

Multi-Criteria Routing in Wireless Sensor-Based Pervasive Environments

QINGLAN LI, JONATHAN BEAVER, AHMED AMER, PANOS K. CHRYSANTHIS
AND ALEXANDROS LABRINIDIS

Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA
Email: {qinglan,beaver,amer,panos,labrinid}@cs.pitt.edu

GANESH SANTHANKRISHNAN

Department of Information Sciences and Telecommunications, University of Pittsburgh, Pittsburgh, PA 15260, USA
Email: ganeshs@sis.pitt.edu

Received: October 24 2005

Abstract—Wireless sensor networks are expected to be an integral part of any pervasive computing environment. This implies an ever-increasing need for efficient energy and resource management of both the sensor nodes, as well as the overall sensor network, in order to meet the expected quality of data and service requirements. There have been numerous studies that have looked at the routing of data in sensor networks with the sole intention of reducing communication power consumption. However, there has been comparatively little prior art in the area of multi-criteria based routing that exploit both the semantics of queries and the state of sensor nodes to improve network service longevity. In this paper, we look at routing in sensor networks from this perspective and propose an adaptive multi-criteria routing protocol. Our algorithm offers automated reconfiguration of the routing tree as demanded by variations in the network state to meet application service requirements. Our experimental results show that our approach consistently outperforms, in terms of Network Lifetime and Coverage, the leading semantic-based routing algorithm which reconfigures the routing tree at fixed periods.

Index Terms—Sensor networks, pervasive environment, adaptive semantic routing, quality of service, multi-criteria

I. INTRODUCTION

The computing environment today is changing quickly with the emergence of small sensor devices and sensor networks. Such wireless sensor networks are expected to be an integral part of any pervasive computing environment, since they allow an unprecedented level of interaction with the physical environment. Such sensor-based pervasive services will also be subject to the same requirements for quality of data (QoD) and quality of service (QoS) that we expect of web services today.

There are several problems in providing such qualities of service in these new pervasive environments. Most problems arise from the inherent limitations of sensor nodes: limited

storage, limited network bandwidth, poor inter-node communication, and limited power of the sensor nodes. Power conservation in particular is a major challenge because battery technologies are advancing at a much slower pace than semiconductor technologies (for CPU/memory) and, as such, the power “divide” will remain a challenge for many years to come.

In wireless sensor networks, energy spent on communications typically supercedes all other power consumption costs, such as CPU processing. In order to reduce communication costs, many approaches toward *in-network processing* have been proposed. The main idea behind *in-network processing* is to perform computation in the network itself (i.e., within individual sensor nodes), thus reducing the size of the data to be sent higher up to other nodes in the network. This helps in reducing power consumption, since computation is much cheaper in terms of energy consumed than communication. The chief among the approaches for *in-network query processing* (*in-network aggregation*) are TAG [17] and Cougar [36]. TiNA [28], [3] is a more recent approach that aims to balance the reduction in energy with the loss of QoD by adhering to user-specified QoD requirements, and works on top of existing *in-network aggregation* schemes such as TAG and Cougar.

In *in-network processing*, communication among sensor nodes is structured as a (routing) *tree* with a base station as its root. As *in-network query processing* gains popularity and sensor networks and applications increase in complexity, it becomes imperative that the creation of the routing tree itself be based on the semantics of the query in addition to standard criteria that are already being used (like minimizing the distance among sensor nodes). This brings the need to develop adaptive routing protocols that consider *multiple* criteria, as opposed to the single-criteria approaches in use today. Such criteria should consider the semantics of the query, as well as of the node characteristics such as energy remaining at the sensor nodes and the power consumption model of the sensor nodes. There have been numerous studies that have looked at the routing of data in sensor networks with the

This paper is an extended version of “Multi-criteria routing in pervasive environment with sensors” which appeared in the *Proceedings of the IEEE International Conference on Pervasive Services* (ICPS 2005), July 2005 [24]. The last author worked on the project while a student in the Computer Science Department at the University of Pittsburgh.

sole intention of reducing communication power or energy consumed. However there has been comparatively less prior work in the area of semantic routing and multi criteria-based routing algorithms that consider other performance goals. In this paper, we explore such adaptive algorithms.

Specifically, the problem we are looking at is as follows. In the process of creating a routing tree in a sensor network, always using the lowest energy path may not be optimal from the point of view of network lifetime and long-term connectivity. Other criteria also need to be considered such as the semantics of in-network processing and energy remaining at nodes when constructing routing trees.

The contributions of this paper include the introduction of a *semantic and multi-criteria based routing protocol*, which has shown significant performance improvement over the state of the art. Also, this scheme is inherently self-optimizing. We demonstrate these performance improvements by considering three main evaluation areas in this paper:

- 1) *Network Lifetime* goals – Requiring a particular Network Lifetime (e.g., 50%) involves defining a running network as needing at least that many nodes available in order to be considered useful. In this manner, the relative performance of different network management algorithms is evaluated based on how much effective running time can be realized before the total number of available nodes drops below this required Network Lifetime value.
- 2) *Coverage* goals – Coverage goals are similar to Network Lifetime, except that the percentage of nodes needed to be alive must satisfy some property. This can either be a requirement that alive nodes are available across the entire physical range covered by the network, that there be a minimum number of nodes available in a specific region of the network, or that there be a minimum percentage of a specific subset of nodes throughout the network alive. We refer to these three requirements as *Network Coverage*, *Regional Coverage* and the *Survivability of Critical nodes* respectively, and define them further in Section V. These metrics represent examples of lifetime requirements that may be imposed by applications that attach an increased importance to particular nodes within the network.
- 3) *The Impact of Adaptation* – The final area of evaluation for our algorithm focuses on the effects of dynamic route update. A strength of our multi-criteria routing algorithm lies in its ability to automatically adjust the network in response to changes in the network status. To fairly evaluate our algorithm, we conduct experiments to isolate the impact of adaptation on its performance.

While different applications may have different definitions of the useful lifetime of a network, it is important to note the general effect that lifetime has on the quality of data. The overall longevity of a sensor network and its nodes has a direct impact on the quality of data that it can provide, with greater longevity and larger numbers of available nodes corresponding to greater opportunities to provide more timely and accurate data.

In summary, the salient features of our proposed multi-criteria routing protocol (MCR) include:

- Demonstrated performance benefits in light of multiple and varied evaluation metrics, each reflecting significantly different user and application requirements.
- Distinct performance benefits gained from both the dynamic nature of the route updates as well as the informed dynamic re-evaluation of the routing tree.
- The inclusion of a mechanism for dynamic route updates that can automatically trigger the rebuilding of routing trees based on a combination of local information and a global goal for the increase of useful network lifetime.

The rest of the paper is structured as follows. We discuss background and related work in the next section, and go on to discuss the multi-criteria routing protocol in Section III. Then we discuss experimental setup and results in Sections IV and V, and conclude with directions for future work in Section VI.

II. BACKGROUND AND RELATED WORK

While the simplest forms of routing depend solely on the passing of messages to neighbors that hear them, the most efficient routing schemes can be classified as either *hierarchical* or *data-centric*. Before we introduce our approach to routing in sensor networks, we will briefly discuss prior art, while touching upon the techniques from ad-hoc and mobile routing as well as quality assurance efforts.

The simplest way to route data is to completely avoid the effort of constructing a route, and to pass the data along through flooding or gossiping [21]. This relies on a maximum number of message hops to guarantee receipt by all nodes. While this is adequate for distributing the data, it is not efficient, and so, techniques for establishing routes were developed that either used the locations and identities of nodes (hierarchical routing) or knowledge of data (data-centric routing).

A. Hierarchical Routing

Hierarchical routing schemes actively maintain and use topological information in constructing routes. The main idea behind hierarchical routing is efficient energy consumption of sensor nodes by involving them in multi-hop communication within a particular cluster and performing data aggregation and fusion in order to decrease the number of transmitted messages to the sink (or destination node). Low-Energy Adaptive Clustering Hierarchy (LEACH) [10] was one of the first hierarchical routing algorithms for sensor networks. Other algorithms include Threshold-sensitive Energy Efficient sensor Network protocol (TEEN) [19], Adaptive Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) [20], techniques that use the router nodes to keep all the sensors connected by forming a dominating set [32], and algorithms based on a three-tier architecture ([37], [38]).

Location awareness was utilized in many routing protocols originally intended for ad-hoc and mobile applications, but are amenable to sensor networks. Examples include GAF [35], Geographic and Energy Aware Routing (GEAR) [39], Minimum Energy Communication Network (MECN) protocol [22], Small Minimum Energy Communication Network (SMECN) [15], and other protocols that actively

attempted to improve the overall network lifetime of an ad-hoc network ([18], [27], [34]).

The Quality of Service in terms of timing has also been considered in the context of hierarchical routing [1]. Examples of such protocols are Sequential Assignment Routing (SAR) ([2], [31]) and the Stateless Protocol for Real-Time Communication in Sensor Networks (SPEED) [8].

B. Data Centric Routing

Many data-centric routing protocols achieve energy efficiency and overhead advantages by avoiding the need to maintain topological information. Examples of data-centric routing include the early Sensor Protocol for Information via Negotiation (SPIN) [9], and the later Directed Diffusion ([13], [6]), and its numerous variants. SPIN used high-level meta-data to allow advertising and on-demand retrieval of data. In Directed Diffusion, the main idea is to query the sensors in an on-demand manner, while the data has been maintained as attribute-value pairs to effectively name the data. This allows nodes to express and maintain lists of attribute-value pairs that represent interests, as well as reply links to neighbors based on the interests they expressed. These reply-links are known as gradients. As such, the nodes in Directed Diffusion have the ability to do in-network data aggregation, which is modeled as a minimum Steiner tree problem [14]. Other algorithms include Constrained Anisotropic Diffusion Routing (CADR) [5], as well as the suggestion of employing multiple pre-planned paths to facilitate the choice of alternate paths without incurring the cost of searching for new routes [7], Rumor routing [4], Gradient-Based Routing [25] and information-directed routing [16].

TAG [17] and COUGAR [36] are data-centric routing protocols that view the network as a huge distributed database. The main idea in TAG and COUGAR is to use declarative queries in order to abstract query processing from the network layer functions and utilize in-network data aggregation to save energy. This abstraction is supported through a new query layer between the network and application layers. An architecture for the sensor database system where sensor nodes select a leader node to perform aggregation and transmit the data to the gateway (sink) was proposed. Thus, both of TAG and COUGAR provide network layer-independent solutions for querying the sensors and are among the most popular data-centric protocols to date. The key difference between TAG and COUGAR is the synchronization method (i.e., the synchronization of message receipt and transmission) between nodes on a single path to the root of the tree. In TAG, synchronization is achieved using the idea of communication slots, whereas in COUGAR it is achieved by having a node wait until it hears from all its children. In both these schemes, simultaneous transmission among different nodes that interfere with each other and waste energy can be avoided by exploring a collision-aware query scheduler ([40], [30]). Further, such a scheduler improves the timeliness and quality of data.

Another data-centric routing protocol is ACtive QUery forwarding In sensoR nEtworks (ACQUIRE) [23], which is designed for complex queries consisting of several sub queries.

The ACQUIRE protocol provides efficient querying via an adjustable range of neighborhood nodes.

C. Construction of Routing Trees

There are several ways in which the routing tree can be built. One relatively simple way is to try to create the tree in such a way that the distance between any two nodes is minimized. This can be done in a greedy manner [12] by having the first node that it hears from be chosen as the parent. The intuition behind this choice is the assumption that if a node is heard from first, it was most likely the closest to the child. This protocol is called *First-Heard-From* (FHF) ([3], [29]) and is used in both TAG and COUGAR. This method as well as other similar methods that consider only the network characteristics, such as link low-loss rate, fail to consider the semantics of the query or the properties/attributes of the sensor nodes and hence cannot take any opportunities for energy savings.

The *Group-Aware Network Configuration* (GaNc) algorithm ([3], [29]) works similar to FHF algorithm: Starting from the root node, nodes transmit the new query. Child nodes select the first node they hear from as their parent and continue the process by further propagating the new query to all neighboring nodes. The process terminates when all nodes have been connected via the routing tree. The main differing point is that in GaNC a child can switch to a better parent while the tree is still being built, unlike in the case of the FHF algorithm. This switch is based on a set of fixed order tie-breaker conditions that go beyond the network characteristics and introduce the semantics of aggregation. A variant of GaNC is GaNCi ([3], [29]) in which a child node could also consider nodes from the same level as possible parents during the process of parent selection. GaNC and its variant have been experimentally shown that are the best performing algorithms in their class of multi-criteria algorithms that consider query semantics.

In the next section, we will present a new data-centric, multi-criteria routing algorithm, which, similarly to GaNC, operates as a layer on top of existing in-network aggregation schemes and which, unlike GaNC, dynamically considers query and node semantics in a goal-driven manner.

III. THE MULTI-CRITERIA ROUTING PROTOCOL

We now describe our algorithm for multi-criteria routing (MCR). MCR is a data-centric routing algorithm in which data is propagated from various locations to a central sink of data, the base station, which becomes the root of the routing tree. As with the First Heard From (FHF) and GaNC algorithms discussed above, the routing tree is created along with the propagation of the query, through the selection of a parent node for each individual node. The base station propagates the query down the network.

Traditionally, signal strength is the main factor considered when constructing the routing tree, where a sensor node would select its parent based on the best link strength. Our multi-criteria routing algorithm provides a mechanism for route construction which considers the best choice of parent node based on the evaluation of the merits of neighboring nodes

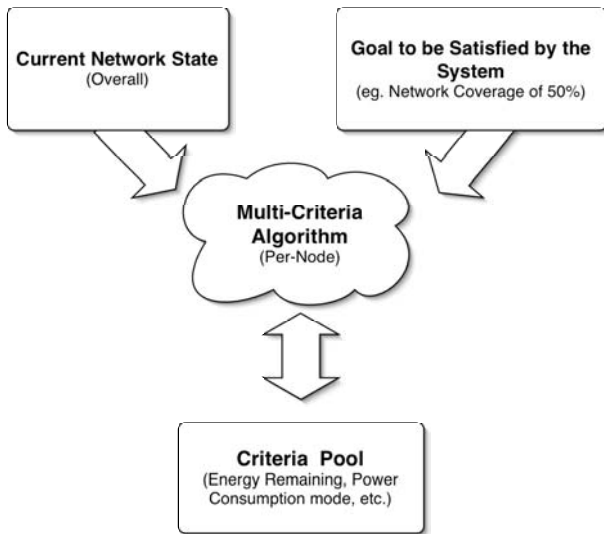


Fig. 1. Overview of the system.

according to an automatically varying mix of criteria. This mix of criteria is varied dynamically by the root node in response to changes in the network state, always with the objective of satisfying an application-defined goal in the best possible way given the current network state. Example of such goals can include a desire to maintain a certain percentage of nodes alive, or a certain distribution of nodes, as per-node energy is depleted.

Figure 1 presents the overall view of our MCR algorithm. MCR accepts the user or application-specified goals as an input to the system, and responds to changes in the network state by triggering a rebuilding of the routing tree. Our algorithm can consider arbitrary criterion C_i , but for the purpose of this paper we have specifically focused on only three. The three criteria that are considered in our current algorithm are the following sensor-node properties:

- 1) the *energy remaining* at the sensor node,
- 2) the *power consumption model* (which specifically refers to the estimated rate of energy consumption at the node),
- 3) the *group membership* of each node.

The algorithm requires a general awareness of the state of the network, which can be achieved through periodic or opportunistic communication with the base station, but the actual construction and modification of the routing tree is performed on a per-node basis (Fig. 2), locally amongst a node n_x and its neighbors.

A. Criteria-Based Route Construction

Our network configuration mechanism considers the semantics of the query and the properties of the individual sensors when dynamically building or rebuilding the routing tree. The initial construction of the routing tree starts with a tree *build request* initiated by the root node, and propagated to the set of neighboring nodes. This message contains an identifier for the sender, the query specification, and a value representing the current level in the tree being constructed, $L(\text{sender})$. Since the base station is the root node of the tree, $L(\text{root}) = 0$. For

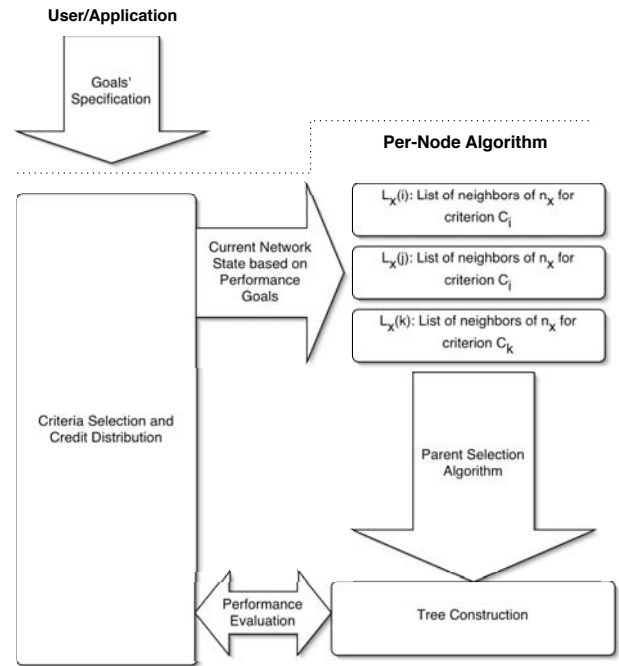


Fig. 2. Separation of the user specification and per node algorithm.

each node receiving the *build request*, the following steps are taken:

- 1) Upon initial receipt of the *build request*, a sensor node i sets its level value $L(i)$ to $L(\text{sender}) + 1$. It also records the parent value (Id) of the sender node, and its group ID. It then sends the tree build request to all its neighbors, after modifying it to reflect the new level, and itself as the sender.
- 2) A node will likely receive multiple *build requests* (from each of its neighbors), and upon subsequent receipts a node may decide to switch to a “better” parent. The definition of a better parent is determined by a weighted combination of the node properties, which is described in more detail below.
- 3) Steps 1 and 2 are repeated until all nodes have propagated the *build request* message.

For selecting a node’s parent we consider: *power consumption model* per node (in Watts), *energy remaining* at nodes (in Joules) and the *group membership* information. The first two criteria are reasonably intuitive, for increased network lifetimes it is better to make use of nodes with more remaining energy, or nodes that appear to be using energy at a lower rate. For example, a node may have fewer sensors, or may be awoken from a low-power state infrequently due to environmental conditions (e.g., by motion sensors), thereby requiring less power.

The significance and usage of the third criteria, *group membership*, requires clarification. Specifically, it is favorable for nodes that will perform in-network aggregation of their data to fall along a common path in the routing tree, i.e., share a parent-child relationship. In-network aggregation depends on the query attributes and the aggregation function. The list of attributes in the group-by clause divides the query result

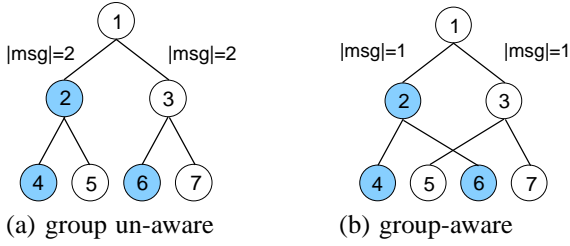


Fig. 3. Benefits of group-aware network configuration.

into a set of groups. The number of groups is equal to the number of combinations of distinct values of the attribute list. Attributes may be static (e.g., a floor location), dynamic based on sensor readings (e.g., levels of light), or dynamic based on other stored values at the node (e.g., color). Hence, readings from two different sensors are aggregated only if they are part of the same group. Since aggregation essentially combines all readings of a particular group into one, a tree in which all members of a group are in the same path incurs smaller message sizes, and is therefore better in terms of overall energy consumed. Further, it requires less space to store the aggregate values at each sensor node. Hence, this criterion is also suitable for space efficiency. To make use of this property, we consider group membership when identifying “better” parent nodes, as they are the neighbors that are members of the same group as the individual sensor node ([28], [29]).

To better illustrate the basic motivation of considering group membership, we use the simple example shown in Fig. 3. In this figure, nodes 2, 4, and 6 (the shaded ones) belong to one group, whereas nodes 1, 3, 5, and 7 belong to a different group. Let us assume that under the standard FHF network configuration (Fig. 3a), nodes 4 and 5 pick 2 as their parent, whereas nodes 6 and 7 pick 3 as their parent. Using in-network aggregation, the message sizes from nodes 2 and 3 to the root of the network will both be 2 (i.e., contain partial aggregates from two groups). On the other hand, if we cluster along the same path nodes that belong to the same group (Fig. 3b) we reduce the size of messages from nodes 2 and 3 in half: each message will only contain the partial aggregate from a single group.

The weighting of these three criteria is dependent on the initial goals offered to the system, and is updated dynamically by individual nodes based on global changes to the goals or local changes in the properties of a nodes neighbors.

B. Neighborhoods and Criteria Lists

Our algorithm uses neighborhoods of nodes, and local per-node lists of such neighboring nodes. The concept of neighborhoods is similar to the use of a neighborhood of nodes (up to d hops away) in ACQUIRE [23], while the maintenance of a local list of nodes is similar to piggy-backing approaches suggested for reliable multi-hop routing [33]. Local to each node, lists of neighboring nodes status are maintained. These lists each represent a priority list based on one of the three evaluation criteria we use for selecting a parent node during tree construction, effectively offering three independent rankings of the neighboring nodes’ desirability based on each of

the evaluation criteria. The first list consists of the neighboring sensor nodes ranked according to their power consumption models (in Watts). The second list consists of the neighboring nodes ranked according to the energy remaining at the nodes. The third list consists of the neighborhood range nodes partially ranked as per the group membership information. This means that nodes in the same group are guaranteed to be nearer to the head of the list than nodes from other groups. The nodes transfer this information in message headers that are transmitted back and forth between nodes and their neighbors. By maintaining ordering information for each of the lists, it becomes possible to efficiently select the most desirable node based on a weighted sum, without being forced to evaluate all nodes in all lists.

The energy remaining at each node decreases throughout the lifetime of the network, from initial deployment and route tree construction, till enough of the right nodes have failed that the network is no longer useful (i.e., dead). The initiation of tree construction and the construction of an initial routing tree are logically illustrated in Fig. 4. While the base station (the root node) is given goal parameters, and thereby an initial global weighting of the three criteria, C_i , $i = 1, 2, 3$, the individual nodes, n_x , namely, A, B, C, D, E, and F, maintain their own lists $L_x(i)$ of their neighboring nodes, indicating their relative merit according to each criterion C_1 , C_2 , or C_3 . The construction of the routing tree is initiated by the root node, which conveys initial weightings of the criteria, and which gathers a global view of the state of the sensor network through piggy-backed node status information. The selection of an individual node’s parent is performed by the node, and the selection is dependent on the criteria weightings and the values of these criteria for the neighboring nodes. For example, A keeps its lists of neighbor nodes and chooses root as its parent according to weighted sums of three criteria C_1 , C_2 , and C_3 . Similarly, C also keeps its lists and chooses A as its parent. It is important to note that this per-node selection is dependent on state information that is purely local to each node and its immediate neighbors.

C. Dynamically Updating Routing Trees

Initially we define a set of goals that need to be satisfied. This is drawn from a pre-determined set of goals that the application might want to fulfill. For instance one possible goal is based on the number of nodes alive such as Network Lifetime of 50%.

Our terminology and approach for the problem is as follows. The “criteria lists” are the per-criteria ranked lists of neighboring nodes; and the corresponding set of weights (for each criterion) are a representation of the suitability of the recommendations with regards to the desired goal (i.e., the current local weighting of each criterion). The ranking of neighbors that a node finally uses to select its parent is based on a weighted combination of the orderings offered by these three criteria lists.

“Weights” are per-node numeric values assigned to individual criteria lists from a central pool, thereby offering a mechanism to assign relative weighting of the criteria based

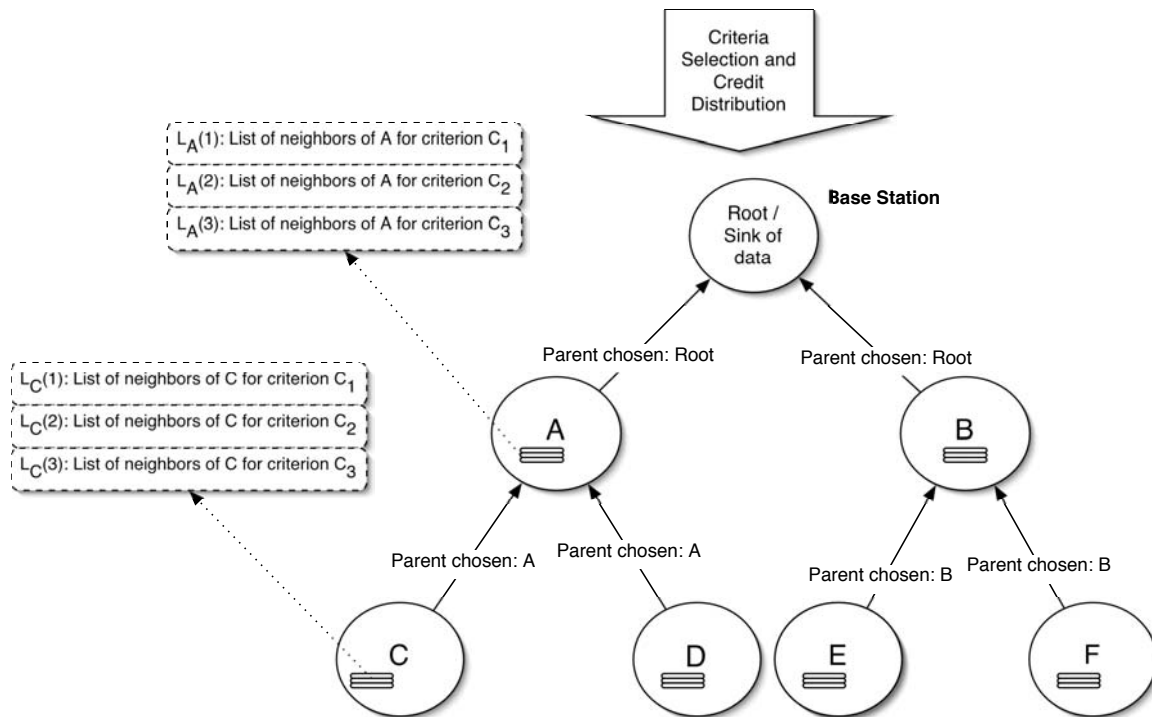


Fig. 4. Constructed routing tree.

on a local node's perception of the criteria's merit. The relative merit of each of our three node criteria is based on how well the individual criterion is seen as contributing to the desired system goal. For example, if the power consumption of a node increases, while its remaining energy is lowered, this node likely becomes a less desirable parent node for its neighbors, as it may now be preferable for the overall lifetime of the system to conserve its energy. We therefore define the distribution of weights depending on the current node status, and desired goals. In other words, weights for the individual lists are in effect specifying the mix of criteria that best achieve the goal. This is in contrast to static schemes such as GaNC, in which the order, and hence the significance, of criteria (i.e., the tie-breaker conditions) is fixed.

Initially the weights are distributed among all nodes. This initial distribution of weights is specified in the build request message that is transmitted from the root to all nodes. Our multi-criteria routing algorithm decides the parent for each node with a weighted average of the criteria list rankings. Depending on the observed outcome (e.g., an observed trend towards failing the goals) the base station may choose to update weights among criteria globally. We now go on to describe the general mechanism of weight updates.

D. Proportional Weight Updates

The redistribution of weights is done globally. In other words, we check periodically if the goal is satisfied. If a certain goal is not satisfied, then the weights are redistributed proportionately and the network is reconfigured. That is, with every reconfiguration, the weights are then sent out in the *build request* message. We have assumed here that the

base station has global information of alive and dead nodes. Whenever a node transmits its reading, it has the opportunity to piggyback such information in the header of the message. In this manner, such knowledge can be periodically or opportunistically acquired for all nodes by the base station. This global information is necessary for MCR to improve optimally, but is not necessary to maintain current performance.

The triggering condition is a system parameter that defines when a reconstruction of the routing tree is to commence. Unlike prior algorithms, this can be purely driven by the state of the system, and requires no intervention or central control beyond the specification of this trigger condition. Such conditions could include a certain reduction in the number of living nodes, a rate of reduction in available nodes, a variation in the distribution of available nodes, or any combination thereof. Node availability can also be refined to consider variations in local energy levels and capabilities. With the detection of such a trigger condition, or any failure to meet the desired system goals, the root node can initiate a rebuild of the routing tree, which again is based solely on the consideration of the relative merits of an individual node's neighbors. But in this instance, the selection has automatically been affected globally through the simple inclusion of these new weights in the build request issued by the root node.

Illustrative Example: Let us illustrate the operation of this proportional update of credits scheme with a high-level example. Initially, the base station notifies all nodes of the current goals and their relative weights W_e , W_r , W_g . These weights represent the relative importance of remaining energy, rate of consumption (power), and a common group. If a trigger condition is detected, and the base station is notified, then a revision

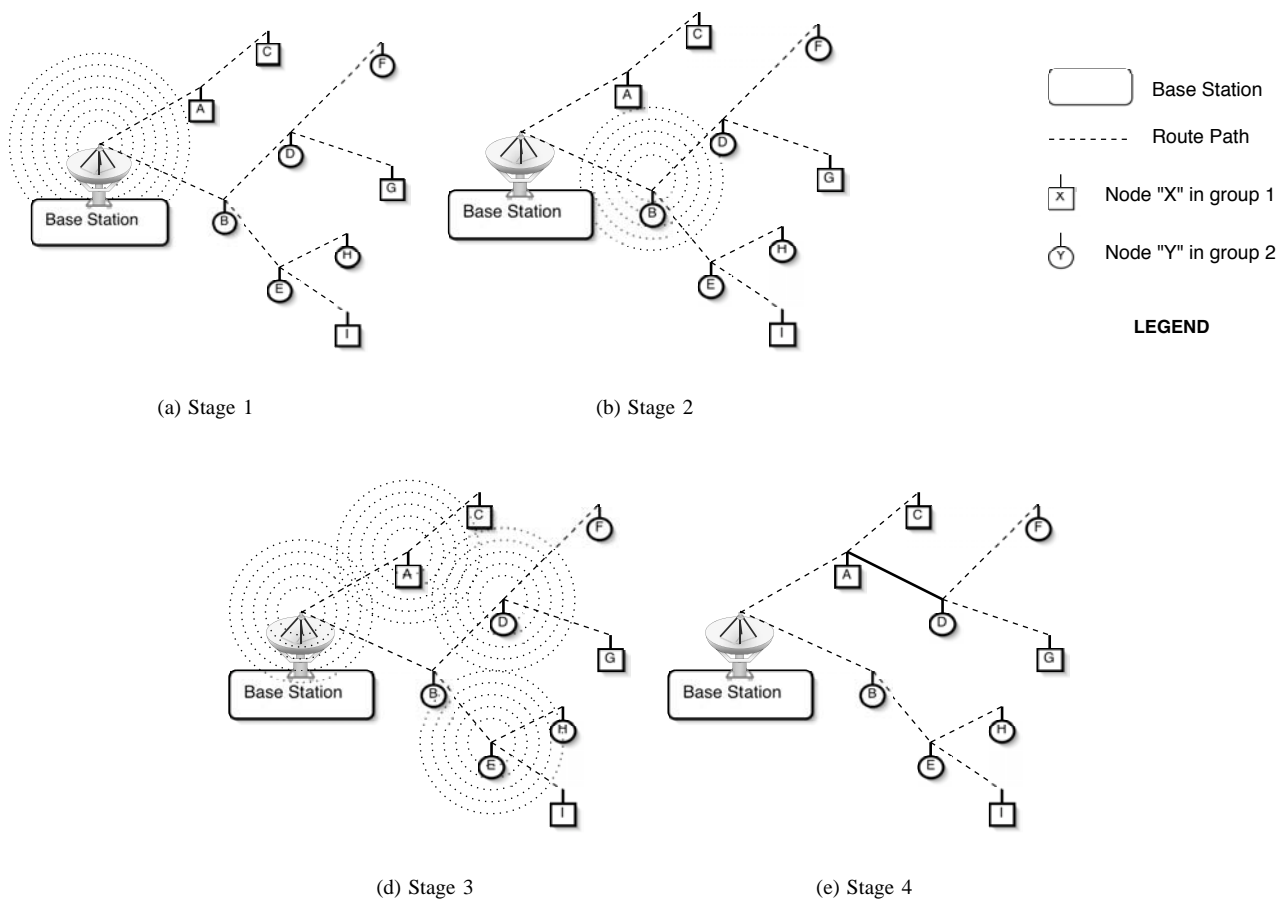


Fig. 5. Illustrative stages for a route update in response to a trigger condition.

of these weights may be required. For example, if a significant percentage of nodes have died (energy depleted), then the importance of the *energy remaining*, and *power* criteria are increased relative to the common *group participation* criteria. The scale of the update is proportional to the difference between the numbers of dead and living nodes. We note here that we have considered the two criteria of energy remaining and power consumption since they exclusively favor the reduction of power consumption and the increased longevity of sensor nodes as opposed to group membership, for which energy conservation is a secondary effect (reducing memory requirements is the primary implication of this criterion).

Figure 5 illustrates the stages that may be involved for a specific route update scenario. Initially a route tree is constructed in response to a build request from the base station (Fig. 5(a)). In this example particular attention should be paid to node **B**, which has been heavily favored by nodes **E** and **D** due its sharing a common group. But as can be seen, **B** now lies in the path of a large number of nodes, and will likely be active frequently, thereby consuming more energy at a higher rate than most other nodes. In the process of regular communication, the reduction of **B**'s energy levels are communicated (Fig. 5(b)), eventually leading to a trigger condition (for clarity we can assume that the trigger for this illustration is an large change in energy level at any node).

The trigger condition results in a new build request being issued by the base station, but it should again be noted that only the neighboring nodes need per-node energy-level and power information (Fig. 5(d)). During reconstruction, individual nodes pick their preferred parent node, and while **A** was not initially the preferred parent of **D**, it is now selected (Fig. 5(e)) due to the reduced energy levels and high power consumption of **B**, and the subsequently increased importance of energy levels over group membership when evaluating peers (the new weights broadcast by the base station).

Table I offers an overview of the the additional data exchanged in MCR, as compared to GaNC. It specifically focuses on the additional data that needs to be exchanged between nodes, and in some cases the root node, to enable the algorithm. We should also note here that, while MCR is effective as an adaptive route update scheme, an ideal dynamic list update scheme is the subject of continued investigation. In the next section, we describe our data collection and experimentation.

IV. EXPERIMENTAL DESIGN

We evaluate our proposed algorithm, MCR, using simulation and compare it to GaNC. We simulated sensor networks that were arranged in the form of a grid, with grid sizes ranging from 15×15 sensors to 50×50 sensors. We assume a grid

TABLE I
DATA EXCHANGED THAT IS SPECIFIC TO MULTI-CRITERIA
ROUTING(MCR)

Data	Description
Build Request:	Sender Identifier, Query Specification, Current Level, Criteria Weights
Local Status	Energy remaining and estimated power consumption rate for the local node. This information is periodically passed to the root node.
Neighbor Status	Group membership, live status, and energy remaining for neighboring nodes. This information is maintained locally by each node for its neighbors. Only timeout or death of neighbors needs to be forwarded to the root node, but this is not specific to MCR.

TABLE II
PARAMETERS IN THE EXPERIMENT

Parameter	Value
Grid Size	15×15, 20×20, 25×25 ... 50×50
Experiment Duration	1 to 91.1 months in simulation
Performance Metrics	Network Lifetime Network Coverage Regional Coverage Survivability of Critical nodes
Termination Condition (minimum system goals)	Network Lifetime 40% to 70% Network Coverage 40% to 70% Regional Coverage 40% to 70% Survivability of 40% to 70% Critical nodes
Functions	SUM, AVERAGE, and MAX
Initial Per-Node Energy	250 Joules
Energy Consumption Rate	1, 2, ... 4 randomly
Group Numbers	1, 2, ... 100 randomly
Initial Weight Distribution	30 to 25000 randomly

of sensors in which the range of transmission is restricted to a single hop. This is in keeping with the basic assumption of various other in-network aggregation schemes.

We also assume that we can piggyback control information along with the regular data transmission by taking advantage of unused space within packets of fixed sizes, that are most often bigger than the actual data to be transmitted. For our experiments, we focused on the standard SQL aggregation functions SUM, AVERAGE, and MAX. We did not include the MIN function, which is similar to MAX, nor did we include COUNT, which is similar to SUM.

We performed extensive experiments, in which we measured the longevity of the overall sensor network, as well as its ability to survive with a specified degree of physical coverage, in the face of node failures due to eventual energy depletion at such nodes. We have also conducted experiments to evaluate the impact of adaptivity on network lifetime. Specifically, we aimed to isolate the impact of our proposed adaptive criteria-weighting scheme upon network longevity, as opposed to strict periodic adaptation.

The simulator was written using C++ and CSim[26]. The energy remaining at each node is measured in Joules; for simulation purposes we define a maximum energy value equivalent to the energy of a typical battery cell. We assume that each node fails when it exhausts its energy reserves. Over a period of time, as the node transmits data and performs various computations, we reduce this value appropriately for the various operations and when the minimum value is reached, we mark the node as dead. We model the power consumption of each sensor node drawn randomly from a distribution and assume a rate of decay for all the sensor nodes.

Group information is modeled as participation based on a group identifier. The group identifier can consider static properties (e.g., grouping all nodes in the same floor together) or dynamic properties (e.g., grouping together all nodes with a light intensity reading above a certain threshold).

Weights at each node are modified in a distributed fashion.

Initially, the base station notifies all nodes of the current goals and their relative weights W_e , W_r , W_g . These weights represent the relative importance of remaining energy, rate of consumption (power), and a common group. We simulated different goals of Network Lifetime, Network Coverage, Re-

gional Coverage, and Survivability of Critical nodes.

For evaluating the trigger condition in our simulations, we evaluated the status of the network every 10 minutes. If more than 5% of nodes have died since the last evaluation period, the root node will update the weights of the three criteria. Specifically, the weighting of the *energy remaining*, and *power* criteria are increased relative to the *common group* participation criteria. The scale of the update is proportional to the difference between the numbers of dead and living nodes. The specific updates applied to the weights are as follows:

$$W_e = W_e \cdot (1 + \text{dead}/\text{all})$$

$$W_r = W_r \cdot (1 + \text{dead}/\text{all})$$

$$W_g = W_g \cdot (1 - \text{dead}/\text{all})$$

As we discussed above, the *energy remaining*, and *power* criteria are selected for increase since they exclusively favor the reduction of power consumption and selection of nodes with greater potential running times.

V. METRICS AND EXPERIMENTAL RESULTS

We evaluate performance in terms of four performance metrics: Network Lifetime, Network Coverage, Regional Coverage and the Survivability of Critical nodes. Network Lifetime and Network Coverage deal with the time during which a percentage of nodes can remain alive, whereas Survivability of Critical nodes focuses on the need to maximize the run time of critical nodes. Table V summarizes the parameters of our experiments.

- *Network Lifetime* is defined as the amount of time during which no less than a certain percentage of nodes remains alive.
- *Regional Coverage* is similar, but is defined in our experiments for only a subset of the nodes, specifically the first 100 nodes. In this manner, Regional Coverage gives an indication of how long a network can remain functional while maintaining a percentage of all nodes active (effectively localized to a specific region of interest).
- *Network Coverage* is similar to Regional Coverage, but is defined with the stricter requirement that no sub-grid,

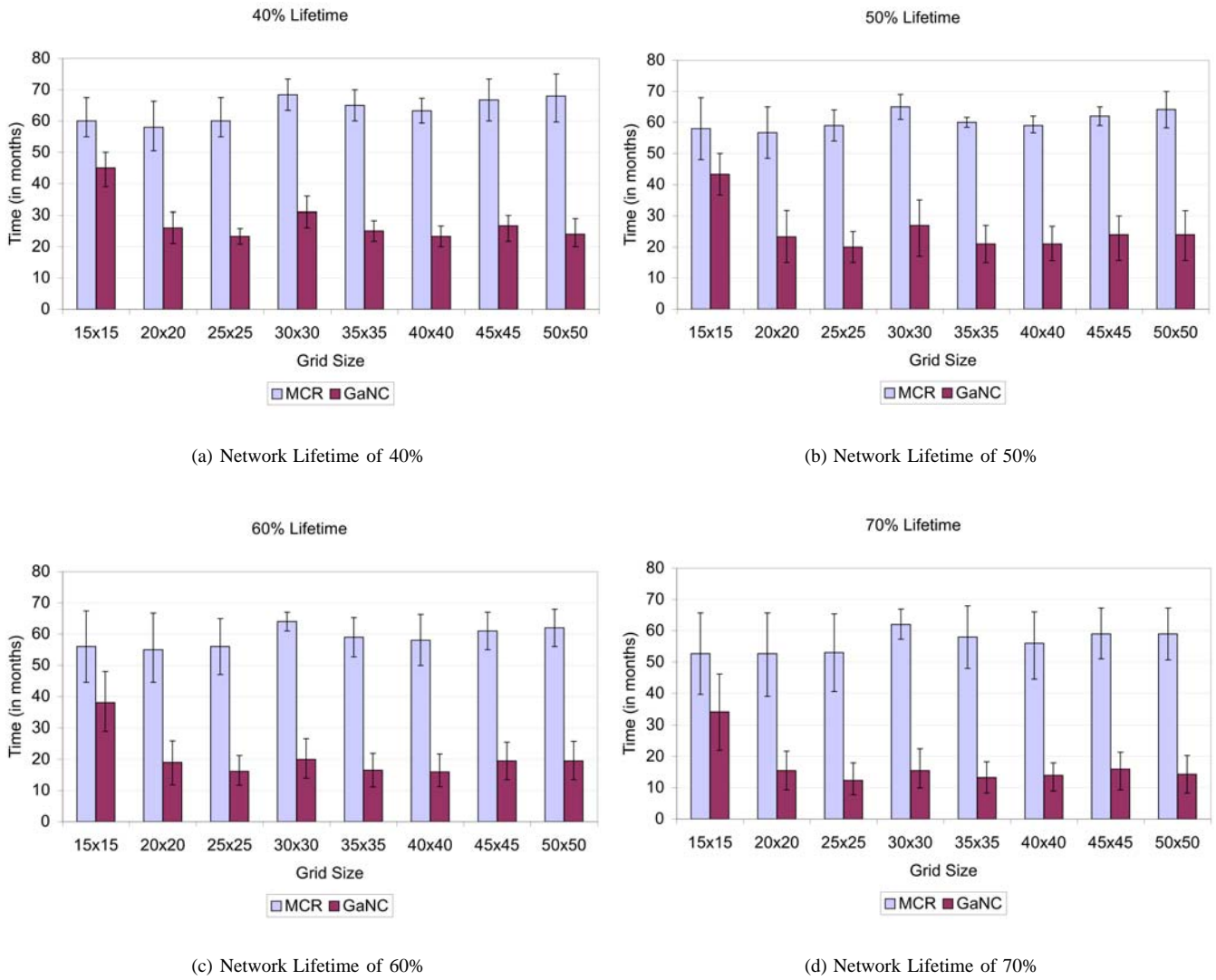


Fig. 6. Comparison of Network Lifetime for varied grid sizes.

across the range of all nodes, suffers a loss of more than a certain percentage of its nodes. This metric measures a network's ability to survive with a balanced number of nodes across the entire network, i.e., without depletion occurring in any localized subset of all nodes.

- *Survivability of Critical Nodes* is defined as the percentage of *critical* nodes alive, where critical nodes are the segment of nodes that need to be preserved the most in terms of importance.

In our experiment, we set lifetime goals ranging from 40% to 70%. This range was used for Network Lifetime, Regional Coverage, Network Coverage, and the Survivability of Critical nodes. The range was chosen to reflect realistic values that could be used when deploying a wireless sensor network.

We present results for Network Lifetime in Section V-A, and demonstrate how MCR performs in comparison to GaNC when considering a goal of keeping a certain percentage of all sensor nodes alive. In Section V-B we compare the algorithms' performance when we are interested in having representation

from all parts of the sensor network (i.e., when we want to force a certain percentage of lifetime, recursively over certain-size sub-grids, instead of a global percentage of lifetime). For both Section V-A and Section V-B we compare MCR to a static GaNC tree, allowing MCR to automatically reconstruct the routing tree in response to changes in the total number of available nodes. GaNC is kept static as it does not provide a similar automated mechanism for deciding when to reconstruct the routing tree. Despite this inherent inability of the GaNC scheme (which is common among almost all reconstruction schemes), to offer a fair comparison, we also compare both algorithms with fixed reconfiguration periods in Section V-C. In this manner, we effectively disable MCR's ability to automatically trigger reconstruction, and thus level the playing field for GaNC, since both algorithms will then adapt their trees at the same frequency.

In our experiments the multi-criteria routing protocol was shown to outperform the Group aware Network Configuration (GaNC) [3] algorithm in all the measured metrics. This

was true when MCR was allowed to automatically initiate route tree reconfigurations, as well as when this ability was artificially impeded by forcing fixed-interval tree rebuilds for MCR.

A. Network Lifetime Goals

Figure 6 presents the results from comparing the MCR and GaNC protocols on the basis of a desired Network Lifetime goal. In these figures, the x-axis is the grid size (which varies from 15×15 sensors to 50×50 sensors). The y-axis is the time at which the network dies, measured in months, assuming a full battery for all sensors initially. We present experiments with Network Lifetimes ranging from 40% to 70%.

As mentioned above, in all our experiments, the redistribution algorithm executes periodically every 10 minutes and checks for a reduction in the number of available nodes in excess of 5%. All results are in time measured in terms of months. As with prior art, we have simulated the values used for transmission consumption as a percentage of the total energy of each sensor node [11]. The energy values are maintained per node and updated (locally to each node) with every transmission that the node performs. In this manner our simulation takes into account the energy overheads of the routing tree construction. All experiments were run multiple times to eliminate any statistical errors.

It is clear from these figures that the multi-criteria routing policy fares consistently better. This can be attributed to the fact that the nodes' parents are redistributed, thus preventing any single node from over-utilization of its energy.

From the results in this set of experiments, we see that the multi-criteria routing policy outperforms the GaNC scheme, offering continually prolonged network activity. This behavior is consistent across different grid sizes and also for different network lifetime goals (40% to 70%). From this point on, we will only report results for the two extreme goal percentages (40% and 70%), since the behavior of the in-between points is as expected.

B. Network Coverage Goals

While it is straightforward to consider a network as effectively alive as long as a certain percentage of all its nodes remain alive, it is often the case that some nodes are more valuable than others. For example, nodes that are physically located at one border of a grid may be acting as a link to base stations, while remote nodes are difficult to access (e.g., in a forest). On the other hand, it may be the case that the usefulness of the network depends on being able to collect readings from a large physical region (e.g., measure temperature at various points of the core in a nuclear reactor). In this case, it's important that for every sub-area (i.e., sub-grid) across the network, there remain sufficient active nodes within each sub-grid. For the latter case, a lifetime goal based on a desired Network Coverage is appropriate, whereas for the former, a goal of Regional Coverage or Survivability of Critical nodes may be the metric of interest. We have found MCR to be an effective improvement over GaNC regardless of which of these metrics we consider.

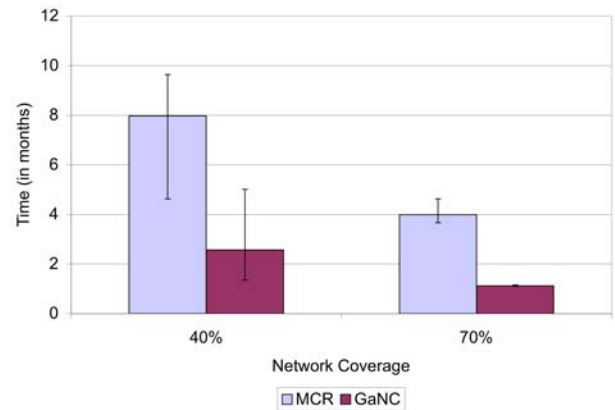


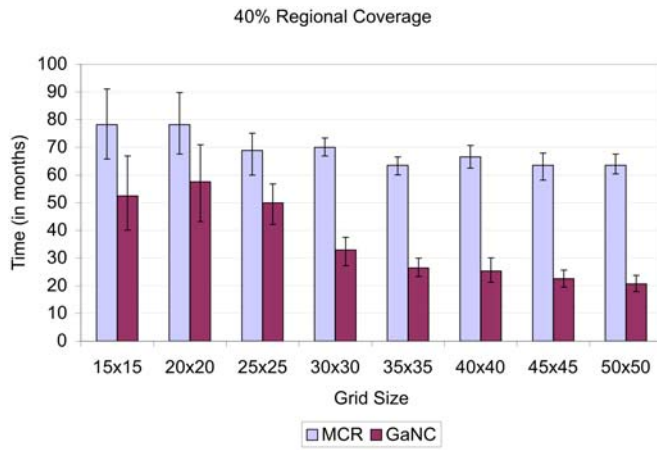
Fig. 7. Longevity for different Network Coverage goals (for a 15×15 grid).

In the set of experiments presented in Fig. 7, we compare MCR to GaNC and focus on network coverage for a 15×15 grid, where the sub-grid was defined to be 5×5 . We explored two different network coverage goals for each sub-grid: 40% and 70%. Clearly, we expect the 70% case to be more “demanding” and thus the sensor network will fail this goal sooner (than the 40% case).

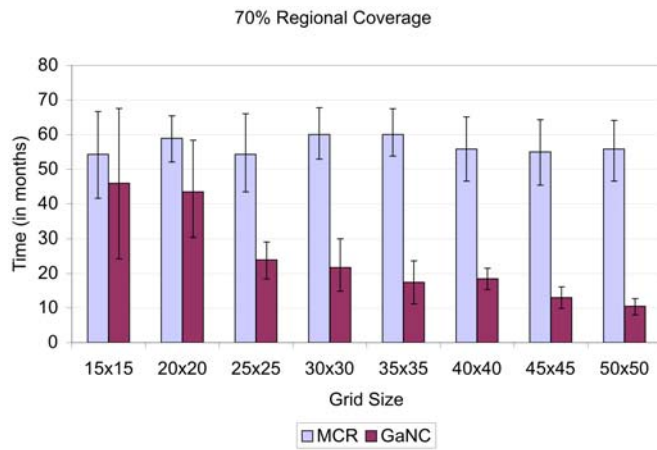
In Fig. 7, MCR can be seen to offer network lifetimes over three times longer than with GaNC alone, consistently, across network goal levels. Although both algorithms result in shorter network lifetimes when we have higher coverage requirements, it is interesting to point out that MCR is performing much better than GaNC, to the level that the lifetime of the sensor network under MCR with a 70% network coverage goal (third bar in Fig. 7), is almost 60% better than that of GaNC with a 40% coverage goal (second bar in Fig. 7).

If we consider the requirement that only a fixed subgroup of nodes maintain the lifetime goal, then our metric of choice is Regional Coverage. Figure 8 presents the results from comparing MCR and GaNC for varying grid sizes on the basis of Regional Coverage. In these figures, the x-axis is grid size (which ranges from 15×15 sensors to 50×50 sensors). The y-axis is the time at which the network dies, measured in months. We consider Regional Coverage goals of 40% to 70%. Clearly, MCR outperforms GaNC across the board.

In Fig. 9, we compare the Survivability of Critical nodes for different Network Lifetime Goals. In this set of experiments, we define a set of critical nodes (not simply a contiguous region) and measure the lifetime of those nodes as the Survivability of Critical nodes in the network. Specifically, we define a fixed random selection of nodes as critical nodes in the network for simulation purposes. In this figure, the x-axis is varying Network Lifetime Goals ranging from 40% to 70%. The y-axis is the Survivability of Critical nodes measured as a percentage of critical nodes alive. We can see that the multi-criteria routing protocol seems to outperform the Group aware Network Configuration (GaNC). It should be noted that MCR is particularly useful in applications where it is possible to identify specific subsets of nodes as critical, since unlike GaNC, it is straightforward to define a variation in the state of such nodes as a trigger condition to initiate, and a criteria to inform, the routing-tree reconfiguration.



(a) Regional Coverage of 40%



(b) Regional Coverage of 70%

Fig. 8. Comparison of Regional Coverage for different grid sizes.

C. The Impact of Adaptation

One of the advantages of MCR is that it can automatically decide on the best time to reconfigure the routing tree, specifically in response to triggering conditions from the state of the network. On the other hand, it is possible to improve the performance of GaNC by periodically reconstructing the routing tree. GaNC would therefore require the selection of such a period – a fixed value.

In this set of experiments, we investigate whether the performance benefits of MCR are due solely to its adaptation alone, or if there is a benefit gained from its ability to dynamically adjust its weighting of evaluation criteria. For this purpose we selected a range of reconfiguration periods (10, 100, and 1000 minutes) and allowed GaNC to reconfigure the routing tree at such a frequency. To allow for comparison against MCR, we artificially forced the reconfiguration of the MCR routing tree at the same frequency as GaNC (i.e., ignoring the triggering capability of MCR). In this manner, both algorithms reconfigure at the same rate, and MCR's main difference is its ability to evaluate neighboring nodes based on a varying

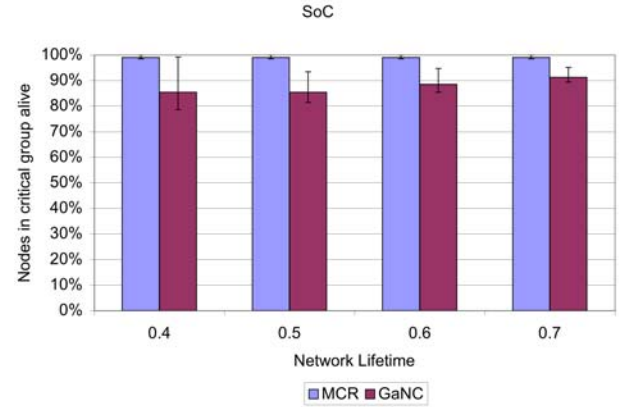


Fig. 9. Comparison of Survivability of Critical nodes (SoC) for different Network Lifetime goals.

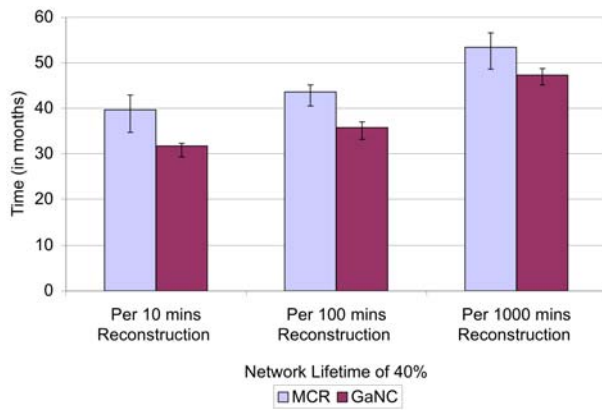
weighting scheme. This configuration is disadvantageous to MCR, preventing the algorithm from triggering tree rebuilds in response to network status. In spite of this artificial handicap, as we will see, MCR consistently outperforms GaNC and offers significant performance improvements.

In Fig. 10 we consider network lifetime goals of 40% and 70%. In these cases, the network is considered dead after less than 40% or 70% of all nodes remain alive. We have forced tree reconfigurations at 10-, 100-, and 1000-minute intervals. As we can see, increasing the interval results in an overall increase in the longevity of the network, suggesting that frequent reconfigurations can result in significant energy overheads. An incorrect selection of such a system parameter (reconfiguration frequency), can result in a 30% reduction in overall network lifetime. This is a big disadvantage on all routing tree reconstruction schemes (like GaNC) that require a fixed period. MCR does not require any such *a priori* selection.

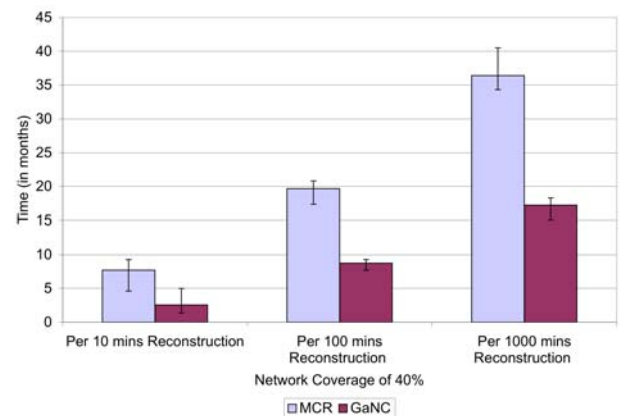
Despite being forced to reconfigure at fixed intervals (same with GaNC), our proposed algorithm MCR, performs consistently better than GaNC, offering a 10% to 20% improvement in network lifetime. This improvement is attributed to MCR's ability to dynamically redefine the importance of node evaluation criteria (dynamic and proportional weight updates).

In Fig. 11 we consider the goal of 40% and 70% Network Coverage. In this experiment, Network Coverage was defined as the requirement that at least this percentage of nodes is alive in every 5×5 sub-grid of the network; in the general case it can be any size sub-grid. As we can see in Fig. 11(b), the strict requirement that 70% of all nodes in all sub-grids remain alive results in a rapid death of the network, but MCR consistently offers an improvement, ranging from 10% to 60% over GaNC. When the requirement is relaxed, allowing the network to continue with as little as 40% of nodes alive per sub-grid (Fig. 11(a)), we see a generally greater improvement in network longevity, but the performance gains of MCR are even more pronounced. From Figs. 10 and 11, we can see that MCR consistently outperforms GaNC, but will offer even greater performance improvements if reconfigurations are frequent, or if there is more leeway in allowing nodes to die without failing the global system goals.

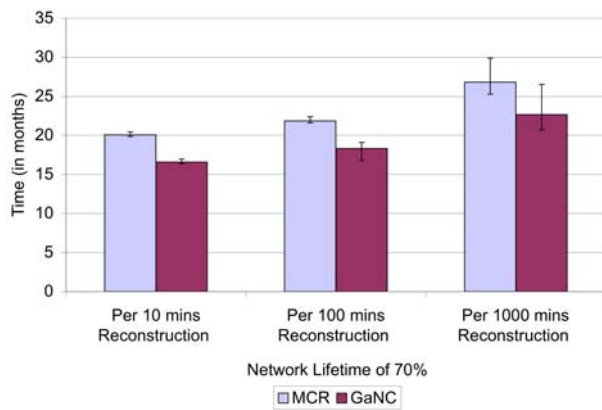
In summary, we can see that the multi-criteria routing pro-



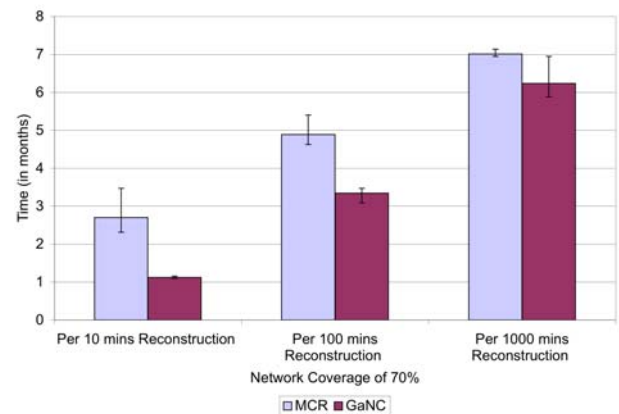
(a) 40% Network Lifetime with forced adaptation



(a) 40% Network Coverage with forced adaptation



(b) 70% Network Lifetime with forced adaptation



(b) 70% Network Coverage with forced adaptation

Fig. 10. Comparison of Network Lifetime under different forced-reconfiguration frequencies.

Fig. 11. Comparison of Network Coverage under different forced-reconfiguration frequencies.

tolcol (MCR) offers significant improvements over the Group aware Network Configuration (GaNC) protocol in terms of Network Lifetime, Network Coverage, Regional Coverage, and Survivability of Critical nodes. Moreover, the overhead of power consumption for tree construction is comparable between the two approaches, and the improvements offered by MCR are due to both its ability to adapt, as well as to its ability to dynamically vary the node selection criteria. Hence MCR is a very good mechanism for in-network aggregation and is quite versatile as it can be deployed over data-centric routing mechanisms such as TAG or COUGAR.

VI. CONCLUSIONS

In this paper, we have designed and implemented a multi-criteria routing scheme for sensor networks with pervasive services in mind. Our scheme exhibits significant performance improvement with minimal overhead when compared to the current state of the art routing algorithms. Our contributions in this paper include:

- The introduction of a multi-criteria routing protocol (MCR) that demonstrates significant improvement in performance across multiple metrics.

- The isolation of the effects and impact of dynamic route updates on network longevity.
- The introduction of a mechanism for dynamic route updates that can automatically trigger the rebuilding of routing trees based on a combination of local information and a goal of increasing the useful lifetime of the network.
- The evaluation of several metrics that reflect different application expectations with regards to the usefulness of a network of sensor nodes.

While we have shown that our adaptive multi-criteria algorithm improves the longevity of system nodes, and the longevity of the connected network, we have also seen that it results in a higher quality of service by allowing the survival of more critical nodes. This example was specific to wireless sensor networks, but any pervasive computing system that depends on the interconnection of its nodes for its services, and the routing of data among them, could potentially benefit from such adaptive multi-criteria algorithms for the management of communication routing. Future work will include the consideration of additional criteria, the refinement of the weight update algorithm, and the development of more application examples

to test the usefulness of our approach.

ACKNOWLEDGMENTS

We would like to thank Mohamed Sharaf for valuable input and contributions to the experimental implementation. This work was supported in part by the National Science Foundation under award ANI-0325353, and by the University of Pittsburgh through faculty startup funding.

REFERENCES

- [1] K. Akkaya and M. Younis. An energy-aware qos routing protocol for wireless sensor networks. *Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003)*, pp. 16–27, 2003.
- [2] I. Akyildiz, W. Su, Y. Sanakarasubramaniam and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
- [3] J. Beaver, M. A. Sharaf, A. Labrinidis and P. K. Chrysanthis. Location-aware routing for data aggregation for sensor networks. In *Proceedings of the Geo Sensor Networks Workshop*, pp. 189–210, 2003.
- [4] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. *Proceedings of the 1st Workshop on Sensor Networks and Applications*, pp. 22–31, 2002.
- [5] M. Chu, H. Haussecker and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3):293–314, 2002.
- [6] D. Estrin, R. Govindan, J. S. Heidemann and S. Kumar. Next century challenges: Scalable coordination in sensor networks. *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 263–270, 1999.
- [7] D. Ganesan, R. Govindan, S. Shenker and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, 2001.
- [8] T. He, J. A. Stankovic, C. Lu and T. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. *Proceedings of the 23rd International Conference on Distributed Computing Systems*, pp. 46–56, 2003.
- [9] W. Heinzelman, J. Kulik and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 174–185, 1999.
- [10] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of 33rd Annual Hawaii International Conference of System Sciences*, pp. 3005–3014, 2000.
- [11] J. L. Hill and D. E. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, 2002.
- [12] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. *22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pp. 457, July 2002.
- [13] C. Intanagonwiwat, R. Govindan and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 56–67, 2000.
- [14] B. Krishnamachari, D. Estrin and S. Wicker. Modelling data-centric routing in wireless sensor networks. *Proceedings of the 21st Conference of the IEEE Communications Society*, 2002.
- [15] L. Li and J. Y. Halpern. Minimum energy mobile wireless networks revisited. *Proceedings of the IEEE International Conference on Communications*, pp. 278–283, 2001.
- [16] J. Liu, F. Zhao and D. Petrovica. Information-directed routing in ad hoc sensor networks. *Proceedings of the 2nd ACM international conference on Wireless Sensor Networks and Applications*, pp. 88–97, 2003.
- [17] S. R. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong. TAG: a Tiny AGgregation Service for ad-hoc sensor networks. *SIGMOD Record*, pp. 131–146, 2002.
- [18] M. Maleki, K. Dantu and M. Pedram. Power-aware source routing protocol for mobile ad hoc networks. *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 72–75, 2002.
- [19] A. Manjeshwar and D. P. Agrawal. TEEN: A protocol for enhanced efficiency in wireless sensor networks. *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, pp. 189, 2001.
- [20] A. Manjeshwar and D. P. Agrawal. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing*, p. 48, 2002.
- [21] E. L. Miller and R. H. Katz. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.
- [22] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. *Proceedings of the IEEE International Conference on Communications*, pp. 1633–1639, 1998.
- [23] N. Sadagopan, B. Krishnamachari and A. Helmy. The ACQUIRE mechanism for efficient querying in sensor networks. *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 149–155, 2003.
- [24] G. Santhanakrishnan, Q. Li, J. Beaver, P. Chrysanthis, A. Amer and A. Labrinidis. Multi-criteria routing in pervasive environment with sensors. *Proceedings of the IEEE International Conference on Pervasive Services*, pp. 7–16, 2005.
- [25] C. Schurgers and M. B. Srivastava. Energy efficient routing in wireless sensor networks. *MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force*, pp. 357–361, 2001.
- [26] H. Schwetman. CSIM user's guide. MCC Corp.
- [27] R. C. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 350–355, 2002.
- [28] M. A. Sharaf, J. Beaver, A. Labrinidis and P. K. Chrysanthis. TiNA: a scheme for Temporal Coherency-Aware in-Network Aggregation. *Proceedings of the Third ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 69–76, 2003.
- [29] M. A. Sharaf, J. Beaver, A. Labrinidis and P. K. Chrysanthis. Balancing energy efficiency and quality of aggregate data in sensor networks. *VLDB J.*, 13(4):384–403, 2004.
- [30] D. Sharma, V. I. Zadorozhny and P. K. Chrysanthis. Timely data delivery in sensor networks using whirlpool. *DMSN '05: Proceedings of the 2nd international workshop on Data management for sensor networks*, pp. 53–60, 2005.
- [31] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, 2000.
- [32] L. Subramanian and R. H. Katz. An architecture for building self-configurable systems. *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 63–73, 2000.
- [33] A. Woo, T. Tong and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 14–27, 2003.
- [34] J. Wu, F. Dai, M. Gao and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communications and Networks*, 4(1):59–70, 2002.
- [35] Y. Xu, J. Heidemann and D. Estrin. Geography-informed energy conservation for ad hoc routing. *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 70–84, July 2001.
- [36] Y. Yao and J. E. Gehrke. The Cougar approach to in-network query processing in sensor networks. *Proceedings of the ACM Symposium on Operating System Design and Implementation*, pp. 131–146, 2002.
- [37] M. Younis, M. Youssef and K. Arisha. Energy-aware routing in cluster-based sensor networks. *Proceedings of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 129–137, 2002.
- [38] M. Youssef, M. Younis and K. Arisha. A constrained shortest-path energy-aware routing algorithm for wireless sensor networks. *Proceedings of the IEEE Wireless Communication and Networks Conference*, pp. 794–799, 2002.
- [39] Y. Yu, R. Govindan and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA/CSD-TR-01-0023, 2001.
- [40] V. I. Zadorozhny, P. K. Chrysanthis and P. Krishnamurthy. A framework for extending the synergy between mac layer and query optimization in sensor networks. *DMSN '04: Proceedings of the 1st international workshop on Data management for sensor networks*, pp. 68–77, 2004.



Qinglan Li is a third year Ph.D. student in the Department of Computer Science at University of Pittsburgh. She holds a BS degree in Computer Science from Northern Jiaotong University, China and a MS degree in Computer Science from Central Michigan University. Her research interests include data dissemination in sensor network, data stream processing, semantic searching, XML schema integration and verification, and XML query processing and optimization. She is a member of ACM SIGMOD.



Jonathan Beaver is a fourth year Ph.D. student in the Department of Computer Science at University of Pittsburgh. He holds a dual BS degree in Computer Science and Finance and a MS degree in Computer Science, all from University of Pittsburgh. His research includes mass data dissemination in hybrid systems, selective data dissemination in sensor networks, and semantic based routing in sensor networks. He currently works for the National Energy and Technology labs of the United States while finishing up his PhD work in multicast based

data dissemination.



Ahmed Amer is an Assistant Professor with the Computer Science Department of the University of Pittsburgh. Dr. Amer received his Ph.D. from the University of California, Santa Cruz, where his research focused on predictive data management. His research interests are primarily aimed at systems issues, particularly storage and file systems, adaptive self-tuning systems, operating systems, and distributed systems. He received his Bachelor's and Master's degrees in Computer Science from the American University in Cairo. He is a member of

ACM (SIGOPS and SIGMOD) and IEEE (Computer, Information Theory, and Geoscience and Remote Sensing societies).



Panos K. Chrysanthis is a Professor of Computer Science at the University of Pittsburgh and an Adjunct Professor at Carnegie Mellon University. He received his B.S. from the University of Athens, Greece and his M.S. and Ph.D. from the University of Massachusetts at Amherst. His current research focus is on mobile and pervasive data management including sensor networks. In 1995, he was a recipient of the National Science Foundation CAREER Award for his investigation on the management of data for mobile and wireless computing. His research

accomplishments have been published in over 100 papers in journals and peer-reviewed conferences and workshops in the field of data management. In addition, his publications include a book and book chapters on transaction processing and data access in distributed, mobile and web databases. He is on the editorial board of the VLDB Journal, and was program chair of several workshops and conferences related to mobile data management. More recently, he was the ICDE 2004 Vice Chair for the area of distributed, parallel and mobile databases and the General Chair of MobiDE 2003 and MDM 2005. He is a member of ACM and IEEE.



Alexandros Labrinidis received a Ph.D degree in Computer Science from the University of Maryland, College Park in 2002. He is currently an assistant professor at the Department of Computer Science of the University of Pittsburgh and a co-director of the Advanced Data Management Technologies Lab. He is also an adjunct assistant professor at Carnegie Mellon University (CS Dept) and the information director for ACM SIGMOD. His research interests include web-databases, realtime data management, data management for sensor networks, quality of

data, data management in p2p networks, data stream management systems, and data management for mobile and tiny devices. In 2005, Dr. Labrinidis served as the program co-chair of MobiDE'05 (the 4th International ACM Workshop on Data Engineering for Wireless and Mobile Access, co-located with SIGMOD) and DMSN'05 (the 2nd International VLDB Workshop on Data Management for Sensor Networks, co-located with VLDB).



Ganesh Santhanakrishnan is a graduate student in the Department of Information Sciences and Telecommunications at the University of Pittsburgh. He has worked on different research areas which have yielded journal and conference publications on e-learning and distance education, knowledge management, caching, pre-fetching, and data management in sensor networks. He earned his Bachelors in Computer Science and Engineering from the National Institute of Technology, Tiruchirapalli and his MS degree in Computer Science from University

of Pittsburgh.