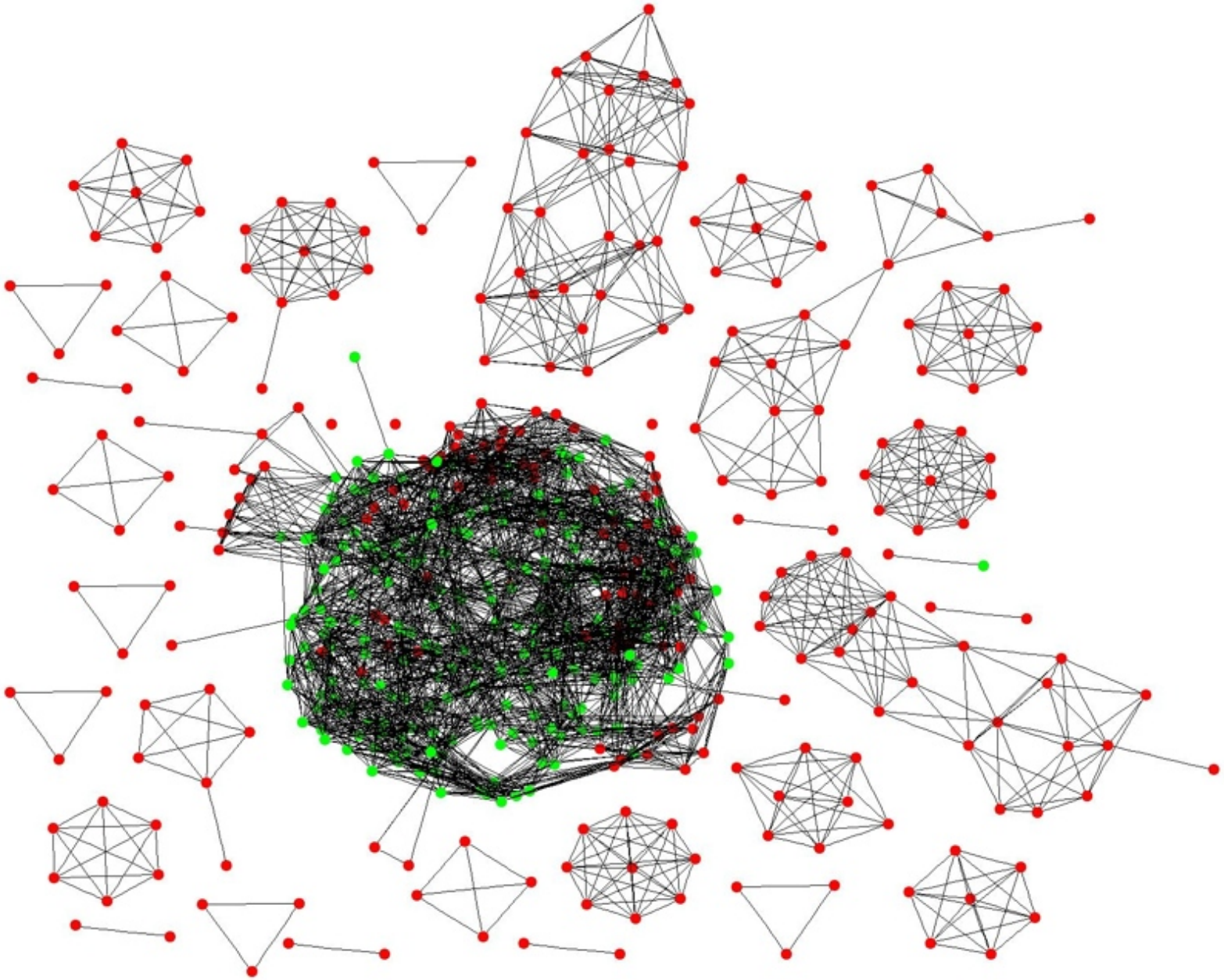
Example

- 500 nodes
- Initial state:
  - All defectors
  - Random interaction network

Cycle 180: Small Defective Clusters

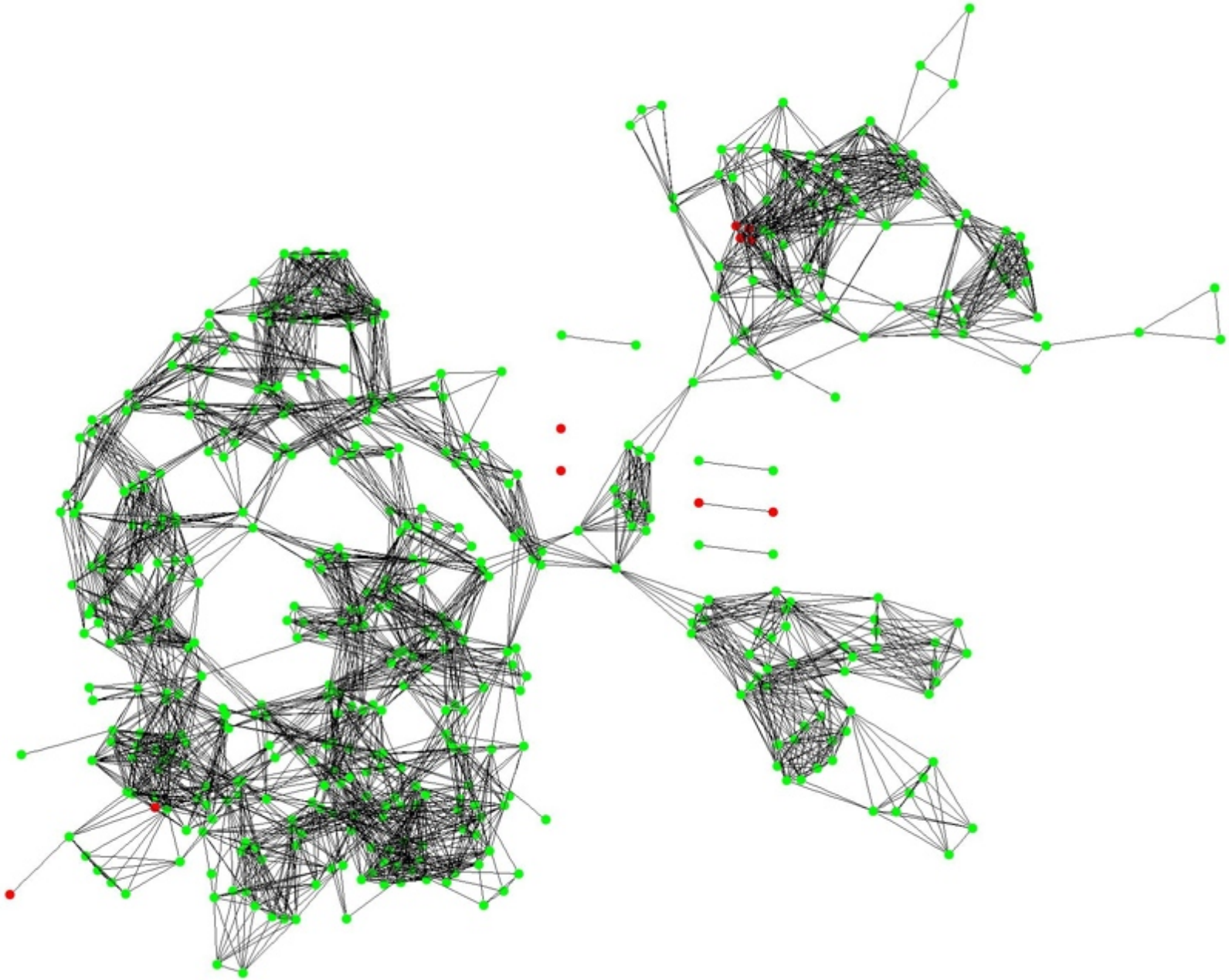


Cycle 220: Cooperation Emerges

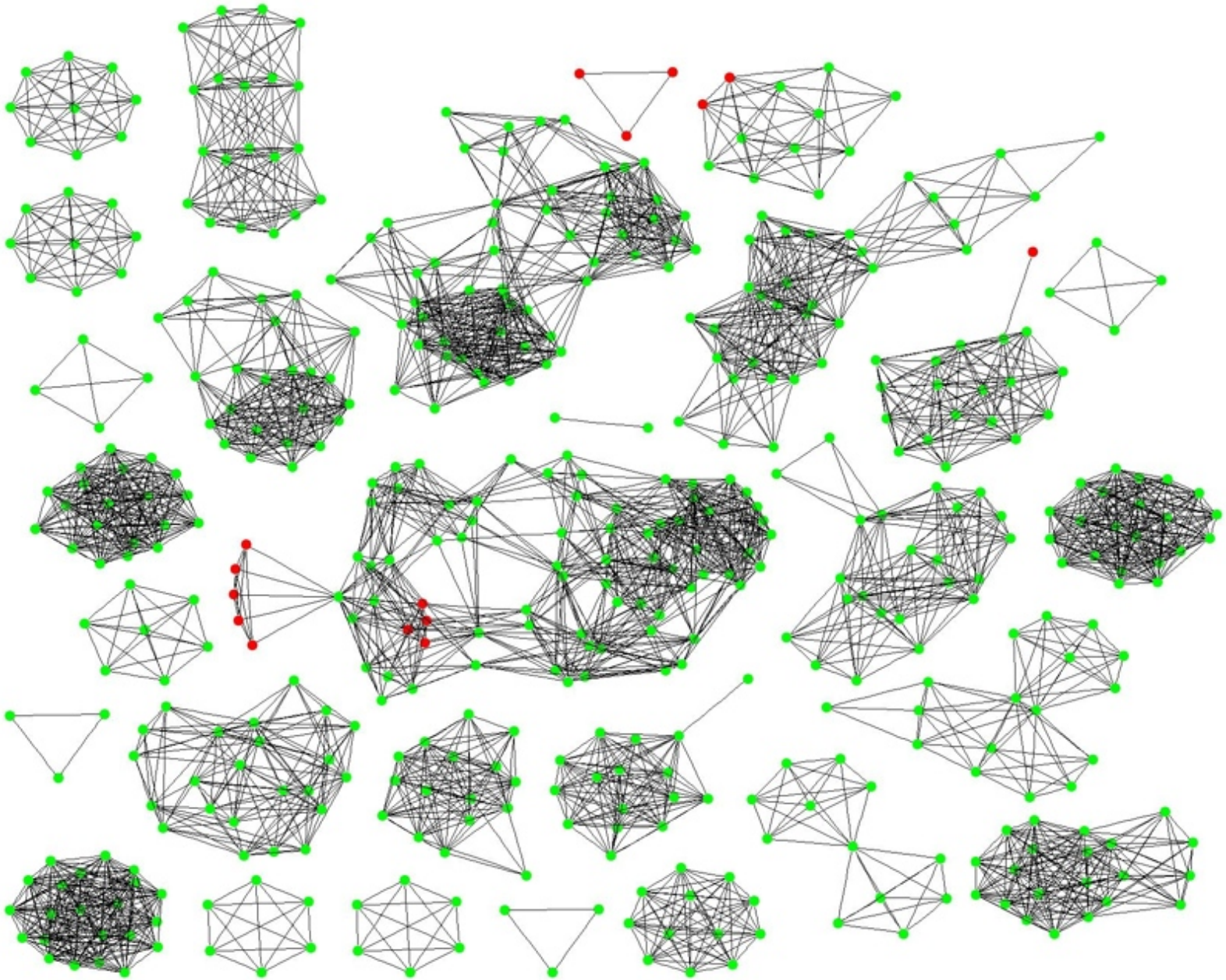


Cycle 230: Cooperating Cluster Starts to Break Apart

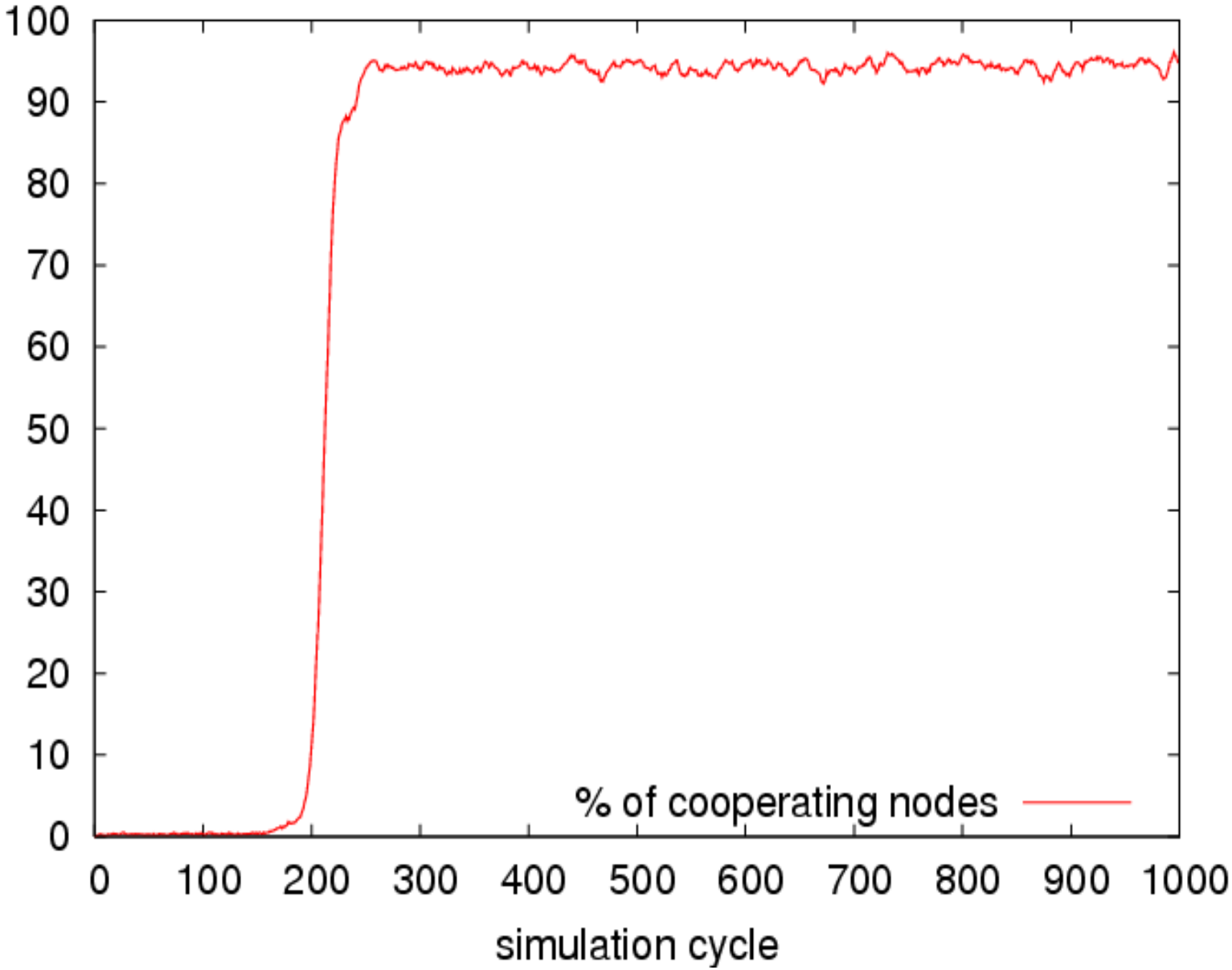
Stefano Arteconi



Cycle 300: Defective Nodes Isolated, Small Cooperative Clusters Formed



Cooperation Trend

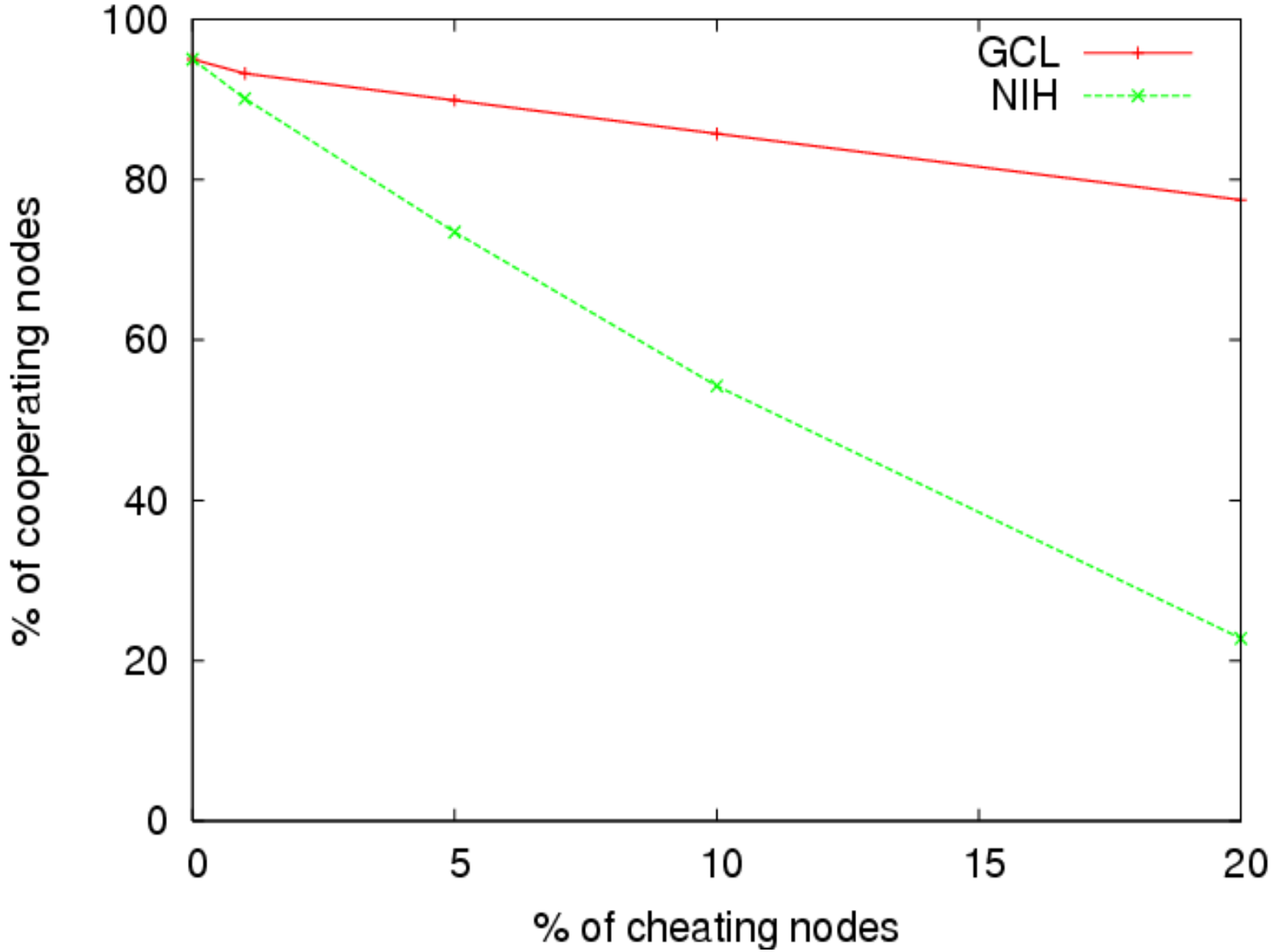


- SLAC requires nodes to honestly report their states (strategy, utility, links)
- What happens if some of the nodes lie in an effort to cheat the system? Will this destroy cooperation?
- We consider two types of cheating:
  - Greedy Cheating Liars (GCL) that want to exploit the system in order to increase their utilities
  - Nihilists (NIH) that want to destroy cooperation in the system and don't care about their own utilities

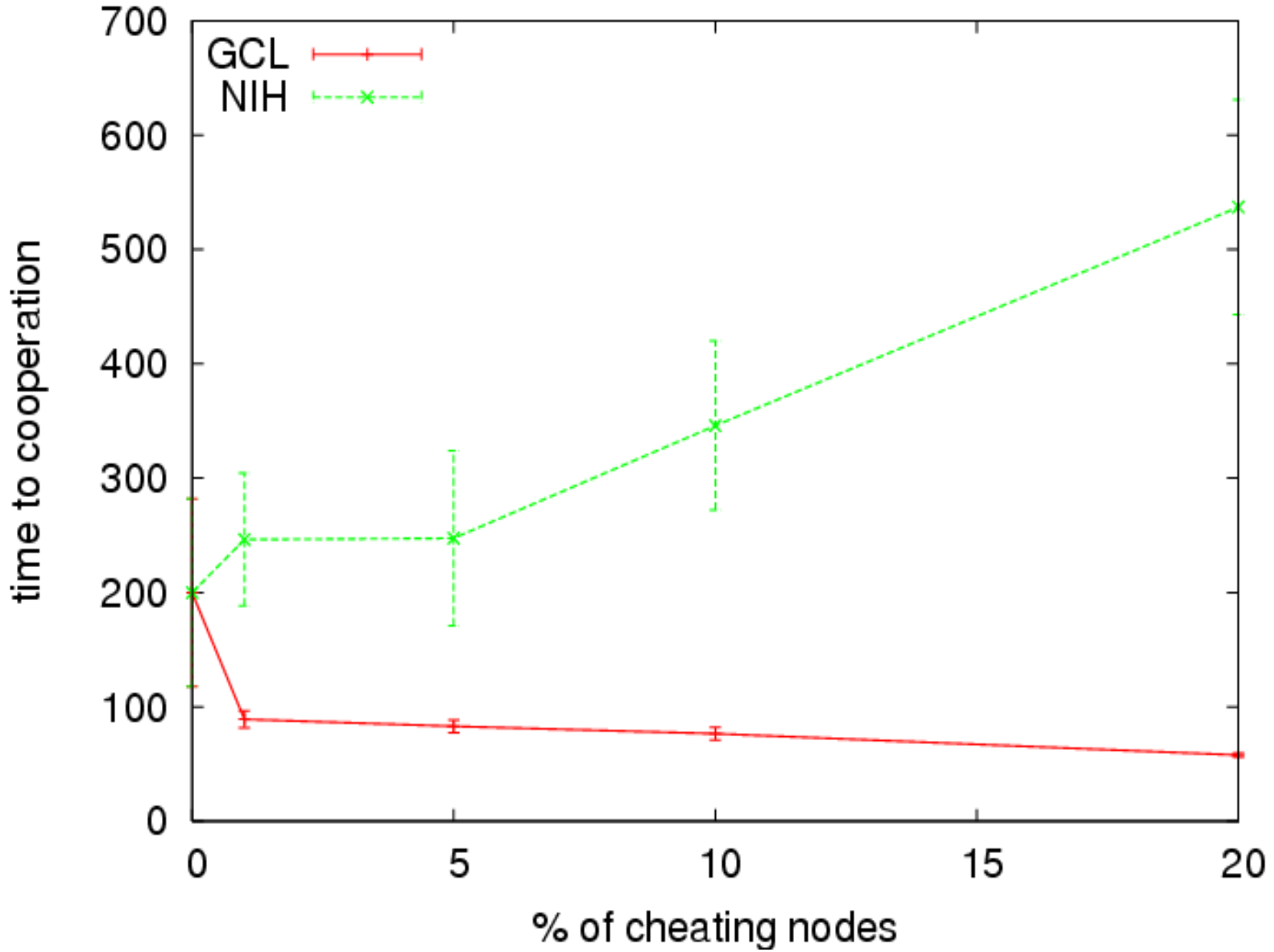
- GCL nodes:
  - Always report high utility (lying)
  - Always report strategy *C* (lying)
  - Always play strategy *D* (cheating)
  - Move away when they are surrounded by only defectors
- In this manner, GCL nodes try to surround themselves with cooperating nodes to exploit

- NIH nodes:
  - Always report high utility (lying)
  - Always report strategy *D*
  - Always play strategy *D* (cheating)
  - Move away when they are surrounded by only defectors
- In this manner, NIH nodes try to turn cooperating nodes to defectors

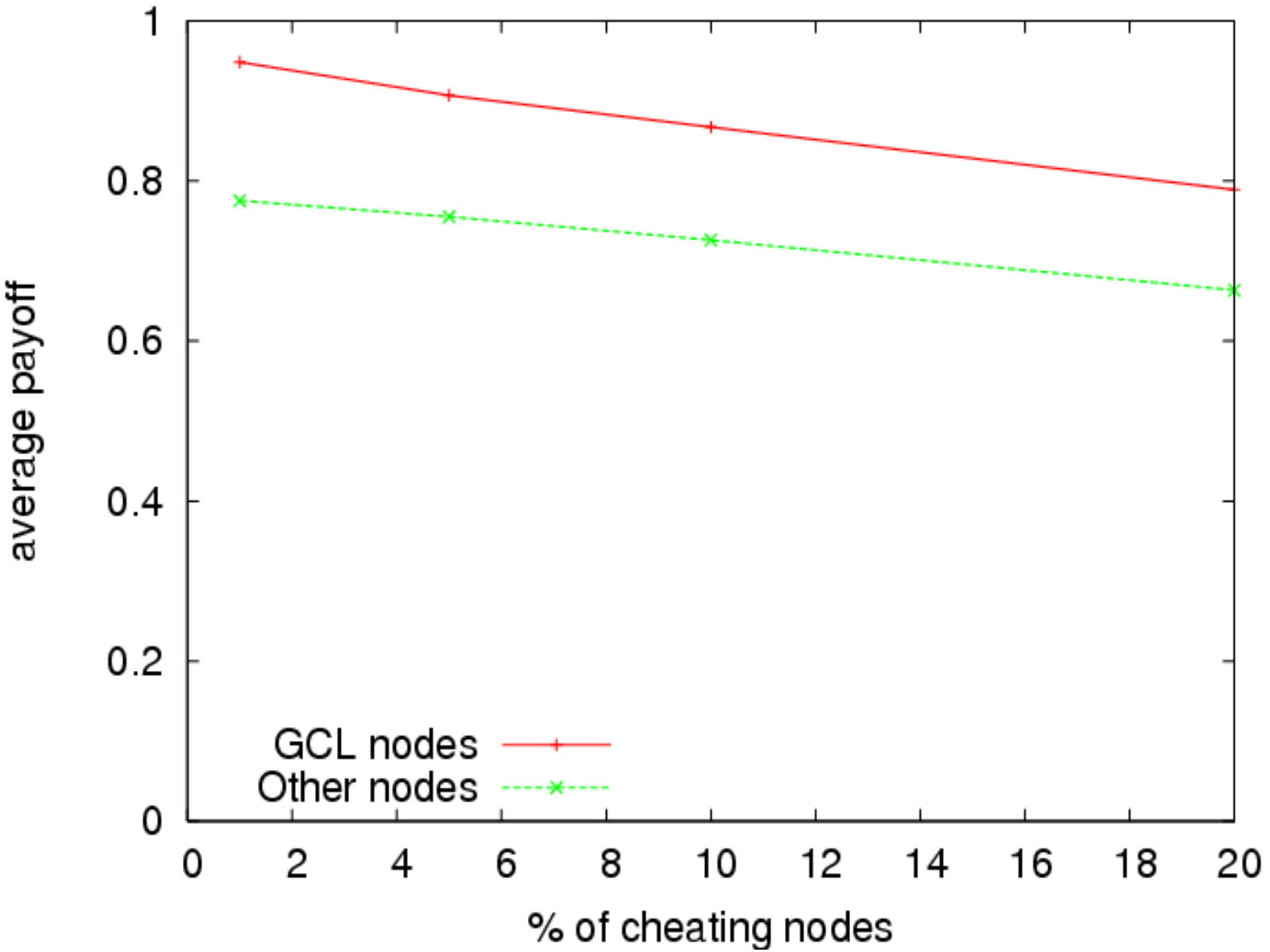
Cooperation in the Presence of Cheaters



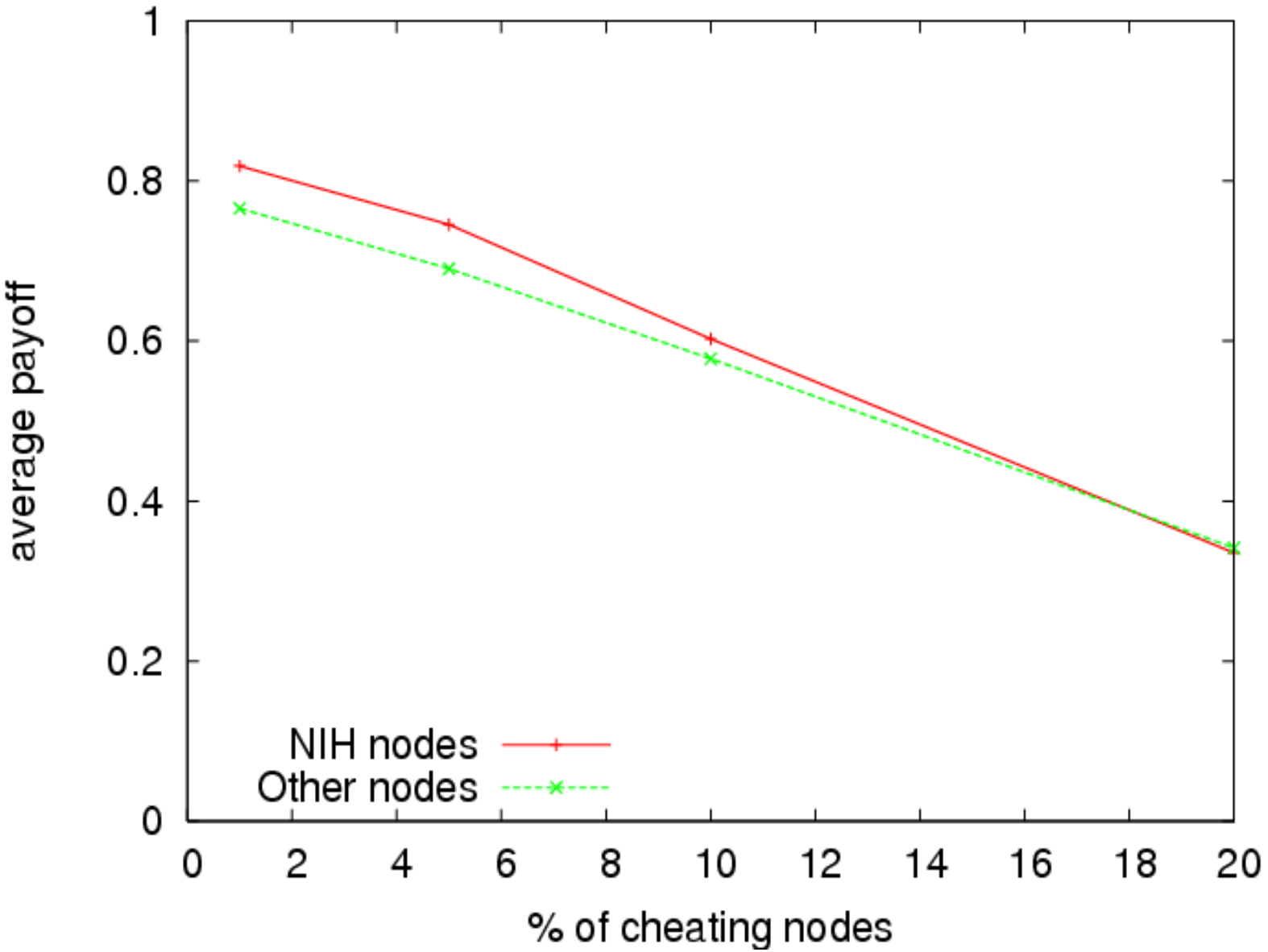
Time to Cooperation with Cheaters



Utilities in the Presence of GCL Nodes



Utilities in the Presence of NIH Nodes



- SLAC can tolerate a high percentage of GCL nodes
  - GCL nodes degrade global performance gracefully
  - Interestingly, increasing percentage of GCL nodes *decreases* the time to cooperation
  - GCLs can be seen as “taxing” the general population in return for more rapid cooperation
- Yet, NIH nodes degrade performance significantly
- Perhaps protocols can be designed to function despite cheating nodes rather than strive to detect and block them

- Copying (and mutation) applied to normal behavior
- Cheating behavior limited to a (fixed) percentage of nodes and does not spread
  - “Normal behavior” akin to running good clients in a P2P system (like BitTorrent)
  - “Cheating behavior” akin to running hacked versions of the P2P client
  - Typically, these hacked versions remain limited to a small group “in the know” and are not made widely available to others

Some Final Comments

- SLAC – simple algorithm based on copying and rewiring
- Induces cooperative behavior even in selfish environments
- Not based on notions of trust and reputation
- As such, no need for maintaining histories of past interactions
- Graceful degradation in the presence of greedy cheating liars
- Rapid degradation in the presence of Nihilists

Bibliography

- D. Hales, S. Arteconi, O. Babaoglu. SLACER: Randomness to Cooperation in Peer-to-Peer Networks. In *Proc. of StoDiS 2005: 1st Workshop on Stochasticity in Distributed Systems*, San Jose, California, USA, December 2005.
- A. Marcozzi, D. Hales, G. P. Jesi, S. Arteconi, O. Babaoglu. Tag-Based Cooperation in Peer-to-Peer Networks with Newscast. In *Proc. of SOAS2005: Int. Conf. on Self-Organization and Adaptation of Multi-agent and Grid Systems*, Glasgow, Scotland, UK, December 2005.
- S. Arteconi, D. Hales. Greedy Cheating Liars and the Fools Who Believe Them. Technical Report UBLCS-2005-21. Available at: http://www.cs.unibo.it/pub/TR/UBLCS/2005/2005-21.pdf
- D. Hales, S. Arteconi. SLACER: A Self-Organizing Protocol for Coordination in P2P Networks. To appear in *IEEE Intelligent Systems*
- PeerSim Simulator and GCL protocols available at: http://peersim.sf.net