I. Proposal to the Defense Advanced Research Projects Agency (DARPA)
BAA 99-16: Sensor Information Technology
Topic/Areas: (2) Sensor Algorithms and Applications,
(3) Distribution and Aggregation Technology

SCADDS: Scalable Coordination Architectures for Deeply Distributed Systems

BY

INFORMATION SCIENCES INSTITUTE
UNIVERSITY OF SOUTHERN CALIFORNIA
LOS ANGELES, CALIFORNIA 90089-1147

PROPOSAL

Period of Performance: 4/1/99 - 3/31/2002
YEAR 1: $922,171 YEAR 2: $980,595 YEAR 3: $994,149

Total Base Cost $2,896,915

ISI Technical Contact:

Deborah Estrin
(310) 822-1511, estrin@isi.edu
FAX: (310) 823-6714

ISI Administrative Contact:

Margie Schroeder
(310) 822-1511, margiej@isi.edu
FAX: (310) 823-6714

Type of Business: Other Educational

Principal Investigators
Deborah Estrin, Ramesh Govindan, John Heidemann

Lloyd Armstrong, Jr.
Provost and Senior Vice President for Academic Affairs
University of Southern California

A. Innovative Claims

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 military reconnaissance, biological field experimentation, industrial assembly line practices, and others.

Our proposal will enable rapid, completely unattended deployment of application-specific sensor networks. Our research will target the following scenario: Several thousand sensors are rapidly deployed 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 the sensing accuracy to the remaining total resources, and re-organize upon sensor failure. Moreover, unlike any existing technology, additional sensors can be added to the network, or moved around within the network, and the sensors will re-organize to take advantage of the added system resources. Such large scale, dynamic, and robust sensor colonies can be deployed in inhospitable physical environments: remote geographic regions, toxic urban locations etc. In addition, they also enable low maintenance sensing in more benign, but less accessible, environments: large industrial plants, aircraft interiors etc. Ongoing programs and commercial efforts provide preliminary hardware to enable these applications. However, radically new coordination architectures are required to make this hardware deployable at the scales required.

Our research contributions will focus on filling this gap by providing mechanisms that enable large numbers of distributed devices, deployed in highly dynamic environments, to coordinate on a specific task. Three primary outcomes of this research will be:

- Primitives for Scalable Coordination: These **directed diffusion** primitives can be thought of as a language for expressing coordination algorithms. They achieve robustness and scale by forcing the sensor network designers to express their algorithm in terms of local sensor node interactions.
- Techniques for Localized Algorithm Design and Analysis: These techniques will be captured in a library comprising sample localized algorithms, and design intuition about the performance and stability implications of local interactions. Designers with little networking experience, but more application domain knowledge can use this library to design networked sensors.
- Application Support Mechanisms: Network location, clustering and security mechanisms will form reusable components of sensor networks. These mechanisms will themselves employ directed diffusion primitives.

Many of these research directions were developed during our participation in the 1998 Simple Systems ISAT study. We expect to prototype many of these coordination algorithms in the context of an object tracking and monitoring application. In conducting our research we hope to use the simulation and emulation infrastructure proposed in SEEDS (Heidemann/Fall/Estrin), and the experimental platform proposed in SNST (Mankin/Estrin/Heidemann).

Our methodology for designing sensor network coordination, and our demonstration of this in the context of sample applications, will be the first rigorous attempt to extend lessons from Internet design to "deeply-distributed" dynamic systems.

B. Technical Rationale

Powerful yet inexpensive embedded processors and sensors and wireless networking standards have enabled sensor networking**.** Groups of small nodes should now be able to collaborate to track and monitor objects, identify vehicles, and detect intruders. We can imagine throwing hundreds of sensor nodes down at a disaster site to monitor chemical leaks, or interacting with dozens of computers embedded in a meeting room and carried by other attendees.

Although the prototype hardware exists and improvements are underway, current software techniques to achieve these ends is woefully inadequate at addressing the key characteristics of these environments: very large scale (many physically distributed nodes) and environment dynamics (node movement, failure, addition, communication changes, etc.).

SCADDS proposes an innovative architecture to meet these challenges: data diffusion. In a network of thousands of tiny sensors, the data is important, not the individual nodes, therefore we turn networking on its head by making it data-centric. We believe that diffusion protocols will scale to sensor nets of 1000s of nodes because they are based on **local algorithms** rather than centralized approaches.

SCADDS' starting assumption is that the power of sensor networks is greatly increased by inter-node **coordination–**individual sensors cooperating to perform a larger sensing task. We use the term **coordination** to mean information exchange between two or more nodes, and subsequent computation based on that information. By this definition, Internet unicast and multicast routing are examples of coordination. So is Internet web caching.

A **coordination architecture** defines which nodes may participate in the information exchange, how far the information from a particular node disseminates, etc. For example, a hierarchical Web document caching scheme defines a parent-child communication and caching scope relationship. Link-state unicast routing is a different example of a coordination architecture, in which a node's state is flooded to all other nodes within an administrative region.

This proposal expands on the constraints unique to sensor nets. We then consider how directed diffusion, based on local algorithms, addresses these constraints. Supporting mechanisms such as location awareness, clustering, and security protocols must be adapted or invented to support environments of thousands of autonomous nodes.

B.1 Unique Constraints of Large Sensor Nets

Sensor networks today are collections of tens of small, often power-constrained, mobile, and wireless nodes that interact to detect or effect (via actuators) events in a small area. Nodes in these systems will be heterogeneous, having a range of sensing, actuation and communication capabilities. Tomorrow's sensor nets will share many of these characteristics, but will be much larger, hundreds to tens of thousands of nodes.

SCADDS will explore and demonstrate scalable coordination architectures for these very large dynamic and deeply distributed sensor networks. The two most challenging aspects these systems will face are scalability and environment dynamics**.**

The problem of large scale is inherent to sensor nets. Sensors have limited ranges and sometimes must be directly attached to the device they're observing. Actuators nearly always are co-located with their target object. Furthermore, redundancy and load balancing require multiple nearby nodes. These reasons make it mandatory to employ large numbers of nodes both to leverage off the pervasive physical locality between nodes and the physical phenomena being monitored, and to allow these systems to exploit the concept of simple, modular, reusable component design.

Environment dynamics follows both from scale and directly from the problem domain of sensor nets. Any large scale system must cope with node failure; the very large numbers of nodes in sensor nets implies that partial failure will be a constant condition rather than an exception. More fundamentally, dynamics is inherent to most environments we come across in the physical world, where sensor networks are likely to have the greatest impact. Wireless communication can be error prone, nodes will fail as they run out of power, nodes will be added to gain improved sensing resolution near an important event. Fortunately, the sensor networks domain allows small, interchangeable, redundant sensing nodes which, properly designed, can mask failure and accommodate environmental changes.

A typical application scenario may consist of a heterogeneous and dynamic set of participating nodes being deployed in rapidly changing topologies with varying operating conditions, with new set of tasks being assigned to the participating nodes, coupled with dynamic availability of system resources.

The large number of nodes as well as difficulty in predicting the dynamics of an environment with a high degree of accuracy, precludes manually configured systems. Hence devices must learn to adapt automatically to the changing environment.

These constraints also preclude centralized approaches for all but the simplest scenarios. Sensor and actuator ranges require many distributed nodes. Communication latency could prevent centralized systems from responding to environment dynamics quickly enough. Lack of robustness, including system breakdowns due to single point failure is another major drawback of the centralized approach.

We believe there is much to learn from Internet-based protocols. For example, the ad hoc routing mechanisms being developed for IP networking applications capture the need for auto-configuration, adaptation to dynamic topologies and accommodation of mobility, wireless links and some power constraints. However, the goal of ad hoc routing, to support end to end IP communications among mobile PC-class nodes (from PDAs to workstations) other handhelds is fundamentally different in its goals in that it assumes an end-node-centric architecture. In sensor nets, the nodes are interchangeable, only the data has value. SCADDS reflects this data-attribute-based addressing and data-specific processing (such as detection, identification, and aggregation) at intermediate nodes.

We propose to develop a coordination architecture for sensor networks that will address the following basic design principles:

- Nodes will be designed to be self-contained and will function independently of one another. Each node will have a clean functional model allowing composition and analysis of the whole system based on simple component design.   The nodes will also be extendable and will have flexible specialization allowing optimization for varying environmental conditions.
- A coordination architecture based on data diffusion will control node interaction. Nodes will interact locally within their network neighborhood to achieve the desired global behavior while accommodating simple incremental deployment, robust operation and scalability in overhead (including power). Nodes will adapt their behavior to the sensed environment and the behavior of their neighbors. This empirical adaptation will allow them to converge to the operating point that is appropriate to the particular state of the environment and network.

Data-centric applications will be demonstrated over this architecture. Nodes will be anonymous and interchangeable. For example, a large sensor net application should not be designed to answer questions like: what is the temperature reading at sensor #27 now? Instead, the DATA generated by these devices should be the focus of all communication, independent of which node collected the data originally, and which nodes currently store it.

Autonomous unattended nodes, diffusion-based coordination, and data-centric applications, represent a paradigm shift away from traditional personal computing. At the same time, we see significant benefit in departing from traditional networking models by viewing the sensor networks as distributed applications instead of lower level transport services for end to end applications. SCADDS will focus on the directed diffusion techniques, adaptive fidelity algorithms, and the system building blocks needed to support this new paradigm. The SCADDS new architecture will enable applications to run on 1000-node, deeply embedded, sensor networks that can be constructed in a cost-effective and timely manner.

B.2 Directed diffusion as a scalable coordination architecture

We have argued in the previous section that traditional coordination architectures are inappropriate for deeply distributed sensor networks. Nevertheless, we can adapt some of the principles behind these architectures to sensor network design. For example, unicast routing achieves its global scaling by applying hierarchical aggregation, localized exchange of information between neighbors, and relatively simple local computation based on data and local events. Similarly, multicast routing protocols are also defined by simple local rules and local exchange of control information. The IP multicast model scales to very large receiver sets because the distribution tree is set up based on hop-by-hop, rather than end-to-end, coordination. Adaptive Web caching, with its local opportunistic dissemination of cached pages, provides another model for designing large, dynamic sensor networks.

B.2.1 Directed Diffusion Concepts

Figure 1: Activity in the southeast quadrant ?

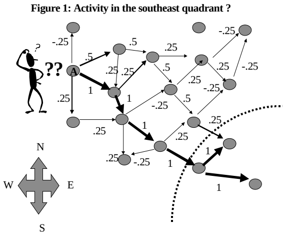


Figure 2: Motion detected

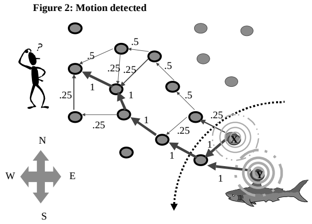


Figure 3: Gradients reenforced proportional to received data rate/quality

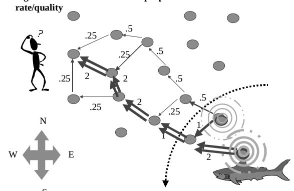
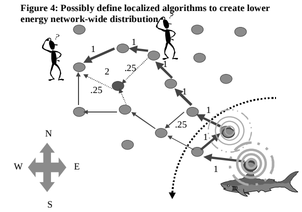


Figure 4: Possibly define localized algorithms to create lower energy network-wide distribution



We will explore directed diffusion as an alternative coordination architecture. We will also develop a tunable diffusion substrate upon which different applications can implement their coordination mechanisms. Directed diffusion was inspired by biological systems and first suggested for application to this domain by Van Jacobson (based on successful application of these ideas to reliable multicast and web caching).
What do we mean by directed diffusion? Briefly, a sensor network based on directed diffusion exhibits the following properties:

- Each sensor node tags data that it generates with one or more attributes.
- Other nodes may express interests, based on these attributes.
- The network establishes **gradients** that direct the diffusion of data.
- As it propagates, data may be locally transformed at each node.

Before describing these concepts in greater detail, we present a scenario (Figure [1]) to illustrate them.

Figure[1] shows a network of nodes, each deployed to pick up motion signals—and possibly other information—within their individual 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). The following list explains the scenario in terms of the diffusion terminology we have introduced in the previous paragraph:

- Node [a] triggers a query for motion in the southeast quadrant. In diffusion terms, this query is an interest for any data that contains a “motion” attribute and whose location attribute matches “the southeast quadrant”.
- Using the location information specified in the query, intermediate nodes direct this query towards the southeast quadrant (Figure [2]). In this manner, queries converge towards the southeastern corner of the sensor network. Queries may be propagated to multiple neighbors, even though one neighbor may seem on the most direct path to the region of interest; in this case, queries sent to less direct neighbors have lower values.
- As the queries propagate, they set up reverse gradients for data with the “motion attribute”. The value of the gradient in this example is set proportionally to the value expressed in the query.

Thus, when source node [x] and [y] detect and advertise motion, the motion signature travels from the southeast quadrant along gradients etched out by queries.

Several variants on this scenario are possible. For example, intermediate nodes may “smear” out the queries a little, resulting in multiple data transmission gradients (Figure [2]). Different gradients may have different “slopes”---aslope is indicative of the desirability of using that path to transmit. (In what follows, we use the word gradient to mean the path established for data, as well the “slope” of that path.) Source nodes may either send different data along different paths, or may send the same data in different resolutions along different paths. Sinks may reinforce one or more of these multiple paths as shown in Figure [3]. Finally, localized algorithms may be used to establish more efficient data distribution as shown in fig [4].

We now discuss the key principles of directed diffusion (data naming, gradient establishment, and data aggregation) in greater detail.

Data Naming

Directed diffusion decouples data from its location (i.e. where the data was generated). In sensor networks, in particular, this decoupling is key to robustness and scalability. Data can continue to “live” in other nodes inside the network even if the originating node is unreachable. Furthermore, data can be cached (and aggregated) inside the network to enable scalable dissemination.

To achieve this decoupling, sensors name data (representing objects and events) using attributes. Data attributes are determined based on the type of information that is expected to be disseminated in and utilized from the system. For example, a movement-detecting system may have attributes like: “Type=seismic, id=12,time-stamp=98.07.22/21:08:15”. Attributes may be hierarchically defined as in: ”location=75N/120E, detected=vehicle/wheeled/over 40-ton”. Data aggregation would be possible with attribute naming like:”location=42-4N/152-178E, detected=vehicle/*”. Attribute-based naming can be used to control the granularity of received data by specifying coarse (e.g. ”detected=*”) or fine grained attribute (e.g. ”detected=vehicle/wheeled/*”) descriptions.

Interests

Data consumers diffuse interests in data. Interests are named by specifying value ranges for attribute values. For example, the interest defined by: “location=40-60N/110-130E, detected=vehicle/*” specifies an interest in any vehicular movement in a particular geographic area. Interests are propagated via a sequence of local interactions. Each node that receives an interest makes a local decision about which direction, and how far, to propagate an interest. Different local decisions can result in different data distribution paths: shared distribution trees, multi-path node-to-node distribution etc. These local decisions may be based on the attributes specified in the interest. For example, queries for fire detection such as, "query=fire? location=south-central" would be allowed to propagate in a way such that they finally converge in the south-central locality of given area. Queries such as "Query=Traffic conditions?, location=all" would be propagated to the whole area covered by the system.

Interest propagation sets up paths in the network. In addition, each hop of the path is associated with a quantity called the *gradient.* Gradients can selectively inhibit or channel the distribution of data. In its simplest form, a gradient is a scalar quantity. In general, 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. Gradients are a powerful design abstraction; mechanism designers are forced to express their coordination algorithms in terms of local interactions and local decisions. These will result in significantly more robust and scalable coordination, as we have argued above.

Finally, interests will be continually *refreshed*, to ensure robustness in the face of network partitions or node failures. An interesting research question is: How will nodes adapt the refresh rate in order to conserve resources?

<u>Local aggregation/transformation</u>
While the data propagates down established gradients, it may be aggregated, transformed, or cached locally. This local processing of the data has two important benefits: reduced power consumption and increased system robustness.

Sending high bandwidth data over long distances consumes enormous amounts of power. Alternatively, data can be locally processed and smaller "decisions" or summary information can be propagated to neighbors. In this way nodes trade local CPU for communication. This is a desirable tradeoff---local processing is getting cheaper and cheaper, while the amount of power it takes to communicate more data over a large radius is not dropping at the same rate.

Caching and aggregation can also 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, as we described above, caching can increasing the salability of coordination by enabling directed propagation of interests.

B.2.2 Design Space

There are many dimensions to the design of diffusion algorithms and different applications will want to make different tradeoffs. We discuss the design space in terms of the gradient mechanisms themselves and the way in which data is disseminated in the presence of gradients. Our proposed research will not only articulate this design space more completely, but also elaborate some of the points in this space.

<u>Gradient mechanism</u>
In the examples we described in the previous section, interests are propagated based on the physical location they specify. In general, however, interests may not contain a location attribute. Consider, for example, an interest that says "tell me about any kind of chemical spill anywhere". Is broadcasting the

interest the only possible alternative for such interests? No. Other possibilities, which we intend to investigate, include:

- Disseminating interests within a limited scope. This is useful in situations where we are interested, for example, in knowing about at least one chemical spill.
- Pre-distributing summary information about events to help guide interests. For example, suppose clusters of sensors (see the Clustering section) occasionally summarize the existence of chemicals in their vicinity. Each node may send interests about chemical spills in the direction that these summaries were heard from.
- Using cached information to direct interests. For example, suppose an earlier, location-based interest had elicited data about chemicals. A node that had cached this data may send interests for chemical spills in the direction that the data originally came from. We say that this data acts as an attractor. Such attractors can be used to build special data distribution patterns (such as minimal-power spanning trees). While the particular realization of the attractor mechanism would be application specific, we expect many aspects of the approach to be common across applications.

<u>Data dissemination</u>

Recall that each hop of a path has an associated gradient, usually a scalar quantity. As we have said above, the interpretation of a gradient is application specific. Some applications might be designed to forward data deterministically only along paths with the highest gradient. Others may choose to probabilistically send data along different paths–the likelihood of selecting a path for sending a data item is proportional to the path's gradient. This latter approach ensures load balancing, while the former offers more predictable behavior and perhaps steady-state efficiency.

Data may be sent entirely over a single path or may employ multiple paths. Moreover, given the multi-path approach, a single one of the multiple paths may be used or data may be sent redundantly down more than one path. In highly volatile environments where change in topology, location of sinks, and events, are the norm, the robustness offered by multi-path dissemination may exceed the overhead associated with redundant communication or storage. On the flip side, resource constraints in some cases may make redundant dissemination an unaffordable luxury.

For some applications, data can be represented at several resolutions. In this case, different gradient strengths may dictate the level of resolution of the forwarded signal. The higher the gradient, the higher the resolution at which data is sent along that path. Video is not the only example of data that can be represented at multiple resolutions; sensor sampling rate is another.

<u>Reinforcement mechanisms</u>

In our early exploration of these algorithms we have identified reinforcement as an important connection between gradients and the dissemination of data. Probabilistic gradient establishment protocols can set up multiple data dissemination paths. Reinforcement mechanisms can help reduce the number of paths that are used in steady state. For example, if initial interests are propagated in multiple directions, the path that delivers the highest resolution (e.g., data rate) can be reinforced by increasing the interest level along that path; while reducing the interest levels sent along other paths.

B.2.3 Evaluation

Sensor networks and other forms of deeply distributed systems require that we develop new evaluation metrics. Traditional networks appropriately emphasize steady state efficiency because dynamics are the exception, not the rule. While some sensor network applications will run in relatively stable environments, others will be subject to continuous movement, communication interference, and varying physical events. The latter networks should not be designed to optimize steady state efficiency when a significant portion of time is spent in a dynamic state. For example, the steady state efficiency of multi-path data distribution may be lower than single path data distribution. However, if evaluated in the context of continuously changing

communication reachability, the multi-path distribution approach may prove the more efficient approach because the data needs not be pulled all the way down from the source after a topology change.
Along side our investigation of coordination architectures, we will refine the evaluation methods used.

B.3 Design and Analysis of Localized Algorithms

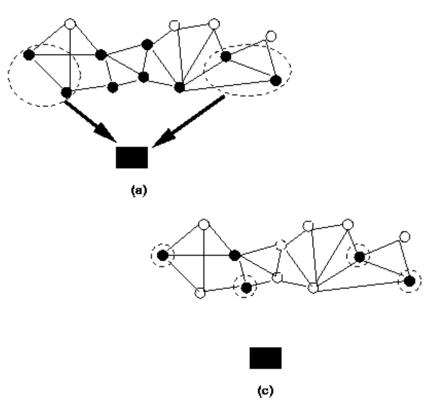
Directed diffusion primitives can be used to express algorithms that set up constraint-based data distribution paths in sensor networks. One example is the energy-efficient spanning tree described in the previous section. Establishing data dissemination paths is one instance of a class of algorithms for network node coordination. Diffusion primitives can also be used for expressing these coordination algorithms. A coordination algorithm is one that allows nodes in a sensor network to collectively perform a single task: examples of such tasks include cluster formation, and collaborative beam-forming.

There are several approaches to designing such coordination algorithms—ranging from the completely centralized to the distributed. In this proposal, we consider a sub-class of distributed coordination algorithms that we call *localized* algorithms. These algorithms achieve a global objective using only simple, localized (within a restricted vicinity) interactions. Such algorithms exhibit two desirable properties. First, with localized interactions, communication overhead increases slowly with network size. This results in good scaling behavior. Second, localized interactions can improve the robustness of the resulting coordination to a variety of failures (node and link failures, network partitions etc.). Intuitively, this is because the impact of a failure is, by definition, localized.

Localized algorithms raise two important research issues. Decomposing a global objective into localized interactions is often extremely difficult. Characterizing the performance of a localized algorithm is equally hard. It will be several years before we can hope to develop a methodology for designing localized interactions, or a calculus for characterizing the resulting performance. In our proposal, we hope to make initial contributions towards the development of these. The following sections discuss these two issues in greater detail.

B.3.1 Localized Algorithms: An Example

To give the reader an example of a localized coordination algorithm, consider the scenario described in Figure 5(a). This figure depicts a sensor network in which some subset of the sensors (the shaded ones) 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 5, we'd like to design a localized algorithm that results in the sensors included in the dashed ellipses participating in the triangulation.

What localized algorithm achieves this? Assume that each sensor can determine its position in 2-space. Assume further that each sensor can specify the approximate direction of the object relative to its own location. Then, there exists a simple rule whereby each sensor can *locally* determine (based on neighbor information alone) whether it should participate in the triangulation computation:
If all neighbors of a sensor lie on the same side of
a line drawn between the sensor the object, then
that sensor elects itself as a participant in the computation.
By this rule, for example, the sensor *X* in Figure 5(b) does *not* elect itself to be a participant. Note that this rule will elect one sensor at each extremity (or more than one if those sensors are aligned with respect to the object). In practice, we're likely to incorporate some fuzziness to the rule that allows a small number of extremal sensors to elect themselves.

To implement this rule, a single message exchange between neighboring nodes suffices. This approach has nice several nice properties:
- The complexity of the algorithm is independent of the number of sensor nodes. This very desirable scaling property is crucial for large, rapidly deployed sensor networks.
- The algorithm is robust to link or node failures and network partitions. It could, however, be slightly inefficient when these pathologies occur. For example, under a single network partition, the algorithm could result in four clusters, instead of two.

Certainly, the rule described above achieves 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 sensors are behind obstacles and cannot see the object, while their neighbors can, the rule can cause several sensors to elect themselves (Figure 5c).

One way to alleviate the impact of such conditions might be to allow a node to diffuse information about itself some distance beyond its immediate neighbors. This diffusion distance may be determined by **localized empirical adaptation**—a node that does not detect an object, but notices that its neighbor has, might choose to propagate information about that neighbor to its other neighbors (as a way of possibly inhibiting spurious self-election). Notice that this sort of behavior is well-explained using the diffusion abstraction.

In this section, we have shown by example of the difficulty of designing localized algorithms. Roughly speaking, localized algorithms are hard to design, for two reasons:
- The design of localized algorithms, given a desired global behavior, is akin to the process of converting a centralized algorithm to a completely distributed one. The latter requires some non-conventional thinking---the rules that define the distributed computation often bear little resemblance to the original distributed computation. Localized algorithms represent a special kind of distributed computation: one that permits only local interactions. For these, it is even harder to construct the rules that achieve desired global behavior.
- 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[18]. 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.

It will be several years before the research community can satisfactorily address these issues. We propose to make initial inroads in these areas by:

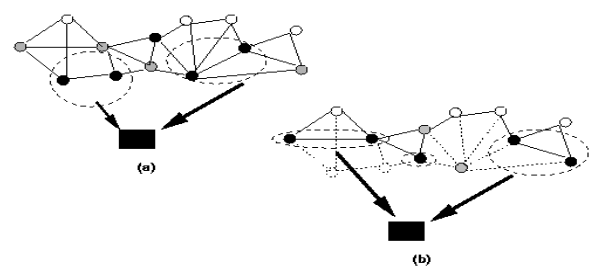
- Designing a representative set of localized algorithms and thereby developing some intuition for the first question posed above.
- Developing performance characterizations for these algorithms as a way of contributing to the second issue.

B.3.2 Adaptive Fidelity Algorithms

In order to develop intuition about the design of localized algorithms, we propose to investigate **adaptive fidelity** algorithms.

Consider a rapidly-deployed sensor network that monitors a remote terrain. If no new sensor nodes are added, this network has a well-defined lifetime; this is the time during which the network can deliver data with acceptable fidelity. Note that, in general, there is a range of acceptable fidelities for a given task. For example, for the triangulation task above, reasonable location accuracy might be obtained with a baseline that is, say 80% of the widest baseline.

An adaptive fidelity algorithm is one that trades off increased network lifetime for possibly reduced data fidelity. Of course, the resulting fidelity must still fall within acceptable bounds. One illustration of this idea is shown in Figure 6. Consider, as before, a sensor network that determines the exact location of sensed objects. Now, instead of every sensor participating in the baseline determination, some sensors turn themselves to conserve power off (the grayed sensors in Figure 6(a)). This has the effect of a smaller baseline and, consequently, lower fidelity triangulation. Subsequently, as some of the currently active sensors "die" (shown with dotted lines), their neighbors "take over" and continue the baseline determination for other objects (Figure 6(b)). If designed correctly, this can result in only slightly degraded baseline determination but a sensor network with nearly double the lifetime.



The adaptive fidelity example described in the previous paragraph doesn't dynamically adapt the sensing fidelity. We can envision other forms of adaptive fidelity:

- One that varies fidelity based on the current sensing task. For example, to pinpoint the location of an intruder, a sensor network might dynamically "turn on" more sensors than those required to pinpoint the location of a small animal.
- Another, more complicated form, tries to extend the lifetime of sensors by reducing fidelity to avoid hotspots. For example, if all objects appear near the same location, the same cluster of sensors will elect themselves for every object. To avoid this, the network might adapt its fidelity to use a smaller baseline (and turn off some sensors) in some cases.

Adaptive fidelity algorithms are essential in rapidly deployed networks for several reasons:

- <u>Cost-effectiveness</u>

In rapidly deployed sensor networks, it will be impossible to determine, *a priori*, what size deployments are necessary. This is because the environment can be completely unpredictable. For this reason (and because we expect sensors themselves to be cheap), sensor network deployments will usually contain larger-than-necessary numbers of sensors. Without adaptive fidelity, these deployments can result in significantly reduced network lifetimes.

- Graceful unplanned incremental deployment

For a similar reason, adaptive fidelity algorithms enable the network to work gracefully in the face of unplanned incremental deployment. Consider a collection of sensors "thrown" into an existing network. We cannot expect that the location of the "thrown" sensors will exactly augment the capacity of that part of the existing network that needs it—that is, many sensors may land near existing sensors that already have significant residual power. Adaptive capacity algorithms help use these sensors carefully to extend the overall lifetime of the network.

We will specify, analyze, and experiment with different forms of adaptive fidelity to understand both the opportunities and limitations of the general approach and to understand what is required to put it into practice in real sensor networks.

B.3.3 Characterizing Global Behavior 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; we intend, in our work, to take the initial steps towards such a methodology.
Localized algorithms achieve a great degree of robustness through local interactions. When deployed in a sensor network, however, localized algorithms may:

- Utilize more resources than necessary in order to achieve a given level of sensing fidelity, or
- May achieve lower sensing fidelity, given a particular resource utilization target.

Intuitively, this may happen because nodes in a localized algorithm inherently "see" only local state; thus, they may be forced into local maxima or minima (resulting in reduced fidelity). To avoid this, the algorithm may attempt to perturb the system from these local maxima or minima, thereby expending more resources. Figure 6 c is a good example of how localized algorithms can tradeoff resource utilization for increased robustness. Recall that this figure describes a situation where certain terrain conditions can cause non-optimal behavior of our simple triangulation localized algorithm. A centralized algorithm would have resulted in only two sensors being elected for triangulation (because this algorithm has a more complete global view).

Localized algorithms can also result in reduced responsiveness to external events. Roughly speaking, this may happen because these local interactions take more time to "diffuse" across the network before the desired global behavior is achieved. Some localized algorithms may be *iterative*; they may converge to the desired global behavior after several steps. On the other hand, the corresponding centralized algorithm might have computed the result in a single step—at the cost of reduced robustness.

A third consequence of localized algorithms is their increased tendency towards pathological failures caused by rippling node failures or system-wide oscillations. Such behaviors are not unknown in highly distributed
systems, like the Internet routing system. Highly localized algorithms (such as the reaction-diffusion algorithms) can also exhibit unstable equilibria. These equilibria can be easily perturbed by random noise in the system. We expect sensor networks to be more prone to these, because they respond to a wider, less predictable, input range---that of their environment.

To understand these tradeoffs, a desirable design tool (analogous to a CAD-tool for VLSI-design) is one that, given any localized algorithm, can quantify its performance---i.e., its reduced fidelity, its increased responsiveness, or its increased susceptibility to pathological failures. It is highly unlikely, given the state of our understanding of these tradeoffs, that we will be able to come up with such a design tool within the lifetime of the proposed project.

Specifically, for the coordination algorithms we choose, our goal is to be able to make statements like:

- "this algorithm uses x% more resources to achieve a tracking fidelity w",
- "this algorithm takes y% longer to converge at a result than a centralized algorithm",
- "the probability of a pathological network failure in our object tracking network is z%".

In some cases, simple system models combined with worst-case analysis will allow us to make statements like these. Fairly detailed simulations that incorporate terrain models will give us more accurate bounds. For the latter, we intend to use the SEEDS simulator, submitted as a separate proposal.

Having done this, what form do we expect our results to take? Recall that localized algorithms consist of local interaction rules. So, based on our investigations, we expect to be able to formulate generalized rules-of-thumb, such as:

- If a local interaction rule is of *this* form, the resulting system has *this* likelihood of succumbing to ripple effects, or
- The locality of the interaction spans a neighborhood of *this* size, this system may need *this* level of resource utilization for *this* sensing fidelity, and so on.

In summary, while we have reason to believe that localized algorithms will offer significant advantages in terms of robustness and scalability, we recognize also that it will be more difficult to characterize the resulting overall system behavior. System designers and users will need to quantify a range of system characteristics such as those discussed below.

- Accuracy: Users will want to know the accuracy with which the sensor network is able to carry out its function. System designers similarly will want to know how much system resource is needed to achieve the required accuracy, i.e., how many sensors, how densely placed, and with what sensor and communication range capabilities.
- Temporal responsiveness: As with accuracy users will want to know the response time of the system to events and system designers will need to determine how much of a resource is needed in order to achieve a particular response time.
- Failure modes: Even as localized algorithms are designed to increase robustness, they bring with them new failure modes. System designers will want to characterize the types of cascading failures that might interrupt network function. Aside from identifying scenarios, designers will want to characterize probabilities and evaluate damping mechanisms.

We will develop intuition for these kinds of performance characterizations by applying it to localized coordination algorithms for object tracking.

B.4 Supporting protocol mechanisms

Finally, we will work on several aspects of the supporting mechanisms that are needed by a wide range of these applications/systems: e.g., Network location Protocol, self-configuring clustering, and security.

B.4.1 Network Location

(Note that this NLP description appears in a companion proposal called SNST. If the SNST proposal is also funded, we can make use of the NLP work conducted under that project. However because NLP is essential to our work described here, we have included it in the event that the SNST work is not funded.)
Location awareness is a fundamental middle-layer service for many sensor network applications. While GPS is suitable in many situations, it does not support all kinds of object tracking: it requires an unobstructed antenna, it is relatively expensive ($100) compared to the price of the objects being tracked, and it has relatively limited accuracy. More importantly, GPS provides only physical location (latitude and longitude) as opposed to logical location (in the office, at home, on the highway, visiting a university). We will therefore design a Network Location Protocol to augment GPS and other commercial tracking systems. Our Network Location Protocols (NLP) will assume a number of mobile objects, some with known locations (due to on-board GPS or a fixed location). Other objects will estimate their location from these first-stratum objects through data communication and direct measurements of timing and communications signal. Thus, network location becomes analogous to time determination with the Network Time Protocol [34].

There are many dimensions of possible network location protocols. Indoor and outdoor communication have substantially different interference and accuracy requirements. A choice must be made between using existing RF- based communications for location and augmenting them with ultrasound sensors. We assume that several base stations are reachable at any time, and will compare different algorithms using this information, from the simple use of the nearest base station, using the mean or weighted mean, or considering signal propagation effects to do multi-lateration [35]. Finally, if specialized hardware is used (possibly sensors or high resolution clocks), what are the trade-offs to placing it at the end system or with in more central “location servers” (possibly co-located with network base stations).

These approaches allow us to determine physical locations: a node’s coordinates in 3-space. End-users don’t care about something at latitude 33.97988N, longitude 118.4399W, altitude 56m, they want to know about the printer in room 1120. We will develop mappings between physical and logical locations, and provide multiple, overlapping, hierarchical coordinate systems suitable for logical location.

Physical constraints are critical in both wireless communication and deep networking. The object tracking application thus will serve as an important case study to verify that our design choices, physical constraints, and models are appropriate for real-world scenarios. Furthermore, indoor location tracking (as required for object tracking) provides substantial challenges (due to multi-path effects and communications blackouts) and provides a pre-existing scheme for logical location (building number and floor).

Preliminary work suggests that network location protocols are feasible. We have investigated several algorithms (nearest, mean, weighted mean, and multilateral) over an outdoor grid of Metricom Ricochet base-stations. Base stations are statically placed about 350-800 km distant from each other. In this environment we compared our estimated position with true position (based on GPS) for each algorithm. From our preliminary studies it appears that the choice of algorithm has substantial effect on accuracy. The best of the relatively simple algorithms considered so far can estimate location within 100 meters with 90% confidence.

This very preliminary result shows what is possible with a naïve software-only implementation. We are currently investigating how minor hardware changes (for example, using a high resolution clock at the base stations) can improve these results while still allowing inexpensive (software-only) mobile nodes.
We expect to use physical and logical network location through the APIs described next.

B.4.2 Clustering in Sensor Networks

We will investigate a self-configuring hierarchical approach to distribute information about the data sensed by nodes in order to efficiently direct interests to the relevant nodes. In this approach sensors automatically organize themselves into an hierarchy. The parent sensors in the hierarchy summarize the data sensed by its children sensors and distribute this summary to farther away sensors. The parent sensors at higher levels in the hierarchy distribute the summary information to larger regions than parent sensors at lower levels. The distributed summary information can then be used to direct interests to the relevant sensors. For simplicity, we use the term cluster to denote the set of a parent sensor and its children sensors.

For example, consider an office environment where sensors monitor the location of various tagged objects such as books and projectors. The sensors organize themselves into an hierarchy and advertise a summary of objects in their clusters. The clusters on the 1st and 2nd floor of a building advertise the presence of books and conferencing equipment on their floors. A query for projector1 from an user on the 2nd floor of the building is then directed to the 1st floor cluster since only this cluster advertised the presence of conferencing equipment. Once the query reaches the 1st floor, it is directed to the relevant smaller clusters i.e., clusters 1 and 2 that advertised the presence of conferencing equipment and thus finally reach the cluster that has projector1.

However, global dissemination of information about events in all the clusters may not be energy efficient especially if queries are largely about just a few events. Hence we intend to adapt our design so that sensors maintain reachability information only to clusters with frequently queried events. The sensors discover reachability to other clusters with less frequently queried events in an on-demand fashion. Although our architecture does not rely on fixed base-stations, we will exploit the presence of any such nodes in our design.

It should be noted that the mobile ad-hoc IP Networking community is concerned with a related problem of discovering routes to destinations in an on-demand fashion. However, the solutions proposed flood queries to initiate a distributed routing procedure to find routes to destinations. Hence these approaches will suffer from scaling limitations when used in sensor networks that consist of thousands of sensors. We have not seen any hierarchical approaches yet in ad-hoc networks. This may be because hierarchical techniques typically require (for scaling purposes) that IP addresses be assigned to hosts such that the reachability of hosts at a particular level in the hierarchy can be summarized by an aggregated address prefix. Dynamic assignment of addresses to hosts to achieve the above structure may be difficult in mobile networks since addresses may have to be frequently changed as hosts move around. Such frequent address changes may cause data packets to be lost or loop in the network.

We next discuss the possible approaches to organize sensors into an hierarchy and the impact of energy constraints on the various aspects of hierarchy construction.

<u>Approaches to construct hierarchy of sensors</u>

Distributed approaches to organize sensors into an hierarchy are better suited to sensor networks since sensors can use rules based on the local environment to construct the hierarchy. Centralized approaches are not robust or energy efficient for sensor networks since collecting topology information at a central server is highly inefficient.

Distributed approaches to construct hierarchies may be either Top-down or Bottom-up.

- Top-down approach: Top-down approaches require a lot of manual configuration or global topology information to be available at nodes in order to produce hierarchies with desired characteristics. Hence these approaches are not well suited to sensor networks.
- Bottom-up approach: In this approach, sensors make decisions based on the local environment to construct the hierarchy. Hence bottom-up approaches are more energy efficient than top-down approaches.

We next discuss some of the choices that need to be made for bottom-up hierarchy construction methods.

- Initial node levels: Sensors at higher levels are responsible for summarizing information from sensors at lower levels. Hence it may be desirable for the initial levels of sensors to be determined based on their energy levels.
- Vicinity discovery: The geographical extent of a sensor's vicinity determines the geographical extent of sensors in a cluster at various levels of the hierarchy. Larger vicinities would result in clusters with sensors that are more widely away from each other. Hence the energy consumed in summarizing information about the cluster at a parent sensor would be excessive. Smaller vicinities on the other hand lead to more levels in the hierarchy thus possibly increases the cost of hierarchy maintenance. We will study how the vicinity sizes affect the overall energy consumption and scaling properties of our design.
- Promotion criteria: The energy expended in summarizing information about a cluster at a sensor may be used as the promotion criteria. It is better to promote sensors that can summarize the information about the cluster in the most energy efficient fashion.
- Demotion criteria: A simple criterion for demoting sensors that do not have any children may be used. Another criterion may be to demote sensors whose clusters respond to relatively fewer queries.
- Parent selection policy: A sensor may choose a parent among the set of potential parents based on the energy expended to reach the parent. We will look at how this policy interacts with the promotion criteria of sensors.
- Algorithm termination conditions: The algorithm can be terminated once a specified number of levels have been constructed. An alternative may be to terminate the algorithm once a single root is elected for the hierarchy. We need to investigate the variation in energy consumed as the number of levels in the hierarchy increases. As the number of levels increases, the cost of hierarchy construction and maintenance may increase but more concise summary information can be distributed throughout the network.
- Adaptation: We need to adapt to various network dynamics such as sensor failures and addition or deletion of sensors. In addition the algorithms need to adapt to the mobility of sensors. We will look at how mobility in terms of the speed, range/extent of movement and frequency of movement of the sensors affects the adaptation algorithm.

B.4.3 Security

Finally, interesting challenges and opportunities arise for securing deeply distributed systems. With so many nodes, some may be subverted (physically or electronically). The large number of nodes requires end-to-end (rather than individual node) security goals. We will explore techniques based on node cross-checking, periodic node shut-down with secure re-initialization, and automatic exclusion of "outliers" from computations. A Byzantine approach to data security is of interest to prevent a few corrupt nodes from substantially altering data aggregation and diffusion.

We can discuss the security issues in sensor networks according to the usual security services: confidentiality, integrity, authentication, and denial of service.

- Confidentiality mechanisms will rely primarily on light-weight encryption and we do not plan to innovate in this area. Rather we hope to benefit from work funded under the nanotechnology area. One interesting opportunity is that with most communication localized, key distribution is somewhat facilitated. On the other hand, with the set of peers so volatile, key distribution is more of a challenge.
- Authentication of nodes and integrity of exchanged data and control is an area that we plan to explore. In the absence of strong cryptographic techniques applied on the data element level, we will explore diagnostic techniques for evaluating the health and integrity of the sensor network. We will explore algorithms to isolate outliers and those nodes that are not behaving as others in the network. In this way

we can exploit the larger numbers of sensors and make it harder to disrupt overall system behavior by taking over one or a small number of nodes.

- Denial of service will also be addressed through background mechanisms that attempt to identify and isolate misbehaving nodes that threaten system operation or availability.

We imagine a very low overhead mechanism that effectively walks the network and causes each node to compare its own state to that of its neighbors for reporting, and for analysis.

B.5 Demonstration applications

During the course of this work we plan to pursue two to three applications. The first will be an indoor application emphasizing scaling and dynamics, object tracking. The second application will emphasize mobility and will be outdoor, but will not achieve the same scaling as object tracking, vehicle to vehicle exchange of road conditions. Finally, we will select a third application that offers opportunity to collaborate with researchers working in a challenging but important domain such as distributed image sensing and identification.

B.5.1 Object Tracking

Our initial test-bed and sample application will be based on tracking and monitoring tagged objects with mobile sensor nodes. We will emulate a system composed of intermittently-mobile, intelligent “sensor nodes” which collect data about the location and state of objects that are identifiable in the region nearby the sensor. The sensor nodes will also communicate with one another and exchange information about observed objects. The sensor nodes will be moved around the sensed environment in a manner that cannot be guided by the goal of monitoring, but rather one that is guided by the otherwise-determined movement of the human beings carrying the sensors. In this sense the tagged objects are the equivalent of vehicles causing seismic vibrations and the sensor nodes in both case, collect such data and disseminate it.
This application will be designed using our diffusion approach and allow us to experiment with various designs for each of the mechanisms: attribute based naming (coordinates in 3-space, object identifier, object-sensed-data sample type); diffusion of information about object location and movement exchanged among the mobile sensor nodes (initial prototypes will run on laptops and PDAs); local aggregation of information; gradients established as a result of queries referring to named data; adaptive fidelity applied to more detailed object tracking in densely populated contexts; and NLP, clustering, and security used by sensor nodes.

Intelligent sensor nodes will be sometimes stationary and sometimes attached to things that move (such as human users). The events being sensed will be the status of objects within the sensing range of the sensor nodes. Status can be as simple as the presence and identity of sensed objects, or as involved as to details of remaining energy, details about the object movement, etc. Sensor nodes will have a limited sensing range and will exchange data with other sensor nodes with whom they communicate. Sensor nodes themselves will operate in power-conserving ways.

This application will force us to deal with dynamics and heterogeneity. The sensor network topology itself will be ad hoc and continuously subject to change as the sensor nodes move around the monitored environment. The change in topology is not just that of a change in the set of neighbors seen by any particular node. In addition, and perhaps more importantly, there will be wide and changing variations in the density of sensor nodes. As the sensor node density varies, the protocol approaches developed in companion work will require that the sensor nodes adapt their behavior to the varying densities. The environment itself will be continuously changing as the monitored objects change position and state. Objects will also have widely varying densities over time.

This application captures key characteristics of an range of envisioned sensor network applications. We will start by developing the basic support for object tracking. We will then build some interesting sample applications of object tracking. For example, we will start with a simple mapping application that will build a map at various granularities of object location. Over time we will develop applications that monitor the more detailed status of objects and generate alerts and perhaps responses to events triggered by significant object-status changes. One can think of this as a distributed condition-based-maintenance infrastructure for the monitored objects.

By exploring condition-based maintenance in the context of our diffusion architecture we will explore the use of aggregation and scaling. In particular, we will build support for dissemination of aggregate condition, as opposed to single device condition. The aggregate condition might be physical or logical. For example, one might want to know the condition of all equipment on the 11th floor when doing facilities surveys, whereas one might want to know the condition of all photocopiers of a particular vendor type when the repair person from that vendor is on sight.

B.5.2 Mobile-Vehicle

In addition to object tracking, we will develop a second application to be determined in consultation with DARPA and after preliminary SCADDS experience (at start plus 6 months).
Data diffusion for communication between moving vehicles is a promising possibility for a second application. There is enormous demand for action in this area: information about traffic congestion and accidents can be passed along, information about changing terrain (bridge out or mud ahead), emergency vehicles can pre-send specific instructions ("clear lane 1 for passage"), and very timely information can be relayed ("car 5 vehicles ahead lost tire---slow immediately!").

Much as with object tracking, traditional techniques such as sending this data to a central server fail to provide the scalability and robustness required for this application. Consider traffic information in the LA area (http://traffic.maxwell.com/la/). An existing web server provides highway traffic speeds, but it is already slow and would collapse under the weight of 10 million simultaneous accesses. Furthermore, it has very limited, coarse coverage with sensors every 2/3rds of a mile for highways---no coverage of other streets.
Vehicle-to-vehicle, diffusion-based communication provides exactly the right solution. Bandwidth and sensor density scales directly with to traffic. Each vehicle provides a traffic speed sensors and local communication relay. Data communication will be localized, making transmission and use of high-resolution information practical. Low-resolution data will be automatically summarized and dispersed over a wide area, allowing long-range planning and analysis. Adaptive fidelity will occur naturally as traffic densities increase.

This application is also well suited to generalize SCADDS research objectives, as well. Basic diffusion algorithms apply to this domain, where data direction, aggregation, reinforcement and queries are key. However, the different environment will force us to generalize these principles. Communication dynamics are stressed as cars pass or follow each other. Outdoor radio communication and location determination have different constraints, as well. We expect linear communication (up and down highways) to offer new opportunities for optimization. We will discover and optimize these paths automatically through gradients while remaining adaptive to changes (due to road detours, for example).

The scale of vehicle-to-vehicle communication limits full-scale experimentation. We will explore this application in a testbed of six vehicles, each with WaveLAN local communication, some with Metricom wide-area communication, some with GPS. We will use this testbed to produce high-fidelity models of the application to allow validated simulations to a much larger scale. (If possible this work will use the SEEDS simulation platform.) Upon successful demonstration of SCADDS technology for this application, we will seek to expand this testbed (in Y3) and obtain matching funds from industry.

There are currently many efforts for a “networked car”, but most focus on communication within the car rather than between cars. Some use wide-area information, but they are based on Internet/server-based technology and so don’t scale. Some automated traffic systems use vehicle-based sensors, but they fail to use vehicle-to-vehicle communication. Other automated systems depend on in-road sensors; these are expensive and slow to deploy. The success of SCADDS approaches to this domain could be quantified in each of these aspects. Direct communication between vehicles will reduce aggregate message travel distance and adaptive fidelity will provide information at a much finer resolution.

B.5.3 Distributed Imaging

The area of distributed imaging is in itself a complex problem, demanding significant expertise in image processing and vision. We do not propose to become experts in these areas per se. Rather we hope to collaborate with researchers in those areas and explore the applicability of our coordination techniques to coordinating object identification and tracking in a highly distributed network of cameras. The details of this application will be worked out during the second year of our work and it is possible that we might change the particular application focus to another domain if we find an alternative area to offer better opportunities and impact.

C. Deliverables (2)

- Software implementation of:
- Diffusion based coordination primitives
- Adaptive fidelity algorithms for selected applications
- Instrumentation to allow evaluation of sensor network algorithms

- Evaluation methods and tools to quantify global behavior of localized algorithms.

- Software implementation of the following sensor network application support mechanisms:
- Network Location
- Clustering
- Security

- Software implementation of application prototypes including but not limited to:
- Distributed object tracking and monitoring
- Vehicle application

- Simulation studies of the coordination methods, algorithms and protocols, using SEEDS or whichever simulation environment will be used for SCADDS development.

- Papers and on-line presentations describing analysis, simulation and implementation of coordination techniques including directed diffusion, localized algorithms like adaptive fidelity, network location protocols and other issues addressing problems for scalable coordination of deeply distributed systems.

- Regular project summaries and status reports

- Participation in Internet working groups and related DARPA sponsored groups as requested.

D. Statement of Work (3)

- Develop algorithms for directed diffusion including attribute naming, diffusion of interest signals in the neighboring nodes, gradient building and data-flow along interest gradients. Distribute software implementation of these algorithms.

- Develop algorithms for selective fidelity. Analyze, experiment and evaluate with different forms of adaptive fidelity. Distribute software implementation of these algorithms.

- Design network location protocols supporting logical and physical location determination. . Distribute software implementation of these algorithms.

- Develop test scenarios to evaluate the algorithms and protocol performances and simulate the scenarios.

- Implement, demonstrate, and evaluate three applications based on the above algorithms.

- Prepare papers covering above issues pertaining to the application of object tracking specifically and deeply distributed networking in general.

E. Milestones (1)

6 months
- Version 0 implementation of diffusion primitive software
- Version 0 of architecture document describing details of object tracking system/application
- Design documents describing:
  - NLP
  - Clustering
  - Security

12 months
- Version 1 implementation of diffusion primitive (software) with limited object tracking application software running over it.
- Version 0 of adaptive fidelity mechanism
- Version 0 implementation of NLP, clustering and security

18 Months

- Release of initial diffusion primitive software to SENSIT community.
- Architecture document describing 2nd and 3rd applications

24 months

- Demonstration of object tracking application with adaptive fidelity and release of associated software.
- Experimental, analytical and simulation results evaluating diffusion architecture, NLP, clustering and security
- Version 0 of second application

30 months

- Demonstration of vehicle and third application

36 months

- Release of object tracking, diffusion software
- Additional analytical and simulation results and conclusions regarding applicability and suitability of the diffusion architecture.

F. Technology Transfer (2)

Our research group has a long history of extensive collaboration with other researchers, members of the DARPA community, and commercial industry. We plan to continue this collaborative approach in this new area of work. We believe that collaboration in the definition, execution, and experimentation phases is critical.

- Presentation at DARPA PI meetings
- Presentations at workshops and conferences
- Interaction with potential vendors of sensor and tag technology.

G. Comparison to other work (3)

Hardware design

The Low Power Wireless Integrated Micro-sensors (LWIM) [1,2] project, the Ultra Low Power Wireless Sensor project [3] and the Wearable Computing project[4] among others, are involved in building devices that have the ability to perform sensing, signal processing and other capabilities in a single module. Micro-electro-mechanical Systems, or MEMS, are integrated micro devices or systems combining electrical and mechanical components. MEMS devices can range in size from micrometers to millimeters. Reference [5] provides some information on the state of MEMS technology and its potential applications.

Some applications of embedded networks

Various process control and automation tasks in factories have traditionally used networks of embedded systems known as control networks [6] to efficiently perform their monitoring tasks. The control network architecture typically consists of a centralized controller coordinating the communication between sensors and actuators connected to the controller. Many control networks are currently wired networks where the sensors are powered by a common power-line. A transition to wireless communication is expected due to the ease of deployment and maintenance of wireless networks.

Tags or transponders are simple circuits that respond to interrogations from a reader by returning some pre-programmed information. These devices find a wide range of applications in object monitoring scenarios such as anti- shoplifting and inventory tracking systems. Reference[7] provides information about the available tag technologies.

Pottie et al.[1] give a high-level systems point of view on the design of Wireless Integrated Network Sensors (WINS). They discuss the necessary hardware technology, signal processing techniques and some details of a self-organizing network architecture in the context of a military situational awareness problem. The networking aspects discussed include TDMA scheduling, automatic discovery of neighbors by nodes to form a connected topology and cooperative detection by sensors.

The Piconet[8] project at the Olivetti and Oracle Research Laboratory (ORL) 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. A low level radio protocol is presented for communication between various embedded objects that use Piconet. Some interesting applications for Piconet in the home and office environments are also discussed.

Active badges [9] are small computing devices worn by personnel for location tracking purposes. Each badge has a globally unique code that is broadcast using an infrared interface. The signal is picked up by infrared receivers placed at various points in the building. In the above fashion, the badge's owner can be tracked as the person moves around in the building. ParcTab[10], equipment tag[11] also perform similar functions. The ORL location system [12] improves on the performance of infra-red based technologies by using ultrasonic sound.

The MIT smart room project [13] and the Forest of sensors project [14] analyze data from video images to make inferences about the presence or absence of various objects or people. PinPoint Corporation has recently developed a Local Positioning System (LPS)[15] to track various assets in home or office. They use baseband radio communication between tags attached to the various objects and a set of controllers placed at various points in the building.

Weiser[16] considers a vision of ubiquitous computing where the various computing elements are seamlessly integrated into the environment. Many types of devices will be available around us that will

each be capable of adapting their behavior based on the conditions. These devices will however be so tightly integrated with the environment that they will be invisible to common awareness.
Stajano and Jones [17] present a system where users can query and control their personalised computing resources and services from their cell-phones. Their emphasis was to not require an user to carry an additional gadget to access some basic services. Instead, they focus on instrumenting the cell phone that many people carry to be able to access some basic computing resources such as checking email or stock quotes.

Relevant techniques in other areas

Biological systems may provide some interesting insights into the design of coordination algorithms for sensor networks. These include reaction-diffusion models for morphogenesis[18] and models based on behavior of ant colonies[19]. Morphogenesis is concerned with the mechanism by which the initially homogenous human cells eventually differentiate themselves to form various tissues and organs. The ant colony algorithms 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.
Distributed robotics may also be an area of research that has some relevance to our design. These include coordination protocols being designed for troops of low-cost robots to explore and acquire maps of unknown environments[20]. 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.

Coore et al.[21] 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. There is also no analysis of the state requirement for nodes in the hierarchy.
Some relevant techniques can be found in web caching such as the self-configuring Adaptive web caching[22] system. In this system, 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.

The PATH[23] project deals with developing coordination and communication protocols to enable auto-motives to more efficiently utilize freeways. A vision based surveillance system for roads that uses state-of- the-art image processing and tracking algorithms is also being developed. Other issues being investigated include automated highway systems, magnetic guidance systems for cars and vehicle navigation systems.
Many problems in distributed computing [24] concern themselves with executing some actions when the state of the system satisfies a particular condition. Hence, the ability to construct a global state and evaluate a predicate over such a state forms the core of solutions to many problems in distributed computing. These solutions are relevant to the design of coordination protocols for embedded networks as well. Distributed programming methods like CORBA[25] may also be relevant to our study.

Ad-hoc networks

Ad-hoc networks refer to self-organizing networks of mobile wireless nodes that do not depend on any fixed infrastructure. Ad-hoc routing protocols can be classified as reactive or proactive. 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. Reactive routing protocols on the other hand start a route computation process only when a packet needs to be sent to some other node.

Proactive routing proposals for ad-hoc networks include distance-vector[26 , 27] and link-state algorithms[28]. These proposals do not apply well to sensor networks as continuous computation of routes to frequently mobile destinations is not energy efficient.

Many of the current reactive routing protocols do not apply well to sensor networks because they flood route requests throughout the network. These proposals[29] include the Temporally Ordered Routing Algorithm (TORA), Dynamic Source Routing (DSR) and Ad-hoc On-demand Distance Vector (AODV) protocols.

The Zone Routing Protocol (ZRP)[30] and Cluster Based Routing Protocol (CBRP)[31] proactively maintain routing information for a local neighborhood while they reactively acquire routes for farther away destinations. These protocols expect to use lesser average control traffic compared to pure reactive or proactive approaches.

Core Extraction Distributed Ad hoc Routing (CEDAR)[32] establishes a virtual backbone consisting of “core”nodes. These “core” nodes are responsible for performing route computations to destinations. This scheme may not apply well to networks of energy constrained nodes since the core nodes on the backbone will be drained of energy much faster than others.

The On-Demand Multicast Routing Protocol (ODMRP)[33] uses a mesh-based rather than a conventional tree-based scheme for multicast communications in ad hoc networks. This approach improves robustness at the expense of redundant packets in the network.

H. List of Personnel (3)

Key Personnel:

Deborah Estrin	Year1: 30%	Year2: 30%	Year3: 30%
Ramesh Govindan	Year1: 30%	Year2: 50%	Year3: 50%
John Heidemann	Year1: 30%	Year2: 50%	Year3: 50%

Deborah Estrin, Ph.D.---Principal Investigator.
Deborah Estrin is a Professor of Computer Science at the University of Southern California in Los Angeles where she joined the faculty in 1986. Estrin received her Ph.D. (1985) and M.S.(1982) from the Massachusetts Institute of Technology and her B.S. (1980) from U.C. Berkeley. In 1987, Estrin received the National Science Foundation, Presidential Young Investigator Award for her research in network interconnection and security. Estrin is a co-PI on the DARPA Virtual Internet Testbed (VINT) project and the NSF Routing Arbiter project at USC's Information Sciences Institute where she spends much of her time supervising doctoral student research. While she continues her research related to protocol scaling and multicast, most recently she has begun to focus on problems related to networking and coordination among very large numbers of physically-embedded devices (sensors, actuators).
Estrin is an active member of the IETF multicast routing related working groups and a long-time member of the End- to-End research group. Estrin is a member of the ACM, IEEE,and AAAS. She has served on numerous panels for The National Science Foundation, The National Academy of Engineering, and DARPA's Information Systems and Technology (ISAT) advisory board. Estrin has served as an editor for the ACM/IEEE Transaction on Networks and the Wiley Journal of Internetworking Research and Experience, and as a member of the program committee for many IEEE Infocom and ACM Sigcomm conferences, and she was program co-chair of ACM Sigcomm '96.

Deborah Estrin is currently supported (on an annual average) at 25% effort on the VINT project at ISI, 5% effort on the NSF RA project at ISI, and 15% effort on the STRESS project at USC. She is involved in several pending proposals with DARPA, including SEEDS 20% (with John Heidemann and Kevin Fall), SCADDS (PI at 45% effort with Ramesh Govindan and John Heidemann), SNST (co-PI at 10% time with Allison Mankin and John Heidemann), all in the area of deeply embedded networking.

Ramesh Govinday, Ph.D.--- Principal Investigator.
Ramesh Govindan is a Project Leader of ISI's NSF-sponsored Routing Arbiter project and the DARPA-funded SCAN project. He is also a Research Assistant Professor of Computer Science at the University of Southern California. Dr. Govindan received his Ph.D (1992) and M.S. (1989) in Computer Science from the University of California at Berkeley, and his B.Tech. (1987) in Computer Science and Engineering from the Indian Institute of Technology at Madras, India. He also worked for two years at Bell Communications Research, Morristown, NJ. While at Bellcore, he was co-PI on the DARPA funded Pip project, and participated actively in the IPng standardization efforts within the
IETF. Dr. Govindan is also the author of the Route Server, now in production use at several Internet exchange points.

Ramesh Govindan is currently supported 50% on the NSF Routing Arbiter project, and 50% on the DARPA-sponsored SCAN project. The Routing Arbiter project is expected to complete by June 1999. He is involved in two pending proposals with DARPA: SNST (5% of his time) and SCADDS (co-PI at 50% effort with Deborah Estrin and John Heidemann.

John Heidemann, Ph.D.--- Principal Investigator.
John Heidemann is a research assistant professor at USC/ISI where he has worked since 1995. He has been conducting ARPA-sponsored operating systems and network research since 1989. His stackable filing

interface is deployed in the 4.4BSD operating system. His networking research has included studies of HTTP and TCP interactions, TCP optimizations, and currently network simulation issues with the VINT project. As a part of VINT he has focused on simulation abstraction to improve scaling and extension of the simulator to new domains. He his a member of ACM, IEEE, and Usenix, is an active reviewer for IEEE/ACM ToN, SIGCOMM, NSF, and is currently co-chair of the USENIX '99 invited talks track.

John Heidemann is currently supported at 100% effort on the VINT project at ISI. He is involved in several pending proposals with DARPA, including SEEDS (PI at 75% effort with Deborah Estrin and Kevin Fall), SCADDS (co-PI at 25% effort with Deborah Estrin and Ramesh Govindan), SNST (co-PI at 25% time with Allison Mankin and Deborah Estrin), all in the area of deeply embedded networking.

I. Facilities (1)

ISI maintains a large University research computing center. ISI has built a reputation for excellence and efficiency in both experimental computer services and production services. ISI originally developed, provided, and now helps to support, many mature software packages for the entire Internet community. It contains a variety of computers, from mainframes to microcomputers. ISI is connected to several wide-area networks, including the Internet, SCAN, CAIRN, and Los Nettos.
ISI is the home of the MOSIS project, which provides the academic VLSI research community with low-cost, rapid- turnaround fabrication of VLSI chips and printed circuit boards. Through the FAST project, ISI also maintains an automated broker for standard electronics parts ordering.
ISI is also home to the Integrated Systems Laboratory (ISL), which has demonstrated advanced packaging and electronics system engineering capabilities. This laboratory has established a local technology base for advanced packaging, which has provided characterization and models for new technologies as well as procedures and sources for assembly and testing of prototype and production systems.

Pursuant to FAR, Subpart 9.5, the University of Southern California/Information Sciences Institute does not support any DARPA offices and therefore has no Organizational Conflict of Interest.

J. Cost breakdown (5)

Personnel

- Drs. Estrin, Govindan, and Heidemann will collaborate and co-supervise the research.

- The two programmers will be responsible for implementation, one at the system level and the second at the application level.

- Each of the four GRAs will likely specialize in one of the following areas: Network location protocol, Clustering algorithms, Security, Diffusion algorithms.

Equipment

- In our experiments, the “sensors” will be the librettos and PDAs. In order to experiment with and demonstrate credible examples of diffusion we need a number close to 100 sensors. 15 librettos with some form of wireless card (total sensor node cost of $1000 each), and 60 CE machines ($400 each); combined with laptops that we already have will allow us to achieve experiments of up to 100 sensor nodes. We will stagger the purchase of these sensor nodes over the first 18 to 24 months.
- In addition we will need development platforms (workstations, laptops) for the programmers and GRAs.

We will deploy the sensor nodes across the networking group, Div7, at ISI. Large numbers of the staff and students will participate in the experiments. In addition we will attach these sensor nodes to certain particularly important and challenging applications first.

- We will also purchase a collection of tags to place on many objects in the environment so as to have a dynamic set of events at adequate scale.

Note that if a companion proposal, SNST, is funded, we will reduce the amount of equipment purchased under SCADDS.

Travel

We will travel to visit other researchers in the field, to deliver papers at conferences, and to attend DARPA PI meetings.

III. Bibliography

References
[1] G. Pottie, W. Kaiser, L. Clare and H. Marcy. Wireless Integrated Network Sensors. Submitted 1998.
[2] The WINS project. http://www.janet.ucla.edu/lpe.lwim/.
[3] The Ultra Low Power Wireless Sensors project. http://www-mtl.mit.edu/~jimg/project_top.html.
[4] Wearable Computing. http://www.cs.cmu.edu/ wearable/.
[5] MEMS Technology Applications Center. http://mems.mcnc.org.
[6] Echelon, The LonWorks Company. LonWorks Solutions. http://www.echelon.com/Solutions/.
[7] Transponder News. http://rapidttp.com/transponder.
[8] F. Bennett, D. Clarke, J. Evans, A. Hopper, A. Jones and D. Leask. Piconet: Embedded Mobile Networking. IEEE Personal Communications, 4(5), October 1997.
[9] R. Want et al. The Active Badge Location System. ACM Transactions on Information Systems, January 1992.
[10] N. Adams. An Infrared Network for Mobile Computers. In Proc. USENIX Mobile and Location-Independent Comp. Symp., Cambridge, MA, August 1993.
[11] A. Harter and A. Hopper. A Distributed Location System for the Active Office. IEEE Network, Special Issue on Distributed Systems for Telecommunications, January 1994.
[12] A. Ward, A. Jones and A. Hopper. A New Location Technique for the Active Office. IEEE Personal Communications, 4(5), October 1997.
[13] A. Pentland. Machine Understanding of Human Action. In Proceedings of 7th International Forum Frontier of Telecommunication Tech., November 1995.
[14] W. Grimson, C. Stauffer, R. Romano and L. Lee.Using adaptive tracking to classify and monitor activities in a site. CVPR, 1998.
[15] J. Werb and C. Lanzl. Designing a positioning system for finding things and people indoors. IEEE Spectrum, September 1998.
[16] M. Weiser. The Computer for the 21st Century. Scientific American, September 1991.
[17] F. Stajano and A. Jones. The Thinnest of Clients: Controlling it all via cellphone. ACM Mobile Computing and Communications Review, October 1998.
[18] 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.
[19] 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.
[20] M. Lopez-Sanchez, R. Lopez de Mantaras, and C. Sierra. Incremental map generation by low cost robots based on possibility/necessity grids. In Proc. of the Conf. on Uncertainty in Artif. Intell., pages 351-357, Providence, RI, 1997.
[21] 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.
[22] 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/.
[23] The PATH project. http://www.path.berkeley.edu/.
[24] S. Mullender, Editor. Distributed Systems. Addison-Wesley, 1993.
[25] Object Management Group. The Common Object Request Broker: Architecture and Specification. Tech. Rep. 91.9.1, December 1991.

[26] J.J. Garcia-Luna-Aceves. Loop-Free Routing using Diffusing Computations. IEEE/ACM Transactions on Networking, February 1993.
[27] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proc. of the ACM SIGCOMM, October 1994.
[28] 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.
[29] 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.
[30] 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.
[31] 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.
[32] R. Sivakumar, P. Sinha and V. Bharghavan. Core Extraction Distributed Ad hoc Routing Protocol (CEDAR) Specification. Internet Draft, draft-ietf-manet-cedar-spec-00.txt, October 1998. Work in progress.
[33] M. Gerla, G. Pei, S. Lee and C. Chiang. On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc Networks. Internet Draft, draft-ietf-manet-odmrp-00.txt, November 1998. Work in progress.
[34] D.L. Mills, Network Time Protocol(Version 2) Specification
and Implementation. Network Working Group RFC-1119.
[35] A.Ward, A.Jones, A. Hopper A new location technique for the Active Office IEEE Personal Communications, Vol.4, No. 5, October 1997, pp.42-47