Threshold-Based Communication Algorithms for Large-Scale Mesh Networks

Thomas Meyfroyt

This Ph.D. thesis deals with the performance evaluation and design of communication algorithms for large-scale mesh networks. Important motivating examples throughout this thesis are sensor networks, such as lighting networks. Such networks typically consist of a large number recourse-constrained devices, which require simple communication algorithms that are able to maintain and disseminate data quickly, efficiently and reliably in a distributed fashion. The goal of this thesis is to analyze the performance of existing communication algorithms and to use the obtained insights for optimizing the design and deployment of these algorithms for large-scale networks.

An important commonality among the communication algorithms that are studied in this thesis, is that they all rely on a threshold-based suppression mechanism for controlling the number of messages being sent out by the nodes of the network. That is, in all algorithms, each node keeps track of the number of messages that were recently sent out either by itself or by neighboring nodes, and when this number exceeds a certain predefined threshold the node will temporarily suppress any outgoing messages. Analyzing the performance of such algorithms requires concepts from probability theory, graph theory and operations research, and this thesis combines techniques from all these fields. Additionally, throughout the thesis, we heavily rely on the use of asymptotic analysis, where we study the performance of these algorithms as the number of nodes in the network tends to infinity.

The thesis consists of two parts, each focusing on their own specific type of communication algorithm. In the first part, we analyze a wireless communication algorithm called Trickle. Trickle is a flooding algorithm designed to quickly propagate updates across a network of nodes. Under Trickle, each node in the network periodically schedules a transmission, which is only sent out if during that period the number of received transmissions from neighboring nodes is less than a predefined global threshold. While Trickle was originally designed for propagating and maintaining code updates in wireless sensor networks, it has shown to be a powerful mechanism that can be applied to a wide range of protocol design problems.

First, we study the message overhead and scalability of Trickle. We show that in a single-hop network Trickle bounds the number of messages per time unit, independent of the number of nodes in the network. Secondly, using Markov renewal processes, we analyze the speed at which Trickle can disseminate data in a multi-hop network, where the number of hops grows large. Additionally, we propose a simple extension of Trickle aimed at improving its propagation speed, without affecting scalability. Third, we also examine the performance of standardized communication protocols which have adopted the Trickle algorithm. We show that Trickle's broadcast suppression mechanism is intrinsically unfair in terms of load distribution and that this has adverse effects on the performance of these protocols. Another extension of Trickle aimed at overcoming these issues is proposed. Finally, we study a generalized version of Trickle, where a node's threshold is allowed to depend on the number of neighbors of that node. Assuming a synchronized operation, we show how to model this suppression mechanism as a special version of random sequential adsorption with nearest-neighbor interaction. Then, focusing on infinite-tree networks, we show how one can calculate the probability that the transmissions of certain nodes are suppressed. We also propose an algorithm which solves the inverse problem of determining how to configure the degree-dependent thresholds in infinite-tree graphs in order to reach some desired suppression probabilities.

In the second part, changing the focus from wireless to wired networks, we investigate token-passing algorithms. In token-passing networks, the nodes in the network cyclically pass around a virtual token. Only when a node holds the token, it is allowed to send out messages to other nodes in the network. The number of messages a node is allowed to send out before it has to pass on the token to the next node in the network is determined by the underlying service discipline. Our main focus is determining which service disciplines are best suited for deadline-critical networks. To this end, the performance of several service disciplines in large-scale symmetric polling systems is analyzed. Quantities of interest are the asymptotic queue length, cycle time and waiting-time distributions as the number of queues grows large. Additionally, we introduce a flexible k-limited service discipline, which extends the traditional k-limited service discipline and aims to achieve a predictable server cycle time, while also ensuring short waiting times.