Abstract

Wireless sensor networks (WSN) enable a myriad of new applications, e.g. human-embedded sensing, habitat exploration and ocean data monitoring. Nevertheless, they have different requirements from conventional systems. Self-configuration, energy-efficient operation, collaboration and in-network processing are examples of important requirements. In order to achieve these requirements, the system software of a sensor node plays a fundamental role: it should provide useful abstractions to enable the development of the applications and at the same time comply with the constrained resources of the sensor nodes.

The range of possible applications in a sensor node covers distinct tasks like clock synchronization, data acquisition, signal processing and data fusion. The traditional approach in this area is to provide operating system concepts with dramatically reduced functionality. In this work, we present an alternative approach. Our OS provides potentially arbitrary functionality that dynamically adapts to the actual profile of requirements. The basic idea is to offer services that are distributed over a cluster of nodes instead of having the entire system on each node.

Cooperation is the keyword to achieve complex tasks using the restricted sensor nodes. Our operating system (OS) supports this cooperation among neighboring nodes using the concept of distributed services. We are combining the typical OS functionality with the middleware one. Our system is responsible to coordinate the migration and placement of those services. For that, we develop a biologically inspired heuristic responsible to drive the placement of the services in theWSN.We develop two version of the heuristic, with different complexity and performance. Both are completely distributed and based on local information and local rules. The communication necessary for organizing the services is done by means of stigmergy.

Further, we present two clustering heuristics responsible to decompose the network graph into connected sub-graphs (called clusters). Each cluster will hold a complete instance of the OS and application. With the network divided in clusters, the organization overhead is reduced, since protocols that rely on some global information are restricted to a single cluster. This enhances the scalability of the system.

The clustering problem is called minimum intracommunication-cost clustering. The idea is that a minimum amount of resources must be present in each cluster and the clusters should be well connected. The first heuristic is able to handle networks with low topology changes, whereas the second can deal with moderate changes. Both heuristics rely on the principle of division of labor in social insects.

We evaluate our heuristics using the Shox wireless network simulator. The service distribution heuristics were able to produce very good assignments, near to the optimal, for most experiments. Our clustering heuristic for systems with low topology changes outperforms an existing heuristic (expanding ring) in term of cost for most cases. It was able to produce clusters that were, at most, in average 1.43 times the optimal for all simulated scenarios. Moreover, the results have low standard deviation.

Several enhancements can be done in our heuristics. In order to better distribute the burden imposed on the clusterhead, clusterhead rotation may be included in the emergent clustering. Moreover, an additional negotiation phase at the end of the heuristic may be included to improve the performance of the heuristic.

Moreover, we aim to combine the concept of our OS with the control script mechanism present in virtual machines for WSN. This will enable a straightforward development of data-centric scripts that use the extensive functionality of our distributed services to achieve complex goals.