Scalable and Distributed Self-Healing Algorithms for Reconfigurable Networks

Amitabh Trehan    Jared Saia

Department of Computer Science
University of New Mexico

IEEE International Parallel and Distributed Processing Symposium
Motivation

Skype network crashes
Self-healing

- **Self-healing**: A process of recovery motivated by and directed by the patient.

- Our Goal:
  Make this concept concrete.
Brain vs. Computer

- Components fail in both systems.
- Brain rewires automatically to maintain functionality.
- A computer does not rewire automatically.
Outline

1. Problem
2. Algorithm (DASH)
3. Theorems
4. Experiments
Outline

1. Problem
2. Algorithm (DASH)
3. Theorems
4. Experiments
Our Problem

- Given: A connected network.
- Goal: Keep the network connected and "small".
- Adversary: deletes nodes.
- Algorithm: adds edges.
The original network
The attack
After the attack
After the attack
After the attack
Network broken
Self-healing

Amitabh Trehan, Jared Saia
Heal thy self
Our Model

- **The Adversary:**
  - Omniscient: knows our network and algorithms.
  - Deletes one node at a time.

- **The Home team (Nodes):**
  - Have a small time to recover after each attack.
  - Can add new links (reconfigure).
  - Maintain neighbour-of-neighbour information.
Reconfigurable Networks.

- **Reconfigurable:** can add new edges.
- **Examples:**
  - Peer-to-Peer (P2P) networks
  - Social Networks
  - Ad-hoc networks
Applications

- P2P Networks
  - Node: Peer
  - Edge: Communication link

- Social Networks
  - Node: Person
  - Edge: Social connection

- Ad-hoc Networks
  - Node: Sensor
  - Edge: Communication link
Self-healing goals:
- Maintain connectivity
- Keep degree increase small
- The algorithm must be efficient: latency, bandwidth “small”
- Keep pair-wise distance increase small
Outline

1. Problem
2. Algorithm (DASH)
3. Theorems
4. Experiments
One approach

- Reconnect neighbours of deleted nodes in a line. [BASS ’06].
Pluses

- Keeps degrees small
- Ensures connectivity
- Simple algorithm
Problems

- Not distributed.
- Too many messages exchanged $O(n)$.
- Too slow $O(n)$.
- Distances may increase dramatically.
Some definitions

For a fixed time $t$:

- $G(V, E)$: The network.
- $E'$: The edges added by our algorithm ($E' \subseteq E$).
- $G' = (V, E')$: $G'$ will be a forest.
Definitions (Continued)

- \(N(v, G')\): neighbors of \(v\) in \(G'\).
- \(UN(v, G)\) (Unique Neighbours): Set of neighbours of \(v\) in \(G\) such that no tree in \(G'\) has more than one representative.
DASH: Degree-Based Self-Healing.

1. **Init:** Initialise each vertex with a random number $ID$ between $[0,1]$ selected uniformly at random.

2. **When a vertex $v$ is deleted do:**
   1. Nodes in $UN(v, G) \cup N(v, G')$ are reconnected into a complete binary tree sorted top-down in increasing order of degree increase.
   2. 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 $G'$. 
DASH Timeline: Deletion 10
DASH Timeline: Deletion 30
DASH Timeline: Deletion 50
DASH Timeline: Deletion 90
Outline

1. Problem
2. Algorithm (DASH)
3. Theorems
4. Experiments
Theorem

DASH guarantees the following properties even if up to all the nodes in the network are deleted:

- The network stays connected.
- The degree of any vertex is increased by at most \(2 \log n\).
- Number of messages any node of initial degree \(d\) sends out and receives is no more than \(2(d + 2 \log n) \ln n\) whp.
- The latency to reconnect is \(O(1)\) after attack; and the amortized latency to update the state of the network over \(\theta(n)\) deletions is \(O(\log n)\) with high probability.
- The algorithm is completely distributed.
Consider any locality-aware algorithm that increases the degree of any node after an attack by at most a fixed constant. Then there exists a graph and a strategy of deletions on that graph that will force the algorithm to increase the degree of some node by at least $\log n$. 
Observation

For a tree, deletion of a node of degree $d$ increases the sum total of degrees of its neighbors by $d - 2$ for a locality-aware acyclic healing strategy.
Outline of the proof

Prune (v,x) : Given node v and its subtree headed by node x, deletion of all the nodes in that subtree including x, despite self-healing. Accomplished by repeated deletion of leaf nodes in the subtree.
Outline of the proof

- Graph: M+2-ary tree for a M-bounded self-healing algorithm
- Attack Strategy: LEVELATTACK
Lower bound illustration (Ternary Tree, 2-Degree bounded (DASH))
Lower bound illustration
Lower bound illustration

Amitabh Trehan, Jared Saia
Heal thy self
Lower bound illustration
Strategies and Heuristics

- **Attack strategies:**
  - Max Degree (MD) delete: Delete node of maximum degree.
  - MD Neighbor delete: 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)
  - SDASH (Surrogate DASH): Let a node surrogate for the deleted node whenever feasible.
Degree increase: MD neighbor delete
Stretch and SDASH

**Stretch:** \( \max_{u,v,t} \left( \frac{\delta_t(u,v)}{\delta_0(u,v)} \right) \), where \( \delta_t \) is distance in graph \( G_t \), \( \delta \) distance in original graph \( G_0 \).

**Surrogation:** A neighbor of the deleted node takes over all the deleted edges.

**SDASH:** If degree increase using surrogation < degree increase using DASH, do surrogation.
Stretch: Max degree delete

Amitabh Trehan, Jared Saia

Heal thy self
Summary

- Concrete definition of self-healing: maintaining an invariant over multiple attacks
- Provably efficient algorithm for self-healing
- Provably ensure: connectivity and small degree increase
- Empirically small stretch
Future Work

- Keeping stretch of the network low

[Cartoon of a tree healing itself]

Amitabh Trehan, Jared Saia
Future Work

- Self-healing in sensor networks.
- Self-healing in social networks.
- Functional self-healing: maintaining functionality (circuit boards, the brain).
Question Time
Handling insertions

- Easily handled. The new node and neighbors simply update their data structures.
- True degree no more than degree if only insertions + 2log $n$ + 1.