Middleware for Adaptive Group Communication in Wireless Sensor Networks

Klaas Thoelen, Sam Michiels, and Wouter Joosen

IBBT - DistriNet, Katholieke Universiteit Leuven
3001 Leuven, Belgium
klaas.thoelen@cs.kuleuven.be

Abstract. While the size and heterogeneity of wireless sensor networks confirm the need and benefit of group communication, an intelligent approach that exploits the interaction pattern and network context is still missing. This paper introduces sensor middleware to dynamically select the most efficient alternative from a set of group communication mechanisms. The proposed solution leverages on an empirical analysis of the ODMRP multicast protocol and was evaluated by a proof-of-concept prototype running on the SunSPOT platform. Results show that network overhead is considerably reduced when using the sensor middleware for software deployment, reconfiguration and periodic data monitoring.

Keywords: Group Communication, Context-Awareness, Middleware, Wireless Sensor Networks, Multicast.

1 Introduction

As wireless sensor network (WSN) platforms mature and standardization in their networking stack advances, WSN deployments become larger and more heterogeneous. This heterogeneity is exposed by differences in hardware, software and even ownership of the individual sensor nodes. As a result, sensor nodes emerge with different responsibilities (provided services, quality levels), capabilities (processing power, sensor accuracy) and general properties (ownership, types of sensors, energy source). Paradoxically, this heterogeneity also leads to the opportunity of defining groups of sensor nodes with common characteristics.

Group communication support is needed so that a subset of nodes adhering to a certain commonality can easily be contacted. In the scope of large scale, long lived and dynamic WSN environments, we identify three interaction patterns that benefit from group communication [9,13]: (1) software deployment, (2) dynamic configuration and (3) periodic actuation and monitoring. Evolving requirements on deployed services and the network as a whole trigger both software deployment and configuration. The difference lies however in the frequencies of these interactions and the amount of data to be exchanged. (1) Software deployment involves the dissemination of relatively large chunks of data and should be kept to a minimum to extend the battery lifetime of the sensor nodes. (2) Reconfigurations can be performed more frequently than deployment, given that the
network overhead is considerably smaller. (3) The exchange of monitoring and actuation data is highly frequent and often periodic in nature; in addition, the frequency of this periodic interaction is not static but susceptible to (changing) application preferences. The size of the data that is exchanged for monitoring however, is typically small in comparison with software deployment. These differences in interaction pattern characteristics influence the applicability of the various group communication mechanisms at hand.

In traditional networking, group communication is provided by multicast protocols that represent a group of nodes by a single address. The member nodes are not always bound by geographical properties and can be scattered throughout the network. The primary goal of multicast protocols is to deliver the intended packets to all members of the group with the least possible amount of transmissions per packet. This is typically achieved by the usage of an overlay tree or mesh in which all sources, members and needed intermediary nodes are connected. Creating these overlays, typically involves broadcasting the network in search of members and subsequent replies to the initiator of this so-called discovery phase. This exchange of control packets is commonly called protocol overhead, as it does not take part in delivering the actual data to the group members. In the face of mobility, changing connectivity and changing group memberships, these discovery phases need to be repeated at an appropriate frequency to keep the overlay up-to-date.

The problem that we address in this paper is that the protocol overhead introduced by multicast protocols is not always justifiable. Alternatives like broadcast and gossip mechanisms, introduce no or less protocol overhead [2,4] and although not group-aware, can be more efficient than multicast protocols. We argue that both the characteristics of the current interaction pattern and the network context need to be taken into account when performing group communication. Only in this way can we ensure that the most efficient communication mechanism is used.

The contributions of this paper are (1) the design of a context aware middleware layer that allows the selection of the most efficient mechanism at hand to perform group communication; to enable this, we performed (2) an efficiency analysis of a representative multicast protocol with regards to protocol overhead; (3) the evaluation of our prototype on the SunSPOT platform confirms the efficiency benefits of the proposed solution.

The remainder of this paper is structured as follows. In Section 2 we report on our efficiency analysis of the ODMRP multicast protocol and discuss the main results. We introduce our Communication Management Middleware in Section 3. We evaluate our prototype in Section 4 and elaborate on related work in Section 5. We conclude in Section 6.

2 Multicast Efficiency Analysis

With respect to the interaction patterns identified in the introduction, we compare a multicast and broadcast approach in terms of the amount of transmissions