Modeling In-Network Processing and Aggregation in Sensor Networks

Ajay Mahimkar

Department of Electrical and Computer Engineering

The University of Texas at Austin, TX 78713, USA

mahimkar@ece.utexas.edu

Abstract

The rapid advances in processor, memory and radio technology have enabled the development of distributed networks of sensor nodes capable of sensing and communicating using wireless media. The basic operation in a sensor network is the systematic gathering and transmission of sensed data to the end-user. The severe energy constraints and limited computing capabilities of the sensors present major challenges to its design. In this project, we aim to develop an energy-efficient algorithm for in-network processing and aggregation in a sensor network. We also propose to present a model for a sensor node and communication in a sensor network.

Index Terms

Sensor Networks, In-Network Processing, Data Aggregation, Energy-Efficient Algorithms

I. INTRODUCTION

A sensor network comprises of a large number of low-power wireless sensors spread across a geographical area that can be used to monitor and control the physical environment from remote locations. Each sensor node is battery powered and equipped with integrated sensors, data processing capabilities and short-range radio communications. The readings sensed by the sensors are routed to the end user by a multi-hop infrastructureless architecture through the base station as shown in Fig. 1. Potential applications of sensor networks include real-time traffic monitoring, battlefield surveillance, nuclear attack detection, wireless meter reading services, etc.

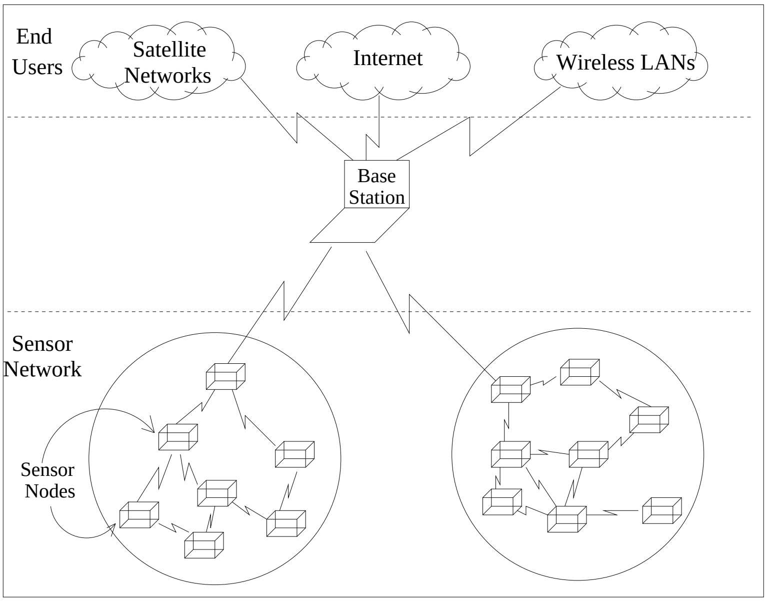

Fig. 1. A representative Sensor Network

The basic operation in sensor networks is the systematic gathering of sensed data to be eventually delivered to the base station (referred to as a *sink*). Since the sensor nodes are energy-constrained, communication between the base station and the sensors must be energy-efficient. The key challenge in such an environment is the design of communication protocols that maximize the network lifetime. Network Lifetime is the time at which the first sensor node in the network dies i.e. it completely exhausts its battery resources. Data fusion or aggregation has emerged as a useful paradigm in sensor networks. The key idea is to combine data from different sensors to eliminate redundant transmissions, thereby leading to efficient use of the energy resources [16]. For example, in a reconnaissance-oriented sensor network, sensor readings indicate detection of a target, while the aggregation can be used for tracking and identifying the detected target.

1) Sensor Node Hardware Configuration: An example of a sensor node's hardware configuration is the Berkeley Mica Motes. The Mica Mote is a small (several cubic inch) sensor/actuator unit with a CPU, power source, radio and several optional sensing elements. The processor is a 8-bit 4 MHz Atmel ATmega 128L processor with 128KB of program memory, 4KB of RAM for data and 512KB of flash memory. The processor only supports a minimal RISC-line instruction set, without support for multiplication or variable-length shifts or rotates. The radio is a 916 MHz low-power radio, delivering upto 40 Kbps bandwidth on a single shared channel and with a range of upto a few dozen meters. The radio consumes 4.8 mA in receive mode, upto 12 mA in transmit mode and 5 μA in sleep mode.

II. Problem Definition

In this work, our main consideration is wireless sensor networks where the sensors are randomly distributed over an area of interest. The sensor nodes are static and communicate to the base station in a multi-hop fashion. The sensor nodes periodically sense the environment and have data to send in each round of communication. The end user can query the network for a particular attribute through a base station or a sensor node (this is referred to as a *sink*). The intermediate nodes on the path to the sink, aggregate the data they receive from the others and forward the aggregate towards the sink. The problem is to find a routing scheme to deliver data packets collected from sensor nodes to the base station in such a way, that maximizes the lifetime of the sensor network.

III. EXISTING APPROACHES

A. *Directed Diffusion*

Directed Diffusion is a data-centric communication paradigm for sensor networks [1]. In directed diffusion, data generated by the sensors is named by attribute-value pairs. A sensing task (initiated by the sink) is disseminated throughout the sensor network as an *interest* for the named data. This interest dissemination sets up *gradients* within the network that point to the neighbor from which an interest was received. Sensors matching the interest send their data to the sinks along multiple gradient paths, initially and then gradually reinforce better paths. Intermediate nodes aggregate the data and forward the fused data to the next node till it reaches the sink. Fig. 2 gives an illustration of directed diffusion.

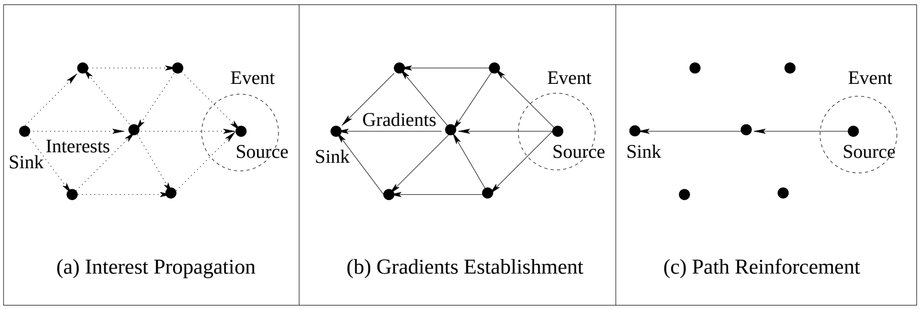

Fig. 2. Schematic for Directed Diffusion

B. *SPIN - Sensor Protocols for Information via Negotiation*

SPIN efficiently disseminates information among sensors in an energy-constrained wireless sensor network [5]. Nodes running SPIN name their data using high-level data descriptors, called meta-data. SPIN nodes base their communication decisions both upon application-specific knowledge of the data and upon knowledge of the resources that are available to them. This allows the sensors to efficiently distribute data given a limited energy supply.

C. Power-Efficient Algorithms

1) LEACH - Low Energy Adaptive Clustering Hierarchy: The LEACH protocol presented in [6] is an elegant solution to the data aggregation problem where clusters are formed in a self-organized manner to fuse data before transmitting to the base station or sink. In LEACH, a designated node in each cluster, called the *clusterhead* is responsible for collecting and aggregating the data from sensors in its cluster and eventually transmitting the result to the base station. An improved version of LEACH, called LEACH-C [7] does cluster formation at the beginning of each round using a centralized algorithm by the base station.

2) PEGASIS - Power-Efficient GAthering in Sensor Information Systems: In [2], the authors propose a new chain-based protocol called PEGASIS that minimizes the energy consumption at each sensor node. PEGASIS achieves reduction in energy consumption as compared to LEACH since it requires only one designated node to send the combined data to the base station. The key idea is that nodes organize to form a chain and each node take turns being the leader for communication to the base station. The data is collected starting from each endpoint of the chain and aggregated along the path to the designated head-node. Unlike LEACH [6] (that uses hierarchical clustering), PEGASIS uses a flat topology thereby eliminating the overhead of dynamic cluster formation. PEGASIS achieves a better performance than LEACH by between 100% and 300% in terms of network lifetime.

3) PEDAP - Power Efficient Data Gathering and Aggregation Protocol: In [4], the authors propose a new minimum spanning tree-based protocol, called PEDAP and its power-aware version (PEDAP-PA). The data packets are routed to the base station over the edges of the minimum spanning tree. PEDAP outperforms LEACH and PEGASIS by constructing minimum energy consuming routing for each round of communication. The advantage with PEDAP-PA is that it minimizes the total energy of the system while distributing the load evenly among the nodes. This leads to increased system lifetime.

4) APTEEN - Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network Protocol: A. Manjeshwar et al. develop a protocol called APTEEN [3] that uses an enhanced TDMA schedule to efficiently incorporate query handling. APTEEN provides a combination of proactive (by requiring nodes to periodically send data) and reactive (by making nodes to respond immediately to time-critical situations) policies.

IV. Modeling Environments

A. NS-2

NS-2 [8] is a well-established open-source network simulator from UCB. It is a discrete event simulator with substantial support for simulation of TCP, routing and multicast protocols over wired and wireless networks.

B. OPNET

OPNET Modeler [9] developed at MIT offers sophisticated modeling and simulation of communication networks. An OPNET model is hierarchical where the top level contains the communication nodes and the topology of the network. It uses a discrete-event simulator to execute the entire model.

C. Ptolemy-II

Ptolemy-II [10] [11] integrates diverse models of computation, such as continuous-time, discrete-event, finite state machines, process networks, synchronous dataflow, synchronous/reactive. This capability can be used, for example to model the physical dynamics of sensor nodes, their digital circuits, energy consumption, signal processing or real-time software behavior. In a recent paper [12], P. Baldwin et al. describe a modeling and simulation framework called *VisualSense* for wireless sensor networks that builds on and leverages Ptolemy II. *VisualSense* supports actor-oriented definition of sensors, wireless communication channels and provides an extensible visualization framework.

D. TOSSIM - A Simulator for TinyOS wireless sensor networks

TOSSIM [13] is a discrete-event simulator for TinyOS [15] wireless sensor networks. It captures the behavior and interaction of networks of thousands of TinyOS Mica motes at network bit granularity. TOSSIM is composed of five parts: support for compiling TinyOS component graphs into the simulation infrastructure, a discrete event queue, a small number of re-implemented TinyOS hardware abstraction components, mechanisms for extensible radio and ADC models.

E. TinyGALS - Globally asynchronous and locally synchronous

E. Cheong et al. have designed and implemented a globally asynchronous and locally synchronous model, called TinyGALS [14], for programming event-driven embedded systems. Their model is influenced by the TinyOS project [15]. TinyOS lacks explicit management of concurrency, thereby making components hard to develop. The TinyGALS model uses two levels of hierarchy to build an application. At the system level, a set of modules communicates asynchronously through message passing. Within each module, components communicate via synchronous method calls.

V. Objectives

Having looked at the existing approaches and the simulation tools available, we aim to to develop a model for a sensor node, wireless communication channels, energy consumption. We also propose to develop a new model for in-network processing and aggregation in a wireless sensor network. The communication in a sensor network should be such that it maximizes the network lifetime. The problem with the existing power-efficient algorithms is that they optimize the energy metric locally. This may not lead to an optimal use of the energy resources. We propose a novel algorithm that globally maximizes the energy of the sensors and balances the energy consumption in proportion to the energy reserves.

The deliverables for the project will be as follows :

- A Model for a sensor node, that includes sensing, communication, switching to a low power mode.
- A novel distributed algorithm that globally maximizes the energy of the sensors and balances the energy consumption in proportion to the energy reserves.
- Performance evaluation of our algorithm with existing algorithms.

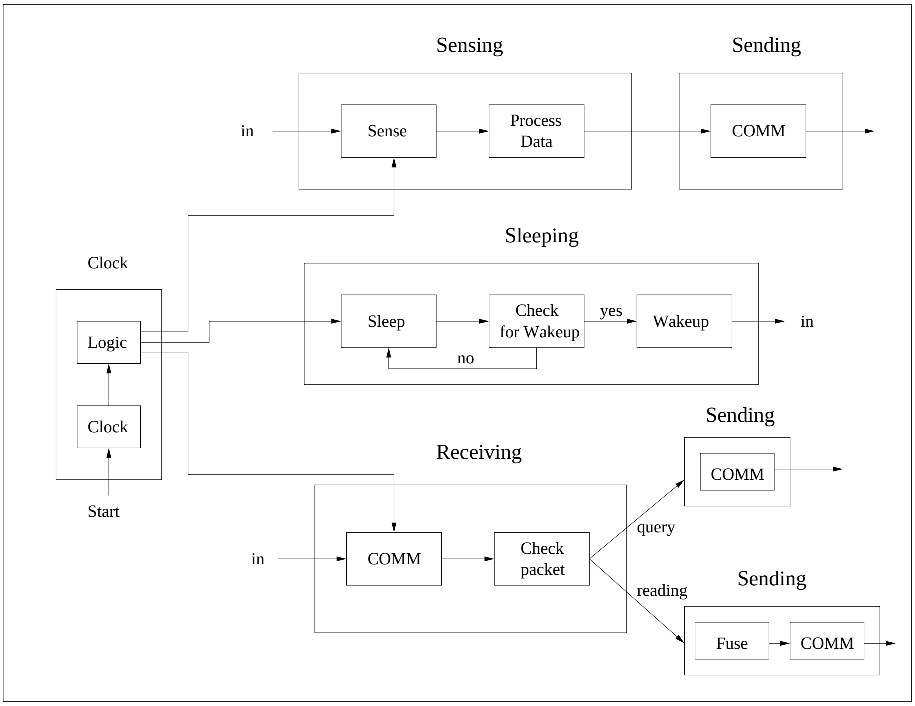

Fig. 3. A model for a Sensor Node

Our model for describing a typical sensor node is depicted in Fig. 3. The key blocks are

1) *Sensing* - If the sensor node is awake, it can sense the physical environment and report readings.
2) *Receiving* - The sensor can receive data along its communication block. On receiving a packet, each sensor node will check whether it is a *query* message or a *reading*. If it is a *query* message, it broadcasts it to the remaining sensors. If it is a *reading*, then it can aggregate the readings and forward it to the base station along the best available path.
3) *Sending* - When the sensor node has data to be sent, it uses the communication block for trans-

mission. The data could either be a query, a sensed reading or an aggregate.

4) *Sleeping* - In order to conserve energy, a sensor can be powered off when the network does not require its services. The logic block determines when a sensor could be powered off. Periodically, each sensor checks if it needs to be powered on. If it does, then it wakes up and provides services to the network.

REFERENCES

[1] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, "Directed Diffusion for Wireless Sensor Networking", *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 2-16, Feb. 2003.

[2] S. Lindsey, C. Raghavendra, K. M. Sivalingam, "Data Gathering Algorithms in Sensor Networks using Energy Metrics", *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924-935, Sept. 2002.

[3] A. Manjeshwar, Q-A. Zeng, D. P. Agarwal, "An Analytical Model for Information Retrieval in Wireless Sensor Networks using Enhanced APTEEN Protocol", *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 12, pp. 1290-1302, Dec. 2002.

[4] H. O. Tan, I. Korpeoglu, "Power Efficient Data Gatherting and Aggregation in Wireless Sensor Networks", *Proc. ACM Int. Conf. Management of Data (ACM SIGMOD)*, vol. 32, no. 4, pp. 66-71, Dec. 2003.

[5] W. Heinzelman, J. Kulik, H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Proc. ACM Int. Conf. Mobile Computing and Networking (ACM/IEEE Mobicom)*, pp. 174-185, Aug. 1999.

[6] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", *Proc. IEEE Int. Conf. System Sciences*, vol. 8, pp. 8020, Jan. 2000.

[7] W. Heinzelman, "Application-Specific Protocol Architectures for Wireless Networks", PhD thesis, Massachusetts Inst. of Technology, June 2000.

[8] NS-2, http://www.isi.edu/nsnam/ns/, 2004.

[9] OPNET Technologies, Inc., "OPNET Modeler", http://www.opnet.com/products/modeler/home.html, 2004.

[10] C. Hylands, E. A. Leem J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng, "Heterogeneous Concurrent Modeling and Design in java: Volume 1: Introduction to Ptolemy II", Tech. Memorandum UCB/ERL M03/27, Univ. of California, Berkeley, CA USA, July 2003.

[11] C. Hylands, E. A. Leem J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng, "Heterogeneous Concurrent Modeling and Design in java: Volume 3: Ptolemy II Domains", Tech. Memorandum UCB/ERL M03/27, Univ. of California, Berkeley, CA USA, July 2003.

[12] P. Baldwin, S. Kohli, E. A. Lee, X. Liu, Y. Zhao, "Modeling of Sensor Nets in Ptolemy II", *Proc. Information Processing in Sensor Networks (IPSN)*, Apr. 2004.

[13] P. Levis, N. Lee, M. Welsh, D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications", *Proc. ACM Int. Conf. Embedded Networked Sensor Systems (ACM SenSys)*, pp. 126-137, Nov. 2003.

[14] E. Cheong, J. Liebman, J. Liu, F. Zhao, "TinyGALS: A Programming Model for Event-Driven Embedded Systems", *Proc. ACM Symposium Applied Computing (ACM SAC)*, pp. 698-704, Mar. 2003.

[15] D. E. Culler, J. Hill, P. Buonadonna, R. Szewczyk, A. Woo, "A Network-Centric Approach to Embedded Software for Tiny Devices", *Proc. Int. Workshop Embedded Software (EMSOFT)*, pp. 114-130, Oct. 2001.

[16] G. Asada, A. Burstein, D. Chang, M. Dong, M. Fielding, E. Kruglick, J. Ho, F. Lin, T. H. Lin, H. Marcy, R. Mukai, P. Nelson, F. Newberg, K. S. J. Pister, G. Pottie, H. Sanchez, O. M. Stafsudd, S. Valoff, G. Yung, W. J. Kaiser, "Low power systems for wireless microsensors", *Proc. IEEE Int. Symposium Low Power Electronics and Design*, pp. 17-21, Aug. 1996.