Next Century Challenges: Scalable Coordination in Sensor Networks

Deborah Estrin
Ramesh Govindan
John Heidemann
Satish Kumar

USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292, USA
{estrin,govindan,johnh,kkumar}@isi.edu

Abstract

Networked sensors—those that coordinate amongst themselves to achieve a larger sensing task—will revolutionize information gathering and processing both in urban environments and in inhospitable terrain. The sheer numbers of these sensors and the expected dynamics in these environments present unique challenges in the design of unattended autonomous sensor networks. These challenges lead us to hypothesize that sensor network coordination applications may need to be structured differently from traditional network applications. In particular, we believe that *localized algorithms* (in which simple local node behavior achieves a desired global objective) may be necessary for sensor network coordination. In this paper, we describe localized algorithms, and then discuss *directed diffusion*, a simple communication model for describing localized algorithms.

1 Introduction

Integrated low-power sensing devices will permit remote object monitoring and tracking in many different contexts: in the field (vehicles, equipment, personnel), the office building (projectors, furniture, books, people), the hospital ward (syringes, bandages, IVs) and the factory floor (motors, small robotic devices). Networking these sensors—empowering them with the ability to *coordinate* amongst themselves on a larger sensing task—will revolutionize information gathering and processing in many situations. Large scale, dynamically changing, and robust sensor colonies can be deployed in inhospitable physical environments such as remote geographic regions or toxic urban locations. They will also enable low maintenance sensing in more benign, but less accessible, environments: large industrial plants, aircraft interiors etc.

To motivate the challenges in designing these sensor networks, consider the following scenario. Several thousand sensors are rapidly deployed (*e.g.,* thrown from an aircraft) in remote terrain. The sensors coordinate to establish a communication network, divide the task of mapping and monitoring the terrain amongst themselves in an energy-efficient manner, adapt their overall sensing accuracy to the remaining total resources, and re-organize upon sensor failure. When additional sensors are added or old sensors fail, the sensors re-organize themselves to take advantage of the added system resources.

Several aspects of this scenario present systems design challenges different from those posed by existing computer networks (Section 2). The *sheer numbers* of these devices, and their unattended deployment, will preclude reliance on broadcast communication or the configuration currently needed to deploy and operate networked devices. Devices may be battery constrained or subject to hostile environments, so individual *device failure* will be a regular or common event. In addition, the configuration devices will *frequently change* in terms of position, reachability, power availability, and even task details. Finally, because these devices interact with the physical environment, they, and the network as a whole, will experience a significant range of *task dynamics*.

The WINS project [1] has considered device-level communication primitives needed to satisfy these requirements. However, these requirements potentially affect many other aspects of network design: routing and addressing mechanisms, naming and binding services, application architectures, security mechanisms, and so forth. This paper focuses on the principles underlying the design of services and applications in sensor networks. In particular, since the sensing is inherently distributed, we argue that sensor network applications will themselves be distributed.

Many of the lessons learned from Internet and mobile network design will be applicable to designing sensor network applications. However, this paper hypothesizes that sensor networks have different enough requirements to at least warrant re-considering the overall structure of applications and services. Specifically, we believe there are significant robustness and scalability advantages to designing applications using *localized* algorithms—where sensors only interact with other sensors in a restricted vicinity, but nevertheless collectively achieve a desired global objective (Section 3). We also describe *directed diffusion*, a promising model for describing localized algorithms (Section 4).

Our research project is starting to investigate the design of localized algorithms using the directed diffusion model. These ideas were developed in the context of a DARPA ISAT study, chaired by one of the authors (Estrin). The idea of applying directed diffusion to this problem domain is due to Van Jacobson, based on experiences with reliable multicast [2] and adaptive Web caching design.

2 Sensor Network Challenges

By early next century, sensor integration, coupled with unceasing electronic miniaturization, will make it possible to produce extremely inexpensive sensing devices. These devices will be able to monitor a wide variety of ambient conditions: temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects, and so on. These devices will also be equipped with significant (*i.e.,* comparable to today's high-end portable computers) processing, memory, and wireless communication capabilities.

Emerging low-level and low-power wireless communication protocols will enable us to *network* these sensors. This capability will add a new dimension to the capabilities of sensors: Sensors will be able coordinate amongst themselves on a higher-level sensing task (*e.g.,* reporting, with greater accuracy than possible with a single sensor, the exact speed, direction, size, and other characteristics of an approaching vehicle).

Networking inexpensive sensors can revolutionize information gathering in a variety of situations. Consider the following scenarios, arranged in increasing order of complexity:

- Each item of inventory in a factory warehouse or office complex has, attached to it, a tag. Stick-on sensors, discreetly attached to walls, or embedded in floors and ceilings, track the location history and use of items. The sensor network can automatically locate items, report on those needing servicing, analyze long-term correlations between workflow and wear, report unexpected large-scale movements of items or significant changes in inventory levels. Some systems today (for example, those based on bar-codes) provide inventory tracking; full sensor-net based systems will eliminate manual scanning and provide more data than simply location.

- Thousands of disposable sensors are densely scattered over a disaster area. Some of them fall into regions affected by the disaster, say a fire—these sensors are destroyed. The remaining sensors collectively map these affected regions, direct the nearest emergency response teams to affected sites, or find safe evacuation paths. Disaster recovery today is by comparison very human intensive.

- Every vehicle in a large metropolis has one or more attached sensors. These sensors are capable of detecting their location; vehicle sizes, speeds and densities; road conditions and so on. As vehicles pass each other, they exchange information summaries. These summaries eventually diffuse across sections of the metropolis. Drivers can plan alternate routes, estimate trip times, and be warned of dangerous driving conditions. Unlike the centralized systems sometimes seen today, one based on local communication would scale as the number of vehicles grows and provide much greater local detail.

These futuristic scenarios bring out the two key requirements of sensor networks: support for very large numbers of unattended autonomous nodes and adaptivity to environment and task dynamics.

Many large-scale networks exist today; the Internet is a prime example. Sensor networks present a fundamentally more difficult problem, though, because the ratio of communicating nodes to users is much greater. Each personal computer on the Internet has a user who can resolve or at least report all manner of minor errors and problems. This human element allows the Internet to function with much less robust software. Sensor networks, by comparison will exist with the ratio of thousands of nodes per user (or more). At such ratios, it is impossible to pay special attention to any individual node. Furthermore, even if it were possible to consider each node, sensors may be inaccessible, either because they are embedded in physical structures, or thrown into inhospitable terrain. Thus, for such a system to be effective, it must provide *exception-free*, unattended operation (the term exception-free is due to Mark Weiser).

It is not completely true that there are no large scale unattended systems today. Automated factories, for example, may contain hundreds of largely unsupervised computers. This example illustrates the second requirement of sensor networks: they operate and must respond to very dynamic environments. Automated factories are deployed with very careful planning and react to very few external events. Sensor networks instead will be deployed in a very ad hoc manner (possibly thrown down at random). They will suffer substantial changes as nodes fail due to battery exhaustion or accident, new nodes are added, nodes move or are carried. User and environmental demands also contribute to dynamics as what is being sensed moves and what is considered interesting changes. Thus sensor networks must automatically adapt to changes in environment and requirements.

One hypothesis for the overall design of a sensor network is that it is sufficient to design sensor network applications using Internet technologies coupled with ad-hoc routing mechanisms. In such a design, each sensor node is an Internet-capable device (has one or more IP addresses) and can run applications and services. When deployed, sensor nodes establish an ad-hoc network amongst themselves; thereafter, application instances running on each node can communicate with each other. Applications, aided by directory and resource discovery services, are structured much the same way as traditional Internet applications.

We believe, however, that sensor network requirements are different enough from those of traditional wired and wireless networks to warrant considering a different design. This design has the following features:

Data-Centric Unlike traditional networks, a sensor node may not need an identity (*e.g.,* an address) [1]. That is, sensor network applications are unlikely to ask the question: What is the temperature at sensor #27? Rather, applications focus on the **data** generated by

[1] In some situations, for example, for querying a specific faulty sensor, the ability to address an individual sensor is clearly necessary.

sensors. Data is named by attributes and applications request data matching certain attribute values. So, the communication primitive in this system is a request: Where are nodes whose temperatures recently exceeded 30 degrees? This approach decouples data from the sensor that produced it. This allows for more robust application design: even if sensor #27 dies, the data it generates can be cached in other (possibly neighboring) sensors for later retrieval.

Application-Specific Traditional networks are designed to accommodate a wide variety of applications. We believe it is reasonable to assume that sensor networks can be tailored to the sensing task at hand. In particular, this means that intermediate nodes can perform application-specific data aggregation and caching, or informed forwarding of requests for data. This is in contrast to routers that facilitate node-to-node packet switching in traditional networks.

If we admit this architecture, how might we design applications on top of a sensor network that provided this kind of communication? Recall that sensor network applications of interest to us are those in which sensor nodes *coordinate* to perform a higher-level sensing task (*e.g.,* Is it time to order more inventory? At what speed and in what direction was that elephant traveling?). Clearly, this kind of coordination can be structured in a centralized manner. Individual sensors report their data to a central node, which then performs the computation required for the application. This centralized structure is a bad choice for several reasons: it provides a single point of failure, it can be energy inefficient, and it doesn't scale to large networks.

We hypothesize that sensor network coordination applications are better realized using *localized* algorithms. We use this term to mean a distributed computation in which sensor nodes only communicate with sensors within some neighborhood, yet the overall computation achieves a desired global objective. What is the rationale for using localized algorithms in sensor networks? Since the sensors themselves are physically distributed, it is not unnatural to design sensor networks using distributed algorithms. Furthermore, localized algorithms have two attractive properties. First, because each node communicates only with other nodes in some neighborhood, the communication overhead *scales* well with increase in network size. Second, for a similar reason these algorithms are *robust* to network partitions and node failures. We are just beginning the work of validating this hypothesis through design and experimentation.

In the next section, we describe the challenges posed by the design of localized algorithms in data-centric, application-specific sensor networks.

3 Localized Algorithms for Coordination

Clustering allows sensors to efficiently coordinate their local interactions in order to achieve global goals. In particular, localized clustering can contribute to more scalable behavior as number of nodes increase, improved robustness, and more efficient resource utilization for many distributed sensor coordination tasks. One such sensor coordination task is the election of extremal sensors to form the widest baseline for locating external objects. Especially when this "triangulation" is performed frequently, it may be more energy efficient for cluster heads alone, rather than all the sensors in the network, to participate in this election. Data aggregation is a second example of the use of clustering. Consider an office environment where sensors monitor the location of various tagged objects such as projectors and books. Cluster-heads could summarize the objects located in their clusters to provide a less detailed view to distant nodes. The disseminated summary information can then be used to locate objects such as the nearest projector or a missing book.

We first present a localized clustering algorithm and later discuss an application that makes use of the clustered sensors to efficiently pinpoint the location of objects. We assume that a link level procedure is run on each sensor that adjusts the transmission power and thus the communication range to a minimum value that maintains full network connectivity. The clustering algorithm then elects cluster-head sensors such that each sensor in the multi-hop network is associated with a cluster-head sensor as its parent. The parent-child relationships are established only between sensors that are able to communicate with each other thus preventing inconsistencies due to asymmetric communication. The clusters adapt to network dynamics and changing energy levels of nodes. For simplicity, we describe a two-level cluster formation algorithm in this paper. The algorithm can be recursively applied to build a cluster hierarchy.

In our algorithm, we associate sensors at a particular level with a radius. The radius specifies the number of physical hops that a sensor's advertisements will travel. Sensors at a higher level are associated with larger radii than those at lower levels. All sensors start off at the lowest level of 0. Each sensor then sends out periodic advertisements to sensors within *radius* hops. The sensor advertisements carry the sensor's hierarchical level, parent ID (if any) and remaining energy of the sensor. Sensors then wait for a certain *wait time* that is proportional to their radius in order to allow advertisements from various sensors to reach each other. At the end of the above wait period, a level 0 sensor starts a *promotion timer* if it does not have a parent. The promotion timer is set to be inversely proportional to the sensor's remaining energy and the number of other sensors from whom level 0 advertisements were received. This would cause sensors located in relatively dense regions and with higher remaining energy to have smaller timeout values.

When the promotion timer expires, a sensor promotes itself to level 1 and starts sending periodic advertisements at the level 1 radius. In these advertisements, the newly promoted sensor lists its potential child sensors that consists of the level 0 sensors whose advertisements it previously received. Only the level 0 sensors that appear in this potential children list can choose the level 1 sensor to be their parent. This ensures that parent-child relationships are established only between sensors that can see each other's advertisements and thus are able to communicate with each other. A level 0 sensor picks the closest potential parent that it sees to be its parent. Once a level 0 sensor picks a parent, it cancels its promotion timer if running and thus drops out of the election process. After promotion, the level 1 sensors start a wait timer proportional to their new larger radius. At the

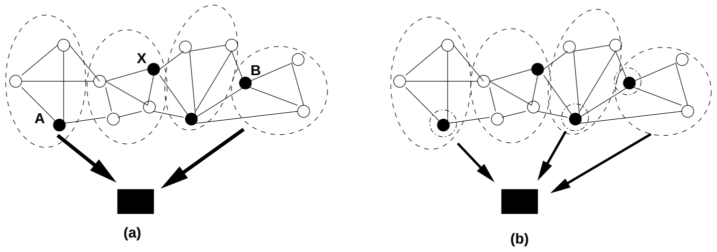


Figure 1: *Object Triangulation in Sensor Networks:* The high-level task of this sensor network is to accurately pinpoint the location of the object (represented by a dark square). Under certain assumptions, there exists a localized algorithm which can do this, which may be non-optimal under certain circumstances.

end of the wait period, the level 1 sensor may demote itself if it does not have any child sensors or if its energy level is less than a certain threshold function of its children's energy (e.g., less than 50% of the maximum energy among its children). All level 0 and level 1 sensors periodically enter the wait state. Thus, any change in network conditions, or in sensor energy levels results in re-clustering with bounded delay.

Many clustering proposals can be found in the literature. However, they do not adequately address the primary constraints of the wireless sensor networks environment. Some proposals [3, 4] exhibit non-localized behavior where nodes need to communicate with other distant node(s) to elect leaders. The *Landmark* hierarchy [5] [2] and other localized clustering proposals [6, 7] suggest significant improvements but do not handle two key design constraints: asymmetric communication in the network and limited energy of sensors. Asymmetric communications may cause a sensor in the network to have inconsistent information about its own cluster. For example, a child sensor might choose a particular cluster-head to be its parent (thereby joining its cluster) even though the cluster-head cannot see advertisements from the child sensor (due to the child's advertising radius being smaller than the longer reverse path to the parent). Energy-insensitive design may lead to quick depletion of energy levels of sensors thus reducing the life of the network.

We next illustrate an application of the clustering algorithm in pinpointing object locations. Consider the scenario shown in figure. This figure depicts a sensor network organized into clusters where no single level 1 cluster encompasses all sensors. The only active sensors in the network are the cluster-head sensors (the shaded ones) that detect an object (the dark square). Each sensor can determine the general direction of the object. The task of this sensor network is to pinpoint, in an energy-efficient manner, the exact location of the object. To accurately determine the location of the object, we need the widest possible measurement baseline. To achieve energy efficiency, we need the fewest number of sensors participating in this triangulation. That is, for the network shown in Figure 1(a), we would like to design a localized algorithm that results in cluster-heads A and B participating in the triangulation.

Assume that each sensor can determine its position in 2-space, and that each sensor can specify the approximate direction of the object relative to its own location. Then, there exists a simple rule whereby each cluster-head sensor can locally determine (based on information from neighbour cluster-heads alone) whether it should participate in the triangulation computation: If all the neighboring cluster-heads of a cluster-head sensor lie on the same side of a line drawn between the sensor and the object, then that cluster-head sensor elects itself as a participant in the computation. By this rule, for example, the cluster-head X in Figure 1(a) does not elect itself to be a participant. This rule will elect the cluster-head sensor at each extremity (or more than one if those sensors are aligned with respect to the object) Once elected, these extremal sensors report their readings to an external observer.

To implement this rule, a single message exchange between neighboring cluster-head nodes suffices. The above cluster-based approach for base-line estimaton has several nice properties. First, because these sensor algorithms use only local information, they should have generally lower energy consumption than those that entail global communication. Intuitively, these algorithms have the potential to demonstrate scaling complexity such that the overhead of the algorithm run at each sensor node is a sublinear function of the total number of sensor nodes, and is proportional to the local population density. Second, the algorithm is robust to link or node failures and network partitions. As we show below, however, it could be slightly inefficient when these pathologies occur. Third, because all communication is inherently localized, mechanisms for self-configuration can be simpler than for other networks. This enables rapid deployment and robust unattended operation. Finally, local communication and per-hop data filtering can avoid transmitting large amounts of data over long distances, thereby preserving node energy resources. Node energy resources are also better utilized since the cluster-heads adapt to changing energy levels. Essentially, the sensors in a cluster take turns at being the cluster-head based on their current energy levels thus leading to more efficient energy usage.

The previously described rules achieve our objective: electing sensors that form the longest-baseline for triangulation. So, what is hard about designing such localized algorithms? This simplified algorithm can be non-optimal under certain terrain conditions. For example, if some cluster-head sensors are behind obstacles and cannot see the object, while their neighbor cluster-heads can, the rule can cause several sensors to elect themselves (Figure 1(b)). One way to alleviate the impact of such conditions might be to allow a cluster-head to switch on some number of child sensors in its cluster to do object location. A cluster-head can then communicate to its neighbour cluster-heads that it detects an object if any

[2] Recursive application of our algorithm leads to the construction of a Landmark hierarchy

of its selected child sensors detect the object. A cluster-head that is elected to participate in the triangulation can report back readings from the extremal child sensor in its cluster that detects the object.

The preceding paragraph shows by example the difficulty of designing localized algorithms. Localized algorithms are hard to design for two main reasons. First, local algorithms must provide a desired global behavior with at best indirect global knowledge. Thus the process of crafting local algorithms from the global behavior is akin to the process of converting a centralized algorithm to a completely distributed one. The resulting rules often bear little resemblance to the original distributed computation. Second, some kinds of localized algorithms are parametrically sensitive; different choices of algorithm parameters can lead to radically different kinds of global behavior. An example of this is the reaction-diffusion systems studied by Turing [8]. It is difficult to design localized algorithms that both empirically adapt to a wide range of environments and converge to the desired global behavior over that entire range.

We believe that the following two-pronged approach can be used to overcome these difficulties.

- First, develop intuition for localized algorithms by designing and prototyping some algorithms. A class of algorithms pertinent to sensor networks that we plan to explore are *adaptive fidelity* algorithms. An adaptive fidelity algorithm is one where the quality (fidelity) of the answer can be traded against battery lifetime, network bandwidth, or number of active sensors. (Of course, the resulting fidelity must still fall within acceptable bounds.) One illustration of this idea is shown in Figure 2. Consider, as before, a sensor network that determines the exact location of sensed objects. Now, instead of every cluster-head sensor participating in the baseline determination, some cluster-head sensors turn themselves off to conserve power (the grayed sensors in Figure 2(a)). This has the effect of a smaller baseline and, consequently, lower fidelity triangulation. Subsequently, as some of the currently active sensors "die" (due to battery failure or accidents, shown with dotted lines), new cluster-heads are elected and their neighbor cluster-heads take over and continue the baseline determination for other objects (Figure 2(b)). If designed correctly, this can result in only slightly degraded baseline determination but a sensor network with nearly double the lifetime.

- Second, develop techniques for characterizing the performance of localized algorithms. Localized algorithms exhibit good robustness and scaling properties. To achieve these properties, however, these algorithms may sacrifice resource utilization or sensing fidelity, responsiveness, or immunity to cascading failures. It is desirable to develop a methodology that can characterize these tradeoffs.

4 Directed Diffusion

Localized algorithms have many desirable properties in the context of sensor networks. However, these algorithms are hard to design and characterize. It would be convenient to define a set of abstractions that describe the communication patterns underlying many localized algorithms. In this section, we briefly explore one such set, *directed diffusion*.

A sensor network based on directed diffusion exhibits the following properties. Each sensor node *names* data that it generates with one or more attributes. Other nodes may express *interests*, based on these attributes. Network nodes propagate interests. Interests establish *gradients* that direct the diffusion of data. As it propagates, data may be *locally transformed* at each node. We explain these concepts with the aid of the simple scenario shown in Figure 3.

Figure 3 shows a sensor network in which each node can detect motion (and possibly other information) within some vicinity. One or more *sink* nodes may query the sensor network for motion information from a particular section of the terrain (*e.g.,* from the southeast quadrant). One goal of the sensor network is to robustly compute a data dissemination path from source to sink. The following four paragraphs describe this path finding algorithm using the diffusion model.

Attribute-based naming is the first characteristic of directed diffusion systems. In our example, each sensor names data that it generates using a single attribute motion, which has a geographic location (*e.g.,* latitude/ longitude, or relative location with respect to some landmark) as its value. In general, motion data may be described using several attributes: (type=seismic, id=12, timestamp=99.01.22/21:08:15, location=75N/120E, footprint= vehicle/wheeled/over-40-ton). As this example shows, an attribute's value may also have a hierarchical structure.

A sink (such as the node a in Figure 3(a)) may query for motion information by disseminating an *interest*. Syntactically, an interest is simply a range of values for one or more attributes. In our example, the node a specifies south-east quadrant as the value of the motion attribute in its interest. More generally, interests may have complex structure (type=seismic, timestamp=99.01.22/*, location=70-80N/100-140E).

Each node disseminates interests based on the contents of the interest. In our example, intermediate nodes send the interest towards the neighbor in the direction of the south-east quadrant. Conceptually, the path of interest propagation sets up a reverse data path *for data that matches the interest*. Then, when nodes x and y in the southeast quadrant detect motion, the motion signature travels towards a along data propagation path.

In the diffusion model, we say that this data propagation path has an associated *gradient*. The notion of gradient is useful when, for robustness, each intermediate node propagates the interest towards multiple neighbors (Figure 3(b)). We say that the "strength" of the interest is different towards different neighbors, resulting in source-to-sink paths with different gradients. In its simplest form, a gradient is a scalar quantity. Negative gradients inhibit the distribution of data along a particular path, and positive gradients encourage the transmission of data along the path. The value of a particular gradient may have application-specific semantics. In our motion sensing scenario, for instance, if a node has two outgoing paths, one with a gradient of 0.8 and another with a gradient of 0.4, then the node may send twice as much detail along the higher gradient path than along the lower.

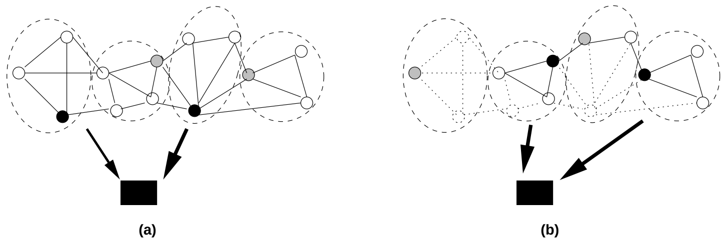


Figure 2: *Adaptive Fidelity Algorithms:* These localized algorithms selectively turn off some sensors to conserve system resources. As individual sensors die, others take their place. This technique extends network lifetime while possibly reducing sensing fidelity.

Although not described in our scenario, the diffusion model allows intermediate nodes *to cache or locally transform* (*e.g.,* aggregate) data. This aspect of the model leverages the application-specificity that is possible in sensor networks. Caching and aggregation can increase the efficiency, robustness and scalability of coordination. Locally cached data may be accessed by other users with lower energy consumption than if the data were to be resent end to end. Intermediate node storage increases availability of the data, thereby improving robustness. Finally, intermediate nodes can increase the scalability of coordination by using cached information to carefully direct interest propagation.

The diffusion model's data naming and local data transformation features capture the data-centricity and application-specificity inherent in sensor networks (Section 2). The model allows for neighbor-to-neighbor interest propagation and local data transformation rules; these elements capture the communication patterns expected of localized algorithms. Finally, gradients model the network-wide results of these local interactions. We believe that the diffusion model can be used to describe not only other data dissemination patterns (shortest path multicast trees, energy-efficient spanning tree multicast), but also other coordination algorithms (such as the triangulation example of Section 3).

To the extent that diffusion primitives help set up communication paths between nodes in sensor networks, they play the role of the routing system in traditional data networks. Because we expect most sensor applications to be localized, we think sensor networks are unlikely to incorporate a reactive routing system like that found in today's Internet. Instead, we expect the routing function in a sensor network to be tightly integrated with the application. Applications will use a combination of proactive and reactive schemes [9, 10, 11, 12, 13, 14] to achieve energy-efficient communication.

5 Related Work

Several projects have already demonstrated the feasibility of low power integrated sensors [1, 15, 16, 17] and MEMS-based microsensors [18].

Different contexts have been proposed for the application of small-scale networks of devices for sensing and actuation tasks.

Various process control and automation tasks in factories have traditionally used networks of embedded systems known as *control networks* [19] to efficiently perform their monitoring tasks. These control networks today consist of a centralized processor coordinating the communication between many tens of sensors and actuators.

Simple networks of integrated sensors have been proposed for military awareness situations [1]. These networks schedule sensor transmission using TDMA methods, and provide for automatic discovery of neighbors and cooperative detection by sensors.

The Piconet [20] project is developing a prototype embedded network. Piconet is a low-rate (about 40 Kb/s), low-range (5 meters) ad-hoc radio network. Sensors can use the Piconet module to enable wireless connectivity. The project has designed a low level radio protocol for communication between various embedded objects. They have also prototyped some interesting home and office information discovery applications [21].

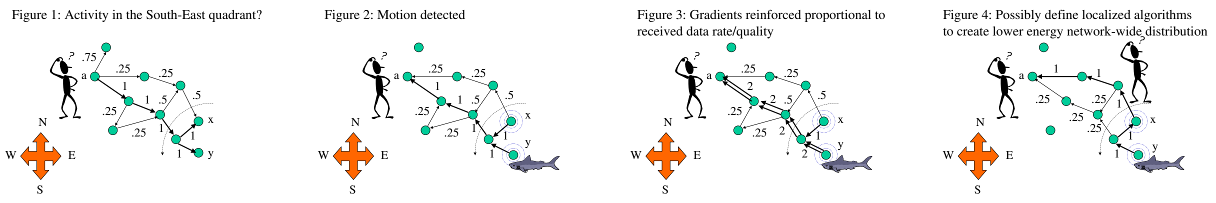
Embedded networks have long been used for personal location, equipment tracking, or information gathering. Active badges [22], the ParcTab [23], the ORL location system [24], the PinPoint Positioning System (LPS) [25], and the Factoid [26] are all examples of this class of networked sensors.

The Ubiquitous Computing project at Xerox PARC [27] explored a generalized version of these applications: seamless integration of computing devices into the environment. Finally, the MIT smart room project [28] and the Forest of Sensors project [29] analyze data from video images to make inferences about the presence or absence of various objects or people.

Several research efforts provide insights into the design of coordination algorithms for sensor networks.

Biological Systems: The reaction-diffusion models for morphogenesis [8] describe the mechanism by which the initially homogeneous human cells eventually differentiate themselves to form various tissues and organs. Models of ant colonies [30] are based on the fact that the almost blind ants seem to be able to find shortest paths to destinations using the *pheromone* trails deposited by other ants as the only information. These contain some elements of the diffusion model: localized interactions that lead to the formation of network gradients.

Distributed robotics: Of relevance here are coordination protocols being designed for troops of low-cost robots to explore and acquire maps of unknown environments [31]. In these schemes, the robots move about in a partially random fashion and cooperate with each other by transferring the collected information when they meet. After the robots complete their exploratory run, they deliver their partial maps to a host computer that then derives a complete map of the area.



(a) (b) (c) (d)

Figure 3: *Diffusion for path finding:* This illustrates the basic diffusion constructs: data naming, interests that set up gradients, data diffusing along gradients.

Amorphous Computing: Coore et al. [6] present distributed algorithms to organize unstructured processing elements into a hierarchy of cooperative groups, called an AC hierarchy. However adaptation to network failures has not been discussed.

Internet Multicast and Web Caching: The design of Internet multicast tools [32] has led to significantly improved understanding of robust application design. Specifically, techniques such as lightweight sessions and soft state management are also applicable in the sensor network context. In adaptive Web caching schemes [33], cache servers self-organize themselves into overlapping multicast groups. The mesh of overlapping groups form a scalable, implicit hierarchy that is used to diffuse popular web content towards the demand. These techniques are relevant because frequently queried information in sensor networks also needs to be efficiently diffused to the interested users.

Perhaps most directly relevant to sensor networks is ongoing work on *ad-hoc networks*. Ad-hoc networks refer to self-organizing networks of mobile wireless nodes that do not depend on any fixed infrastructure. A central focus of the work on ad-hoc networks has been the design of *proactive* [9, 10, 11] and *reactive* [12] routing protocols, and combinations thereof [13, 14]. Proactive routing protocols continuously compute routes to all nodes so that a route is already available when a packet needs to be sent to a particular node. Such continuous route computation may energy-inefficient. Reactive routing protocols on the other hand start a route computation process only when a packet needs to be sent to some other node. However, they may redundantly flood requests throughout the network. A combination proactive and reactive scheme may overcome these disadvantages, but may still not perform as effectively as schemes that use application knowledge to route data and queries.

Acknowledgements

Many of the ideas presented in this paper were the outcome of a DARPA ISAT study, chaired by one of the authors (Estrin). Two of the other authors (Govindan, Heidemann) participated in the study. The other study participants were: Rod Brooks, Mani Chandy, Dave Clark, Steve Deering, Rich Ivanetich, Van Jacobson, Butler Lampson, Bill Mark, Vern Paxson, Greg Pottie, and Mark Weiser. Others who contributed to the ideas described in this paper include: Cengiz Alaettinoglu, Nirupama Bulusu, Padma Haldar, and Chalermek Intanagonwiwat. The authors are also grateful to the anonymous reviewers for suggestions which led to the improved readability of the paper.

References

[1] G. Pottie, W. Kaiser, L. Clare and H. Marcy. Wireless Integrated Network Sensors. Submitted for publication, 1998.

[2] S. Floyd, V. Jacobson, C. Liu, S. Mccanne and L. Zhang. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *IEEE/ACM Transactions on Networking*, November 1996.

[3] David G. Thaler and Chinya V. Ravishankar. Distributed top-down hierarchy construction. In *Proc. of the IEEE INFOCOM*, 1998.

[4] P. Sharma, D. Estrin, S. Floyd and L. Zhang. Scalable Session Messages in SRM. Technical Report 98-670, University of Southern California, 1998.

[5] Paul F. Tsuchiya. The Landmark hierarchy: A new hierarchy for routing in very large networks. In *Proc. of the ACM SIGCOMM*, 1988.

[6] D. Coore, R. Nagpal and R. Weiss. Paradigms for Structure in an Amorphous Computer. Technical Report 1614, Massachusetts Institute of Technology Artificial Intelligence Laboratory, October 1997.

[7] A. Rosenstein, J. Li and S.Y. Tong. MASH: The Multicasting Archie Server Hierarchy. *ACM Computer Communication Review*, 27(3), July 1997.

[8] A. M. Turing. The chemical basis of morphogenesis. *Phil. Trans. Roy. Soc. of London*, Series B: Biological Sciences(237):37–72, 1952.
A reaction-diffusion model for development.

[9] J.J. Garcia-Luna-Aceves. Loop-Free Routing using Diffusing Computations. *IEEE/ACM Transactions on Networking*, February 1993.

[10] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. of the ACM SIGCOMM*, October 1994.

[11] P. Jacquet, P. Muhlethaler and A. Qayyum. Optimized Link State Routing Protocol. *Internet Draft, draft-ietf-manet-olsr-00.txt*, November 1998. Work in progress.

[12] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. of the ACM/IEEE MobiCom*, October 1998.

[13] Z. Haas and M. Pearlman. The Performance of Query Control Schemes for the Zone Routing Protocol. In *Proc. of the ACM SIGCOMM, Vancouver, Canada*, September 1998.

[14] M. Jiang, J. Li and Y. Tay. Cluster Based Routing Protocol (CBRP) Functional Specification. *Internet Draft, draft-ietf-manet-cbrp-spec-00.txt*, August 1998. Work in progress.

[15] The WINS project. `http://www.janet.ucla.edu/lpe.lwim/`.

[16] The Ultra Low Power Wireless Sensors project. `http://www-mtl.mit.edu/~jimg/project_top.html`.

[17] Wearable Computing. `http://www.cs.cmu.edu/~wearable/`.

[18] MEMS Technology Applications Center. `http://mems.mcnc.org`.

[19] Echelon, The LonWorks Company. LonWorks Solutions. `http://www.echelon.com/Solutions/`.

[20] F. Bennett, D. Clarke, J. Evans, A. Hopper, A. Jones and D. Leask. Piconet: Embedded Mobile Networking. *IEEE Personal Communications*, 4(5), October 1997.

[21] F. Stajano and A. Jones. The Thinnest of Clients: Controlling it all via cellphone. *ACM Mobile Computing and Communications Review*, October 1998.

[22] R. Want et al. The Active Badge Location System. *ACM Transactions on Information Systems*, January 1992.

[23] N. Adams. An Infrared Network for Mobile Computers. In *Proc. USENIX Mobile and Location-Independent Comp. Symp., Cambridge, MA*, August 1993.

[24] A. Ward, A. Jones and A. Hopper. A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5), October 1997.

[25] J. Werb and C. Lanzl. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, September 1998.

[26] The Factoid project. `http://www.research.digital.com/wrl/projects/Factoid/index.html`.

[27] M. Weiser. The Computer for the 21st Century. *Scientific American*, September 1991.

[28] A. Pentland. Machine Understanding of Human Action. In *Proceedings of 7th International Forum Frontier of Telecommunication Tech.*, November 1995.

[29] W. Grimson, C. Stauffer, R. Romano and L. Lee. Using adaptive tracking to classify and monitor activities in a site. *CVPR*, 1998.

[30] M. Dorigo, V. Maniezzo and A. Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 26(1), 1996.

[31] M. López-Sánchez, R. López de Mántaras, and C. Sierra. Incremental map generation by low const robots based on possibility/necessity grids. In *Proc. of the Conf. on Uncertainty in Artif. Intell.*, pages 351–357, Providence, RI, 1997.

[32] S. McCanne. Scalable Multimedia Communication with Internet Multicast, Lightweight Sessions, and the Mbone. Technical Report Technical Report CSD 98-1002, University of California, Berkeley, March 1998.

[33] L. Zhang, S. Michel, S. Floyd, V. Jacobson, K. Nguyen and A. Rosenstein. Adaptive Web Caching: Towards a New Global Caching Architecture. In *Third International Caching Workshop*, June 1998. Available from `http://irl.cs.ucla.edu/AWC/`.