Scalable and Distributed Self-Healing Algorithms for Reconfigurable Networks

Amitabh Trehan*†, Jared Saia†

†Computer Science, University of New Mexico

I. INTRODUCTION

Self-healing involves maintainance of basic properties of the network, such as connectivity, in face of an attack. Self-healing can occur on *Reconfigurable Networks*. Reconfigurable Networks are networks in which we can add new connections between nodes. Examples of such networks are peer-to-peer networks, cellular networks and ad-hoc networks . There has been significant past work on self-healing algorithms [1] [6] [4] [2].

In this paper, we introduce an algorithm for self healing of reconfigurable networks, called *DaSH*, which is an acronym for *Degree based Self-Healing*. We prove that *DaSH* is an efficient algorithm i.e. with low latency and small number of messages, which meets our objectives. We build upon the earlier work done in [3], which used a line algorithm for self-healing.

II. DASH: ALGORITHM

We define the following objectives for self-healing:

1. The network stays connected, after self-healing.
2. No node is overloaded, after the healing i.e. there is only a reasonable ($O(logn)$) increase in degree of any node in the network.
3. The Algorithm must be efficient i.e. it has $logn$ latency and $logn$ number of messages are exchanged.

Our algorithm uses only local information, therefore, it is scalable and can be implemented in a Distributed manner.

We assume our network is attacked by an adversary that deletes nodes. Our adversary is omniscient. Thus, it has knowledge of our network and our algorithms. However, we assume that it attacks one node at a time and after a deletion, there is a small amount of time in which the network detects the attack and tries to heal itself. Also, we assume nodes maintain Neighbour-of-Neighbour (*NoN*) information [5] i.e. if x and y are neighbours, then x knows all of y's neighbours. The nodes in the network, especially the neighbours of the deleted node may communicate with each other and set up links to heal the network. Our Algorithm *DaSH* is shown as Algorithm 1.

We introduce the following notations and definitions for *DaSH* and its analysis:

- The actual network at a particular time step is $G(V, E)$.
- Let E' be the edges (i.e. *healing edges*), that have been added by the algorithm up to that time step. (note $E' \subseteq E$).
- Let $G' = (V, E')$. Note that G' is a forest.
- Let $N(v, G)$ be the neighbors of vertex v in graph G.
- Let $N(v, G')$ be the neighbors of vertex v in graph G'.
- Let $UN(v, G)$ (*Unique Neighbours*) be the neighbours of v in G with ID not equal to ID of v such that no two members of $UN(v, G)$ have the same ID. If two or more neighbours of v have the same ID we include the one with the lowest initial ID. Note that $UN(v, G) \cap N(v, G') = \phi$ and $UN(v, G) \cup N(v, G') \subseteq N(v, G)$
- Let $T(v, x)$ be the tree in $G' - x$ that contains v.
- Each vertex v will have a weight, $w(v)$. The weight of a vertex will start at 1 and may increase during the algorithm. If v is deleted, $w(v)$ is added to an arbitrarily chosen neighbour in G'.
- Let $d'(v)$ be the degree of the vertex v in G'.
- For vertices v and x, let $W(v, x) = \sum_{v' \in T(v,x)} w(v')$
- For vertex v, let $rem(v) = \sum_{u \in N(v,G')} W(u, v) - \max_{u \in N(v,G')} W(u, v) + w(v)$

Algorithm 1 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) ∪ 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 d' value.
5:   Let MINID be the minimum ID of any node in UN(v,G) ∪ N(v,G').
     Propagate MINID to all the nodes in the tree of
     UN(v,G) ∪ N(v,G') in G'.
6: end while
```

Our main results about *DaSH* are stated in theorem 1.

Theorem 1. DaSH *has the following properties:*

- *The degree of any vertex is increased by at most* $2(logn) + 1$.
- *The latency to do healing after a deletion is constant.*

- *The number of messages any node sends out and receives is* $O(logn)$ with high probability.
- *The algorithm is completely distributed.*

Proof. We give only a brief sketch of the proofs here:

- *The degree of any vertex is increased by at most* $2(logn) + 1$.

  To prove this, we state here lemma 1, without proof.

Lemma 1. *For any node v,* $rem(v) \geq 2^{(d'(v)-1)/2}$

Corrolary 1. *The value* $rem(v)$*, at least, doubles each time* $v's$ *degree increases due to a deletion.*

Proof. When $v's$ degree goes from $j - 1$ to j or to $j + 1$, $rem(v)$ goes from $2^{\frac{j-2}{2}}$ to at least $2^{\frac{j}{2}}$, thus $rem(v)$ at least doubles. □

Lemma 2. *For all vertices* v*,* $rem(v)$ *is always no more than n.*

Proof. No vertex is counted twice in a rem value since the subtrees of a vertex are disjoint. Since the number of vertices in the subtrees cannot be more than the number of vertices remaining, the rem value is always no more than the sum of the weights of all undeleted vertices in G'.

We define W^* = sum of weights of all undeleted vertices in G'.

After initialization, $W^* = n$, since there are n vertices.
At each step of the algorithm, $W^* = n$, since the weight of the deleted vertex is added to one of the remaining vertices.
Thus, for node v, $rem(v) \leq n$. □

The following proves our theorem:
Every vertex v starts with $rem(v) = w(v) = 1$.

$$
\begin{aligned}
rem(v) &\geq 2^{(d'(v)-1)/2} \quad \text{by Lemma1} \\
2^{(d'(v)-1)/2} &\leq n \quad \text{by Lemma II} \\
(d'(v) - 1)/2 &\leq logn \quad \text{(by taking log of both sides)} \\
\Rightarrow d'(v) &\leq 2logn + 1
\end{aligned}
$$

- *The latency to do healing after a deletion is constant.*

  During the reconnection process, *DaSH* requires communication only between nodes one hop away, thus, the latency is a constant.

- *The number of messages any node sends out and receives is* $O(logn)$ *with high probability.*

  In *DaSH*, after the reconnections have been made, messages are sent out by nodes when the minimum ID has to be propagated. With similarity to the *record breaking problem*, it can be shown that *w.h.p.*, a node has its ID reduced only $logn$ times, where the record is the node's ID. These are the only messages the node needs to transmit or receive. Thus, it sends or receives $O(logn)$ messages.
- *The algorithm is completely distributed.*

  DaSH is completely distributed since it requires no global communication and can be implemented by nodes locally using local communication.

□

III. EXPERIMENTS

We carried out a number of experiments to test *DaSH* (also referred to by it's working name: *BinaryTree Degree Heal* in the figure). The adversial strategies included deletion of maximum degree node (*MaxNodeStrategy*), or deletion of neighbour of maximum degree node (*Max-NeighbourStrategy*. We implemented a number of healing strategies based on the idea of reconnecting neighbours of deleted nodes as binary trees or graphs. Our experiments confirmed that *DaSH* succeeded while naive algorithms failed to meet our objectives.

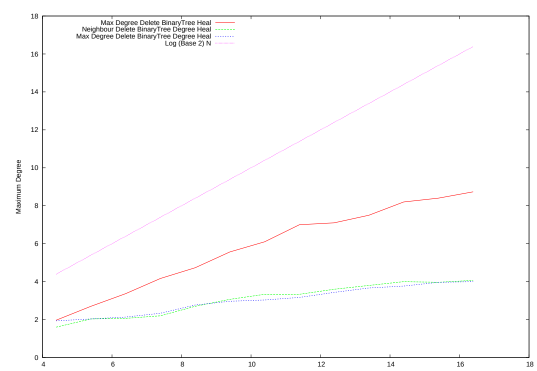

FIG. 1: Self-healing demonstrated by *DaSH* and related Algorithms.

[1] M. Amin. Towards self-healing energy infrastructure systems. *IEEE Computer Applications in Power*, 14:20–28, 2001.

[2] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. *SIGOPS Oper. Syst. Rev.*, 35(5):131–145, 2001.

[3] I-Ching C. Boman. Algorithms for self-healing networks. *M.S. Thesis, Computer Science, University of New Mexico.*, 2006.

[4] Sanjay Goel, Salvatore Belardo, and Laura Iwan. A resilient network that can operate under duress: To support communication between government agencies during crisis situations. *Proceed-*

ings of the 37th Hawaii International Conference on System Sciences, 0-7695-2056-1/04:1–11, 2004.

[5] Gurmeet Singh Manku, Moni Naor, and Udi Wieder. Know thy neighbor's neighbor: the power of lookahead in randomized p2p networks. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 54–63, New York, NY, USA, 2004. ACM Press.

[6] Yijun Xiong and Lorne G. Mason. Restoration strategies and spare capacity requirements in self-healing atm networks. *IEEE/ACM Trans. Netw.*, 7(1):98–110, 1999.