Attribute Hierarchies as Adaptive Overlays for Energy Efficient Inquiry Forwarding in Sensor Networks*

T.D.C. Little, P. Basu, W. Ke and S. Abu Ayyash
Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts
{*tdcl,pbasu,ke,saayyash*}*@bu.edu*

MCL Technical Report No. 04-01-2003

Abstract– New hardware technology allows many types of sensors to be interconnected in wired or wireless networks. This interconnectivity enables a multitude of novel modes of data collection, analysis, and system control. However, under typical deployment methodologies, sensor networks can be difficult to re-task once in the field. We argue that it is impossible to predict all potential uses of a sensor net in advance of deployment, and propose an enabling approach that defines classes of solutions from which instances can be cast using network overlays.

We investigate and develop a technique to dynamically re-task pre-existing sensor networks with new missions. With this approach a sensor net can be optimized based on mission goals and cost constraints including bandwidth, power and reliability. The proposed solution allows construction of overlays based on data-flow and aggregation required by the scientific inquiry presented to a sensor network. We outline algorithms to build overlays with desired attributes and to deal with changing mission goals and failures under varying resource constraints. We also show by analytical cost comparison that our framework has significant bandwidth gains over a flooding-based scheme (directed diffusion) when the rate of inquiries is high and the probability of inquiries are skewed to specific regions.

Specific outcomes of our research include: (1) definition of mission specification including representation, programmatic issues, representation of dataflow, database queries, and other scientific inquiry; (2) optimization under different goals of energy consumption, bandwidth, reliability, latency; (3) overlay construction, management and reliability.

*This work was supported by the NSF under grant No. ANI-0073843. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

Large-scale deployment of wireless sensors and actuators in the physical environment is poised to radically transform the world that we know. Already a typical home or automobile has $10^1 - 10^2$ embedded computational elements, sensors, and actuators. In the future we will see sensors for physical phenomena such as temperature, pressure, humidity, light, sound, motion embedded in even higher densities in physical locations such as buildings, automobiles, forests, oceans, battlefields, and the atmosphere. These sensors will be deeply networked by means of wireless communication and will form increasingly a massively distributed sensor resource that can be envisioned to support a variety of queries about local or remote physical phenomena accessed by users in different parts of the world.

Users of such sensor networks will be diverse – scientists, factory workers, vehicle drivers, museum curators, students, investigators etc. With access to the right sensor network, a user can perform a plethora of tasks such as monitoring natural or artificial phenomena, tracking objects over time, generating reports on behaviors, or even regulation of traffic flow in busy city streets. Examples include sensing the presence of vacant parking spots in a busy city [2], tracking of wild animals in forests or that of marine life in the oceans, or currents for $El Niño$. A group of heterogeneous sensors with different sensing modalities can also perform more complex sensing tasks in a collaborative manner. One example of such a complex task is forming contour maps of physical phenomena in a flat area [10].

Because most sensors in a sensor network are intended to monitor a phenomena and report results elsewhere, they can be collectively modeled as a large spatially distributed database with the results of the sensed phenomena being the data stored in the database [3]. Thus the most fundamental primitive for addressing the sensor network database is a *query* posed by a user using the attributes of sensors. Under a relational model, this includes primitives such as selection, projection, and join. However, there are additional models for use and addressing sensors under this model. These include support of local control loops (interconnecting sensors to actuators), data aggregation as represented by data flow processes, and embedded functions. We therefore use the term *inquiry* to define the general set of ways of tasking a sensor net for the purpose of yielding a result. This term is chosen to represent the general concept of scientific inquiry potentially posed to a sensor net. Important here is that we believe that most scientific investigators who use sensor data want access to raw data in support of their hypotheses, however, there is great advantage to localized adaptations of data aggregation and processing. A sensor network inquiry model should have some basic components which include attributes of devices needed for the sensing task, constraints on the measured data, constraints on the quality of the response and support for embedded functions of aggregation, transformation, collaborative tasks, and nesting of inquiries. These embedded functions can be applied “in network “ on sensed data, near the sensors. This has been recognized as an important theme in many current research projects [16, 21, 20] and can either be achieved by executing functions that a sensor understands or by shipping the functions along with the inquiry [4]. A few examples of inquiries that might be posed include:

Inquiry 1 (*Relational database query*) A scientist inquires: how many nests in Cypress trees in the Northeast section of the Amazon forest currently have birds in them?

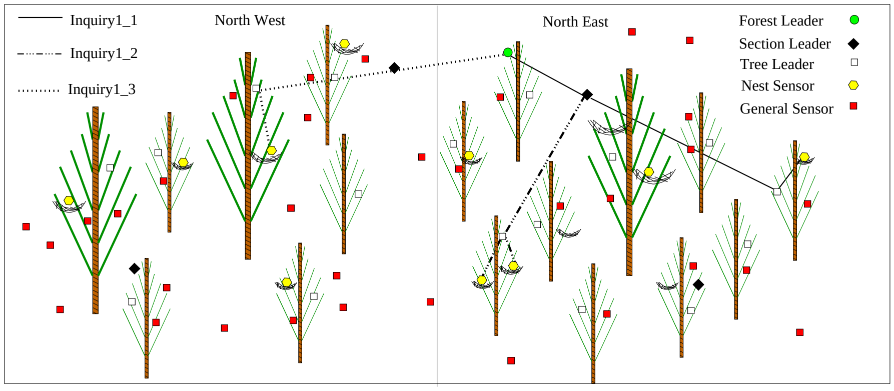


Figure 1: Inquiry Forwarding using Multiple Overlays on the Forest example Inquiry 1: Sensors reside on trees, nests and on the ground. They have been tagged with the values of their corresponding attributes: `[forest]`, `[section]`, `[tree]`, and `[nest]` which form a spatial containment hierarchy. The instances of overlays depicted here exploit these containment relationships. For the first two instances of the inquiry (Query1_1 and Query1_2), the instantiation of the overlay is such that they share the same "`section` leader" node.

Inquiry 2 (*Embedded function*) A physical plant employee inquires: give me average temperature readings in the laboratories in the basement of Building 10 in which there is nobody inside and the energy usage is greater than 30 KWh.

Inquiry 3 (*Dataflow processing task*) A traffic officer tasks a sensor network to monitor vehicular traffic during a special sporting event (e.g., the Boston Marathon), to locate congestion, and to control actuators (traffic lights) to relieve the congestion. This example can be formulated using a data flow model represented by component tasks.

Inquiry 4 (*Grid-based sensing*) Motion sensors are dispersed across a battlefield, deployed via artillery, for the purpose of tracking soldiers. A tactician can pose the following inquiry: are there any enemy soldiers in a quadrant that has an approximate GPS location of $< LON, LAT >$? Are there any friendly soldiers there too?

In each of these examples our intent is to develop the framework in which these types of inquiries are easily rendered, while benefiting from the distribution of the inquiry into the sensor network. In contrast, related efforts report techniques that usually rely on data flooding to achieve response to and from sensor nodes [12, 16, 3]. We argue that flooding of large networks with inquiries in an uncontrolled fashion can lead to rapid depletion of resources and can also quickly lead to network congestion. The latter problem is acute especially when multiple users flood the network repeatedly with diverse inquiries, for example, using directed diffusion [12]. This can severely limit the network's capacity for supporting multiple inquiries.

Distributed Locally Managed Sensor Catalogs In large dynamic systems, it is not feasible (energy/ bandwidth efficient) to perform global optimization based on global knowledge of sensor state. Throughout this paper, we describe local (often greedy) techniques of achieving the general goals of reduced overhead, energy and bandwidth. These optimizations need only rely on locally collected information. Therefore, managing and maintaining distributed knowledge bases of information is of great importance. We create a distributed database built out of components that are locally managed. As in [3] we refer to these components as *catalogs*. Catalog information aids in processing the inquiry in an efficient manner based on the cataloged state and attributes of a group (cluster) of sensors.

Hierarchical Organization with Overlays As sensor networks become more ubiquitous, a mechanism is required for incorporating new and potentially complex inquiries while also ensuring their successful execution. We propose the use of a hierarchical structure primarily based on *containment*, although more generalizable based on other assigned attributes. These attributes allow the construction of relationships that facilitate flexible implementation of inquiries. Once these relationships are established, network overlays can be defined that represent the logical organization that is relevant to an inquiry while also permitting achievement of optimization goals. One of these key goals is to minimize energy usage over the lifetime of the sensor network, in other words, to increase its lifetime.

In the context of our work, an overlay describes a logical connectivity structure between a few key sensor nodes in the sensor network which facilitate scalable forwarding of a user inquiry to relevant sensor nodes. The logical connectivity reflects containment relationships between spatial sensor node attributes related to inquiries. This enables a hierarchical approach to clustering spatially proximate sensors based on the attribute values at each level of the containment relationship. For each cluster of sensors, a node will be elected, called the *leader*, to represent a logical node in this overlay. The process of mapping of this hierarchical logical structure of the overlay to physical leader sensor nodes is referred to as *instantiation* of the overlay. Fig. 1 illustrates overlay instantiation for instances of type of Inquiry 1.

Our overlay instantiation techniques seek to re-use and strengthen already existing overlay structures for executing future inquiries containing a similar set of attributes (e.g., add more branches to the overlay tree). An overlay may exist prior to an inquiry and its instantiation may be relevant to multiple inquiries over a period of time. On the other hand, a new overlay may need to be produced and instantiated on demand based on new attributes required by an inquiry that have not been addressed before.

Adaptation to Inquiry Traffic We propose that the principal means to achieve the aforementioned goal of minimizing resource utilization is through adaptation based on the frequency of use of different types of inquiries. For example, if most inquiries are of the form "give me all temperature samples less than 0^oC from forest 1016," then directed diffusion [12] appears to be the most relevant data access and routing method because all temperature sensors must be queried. However, as the sensor network evolves over time, if more refined inquiries such as Inquiry 1 become frequent, then there is benefit from the creation of a hierarchical overlay structure for more bandwidth and energy efficient routing of inquiries. Sec. 4 gives preliminary cost analysis under these varying conditions. Our vision is to design protocols that adapt the amount of hierarchical structure present in the sensor network according to the inquiry traffic to satisfy the key aforementioned goals, that is, as a given query becomes more frequently used, the structure is optimized for repeated use.

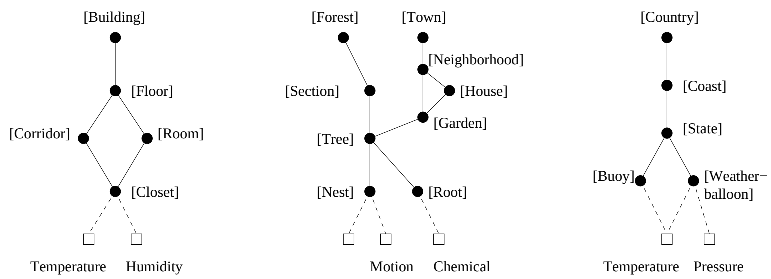


Figure 2: Examples of Containment DAG (C-DAG) Relationships

Re-tasking Sensor Networks Our framework also seeks to achieve adaptation in a sensor network via the *re-tasking* of sensor functions and users missions. For example, we might re-task a set of sensors to perform a new sensing task that cannot be performed by existing individual sensors and that has not been performed before. For example, processing of Inquiry 2 assumes that motion, temperature and card access sensors can collaboratively estimate whether there is a person inside a room with a high probability. This function can be embedded into the sensor network on-demand.

Leading sensor network proposals [12, 16, 3, 9] do not adequately address the aforementioned issues. Although the principles of hierarchical organization have been utilized in the works [8, 11, 19], our techniques are more general and flexible. Additional related work is considered in Sec. 5.

This paper is organized as follows: in Sec. 2 we discuss our framework which is comprised of the attribute relationships which drive the building of the overlay structures (Sec. 2.1), a new attribute based clustering algorithm presented (Sec. 2.2), and mechanisms for overlay instantiation and routing (Sec. 2.3). In Sec. 3 we present different methodologies for re-tasking a sensor network. In Sec. 4 we present our initial results for cost effectiveness of utilizing overlays instead of flooding. Related work is discussed in Sec. 5 and finally, future work and conclusions are presented in Sec. 6.

2 Adaptive Framework for the Construction of Overlays in Sensor Networks

An inquiry defines the mission associated with how sensors are tasked. Because there are attributes associated with sensors and the inquiry, one can steer a request for sensor data in a manner that avoids unnecessary flooding as an inquiry dissemination technique. We propose a more efficient technique using overlays in this section.

2.1 Attribute Containment Relationships and Structure of an Overlay

We exploit the multi-level virtual containment relationships between sensor nodes' spatial attributes to define the hierarchy of an overlay. This containment relationship will be general enough initially so that it can address a multitude of inquiries posed to the sensor network. For example, the following succession of attributes describes a spatial containment relationship that is understood by the network: a `nest` is in a `section` of a `forest` or notationally, `nest` ⊂ `section` ⊂ `forest`. Each of these attributes determine a level in the overlay needed to process inquiries related to them. Overlays are constructed in a top-down manner using the structure of this containment relationship which in turn is stored in the form of a directed acyclic graph (DAG).

A few examples of containment DAGs (C-DAGs) are shown in Fig. 2. Although only a simple tree structure might be relevant per inquiry, we propose to store a more encompassing DAG structure that can be pruned for various paths to form a tree relevant to an inquiry. We will also consider that a C-DAG can be used in its entire form based on inquiry content. This implies an attribute containment scheme that is more general than, for example, the Intentional Naming Service [1]. Sensors are tagged based on an initial C-DAG that best describes the most general containment relationship with respect to an initial positioning of a sensor and an assumed *a priori* distribution function of general incoming inquiry types.

Why a Structured Scheme? Existing schemes that are based on pure flooding of a request (such as [12]) may execute [Inquiry 1] by setting the attributes of diffusion to [`forest=F`, `section=North-East`, `nest=occupied?`] in the inquiry. However, we argue that since sensors are distributed over large networks which can be dense, some degree of organization and structure will be greatly beneficial toward bandwidth saving. In particular, for inquiries that inherently possess a hierarchy of relationships, it is feasible to exploit the relationships to perform multi-level inquiry processing based on the spatial containment hierarchy, rather than go directly to the leaves. For [Inquiry 1], the inquiry will travel successively to one or more leader nodes (defined in detail in Sec. 2.2) representing the *forest*, then leader nodes representing `section = North-East`, then leader nodes representing all the `trees` in this `section`, and then to the sensors on `nests` on the chosen trees (Fig. 1). In Sec. 4 we study the trade-offs between the costs incurred in terms of energy/bandwidth usage in maintenance of overlays and energy saved for different types or patterns of inquiries.

2.2 Hierarchical Clustering based on Attribute Containment Relationships

Having identified the different containment relationships and tagged the sensors with their attribute values, we can group them into clusters of same-attribute sensors and elect a leader. This leader will gather information of all the sensors in the cluster (i.e., compile the catalog) and become their "representative." With possession of the catalog information, the leader becomes the point of control for managing efficient passing or blocking of inquiries to its cluster members depending on whether they are relevant to a given inquiry. When an inquiry reaches inside a cluster, it remains within the cluster (i.e., sensors from neighboring clusters do not propagate the inquiry). With this approach, it is essential that sensors be aware of their attribute values, which explains the importance of tagging, as mentioned previously.

A hierarchical clustering system following the containment relationships achieves additional scalability and allows us finer control over the selection of sensors that will receive an inquiry. Unicast routes are established between sensor nodes that are leaders at different C-DAG levels within

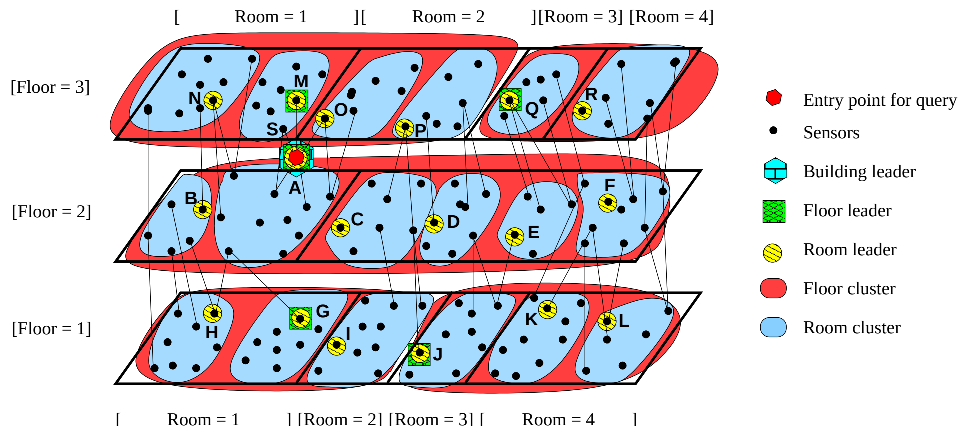


Figure 3: Attribute Containment based Clustering. Cluster formation starts from A, which elects itself as building, floor, and room leader. When it broadcasts the cluster formation packet, M accepts A's building leadership, but notices that the packet came from a different floor and room, and elects itself as leader of its floor and its room. Upon M's broadcast, O accepts M's floor leadership, but keeps its own room leadership candidacy and eventually becomes room leader. S accepts leadership from A and M, canceling any candidacy timers it may have. As cluster formation packet propagates, new room clusters are formed if the rooms are large (e.g., rooms 1 and 2 on floor 3) but since different floor clusters cannot be formed in the same room, there is only one floor cluster on floor 2. On floor 1, G and H both broadcast their floor candidacy close to one another, but G is the "most suitable" leader because we used the "lowest" id function as tie breaker. H remained room leader because of the hop distance between itself and G. The building cluster encompassing all sensors has not been shown for the sake of clarity.

a hierarchy[1], and catalog summaries are sent bottom-up. Thus a top-level leader may very quickly eliminate inquiry propagation to large clusters of irrelevant sensors (with respect to the inquiry), saving energy and bandwidth.

Because leaders have the increased burden of inquiry processing and route maintenance, a rotation scheme can be used to achieve fairness with respect to load distribution and energy consumption. Alternately, specific nodes optimized for this purpose can be selected.

We developed clustering algorithms designed to address the issues mentioned above. The algorithms form same-attribute clusters with one leader and rotate leaders among cluster members. Cluster sizes are constrained to be within designed limits whenever possible. Devices with higher energy levels are selected in the rotation process. Unicast routes are naturally established among adjacent level leaders in the process. In addition the algorithms detect and recover from leader failures and support dynamic membership updates, effectively allowing dynamic C-DAG level updates (i.e., the containment relationships may adapt to the types of inquiries). An example of how clustering works can be found at Fig. 3, and some inquiry forwarding issues within the hierarchical clustering systems is illustrated in Fig. 4. For a fuller specification of the clustering algorithms, please refer to [13].

[1] We use terms like "high/low level leaders" at many places in this paper. "High" corresponds to a low value of the $level$ attribute which is 1 for the top level of the C-DAG hierarchy.

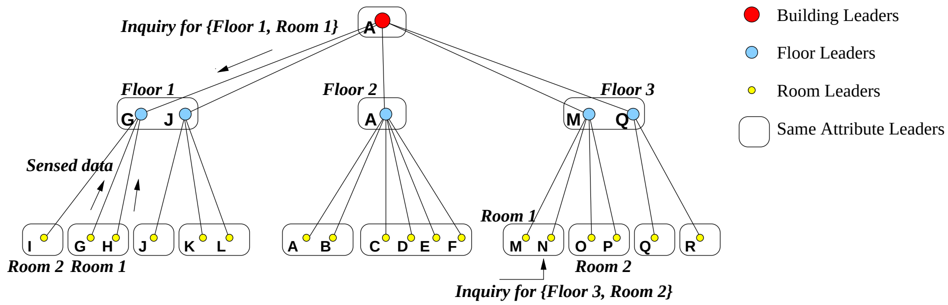


Figure 4: Inquiry Forwarding in Overlay Trees. The figure represents the *leaders* in the overlay instance corresponding to Fig. 3. Suppose an inquiry from A to {`floor=1, room=1`} must be processed through an *embedded function*. Since all clusters in {`floor=1, room=1`} belong to G (leader in Floor 1), G can apply the embedded function on sensed data. This would not be possible if H belonged to J (at Floor 1) – we would need to reach A before applying the embedded function. Thus early aggregation is facilitated. If an inquiry is received by N regarding {`floor=3, room=2`}, and it reaches M (at `floor=3`), M can redirect the inquiry *directly* to O and P. If M and Q were both floor leaders for Room 2, the inquiry would have to be forwarded to A. Note that because inquiries do not cross cluster boundaries, an inquiry that reaches O will not be forwarded to P and vice-versa.

2.3 Instantiation of an Overlay

As described in Sec. 2.1, an overlay is a subset of a C-DAG containment relationship with respect to an inquiry. In order to facilitate scalable routing of inquiries, an overlay must be instantiated or mapped to physical sensors (leaders of clusters at different lower levels of the C-DAG hierarchy which satisfy the inquiry constraints). With the goal of saving energy and reducing the scope of uncontrolled floods over large sensor networks, we propose a greedy instantiation approach. At any instant of time, the chosen leaders may not be the best possible choice for nodes in the overlay. This is because of periodic leader rotation within their respective clusters over the lifetime of an overlay (see Sec. 2.2). If an overlay exists in the network to forward inquiries of a certain type, a fresh inquiry with a similar set of attributes can be forwarded in a staged manner through the overlay (from one leader to the next) until it reaches the intended sensor nodes. However, if no overlay exists when the inquiry is issued, one has to be instantiated reactively.

On-demand instantiation of an overlay can be achieved by performing directed diffusion in stages. Normally, leaders should exist in every cluster even if they have not received inquiries for a long time.[2] In this situation, a level k leader diffuses the inquiry after setting $\texttt{leader}_{k+1} = \texttt{all}$ so that all existing leaders at level $k+1$ in the C-DAG respond to the inquiry by either forwarding it to or instantiating lower level leaders. We expect that on-demand instantiation to be infrequent when the system is in steady state. The cost of flooding due to this action should be amortized over the lifetime of the sensor network when the overlay instance gets reused by subsequent inquiries.

Routing between Levels of an Overlay As stated earlier, in a large-scale sensor network, a diffusion style routing algorithm used in an uncontrolled manner for every inquiry can deplete node resources,

[2]Although their rotation periods will be very high since there is no fairness concern in the lack of traffic (see Sec. 2.2).

cause network congestion and therefore escalate the rate of node failures. However directed diffusion has significant benefits for smaller networks as it does not require route establishment or maintenance in a resource intensive manner. In diffusion, network-wide *gradients* are setup as a result of periodic floods in the sensor network and every sensor node can route to the sink using these gradients. We propose to combine the benefits yielded by the above scheme and periodic local broadcasts that happen in clusters at each level of the instantiated overlay. This approach is very effective since we do not have to specifically discover unicast routes on-demand unless there are node failures. Interestingly, unicast routes between leaders at adjacent tiers are obtained as a *side effect* of the cluster leader rotation process outlined in Sec. 2.2. We illustrate this fact in Fig. 5. Detailed explanations are skipped due to space limitations.

Tolerance to Node Failures Our approach may suffer from problems if an intermediate node x that forwards unicast packets between leader sensors at levels k and $k+1$ fail. In that case the older unicast route will be lost. However, nodes in the neighborhood of x (child and parent in the routing tree) will have multiple routes to either leader in a dense network, as much as multiple gradients formed between source and sink in directed diffusion. Hence route repair can happen locally in case of node failures. In the future, we hope to investigate the effects of various single and burst failure models and node density on this local recovery scheme.

Routing in Tier-0 Clusters Communication inside a lowest level cluster is achieved via pure attribute based diffusion like mechanisms (as depicted in Fig. 5). This diffusion is cluster-ID based which means that any messages routed outside the cluster will be dropped immediately, and therefore the scope of this diffusion is limited to the clusters. We argue that nodes within a cluster rarely communicate over multiple hops with each other as most of the communication is with respect to the leader of the cluster, hence unicast routing between any two generic nodes in the sensor network is not a requirement.

Issues in Route Maintenance Unicast routes produced by the above mechanism generally do not need proactive maintenance. Any inquiry traffic along such a route reinforces its lifetime. On the other hand, if a particular route is not used for a long time, it will be purged from each node's cache. However there are a few trade-offs involved in route maintenance at higher levels. Since higher level C-DAGs tend to span a much larger area than the lower levels, the maximum bandwidth savings happen because of containing the flooding at the higher levels of the overlay. Therefore, purging of unicast routes from higher level leaders should be done conservatively. But, we recognize that unicast routes between higher level leaders in an overlay instance are likely to be long; also, infrequent leader rotations at the higher levels of the overlay causes infrequent catalog exchanges and that increases the probability of a unicast route failing significantly. To solve this problem, we propose to use periodic but extremely lightweight (zero byte payload) route reinforcement messages from an upper level leader to those at the lower adjacent level. Note that this happens only during the lifetime of an overlay instance. In other words, reinforcement messages are not sent if that route has not been used by any inquiry in the recent past. In the future trade-offs due to the effects of varying different parameters in our routing scheme (e.g., node failure probabilities, route timeout values) need to be studied.

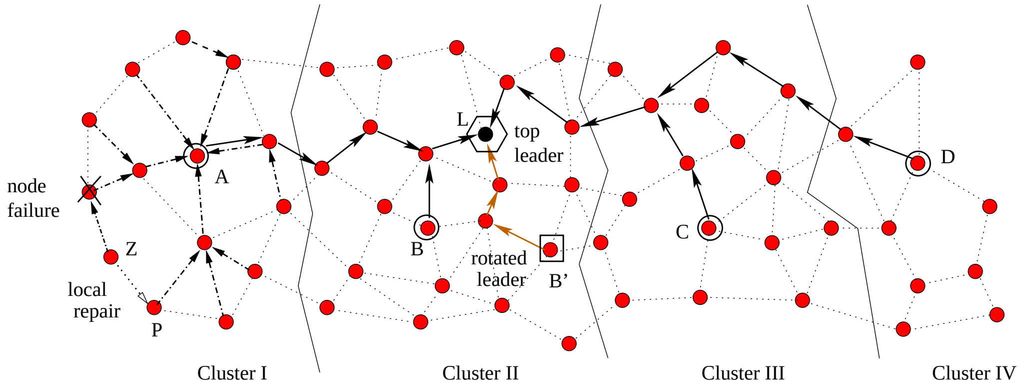


Figure 5: Creation and Maintenance of Unicast Routes between Tier Leaders: A, B, C and D are cluster leaders at the second tier in the overlay instance, and they are at the root of a routing tree that spans all sensors in their clusters (paths shown for cluster I only). The routing trees were formed through an intra-cluster flooding (at that level in the hierarchy) at the time they became leaders. Thus A, B, C and D are also able to keep unicast routes to their top tier leader L. When leadership in cluster II rotates from B to B', the latter does not need to discover a route to L because it already had one since the time L became top leader. The route from L to B' is established when the latter contacts the former with catalog update information. If a sensor's path to its leader becomes disrupted due to an intermediate node failure, local repair will be attempted (Z contacts its neighbor P), since all sensors in a cluster have a path to the corresponding cluster leader.

Reliability in Inquiry/Data Delivery Since we are proposing the use of multihop unicast messages between cluster leaders at adjacent levels of the overlay instance, reliability of message forwarding is a concern. We advocate the use of leader-to-leader acknowledgment (ack) for important unicast inquiries or data. If an ack does not arrive before the expiration of the timer, the message is resent. An intermediate node which detects a link failure will attempt to discover an alternate route for forwarding the message by *scoped* diffusion in its neighborhood (it will propagate an interest regarding a route to the destination). However if the hop count to the destination is low, it may be possible to just diffuse the data so that it reaches the destination without any explicit discovery of routes. We will investigate these trade-offs in detail during the research.

Adaptation to Frequency of Inquiries We emphasize that every leader of a cluster at each level can make local decisions based on the frequency of inquiries regarding maintenance of discovered routes to a subset of lower level leaders. A leader can weigh the resources (energy) spent during the initial route discovery with the cost penalty from broken routes if those are not purged in a timely fashion to determine an adaptive time period for purging stale unicast routes. In the future, we will investigate such trade-offs for various types of inquiries.

Execution of an Inquiry There are different levels of complexity in executing an inquiry. Consider the case in which values for each attribute are specified in the inquiry, e.g., one wants temperature sensor values from all `rooms` on `floors = 3, 4 and 5` in `building = 10`. In this case the leaders of `floors` 3, 4 and 5 will possess knowledge of the `room` leaders' properties (have temperature sensor?) in their catalogs and therefore, only those `floor` leaders that can satisfy the inquiry will forward the query to respective `room` leaders and `building = 10` leaders will forward the inquiry to all relevant `floor` leaders. In case the inquiry contains no constraints on the value of any sensor attribute, it must be forwarded via unicast to all leader sensors at all levels of the overlay. Responses are aggregated by the same leaders and then transmitted back via the unicast paths existing between leaders.

Optimizations geared towards future gains in bandwidth utilization/energy are indeed possible after an inquiry is executed. For example, if an inquiry specifies `room = 25` without specifying the value of the intermediate attribute `floor`, the initial cost of building an overlay instance can be amortized by learning the value of `floor` attribute and using it for similar inquiries in future. This cached information has a lifetime that can be renewed every time it is reused.

The functions of aggregation, interpretation and transformation of sensed data utilize the same overlay instance on the reverse path. Some leaders will be tagged with a temporary attribute, such as `aggregator` with respect to a class of inquiries that have been processed in the past. How to automatically choose these leaders is dependent upon the specific inquiries and the complexity of the embedded functions involved. In the future, we will explore mechanisms to enable this process by extracting local information from the inquiry and the distributed catalogs.

3 Mechanisms of Re-tasking a Sensor Network

The generality of our approach allows us to work with any kind of sensor data and a large set of inquiry types. The concept of re-tasking of a sensor network must be enabled such that we can accomplish this general framework. Central to re-tasking a sensor network is the fact that we use adaptive overlays which can be re-used to support the new tasks while minimizing resource usage and even time. Within these overlays we describe the following mechanisms that allow us to achieve a flexibility in re-tasking.

Re-tasking via Embedded Functions We envision a framework for supporting re-tasking of a sensor network by using the concept of embedded functions. For example, a temperature sensor network, using an embedded function can re-interpret its own temperature reading to indicate whether a living object is in its vicinity or not. Specifically, embedded functions allow inquiries to be posed in terms of attributes that are seemingly unsupported by the sensor network. Consider another example: suppose an inquiry requires the computation of altitude of certain weather balloons in a specified geographical area (say, US east coast). Since the attribute *altitude* is not directly supported in any of the sensor nodes in the weather balloons, the user can explicitly specify in the inquiry an embedded function $ComputeAlt : Temperature \times Pressure \rightarrow Altitude$ which can calculate altitude from temperature and pressure samples in weather balloons. Embedded functions can be distributed along with inquiries and stored in the sensor network at key locations and this new knowledge (how to compute altitude), in addition to the new attribute *altitude* will become part of the new capabilities of the sensor network, thus supporting future querying on that attribute directly. We believe that this re-tasking aspect of our work is novel in this form and is very powerful as it gives researchers and scientists (or any competent user) power to use the sensed data for alternate purposes that were not envisioned at the time of deployment, and this can be achieved without retrieving all the sensed data back to the point of inquiry.

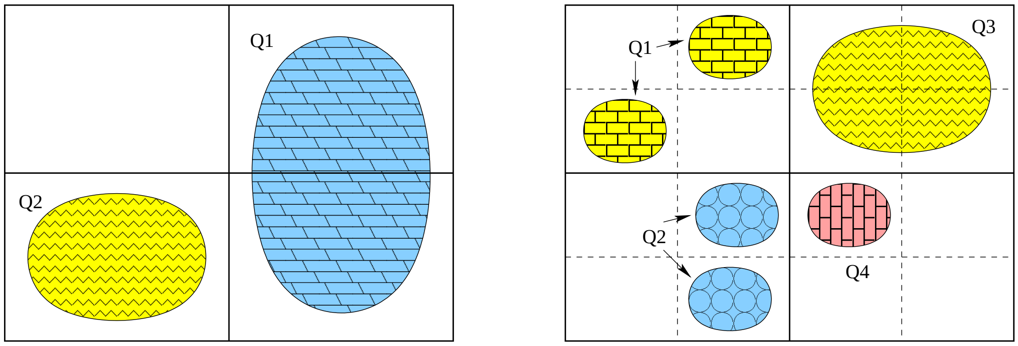
Re-tasking via Tag Change Sensors are tagged based on an initial containment relationship attribute, which can then be updated dynamically over time by joining several C-DAG lower levels into a single higher one or splitting of a higher-level C-DAG into several lower-level ones. For example, if the attribute `tree` is added when sensors are installed in trees (in addition to those *in* the nests). The new relationship can be described as: `nest` $\subset$ `tree` $\subset$ `section` $\subset$ `forest` (e.g., [Inquiry 1]). This C-DAG relationship has to be updated in all key locations of the sensor network. A global update of any new C-DAG relationship will be flooded to all leaders at all levels using the most encompassing C-DAG relationship. If a leader receives a request from a higher-level with a C-DAG attribute that it is not aware of, then it must propagate to higher levels the fact that it missed the update. We assume that C-DAG updates (see Sec. 2.2) are infrequent and will not generate frequent broadcast traffic in the sensor network.

Re-tasking via Dynamically Assigned Attributes Designating nodes as *aggregators*, or in general, as sensors where an embedded function $f(.)$ is applied can be seen as an association of an attribute to the corresponding sensors. The lifetime of this attribute might be that of an inquiry or even of an overlay supporting multiple similar inquiries. Note that implementation of arbitrary embedded functions requires some limited mobile code execution support from the computing environment on at least some sensor nodes. Arguably, this suffers from the security problems that plague mobile code based systems. We advocate that a sandboxed execution environment with strict protection rules can mitigate the security threats and enable execution of user specified embedded functions on sensor inputs [4]. However the creation of such a secure environment is beyond the core focus of this work.

4 Comparative Cost Analysis of Overlays and Flooding Techniques

In this section we present an analysis to establish the effectiveness of creating and maintaining overlays over the lifetime of a sensor network as compared to a flooding-based scheme. We focus on the communication cost for the dissemination of inquiries since power consumption in a sensor node is dominated by radio communication [15]. In both flooding- and overlay-based schemes, the return data path is built during inquiry propagation. Sensed data are sent back along such paths, which form an inverted tree structure: sensors with data are leaves, and the node which first started the inquiry is the root. The exact number of transmissions needed to send the collected data back is dependent on the tree structure, and is left for future research. However, since the underlying mechanism is the same (tree structures), we believe that the order of magnitude of the number the transmissions in both cases is similar. We note that while in diffusion [12], there is a cost in populating the multiple variable-rate return paths, in our case, there is a cost in maintaining the unicast routes and insuring reliability as described in earlier sections. These costs are likely to be comparable, depending on the level of reliability yielded by each scheme.

In our model, inquiries arrive into a sensor network at a mean rate λ in a large time epoch T; the network comprises of N sensors distributed uniformly over a square area. The cost incurred by



(a) 2 Levels in the Containment DAG

(b) 3 Levels in the Containment DAG

Figure 6: Querying in a Attribute Partitioned Space: Each pattern corresponds to a type of query.

directed diffusion in terms of bandwidth consumption for dissemination of the inquiry is given by:

$$Cost_{diff} = \lambda T N \tag{1}$$

In this analysis we consider the case where the entire space is divided into four regions with each region having the same value for a certain attribute in the C-DAG hierarchy, and this subdivision is continued within each smaller region as the number of levels in the C-DAG hierarchy increases. We assume that each subdivision results in four new regions. Fig. 6 illustrates two different cases of subdivision that we study in this analysis.

With our overlay structure, first the inquiry is forwarded from the point of entry (e.g., a base station) to the top level ($level = 1$) leader. If the inquiry is for the whole network, the latter floods it, otherwise it forwards the inquiry to appropriate leaders at $level = 2$ (with appropriate region attribute). These will likewise determine whether the inquiry is for their whole region, in which case they flood the region, or forward the inquiry to appropriate sub-region leaders (these will then flood their sub-region, and so on). The cost of flooding the network is N, while that of a region with $level = 2$ is $N/4$ and a sub-region with $level = 3$, $N/16$ etc. Unicasts from the base station to the top level leader have estimated cost bounded by $\sqrt{2N}$ since there are as many hops in the longest path in the square area. Likewise, the cost bound for forwarding the inquiry from a level 2 leader to a level 3 leader is $\frac{\sqrt{2N}}{2}$.

Fixed Cost in Clustering Attribute based clustering has an associated "fixed cost" for the entire epoch due to the periodic rotation of clusterheads. Suppose the clusterhead rotation period at $level = i$ is T_i for $i = 1$ to ℓ_{max}. The total fixed cost is then given by:

$$Cost_{overlay}^{(fixed)} = N + 2N \sum_{i=1}^{\ell_{max}} \frac{T}{T_i} + \sqrt{2N} \sum_{i=2}^{\ell_{max}} 2^i \frac{T}{T_i} \tag{2}$$

Initial clustering involves one network-wide broadcast that contributes N (first term in Eq. 2) to the cost since each node transmits a broadcast packet only once. The rest of the terms correspond to cluster maintenance costs. There are $\frac{T}{T_i}$ clusterhead rotations at $level = i$. Each rotation at $level = i$ requires one broadcast at that level followed by all sensors in the cluster responding to update the catalog information. The broadcast contributes N to the cost at each level and so does the catalog update step. This accounts for the second term in Eq. 2. The third term corresponds to the unicast cost of communication of catalogs between cluster leaders, and is a simplification of $4\sqrt{2N}\frac{T}{T_1} + 16\frac{\sqrt{2N}}{2}\frac{T}{T_2} + \cdots$.

Cost of Inquiry Dissemination Now, consider a model where one particular region (say, R) at $level = \ell_{max}$ receives an inquiry with probability p. For example, the region getting inquiry Q2 in Fig. 6(a) or Q4 in Fig. 6(b). For simplicity, we assume that inquiries involving the rest of the possible combinations are equiprobable with probability q, e.g., $q = \frac{1-p}{14}$ for $\ell_{max} = 2$. In this model, the average cost incurred for dissemination of inquiries over time T is given by:

$$Cost_{overlay}^{(inq)} = \lambda T\{\sqrt{2N} + \sum_{Q \in S} P_Q C_Q\} \quad (3)$$

In Eq. 3 the estimated cost of forwarding an inquiry from the base station to the top level leader is upper bounded by $\sqrt{2N}$. This analysis assumes the presence of one leader per attribute-value region. The second term expresses the cost of disseminating the inquiry to its intended destinations while using the constructed overlay. The summation occurs over all elements Q in the set S of all possible combinations of sub-regions in the sensor network. In general there are $s = 4^{\ell_{max}-1}$ sub-regions and hence $|S| = 2^s - 1$. P_Q is the probability of an inquiry involving the particular combination of sub-regions Q from the set S and C_Q is the cost of disseminating that particular style of inquiry. If Q spans all sub-regions in the network ($level = 1$), then $C_Q = N$; if it only spans $m < 4$ sub-regions at $level = 2$, then $C_Q = m(\sqrt{2N} + \frac{N}{4})$. If Q involves m sub-regions $r_1, r_2, \ldots, r_m$ at $level = 2$ and also involves specific subregions inside each of these r_k's at $level = 3$ (say, $\{r_{11}, \ldots, r_{1n_1}; r_{21}, \ldots, r_{2n_2}; \cdots; r_{m1}, \ldots, r_{mn_m}\}$, then the cost is given by:

$$C_Q = \sum_{k=1}^{m}\{\sqrt{2N} + n_k(\frac{\sqrt{2N}}{2} + \frac{N}{16})\} \quad (4)$$

The C_Q term for a general level $i \leq \ell_{max}$ can be expressed similarly as a sum of costs due to unicast and scoped broadcast within attribute sub-regions as have been illustrated above (not presented here). For $\ell_{max} = 2$ the total average cost incurred for dissemination of inquiries for the epoch T is given by:

$$\begin{aligned} Cost_{overlay}^{(inq)} &= \lambda T\{p(\sqrt{2N} + \frac{N}{4}) + \frac{3}{14}(1-p)(\sqrt{2N} + \frac{N}{4}) + \frac{6}{14}(1-p)(2\sqrt{2N} + \frac{N}{2}) \\ &\quad + \frac{4}{14}(1-p)(3\sqrt{2N} + \frac{3N}{4}) + (1-p)N + \sqrt{2N}\} \end{aligned} \quad (5)$$

The total communication cost corresponding to our overlay-based scheme is given by:

$$Cost_{overlay} = Cost_{overlay}^{(fixed)} + Cost_{overlay}^{(inq)} \quad (6)$$

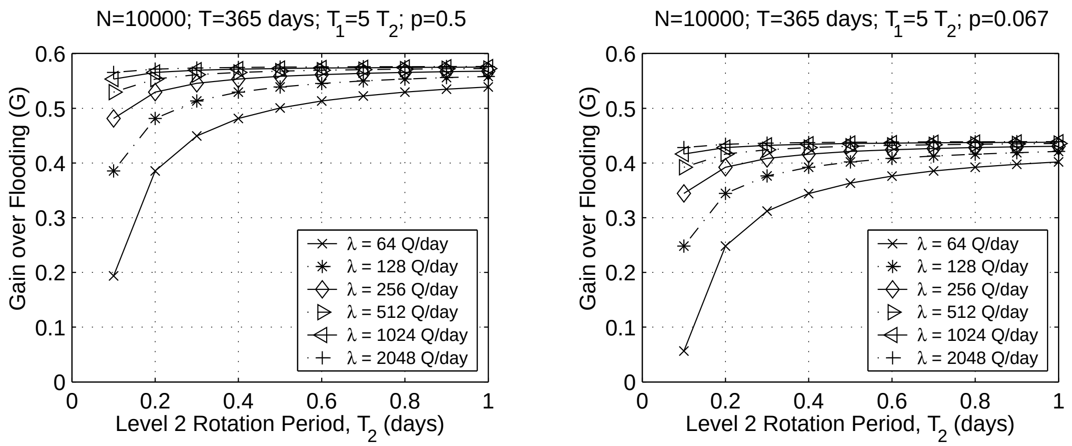


Figure 7: Effect of Rate of Inquiry and Clusterhead Rotation Period on Gains: 2 levels in C-DAG

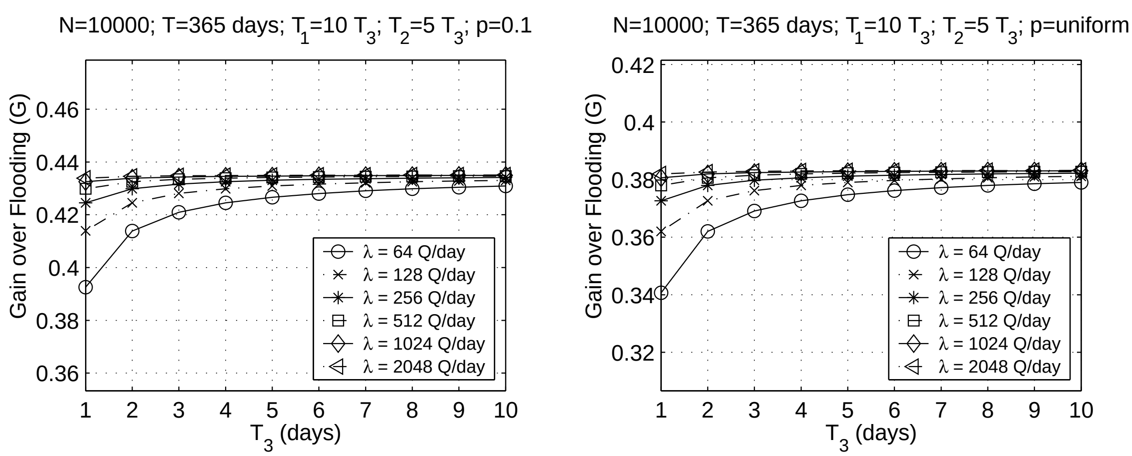


Figure 8: Effect of Rate of Inquiry and Clusterhead Rotation Period on Gains: 3 levels in C-DAG

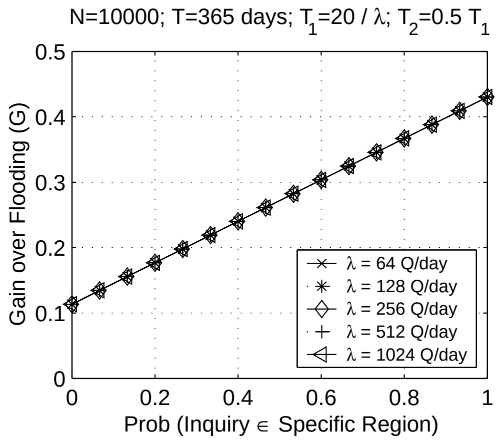


Figure 9: Gain vs. probability for proportional rotation periods : 2 levels in C-DAG

We define our performance index, G, by:

$$G = \frac{Cost_{diff} - Cost_{overlay}}{Cost_{diff}} = \frac{Cost_{diff} - Cost^{(fixed)}_{overlay} - Cost^{(inq)}_{overlay}}{Cost_{diff}} \tag{7}$$

Current sensor technology such as Mica motes have a lifetime in the range of approximately 6 months using AA batteries and a duty cycle of 2% (between active and sleep modes) [15]. The lifetime and energy efficiency of such sensors are likely to increase in the near future. In this analysis we assume an operating life of one year. In general, overlays tend to outperform flooding-based schemes for larger time epochs due to amortization of the clustering cost.

First we study the case in which inquiries for one sub-region are extremely popular ($p = 0.5$). Results for this case are shown in Fig. 7(a). We see that as λ increases, the dependence of G over the the rotation periods T_1, T_2 diminishes. This is expected as T_1, T_2 influence the fixed cost due to attribute based clustering – as more inquiries arrive into the sensor network, the fixed cost penalty almost vanishes. In Fig. 7(b) we study the case in which all 15 combinations of regions are equiprobable ($p = \frac{1}{15}$). We see similar behavior except that the gains are slightly lower in this situation. This is also expected because more possible destinations for the inquiries correspond to greater unicast costs in its dissemination. Similar results have been shown for the case of 3 C-DAG levels in Fig. 8.

One interesting phenomena that can be observed from these curves is that the gains stabilize after λ is increased past a certain value for every value of p. This is because for high λ the contribution of $Cost^{(fixed)}_{overlay}$ towards G is minimal after a certain threshold even for frequent rotation periods. The dominant contributor to the cost is thus $\frac{Cost^{(inq)}_{overlay}}{\lambda TN}$ which is primarily linear in p for large N. For this reason we observe different asymptotic values of G as p is varied.

If rotation periods T_i's are made inversely proportional to the mean arrival rates, e.g., $T_i = \frac{a_i}{\lambda}$, then Eq. 7 becomes:

$$G = 1 - (\frac{1}{\lambda T} + 2\sum_{i=1}^{\ell_{max}} \frac{1}{a_i} + \sqrt{\frac{2}{N}} \sum_{i=2}^{\ell_{max}} \frac{2^i}{a_i}) - (\sqrt{\frac{2}{N}} + \frac{1}{N} \sum_{Q \in S} P_Q C_Q) \tag{8}$$

In this case gain G essentially becomes independent of λ and linearly increases with probability p. This can be seen in Fig. 9.

We note that in our architecture the cluster leaders perform greater computation and communication tasks than other sensor nodes. Hence their resources are likely to get depleted sooner. A fair leader rotation policy would warrant lower rotation periods (values of T_i's lower than the ones shown here) to allow all sensors to participate as leaders during T, and that could be detrimental to the gains of overlay construction. Also, frequent leader rotation results in higher network traffic and therefore faster depletion of resources at sensors. Since the sensor network is large and dense, there are likely to be many new candidates for assuming the role of a leader after an old leader dies due to resource depletion. We advocate the policy of keeping a reasonable value for T_i (i.e., not too small) while letting the adaptive re-clustering algorithm (Sec. 2.2) choose leaders with maximum remnant energy. In this manner, the performance gains will be preserved without depleting resources at all sensor nodes. However, T_i has to be small enough in order to detect failures and network partitions. We found that fairness considerations can be balanced with cost savings by adjusting T_i's at different levels. We intend to investigate these trade-offs in more detail in future.

In this section, we analytically demonstrated that overlays yield gains over flooding-based schemes when there are sub-regions in the sensor network that are more targeted than others, i.e., when the distribution of inquiries is *not* uniformly distributed over time and space. We also showed that with increase in inquiry rate λ, overlays perform better since their structures can be re-used and are more directed towards specific target regions, whereas in a flooding-based scheme, a network-wide broadcast is necessary for each inquiry (e.g., interest propagation in Directed Diffusion).

5 Related Work

Recently several research projects have begun to treat a sensor network as a distributed database. Cornell University's COUGAR System [3] is a platform for testing query processing techniques over ad-hoc sensor networks. UC Berkeley's TinyDB project [17] is another query processing system for extracting information from a network of sensors. One of their focuses is power efficient algorithms for ”in network” aggregation [16, 10].

There have been several initiatives toward the development of adaptive delivery protocols for sensor data. Information Sciences Institute's (ISI) Directed Diffusion [12] is a scheme that floods an *interest* throughout the sensor network and the sensors satisfying the request form a gradient toward the source of query; sensor data flows through these gradients. MIT's SPIN family of protocols [9, 8] use data negotiation (with named metadata) and resource-adaptive algorithms to efficiently disseminate information in a wireless sensor network while conserving valuable communication and energy resources.

Clustering has been a very important topic for research in the mobile ad hoc networks context toward the goal of achieving scalability in routing [5, 7, 18]. Other power efficient hierarchical clustering schemes for sensor networks can be found in [6], [8] and more recently [19, 14]. Other hierarchical schemes include SINA [19] which is also attribute based and has naming and location awareness as salient features. TTDD [14] is a two-tier data dissemination scheme for large sensor networks with efficient forwarding of queries from mobile sinks to the source. MIT's Intentional Naming System (INS) [1] uses *nametrees* to represent resources with hierarchical attribute structures.

Although we rely and build on some of the aforementioned schemes, our approach is distinct in the following manner. We perform hierarchical clustering based on the containment relationships between sensor location attributes. This allows us to achieve scalable routing of queries on named attributes using the created overlay. Another salient difference is that none of these approaches share our vision of adaptive maintenance of structure for scalability as a function of the query traffic and popularities. In fact, only recently have researchers started to recognize the scalability problems faced by the current systems when multiple queries are issued to a sensor network [20]. We believe that our work can address this issue by means of query dependent creation, adaptation, and reuse of a hierarchical overlay structure in the sensor network. In this way, the system can be simple and efficient like directed diffusion for certain types of queries and more scalable for more directed queries.

6 Conclusions and Future Work

In this paper we presented techniques for the efficient execution of user queries (or inquiries) submitted to a large heterogeneous sensor network. This is achieved by creating efficient overlays on top of an existing sensor network and then executing the inquiries by utilizing such overlays. The creation and maintenance of the overlays is facilitated by hierarchical clustering algorithms that utilize the containment relationships between spatial attributes of sensor nodes and that facilitate our routing schemes. With regard to inquiry dissemination cost, we have shown considerable gains in using our schemes (over 50% in some cases) as compared to a pure flooding.

Future work will consist of extensive simulations of the proposed algorithms to further establish the feasibility of this approach and to better understand its design trade-offs and optimizations. Other relevant and interesting challenges of interest to us include guaranteeing better unicast messaging reliability and the adaptation of our clustering algorithms to the rate of inquiries and popularities of target regions, among others.

References

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. "The Design and Implementation of an Intentional Naming System". In *Proc. ACM Symposium on Operating Systems Principles (SOSP)*, Kiawah Island, SC, December 1999.

[2] P. Basu and T. D. C. Little. “Networked Parking Spaces: Architecture and Applications”. In *Proceedings of the 56th IEEE Vehicular Technology Conference - Fall*, Vancouver, Canada, September 2002.

[3] P. Bonnet, J. E. Gehrke, and P. Seshadri. “Querying the Physical World”. *IEEE Personal Communications*, 7(5):10–15, October 2000. Special Issue on Smart Spaces and Environments.

[4] A. Boulis and M. B. Srivastava. “A Framework for Efficient and Programmable Sensor Networks”. In *Proc. OPENARCH*, New York, NY, June 2002.

[5] A. Ephremides, J. E. Wieselthier, and D. J. Baker. “A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling”. *Proceedings of the IEEE*, 75(1):56–73, January 1987.

[6] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. “Next Century Challenges: Scalable Coordination in Sensor Networks”. In *Proc. International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, WA, August 1999.

[7] M. Gerla and J. T.-C. Tsai. “Multicluster, Mobile Multimedia Radio Network”. *Wireless Networks 1*, pages 255–265, 1995.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. “Energy-Efficient Communication Protocols for Wireless Microsensor Networks”. In *Hawaaian International Conference on Systems Science (HICSS)*, Hawaii, January 2000.

[9] W. Heinzelman, J. Kulik, and H. Balakrishnan. “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks”. In *Proc. International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, WA, August 1999.

[10] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek. “Beyond Average: Towards Sophisticated Sensing with Queries”. In *Proc. International Workshop on Information Processing in Sensor Networks (IPSN)*, Palo Alto, CA, March 2003. to appear.

[11] T. Imielinski and S. Goel. “DataSpace - Querying and Monitoring Deeply Networked Collections in Physical Space”. *IEEE Personal Communications Magazine*, October 2000. Special Issue on ”Networking the Physical World”.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin. “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”. In *Proc. International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, August 2000.

[13] Wang Ke, Prithwish Basu, Salma Abu Ayyash, and Thomas D.C. Little. Algorithms for Attribute Based Clustering in Wireless Sensor Networks. Technical report, Boston University, 2003. MCL Technical Report No. 03-24-2003.

[14] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang. “TTDD: A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks”. In *Proc. International Conference on Mobile Computing and Networking (MobiCom)*, Atlanta, GA, September 2002.

[15] S. Madden, M Franklin, J. Hellerstein, and W. Hong. The Design of an Acquisitional Query Processor for Sensor Networks. In *Proc. ACM SIGMOD*, San Diego, CA, June. 2003.

[16] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. “TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks”. In *Proc. Symposium on Operating Systems Design and Implementation (OSDI)*, December 2002.

[17] S. Madden, J. Hellerstein, and W. Hong. “TinyDB: In-Network Query Processing in TinyOS”. Technical Report IRB-TR-02-014, Intel Research, Berkeley, CA, October 2002.

[18] R. Ramanathan and M. Steenstrup. “Hierarchically-Organized Multihop Mobile Networks for Quality-of-service Support”. *ACM/Baltzer Journal on Mobile Networks and Applications*, 3(2), August 1998.

[19] C. C. Shen, C. Srisathapornphat, and C. Jaikaeo. “Sensor Information Networking Architecture and Applications”. *IEEE Personal Communications*, pages 52–59, August 2001.

[20] Y. Yao and J. E. Gehrke. “Query Processing for Sensor Networks”. In *Proc. CIDR Conference*, Asilomar, CA, January 2003.

[21] J. Zhao, R. Govindan, and D. Estrin. “Computing Aggregates for Monitoring Wireless Sensor Networks”. In *Proc. IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, May 2003. to appear.