Scalable and Distributed Self-Healing Algorithms for Reconfigurable Networks

Amitabh Trehan Jared Saia

Department of Computer Science
University of New Mexico

CS UNM Student Conference, 2007

Self-healing.

- *Self-healing:* A phrase applied to the process of recovery (generally from psychological disturbances, trauma, etc.), motivated by and directed by the patient, guided often only by instinct. [Wikipedia]
- Our Goal?
  Make this concept concrete.

Self-healing.

- *Self-healing:* A phrase applied to the process of recovery (generally from psychological disturbances, trauma, etc.), motivated by and directed by the patient, guided often only by instinct. [Wikipedia]
- Our Goal?
  Make this concept concrete.

Our Problem

- Given: a connected network.
- Goal: Keep the network connected and "small".
- Problem: An adversary deletes nodes in the network.
- Technique: Add edges.

Outline

1. Introduction
   - Self-healing in face of attacks
   - Previous Work
2. Our Work
   - Our Model
   - DaSH: Algorithm
   - Experiments

- The network: a Graph G(V,E)
- The attack: Deletion of nodes.
- Self-healing goals:
    - Maintain connectivity.
    - Ensure degrees of all nodes stay small.
    - The algorithm must be efficient.

Outline

Reconfigurable Networks.

- Networks in which we can add new connections between nodes.
- Examples:
  - Peer-to-Peer (P2P) networks.
  - Cellular networks.
  - Ad-hoc networks.
  - Social Networks.

Reconfigurable Networks.

- Networks in which we can add new connections between nodes.
- Examples:
  - Peer-to-Peer (P2P) networks.
  - Cellular networks.
  - Ad-hoc networks.
  - Social Networks.

Applications

- Sensor Networks
  - Node: Sensor.
  - Edge: Communication link.
- P2P Networks
  - Node: Peer.
  - Edge: Communication link.
- Social Networks
  - Node: Person.
  - Edge: Social connection.

Applications

- Sensor Networks
  - Node: Sensor.
  - Edge: Communication link.
- P2P Networks
  - Node: Peer.
  - Edge: Communication link.
- Social Networks
  - Node: Person.
  - Edge: Social connection.

Applications

- Sensor Networks
  - Node: Sensor.
  - Edge: Communication link.
- P2P Networks
  - Node: Peer.
  - Edge: Communication link.
- Social Networks
  - Node: Person.
  - Edge: Social connection.

Applications

- Sensor Networks
  - Node: Sensor.
  - Edge: Communication link.
- P2P Networks
  - Node: Peer.
  - Edge: Communication link.
- Social Networks
  - Node: Person.
  - Edge: Social connection.

Outline

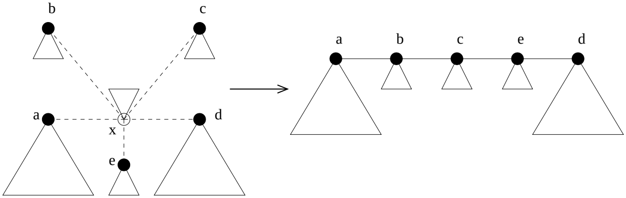
1. Introduction
   - Self-healing in face of attacks
   - Previous Work
2. Our Work
   - Our Model
   - DaSH: Algorithm
   - Experiments

Non-adaptible networks.

- Spare capacity and rerouting.[XM 1999]
- Redundant trees. [MFB 1999]
- Resilient Overlay networks. [ABKM '01]
- Independent redundant network components. [GBI '04]

Line Algorithm

- Reconnecting neighbours of deleted nodes in a line. [BASS '06].



Pluses

- Keeps degrees small.
- Ensures connectivity.
- Simple algorithm.

Problems

- Not scalable.
- Too many messages exchanged $O(n)$.
- Too slow $O(n)$.
- Diameter can increase.

Outline

1. Introduction
   - Self-healing in face of attacks
   - Previous Work
2. Our Work
   - Our Model
   - DaSH: Algorithm
   - Experiments

Our Model

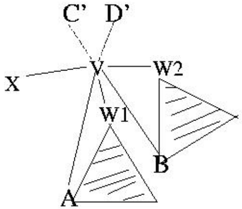
- The Adversary:
  - Eats Nodes.
  - Omniscient: has knowledge of our network and algorithms.
  - Eats one node at a time.
- The Home team (Nodes):
  - Have a small time to recover after each attack.
  - Can set up new links (reconfigure).
  - Maintain Neighbour-of-Neighbour information.

Outline

Some definitions

For a fixed time t:

- $G(V,E)$: The actual network.
- E': The edges added by algorithm.($E' \subseteq E$).
- $G' = (V,E')$: G' will be a forest.
- $N(v,G')$: neighbors of v in G'.
- $UN(v,G)$ (*Unique Neighbours*): Set of neighbours of v in G such that no subtree in G' has more than one representative.



DaSH: Degree-Based Self-Healing.

1. *Init:* for given network $G(V, E)$, Initialise each vertex with a random number *ID* between [0,1] selected uniformly at random.
2. **While** true **do**
3. *If a vertex v is deleted, do*
4. Nodes in $UN(v, G) \cup N(v, G')$ are reconnected into a *complete binary tree*. To connect the tree, go right to left, bottom up, mapping nodes to the *complete binary tree* in decreasing order of degree value.
5. Let *MINID* be the minimum *ID* of any node in $UN(v, G) \cup N(v, G')$. Propagate *MINID* to all the nodes in the tree of $UN(v, G) \cup N(v, G')$ in G'.
6. **end while**

DaSH Demo.

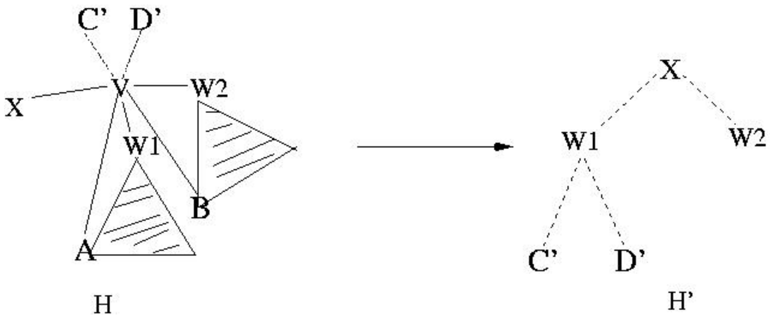
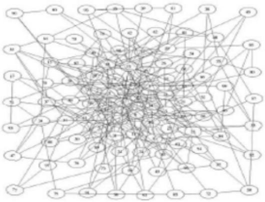
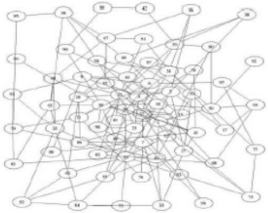


Figure: Reconfiguration on deletion of node V.

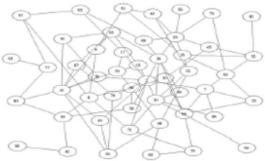
DaSH Timeline.



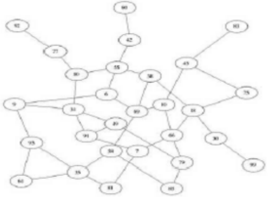
N=100



N=50



N=50



N=30

DaSH Properties.

Theorem

DaSH has the following properties:

- *The degree of any vertex is increased by at most* $2(log n) + 1$.
- *The latency to do healing after a deletion is constant.*
- *The number of messages any node sends out and receives is* $O(log n)$ *with high probability.*
- *The algorithm is completely distributed.*

Outline

- Attack strategies:
  - Max degree: Delete node of maximum degree.
  - Max Degree Neighbour: Keep deleting neighbours of maximum degree node.
- Healing strategies:
  - Binary Graph: reconnect all neighbours; naive.
  - Binary Tree: reconnect neighbours keeping G' as forest.
  - Degree based Binary Tree (DaSH)

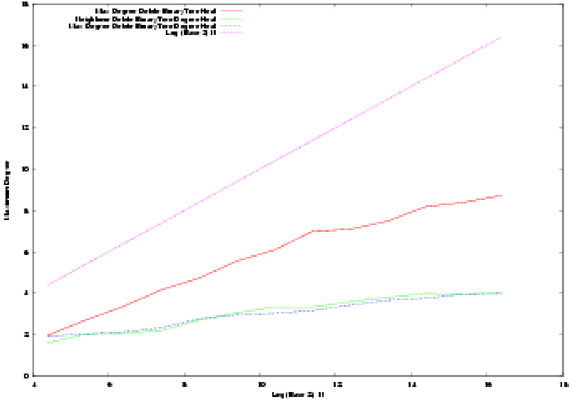


Figure: Self-healing demonstrated by *DaSH* and related Algorithms.

Summary

- Concrete definition of self-healing: maintaining an invariant over multiple attacks.
- Provably efficient algorithm for maintaining networks.

Future Work

- Additionally, keep Stretch[1] of the network low.

[1] maximum $\frac{\delta'(u,v)}{\delta(u,v)}$ for all nodes u, v, where δ' is distance in new graph, δ distance in original graph.

Question Time