Distributed Computing in Wireless Sensor Networks

Hong Huang
New Mexico State University, USA

INTRODUCTION

Wireless sensor networks (WSNs) recently have attracted a great amount of attention because of their potential to dramatically change how humans interact with the physical world (Estrin, Culler, Pister, & Sukhatme, 2002; Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002). A wireless sensor network is composed of many tiny, wirelessly connected devices, which observe and perhaps interact with the physical world. The applications of WSN are many and wide-ranging, including wildlife habitat monitoring, smart home and building, quality monitoring in manufacturing, target tracking in battlefields, detection of biochemical agents, and so forth.

The emerging WSN technology promises to fundamentally change the way humans observe and interact with the physical world. To realize such a vision, *distributed computing* is necessary for at least two reasons. First, sending all the raw data to a base station for centralized processing is very costly in terms of energy consumption and often impractical for large networks because of the scalability problem of wireless networks' transport capacity (Gupta & Kumar, 2000). Second, merely using a large number of inexpensive devices to collect data hardly fundamentally changes the way humans interact with the physical world; and it is the intelligence embedded inside the network (i.e., distributed computing) that can have a profound impact.

WSN presents a very difficult environment for distributed computing. Sensors have severe limitations in processing power and memory size, and being battery-powered, they are particularly energy constrained. As a reference, the hardware capabilities of a typical sensor node are listed in Table 1. In WSN, device failures can be frequent, sensory data may be corrupted by error, and the wireless communications exhibit complex and unpredictable behavior. In such an environment, traditional methods for distributed computing face fundamental difficulties. Now, communications links are neither reliable nor predictable: they can come and go at any time. Packet routing is difficult since maintaining and storing routing tables for a massive number of nodes is out of the question. Routing to a single destination seems to have a solution (Intanagonwiwat, Govindan, & Estrin, 2000), complex message routing for distributed computing remains difficult. Also, distributed organizing and grouping of sensory data using traditional methods is costly in terms of protocol message overhead.

This article is organized as follows. We first describe WSN infrastructures required to support distributed computing, followed by a description of typical, important distributed computing applications in WSN; we the conclude the article.

INFRASTRUCTURE SUPPORT FOR DISTRIBUTED COMPUTING IN WSN

In order for WSN to effectively perform distributed computing, some necessary infrastructure needs to be established. The type of infrastructure required varies according to the specific application in question, but the common ones include neighbor discovery and management, synchronization, localization, clustering and grouping, and data collection infrastructure. We elaborate on each item as follows.

Table 1. Hardware capabilities of typical sensor nodes (Crossbow Technology —www.xbow.com)

	CPU	Nonvolatile Memory	Radio Transceiver	Power
MICA2	ATMega 128L 8 MHz, 8 bit	512 KB	869/915, 434, 315 MHz, FSK ~40 Kbps	2 AA 2850 mAh
MICA2DOT	ATMega 128L 8 MHz, 8 bit	512 KB	869/915, 434, 315 MHz, FSK ~40 Kbps	Coin cell 1000 mAh

Neighbor discovery and management refers to the process in which sensors discover their neighbors, learn their properties, and control which neighbors to communicate with. Discovery is typically done through sensors exchanging hello messages within radio range. In the process, sensors discover not only neighbors' presence, but also optionally their node type, node identifier, power level, location/coordinates, and so forth. Frequently, sensors can also control how many neighbors to communicate with through the use of power control—that is, a sensor can increase or decrease the scope of its immediate neighborhood by increasing or decreasing its transmitting power, respectively. This is also called topology control (Li & Hou, 2004), and its purpose is to allow sensors to use just enough, but no more power to ensure adequate connectivity.

Synchronization refers to the process in which sensors synchronize their clocks. Synchronization is necessary because sensory data is often not useful if not put in a proper temporal reference frame. Traditional methods for synchronization in a network, such as NTP (Mills, 1994), do not apply very well in WSN. This is because the assumptions on which tradition network synchronization methods are based, such as availability of high-precision clocks, stable connections, and consistent delays, are no longer true in WSN, causing considerable difficulties. The approach to deal with such difficulties in WSN is to relax the requirements. For example, only local, not global, synchronization is maintained, or only event ordering, not precise timing, is kept (Romer, 2001).

Localization refers to the process in which sensors obtain their position/coordinates information. Similar to synchronization, localization is necessary because sensory data needs to be put in a spatial reference frame. Sensors with global positioning system (GPS) capability are currently commercially available; they obtain their coordinates from satellites with a few meters' accuracy. The downside with using GPS is the cost, and the unavailability indoors or under dense foliage. In a WSN without GPS capability, it is still possible to localize relatively to a few reference points in the network (Bulusu, Heidemann, & Estrin, 2000).

Clustering and grouping refers to the process in which sensors organize themselves into clusters or groups for some specific function. A cluster or a group typically consists of a leader and a few members. The leader represents the cluster or group and maintains external communication, while the members report data to the leader and do not communicate with the outside. Such organization is advantageous for scalability, since a large network can now be reduced to a set of cluster or groups. Task-specific clusters or groups can be formed. For example, sensors around a moving target form a tracking group, which moves with target, while sensors not in the tracking group can be put to sleep to save energy (Liu, Reich, Cheung, & Zhao, 2003).

Data collecting infrastructure ensures that sensory data is transported correctly and efficiently to one or a few collection points, sometimes called data sinks. A typical approach is publish and subscribe with attribute-based naming, where a sink broadcasts its interest for some data attributes, and sensors send their data if it matches the interests. An example of such an approach is directed diffusion, in which an infrastructure based on the hop count to the sink is established and refreshed periodically (Intanagonwiwat et al., 2000).

TYPICAL DISTRIBUTED COMPUTING APPLICATIONS IN WSN

In this section, we describe typical distributed computing applications in WSN which include distributed query and search, collaborative signal processing, distributed detection and estimation, and distributed target tracking.

Distributed query and search refers to the process in which a user query or search for an event or events inside the network in a distributed fashion. There are two major types of such applications: blind and structured. In a blind search, no prior information about the target exists. In a structured search, some kind of infrastructure exists which points to the target location in a distributed manner. We elaborate on these two types of searches below.

There are three major approaches to perform blind search. The first one is flooding, in which the query message is flooded to the entire network and the target responds with a reply. The advantages of flooding are simplicity and low response latency. The disadvantage is the high communications cost in terms of number of messages transmitted. To mitigate the high communication cost of flooding, a second approach, iterative, limited flooding, can be used (Chang & Liu, 2004). In such an approach, a sequence of limited broadcasts of increasing hop-count limits is tried until the target is found, in the hope that the target will be found during a low-cost, limited broadcast. The expected communications cost reduction of this approach comes at the expense of higher search latency. In the third approach, a query packet carries out a random walk in the network, which continues until the search target is encountered (Avin & Brito, 2004). This approach can further reduce communications cost but at the expense of even higher latency.

In a structured search, indices or pointers for targets are distributed in the network. A typical approach uses a distributed hash table, where the name of a target is randomly and uniformly hashed to a number that identifies a node, or a location, where the target information is stored (Ratnasamy et al., 2002). The search becomes a simple matter of evaluating the hash function of the target name which points to a node that stores the target information. This simplification comes at the cost of maintaining an infrastructure that stores target information in a distributed manner, which can be costly if