On Scheduling Sensor Networks

Lloyd G. Greenwald
Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104
lgreenwald@mcs.drexel.edu

Harish Sethu
Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104
sethu@ece.drexel.edu

Sensor Networks

The emerging technology of wireless sensor networks (D. Estrin *et al.* 2001) holds the promise of facilitating a diverse array of real-time decision support applications in environmental monitoring, military deployments, and robotic explorations. Such networks consist of a distributed set of sensor nodes, each of which typically uses battery-operated sensing and computing devices and a low-power radio transmitter and receiver. These sensor nodes sense, compute, and communicate with each other to co-operatively accomplish, typically, a common set of tasks. Effective integration of computation and communication in sensor networks is a challenging problem that will draw from as well as contribute to the fields of parallel and distributed computing, network protocol design, and stochastic optimization. Aspects of this problem that are unique in comparison to traditional distributed systems include:

Dynamic network topology: Nodes with depleted power cells may disappear from the network; mobile nodes may move in and out of low-power wireless range.

Tight power constraints: Re-charging power cells deployed in the field may be extremely difficult, especially in hostile or otherwise dangerous environments; total power used throughout the network must be minimized while avoiding any unbalanced power consumption that could change network topology.

Self-configurability: Collaborative processing among sensor nodes has to be achieved in the absence of a global synchronizing clock or common node identifiers.

Limited memory and computational power: Sensor nodes may be constrained by limited memory and limited processing due to power constraints as well as practical issues such as price and feasibility of deployment.

Significant potential applications of this work include environmental monitoring, military deployments, biomedicine, and robotic explorations. For example, a sensor network deployed as an early warning system against forest fires may track changes in temperature gradients both spatially and in magnitude. A network of vibration sensors might track the movement of enemy troops and tanks on a battlefield. A network of mobile robots might explore Mars in search of a suitable region to build an encampment. In some of these applications, the sensor network may be connected to a general communications network for additional processing, tasking, or delivery.

In general, the control, management or exploration of areas that are unreachable or are too dangerous for humans, such as hazardous waste sites, military battlefields and unexplored remote environments, can often benefit from the deployment of sensor networks. Once deployed, these networks must make effective use of their resources to complete specific tasks in reasonable time frames. Keeping humans away from danger requires that sensor networks operate autonomously or semi-autonomously for extended periods of time, and behave robustly in the face of uncertainty.

We are investigating protocols for integrated computation and communication that schedule local broadcasts and receptions in various regions of the network while simultaneously scheduling computations throughout the network. Sensor network protocols can be evaluated according to a number of criteria, including:

Computation Convergence and Responsiveness: From an initial state, how quickly does the network converge to usable results? Given any changes in sensor readings, how quickly does the network respond?

Energy Efficiency: How much total energy is consumed for a given computation? Is there an even distribution of energy depletion across all nodes?

Robustness and Scalability: How much does the protocol rely on the network design? Can the protocol adapt to node failure or energy depletion? Can it scale to node additions or the merging of mobile networks?

Associative Combination Computations

Consider a network assigned the task of determining the maximum temperature in the region covered by the network, and have this information be available at each of the sensor nodes. A naive means of accomplishing this would be to implement a periodic many-to-many multicast through the multi-hop wireless network, and have each sensor node compute the maximum of all the temperatures reported by each of the sensors in the network. A more energy-efficient method, however, is through *data aggregation* that eliminates redundancy through forwarding only the relevant information periodically. For example, a sensor node would transmit to another sensor only the maximum of the temperatures reported to it thus far. With aggregation, data from multiple sources is combined at each hop, reducing both the number and size of transmissions and thus achieving higher energy efficiency.

The majority of applications intended for sensor networks can take advantage of data aggregation. In addition, most such applications exhibit the property of associativity in combination, i.e., data from multiple sources can be combined in any order to yield identical results. We refer to this class of computations as *associative combination algorithms* (Sethu & Wagh 1999).

Scheduling Sensor Networks

Our strategy for designing protocols for effective use of sensor networks is to formulate an optimization problem whose solution is a *schedule* of computations and communications. Constraints within the optimization problem ensure that local decisions made by each node converge to global solutions. We then solve this optimization problem *off-line*. In other words, we design the protocol centrally with full knowledge of the sensor network properties, constraints, and optimization criteria. The result is a distributed protocol that is then independently executed *on-line* by each node, without global synchronization. The following captures a few example constraint types:

> *Communication Concurrency Constraints:* Communication is constrained by potential collisions between nodes in nearby regions of a sensor network. Only one node within a broadcast region may successfully transmit at a time. Since nodes may be mobile, these constraints are time-varying. The broadcast range of communications captured in these constraints may vary with power level fluctuations and environmental interference.
>
> *Task-Specific Communication and Computation Constraints:* These constraints define the flow of computations. For example, associative combination algorithms induce a tree-like set of constraints. A computation cannot begin until its input data is received. Thus, reception actions must precede computation actions that must then precede transmission actions. Receptions and transmissions in neighboring nodes must be synchronized to minimize energy consumption and remain consistent with the communication concurrency constraints.
>
> *Task-Specific Computation Requirements:* Computation constraints that restrict flow must be balanced with constraints to require minimum flow (enough to solve the desired task). For an associative combination algorithm, these constraints take the form of specifying the minimum amount of computation that must be aggregated at the sink. These constraints interact to effectively distribute the required computation and communication.

Additional constraints that capture different problem features might include: redundant communication paths to aid robustness; mixed wired/wireless sensor network communication patterns; heterogeneous computation and communication capabilities; randomized patterns of integrated computation and communication to aid security; and stochastic constraints to capture the dynamics of mobile nodes.

In addition to the various communication and computation constraints, we specify an objective function that captures a desired trade-off between objectives such as responsiveness and energy efficiency. This can be captured in associative combination computations by weighing the total cost of the energy consumed by communication and computation with the earliest time that a result is available at a sink node. The constructed optimization problem can be solved at design time (off-line) using numerous standard techniques. The resulting schedule of computations and communications captures the desired trade-offs while guaranteeing that communications do not cause collisions despite the broadcast nature of wireless sensor network communications.

Beyond Static Schedules

Due to uncertainties in the conditions under which the sensor network will operate, such as unknown initial network topology, node mobility, and node failure, off-line optimization must be combined with on-line adaptation. Our ongoing work involves limiting on-line adaptation by augmenting the off-line scheduling problem with a collection of constraints to produce solutions that capture a finite set of *node roles*. Each node is programmed with the local schedule for each role. We then provide an on-line protocol for nodes to discover their initial roles and to switch to new roles as conditions change. This approach is reminiscent of our prior work on *conditional scheduling* for designing real-time systems (Greenwald & Dean 1998).

A node's role in a sensor network is determined by its place in the network topology and its proximity to the sink node. A node might be a *leaf* node that computes immediately without waiting for input from neighbors, an *aggregation* node that waits for input from all but one of its neighbors prior to computation and transmission, or a *forwarding* node that serves to transmit data from one part of the network to another. Each example role is defined by a discernible pattern of computation and communication actions.

While single task problems are rich in research challenges, the challenges are multiplied if we consider using the same sensor network to compute the solution to more than one task. This might involve multiple tasks concurrently competing for the resources of a single sensor network or, more simply, the re-use of an existing network for a new task. We are investigating approaches to this problem that include (1) aggregating multiple tasks into a single task, (2) solving each task independently and providing dynamic protocols to resolve resource contentions, and (3) developing loosely coupled optimization problems in which a subset of constraints correspond to each task, and the remaining constraints model the coupling of tasks.

References

D. Estrin *et al.* 2001. Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers. Technical Report 0-309-07568-8, Computer Science and Telecommunications Board.

Greenwald, L., and Dean, T. 1998. A conditional scheduling approach to designing real-time systems. In *Fourth International Conference on AI Planning Systems*.

Sethu, H., and Wagh, M. D. 1999. Node-degree requirements for time-optimal execution of a class of parallel algorithms. In *Proceedings of the 11th IASTED Int'l Conf. on Parallel and Distributed Computing and Systems*, 472–478.