

## Abstract

End-to-end communication is an essential part of any communication network. The main idea that has driven and still drives the further development of end-to-end communication in networks of every sort today, is the need for reliable, stable, and fast file transfer in a distributed network. The file size should also not be of any importance; we strive to guarantee reliable data transfer regardless of the amount.

Sensor networks have come to grow in size and popularity over the past decades. The need for a reliable way of collecting the data that these sensors measure, has grown with it. Sensors are inherently more prone to interference of any kind. On top of that, they are often used in harsh environments where the unreliable communication channel holds a risk for corrupting the data that the sensors collect. That is why we keep looking for new ways to improve the communication channel, and provide reliable transfer of the data that these sensors measure, to the collection point where the data is processed.

At the Chalmers University of Technology in Göteborg, Sweden, researchers have thought of a new algorithm that could potentially solve these problems of bad communication in unreliable networks. They provide a self-stabilizing, end-to-end communication in bounded capacity, omitting, duplicating, and non-FIFO dynamic networks.

We do our research towards finding any effort or literature that is working towards the practical implementation of such an algorithm. Next, we look into the background of distributed networking and wireless sensor networks. We explain the thought process behind the algorithm and the theoretical inner workings.

Before we start the practical implementation, we look at the Challenges ahead and formulate the central research question.

In the methods section, we provide a detailed look into the development process of the algorithm. We give an overview of the tools needed to complete this work, and the obstacles that had to be overcome.

The results form the key to this work, providing the reader with the proof that the algorithm works in a practical implementation, and not only on paper. We execute a number of simulations in different scenario's, and we try to identify the optimal value for the variables involved in the performance of the algorithm. The last part of this chapter focuses on the hard, solid proof. Using a debugging tool developed at Chalmers, we test the self-stabilization criteria and see if they hold up to the theory.

Finally we provide the reader with a short overview of what can be done with these results, and we propose a direction in what future research should go.