Consensus with Unknown Participants or Fundamental Self-Organization

David Cavin, Yoav Sasson, and André Schiper

École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Switzerland
{david.cavin,yoav.sasson,andre.schiper}@epfl.ch

Abstract. We consider the problem of bootstrapping self-organized mobile ad hoc networks (MANET), i.e. reliably determining in a distributed and self-organized manner the services to be offered by each node when neither the identity nor the number of the nodes in the network is initially available. To this means we define a variant of the traditional consensus problem, by relaxing the requirement for the set of participating processes to be known by all at the beginning of the computation. This assumption captures the nature of self-organized networks, where there is no central authority that initializes each process with some context information. We consider asynchronous networks with reliable communication channels and no process crashes and provide necessary and sufficient conditions under which the problem admits a solution. These conditions are routing and mobility independent. Our results are relevant for agreement-related problems in general within self-organized networks.

1 Introduction

The paper addresses the problem of bootstrapping a self-organized MANET. More precisely, the paper addresses the following question. Consider some geographical region $R$ that is initially empty. At some point, one or more mobile nodes enter the region and want to deploy one or more services. However, to deploy the service(s), it is necessary for the first nodes that enter $R$ to agree on an initial set of nodes, in order for these nodes to decide which node is going to provide what service. Let us call these nodes $I$-nodes (Infrastructure nodes).

To decide which node is going to provide which service, we need to solve an agreement problem that decides on a set $I$-nodes, and outputs $I$-nodes at each node in $I$-nodes. Once this problem is solved, each node in $I$-nodes, based on the knowledge of $I$-nodes, can locally determine which node is responsible for providing what service(s). For example, assume that the agreement is on $I$-nodes $= \{n_1, n_2, n_3\}$, and consider that there are five services $s_1$ to $s_5$ to provide. Based on the knowledge of $I$-nodes, $n_1$ will provide the services $s_1$ and $s_2$, $n_2$ will provide the services $s_3$ and $s_4$, and $n_3$ will provide the service $s_5$. Or, if there is

* The work presented in this paper was supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant 5005-67322.
only one service to provide, \( n_1 \) will provide it, and \( n_2, n_3 \) know that they have no service to provide.

The problem is easily solved if there is some fixed node \( f_n \) that is always in \( R \): \( f_n \) can act as a centralized decision point. However, this solution is not self-organized, since it relies on some preexisting infrastructure. This leads to the following question: is it possible to solve the problem without any preexisting centralized infrastructure, i.e., in a fully self-organized way?

Deciding on the set I-nodes can be modeled as a consensus problem \([1]\). In the consensus problem, a set \( \Pi \) of processes have to agree on a common value (called the decision value) that is the initial value of one of the processes. Consensus has been extensively studied in traditional networks with process failures, and algorithms based on various system models have been developed \([2,3,4]\). The importance of consensus is due to the fact that it is a basic building block for solving several other important fault-tolerant distributed problems. However, there is a fundamental difference between the classical consensus problem, and the problem addressed in the paper: in the paper the set \( \Pi \) is unknown (and \( \Pi \) is precisely the information we want to obtain). This makes it a new problem that we call Consensus with Unknown Participants or simply CUP. Note that the notion of consensus with uncertain participants appears in \([5]\). However, the specification is different, and the context is also different (it is used as a building block for implementing a dynamic atomic broadcast service in a wired synchronous network). Thus, the results in \([5]\) are unrelated to the results established in this paper.

The CUP problem is formally defined in Section 2, which also defines the model in which the problem is solved. The classical consensus problem is hard to solve because processes may crash. With CUP, the difficulty of the problem is due to the unknown participants. So, for simplification, we assume in the paper that processes (i.e., mobile nodes) do not crash. We also assume that the nodes in \( R \) always form a connected network, and we assume the existence of an underlying multihop routing protocol: if some node \( n \) knows the existence of a node \( n' \) in \( R \), then \( n \) can reliably send a message to \( n' \). Moreover, \( n \) can only send reliably a message to nodes that it knows. Given these assumptions, the results obtained in the paper are independent of the underlying routing algorithm or mobility pattern of the nodes.

Clearly if all nodes in \( R \) do not know any other nodes, then no node can send any message to any other node, and CUP cannot be solved. This leads us to add to our model the notion of participant detectors, which are distributed oracles attached to each node \( n \). The participant detector of \( n \) provides to \( n \) a (possibly small) subset of the nodes in \( R \), e.g. by receiving messages or beacons. In Section 3, we introduce various classes of participant detectors, and we compare them based on the classical notion of reduction. In Section 4, we identify necessary and sufficient conditions for solving CUP, and we solve CUP using the participant detector named one sink reducibility. In Section 5, we illustrate how CUP can be used for solving the bootstrapping problem described in this section. Finally, we conclude and present future work in Section 6.