

Energy-Aware Management for 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

Networking unattended sensors is expected to have a significant impact on the efficiency of many military and civil applications. 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. In this paper, we present a novel approach for energy-aware management of sensor networks that maximizes the lifetime of the sensors while achieving acceptable performance for sensed data delivery. The approach is to dynamically set routes and arbitrate medium access in order to minimize energy consumption and maximize sensor life. The approach calls for network clustering and assigns a less-energy-constrained gateway node that acts as a cluster 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. We also describe a time-based Medium Access Control (MAC) protocol and discuss algorithms for assigning time slots for the communicating sensor nodes. Simulation results show an order of magnitude enhancement in the time to network partitioning, 11% enhancement in network lifetime predictability, and 14% enhancement in average energy consumed per packet.

Keywords: Sensor networks Energy-aware network management, Power-Aware Communication, Energy-efficient design.

1. Introduction

Recent advances in miniaturization and low-power design have led to active research in large-scale, highly distributed systems of small-size, wireless unattended sensors. Each sensor is capable of detecting ambient conditions such as temperature, sound, or the presence of certain objects. Over the last few years, the design of sensor networks has gained increasing importance due to their potential for some civil and military applications [1],[2]. A network of sensors can be used to gather meteorological variables such as temperature and pressure. These measurements can be used in preparing forecasts or detecting natural phenomena. In disaster management situations such as fires, sensor networks can be used to selectively map the affected regions directing the nearest emergency response unit to the fire. In military situations, sensor networks can be used in surveillance missions and can be used to detect moving targets, chemical gases, or presence of micro-agents. Sensors in such environments are energy constrained and their batteries cannot be recharged. Therefore,

designing energy-aware algorithms becomes an important factor for extending the lifetime of sensors.

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 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 [7].

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 sensor 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.

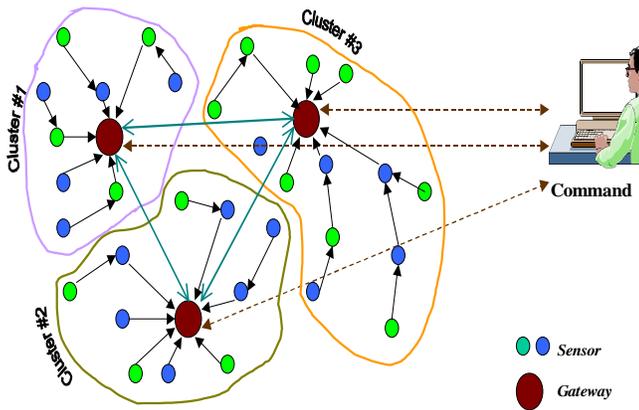


Fig. 1: Multi-gateway clustered sensor network

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 network and MAC layer protocols. 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 reduce 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. Section 2 describes our approach to energy-aware routing in sensor networks. In Section 3 we introduce our energy-aware MAC protocol. Description of the simulation environment and analysis of the experimental results can be found in Section 4. Finally, section 5 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 [6][19]. In our model, the gateways collaboratively locate the deployed sensors and group them into clusters so that sensors' transmission energy is minimized while balancing the load among the gateways [34][35][36]. 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.

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 its cluster as deemed by the command node. 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 performs system-level fusion of 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 Global GPS. 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. 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. [23].

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 [1]- [3]. 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 [13]. 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 and MAC 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 [29]. 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.

The effectiveness of three power-aware routing algorithms: Minimum total Transmission Power, Min-Max Battery Cost, and Max-Min Battery Capacity, is compared in [29]. 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 [8]. 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 nodes in proportion to their energy reserves. Our algorithm balances these considerations with other QoS metrics such as end-to-end delay. In addition, we consider the sensor role in a mission in the routing decision.

Achieving energy saving through activation of a limited subset of nodes in an ad-hoc wireless network has been the goal of some recent research such as SPAN [11], GAF [30] and ASCENT [8]. 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 processing duties in the sensor's state transition.

A signaling channel is used in [24] to intelligently turn off nodes that are not active, however nodes use a complex probe mechanism. Store-and-forward schemes of wireless networks, such as IEEE 802.11, have a sleep mode in which nodes are turned off [22],[31].

A power-aware Time Division Multiple Access (TDMA) Medium Access Control (MAC) protocol that coordinates the delivery of data to receivers based on the base station control is given in [12]. There are three phases in this TDMA: up-link phase in which nodes transmit data to the base station, down-link phase in which the base station transmits data to the nodes, and reservation phase in which nodes request new connections. The base station dictates a frame structure within its range. A frame consists of a number of data cells and a traffic control cell. Nodes with scheduled traffic are indicated in a list, which allows nodes without traffic to rapidly reduce power. The traffic control is transmitted by the base station and contains information about the subsequent data-cells, including when the next traffic control cell will be transmitted. Nodes explicitly request transmission from the base station, in a distributed manner, during the reservation phase. In our approach, the gateway performs the slot assignment based on its routing decisions. Our approach, as explained later, has four phases some of them have different functionality than their approach. Their approach requires the three phases to be present in every frame while in our approach the data send phase (up-link phase in their approach) is more frequent than the other phases leading to less control overhead and thus higher bandwidth efficiency. The gateway informs each node of its state so that a node can turn itself off. They did not discuss the effect of transmission errors on collision and network performance.

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.

Route setup in a cluster is centralized at the gateway. 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 [24]. 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 can account for resource requirements at the gateway node to cope with the responsibility of managing the assigned sensors [35]. Dependability issues related to the centralized network control can be addressed by fault-tolerance techniques [21] or through limited-scope re-clustering [36].

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 incoming 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 towards 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

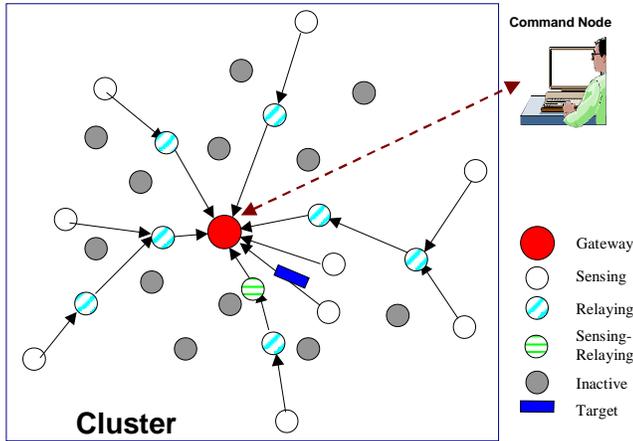


Fig. 2: A Typical Cluster in a Sensor Network

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. It should be noted that our approach is transparent to the method of selecting the nodes that should sense the environment. Fig. 2 shows a typical cluster tasked with a target-tracking mission.

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. 3 shows an example for energy model update.

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, send 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.

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

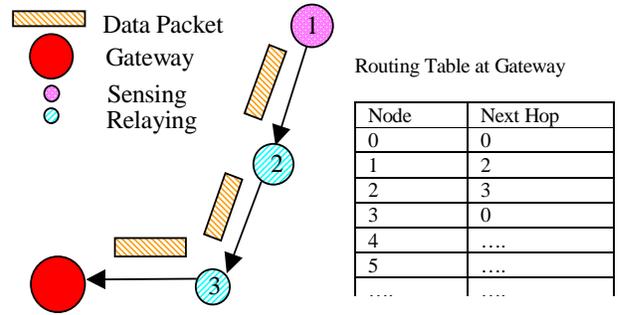


Fig. 3: When the gateway receives a packet from node1, it uses the routing table to update the energy model of nodes 1, 2, and 3, which are on the path from node1 to the gateway

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 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 is 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 in Section 3, can reduce the probability of collision. The energy model we used in the simulation is described in Appendix A.

2.2. Routing Approach

Since we have chosen a centralized approach for network management, source routing methodologies can be followed [28]. 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 delay of refresh messages. The model approximation is still accepted since we believe that frequent refreshing, together with fine-tuning of routing parameters, can keep deviation within tolerable limits. A detailed analysis of the effect of the model accuracy on performance is given in Section 4.

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 and then 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 .
- CF_1 : Energy stock = $c_1 \times f(energy_j)$ r node j . This cost factor favors nodes with more energy. The more energy the node contains, the better it is 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 relay-enabled nodes for routing rather than inactive nodes.
- CF_4 : Sensing-state cost = c_4 , where c_4 is a constant added when the node j is in a sensing-sate. This factor does not favor selecting sensing-enabled nodes to serve as relays, since they have committed some energy for data processing.
- CF_5 : Maximum connections per relay: once this threshold is reached, we add an extra cost c_5 to avoid setting additional paths through it. 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. This factor favors closer nodes.
- CF_7 : Queuing Cost = $c_7 \times \lambda / (\mu - \lambda)$, 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). Assuming an $M/M/1$ queuing model, this factor reflects the average queue length. Assuming equal service rate μ for each relay as well as equal data-sensing rate λ_s for each sensing-enabled node, CF_7 can be mathematically simplified to be the overall load on the relay node. The

overall load is the total number of sensing-enabled nodes whose data messages are sent via routes through this node. Thus, CF_7 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. For the gateway node, the values of the cost factors CF_1 , CF_2 , and CF_7 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 [10]. Path-optimization problems are usually solved using a shortest path (least-cost) algorithm [17]. 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 [32]. 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 [33] 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 suit centralized routing [28]. Therefore, we use Dijkstra's algorithm with the link cost d_{ij} for the link between the nodes i and j , redefined as $d_{ij} = \sum_k CF_k$.

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

3. MAC Layer Protocol

Although the new routing protocol is independent of the MAC layer protocol, choosing a certain MAC layer protocol may enhance the performance. Recent research results pointed out that the wireless network interface consumes a significant fraction of the total power. Measurements show that on a typical application like web-browser or email, the energy consumed when the interface is on and idle is more than the cost of receiving packets. This is because the interface is generally longer idle than actually receiving packets. Furthermore, switching between states (i.e. off, idle, receiving, transmitting) consumes time and energy [14]. 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 about the slots, which the node can use for its own transmission. The advantages of using a TDMA MAC layer are:

- Clock synchronization is built in the TDMA protocol. Recall that we need synchronization for the energy model refresh and sending rerouting decision from the gateway to the nodes.
- Collision among the nodes can be avoided since each node has its own assigned time slots. Problems can occur with the existence of communication errors: a packet containing the slot assignment can be dropped. If a node that does not hear the gateway decision turns itself off, then no collision can occur. However, we choose to implement the other alternative that a node retains its previous state if it does not receive a routing packet from the gateway in the pre-specified time slot, which leads to potential collisions. However, this collision probability is limited due to the following reasons:

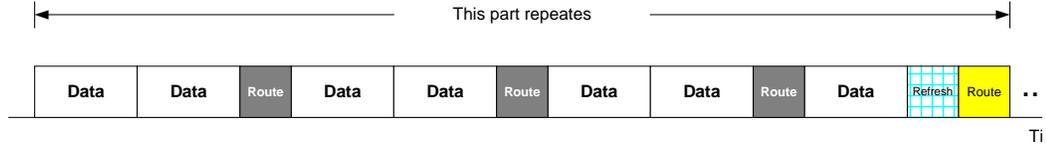


Fig. 4: MAC protocol time-based phases

- A node’s new state and forwarding table is highly probable to remain the same during consecutive rerouting phases.
- The wrong state of the node will be corrected during the next rerouting cycle, which means that the collision period is limited.
- If the node’s *previous* state was inactive, no collision will happen.
- If the node’s *new* state is inactive, no packets will be destined to it reducing the collision probability (recall that a node can overhear other nodes’ transmissions.)
- If the node receives a packet that is not in its forwarding table, this packet is dropped.
- Collision can only occur if the node happens to use the same time slot for transmission as a neighboring node since during transmission, we use the minimum transmission power required for reaching the destination. The same thing happens during receiving.

In the following subsections, we present the details of the MAC layer protocol.

3.1. Protocol Phases and Packet Format

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. Figure 4 shows an example of a typical sequence of phases.

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. The slot size in this phase is less than the slot size in the data transfer phase as will be explained later.

As previously discussed in Section 2, rerouting is performed when the sensor energy drops below a certain threshold, after receiving a status update from the sensors and when there is a change in the sensor organization. Since the media access in our approach is time-based, rerouting has to be kept as a synchronous event that can be prescheduled. To accommodate variations in the rate of causes of rerouting, two phases are designated for rerouting and scheduled at different frequencies. The first phase is called event-based rerouting and allows the gateway to react to changes in the sensor organization and to drops in the available energy of one of the relay sensors below a preset acceptance level. The second rerouting phase occurs immediately after the refresh phase terminates. During both phases, the gateway runs the routing algorithm and sends new routes to each node in its pre-assigned slot number and informs each sensor about its new state and slot numbers as shown in Table 1. Given that events might happen at any time and should be handled within acceptable latency, the event-based rerouting phase is scheduled more frequently than the refresh-based rerouting. If there has not been any event requiring messages rerouting, the event-triggered rerouting phase is shortened.

The lengths of the refresh and reroute phases are fixed since each node in the sensor network is assigned a slot to use in transmission during the refresh phase and to receive in it during the reroute phases. Similarly, the length of the data transfer phase is fixed. Although the number of active nodes changes from a rerouting phase to another, the length of the data transfer phase should be related to the rate of data sending and not to the number of active nodes. If the length of the data transfer phase is dependent on the number of active nodes, then a

Table 1: Description of MAC Protocol Phases

Phase	Initiator	Schedule	Actions
Data send	Active sensors	Assigned time slot	Send/forward data packets
Refresh	All sensors	Pre-assigned time slot	Inform gateway of sensor state
Refresh-based rerouting	Gateway	After refresh phase	Setup routes based on updated model.
Event-triggered rerouting	Gateway	Periodic	Setup routes to handle changes in sensor selection and energy usage.

node may consume power while it has nothing to do. It should be noted that during system design the size of the data transfer phase should be determined to accommodate the largest number of sensors that could be active in a cluster. Since the length of all phases is fixed, the period of the refresh and rerouting phases can be agreed upon from the beginning and does not have to be included in the routing packets.

The description for the packets of the corresponding phases is shown in the Table 2. The data packet used in the data transfer phase includes the originating sensor ID so that the gateway can adjust the energy model for the sender and relay sensors. In addition the sensor ID identifies the location and context of the sensed data for application-specific processing. The refresh packet includes the most recent measurement of the available energy. The optional location coordinates can be used to support sensor mobility.

The content of a routing packet depends on the new state of the recipient sensor node. If the sensor is to be *Inactive*, the packet simply includes the ID of the destination node. In case of a node that is set to sense the environment, the packet includes the data sending rate and the time slots during which these data to be sent. In addition, these sensing nodes will be told the transmission range, which the node has to cover. Basically the transmission power should be enough to reach the next relay on the path from this node to the gateway, as specified in the routing algorithm. Relay sensors will receive the forwarding table that identifies where data packet to be forwarded to and what transmission to be covered.

The forwarding table consists of ordered triples of the form: (time slot, data-originating node, transmission

Table 2: Description of various packet types

Source	Target	Type	Contents
Sensor	Gateway	Data	Orig. ID, Data
Sensor	Gateway	Refresh	Orig. ID, Source battery level, Source Location
Gateway	Inactive Sensor	Rerouting	Dest. ID
Gateway	Sensing Sensor	Rerouting	Dest. ID, Data send rate, Trans range, Time slots
Gateway	Relaying Sensor	Rerouting	Dest. ID, Forward table, Time slots

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 is proportional to the transmission power needed to send 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 to maintain and the changes to the routes will be quicker to communicate among the sensors. Such simplicity is highly desirable to fit the limited computational resources that sensors would have. We rely on the sensor organization and smart data fusion to tolerate lost data packets by allocating redundant sensors and applying analytical techniques [7].

3.2. Slot Size and Assignment

The slot sizes for the refresh and reroute phases are equal since they cover all sensor nodes in the cluster. Both slots are smaller than the slot for the data transfer phase. This is due to two reasons. First, the routing packet is typically less than the data packet. Second, during the data transfer phase many nodes are off which allows for larger slot sizes. In the other phases, all nodes must be on and communicating with the gateway. To avoid collision while packets are in transient, the slot size in the refresh and reroute phases should be equal to the time required to send a routing packet with maximum length plus the maximum propagation time in

the network, as calculated by the gateway. The slot size of the data-transfer phase equals the time required to send a data packet with maximum length plus the maximum propagation time in the network.

Slot assignment is performed by the gateway and communicated with the nodes during the rerouting phases. Different algorithms can be used for slot assignment. We assign each node a number of slots for transmission based on its current load. This leads us to two approaches for handling the TDMA-based MAC slot assignment problem, namely breadth and depth techniques. In the breadth slot assignment technique we follow a breadth-first-search (BFS), commonly used for graph parsing, to assign time slot numbers starting from the outmost active sensors. These outermost sensors are all sensing enabled since they are the source nodes of our data, and thus the initiator nodes in the routes towards the gateway. Such assignment is supposed to provide contiguous time slot numbers assigned for each relaying node to receive at, and thus saving the energy consumed in switching between on and off states. The other technique, namely depth assignment is based upon a depth-first-search (DFS) like. It tends to assign time slots contiguously over each route from the sensing node towards the gateway. Although this approach does not save the energy of switching between on and off states as the breadth technique, it still avoids the buffer overflow problem. In most cases each received packet will not wait in the buffer of the relay node and will be forwarded in the next time slot.

Figure 5 shows an example of the two slot assignment techniques. Nodes *A*, *B*, and *D* acts as sensor so they are assigned one slot for transmitting their data. Node *C* serves as a relay for nodes *A* and *D*, so it is assigned two slots. Node *E* acts as a sensor and a relay. It is assigned one slot for transmitting its own sensor data and 3 slots to relay other nodes' packets. In this

example, the gateway informs each node of the slots it is going to receive packets from other nodes and the slots it can use to transmit the packets.

Now for the breadth technique, the gateway informs nodes *A*, *B* and *D* to transmit their packets at time slots 1, 2 and 5 respectively. For node *C*, it is informed to listen to packets at time slots 1 and 2, and to forward them at time slots 3 and 4 respectively. Node *E* is assigned to turn its receiver on at time slots 3-5 (corresponding to the transmission slots of nodes *C* and *D*) to receive packets. And that it can use time slot 6 to transmit its own packet, as well as time slots 7-9 to forward packets. It should be noted here that this slot assignment algorithm provides contiguous slot numbers for each node, thus reducing the energy needed to switch between on and off states. However, it might lead to instantaneous buffer overflow. For example, if node *E* in Fig. 5 has only a buffer for 2 packets, then it can happen that it receives, in slots 3-5 3 packets from nodes *C* and *D*. This may lead to packet drop due to buffer overflow. However, if transmission and receiving slots were interleaved, this overflow cannot happen, as in the depth technique.

For the same example we apply the depth technique, as shown in the right side Fig. 5. For the packet generated by node *A*, it is assigned time slots 1 to send by node *A*, 2 to forward by node *C*, 3 to forward by node *E* to the Gateway. Similarly, packets generated by node *B* are assigned time slots 4, 5 and 6 to be sent by nodes *B*, *C* and *E* respectively. Similarly, node *D*'s packets are sent at time slots 7 and 8 by nodes *D* and *E* respectively. For node *E*'s own packets, they are assigned time slot 9. It is obvious that this technique avoids packet drops due to buffer overflow. However, nodes switch more frequently between on and off states.

The performance of both the depth and breadth techniques is compared via simulation, as reported in the next section.

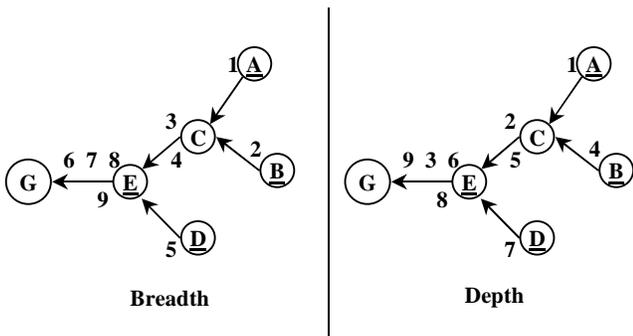


Fig. 5: An example of slot assignment techniques

4. Experimental Validation

The effectiveness of the routing and MAC protocols is validated through simulation. This section describes performance metrics, simulation environment and experimental results.

4.1. 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 a cluster is said to be partitioned ([9],[16],[26], [25]), reflecting the fact that some routes become invalid.
- *Time for last node to die*: This metric, along with the time to network partition, 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. Although efficient energy management is needed, some sensor network missions are delay sensitive.
- *Network Throughput*: Defined as the total number of packets received at the gateway divided by simulation time.
- *Average energy consumed per packet*: A routing algorithm that minimizes the energy per packet will, in general, yields better energy savings.

4.2. Environment Setup

In the experiments the cluster consists of 100 randomly placed nodes in a 1000×1000 meter square area. The gateway is randomly positioned within the cluster boundaries. A free space propagation channel model is assumed [4] 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 [18]. A node is considered non-functional if its energy level reaches 0. For the term CF_l , we used the linear discharge curve of the alkaline battery [25].

For a node in the sensing state, packets are generated at a constant rate of 1 packet/sec [23]. Each data packet is time-stamped when generated to allow tracking delays. In addition, each packet has an energy field that is updated during the packet transmission to calculate 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.

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 the sensors of the cluster. The set of sensing nodes change as targets move. 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 targets 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 four m/s to six m/s and a constant direction chosen uniformly depending on the initial target position in order for the target to cross the convex hull region.

Targets arrive in the deployment area according to a Poisson arrival process. The average inter-arrival time is chosen such that the average number of targets per unit time ranges from 1 to 16. Each target remains active until it leaves the deployment region area.

4.3. 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 sensor data packets. Unless mentioned otherwise, a refresh phase is scheduled periodically every 20 data phases.

4.3.1 Comparison between routing algorithms

In this section we present 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 performance of the new algorithm is compared with the following routing algorithms:

- *Direct routing algorithm*: In this algorithm, each node sends its data directly to the gateway [16].
- *Minimum transmission energy routing algorithm*: This algorithm chooses the intermediate nodes such that the transmit amplifier energy is minimized. The chosen cost function tries to minimize the sum of the distance squared between a node and gateway [16].
- *Linear battery*: This routing algorithm chooses the paths such that nodes with depleted energy reserves do not lie on many paths. In this routing algorithm, the node remaining lifetime is taken to be a linear function of its remaining energy, which is the normal behavior of some alkaline batteries [25].

Figures 6 through 12 summarize the comparative results. We can see from the figures that some algorithms, such as the minimum transmission energy routing algorithm fails to work at high targets arrival rates as the number of time slots becomes inadequate. This can be explained by noticing that the minimum transmission energy routing algorithm tries to minimize the transmission energy by taking short distances leading to more hops and thus more relays. Each relay requires a number of time slots for transmitting its own data. As the number of targets increases, the number of slots required becomes more than the number of available slots and thus the algorithm fails. It is worth mentioning here that the minimum transmission energy routing algorithm may still work under a different MAC layer protocol. However, choosing a contention-based MAC layer protocols may consume more energy due to contention and collisions. The linear battery routing algorithm ensures that the shortest-hop routing will be used when the network starts operation but as the network nodes approach the end of their lifetimes, the packets are routed so that no node dies before the others. This explains the similarity in performance between the direct routing algorithm and the linear battery routing algorithm especially in figures 6, 8, and 12.

Figure 6 shows that regardless of the minimum transmission energy routing algorithm, which fails at high target arrival rate, the new algorithm gives the best time for network partitioning. This is expected, as the new algorithm is the only algorithm of the remaining algorithms that takes energy consumption into consideration. At high load, the new algorithm gives an order of magnitude enhancement over the other algorithms.

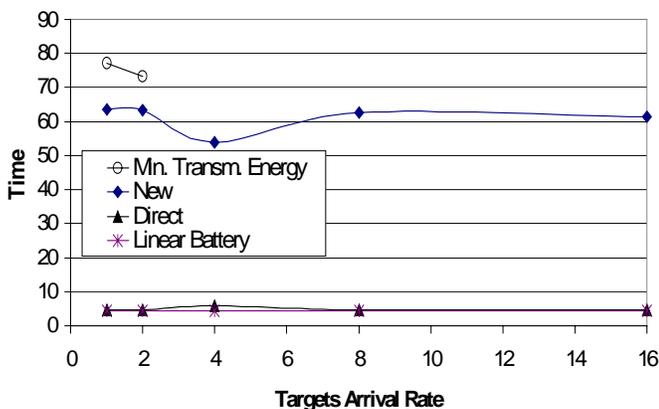


Fig. 6: Comparison between different routing algorithms (Time to network partition)

Figures 7 and 8 show the time for the last node to die and the average node lifetime respectively. The curves show that the new algorithm performs well under low and high target arrival rates. However these curves alone may be misleading without looking at Fig. 9, which shows the standard deviation of the nodes lifetime. The direct routing algorithm is in the lead in figures 7 and 8, since in this routing algorithm there is no packet relaying. Therefore, the node consumes energy only when it has data to send. Nodes very close to the gateway will consumes very little energy and their batteries will last longer. On the other hand Fig. 9 shows that the new algorithm gives the best standard deviation after the minimum transmission energy algorithm, which is an indication of the good predictability of the performance of the new algorithm. Under high load, the new algorithm is the most predictable with 11% enhancement over the other algorithms.

Figure 10 shows the average energy consumed per packet. The figure shows that the new algorithm's performance is consistent under different target arrival rates. Moreover, under heavy load, the new algorithm gives the best average energy consumed per packet with a 14% enhancement. This is expected as the new algorithm tries to minimize the energy consumption while other algorithms either fail to work or do not take energy consumption into consideration in the routing decision. Although the linear battery routing algorithm tries to conserve each node's battery, it does not try to reduce the energy consumed per packet. For example, the linear battery routing algorithm may choose a far away node with a large remaining battery level over a near node with moderate energy level leading to a large amount of transmission energy per packet.

The network throughput is shown in Fig. 11. The new algorithm's performance is accepted under different target arrival rates. The best throughput is achieved using the direct routing algorithm as it gives the minimum average delay per packet as shown in Fig. 12. However the nodes do not stay long under direct routing even under light load, as previously concluded from figures 6-8.

Figure 12 shows the average delay per packet for the different routing algorithms. The figure shows that the new algorithm performance is also consistent under different target arrival rates. The best average delay per packet is achieved by using the direct routing algorithm while the worst average delay per packet is achieved when the minimum transmission energy routing algorithm is used. Again, the minimum transmission energy routing algorithm tries to minimize the transmission power by taking short distances and larger number of hops leading

to increased delay. The opposite reasoning is applied to the energy consumed per packet shown in Fig. 10.

The above results show that the new algorithm gives a relatively good performance for all the metrics. Other algorithms may slightly outperform our algorithm in some metrics. However, the same algorithms perform

poorly on other metrics. Moreover, under heavy load, the new algorithm gives the best values in terms of time to network partitioning (with an order of magnitude enhancement), predictability (with 11% enhancement), and average energy consumed per packet (with 14% enhancement).

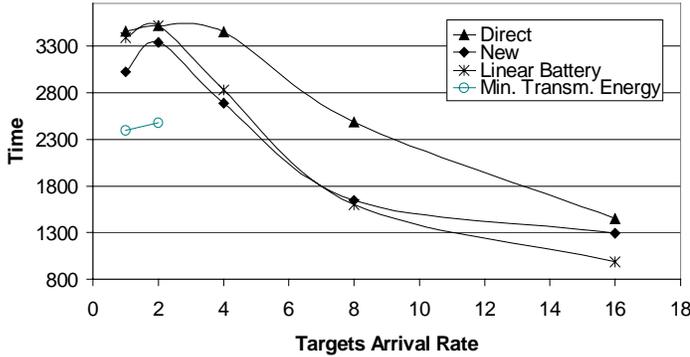


Fig. 7: Comparison between different routing algorithms (Time for last node to die)

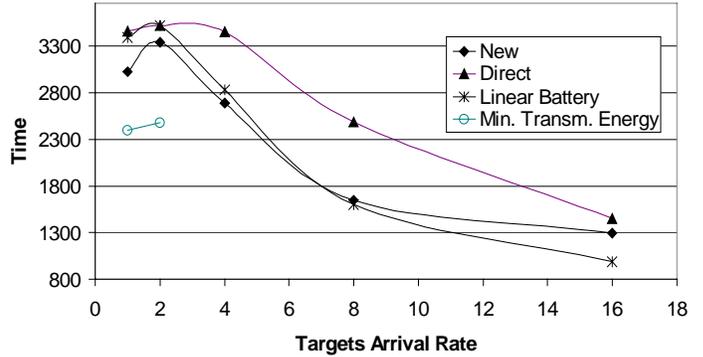


Fig. 8: Comparison between different routing algorithms (Average node lifetime)

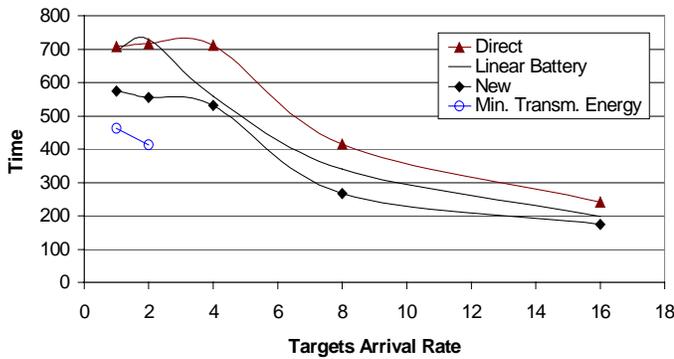


Fig. 9: Comparison between different routing algorithms (Standard deviation of nodes lifetime)

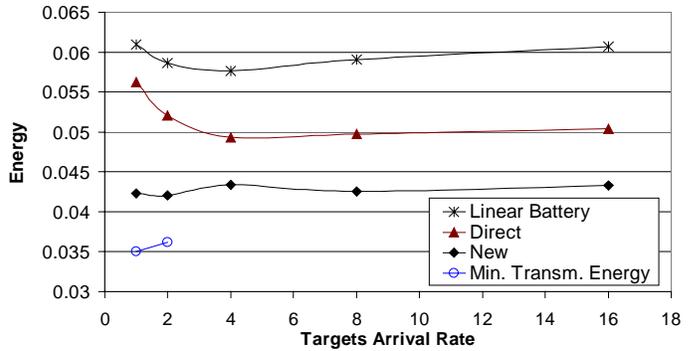


Fig. 10: Comparison between different routing algorithms (Avg. energy consumed per packet)

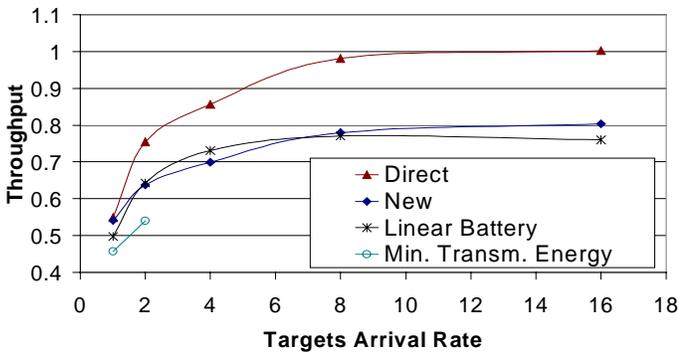


Fig. 11: Comparison between different routing algorithms (Network Throughput)

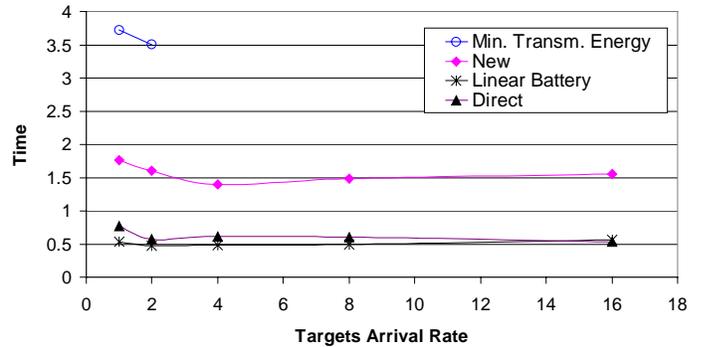


Fig. 12: Comparison between different routing algorithms (Average delay per packet)

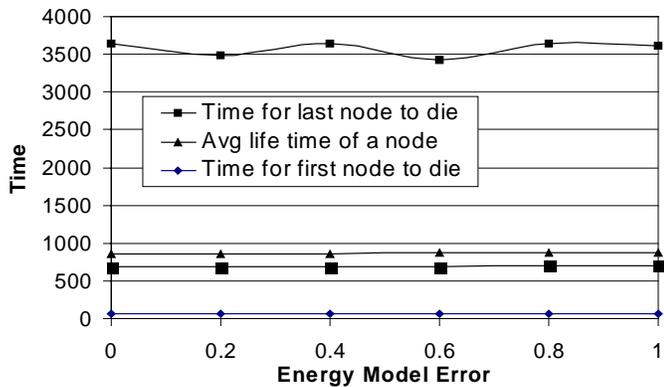


Fig. 13: Effect of energy model accuracy on network lifetime

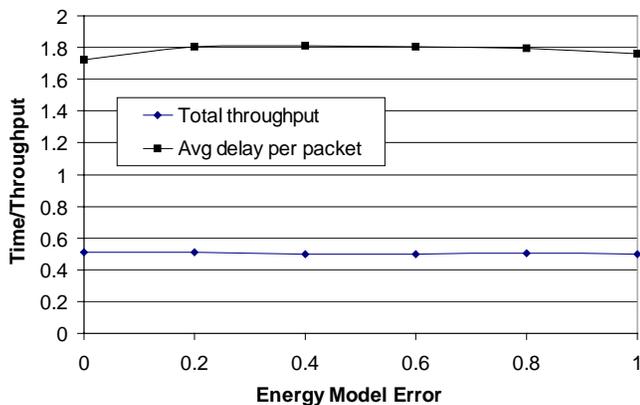


Fig. 14: Effect of model accuracy on throughput and delay

4.3.2 Effects of energy model accuracy

For this experiment, we introduce a percentage error in the energy model. This percentage error is taken to be a uniform random variable whose lower bound is 0 and upper bound ranges from 0% to 100% for different experiments. In this experiment, the energy model was taken to underestimate the actual node energy. The results are shown in figures 13 and 14. The 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 studied the effect of overestimating the node energy level and similar results were obtained.

4.3.3 MAC layer protocols evaluation

In this section, we use simulation to compare the performance of the two proposed slot assignment techniques, the breadth and depth slot assignment. We use another performance metric which is the number of state changes between on and off for the radio circuitry

per sensor which represents the overhead of the power saving process. All the performance metrics are plotted against increasing buffer sizes at the sensor nodes.

In Figure 15, we can see the advantage of the depth technique over the breadth in terms of packet drop count. The number of packets dropped due to buffer overflow in the case of the depth slot assignment is not zero. This is due to two reasons: (a) we do not know when a sensing node will generate its data, and (b) a node retains its buffer when the slot assignment changes. Both

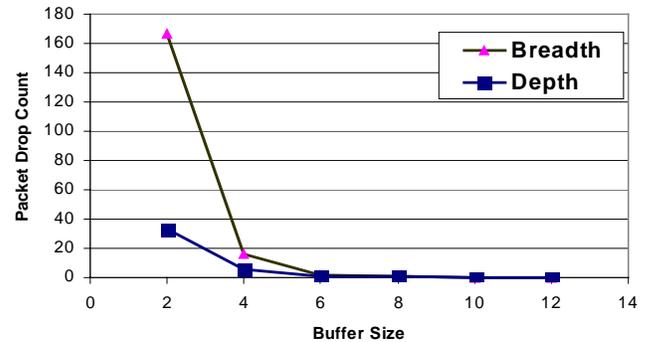


Fig. 15: Effect of buffer size on packet drop count

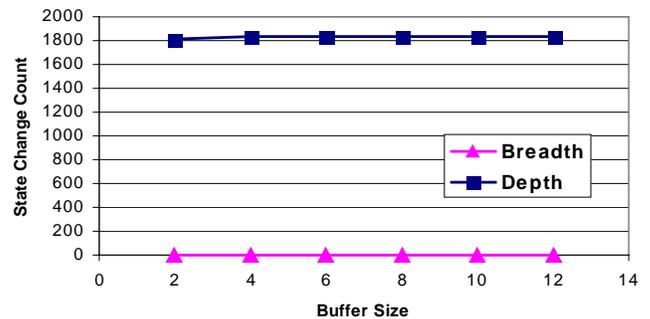


Fig. 16: Effect of buffer size on transmitter state change count

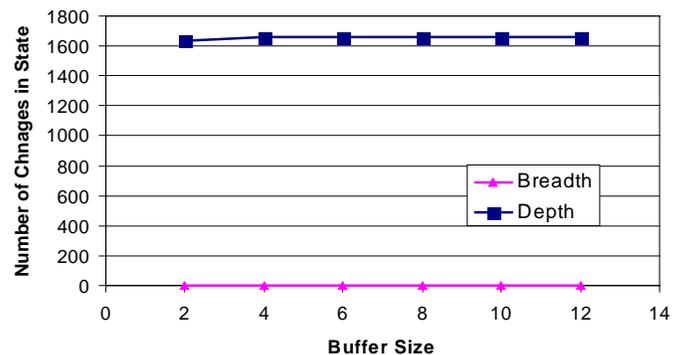


Fig. 17: Effect of buffer size on receiver state change count

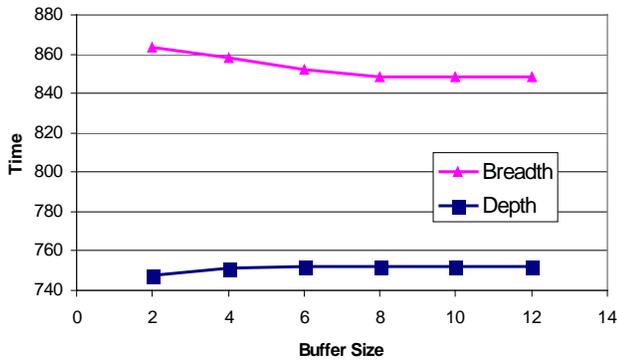


Fig. 18: Effect of buffer size on average node lifetime

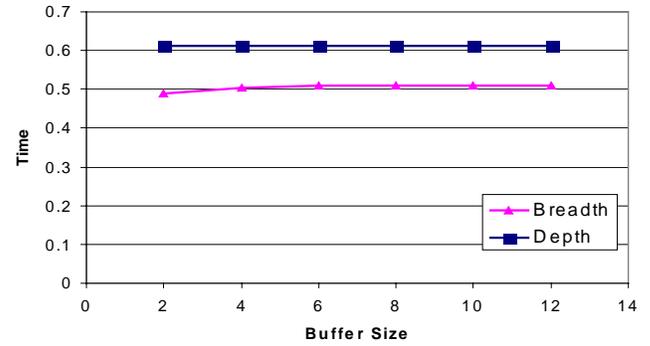


Fig. 20: Effect of buffer size on throughput

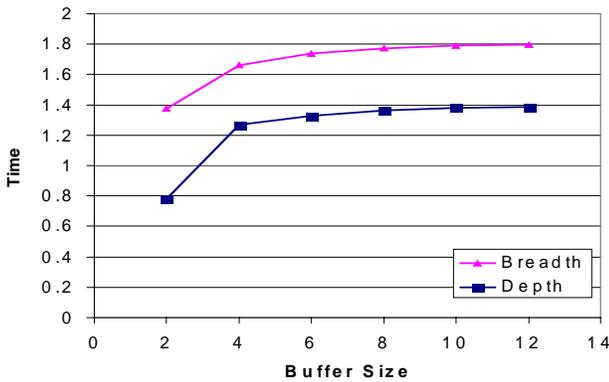


Fig. 19: Effect of buffer size on end-to-end delay

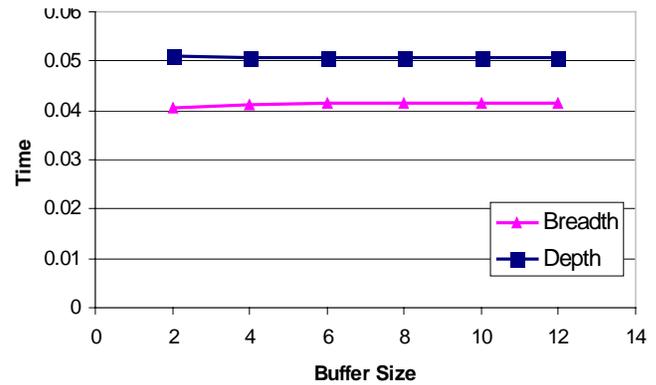


Fig. 21: Effect of buffer size on energy per packet

reasons can lead to transient buffer build-up and hence packet dropping, especially for small buffer sizes.

Figures 16 and 17 show that for the breadth method, the number of changes in state is zero. Thus the breadth technique saves energy. The number of state changes for the transmitter is higher than for the receiver. This is expected as each node at least transmits what it receives (if it does not generate new packets.) This means that the number of transmission slots is larger than the number of receiving slots. Therefore, it is more probable to change state while you are transmitting than when you are receiving.

There is a slight increase in the number of state changes as the buffer size increases. As the buffer size increases, the number of packets that reaches the gateway increases leading to a more accurate model at the gateway. This also explains the decrease of the average energy consumed per packet shown in Fig. 21.

Figure 18 shows that the average node lifetime. Lifetime in case of breadth technique is higher as more packets are dropped and not forwarded saving the energy

of the nodes, but with lower throughput, as shown in Fig. 20.

In Fig. 19, when the buffer size increases, the average delay per packet increases due to the increased queuing delay. However in Fig. 20, the throughput does not decrease as less number of packets is dropped due to more available buffer size.

As seen from Fig. 20, throughput is lower in case of breadth technique since the number of packets dropped is higher.

In summary, the above results show that the breadth technique is better when the energy required for changing the sensor's state between on and off is critical. However, the depth technique is more reliable regarding packet delivery since it avoids packet drops due to buffer overflow. The depth technique is also superior with respect to end-to-end delay as well as throughput.

5. Conclusion and Future Work

In this paper, we have introduced a novel approach for energy-aware management of wireless 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 results show that the deviation in the model has little effect on performance with infrequent periodic model adjustment.

We have also presented in details a new MAC layer protocol. We have proposed two major techniques for slot assignment. Simulation results demonstrate a comparative evaluation of the breadth and depth slot assignment techniques with increasing buffer sizes. The simulation results demonstrated that the breadth technique is recommended in case the energy consumed for changing the sensor's state is high. On the other hand, the depth technique offers more reliable data packet delivery since it is more tolerant to packet drops caused by buffer overflow. The depth technique also gives better results regarding end-to-end delay as well as throughput.

Using the proposed protocols, Simulation results show an order of magnitude enhancement in the time to network partitioning, 11% enhancement in network lifetime predictability, and 14% enhancement in average energy consumed per packet.

Our future plan includes extending the system model to allow for node mobility. We are currently addressing inter-cluster interaction and operations, resources at the cluster level, and dynamic and reservation-based TDMA slot assignment techniques in the MAC layer, among others.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramanian, E. Cayirci, A survey on sensor networks, *IEEE Communications Magazine*, August 2002.

[2] I. Akyildiz, W. Su, Y. Sankarasubramanian, E. Cayirci, Wireless sensor networks: a survey, *Computer Networks* 38 (2002) 393-422.

[3] N. Bambos, Toward power sensitive network architectures in wireless communication: concepts issues and design aspects, *IEEE Personal Communications*, June 1998.

[4] J. Andresen, T. Rappaport, S. Yoshida, Propagation measurements and models for wireless communications channels, *IEEE Communications Magazine*, 33 (1) (1995).

[5] M. Bhardwaj, et. al, Upper bounds on the lifetime of sensor networks, In *Proceedings of ICC 2001*, June 2001.

[6] A. Buczak, V. Jamalabad, Self-organization of a heterogeneous sensor network by genetic algorithms, *Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et. al. (eds.), Vol. 8, ASME Press, 1998.

[7] R. Burne, et. al, A self-organizing, cooperative UGS network for target tracking, *Proceedings of SPIE Conference on Unattended Ground Sensor Technologies and Applications II*, Orlando, April 2000.

[8] A. Cerpa, D. Estrin, ASCENT: adaptive self-configuring sensor networks topologies, *Proceedings INFOCOM 2002*, New York, June 2002.

[9] J. Chang, L. Tassiulas, Energy conserving routing in wireless ad-hoc networks, *Proceedings of IEEE Infocom*, 2000.

[10] S. Chen, Routing support for providing guaranteed end-to-end quality of service, Ph.D. Thesis Dissertation, University of Illinois at Urbana-Champaign, 1999.

[11] B. Chen, et al., Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, *Proceedings of MobiCom 2001*, Rome, Italy, July 2001.

[12] P. Havinga, G. Smit, Energy-efficient TDMA medium access control protocol scheduling, *Proceedings Asian International Mobile Computing Conference (AMOC 2000)*, November 2000.

[13] P. Havinga, G. Smit, Design techniques for low power systems, *Journal of Systems Architecture*, 46 (1) (2000).

[14] 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.

[15] W. Heinzelman, et. al, Energy-scalable algorithms and protocols for wireless microsensor networks, *Proceedings International Conference on Acoustics, Speech and Signal Processing (ICASSP '00)*, June 2000.

[16] 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.

[17] M. Hung and J. Divoky, A computational study of efficient shortest path algorithms, *Computers and Operations Research*, Vol. 15 (1988), 567-576.

[18] M. Gerla, G. Pei, and S. Lee, Wireless, mobile ad-hoc network routing, *IEEE/ACM FOCUS'99*, May 1999.

[19] C. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, *IEEE Journal on Selected Areas of Communications*, 15 (7) (1997).

[20] R. Min, et. al, An architecture for a power-aware distributed microsensor node, *IEEE Workshop on Signal Processing Systems (SiPS '00)*, October 2000.

[21] D. Pradhan, *Fault-tolerant computer system design*, Prentice Hall, New Jersey, 1996.

[22] C. Röhl, H. Woesner, A. Wolisz, A short look on power saving mechanisms in the wireless LAN standard draft IEEE 802.11, *Proceedings of the 6th WINLAB Workshop on third generation Wireless Systems*, New Brunswick, New Jersey, March 1997.

[23] SenTech Inc., Data sheet for the acoustic ballistic module, <<http://www.sentech-acoustic.com/>>

[24] S. Singh, C.S. Raghavendra, PAMAS: power aware multi-access protocol with signaling for ad hoc networks, *ACM Computer Communications Review*, July 1998.

[25] S. Singh, M. Woo, C. S. Raghavendra, Power-aware routing in mobile ad hoc networks, *Proceedings of ACM MobiCom'98*, Dallas, Texas, October 1998.

- [26] J. Chang and L. Tassiulas, Routing for maximum system lifetime in wireless ad hoc networks, In Proceedings of 37th Annual Allerton Conference on Communication, Control and Computing, 1999.
- [27] A. Sinha, A. Chandrakasan, Energy aware software, Proceedings of the 13th International Conference on VLSI Design, pp. 50-55, Calcutta, India. January 2000.
- [28] W. Stallings, Data and computer communications, Macmillan Publishing Company, 3rd edition, 1991.
- [29] C-K. Toh, Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks, IEEE Communications Magazine, June 2001.
- [30] Y. Xu, J. Heidemann and D. Estrin, Geography-informed energy conservation for ad hoc routing, Proceedings of MobiCom 2001, Rome, Italy, July 2001.
- [31] S. Xu, T. Saadawi, Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?, IEEE Communications Magazine, June 2001.
- [32] F. Zhan, Three fastest shortest path algorithms on real road network: data structures and procedures, Journal of Geographic Information and Decision Analysis, 1(1) (1998).
- [33] F. Zhan, C. Noon, Shortest path algorithms: an evaluation using real road networks, Transportation Science, 1996.
- [34] R. Mathew and M. Younis, "Energy-Efficient Bootstrapping Protocol for Sensor Network," in the *Proceedings of the International Conference on Wireless Networks (ICWN'02)*, Las Vegas, Nevada, June 2003 (to appear).
- [35] G. Gupta, M. Younis, "Load-Balanced Clustering in Wireless Sensor Networks," in the *Proceedings of the International Conference on Communication (ICC 2003)*, Anchorage, Alaska, May 2003 (to appear).
- [36] G. Gupta, M. Younis, "Fault-Tolerant Clustering of Wireless Sensor Networks," in the *Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003)*, New Orleans, Louisiana, March 2003.

Appendix A: Sensor's Energy Model

A typical sensor node consists mainly of a sensing circuit for signal conditioning and conversion, digital signal processor, and radio links [5],[20]. The following summarizes the energy-consumption models for each sensor component.

Communication Energy Dissipation: We use the model of [5],[15]. 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 A.1 summarizes the meaning of each term and its typical value.

Computation Energy Dissipation: We assume the leakage current model of [15],[20],[28]. The model

depends on the total capacitance switched and the number of cycles the program takes. We used parameter values similar to those in [27].

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 [5]:

$$E_{sensing} = \alpha_3 * r$$

For the Ballistic Audio sensor [23], the energy

Table A.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.

dissipated for sensing a bit is approximately equal to the energy dissipated in receiving a bit. Therefore, α_3 is taken equal to α_{12} .