Abstract

Reducing sizes and increasing capabilities of wireless communication devices as well as electronic and electro-mechanical systems make it possible to not only continuously monitor a physical phenomenon by staying very close to or even inside it, but also to quickly and autonomously perform the required action upon sensing an event. Distributed, large-scale networks that exhibit this capability are Wireless Sensor and Actor Networks (WSAN). WSAN consist of dense deployment of large number of small, low-cost and low-power sensor nodes and additionally contain few powerful and resource-rich actor nodes. WSAN need to address issues pertaining to energy conservation and unattended, untethered operation present in Wireless Sensor Networks (WSN) and additionally face challenges due to heterogeneity of nodes and real-time requirements of the applications.

In sensor networks, lifetime of sensor nodes determines the lifetime of the network. Among the important techniques proposed for prolonging the network lifetime by exploiting redundant deployment is using hierarchical architecture or clustering. However, the primary drawback of hierarchical control is the control message overhead; it is essential that the overhead does not dominate the network operations cost. The clustering algorithm should also scale to the network sizes and must preserve the desired sensing coverage of the network. If the clustering gives way to designing efficient networking protocols, it can lead to further prolonging the network lifetime.

Typical WSAN applications require real-time routing of data from the sensors to the actors, and the actor nodes will most likely be mobile. Also, actor nodes need to send periodic control messages to the sensors. Such dissemination of data in sensor networks is a challenging problem given the network sizes and the resource constraints.

The first part of this thesis proposes ‘Ripples’, an extremely lightweight and scalable clustering algorithm. Inspired by cellular infrastructure model, ‘Ripples’ produces clusters of bounded geographic size (given a node distribution density this guarantees a bound on the number of nodes in a cluster as well) and handles perturbations in the network locally in space and in time. The novelty of the algorithm lies in its deterministic operation to optimally exploit the redundant deployment and produce balanced clusters with minimum possible control overhead. It also preserves the desired sensing coverage of the network.
In the literature, two types of methods have been proposed for clustering. In bottom-up approaches, nodes volunteer to be cluster heads based on some criteria and other nodes join these clusters. These, 'Rapid' and 'Persistent' [21] have been shown to have better message-efficiency. 'Rapid' achieves better message efficiency at the cost of producing worst-case cluster sizes smaller than expected whereas 'Persistent' uses additional messages to form clusters of desired size. Top-down methods begin with one or more initiator nodes and form clusters in top-down fashion. To the best of our knowledge, 'GS3' [29] is the only method proposed in this category.

The approach to clustering in 'Ripples' is similar to that in 'GS3', and we have compared its performance with 'Rapid', 'Persistent', and 'GS3'. 'Ripples' uses as less as 50% of messages compared to 'Rapid' and 75% compared to 'GS3' to produce clusters of desired size like 'Persistent' and 'GS3'.

'Ripples' is designed to facilitate the design of efficient networking protocols. This is demonstrated in the second part of the thesis that proposes 'Attractor' and 'Spread'.

'Attractor' is a prediction-based routing algorithm that takes advantage of the regular infrastructure created by 'Ripples' to route the data in real-time and in an energy-efficient manner. 'Attractor' has been compared with 'Geographical and Energy Aware Routing (GEAR)' [56] protocol that performs routing based on the geographical locations of the source and destination nodes. 'Attractor' generates routes similar to those generated by 'GEAR'; however, by using the information generated by 'Ripples', 'Attractor' eliminates the need for all the overheads required by GEAR.

'Spread' is a data dissemination algorithm. 'Spread' also exploits the clustering structure formed by 'Ripples' to disseminate data to all the nodes in a given region in an economical fashion. 'Spread' uses order of magnitude less messages compared to classical flooding techniques.

Simulation results and theoretical analysis have been presented to substantiate the claims made in the thesis.