

Energy-Aware Routing in Cluster-Based Sensor Networks

Mohamed Younis*

Dept. of Computer Science and Elec. Eng.
University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250
younis@cs.umbc.edu

Moustafa Youssef

Dept. of Computer Science
Univ. of Maryland College Park
A.V. Williams Building
College Park, MD 20742
moustafa@cs.umd.edu

Khaled Arisha

Honeywell International Inc.,
Advanced Systems Tech. Group,
7000 Columbia Gateway Drive
Columbia, MD 21046
khaled.arisha@honeywell.com

Abstract

Recently there has been growing interest in the applications of sensor networks. Since sensors are generally constrained in on-board energy supply, efficient management of the network is crucial in extending the life of the sensor. In this paper we present a novel approach for energy-aware and context-aware routing of sensor data. The approach calls for network clustering and assigns a less-energy-constrained gateway node that acts as a centralized network manager. Based on energy usage at every sensor node and changes in the mission and the environment, the gateway sets routes for sensor data, monitors latency throughout the cluster, and arbitrates medium access among sensors. Simulation results demonstrate that substantial energy saving can be achieved using our approach.

Keywords: Network management, Routing algorithms, Power-Aware Communication, Sensor networks, Energy-efficient design.

1 Introduction

Networking unattended sensors is expected to have significant impact on the efficiency of many military and civil applications, such as combat field surveillance, security and disaster management. These systems process data gathered from multiple sensors to monitor events in an area of interest. Sensors in such systems are typically disposable and expected to last until their energy drains. Therefore, energy is a very scarce resource for such sensor systems and has to be managed wisely in order to extend the life of the sensors for the duration of a particular mission.

Sensors are generally equipped with data processing and communication capabilities. The sensing circuit measures parameters from the environment surrounding the sensor and transforms them into an electric signal. Processing such a signal reveals some properties about objects located and/or events happening in the

* Mohamed Younis is the contact author and is reachable by phone at (410) 455-3968 and by FAX at (410) 455-3969

vicinity of the sensor. The sensor sends such sensed data, usually via radio transmitter, to a command center either directly or through a data concentration center (a gateway). The gateway can perform fusion of the sensed data in order to filter out erroneous data and anomalies and to draw conclusions from the reported data over a period of time. For example, in a reconnaissance-oriented sensor network, sensor data indicates detection of a target while fusion of multiple sensor reports can be used for tracking and identifying the detected target [1].

Signal processing and communication activities are the main consumers of sensor's energy. Since sensors are battery-operated, keeping the sensor active all the time will limit the duration that the battery can last. Therefore, optimal organization and management of the sensor network is very crucial in order to perform the desired function with an acceptable level of quality and to maintain sufficient sensors' energy to last for the duration of the required mission. Mission-oriented organization of the sensor network enables the appropriate selection of only a subset of the sensors to be turned on and thus avoids wasting the energy of sensors that do not have to be involved. Energy-aware network management will ensure a desired level of performance for the data transfer while extending the life of the network.

Similar to other communication networks, scalability is one of the major design quality attributes. A single-gateway sensor network can cause the gateway to overload with the increase in sensors density, system missions and detected targets/events. Such overload might cause latency in communication and inadequate tracking of targets or a sequence of events. In addition, the single-gateway architecture is not scalable for a larger set of sensors covering a wider area of interest since the sensors are typically not capable of long-haul communication. To allow the system to cope with additional load and to be able to cover a large area of interest without degrading the service, network clustering is usually used by involving multiple gateways, as depicted in Fig.1. Given the constrained transmission range of the sensor and the need for conserving energy, the gateway needs to be located as close as possible to the sensors.

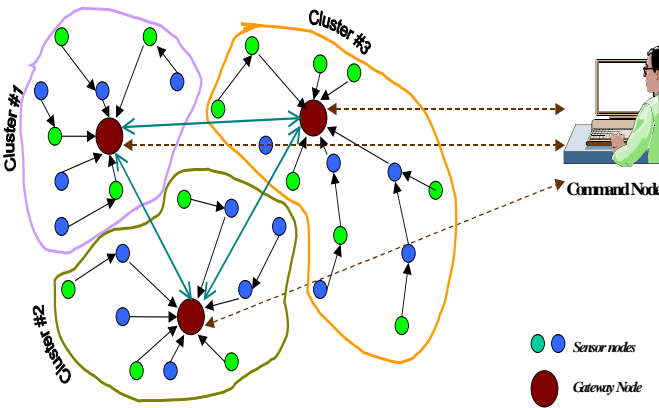


Fig. 1: Multi-gateway clustered sensor network

The multi-gateway architecture raises many interesting issues such as cluster formation, cluster-based sensor organization, network management, inter-gateway communication protocol and task allocation among the gateways. In this paper, we only focus on the issue of network management within the cluster, particularly

energy-aware routing. The gateway of the cluster will take charge of sensor organization and network management based on the mission and available energy in each sensor. Knowing which sensors need to be active in signal processing, we have developed algorithms to dynamically adapt the network topology within the cluster to minimize the energy consumed for communication, thus extending the life of the network while achieving acceptable performance for data transmission. We are not aware of any published work that considers sensor energy consumption related to both data processing and communication in the management of sensor networks.

In the balance of this section we define the architectural model and summarize the related work. The next section describes our approach to energy-aware routing in sensor networks. Description of the simulation environment and analysis of the experimental results can be found in section III. Finally section IV concludes the paper and discusses our future research plan.

1.1 System Model

The system architecture for the sensor network is depicted in Fig. 1. In the architecture sensor nodes are grouped into clusters controlled by a single command node. Sensors are only capable of radio-based short-haul communication and are responsible for probing the environment to detect a target/event. Every cluster has a gateway node that manages sensors in the cluster. Clusters can be formed based on many criteria such as communication range, number and type of sensors and geographical location [23][24]. In this paper, we assume that sensor and gateway nodes are stationary and the gateway node is located within the communication range of all the sensors of its cluster. Clustering the sensor network is performed by the command node and is beyond the scope of this paper. The command node will inform each gateway node of the ID and location of sensors allocated to the cluster. While the gateway can be a single point of failure within the cluster, we plan to extend our approach to support recovery from a gateway failure.

Sensors receive commands from and send readings to its gateway node, which processes these readings. Gateways can track events or targets using readings from sensors in any clusters as deemed by the command node. However, sensors that belong to a particular cluster are only accessible via the gateway of that cluster. Therefore, a gateway should be able to route sensor data to other gateways. Our focus in this paper is the management of the sensor network within a cluster while inter-gateway communication is not addressed.

Gateway nodes, which are significantly less energy-constrained than the sensors, interface the command node with the sensor network via long-haul communication links. The gateway node sends to the command node reports generated through fusion of sensor readings, e.g. tracks of detected targets. The command node

presents these reports to the user and performs system-level fusion of the collected reports for an overall situation awareness.

The sensor is assumed to be capable of operating in an active mode or a low-power stand-by mode. The sensing and processing circuits can be powered on and off. In addition both the radio transmitter and receiver can be independently turned on and off and the transmission power can be programmed based on the required range. It is also assumed that the sensor can act as a relay to forward data from another sensor. The on-board clocks of both the sensors and gateways are assumed to be synchronized, e.g. via the use of GPS¹. It is worth noting that most of these capabilities are available on some of the advanced sensors, e.g. the Acoustic Ballistic Module from SenTech Inc. [2].

1.2 Related Work

In wired networks, the emphasis has traditionally been on maximizing end-to-end throughput and minimizing delay. In general, paths are computed to minimize hop count or delay. While wireless networks inherited such design metrics from the wired counterparts energy constraints and signal interference have become central issues. Signal interference has received the most attention from the research community due to the growing popularity of wireless consumer devices. Only recently energy efficiency has started to receive attention, especially with the increasing interest in the applications of unattended sensor networks.

Although energy efficiency can be improved at various layers of the communication protocol stack, most published research has focused on hardware-related energy efficiency aspects of wireless communications. Low-power electronics, power-down modes, and energy efficient modulation are examples of work in this category [4]. However, due to fundamental physical limitations progress towards further energy efficiency is expected to become mostly architectural- and software-level issues. Given the scope of this paper, we focus on work related to network layer protocols.

Energy-aware routing has received attention in the recent few years, motivated by advances in wireless mobile devices. Since the overhead of maintaining the routing table for wireless mobile networks is very high, the stability of a route becomes of a major concern. Stable routes are reliable and long living [7]. Therefore, a stable route requires each mobile node involved to have enough power and to stay for the longest time within a reachable range of the next node on a link. Stability-based routing is different from ours since it is simply route-centric and does not consider network-wide metrics, as we do.

¹ While the GPS consumes significant energy, it has to be turned on for a very short duration during cluster formation. We use time-based approach for media access control that enables the maintenance of clock synchronization afterward.

A comparison between the direct routing protocol, in which each sensor sends its data directly to the gateway, and the minimum energy routing protocol that tries to minimize the transmission power is reported in [9]. The performance results show that a routing protocol that considers only the energy of the transmitter and ignores the energy dissipation of the receiver in determining the routes can perform worse than direct communication. Such results further justify our approach. We consider the energy that the sensor possesses and consumes in data processing in the routing decision. In addition, we use time-based approach for media access control (MAC) to further limit the energy consumed by the receiver. The MAC approach is briefly discussed in section II.

The effectiveness of three power-aware routing algorithms: Minimum total Transmission Power, Min-Max Battery Cost, and Max-Min Battery Capacity, is compared in [7]. The results pointed out that the battery power capacity, the transmission power, and the stability of routes are among the issues to be considered in designing a power efficient routing protocol. Similar conclusions were drawn in [11]. The reported results have indicated that in order to maximize the lifetime, the traffic should be routed such that the energy consumption is balanced among the nodes in proportion to their energy reserves. Our algorithm balances these considerations with other metrics such as end-to-end delay. In addition, we consider the sensor role in an application mission in the routing decision.

A position-based protocol to setup and maintain a minimum energy network between randomly deployed nodes is presented in [10]. The algorithm fits networks that require strong connectivity among all nodes, a feature that is not required in our model. In addition end-to-end delay appears to be ignored in the algorithm while allowing unlimited levels of relays. Although our approach also uses positions to calculate different cost metrics such as transmission energy and propagation delay, we try to balance different performance metrics instead of getting the minimum energy route sacrificing other performance metrics.

Achieving energy saving through activation of a limited subset of nodes in an ad-hoc wireless networks has been the goal of some recent research such as SPAN [26], GAF [27] and ASCENT [28]. Both SPAN and GAF are distributed approaches that require nodes in close proximity to arbitrate and activate the least number of nodes needed to ensure connectivity. Nodes that are not activated are allowed to switch to a low energy sleep mode. While GAF uses nodes' geographical location to form grid-based cluster of nodes, SPAN relies on local coordination among neighbors. In ASCENT, the decision for being active is the courtesy of the node. Passive nodes keep listening all the time and assess their course of actions; stay passive or become active. In our approach node's state is determined at the gateway while considering the signal processing duties in the sensor's state transition.

2 Energy-Conscious Message Routing

In this section we discuss a novel approach for managing the sensor network with a main objective of extending the life of the sensors in a particular cluster. We mainly focus on the topology adjustment and the message routing. Sensor energy is central in deciding on changes to the networking topology and in setting routes. Messages are routed through multiple hops to conserve the transmission energy of the sensors. Latency in data delivery and other performance attributes are also considered in the routing decision. In addition, message traffic between the sensors and the gateway is arbitrated in time to avoid collision and to allow turning off the sensor radio when not needed.

Setting routes for sensor data can be performed in a central node that knows the network topology, e.g. the gateway, or distributed among the sensors themselves. Both centralized and distributed routing requires maintenance of the routing table every time the network topology changes. While distributed approaches are scalable for larger networks, updating routing tables and ensuring consistency among the local versions that the sensor nodes have consumed significant computation and communication resources, thus limiting the portion of the already limited sensor energy that can be dedicated to serve the application [13]. In addition, exchanging routing messages among the sensors will create excessive traffic that drains energy since radio receivers on the sensors may overhear routing message transmissions not destined to them.

On the other hand, centralized routing is simple and fits the nature of the sensor networks. Since the sensor is committed to data processing and communication, it is advantageous to offload routing decision from the resource-constrained sensor nodes. In addition, since the gateway has a cluster-wide view of the network, the routing decisions should be simpler and more efficient than the decisions based on local views at the sensor level. Given that the gateway organizes the sensor in the cluster, it can combine the consideration for energy commitments to data processing, remaining sensor energy, sensor location, link traffic and acceptable latency in receiving the data in efficiently setting message routes. Moreover, knowledge of cluster-wide sensor status enhances the robustness and effectiveness of media access control because the decision to turn a node receiver off will be more accurate and deterministic than a decision based on a local MAC protocol [6]. Although centralized routing can restrict scalability as the number of sensors per cluster increases, more gateways can be deployed. The system architecture promotes the idea of clustering to ensure scalability. Cluster formation approaches, which are not addressed in this paper, account for resource requirements at the gateway node to cope with the responsibility of managing the assigned sensors. Dependability issues related to the centralized network control can be addressed by fault-tolerance techniques [12].

2.1 Sensor Network State

In the system architecture, gateway nodes assume responsibility for sensor organization based on missions that are assigned to every cluster. Thus the gateway will control the configuration of the data processing circuitry of each sensor within the cluster. Assigning the responsibility of network management within the cluster to the gateway can increase the efficiency of the usage of the sensor resources. The gateway node can apply energy-aware metrics to the network management guided by the sensor participation in current missions and its available energy. Since the gateway sends configuration commands to sensors, the gateway has the responsibility of managing transmission time and establishing routes for the outgoing messages. Therefore, managing the network topology for message traffic from the sensors can be seen as a logical extension to the gateway role, especially all sensor readings have to be forwarded to the gateway for fusion and application-specific processing.

The nodes in a cluster can be in one of four main states: sensing only, relaying only, sensing-relaying, and inactive. In the *sensing state*, the node sensing circuitry is on and it sends data to the gateway in a constant rate. In the *relaying state*, the node does not sense the target but its communications circuitry is on to relay the data from other active nodes. When a node is both sensing the target and relaying messages from other nodes, it is considered in the *sensing-relaying state*. Otherwise, the node is considered *inactive* and can turn off its sensing and communication circuitry. The decision for determining the node's state is done at the gateway based on the current sensor organization, node battery levels, and desired network performance measures. We are not aware on any published work that considers sensor energy consumption related to both data processing and communication in the management of sensor networks. It should be noted that our approach is transparent to the method of selecting the nodes that should sense the environment.

In a cluster the gateway will use model-based energy consumption for the data processor, radio transmitter and receiver to track the life of the sensor battery. This model is used in the routing algorithm as explained later. The gateway updates the sensor energy model with each packet received by changing the remaining battery capacity for the nodes along the path from the source sensor node to the gateway. Fig. 2 shows an example for energy model update.

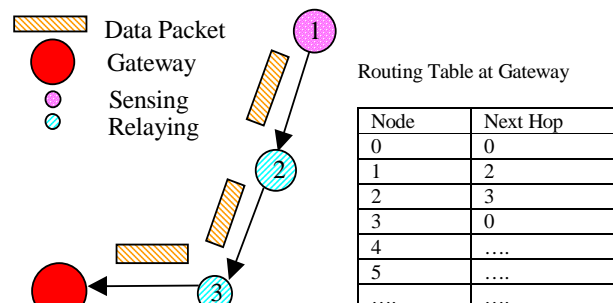


Fig. 2: When the gateway receives a packet from node1, it uses the routing table to update the energy model of nodes 1, 2, and 3, which are on the path from node1 to the gateway

The typical operation of the network consists of two alternating cycles: data cycle and routing cycle. During the data cycle, the nodes, which are sensing the environment sends their data to the gateway. During the routing cycle, the state of each node in the network is determined by the gateway and the nodes are then informed about their newly assigned states and how to route the data. Fig. 3 shows an example of the state of sensors and routes within a typical cluster in a sensor network during a target-tracking mission.

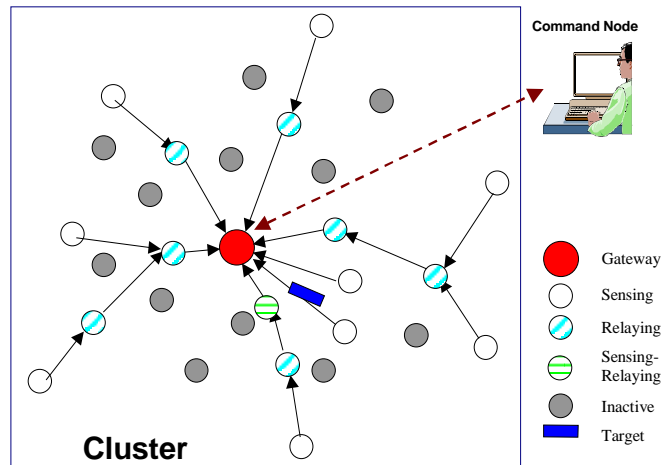


Fig. 3: A Typical Cluster in a Sensor Network

The energy model may deviate from the actual node battery level due to inaccuracy in the model or packet drop caused by either a communication error or a buffer overflow at a node. This deviation may negatively affect the quality of the routing decisions. To compensate for this deviation, the nodes refresh their energy model at the gateway periodically with a low frequency. All nodes, including inactive nodes, send their refresh packets at a pre-specified time directly to the gateway and then turn their receivers on at a predetermined time in order to hear the gateway routing decision. This requires the nodes and gateway to be synchronized as assumed earlier.

If a node's refresh packet is dropped due to communication error, the gateway assumes that the node is nonfunctioning during the next cycle, which leads to turning this node off. However, this situation can be corrected in the next refresh. On the other hand, if a routing decision packet from the gateway to a node is dropped, we have two alternatives:

- The node can turn itself off. This has the advantage of reducing collisions but may lead to loss of data packet if the node should be in the sensing or relaying state. Missing sensor data might be a problem unless tolerated via the selection of redundant sensors and/or the use of special data fusion techniques.
- The node can maintain its previous state. This can preserve the data packets especially if the node new state happens to be the same as its old state. However, if this is not the case, the probability of this node transmission colliding with other nodes' transmissions increases.

We choose to implement the second alternative since it is highly probable for a node to maintain its previous state during two consecutive routing phases. In addition losing data packet may negatively affect the application, e.g. losing track of a target. Using clever MAC protocols, as explained later, can reduce the

probability of collision. A description of the energy model we used in the simulation can be found in Appendix A.

2.2 Routing Approach

Since we have chosen a centralized approach for network management, source routing methodologies can be followed [3]. Although source routing is simple to implement and generates loop-free routes, it requires maintenance of a cluster-wide state that includes all the parameters affecting the routing decision. In our case these parameters are sensor's state, location, remaining energy and message traffic. There is some inaccuracy in the gateway energy model due to the overhead, packet dropping and propagation delays of refresh messages. The model approximation is still accepted since we believe that frequent refreshing, together with fine-tuning of routing parameters, can adjust any deviation to be within tolerable limits.

Because the gateway is not as energy-constrained as the sensors, it is better for the gateway to send commands to the sensors directly without involving relays. Therefore, our problem becomes limited to routing sensor data to the gateway and thus can be reduced to a single-sink unicast routing problem from the sensors to the gateway. Our approach is to use the transpose of a single-source routing algorithm, i.e. single destination routing. This can reduce the complexity of the problem to become solvable using a least-cost or shortest-path unicast routing algorithm.

To model the sensor network within the cluster, we assume that nodes, sensors and gateway, are connected by bi-directional wireless links with a cost associated with each direction. Each link may have a different cost for each direction due to different energy levels of the nodes at each end. The cost of a path between two nodes is defined as the sum of the costs of the links traversed. For each sensing-enabled node, the routing algorithm should find a least-cost path from this node to the gateway. The routing algorithm can find the shortest path from the gateway to the sensing-enabled nodes using the transpose property.

To account for energy conservation, delay optimization and other performance metrics, we define the following cost function for a link between nodes i and j :

$$\sum_{k=0}^7 CF_k = c_0 \times (distance_{ij})^l + c_1 \times f(energy_j) + c_2 / T_j + c_3 + c_4 + c_5 + c_6 \times distance_{ij} + c_7 \times overall\ load$$

- Where:
- $distance_{ij}$: Distance between the nodes i and j
 - $energy_j$: Current energy of each node j
 - CF_k are cost factors defined as follows:

- CF_0 : Communication cost = $c_0 \times (distance_{ij})^l$, where c_0 is a weighting constant and the parameter l depends on the environment, and typically equals to 2. This factor reflects the cost of the wireless transmission power, which is directly proportional to the distance raised to some power l . The closer a node to the destination, the less its cost factor CF_0 and the more attractive it is for routing.
- CF_1 : Energy stock = $c_1 \times f(energy_j)$ for node j . This cost factor favors nodes with more energy. The more energy the node contains, the better it is for routing. Nodes with abundant energy are expected to last long and, thus increase the reliability to the paths selected for routing. The function ' f ' is chosen to reflect the battery remaining lifetime.
- CF_2 : Energy consumption rate = c_2 / T_j , where c_2 is a weighting constant and T_j is the expected time under the current consumption rate until the node j energy level hits the minimum acceptable threshold. CF_2 makes the heavily used nodes less attractive, even if they have a lot of energy.
- CF_3 : Relay enabling cost = c_3 , where c_3 is a constant reflecting the overhead required to switch an inactive node to become a relay. This factor makes the algorithm favor the relaying-enabled nodes to be used for routing rather than the inactive nodes.
- CF_4 : Sensing-state cost = c_4 , where c_4 is a constant added when the node j is in a sensing-state. This factor does not favor selecting sensing-enabled nodes to serve as relays, since these nodes have already committed some energy for data processing. It is preferred not to overload these nodes in order to keep functioning as long as possible.
- CF_5 : Maximum connections per relay: once this threshold is reached, we add an extra cost c_5 to avoid setting additional paths through that relay. This factor extends the life of overloaded relay nodes by making them less favorable. Since these relay nodes are already critical by being on more than one path, the reliability of paths through these nodes increases.
- CF_6 : Propagation delay = $c_6 \times distance_{ij}$, where c_6 is the result of dividing a weighting constant by the speed of wireless transmission. The farther the node from the gateway, the longer it takes to deliver a message. Thus, this factor favors closer nodes.
- CF_7 : Queuing Cost = $c_7 \times \lambda / \mu$, where $\lambda = \sum \lambda_s$ for each sensor node s whose route passes through the node j , λ_s is data-sensing rate for node s and μ is the service rate (mainly store-and-forward process). The expression λ / μ is the average queue length for the $M/M/1$ queuing model. This factor can be mathematically simplified to be the overall load to the relay node, where the overall load is the total

number of sensing-enabled nodes whose data messages are sent via routes through this node to the gateway. Assume equal service rate μ for each relay as well as equal data-sensing rate λ_s for each sensing-enabled node. With this assumption, the constant μ can be reduced inside c_7 , and λ can be reduced to the overall load times the constant data sensing rate for each sensor λ_s . Thus $CF_7 = c_7 \times \text{overall load}$. This factor does not favor relays with long queues to avoid dropping or delaying data packets.

It should be noted that some of the CF_i 's factors are conflicting. For example, in order to minimize the transmission power, we need to use multiple short distances leading to more number of hops and thus increasing the delay. The routing algorithm is to balance among these factors. The weighting constants c_i 's are system-defined based on the current mission of the network. The values of the cost factors CF_1, CF_2 , and CF_7 for the gateway node are set to zero since the gateway is not energy-constrained.

Solving the above model is a typical path-optimization routing problem. This problem is proved to have a polynomial complexity [13]. Path-optimization problems are usually solved using a shortest path (least-cost) algorithm [15]. Shortest paths from one (source) node to all other nodes on a network are normally referred to as *one-to-all* shortest paths. Shortest paths from one node to a subset of the nodes is defined as *one-to-some* shortest paths, while those paths from every node to every node is called *all-to-all* shortest paths [14]. Our routing problem can be considered as the transpose of the one-to-some shortest path, since not all sensors are active simultaneously. A recent study by Zhan and Noon [16] suggested that the best approach for solving the one-to-some shortest path is Dijkstra's algorithm. In addition, Dijkstra's algorithm is shown to be more suited for centralized routing [3]. Therefore, we use Dijkstra's algorithm with the link cost d_{ij} for the link between the nodes i and j , redefined as $d_{ij} = \sum_k CF_k$, as explained in the model.

One of the nice features of our approach is that the routing setup can be dynamically adjusted to optimally respond to changes in the sensor organization. For a target-tracking sensor network, the selected sensors vary as the target moves. The routing algorithm has to accommodate changes in the selection of active sensors in order to ensure the delivery of sensors data and the proper tracking of the target. In addition, the gateway will continuously monitor the available energy level at every sensor that is active in data processing, sensing, or in forwarding data packets, relaying. Rerouting can also occur after receiving an updated status from the sensors. Changes to the energy model might affect the optimality of the current routes, and thus new routes have to be generated.

As mentioned before, all nodes turn their receiver on at a predetermined time in order to hear the gateway routing decision and their local routing table, if the node new state is relaying. This means that all rerouting should be done at the same predetermined time. The refresh cycle should be performed at a low frequency in

order to conserve the sensor energy, especially as the refresh packets are transmitted directly from all sensors to the gateway without passing relays.

2.3 MAC Layer Protocol

Careful choice of the MAC layer protocol can leverage the effectiveness of the new routing approach. Recent research results pointed out that the wireless network interface consumes a significant fraction of the total energy although the interface is actually receiving packets during a small fraction of the time [5]. Therefore, in a wireless system the medium access protocols can be adapted and tuned to enhance energy efficiency.

We choose to implement a time division multiple access (TDMA) based MAC layer whose slot assignment is managed by the gateway. The gateway informs each node about slots in which it should listen to other nodes' transmission and slots, which the node can use for its own transmission. The TDMA MAC layer provides two features that are advantageous to our approach. First, clock synchronization is built in the TDMA protocol. Second, collision among the nodes can be avoided with assigning non-overlapping time slots.

To set the routes, the gateway sends to sensing nodes the transmission range to cover so that data can reach the next relay node on the route. In addition, the gateway sends relay nodes a forwarding table. The forwarding table consists of ordered tuple of the form: (time slot, data-originating node, transmission range). The "time slot" entry specifies when to turn the receiver on in order to listen for an incoming packet. The "source node" is the sensor node that originated this data packet, and "transmission range" specifies the transmission power to use in re-sending the data. This transmission power should be enough to reach the next relay on the path from the originating node to the gateway. It should be noted that the intermediate nodes on the data routes are not specified. Thus it is sufficient for the relaying nodes to know only about the data-originating node. The transmission range ensures that the next relay node, which is also told to forward that data packet, can clearly receive the data packet and so on. Such approach significantly simplifies the implementation since the routing table size will be very small and the changes to the routes will be quicker to communicate with sensors. Such simplicity is highly desirable to fit the limited computational resources that sensors would have.

The protocol consists of four main phases: data transfer, refresh, event-triggered rerouting, and refresh-based rerouting phase. In the data transfer phase, sensors send their data in the time slots allocated to them. Relays use their forwarding tables to forward this data to the gateway. Inactive sensor nodes remain off until the time for sending a status update or to receive route broadcast messages. The refresh phase is designated for updating the sensor model at the gateway. This phase is periodic and occurs after multiple data transfer

phases. Periodic adjustments to sensor status enhance the quality of the routing decisions and correct any inaccuracy in the assumed sensor models. During the refresh phase, each node in the network uses its pre-assigned time slot to inform the gateway of its state (energy level, state, position, etc). Any node that does not send information during this phase is assumed to be nonfunctioning. If the node is still functioning and a communication error caused its packet to be lost, its state may be corrected in the next refresh phase.

This section only provides a brief overview of some of features of the MAC protocol referring the reader to [25] for detailed discussion. It is worth noting that simulation experiments demonstrated the effectiveness of our MAC approach. The results indicate that the MAC protocol can increase the life of the network by an order of magnitude when combined with our routing approach. The next section describes performance evaluation of the energy-aware routing approach via simulation.

3 Experimental Validation

The effectiveness of the routing approach is validated through simulation. This section describes the performance metrics, simulation environment, and experimental results. The performance results are also compared with other routing approaches.

3.1 Performance Metrics

We used the following metrics to capture the performance of our routing approach and to compare it with other algorithms:

- *Time to network partition*: When the first node runs out of energy, the network within the cluster is said to be partitioned [8]. The name network partitioning reflects the fact that some routes become invalid and cluster-wide rerouting may be imminent.
- *Time for last node to die*: This metric, along with the time to network partition metric, gives an indication of network lifetime.
- *Average and standard deviation of node lifetime*: This also gives a good measure of the network lifetime. A routing algorithm, which minimizes the standard deviation of node life, is predictable and thus desirable.
- *Average delay per packet*: Defined as the average time a packet takes from a sensor node to the gateway. Most energy aware routing algorithms try to minimize the consumed energy. However, some sensor network missions are delay sensitive, so this metric is important.
- *Network Throughput*: Defined as the total number of data packets received at the gateway divided by the simulation time.

- *Average energy consumed per packet*: This metric represents the average energy consumed in transmitting, and receiving a data packet. A routing algorithm that minimizes the energy consumed per packet will, in general, yields better energy savings.

3.2 Environment Setup

In the experiments the cluster consists of 100 randomly placed nodes in a 1000×1000 meter square area. The gateway position is determined randomly within the cluster boundaries. A free space propagation channel model is assumed [17] with the capacity set to 2Mbps. Packet lengths are 10 Kbit for data packets and 2 Kbit for routing and refresh packets. Each node is assumed to have an initial energy of 5 joules and a buffer for up to 15 packets [22]. A node is considered non-functional if its energy level reaches 0. For the term CF_l in the cost function, we used the linear discharge curve of the alkaline battery [8].

For a node in the sensing state, packets are generated at a constant rate of 1 packet/sec. This value is consistent with the specifications of the Acoustic Ballistic Module from SenTech Inc. [2]. Each data packet is time-stamped when it is generated to allow the calculation of average delay per packet. In addition, each packet has an energy field that is updated during the packet transmission to calculate the average energy per packet. A packet drop probability is taken equal to 0.01. This is used to make the simulator more realistic and to simulate the deviation of the gateway energy model from the actual energy model of nodes.

We assume that the cluster is tasked with a target-tracking mission in the experiment. The initial set of sensing nodes is chosen to be the nodes on the convex hull of sensors in the cluster. The set of sensing nodes changes as the target moves. Since targets are assumed to come from outside the cluster, the sensing circuitry of all boundary nodes is always turned on. The sensing circuitry of other nodes are usually turned off but can be turned on according to the target movement. Targets are assumed to start at a random position outside the convex hull. We experimented with different types of targets but for this paper we choose the linearly moving targets. These targets are characterized by having a constant speed chosen uniformly from the range 4 meters/s to 6 meters/s and a constant direction chosen uniformly depending on the initial target position in order for the target to cross the convex hull region. It is assumed that only one target is active at a time. This target remains active until it leaves the deployment region area. In this case, a new target is generated.

Rerouting occurs when the node energy level falls below a percentage of its initial energy. This percentage is taken equal to 80%. Each time this threshold is reached, it is reset to 0.8 of its previous value. The sensor energy-consumption model and the values of the parameters can be found in Appendix A. For each experiment, we ran the simulator for different network topologies and calculated the above measures. We also calculate the standard deviation for average measures.

3.3 Performance Results

In this section we present some results obtained by simulation. For the purpose of our simulation experiments the values for the parameters $\{c_i\}$ are initially picked based on sub-optimal heuristics for best possible performance. The reported performance results are based on about 5000 sensors' data packets. Unless mentioned otherwise, a refresh phase is scheduled periodically every 20 data phases.

Comparison between routing algorithms

We ran a set of experiments to compare the performance of our approach with other routing algorithms. The results are shown in figures 4 through 7. The figures show that the new algorithm gives a relatively good performance for all the metrics. Other algorithms may slightly outperform our algorithm for some metrics; however, the same algorithms perform poorly on other metrics. For example, the minimum distance routing algorithm gives a 1.57 improvement factor over the new algorithm in terms of average delay per packet (Fig 7.). However, our algorithm outperforms this algorithm by a factor of 13.91 in terms of time to network partitioning, as indicated in Fig. 4.

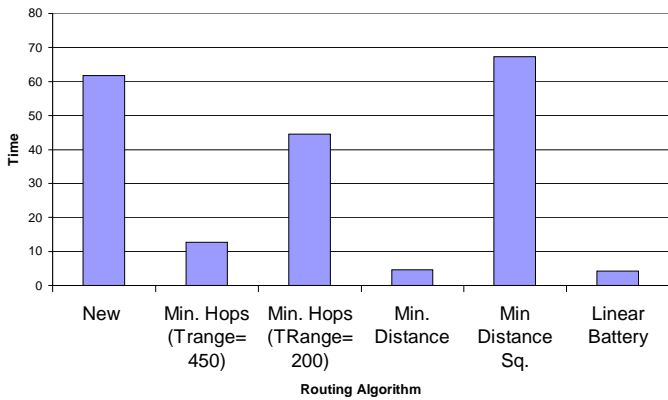


Fig. 4: Time to network partition for routing algorithms

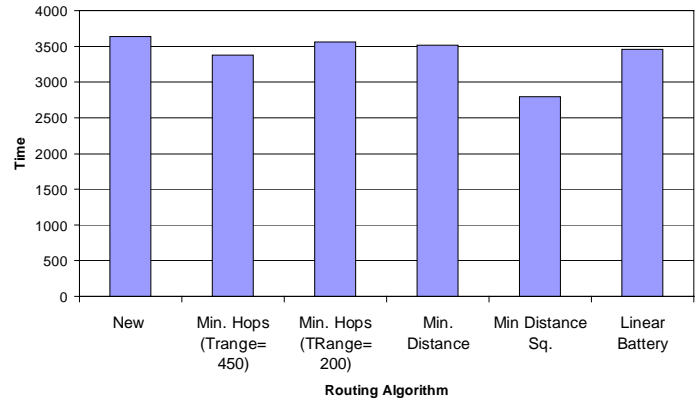


Fig. 5: Time for last node to die under various algorithms

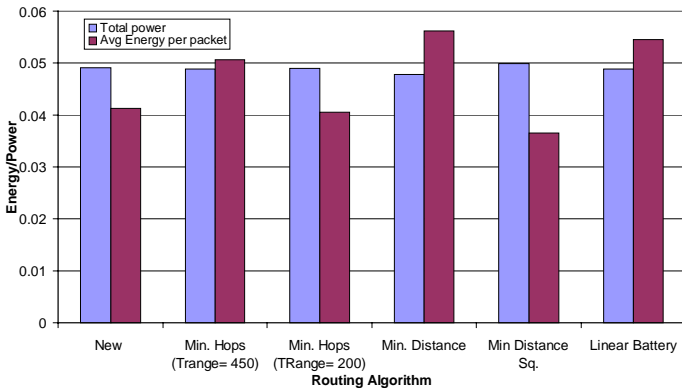


Fig. 6: Comparing energy metrics among routing algorithms

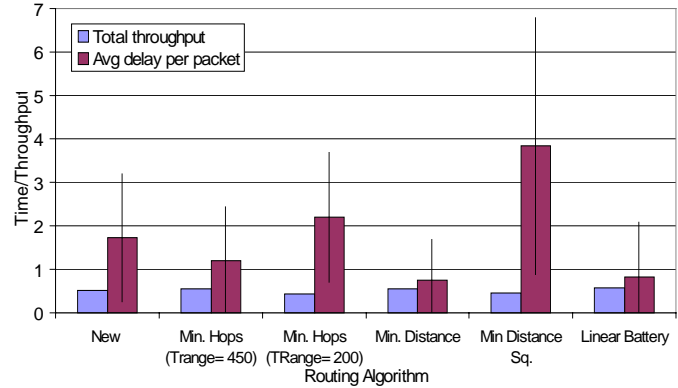


Fig. 7: Throughput and delay for routing algorithms

The minimum distance algorithm gives the best average delay per packet, as indicated in Fig. 7. This is expected as all packets take only one hop. Meanwhile, the minimum distance squared routing algorithm gives the worst average delay per package as it tries to minimize the transmission power by taking short distance and large number of hops. This makes packets visit multiple nodes incurring more transmission and queuing delay. The opposite reasoning applies to the time to network partition. In that case, the best algorithm is the minimum distance squared and the worst is the minimum distance.

Although the minimum distance square leads the way in terms of average energy per packet and time to network partition, figures 4 and 6., it offers the worst average delay per packet as depicted in Fig. 7. Our algorithms lead to the best time for the last node to die, as indicated in Fig. 5. This is expected as the new algorithm balances the factors affecting the lifetime of the node leading to increased network lifetime.

As demonstrated by the experimental results, different routing algorithms can enhance one performance metric while worsening another. Therefore choosing the routing approach is greatly influenced by the performance qualification metrics, which are highly dependent on the nature of application. For example, if data latency and packet loss are not issues of concern while energy conservation is of great interest, one might pick distance squared routing. However, we believe that we are handling the major issues for applications of sensor networks in military and disaster management applications. Such applications are very dynamic and long lasting and thus require energy-efficient, robust and responsive network operation.

Effect of packet drop probability on performance

To study the effect of packet drop probability on performance, we varied the probability of packet drop from 0.1 to 0.5. Figures 8 and 9 show the results. The average delay per packet decreases. This can be explained by noting that as the number of hops the packet traverse increases, the probability that it will be dropped increases. This means that the packets that arrive to the gateway are most probable to take a small number of hops and thus incurring less delay. As expected, the throughput decreases due to lost packets. The average node lifetime increases since not all packets reach their destination and thus the node energy is conserved.

Effect of refreshing rate on model accuracy

To study the effect of refreshing rate on the accuracy of the gateway's energy model, we varied the refresh phase frequency from 20 data cycles to 160 data cycles. Figures 10 shows the result. As the period between energy model refreshes is increased, the gateway energy model deviates more from the node actual energy level leading to more root mean square error in the energy model as shown. The effect of the energy model accuracy on performance is discussed next.

Effect of energy model accuracy on performance

For this experiment, we introduce a percentage error in the energy model. This percentage error is taken to be a uniform random variable between 0 and 100% (1.0 in figures 11-13). In this experiment, the energy model was taken to underestimate the actual node energy. Figures 11 through 13 shows the results. The

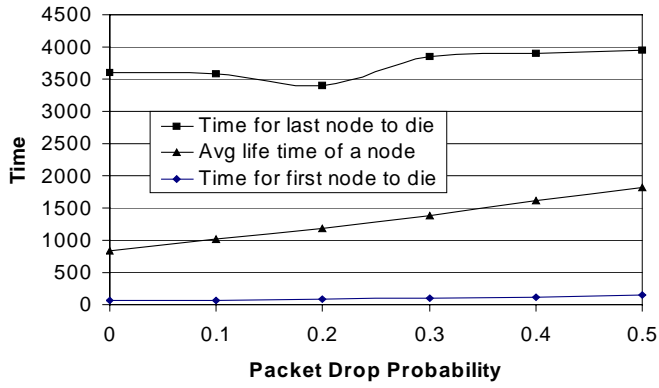


Fig. 8: Effect of packet-drop probability on lifetime Metrics

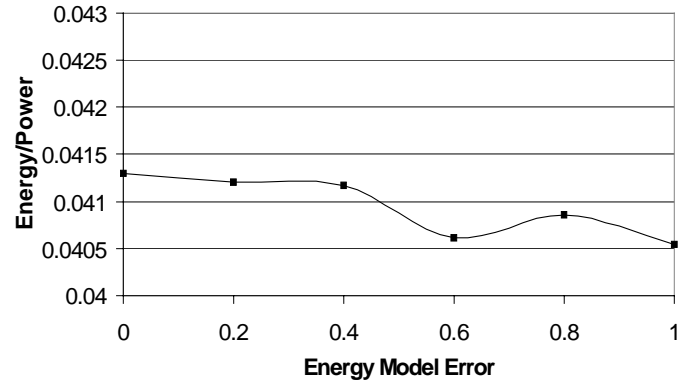


Fig. 11: Effect of energy model accuracy on power metrics

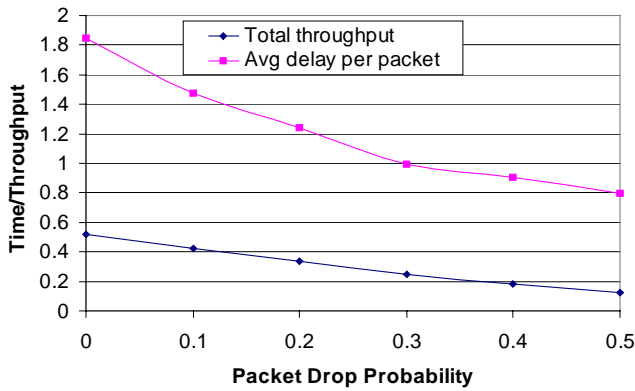


Fig. 9: Effect of packet-drop on Throughput and delay

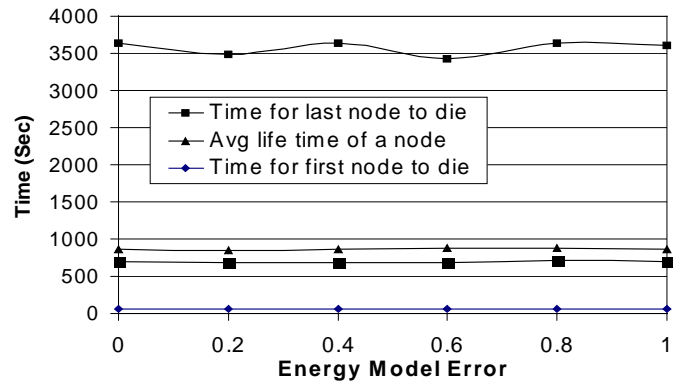


Fig. 12: Effect of energy model accuracy on network lifetime

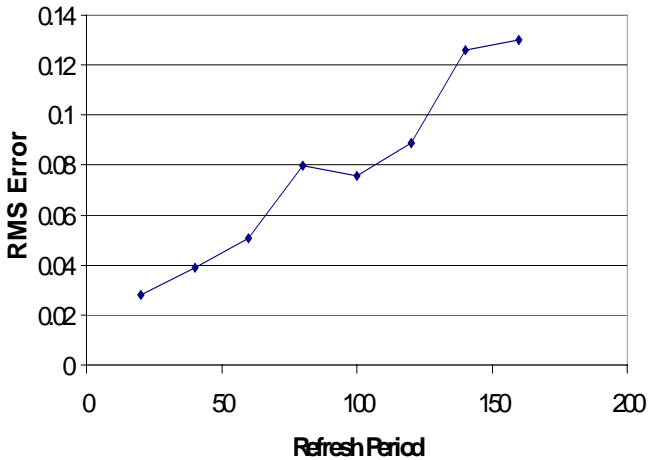


Fig. 10: Effect of refresh rate on energy model

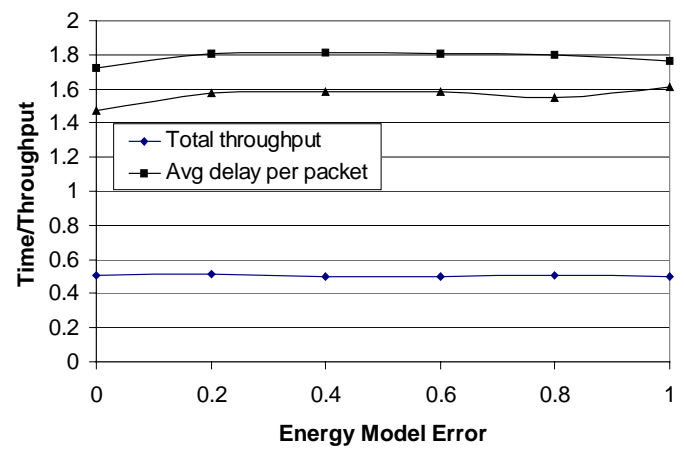


Fig. 13: Effect of model accuracy on throughput and delay

results indicate that the performance is not sensitive to the model accuracy. This is because the refresh phase corrects the data model before it deviates too much from the node actual energy level. We also ran an experiment to study the effect of the energy model overestimating the node energy level. Similar results were obtained and are not included for space constraints.

4 Conclusion and Future Work

In this paper, we introduced a novel energy-aware routing approach for sensor networks. A gateway node acts as a cluster-based centralized network manager that sets routes for sensor data, monitors latency throughout the cluster, and arbitrates medium access among sensors. The gateway tracks energy usage at every sensor node and changes in the mission and the environment. The gateway configures the sensors and the network to operate efficiently in order to extend the life of the network. Simulation results demonstrate that our algorithm consistently performs well with respect to both energy-based metrics, e.g. network lifetime, as well as contemporary metrics, e.g. throughput and end-to-end delay. Although we rely on model of energy usage at the sensor nodes, simulation experiments show that the accuracy of the model has little effect on performance with infrequent periodic model adjustment.

Our future plan includes extending the routing model to handle performance constraints and to allow for node mobility. We would like also to study network clustering approaches, inter-cluster interaction and operations, the management of resources at the cluster level and the handling of sensor or gateway failure.

References

- [1] R. Burne, et. al, "A Self-Organizing, Cooperative UGS Network for Target Tracking," *in the Proceedings of the SPIE Conference on Unattended Ground Sensor Technologies and Applications II*, Orlando Florida, April 2000.
- [2] "Data sheet for the Acoustic Ballistic Module", SenTech Inc., <http://www.sentech-acoustic.com/>
- [3] W. Stallings, *Data and Computer Communications*, Macmillan Publishing Company, 3rd edition, 1991.
- [4] P.J.M. Havinga and G.J.M. Smit, "Design Techniques for Low Power Systems", *Journal of Systems Architecture*, Vol. 46, No. 1, 2000.
- [5] P. Havinga, G. Smit, M. Bos, "Energy efficient adaptive wireless network design", *The Fifth Symposium on Computers and Communications (ISCC'00)*, Antibes, France, July 2000.
- [6] S. Singh and C.S. Raghavendra, "PAMAS: Power Aware Multi-Access protocol with Signaling for Ad Hoc Networks", *ACM Computer Communications Review*, July 1998.
- [7] C-K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks," *IEEE Communications Magazine*, June 2001.
- [8] S. Singh, M. Woo and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", *ACM MOBICOM'98*, Dallas, Texas, October 1998.
- [9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *Hawaii International Conference on System Sciences (HICSS '00)*, January 2000.

- [10] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks", *IEEE Jour. Selected Areas Comm.*, Vol. 17, No. 8, pp. 1333-1344, August 1999.
- [11] J.-H. Chang, L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks", *IEEE Infocom*, 2000.
- [12] D. Pradhan, *Fault-Tolerant Computer System Design*, Prentice Hall, New Jersey, 1996.
- [13] S. Chen, "Routing Support for providing guaranteed End-To-End Quality of Service," *Ph.D. Thesis Dissertation*, University of Illinois at Urbana-Champaign, 1999.
- [14] F. Zhan, "Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures," *Journal of Geographic Information and Decision Analysis*, Vol.1, No.1, pp. 69-82, 1998.
- [15] M. H. Hung and J. J. Divoky, "A Computational Study of Efficient Shortest Path Algorithms," *Computers & Operations Research*, Vol. 15, pp. 567-576, 1988.
- [16] F. B. Zhan, and C. E. Noon, "Shortest Path Algorithms: An Evaluation Using Real Road Networks," *Transportation Science*, 1996.
- [17] J.B. Andresen, T.S. Rappaport, and S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," In *IEEE Communications Magazine*, Vol. 33, No. 1, January 1995.
- [18] M. Bhardwaj, et. al, "Upper Bounds on the Lifetime of Sensor Networks", In *Proceedings of ICC 2001*, June 2001. (12)
- [19] R. Min, et. al, "An Architecture for a Power-Aware Distributed Microsensor Node", *IEEE Workshop on Signal Processing Systems (SiPS '00)*, October 2000.
- [20] W. Heintzelman, et. al, "Energy-Scalable Algorithms and Protocols for Wireless Microsensor Networks," *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00)*, June 2000.
- [21] A. Sinha and A. Chandrakasan, "Energy Aware Software", *Proceedings of the 13th International Conference on VLSI Design*, pp. 50-55, Calcutta, India. January 2000.
- [22] M. Gerla, G. Pei, and S.-J. Lee, "Wireless, Mobile Ad-Hoc Network Routing," *IEEE/ACM FOCUS'99*, New Brunswick, NJ, May 1999.
- [23] A. Buczak and V. Jamalabad, "Self-organization of a Heterogeneous Sensor Network by Genetic Algorithms," *Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et. a.. (eds.), Vol. 8, pp. 259-264, ASME Press, New York, 1998.
- [24] Chunhung Richard Lin and Mario Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, Vol. 15, No. 7, September 1997.
- [25] K. Arisha, M. Youssef, M. Younis, "Energy-Aware TDMA-Based MAC for Sensor Networks," *Proceedings of the IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002)*, New York City, New York, May 2002.
- [26] B. Chen, et al., "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", in the *Proceedings of the 7th ACM Mobile Computing and Communication (MobiCom 2001)*, Rome, Italy, July 2001.
- [27] Y. Xu and J. Heidemann and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," in the *Proceedings of the 7th ACM Mobile Computing and Communication (MobiCom 2001)*, Rome, Italy, July 2001.
- [28] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," In the *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 23-27 2002.

Appendix A: Sensor's Energy Consumption Model

Since the presented routing algorithm uses model-based sensor energy consumption, it is important to use a fairly accurate model to ensure the effectiveness of the approach and minimize the correction made to the gateway-view of the sensor-energy level during the refresh phase. A typical sensor node consists mainly of a sensing circuit for signal conditioning and conversion, digital signal processor, and radio links [18][19]. The following summarizes the energy-consumption models for each sensor component.

Communication Energy Dissipation

We use the model of [9][18][20]. The key energy parameters for communication in this model are the energy/bit consumed by the transmitter electronics (α_{11}), energy dissipated in the transmit op-amp (α_2), and energy/bit consumed by the receiver electronics (α_{12}). Assuming a $1/d^n$ path loss, the energy consumed is:

$$E_{tx} = (\alpha_{11} + \alpha_2 d^n) * r \quad \text{and} \quad E_{rx} = \alpha_{12} * r$$

Where E_{tx} is the energy to send r bits and E_r is the energy consumed to receive r bits. Table 1 summarizes the meaning of each term and its typical value.

Table 1: Parameters for the communication energy model

Term	Meaning
α_{11}, α_{12}	Energy dissipated in transmitter and receiver electronics per bit (Taken to be 50 nJ/bit).
α_2	Energy dissipated in transmitter amplifier (Taken = 100 pJ/bit/m ²).
r	Number of bits in the message.
d	Distance that the message traverses.

Computation Energy Dissipation

We assume the leakage current model of [19][20][21]. The model depends on the total capacitance switched by the program and the number of cycles the program takes. We used parameter values similar to those in [21].

Sensing Energy Dissipation

We assume that the energy needed to sense one bit is a constant (α_3) so that the total energy dissipated in sensing r bits is [18]: $E_{sensing} = \alpha_3 * r$

For the Ballistic Audio Module sensor [2], the energy dissipated for sensing a bit is approximately equal to the energy dissipated in receiving a bit. Therefore, the parameter α_3 is taken equal to α_{12} .