

A Framework for Containing the Degree Growth in Topological Self-stabilization

Thamer Alsulaiman¹, Andrew Berns², and Sukumar Ghosh¹

¹ University of Iowa

² University of Wisconsin-La Crosse

thamer-alsulaiman@uiowa.edu, aberns@uwlax.edu, sukumar-ghosh@uiowa.edu

1 Introduction

Overlay networks are built with logical links over one or more physical edges of a network. Logical links can be added or removed via program actions. Overlay networks mostly operate in fragile environments, and without central supervision. Bad configurations may be caused by node or link failures, by a node *join* or a node *leave*, or deliberate actions of nodes trying to derive undue performance benefits for themselves. Such adversarial actions may alter the network topology in an arbitrary manner. Topological self-stabilization takes a walk through the space of all networks defined by a given set of nodes, starting from a source network that is illegal and ending up at a target network that is legal. In the rest of the paper, we work under the assumption that the corrupted topology remains connected. We propose a framework that caps the degree growth to sublinear bounds while generating the target topology. As an illustration of the proposed technique, we present a self-stabilizing algorithm for building a heap.

2 The Main Idea

Background. The network topology $G = (V, E)$ is an undirected connected graph, where nodes have unique, positive identifiers. Each node $i \in V$ maintains a neighbor set $N(i)$ as a part of its local state, along with, maybe, other variables to help node i reach its goal. We assume a synchronous message-passing model: in each synchronous round, node i reads messages from previous round, performs some computations, and sends out zero or more messages to its neighbors. Neighbor set of a node may change over the course of computations. The efficiency of the detection of illegal topologies largely depends on the distribution of the detectors in the network. For overlay networks that are not locally checkable, there may not be a single detector even if the topology is illegal.

Given a faulty topology $G = (V, E)$, the *detector diameter* $D(G)$ for the given class of networks is the maximum hop distance in G between any node in V and its closest detector, where hop distance implies the shortest distance. The task of notifying every non-detector is time-efficient when the detector diameter is small.

Now, consider an instance of a computation where the topology transitions are represented by the sequence G, G_1, G_2, \dots, G_f , here G is the initial topology and G_f is the final legal topology. Maintaining large degrees, even for an interim period, is challenging for the nodes of any overlay network. This is why we would prefer the computation to steer the topology of the given network through the space of all “low degree” topologies with n nodes. The following definition quantifies the degree growth parameter:

Definition 1. Consider a topological self-stabilization algorithm \mathcal{A} that transforms a given initial topology G into a legal topology G_f . Let δ_{max} be the largest of all the node degrees between G and G_f . Then the degree growth of \mathcal{A} is contained, if in none of the intermediate configurations, the degree of any node exceeds $\delta_{max} + f(n)$, where $f(n)$ is sublinear.

The Framework. Our framework consists of three components. The first component uses a predicate **DETECT** to notify every process that the current configuration is not a legal one. The second component builds an interim **LINEAR** topology out of the given topology G . The third and the final component is a subroutine **REPAIR** that transforms the linear topology into a legal topology of the desired class.

The **LINEAR** network consists of all the nodes of a graph connected in the total order of their identifiers. We adopt the Pure Linearization algorithm from [2], since it caps the degree growth during stabilization. In [2], the degree of a node can increase by at most two in each round. We observe that the degree cannot keep increasing monotonically till the end – at some point the degree growth tapers off.

Theorem 1. For a given node, the degree growth in the Pure Linearization algorithm is bounded by $O(\delta + \sqrt{n})$, where δ is the initial degree of the node.

Further reductions in degree growth can be achieved by allowing nodes to linearize their entire neighborhood in each action, while preventing neighboring nodes from executing concurrent actions.

The **REPAIR** procedure starts after the linearization is over. The node with the largest id acts as the *leader*, and uses the linear chain to collect the identifiers of all the nodes in $O(n)$ rounds. Thereafter, the leader locally computes the legal topology, i.e. the ideal neighbor set $Nbr(i)$ for each node i , and disseminates them to every node i using the same linear pipeline. Each node i connects with the neighbor set $Nbr(i)$, which concludes the **REPAIR** phase.

Theorem 2. For any locally checkable topology, the stabilization time is $O(n)$ rounds.

Using the proposed framework, we illustrate how a binary max-heap topology can be stabilized. This topology is not locally checkable, but we add an extra variable per node to make it locally checkable. We demonstrate that the detector diameter of the heap topology is $O(\log n)$. Using the framework, one can stabilize a binary max-heap topology in $O(n)$ rounds while containing the degree growth. Note that without the degree cap, the heap can be stabilized in $O(\log n)$ rounds using the transitive closure framework [1].

3 Conclusion

To our knowledge, this is the first study of the degree growth containment problem in topological self-stabilization. Additional fixes for time-efficient stabilization from minor topology changes is left as a future exercise.

References

1. Andrew Berns, Sukumar Ghosh, and Sriram V. Pemmaraju. Building self-stabilizing overlay networks with the transitive closure framework. *Theor. Comput. Sci.*, 512:2–14, 2013.
2. Melih Onus, Andréa W. Richa, and Christian Scheideler. Linearization: Locally self-stabilizing sorting in graphs. In *Proceedings of the Nine Workshop on Algorithm Engineering and Experiments, ALENEX 2007, New Orleans, Louisiana, USA, January 6, 2007*, 2007.