ENERGY SAVING ROUTING PROTOCOLS IN WIRELESS SENSOR NETWORKS

by

Orhan Çelik

B.S., Computer Engineering, Boğaziçi University, 2001

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems Control Engineering
Boğaziçi University
2006

ACKNOWLEDGEMENTS

I would like to thank to Systems Control Engineering Institute at Boğaziçi University and, more specifically, Prof. Yorgo İstefanopulos for giving me the chance to be one of their students and conduct this research. I would also like to thank to Boğaziçi University Computer Engineering Department, Electrical and Electronic Engineering Department and their professors for building my infrastructure with B.S and M.S. courses. Most importantly, I owe a lot to my advisor, Prof. Cem Ersoy, for his support, advices, ideas, corrections and observations throughout the writing of the thesis.

I would like to acknowledge the support of my family throughout my master studies, especially of my parents, İbrahim and Birsen Çelik, whose strength and love have always provided me with an important source of motivation. I should also thank to my colleagues and management in PricewaterhouseCoopers who created opportunities for me to attend my master courses. Finally, I would like to thank Maria Demesticha, for her constant encouragement throughout my master studies.

ABSTRACT

ENERGY SAVING ROUTING PROTOCOLS IN WIRELESS SENSOR NETWORKS

Developments in wireless communications and electronics have made designing low-cost sensor networks possible. The sensor networks have many application areas such as health, military, home, agriculture, environmental. Because each sensor has to be low-cost, they have very limited battery and lifetime of the network depends heavily on saving energy. One way of saving energy is designing appropriate routing protocols. In this thesis, we propose some new routing protocols that save more energy and increase the network lifetime comparing to the classical protocols especially in networks with a large number of sensors.

ÖZET

KABLOSUZ ALGILAYICI AĞLARINDA ENERJİ TASARRUFU SAĞLAYAN YÖNLENDİRME PROTOKOLLERİ

Kablosuz iletişim ve elektronik teknolojilerindeki gelişmeler düşük masraflı kablosuz algılayıcı ağların yapımına imkan tanıdı. Algılayıcı ağların sağlık, askeri, ev ve tarım, çevresel izleme gibi birçok uygulama alanı vardır. Algılayıcılar ucuz olmak zorunda olduklarından kısıtlı pil kaynakları vardır ve algılayıcı ağın çalışma ömrü büyük ölçüde enerji tasarrufuna bağımlıdır. Uygun yönlendirme protokolleri tasarımlamak, enerji tasarrufu yapmanın etkin bir yoludur. Bu yüksek lisans tezinde, özellikle çok sayıda algılayıcı içeren ağlar için klasik yönlendirme protokollerine göre daha çok enerji tasarrufu yapıp ağ ömrünü uzatan yönlendirme protokolleri önerip detaylandıracağız.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

AA	Always Alkaline
A/D	Analog to Digital
BAS	Remaining Battery over Area Sensed Algorithm
BAS-H	Remaining Battery over Area Sensed with Traffic History Algorithm
$B_{i,t}$	Sensor *i*'s remaining battery at a given time *t*
$C_{i,j}$	Cost metric between sensors *i* and *j*
$Cap_{i,j}$	Capacity of node *i* to *j*, i.e. Remaining Battery of node *i* over energy required to send a packet to *j*
$Cost_{i,\ sink}$	Cost metric between sensors *i* and the base station
$Cost(N_i)$	Cost field of the request packet *i*
DC	Direct Current
EAR	Energy Aware Routing Protocol
EAR+A	Altruistic Energy Aware Routing Protocol
e_{ij}	Energy used to transmit and receive on the link between *i* and *j*
	$Ef_{i,j}(k, d_{i,j})$ Forwarding energy used in node *i* for forwarding package of length *k* to node *j* at a distance of *d*
$Er_i(k)$	Receiving energy used in node *i* for receiving package of length *k*
	$Et_i(k,d)$Transmission energy used in node *i* for transmitting package of length *k* to distance *d*
$Ew_{i,t}$	Waiting energy spent in sensor *i* for time *t*
FPS	Protocol for Radio Power Scheduling
FT_i	Forwarding Table of node *i*
GEAR	Geographical and Energy Aware Routing
IEEE	Institute of Electrical and Electronic Engineers
LEACH	Low-Energy Adaptive Clustering Hierarchy
LoC	Largest of Capacities Algorithm
LR-WPANs	Low-Rate Wireless Personal Area Networks
LWIM	Low Power Wireless Integrated Microsensors
MAC	Medium Access Control Layer

MIT	Massachusetts Institute of Technology
$MonArea_t$	Size of the area that all the live sensors can monitor at a given time
MTE	Minimum Transmission Energy Algorithm
MUP	Minimum of the Used Percentages Algorithm
Ni	Neighbors of node *i*
NL_1	Time passes until one sensor dies in network
NL_2	# of events that cannot reach the base station
NL_3	Percentage of the area that all the live sensors can monitor comparing to the monitoring area of first deployment
$NormBat_i$	Updated $Battery_i$, which is the theoretical battery level of *i*, used in the cost computations for the BAS algorithm
$Per_{i,j}$	Percentage of node *i*'s energy spent for sending a packet to *j*
PHY	Physical Layer
PP	Projected Percentages Algorithm
Proc.	Processor
$ProjectedPer_{i,j}$	Percentage of node *i*'s enegy spent for sending a packet to *j* multipled by *i*'s traffic load.
RAM	Random Access Memory
Ri	Residual energy at sensor *i*
ROM	Read Only Memory
SensAreaPer(i,j)	Shared sensing area percentage of sensors *i* and *j* over the whole sensing area of sensor *i* in BAS algorithm
T(n)	Probabilistic threshold value for the node *n* becomes a cluster-head for that round
WINS	Wireless Integrated Network Sensors
µAMPS	Micro Adaptive Multi-domain Power-aware Sensors
µ-OS	Micro Operating System
α	EAR protocol's transmission energy exponent
β	EAR protocol's residual battery exponent
γ	Path-loss exponent for radio transmission
δ	In BAS algorithm, weight factor effecting how much the neighbors will contribute to the sensor

1. INTRODUCTION

Wireless Sensor Networks are made out of many tiny sensors that are spread across an area for monitoring a phenomenon. Sensors communicate with other sensors to forward the information about that phenomenon to a base station- sink. Sensors are low cost and have limited battery; sink(s) are relatively large processing units that do not have power restrictions and pass the information to the processing center, that can be far from the event area, with their long range communication capability.

The sensor network application areas have a wide range, like disaster detection such as forest fire or flood detection, patient monitoring and micro-surgery, home and office accessories communication, military intrusion detection, agricultural crop monitoring, pricing goods in the markets, inventory handling and wildlife habitat monitoring. They can also be used for interaction of cars in traffic for safety, virtual keyboards for PC and musical instruments, commanding industrial robots, making social studies on human interaction, hostile environment exploration, monitoring seismic activity, and the monitoring of freshwater quality. They can be used for: civil engineering; monitoring buildings, urban planning and disaster recovery; for other military applications like military asset monitoring, surveillance and battle-space monitoring, urban warfare and self-healing minefields [1].

Sensors control a wide area or many goods, and as their communication range is smaller than the area they control, they need to be many in numbers and use multi-hop communication to the sink. Each of them should be cheap, so they have a limited battery. Since they are many in numbers, and because in many cases they operate in areas where human intervention is hard, it is generally not possible or feasible to replace the batteries.

In the following sections, we will explain the sensor node architecture, and why energy saving is the biggest problem in Wireless Sensor Networks. We will also explain the factors affecting the design of Wireless Sensor Networks.

1.1. Sensor Node Architecture

Sensor nodes have four main parts, the embedded sensor unit, the processor unit, the radio unit and the battery unit. As stated in [2], observations about the environment are gathered using sensor units consisting of sensors connected to an analog-to-digital (A/D) converter. After enough data is collected, or once an important event has occurred, the processing unit of the node can process the data prepared for sending to a nearby node or to the far away base station. The processor unit is formed by a microprocessor that has RAM and flash ROM for data and program storage, and a "μ-OS", an operating system with light memory and restricted computational capabilities. To send data or control messages to neighboring nodes, they are passed to the node's radio unit. The battery unit with DC-DC conversion provides the power for the node needed in the previous tasks. A sample sensor node's units are depicted in Figure 1.1.

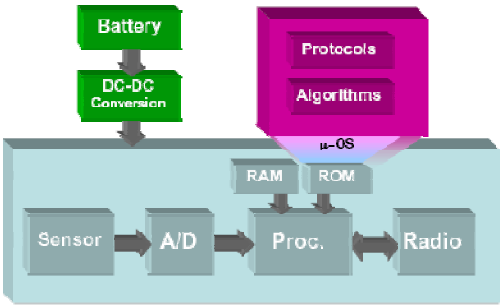

FIGURE 1.1. Sensor node architecture [2]

1.2. Wireless Sensor Networks Design Factors

The design of routing protocols in Wireless Sensor Networks is influenced by many challenging factors and design issues. As stated in Karaki et al. [3], these factors are: *Node deployment* in WSN is application-dependent and can be either manual (deterministic) or randomized. If the deployment is deterministic, data can be routed through pre-determined paths. *Energy consumption* is the biggest problem. Sensors should both sense their environment and be a relay node to the traffic coming from other sensors. *Data reporting method* can be time-driven, event-driven, query-driven or a hybrid of all of these methods. The reporting method is decided according to the application requirements. *Node/link heterogeneity* is another design issue. In many studies, all sensor nodes were assumed homogeneous and considered to have equal computation, communication, and power capacity, but this is not the case for some real-world applications. *Fault tolerance* is another factor; the failure of some of the sensor nodes should not affect the overall task of the sensor network. *Scalability* is another requirement: The number of sensor nodes deployed in the sensing area may be hundreds or thousands, and routing algorithms should be able to handle each case. *Mobility* is also important; in many studies, sensor nodes are assumed to be fixed in their positions. However, in many applications, both the base station(s) and sensor nodes can be mobile. Since the *transmission medium* is wireless, handling collusions and re-transmissions is important. Time division multiple access (TDMA) MAC layers work better than carrier sense multiple access (CSMA) MAC layers. *Connectivity* and *coverage* are other important factors that mainly depend on the architecture of nodes: the communication and sensing ranges, and the deployment method: number of nodes deployed and their locations. *Data aggregation* can be used on similar packets from sensors sensing similar events to reduce the number of transmissions. For some applications, data should be delivered to the base station in a limited time, therefore latency should be small. For these applications, *quality of service* can be as important as the network lifetime.

1.3. Need for Energy Saving

The main problem with the power of wireless sensor networks is the development of battery technology. The battery technology depending chemistry is not developing fast enough to meet the needs of fast developing technologies like processing units or radios. Nodes are expected to operate for 5 to 10 years with an AA-sized battery, an aim that is far from current wireless technology. Due to this reason, the primary concern designing the sensor’s hardware is energy saving.

Power saving can be achieved in the node architecture, physical, MAC and Network layers. There is substantial research on system-level power awareness such as dynamic voltage scaling [4], radio communication hardware, low duty cycle issues, system partitioning, sleeping schedules [5], and energy aware MAC protocols [6], [7], [8], [9], [10]. In addition, there is a lot of ongoing research over solving inter-operating issues, like the IEEE Standard 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) [11], [12].

In Sensor Networks, as wireless communication drains the most energy, we need to use energy efficient routing protocols. In addition, these routing protocols should be able to handle the changes in the network. In a sensor network, some of the sensors can consume all of their energy or simply they may stop working because they are made out of cheap materials and are prone to failures. The network protocol should be able to handle changes in the infrastructure and continue operating even if some of the sensors die. It is also desirable for the network to decide on routing locally: every sensor should decide where to forward packets.

In this thesis, we will attempt to improve the network lifetime in the network layer by proposing new routing protocols. Our routing algorithms aim to meet these requirements: save more energy to increase the network lifetime in comparison to classical protocols especially for networks with a large number of sensors. Although network algorithms should be defined according to the underlying physical and MAC layer, we have supposed that these layers are ideal and give optimal performance for all of the algorithms we have worked with.

The rest of the thesis is organized as follows: In the second chapter, we will name the main research groups and their projects. Then, we will explain how some of the important WSN routing protocols are working and their performance improvements against their competitors. In the third chapter, we will define our research problem's details. We will explain the network lifetime metric and energy usage model. In the fourth chapter, we will explain our starting point routing protocol and over it, we will offer some other routing protocols. Our main emphasis will be including the remaining battery, traffic over the sensors and existence of near sensors monitoring similar areas parameters to the routing decision. In the fifth chapter, we will explain the network model we used in our simulations and present our results of simulations. Finally, we will share our conclusions and future work in the last chapter.

2. WIRELESS SENSOR NETWORKS ROUTING ALGORITHMS

2.1. Research Groups and Projects for Wireless Sensor Networks

As wireless sensor networks will have a wide application area in the future, they have gained substantial research interest. There are many groups working on the area of Wireless Sensor Networks. Some of them are as follows. IEEE 802.15 Working Group for Wireless Personal Area Network (WPAN) [13] is defining Physical and MAC layer [14] industrial standards. Wireless Integrated Network Sensors (WINS) [15] of UCLA Electrical Engineering Department, has developed LWIM (Low Power Wireless Integrated Micro sensors) and WINS communication protocol working in collaboration with the Rockwell Science Center. MIT µAMPS (Micro Adaptive Multi-domain Power-aware Sensors) Project [16] has developed the µAMPS hardware and LEACH algorithm. Berkeley WEBS: Wireless Embedded Systems Group [17] has worked on the Smart-Dust and Pico-Radio Projects. It is also working on other projects like TinyOS: Operating System support for tiny-networked sensors, and FPS: a network protocol for radio power scheduling in Wireless Sensor Networks.

2.2. Classification of Wireless Sensor Networks Routing Algorithms

Many surveys have been written for Wireless Sensor Networks and their routing schemas: Karaki et al. [3], Akyildiz et al. [18], Akyildiz et al. [19], Demirkol et al. [20], Akkaya et al. [21], Rentala et al. [22], Xu [23], Royer et al. [24], Sahni et al. [25]. They generally classify the routing protocols according to network structure as flat network routing, hierarchical network routing and location-based routing. Classification can be made according to their protocol operation as negotiation-based routing, multipath-based routing, query-based routing, and QoS-based routing. The initiator of communications as the source or destination can also be used to categorize them. Path establishment can be made proactive, reactive or a hybrid of both.

Generally, Static Routing schemes tend to try to minimize the energy used in the routing process. Drawback is generally these schemes heavily load a few of the sensors and after burning all of the energy of these few sensors, these schemes are open to the creation disconnected networks. The first sensor death in these schemes happens very early. To overcome this problem and to increase the network lifetime, dynamic routing protocols are developed. They do not use the same routing path for a long time: instead they alternate the routing paths according to the energy remaining in the sensors, and form clusters and other methods to increase the network lifetime. In the next sections, we summarize some of the important routing algorithms from the literature.

2.3. Routing Algorithms in the Literature

2.3.1. Sensor Protocols for Information via Negotiation (SPIN)

A group of adaptive protocols called SPIN [26] was designed to solve the problems of the classical protocols. *Flooding* and *gossiping* are classical routing protocols that were first applied to Sensor Networks, but they had disadvantages in this domain. Applying flooding to Sensor Networks causes *implosion*, which is duplicate messages arriving to the same node; *overlap*, when two nodes that are in the same region send similar messages to the same neighbor and *resource blindness*, that is nodes not taking energy constraints into consideration.

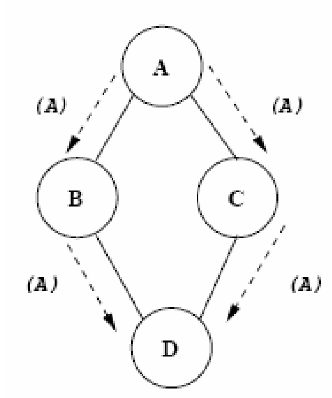

FIGURE 2.1. Implosion Problem: A’s information is sent to D from both B and C [26]

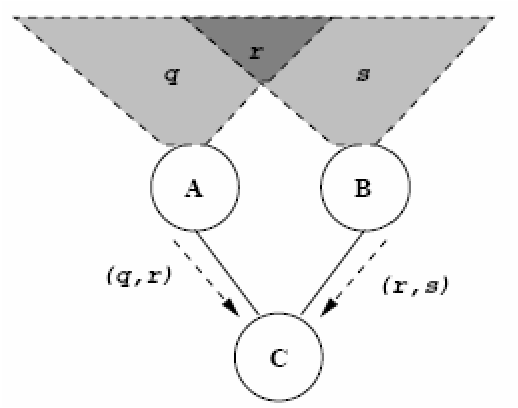

FIGURE 2.2. Overlap Problem: C receives information of region *r* from both A and B [26]

For solving *implosion*, gossiping randomly selects one or a sub-set of neighbors and then sends the message only to those neighbors, but this brings a propagation delay. SPIN solves these problems by using negotiation and resource adaptation. SPIN has a flat network structure and reactive routing and is a source-initiated protocol. Whenever a sensor has data, it sends its neighbors a description of the data (*meta-data*) with an advertisement packet (ADV), the interested neighbors answer the advertisement with a request packet (REQ), and the sensor sends the entire data packet (DATA) only to the interested neighbors. The importance of SPIN is that it is one of the first algorithms that introduce local messaging: sensors keep routing information only about their direct neighbors, which brings scalability. SPIN-2 is the resource aware version that refrains from going into excessive communication when the battery level becomes low. The problem with SPIN protocols is: even if some sensors (sinks) are interested in the data, if the sensors between the source and the sink are not interested in the data, the data cannot reach the destination.

2.3.2. Directed Diffusion

Directed diffusion [27] is an important milestone in data-centric wireless networks, and many algorithms are developed based on it. Directed diffusion has a flat network structure. It is a destination-initiated protocol that uses reactive routing. The protocol uses data-centric routing, where queries are answered by a sub-set of all sensors that have a certain kind of information and not the whole network. This special information query can be a question like, "What is the temperature at region *r*?" or, "Which are the areas that have a temperature over 10˚C?" Directed diffusion consists of three stages: *interest propagation, gradient setup* and *data delivery*.

In "*interest propagation"*, the sink node floods an interest for a kind of data through the network. The reason for using interest requests is to eliminate the possibility of receiving undesired or irrelevant data. The initial interest also specifies the initial frequency data flow from sensors to the sink, which could be every minute, and includes a timestamp for the nodes to stop sending data, for example after ten minutes. Nodes add the interests they receive to their interest cache. These interest entries contain the ID of each neighbor from whom the interest was received and the data rate towards that neighbor.

In the second stage of directed diffusion, "*gradient setup*", nodes having relative information meeting the interest start sending the data to all of their neighbors at the specified frequency. This frequency of sending data is a gradient: it is the frequency (data rate) at which to send data about a specific interest to a specific neighbor. Directed diffusion also uses data aggregation. Nodes receiving data for the sink add the data to their data cache. Nodes check the data cache when they receive a data message to see if the data is new. If the data is already in the cache, it means that the data has already been forwarded and the node will disregard the message. When data reaches the sink, the sink reinforces one or more paths by sending another interest. This interest is for the same data, but it is sent to a specific source node along one path, asks for a higher data sending frequency, and has a longer time-out value. This reinforced path can be found by sending data only to the node from which the interest response was first received at each hop.

During the last stage of the protocol, "*data delivery*" a node that was reinforced sends data to the sink at the data frequency specified in the reinforcement message. The data is sent along the single path that was established.

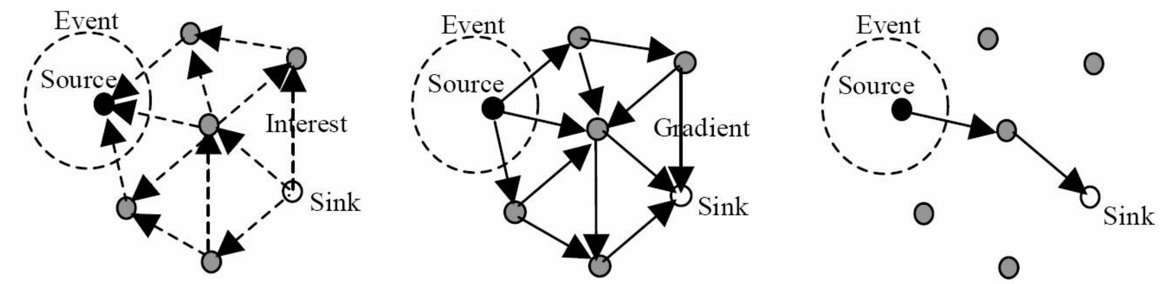

FIGURE 2.3. Phases of Directed Diffusion: Interest propagation, gradient setup and data delivery along reinforced path [28]

The problems of directed diffusion are the overheads in the protocol and the large memory requirements of the nodes. Interests and initial replies are flooded through the network, which means a lot of overhead energy is consumed. Also the nodes cache the interest requests and the data passing over them, this requires a lot of memory. Directed diffusion is not suitable for applications where all nodes send data at frequent intervals.

2.3.3. Rumor Routing

Rumor routing [29] is an improvement of directed diffusion algorithm. It uses a flat network structure. It is also a destination-initiated protocol using hybrid routing. Rumor routing floods the events but not the interest; if the number of events is small and the number of queries is large flooding the events creates an advantage. To flood an event, rumor routing creates packets that are called agents which have a certain time to live (TTL). When a node senses an event, it adds the event to its event table and floods an agent with a certain TTL. The agent contains a table of events observed by the node. As the agent is flooded through the network, the nodes update their event tables after receiving the agent. If an agent observes another event, it also updates its event table and propagates the new event along with the original event. An agent keeps a list of all nodes it has visited and as the next hop, chooses a neighbor that is not in the list. When an agent arrives at a new node, it decrements its TTL before it hops to another node. The agent is discarded and not sent further when its TTL is zero. When the sink sends an interest, the interest travels randomly until it finds a node with a path to the relevant event.

The protocol works efficiently on networks with few events and many interests, but does not provide energy efficiency in other kinds of networks. With the events table and the list of visited nodes, agent packet size can grow very large in networks with frequent events. Agents contain one route to each event, and if there are many interests for these events, the nodes over that route can finish their batteries quickly. Agents' choices for selecting their next hop affect the network lifetime because the queries are routed through that path.

2.3.4. Gradient-based Routing

Gradient-based routing (GBR) [30] is another improvement on directed diffusion. GBR uses a flat network structure and it is a destination-initiated protocol with reactive routing. GBR keeps the hop count when interest is diffused through the network. Each node keeps a parameter called the height of the node, that is, the number of hops required to reach the sink. The difference between a node's height and that of its neighbor is called the gradient and the packets are forwarded to the neighbor that has the largest gradient, which is the level of advancing to the sink. This is similar to the EAR protocols we will explain next, with the cost metric as the hop count to the sink.

GBR improves directed diffusion with data aggregation and traffic spreading. Traffic spreading can be done by three methods: The first method is choosing one of the neighbors that has the same gradient at random. The second method is to increase the height of the nodes whose energy drops below a threshold. The third method is routing new message streams through neighbors that do not already have a different message stream. Simulation results of GBR shows that it outperforms directed diffusion in terms of total communication energy. Interests are still flooded through the network; therefore, GBR has the same flooding overhead problem as directed diffusion.

2.3.5. Energy Aware Routing Protocol (EAR)

In the paper "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks" [31], Energy Aware Routing Protocol (EAR) is explained. EAR uses a flat network structure and is a source-initiated protocol with proactive routing. The protocol is used on the "Pico Radio" project of Berkley University. This protocol uses the cost metric as $C_{ij}= (e_{ij})^{\alpha}(R_i)^{\beta}$. Here Cij is the cost metric between sensors i and j, e_{ij} is the energy used to transmit and receive on the link, while R_i is the residual energy at sensor i normalized to the initial energy. The routing paths are formed as follows:

1. The destination sensor initiates the connection by flooding the network in the direction of the source sensor. It also sets the "Cost" field to zero before sending the request.
2. Every intermediate sensor forwards the request only to the neighbors that are closer to the source sensor than oneself and farther away from the destination sensor.
3. On receiving the request, the energy metric for the neighbor that sent the request is computed and is added to the total cost of the path.
4. Paths that have a very high cost are discarded and not added to the forwarding table. Only the neighbors N_i with paths of low cost are added to the forwarding table FT_j of N_j.
5. Sensor N_j assigns a probability to each of the neighbors N_i in the forwarding table FT_j, with the probability inversely proportional to the cost.
6. Thus, each sensor N_j has a number of neighbors through which it can route packets to the destination. N_j then calculates the average cost of reaching the destination using the neighbors in the forwarding table.
7. This average cost, $Cost(N_j)$ is set in the "Cost" field of the request packet and forwarded along towards the source sensor as in Step 2.

Their energy consumption model is a follows: Transmission used 20 nJ/bit + 1pJ/bit/m^3 (i.e. energy drop-off was r^3, which is a moderate indoor environment). The energy for reception was 30nJ/bit. These numbers are typical values for Bluetooth radios. The packets were 256 bits in size. The similar minimum cost forwarding schema is used in our algorithms with different cost metrics, as well as in other works as minimum cost forwarding algorithm (MCFA) [32], minimum energy communication network (MECN) [33], and small minimum energy communication network (SMECN) [34]. Some algorithms like Constrained Random Walks on Random Graphs [35] consider decisions of routing based on probabilities related to costs, to be able to spread the traffic and achieve load balancing.

The EAR algorithm performs 20 per cent to 40 per cent better than its competitor algorithm, diffusion routing.

2.3.6. Altruistic Energy Aware Routing Protocol (EAR+A)

This protocol is an improvement to the EAR protocol; it uses the notion of altruistic nodes that are willing to forward traffic in the name of their neighbors [36]. It is as well developed in the "Pico Radio" project of Berkley University. Like EAR, EAR+A uses a flat network structure and is a source-initiated protocol with proactive routing. The Altruistic nodes can be nodes with access to a power-line or a node's probability to become altruistic which can depend on its remaining energy, the number of altruists in its neighborhood, or the time elapsed since it was an altruist last time, etc. When a node receives a data packet and has to decide about the next data forwarder, it first looks up all of the possible neighbor nodes j and their respective costs c_j from the interest cache. The costs c_j of those upstream nodes j which are currently altruists (according to the altruist cache) are reduced by a fixed factor of $0 \leq \alpha \leq 1$ (called the cost reduction factor).

In the simulations, the network lifetime is taken as the time that 50 per cent out of the total number of nodes die due to energy depletion. The EAR+A scheme gives in the mean some advantage over EAR, the gain increases with the percentage of unconstrained nodes and reaches from 8.5 per cent up to 70 per cent. However, the altruist scheme is not always better; since with fixed unconstrained node percentages there are some random seeds, for which EAR gives a better network lifetime. In addition, it should be noted that these results are taken in relation to the existence of altruistic nodes with access to a continuous power supply like a power line.

2.3.7. Minimum Transmission Energy Algorithm (MTE)

The MTE algorithm is the competitor algorithm in many works [37], [38], [39], [40], [41], [42], [43]. In our algorithms as well, MTE is our reference for showing the performance of our algorithms. In this algorithm, every sensor calculates the energy required to receive a packet from another sensor and transmit it to the sink, and uses this path when an event happens in its sensing range or when it receives a packet to forward to the sink. If a sensor can directly transmit to the sink, it calculates the energy required to send the packet to the sink, and then checks with its neighbors if there is another path that expends a smaller total energy amount. In many usages of this algorithm, only transmission power is considered, and receiving power -which is as big as receiving power- is not considered. In our comparisons, the version that includes both transmission and the receiving power on a node is used. This makes it a more competitive algorithm and it is hard to outperform this version in the long run since it minimizes the total energy spent.

The algorithm uses the distributed Bellman-Ford algorithm as follows. Each Sensor i calculates its cost to its neighbors j ($Cost_{i,j}$) as total of transmission ($Et_{i,j}$) and receiving energies (Er_j). If a sensor has the sink in its communication range, it updates its cost to the sink ($Cost_{i,sink}$) as only the transmission energy to the sink ($Et_{i,sink}$). If it does not have the sink in its range, it assumes the cost as infinite. Then, for the maximum path length times, it updates its cost to the sink ($Cost_{i,sink}$) as the sum of its neighbor's cost to the sink ($Cost_{j,sink}$) and its cost to the neighbor ($Cost_{i,j}$), if this total is smaller than its previous cost to the sink. Then the routing to the sink is made through the neighbor that has the lowest cost. If this neighbor is the sink itself, the packet is sent directly to the sink.

MTE Algorithm

begin
for each $j \in Ni$ **do**
  $Cost_{i,j} = Et_{i,j} + Er_j$
end for
for each i **do**
  if $i \in Nsink$ **then**
    $Cost_{i,\,sink} = Et_{i,sink}$
  else
    $Cost_{i,\,sink} = \infty$
  end if
end for
for *maxpathlength* times do
  if $\text{Cost}_{i,sink} > \text{Cost}_{i,j} + \text{Cost}_{j,sink}$ **then**
    $Cost_{i,sink} = Cost_{i,j} + Cost_{j,sink}$
  end if
end for
end

The problem with MTE is, it puts a lot of load on the "best" path and drains the battery of the sensors on this "best" path very quickly. Therefore, the use of dynamic routing techniques is required.

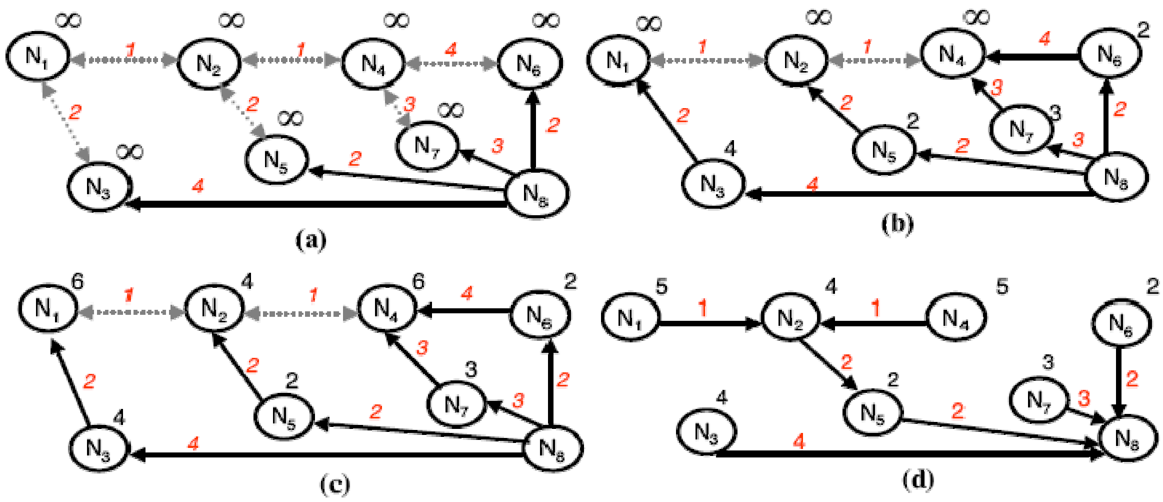

FIGURE 2.4. Operation of the distributed Bellman-Ford algorithm [44]

The example in Figure 2.4. shows the calculation of the shortest paths to N_8.

(a) Shortest-path metrics from each node are initially set to infinity.

(b) Shortest one-hop paths are calculated.

(c) Shortest two-hop paths are calculated.

(d) As all nodes are within three hops of N_8, the three-hop solution is the final one. Note that the metrics for N_1 and N_4 have changed, as a three-hop solution offers a shorter path than the two-hop solution.

MTE is a special case of EAR for α=1 and β=0. General implementations of MTE do not take into account the overhead energy. In "Overhead Energy Considerations for Efficient Routing in Wireless Sensor Networks" [43], improvements of the network lifetime are inspected if the overhead energy is considered during the routing process. Overhead energy is the Receiver Energy, the Computation Energy, and the Sensing Energy. The Minimum Transmission Energy protocol is used with and without the overhead energy considerations. It is shown that the network lifetime can be increased by 50 to 65 per cent by considering the overhead energy.

2.3.8. Low-Energy Adaptive Clustering Hierarchy (LEACH)

The LEACH protocol was developed within the μAMPS project at MIT. In the LEACH (Low-Energy Adaptive Clustering Hierarchy) [42], [45], [46] algorithm, the sensors organize themselves into local clusters, with one sensor acting as the local base station or *cluster-head*. LEACH uses a hierarchical network structure and is a source-initiated protocol with proactive routing. If the cluster heads were chosen a priori and fixed throughout the system lifetime, as in conventional clustering algorithms, sensors chosen to be cluster-heads static would die quickly, ending the useful lifetime of all sensors belonging to those clusters. Thus, LEACH includes randomized rotation of the high-energy cluster-head position such that it rotates among the various sensors in order not tı drain the battery of a single sensor. In addition, LEACH performs local data fusion to "compress" the amount of data being sent from the clusters to the base station, further reducing energy dissipation and enhancing system lifetime. Clusters are being re-created every round, and each node decides whether to become a cluster-head for the current round. The node picks a random number; if it is smaller than a threshold *T(n)*, the node becomes a cluster-head for that round. *T(n)* is the threshold value for each node *n*.

$$T(n)= \frac{p}{1 - p*(r \bmod \frac{1}{p})}$$ if the node has not been a cluster head in the last 1/p rounds (2.1)

LEACH reduces the communication energy by as much as 8 times compared with direct transmission and minimum transmission energy routing. The first node death in LEACH occurs over 8 times later than the first node death in direct transmission, minimum-transmission-energy routing, and a static clustering protocol, and the last node death in LEACH occurs over 3 times later than the last node death in the other protocols. Please note that the energy savings are due to aggregation of data. The problem with LEACH is that it requires direct communication to the sink node; LEACH is not designed for networks where the sink node is to be located outside the communication range of sensor nodes. Another problem is dynamic clustering overheads as head changes and advertisements may consume the energy that is gained from communication.

2.3.9. Power-Efficient Gathering in Sensor Information Systems (PEGASIS) and Hierarchical PEGASIS

PEGASIS [47] is a chain-based protocol and is an enhancement over the LEACH protocol. To save energy, nodes only communicate with their closest neighbors, and one of them communicates with the base station (sink) in one turn. PEGASIS has a hierarchical network structure and is a source-initiated protocol with proactive routing. After all nodes communicate with the base station in turns, one round ends, and a new round starts. Power required to transmit data per round is reduced and power draining is spread uniformly over all nodes. PEGASIS has two objectives. First, the lifetime of each node is increased by using collaborative techniques. Second, local coordination between nodes that are close together is allowed so the bandwidth consumed in communication is reduced. Unlike LEACH, PEGASIS does not form clusters; it organizes the nodes in chains and uses only one node to transmit to the base station instead of multiple nodes. Each node uses the incoming signal strength to measure its distance to neighbors to find the closest neighbor, and then adjusts its signal strength so that only that neighbor can hear it. The chain in PEGASIS will consist of those nodes that are closest to each other and form a path to the base station. Data from sensors moves from node to node, aggregated to each sensors' own data and eventually sent to the base station. The chain construction is made in a greedy way.

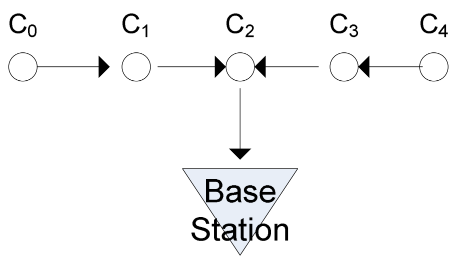

FIGURE 2.5. Chaining in PEGASIS

As presented in Figure 2.5., node c_0 passes its data to its closest neighbor node c_1. Node c_1 aggregates node c_0's data with its own data and then transmits to the leader of that turn. Node c_4 transmits its data to its closest neighbor node c_3. Node c_3 aggregates node c_4's data with its own and then transmits it to the leader. Node c_2 waits to receive data from both neighbors and then aggregates its data with its neighbors' data. Finally, node c_2 transmits one message to the base station. It is shown that PEGASIS performs up to three times better than LEACH for different network sizes and topologies. The performance gain is achieved without spending the overhead caused by dynamic cluster formation in LEACH and with decreasing the number of transmissions and reception using data aggregation. The disadvantages of PEGASIS are, there is excessive delay for distant nodes on the chain and the single leader can become a bottleneck. Like LEACH, PEGASIS also assumes that the sink is in the communication range of all of the nodes.

To decrease the delay of distant nodes, an extension to PEGASIS, called Hierarchical PEGASIS, is introduced in Lindsey et al. [48]. To reduce the delay, the use of simultaneous transmissions is proposed, and collusions are avoided with signal coding and spatial transmissions. Only spatially separated nodes are allowed to transmit at the same time. CDMA-capable nodes construct a chain of nodes that form a tree-like hierarchy, and each node at a particular level transmits data to a node in the upper level of the hierarchy as shown in Figure 2.6.

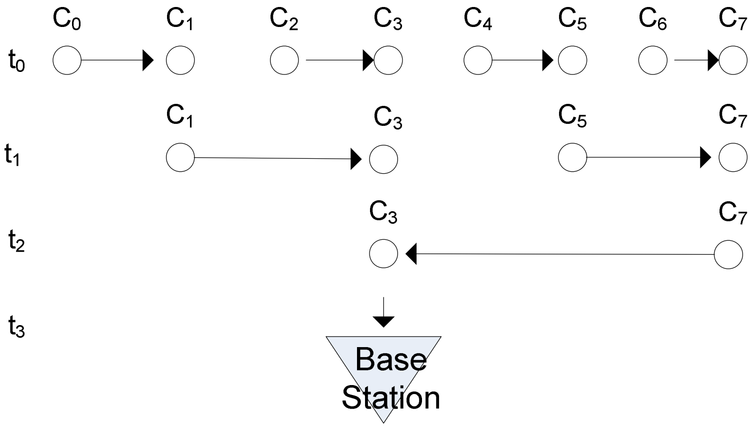

FIGURE 2.6. Chain hierarchy in Hierarchical PEGASIS

This method enables the parallel transmission of data and reduces the delay in PEGASIS: Hierarchical PEGASIS is shown to outperform the regular PEGASIS by about 60 per cent.

2.3.10. Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN) and APTEEN

The TEEN [49] protocol organizes sensors in multiple levels of hierarchy, and data is transmitted through the cluster heads to the base station. The cluster organization is similar to LEACH; the difference is in the data-sending schedule. Sensor nodes sense the medium continuously, but data transmission is done less frequently. The sensor nodes monitor the environment for changes in the sensing environment like temperature change, and respond by comparing the measured value with the hard and soft thresholds.

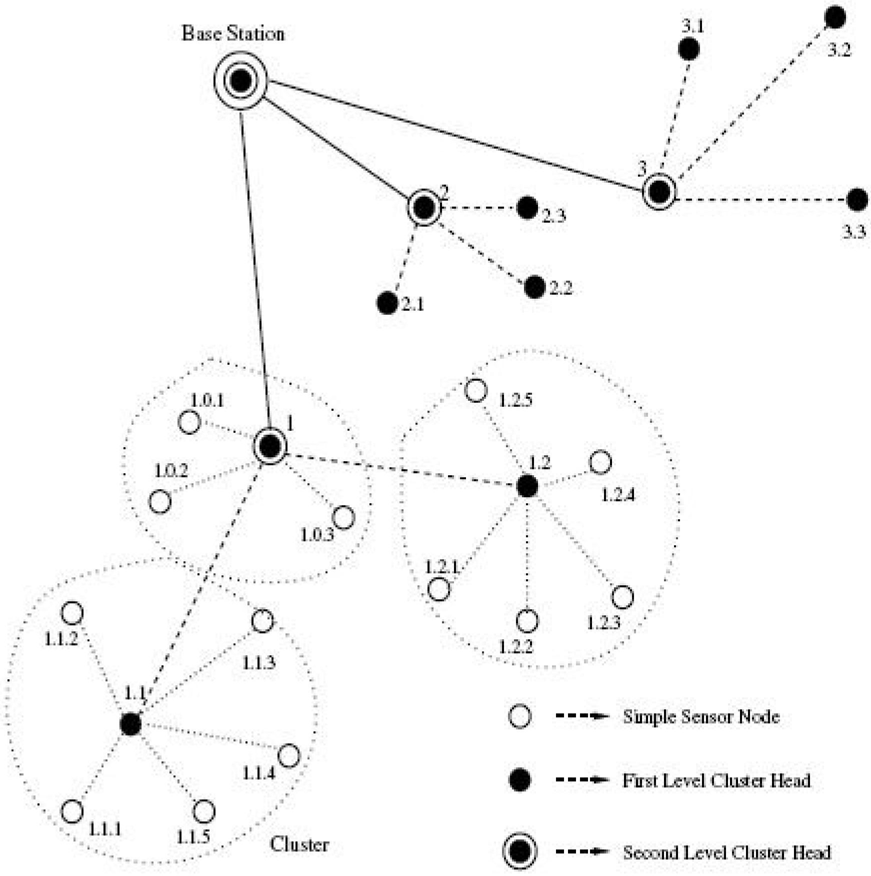

FIGURE 2.7. Hierarchical clustering in TEEN [49]

If the measured value exceeds the hard threshold limit, or if it is multiple of the soft threshold limit, the data is sent to the upper level hierarchy towards the base station. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the tradeoff between energy efficiency and data accuracy. Important features of TEEN include its suitability for time-critical sensing applications. Since message transmission consumes more energy than data sensing, the energy consumption in this scheme is less than in proactive networks. Since TEEN is based on fixed threshold limits, it is not suitable for periodic reports required by some applications. An advancement of TEEN is APTEEN [50], which is an improvement as it both services periodic inquiries and reports sensed-attribute changes.

2.3.11. Geographical and Energy Aware Routing (GEAR)

Some applications may require requesting information from some of the regions of the monitored area. The GEAR (Geographical and Energy Aware Routing) [51] algorithm has been developed to meet this requirement. GEAR uses location-based routing as the network structure. The process of forwarding a packet to all of the sensors in the target region consists of two phases:

Forwarding the packets towards the target region: GEAR uses a geographical and energy aware neighbor selection method to route the packet towards the target region. There are two cases to consider: When a closer neighbor to the destination exists, GEAR picks a next-hop sensor among all neighbors that are closer to the destination. When all neighbors are further away, meaning, there is a hole, GEAR picks a next-hop sensor that minimizes the cost value of this neighbor.

Forwarding the packet within the region: GEAR uses a Recursive Geographic Forwarding algorithm to forward a packet within the region. The region is divided into four sub-regions and four copies of the packet are sent to these regions. This splitting and forwarding process continues until regions with only one node are left. However, under some low-density conditions, recursive geographic forwarding sometimes does not terminate, routing uselessly around an empty target region before the packet's hop-count exceeds a limit. In these cases, GEAR uses restricted flooding.

GEAR's performance metrics are the number of data packets sent and successfully delivered before network partition and fraction of pairs still connected after partition. It is shown that for non-uniform traffic, GEAR delivers 70 per cent to 80 per cent more packets than its competitor GPSR. For uniform traffic, GEAR delivers between 25 per cent and 35 per cent more packets than GPSR.

3. SENSOR NETWORK PROBLEM DEFINITIONS

A sensor network is represented as a collection of *n* nodes arbitrarily distributed in the deployment area, which is an Euclidean plane R^2 in our thesis. A sensor network is modeled as a graph $G = (V, E)$ with the set of nodes $V \in R^2$ and the set of wireless undirected connections *E*. We assume that every node in the network has the same transmission range. The purpose of this thesis is to develop routing algorithms for wireless sensor networks to increase the network lifetime. Increasing the network lifetime depends on many factors. The sensor architecture, their hardware and their operating system affects lifetime. In addition, the network setup, the location of the sensors and their base station affects the lifetime. The network's operation aim, and the nature of the events being monitored are important criteria for choosing a routing schema.

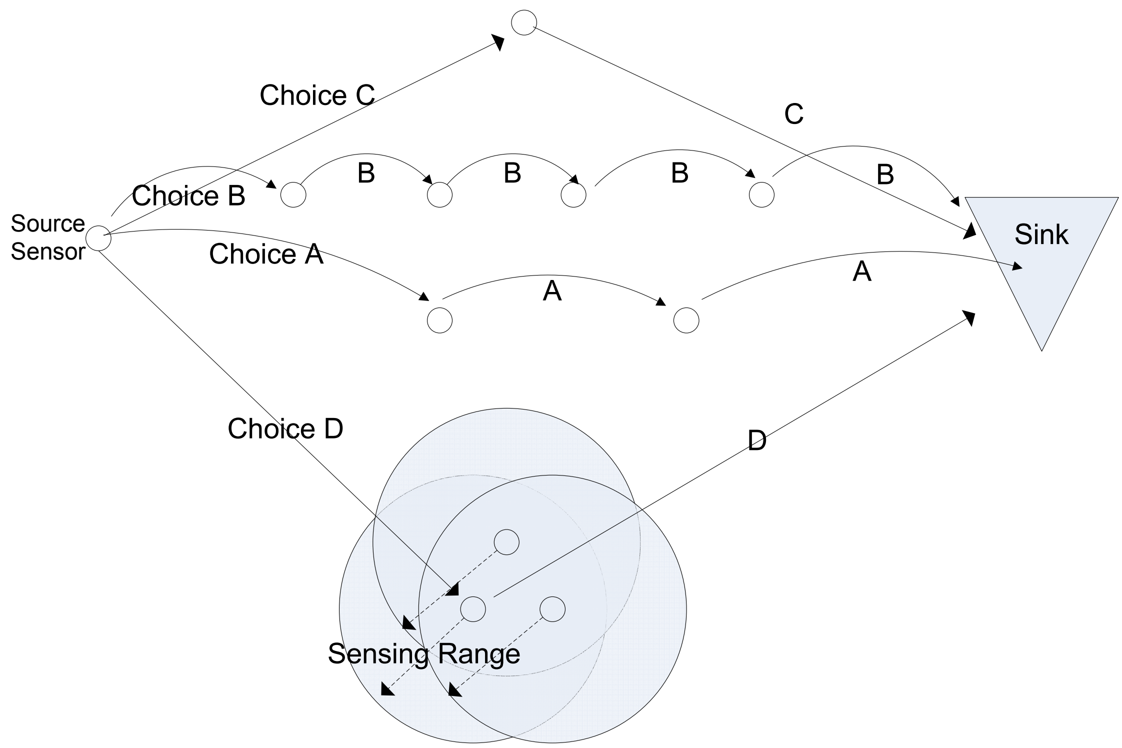

FIGURE 3.1. Alternative routing choices

This is an optimization problem of using limited batteries of sensors for reporting the events in the monitored area to the base station. In our work, we are offering solutions to spread the network traffic to all of the sensors in the region so that the sensors' energies are depleted late and the lifetime of the network is improved. Sensors can be located in good positions to become the next hop, because they may be shortening the hop count (Choice A in Figure 3.1), reducing the fixed energy expenditure for the transmitter energy and the receiving energy. They may be minimizing the energy used in the amplifier energy due to the path loss exponent, by offering short hops (Choice B in Figure 3.1). Alternatively, they may be located in a position away from the best route considering the last two factors, but they may be a feasible choice to forward the packets because their remaining battery levels can be high (Choice C in Figure 3.1). Some sensors may be deployed more densely in comparison with the other regions in the area, and they may be a good choice for forwarding the packets, because even if some of these densely deployed sensors die, the others may continue to monitor the same region (Choice D in Figure 3.1). Our algorithms aim to use a combination of these factors by assigning appropriate costs to choices and improve the network lifetime.

3.1. Energy Dissipation

In our thesis, we are working on event driven sensors. The sensors wait for an event to happen, and when an event happens in their sensing range, they forward the event information to the base station. In the sensing mode, their sensing hardware is working and it expends some energy. The sensing energy spent at sensor *i* is proportional to the time *t* passed.

$$Ew_{i,t}(t) = t * E_{waiting} \tag{3.1}$$

In our thesis, we assume that sensors are distributed randomly in an area to be monitored. They sense an event happening in their sensing range and forward this information to the sink, which is also placed randomly at some point in the area. If the sink is in their communication range, they may pass the packet directly to it, or alternatively they can forward the packet to another sensor in their communication range, to be passed to the sink.

The energy spent in Sensor *i*, for sending information of length *k* to Sensor *j* which is at distance *d* is $Et_i(k,d)$. Transmission energy has two parts, the transmitter electronics energy and the amplifier electronics energy. The transmitter electronics energy is similar to the receiving energy which is the energy needed for running transmitter electronics. Amplifier electronics energy is a multiple of packet length and some *path-loss exponent* (γ) of transmission distance. γ is two for ideal free space propagation that is the square of distance. In case there is attenuation on obstacles, γ can be three or up to five [52].

$$Et_{i,j}(k,d_{i,j}) = E_{t.elec}*k + E_{amp}*k*(d_{i,j})^{\gamma} \quad (3.2)$$

In addition, the sensor *i* can be an intermediate sensor that forwards information received from other sensors towards the base station. The forwarding energy spent in sensor *i*, for forwarding information of length *k* to sensor *j* which is at distance *d* is energy spent for receiving this packet from a previous sensor in the forwarding chain plus energy spent for transmitting it to *j*.

$$Etot_{i,j}(k,d_{i,j}) = Er_j(k) + Et_{i,j}(k,d_{i,j}) \quad (3.3)$$

For an intermediate sensor, energy spent for receiving a packet is proportional to the packet length, which is the energy needed for running receiver electronics.

$$Er_i(k) = E_{r.elec} * k \quad (3.4)$$

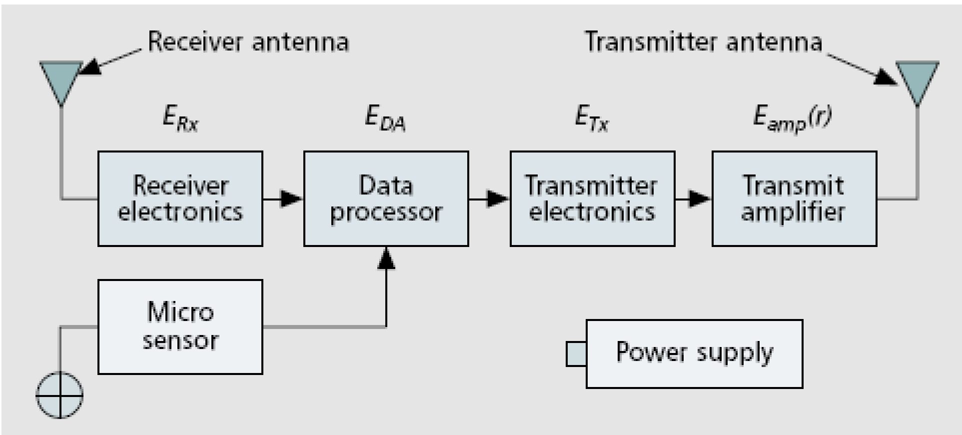

FIGURE 3.2. Simple Radio Energy Model [53]

We included all of the other energy consumptions as the waiting energy. The sensor expends its battery to the waiting energy, to the sending energy for the events that it has sensed, and to the forwarding energy for the events that it has forwarded towards to the base station. For any given time t, sensor i's remaining battery is the original battery $B_{i,0}$ minus the energy spent for the other tasks.

$$B_{i,t} = B_{i,0} - Ew_{i,t} - Et_{i,j} - Ef_{i,j} \quad (3.5)$$

When its battery power is diminished by all of these tasks, it is considered dead and is removed from the network. There are some formulations that consider the radio startup energies as well, but in our thesis, we have excluded them. We have also excluded the processing energy E_{DA} since they can be considered much smaller than the transmission energy.

3.2. Network Lifetime

When we make simulations, we should define what the network lifetime is. System lifetime can be considered as the time passed before the death of the first sensor in the network. Although this can be a metric, it should not be the only criterion to decide if one algorithm is better than another algorithm. Since the sensor networks consist of many sensors, they are robust to single or few sensor failures. For dense networks, few failures do not affect operation. Let us assume that in Network A, α number of sensors can monitor an area sufficiently. Assume another deployment is made in Network B in a similar area with $2^*\alpha$ number of sensors, with two sensors in the same position instead of one sensor in Network A. Network B can continue its operation as good as A even after one of the double sensors die. Therefore, networks can continue their operations even if a big number of their sensors die, if they were deployed densely enough.

$$NL_1 = \text{Time passes until one sensor dies in network} \quad (3.6)$$

Another metric can be the number of events that could not reach the base station (sink) in a given time. The events may not reach the base station (sink) because either they may happen outside the sensing range of live sensors or because the packets carrying the information about these events cannot reach the sink floowing the network being divided into disconnected sub-groups.

$$NL_2 = \text{\# of events that cannot reach the base station} \quad (3.7)$$

The most effective metric can be the percentage of the area that can be monitored compared to the first deployment. If a network can continue to monitor β per cent of the area after some time passes, this network can be considered fully functional. β can be 90 per cent or 95 per cent if it is critical for the application to not miss any event. This can be the case for mission critical applications like military surveillance or, for example, a nuclear plant leakage monitoring. However, if the samples collected from the area can give information of the general picture, and if the application is not time sensitive, this metric can be even 50 per cent. This can be the case for temperature monitoring for crops in a field.

There are many ways of calculating the coverage area. Some of the research like Onur et al. [54], [55] calculates the coverage with an *exposure-based model*, especially on intrusion detection applications. In this model, the sensing abilities of the sensors diminish as their distance to the target increases. This kind of model is used when the network has the aim of *barrier coverage* [56]. In our work, we assume binary sensing [57], that is, sensors have a fixed sensing range that has the same sensing quality, and they are assumed to sense all of the events happening in this sensing range and no event outside this sensing range. This kind of model is used in the applications that aim for *area coverage* [56]. Formula 2.3 gives our third network lifetime metric definition; in our simulations we use this metric with the requirement

$$NL_3 = \text{time t passed for reaching a specific } \frac{MonArea_t}{MonArea_0} \text{ rate} \quad (3.8)$$

$MonArea_t$ is the size of the area that can be monitored at a given time and $MonArea_0$ is the size of the area that could be monitored in the original deployment time.

4. PROPOSED ROUTING ALGORITHMS

After deployment, sensors make a network discovery with an exchange of information with their neighbor sensors. The find their relative positions by signal latency, by incoming signal strengths [58] and with the knowledge from the few GPS-enabled sensors. The set of neighbors in our algorithm are all of the sensors that are in the original sensor's communicating distance; the sensors are assumed to run network discovery algorithm and to have all of the required knowledge about the topology and the sensors around them. In some of the research on sensor networks, like the PRADA algorithm [59], it has been found out that considerable energy can be saved by trying to learn about the positions of not all of the neighbors but a sub-set of them. We have compared our results against the static routing algorithm the Minimum Transmission Energy Algorithm which just sends the packets through the paths that will expend the least energy.

4.1. Largest of Capacities Algorithm (LoC)

We have not used this algorithm in our simulations because its general performance is not better than the other algorithms. To show the relationship between the capacity of a sensor and the cost assigned to it, we decided that it would be good to explain this algorithm. Our motivation started with trying to calculate how many packets sensor *i* can transfer to *j*. This dynamic metric changes with time, because it depends on the residual energy. We have offered a routing algorithm that calculates the capacity of the sensors, that is, a metric that shows how many packets they can send to each of their neighbors with their residual energy. So, for a given time *t*, $Cap_{i,j}$ is *Remaining* $Battery_i$ over Energy required to send a packet to *j*.

$$Cap_{i,j} = Batt_i \,/\, [Et_{i,j}(k,d_{i,j}) + Er_i(k)\,] \qquad (4.1)$$

Our algorithm looks at the capacities of all sensors on the paths to the sink, and labels the paths with the Capacities on it. The algorithm then chooses the path with the largest capacity. This algorithm is called the Largest of the Capacities (LoC). It can be summarized as follows. Each Sensor *i* calculates its capacity to its neighbors *j* ($Cap_{i,j}$) as their current battery level over total of transmission ($Et_{i,j}$) and receiving energies (Er_j). If a sensor has the sink in its communication range, its updates its capacity to the sink ($Cap_{i,sink}$) as only its current battery level over the transmission energy to the sink ($Et_{i,sink}$). If it does not have the sink in its communication range, it assumes the capacity is zero. Then, for the maximum path length times, it updates its capacity to the sink ($Cap_{i,sink}$) as minimum of its neighbor's capacity to the sink ($Cap_{j,sink}$) and its capacity to the neighbor ($Cap_{i,j}$) if both its neighbor's capacity to the sink ($Cap_{j,sink}$) and its capacity to the neighbor ($Cap_{i,j}$) is bigger than its current capacity to the sink.

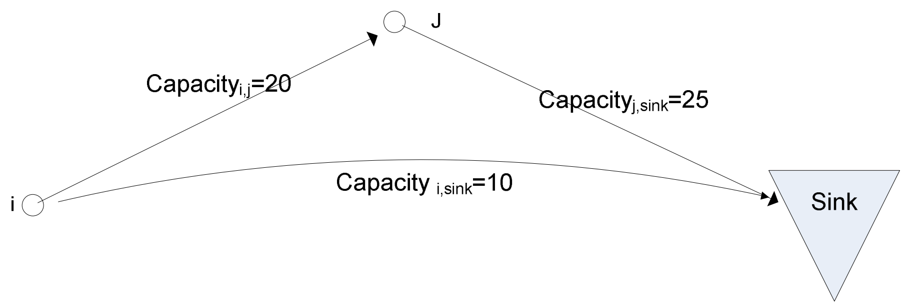

FIGURE 4.1. Largest of the Capacities- *i*'s routing table is updated from *i,sink* to *i,j*

Then the routing to the sink is made through the neighbor that has the highest capacity. If this neighbor is the sink itself, the packet is sent directly to the sink.

Figure 4.1 explains how node *i* updates its routing table for the best path to the sink from *i,sink* to *i,j.* Initially, node *i* has a direct route in its routing table to the sink in the first round of building the routing table. It has the capacity to send 10 units of information through that *i, sink* link and this forms the best path for one-hop routes that were discovered in the first round. In the first round, node *j* has the best path of *j,sink* with the capacity to send 25 units of information. In the second round, nodes exchange information about their capacities with their neighbors. Node *j* offers its capacity of 25 to node *i,* and node *i* discovers an alternative route of *i,j* to the sink, because it has a capacity of 20 units of information to node *j* and minimum of 20 and 25 is bigger than the old capacity, 10. The new capacity becomes 20, and node *i* starts to broadcast its capacity as 20 to its neighbors in the next rounds of building the routing table process. These rounds of updating neighbors with new capacities are repeated for maximum path length times. Maximum path length is the maximum number of hops that any "source to sink" path is allowed to have in the network.

Algorithm 1 LoC

```
begin
for each j ∈ Ni do
    Cap_{i,j} = Batt_i / ( Et_{i,j} + Er_i )
end for
for each i do
    if i ∈ Nsink then
        Cap_{i, sink} = ( Batt_i / Et_{i,sink} )
    else
        Cap_{j, sink} = 0
    end if
end for
for maxpathlength times do
    if Cap_{i,sink} < Cap_{i,j} AND Cap_{i,sink} < Cap_{j,sink} then
        Cap_{i,sink} = min(Cap_{i,j} , Cap_{i,sink} )
    end if
end for
end
```

The advantage of LoC is that it is supposed to perform better than the other algorithms until the first sensor death occurs. It performs better for small networks with limited battery. However for the networks with a large number of sensors, it uses the battery of the sensors over what is required and makes the sensors die very fast after some time passes. Therefore, we searched for other routing algorithms and we did not use this algorithm.

4.2. Minimum of the Used Percentages Algorithm (MUP)

Our previous algorithm LoC assigns values (Capacity) to sensors showing how many packets they can route. In LoC, we tried to choose the paths with the biggest Capacity. We could not use Bellman-Ford because of this reason. We needed to assign a cost to each connection to be able to use Bellman-Ford. We chose the cost value of the percentage of the remaining energy used for sending a packet.

$$Per_{i,j} = [Et_{i,j}(k,d_{i,j}) + Er_i(k)\ ] \,/\, Batt_i \quad (4.2)$$

$Per_{i,j}$ is in practice, $1/Cap_{i,j}$, where $Cap_{i,j}$ is the Capacity we used in the previous LoC algorithm. We call this algorithm "Minimum of the Used Percentages" (MUP). The routing algorithm calculates the routing table as follows: Each Sensor *i* calculates its cost to its neighbors *j* ($Cost_{i,j}$) as total of transmission ($Et_{i,j}$) and receiving energies (Er_j) over its current battery level. If a sensor has the sink in its communication range, it updates its cost to the sink ($Cost_{i,sink}$) as only the transmission energy to the sink ($Et_{i,sink}$) over its current battery level. If it does not have the sink in its communication range, it assumes its cost as infinite. Then, for the maximum path length times, it updates its cost to the sink ($Cost_{i,sink}$) as any of its neighbor's cost to the sink ($Cost_{j,sink}$) plus its cost to their neighbor ($Cost_{i,j}$) if this total is smaller than its previous cost to the sink ($Cost_{i,sink}$). Then the routing to the sink is made through the neighbor that has the lowest cost. If this neighbor is the sink itself, the packet is sent directly to the sink.

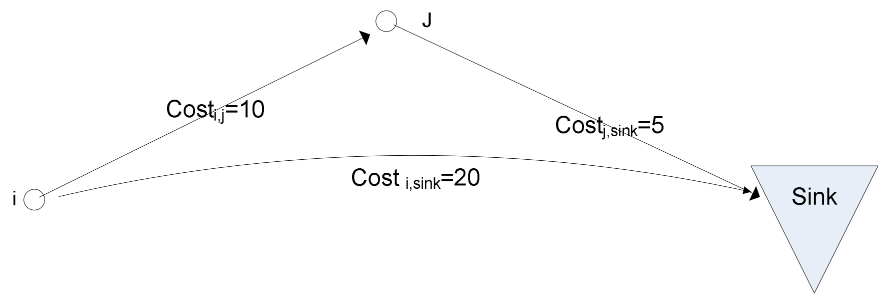

FIGURE 4.2. MUP- *i*'s routing table to the sink is updated from *i,sink* to *i,j*

Figure 4.2 explains how node *i* updates its routing table for best path to the sink from *i,sink* to *i,j*. Initially, node *i* has a direct route in its routing table to the sink in the first round of building the routing table. It has the cost of 20 units for the *i, sink* link and this is the best path for one-hop routes discovered in the first round. In the first round, node *j* has the best path of *j,sink* with a cost of 10 units. In the second round, nodes exchange information about their costs with their neighbors. Node *j* offers its cost of 5 to node *i,* and node *i* discovers an alternative route of *i,j* to the sink, because it has a cost of 10 units of information to node j and a total of 5 and 10 is smaller than the old cost of 20. The new cost of node i is decreased to 15, and node i starts to broadcast its cost as 15 to its neighbors in the next rounds of building the routing table process. Rounds of updating neighbors with new costs are repeated for maximum path length times. Maximum path length is the maximum number of hops that any "source to sink" path is allowed to have in the network.

Algorithm 2 MUP

---

begin
for each $j \in Ni$ **do**
    $Cost_{i,j} = (Et_{i,j} + Er_j)] / Batt_i$
end for
for each i **do**
    if $i \in Nsink$ **then**
    $Cost_{i,sink} = Et_{i,sink} / Batt_i$
    else
        $Cost_{i,sink} = \infty$
    end if
end for
for ***maxpathlength*** **times do**
    if $Cost_{i,sink} > Cost_{i,j} + Cost_{j,sink}$ **then**
        $Cost_{i,sink} = Cost_{i,j} + Cost_{j,sink}$
    end if
end for
end

In fact, this algorithm is very similar to the routing protocol of Pico Radio, Energy Aware Routing Protocol (EAR) [31]. EAR uses the metric $Cost_{i,j}$ given in formula 4.3.

$$Cost_{i,j} = e_{ij}^{\ \alpha} R_i^{\ \beta} \tag{4.3}$$

The weighting factors α and β can be chosen to find the minimum energy path, the path with nodes having the highest energy, or a combination of these [60]. EAR protocol uses this cost metric to assign paths to nodes that use them probabilistically according to their paths. In our case, α is one and β is minus one; the original work did not consider the case where β can take minus values, and report that they choose α as one and β as 50. They calculate R_i as residual energy at node i normalized to the initial energy; in our work we consider just the residual energy. Normalizing the residual energies should not create an advantage, and for the heterogeneous initial battery levels it may create disadvantages because what affects the remaining lifetime is not the sensor's initial battery level, but just the remaining energy. Another difference is that our algorithm does not use probabilistic routing; and instead it chooses the path that gives the least cost.

4.3. Projected Percentages Algorithm (PP)

The Minimum of the Used Percentages algorithm performs better than the other ones, however we wanted to add another parameter to the algorithm- number of packets that passed on the sensor. This is a good parameter for the busy sensors to make projections for the future. If a sensor has heavily loaded traffic and it is frequently being used for relaying other sensors' information to the sink, this sensor is busy and it should not be so willing to share its limited battery by being a relay to other sensors. When the traffic load over a sensor increases, it should decrease its willingness to relay messages by increasing its cost. Although a sensor could be the best choice for energy expenditure, it should not be very willing to accept all of the traffic. This parameter allows the diversion of some of the traffic to less busy sensors. The cost metric is called the Projected Percentage and it is:

$$ProjectedPer_{i,j} = [(Et_{i,j}(k,d_{i,j}) + Er_i(k)) * Packets_i] / Batt_i \qquad (4.4)$$

$Packets_i$ is the traffic load on node i, that is the number of packets that passed from node i since the deployment. The history of the traffic is kept from the beginning of deployment and recent traffic does not have an increased weight. This is due to the fact that the battery of the node is spent from the beginning of the deployment, and the recent traffic does not have a more valuable information than the older traffic. This algorithm is named Projected Minimum of the Used Percentages (PP). It works as follows: Each sensor i sets its initial packet count as one and it calculates its cost to its neighbors j ($Cost_{i,j}$) as the total of transmission ($Et_{i,j}$) and receiving energies (Er_j) multiplied by its packet count ($Packets_i$) over its current battery level. If a sensor has the sink in its communication range, it updates its cost to the sink ($Cost_{i,sink}$) as only the transmission energy to the sink ($Et_{i,sink}$) multiplied by its packet count over its remaining battery level. If a sensor does not have the sink in its communication range, it assumes the cost as infinite. Then, for the maximum path length times, it updates its cost to the sink ($Cost_{i,sink}$) as the sum of its neighbor's cost to the sink ($Cost_{j,sink}$) and its cost to its neighbor ($Cost_{i,j}$), if this total is smaller than its previous cost to sink. Then the routing to the sink is made through the neighbor that has the lowest cost. If this neighbor is the sink itself, the packet is sent directly to the sink. A sensor's packet count is incremented whenever a packet is generated from it or relayed over it.

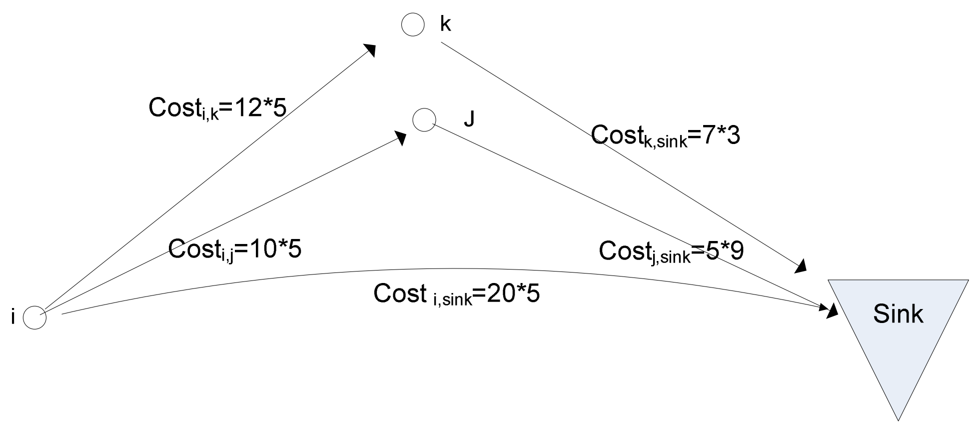

FIGURE 4.3. PP- *i*'s routing table to the sink is updated from *i,sink* to *i,k*

Figure 4.3 explains how node *i* updates its routing table for the best path to the sink from *i,sink* to *i,k.* Initially, node *i* has a direct route in its routing table to the sink in the first round of building the routing table. It has the cost of 20*5=100 units for that *i, sink* link and is the best path for one-hop routes discovered in the first round. 20 is the cost unit proportional to the energy expenditure cost of link *i,sink* over node *i*'s current battery, and 5 is the traffic load (number of packets sent and relayed through node *i*) on node *i*. In the first round, node *j* has the best path of *j,sink* with a cost of 5*9=45 units. 5 is relevant to energy expenditure of link *j,sink* over battery level of node *j* and 9 is the traffic load on node *j*. Node *k* has the best path of *k,sink* with a cost of 7*3=21 units. 7 is relevant to the energy expenditure of link *k,sink* over battery level of node *k* and 9 is the traffic load on node *j*. In the second round, nodes exchange information about their costs with their neighbors. Node *j* offers its cost of 45 to node *i,* and node *k* offers its cost of 21 to node *i.* Node *i* calculates the cost to sink through node *j* as 50+45=95 and the cost to sink through node k as 60+21=81. Node i chooses an alternative route of link i,k to the sink, because the total cost of 81 for link i,k is smaller than the old cost of 100 for link $i,sink$, and is also smaller than the other alternative total cost of 95 for link i,j. The new cost of node i is decreased to 81, and node i starts to broadcast its cost as 81 to its neighbors in the next rounds of building the routing table process. Please note that the route of node i would be updated as the path through node j if the MUP algorithm was used. Rounds of updating neighbors with new costs are repeated for maximum path length times. Maximum path length is the maximum number of hops that any "source to sink" path is allowed to have in the network.

Algorithm 3 PP

begin
for each $j \in Ni$ **do**
  $Packets_i = 1$:For the initial case, when there was no packet flow over sensors
  $Cost_{i,j} = [(Et_{i,j}(k, d_{i,j}) + Er_i(k)) * Packets_i] / Batt_i$
end for
for each i **do**
  if $i \in Nsink$ **then**
    $Cost_{i,sink} = Et_{i,sink} * Packets_i / Batt_i$
  else
    $Cost_{i,sink} = \infty$
  end if
end for
for ***maxpathlength*** **times do**
  if $Cost_{i,sink} > Cost_{i,j} + Cost_{j,sink}$ **then**
    $Cost_{i,sink} = Cost_{i,j} + Cost_{j,sink}$
  end if
end for
increment $Packets_i$ when a packet is transferred over $Sensor_i$
end

4.4. Remaining Battery over Area Sensed (BAS)

All of the algorithms we reviewed in the literature decided on the routing process according to the remaining battery and the costs of passing information to the next nodes.

However, they did not consider the area monitored for use in the routing decision; they propose using this phenomenon to hibernate the abundant sensors. In “Coverage-aware self-scheduling in sensor networks” [61], Lu J. and Suda T. offers to use a parameter called *sensing denomination* for deciding on the hibernation schedule. They define SD as “Since network coverage can be interpreted as the amount of information retrieved from the network, a sensor’s SD can be defined as the loss of network coverage caused by removal of the sensor from the network.” Slijepcevic S. and M Potkonjak [62] propose an algorithm for creating mutually exclusive sub-sets of sensors that give the maximum coverage at a given time. Cardei M. and Du Z. [63] improves Slijepcevic et al.’s algorithm for tracking a limited number of targets, by partitioning the set of all available sensors into disjoint sets such that each set covers all targets. They transform the problem into a maximum flow problem and they use mixed integer programming. They use this disjoint set of information again for the hibernation schedule. In our work, we use the overlapping of sensing coverage as follows.

In most of the cases, sensors are deployed abundantly and more than one sensor can monitor the same area. In this case, the duplicate sensors can be backups of each other, and they can sacrifice their battery if a nearby sensor is monitoring a similar area. Even if one of the sensors dies from excessive forwarding of others’ packets, the other can continue to sense the same area and general network performance is not reversely affected from the death of the sensor. On the other hand, if some sensors are uniquely monitoring some area, and there is no sensor near them to monitor that area, the survival of those sensors is much more important for the network in comparison to the survival of other sensors. We used this property in our novel routing algorithm called the Remaining Battery over Area Sensed (BAS).

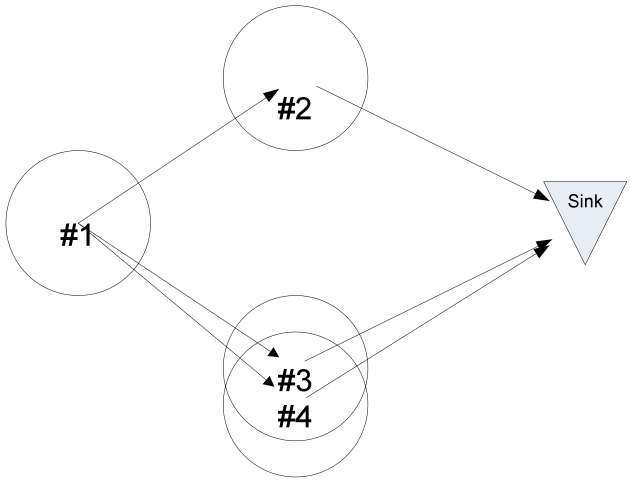

FIGURE 4.4. Sensors monitoring similar areas

Figure 4.4. shows four sensors and the areas that they can monitor. If the Sensor #1 has to send a packet towards the sink and it has to choose one of the Sensors #2, #3 or #4 as a forwarder, according to most of the routing algorithms, it will choose Sensor #2 or #3. However, since Sensors #3 and #4 are monitoring the same area, it does not matter if one of them dies earlier than the other nodes. They should be more eager to use their energy to be a forwarder. In fact, if they are very near to each other, like on top of each other, then they can even work like one sensor with double battery capacity. In the case shown in Figure 4.4, Sensor #2 should start to forward the information after the batteries of Sensor #3 and Sensor #4 fall to approximately half of Sensor #2's battery. This way, equal monitoring coverage in all of the areas can be achieved. In that case, if the batteries of all sensors are equal, the cost of forwarding packets in Sensors #3 and #4 should be half of the cost of Sensor #2. So every sensor's cost should be recalculated according to the batteries of the other sensors that can monitor a part of their monitoring area. Their contribution to each other's cost is shown in Figure 4.5.

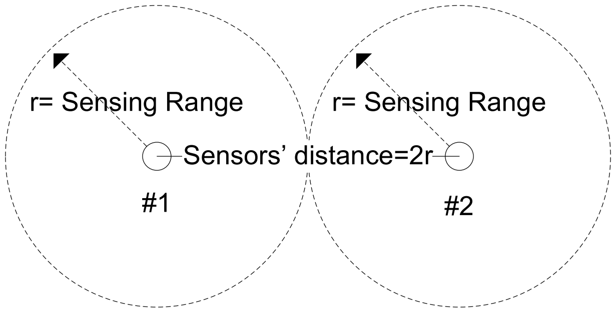

FIGURE 4.5. Sensors not monitoring each others' region

If the sensors are not monitoring any part of each others' region, they are independent. This is the case for sensors if their distance between each other is bigger than *2r*, *r* being the sensing range.

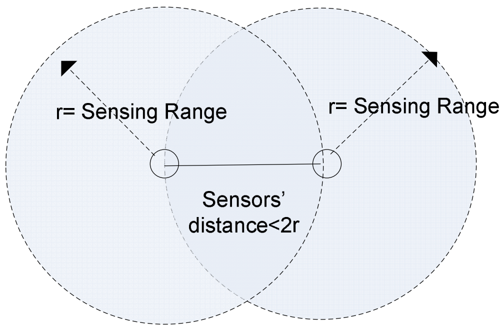

FIGURE 4.6. Sensors monitoring some part of each others' region

If the sensors are monitoring some part of each other's region, they are dependent, and their costs can be rearranged according to the area they monitor together. For example, if Sensor #1 has a battery of 10 Joules and #2 has a battery of 20 Joules. If the area they monitor together is $1/4^{th}$ of each other's whole monitoring area, their new costs should be arranged according to their theoretical battery power. Their normalized theoretical batteries are considered as: $NormBat_1$=10 + ½*(20 * ¼) = 12.5 Joules and $NormBat_2$=20 + ½* (10 * ¼) = 21.25 Joules. The ½ is the parameter δ, which defines how much the neighbor's batteries will contribute to the Normalized Battery. This way, the sensors' costs are reduced if they are in regions with high sensor density, and forwarding traffic is diverted through highly dense areas where the monitoring can be continued for a longer time. We can formulate it as:

$$NormBat_i = OriBat_i + \delta * \sum_{j \in \text{Sensing Neigborhood of } i} (OriBat_j) * (SensAreaPer(i, j)) \quad (4.5)$$

NormBat$_i$ is Updated Battery$_i$ which is the theoretical battery level of *i*, used in the cost computations

OriBat$_i$ is Original Battery level of sensor *i* at the time of computation.

δ is the weight factor affecting how much the neighbors will contribute to the sensor.

SensAreaPer(i,j) is the shared sensing area percentage of *i* and *j* over the whole sensing area *i*.

Sensing Neigboorhood of i contains the sensors *j* that are at a distance smaller than *Sensing Range*$_i$ + *Sensing Range*$_j$

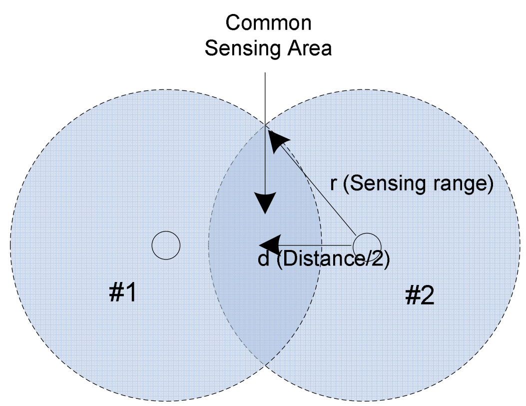

FIGURE 4.7. Common sensing area

We calculated *SensAreaPer(i,j)* as in formulation 4.6.

$$SensAreaPer(i,j) = 4 * \frac{\frac{Cos^{-1}(\frac{d}{r})}{2\pi} * \pi * r^2 \quad - \quad \sqrt{r^2 - d^2} * \frac{d}{2}}{\pi * r^2} \qquad (4.6)$$

where d= *Distance*$_{i,j}$*/2* and r = *SensRange*$_i$ (Sensing Range of Sensor *i*)

In this algorithm, we have defined the cost factor like in the Minimum of the Used Percentages algorithm, but we used Normalized Batteries instead of real batteries. That is

$$Cost_{i,j} = [Et_{i,j}(k,d_{i,j}) + Er_i(k)\ ] \,/\, NormBatt_i \quad (4.7)$$

The routing algorithm calculates the routing table as follows: Each Sensor *i* calculates its cost to its neighbors *j* ($Cost_{i,j}$) as total of transmission ($Et_{i,j}$) and receiving energies (Er_j) over its current normalized battery level ($NormBatt_i$). If a sensor has the sink in its communication range, it updates its cost to the sink ($Cost_{i,sink}$) as only the transmission energy to the sink ($Et_{i,sink}$) over its current normalized battery level. If it does not have the sink in its communication range, it calculates the cost as infinite. Then, for the maximum path length times, it updates its cost to the sink ($Cost_{i,sink}$) as the sum of its neighbor's cost to the sink ($Cost_{j,sink}$) and its cost to the neighbor ($Cost_{i,j}$), if this total is smaller than its previous cost to the sink. Then the routing to the sink is made through the neighbor that has the lowest cost. If this neighbor is the sink itself, the packet is sent directly to the sink.

Algorithm 4 BAS

begin
for each $j \in Ni$ **do**
  $Cost_{i,j} = Cost_{i,j} = [Et_{i,j}(k,d_{i,j}) + Er_i(k)\ ] \,/\, NormBatt_i$
end for
for each *i* **do**
  if $i \in Nsink$ **then**
    $Cost_{i,\ sink} = Et_{i,sink} \,/\, NormBatt_i$
  else
    $Cost_{i,\ sink} = \infty$
  end if
end for
for ***maxpathlength*** **times do**
  if $Cost_{i,sink} > Cost_{i,j} + Cost_{j,sink}$ **then**
    $Cost_{i,sink} = Cost_{i,j} + Cost_{j,sink}$
  end if
end for
end

4.5. Remaining Battery over Area Sensed with Traffic History (BAS-H)

We have tested one more upgrade to the previous algorithm, which is "Remaining Battery over Area Sensed with Traffic History". It works like the previous algorithm, with one exception: it keeps the traffic history of a sensor. This way, it aims to reduce the traffic on busy sensors. Its Normalized Battery is calculated as in formula 4.8:

$$NormBat_i = (OriBat_i / Packets_i) + \delta * \sum_{j \in \text{Sensing Neigborhood of } i} (OriBat_j * SensAreaPer(i, j)) / Packets_j \quad (4.8)$$

$Packets_i$ is the Traffic Flow over Sensor i. It is equal to the packets that passed over Sensor i at a given time from the beginning of the network deployment.

The other parameters and the algorithm's flow are the same as the BAS algorithm. Each Sensor i calculates its cost to its neighbors j ($Cost_{i,j}$) as total of transmission ($Et_{i,j}$) and receiving energies (Er_j) multiplied by its packet count ($Packets_i$) over its current normalized battery level ($NormBatt_i$). If a sensor has the sink in its communication range, it updates its cost to sink ($Cost_{i,sink}$) as only the transmission energy to the sink ($Et_{i,sink}$) multiplied by its packet count ($Packets_i$) over its current normalized battery level. If it does not have the sink in its communication range, it assumes the cost as infinite. Then, for the maximum path length times, the sensor updates its cost to the sink ($Cost_{i,sink}$) as its neighbor's cost to the sink ($Cost_{j,sink}$) plus its cost to the neighbor ($Cost_{i,j}$), if this total is smaller than its previous cost to the sink. Then the routing to the sink is made through the neighbor that has the lowest cost. If this neighbor is the sink itself, the packet is sent directly to the sink.

Algorithm 5 BAS-H

```
begin
for each j ∈ Ni do
    Packets_i=1      :For the initial case, when there was no packet flow over sensors
    Cost_i,j = [(Et_i,j(k,d_i,j) + Er_i(k) ) * Packets_i ] / NormBatt_i
end for
for each i do
    if i ∈ Nsink then
        Cost_i, sink = Et_i,,sink * Packets_i / NormBatt_i
    else
        Cost_i, sink = ∞
    end if
end for
for maxpathlength times do
    if Cost_i,sink > Cost_i,j + Cost_j,sink then
        Cost_i,sink = Cost_i,j + Cost_j,sink
    end if
end for
increment Packets_i when a packet is transferred over Sensor_i
end
```

5. SIMULATIONS AND RESULTS

5.1. Network Model

In our network, we assume that the nodes are deployed to a square area with the size of $x*x$; in our simulations the default value of x is 100 meters.100 sensors are deployed uniformly random distributed to this area, for example, as if they have been thrown from a plane. We place the base station at the center of the area by default. We have made tests with other positions of the sink inside or near the boundaries of the area. The energy expenditure in Node i per unit information transmission from Node i to j is assumed to be

$$Et_{i,j}(k,d_{i,j}) = E_{t.elec}*k + E_{amp}*k*(d_{i,j})^{\gamma} \quad (5.1)$$

where $E_{t.elec}$= 50 nJ/bit and E_{amp}= 100 pJ/bit/m^3. In addition, the energy expenditure in Node j per unit information receiving from Node i to j is assumed to be

$$Er_i(k) = E_{r.elec} * k \quad (5.2)$$

Where $E_{r.elec}$= 150 nJ/bit. These values are very similar to the ones in the energy consumption model used in Chang et al. [37]. The sensors' sensing range (R_s) is assumed to be 10 meters and their communication range (R_c) is assumed to be 20 meters. The sink is considered to have an unlimited power supply. In addition, the sensors are assumed to have batteries with 0.2 Joules energy capacity. Events happen at a rate of, on average, one event per minute uniformly random distributed between zero and two events per minute, at a randomly selected point in the area. The sensors that sense the event in their sensing range collect information about the event and send it to the base station. If they have no event to report, they wait until the next event happens and they spend idle waiting energy (Ew_i) of 50 nJ/min.

Packets generated from event monitoring are 128 bits; sensors forward the event information to the sink without any lag, so aggregation is not used. In some applications of MTE routing algorithms, simulations do not consider the receiving energy or the waiting energy. In our simulations, we consider both for all our algorithms. We have tested the success of our algorithms with the network lifetime metric as γ per cent of the original sensing area is continued to be monitored, and we tested the cases where γ is equal to 98, 95 and 90. We made ten simulations for each random network setup and compared the routing algorithms performances in the same area with the same events. Simulations were made with code written in C++; their running time differed from one minute to several hours on a computer with a 1.86 GHz Pentium M CPU. The running time mainly depended on the network size, number of nodes deployed and energy model parameters.

5.2. Simulations with Different Network Parameters

5.2.1. Network lifetime with different number of sensors deployed

In the following simulations, the effects of deploying different number of sensors on the network lifetime are measured. Different number of sensors also means different node densities, so the effects of node density on the network life are also inspected. Simulations were made with the fixed parameters in Table 5.1:

TABLE 5.1: Default simulation parameters and changed values for different numbers of sensors

PARAMETER	VALUE IN THE SIMULATION
Area:	100x100 meters
Number of Deployed Sensors	Effect is measured; tested values are 50, 100, 150, 200, 250 and 300
Initial Battery Capacity:	0.2 Joules
Sensing Range:	10 meters
Communication Range:	20 meters
Path Loss Exponent γ:	3
$E_{t.elec}$:	50 nJ/bit
$E_{r.elec}$:	150 nJ/bit
E_{amp}:	100 pJ/bit/m^3
Sensor Placement	Random (Uniformly Distributed)
Network lifetime parameter	90 per cent coverage
Sink Placement	Center
Simulations run	10 times

Different numbers of nodes are deployed to the same 100x100 meters square area. When more sensors are deployed the network lifetime increases. This reason for this result is mainly the distribution of relaying energy cost to more sensors with the increased number of sensors. Relaying other sensor's information is a big energy expenditure, and when there are more sensors, this expenditure is spread over many sensors, so the sensors die later. Figure 5.1 shows the performances of algorithms Minimum Transmission Energy (MTE), and the dynamic algorithms Minimum of the Used Percentages (MUP), Projected Percentages (PP), Remaining Battery over Area Sensed (BAS), and Remaining Battery over Area Sensed with History (BAS-H) with the network lifetime covering 90 per cent of the original coverage area with the number of nodes changing.

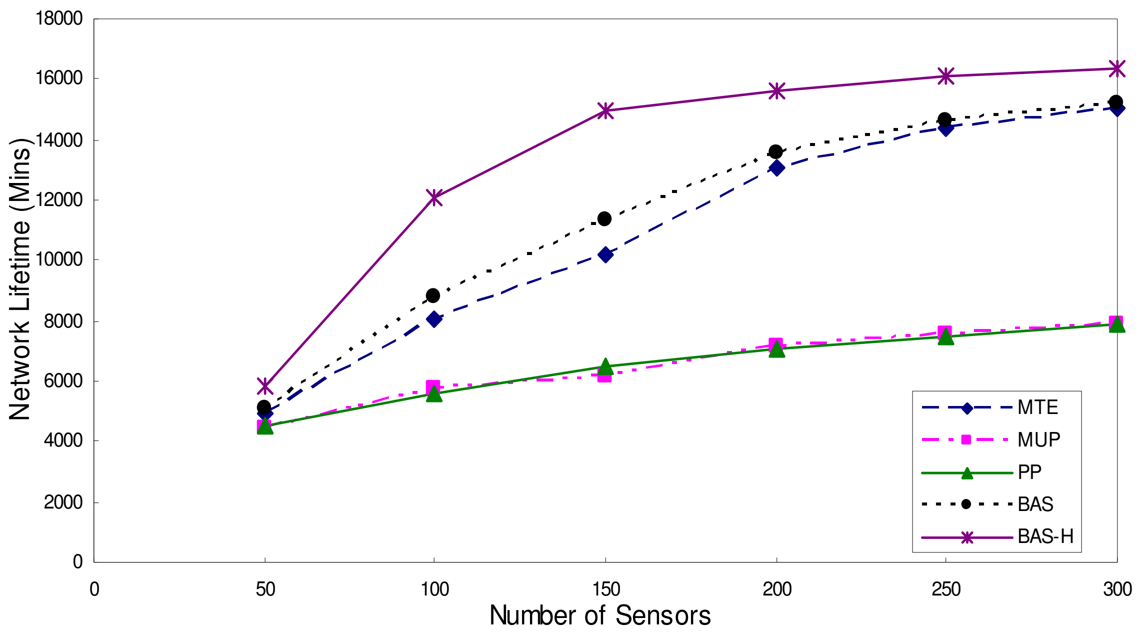

FIGURE 5.1. Network lifetime versus number of sensors deployed

If less than 50 sensors are deployed, in most of the cases more than 10 per cent of nodes are disconnected in the deployment time. Therefore, we started our range from 50 sensors. We can see that the dynamic algorithms BAS-H and BAS perform the best, with MTE in the middle and MUP and PP algorithms performing badly in a very similar way. As expected, the increased number of sensors bring longer lifetime, but the performance increase is saturated after about 200 sensors. We ended our simulations at 300 nodes due to this reason. When we consider the cost of the sensor network, the best lifetime/cost performance would be received somewhere between 100 and 200 sensors for that area. These means a sensor density of 8-16 for the communication range and 3-6 for the sensing range.

5.2.2. Network lifetime with different locations for the sink and different network lifetime parameters

Sink location planning is an important problem in the wireless sensor networks. One of the problems is "finding the best sink locations"- given the sensor network topology. "Minimizing the number of sinks for a predefined minimum operation period" and "Finding the minimum number of sinks while maximizing the network life" are other problems [64]. In this work, we do not offer a solution to these problems; we have made our simulations with a single sink. We simply intend to compare performances of the proposed algorithms with different sink locations. In the following simulations, the effects of the different positions of sink to the network lifetime are measured. The deployment area is 100x100 meters square and the sink is placed in the middle of the area, in the middle of a side, in the corner, 10 meters outside a side and 5x5 meters outside the corner. Simulations were made with the fixed parameters in Table 5.1 and the changed parameter values parameters are given in Table 5.2. The sink (base station) positions are presented in Figure 5.2.

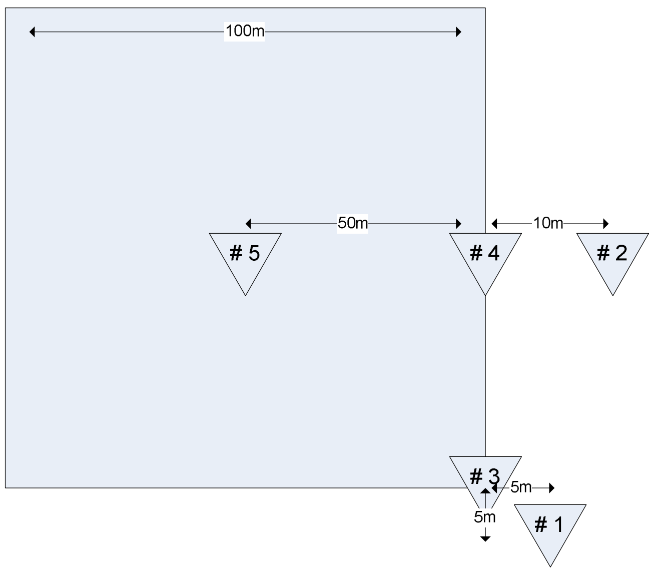

FIGURE 5.2. Positions for the different placements of the sink

TABLE 5.2: Changed simulation parameters for different locations of the sink

PARAMETER	VALUES IN THE SIMULATION
Network lifetime parameter	Tested values are 98, 95 and 90 per cent.
Sink Placement	Tested positions are in the center, on the side and corner, outside the side and corner

Figure 5.3 shows the network lifetimes for the algorithms with the network lifetime parameter covering 98 per cent of the original coverage area for different placements of the sink. As the sink is placed closer to the center, the network lifetime increases due to the following reasons. The traffic around the sink node is higher than for the other areas, and if it is in a central position and if some of the hot-spot sensors' energy is small, the traffic can be re-routed around these busy sensors. For the close to the center cases, the mean distances from the nodes to the sink also fall, so there is less energy consumption. When the sink is away from the center, there are fewer choices of sensors for relaying the traffic and the heavily loaded sensors around the sink die quicker. It can be seen that placing the sink in the corner gives shorter network lifetimes than placing the sink on the middle of a side.

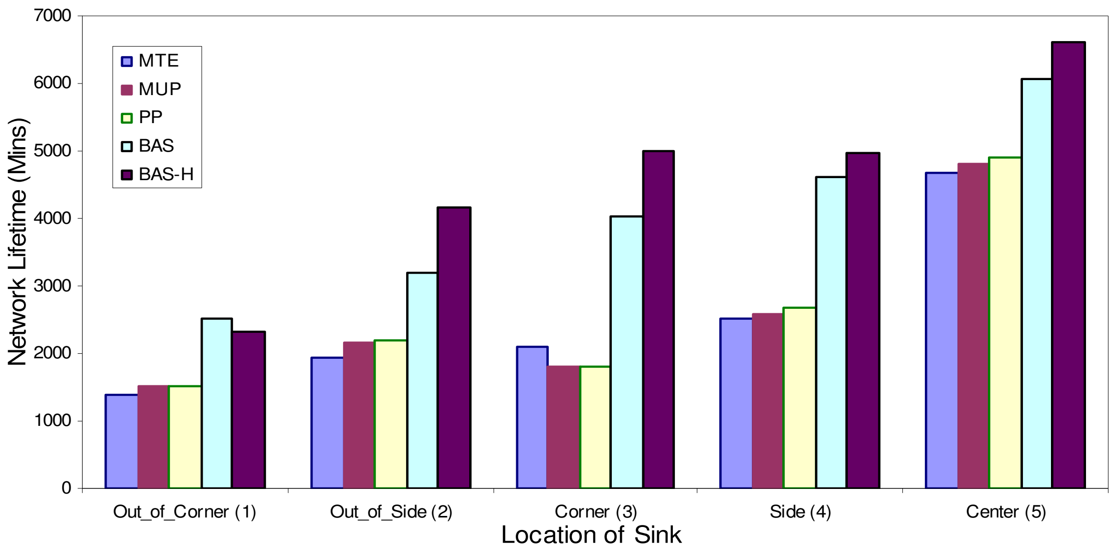

FIGURE 5.3. Network lifetime versus position of the sink for 98 per cent coverage

When the performances are compared, it is seen that the MUP and PP algorithms perform similar to MTE, where BAS and BAS-H perform 50 per cent to 150 per cent better then MTE for 98 per cent coverage. Another interesting result is that BAS performs better for the case with the sink is placed outside the corner. This result can be explained as follows: The difference between BAS and BAS-H is the counting of the number of packets passing on the sensors. The nodes around the sink are the most busy ones and they deplete their energy earlier than the other nodes. For all other node deployments, traffic load can be diverted and re-routed through less busy sensors around the sink and the most critical energy gains can be made around the sink. For the case of outside the corner, there are little choices to re-route the traffic. For this reason, the BAS-H algorithm, which tries to re-route traffic over fewer choices, performs worse than the plain BAS algorithm because it gives less effective path choices.

The coverage of 98 per cent means that the network is considered trustable until just a few sensors die. The dynamic algorithms try to postpone the first sensor deaths out the expense of re-routing traffic over more energy expending paths. They are expected to perform better in the first node deaths. However, for the latter deaths, they gradually perform less advantageously than the MTE, and after more sensors die, they come to a point where they crash. Figure 5.4 shows the network lifetimes for 95 per cent coverage.

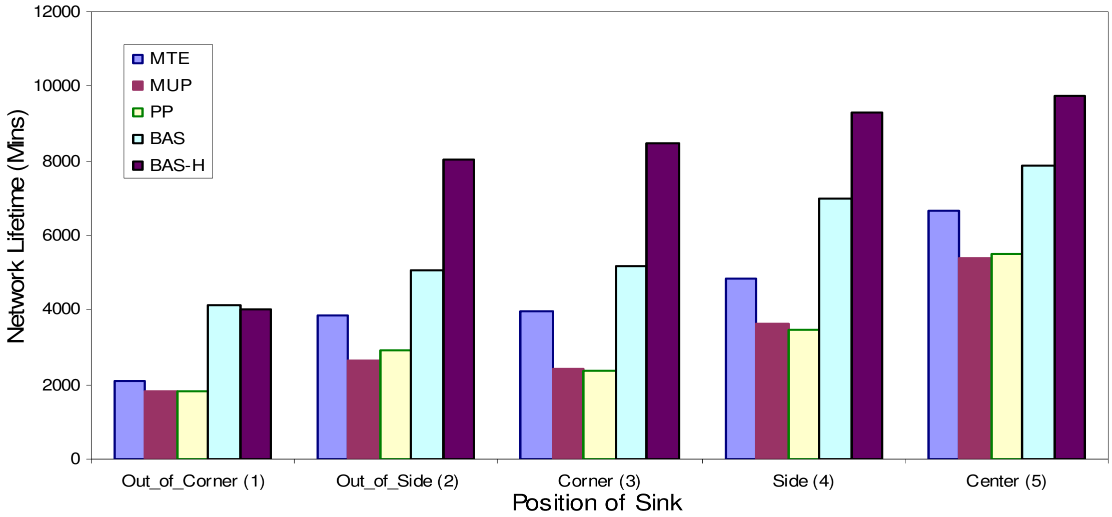

FIGURE 5.4. Network lifetime versus positions of the sink for 95 per cent coverage

For 95 per cent coverage, compared to 98 per cent coverage, MUP and PP algorithms start to perform worse; BAS and BAS-H algorithms' performance is less dominant. 95 per cent coverage means nearly the first 5-10 of the 100 sensors spend all of their energy. The number of sensors giving 95 per cent coverage depends on their location and the overlapping of the sensors with depleted energy.

When the 90 per cent coverage parameter is inspected, the network lifetimes continue to increase as expected, and dynamic algorithms continue to lose their advantages against MTE. It is an expected result, since MTE is spending the least total energy in the operations. The disadvantage of MTE is just that it does not delay the first node deaths, but when the total used energy is considered, MTE spends less energy than the other algorithms. The dynamic algorithms are designed to outperform MTE for delaying the first node deaths. Dynamic algorithms can also divert the traffic through sub-regions where the sensors are more densely deployed and, by this way, they can conserve maximum coverage for longer periods of time. Dynamic algorithms succeed in these objectives at the expense of choosing paths that expend more energy. For this reason, if we consider the maximum possible network lifetime as the time passes for all of the sensors to deplete their energies, the dynamic algorithms perform better than the MTE only at the beginning of this maximum possible network lifetime. If the maximum possible network lifetime is 10 units of time, dynamic algorithms perform better in the first few units of time that passes after deployment. After this limited time, the MTE algorithm has more active sensors and better coverage than the dynamic algorithms, because it spends less energy for the same routing tasks.

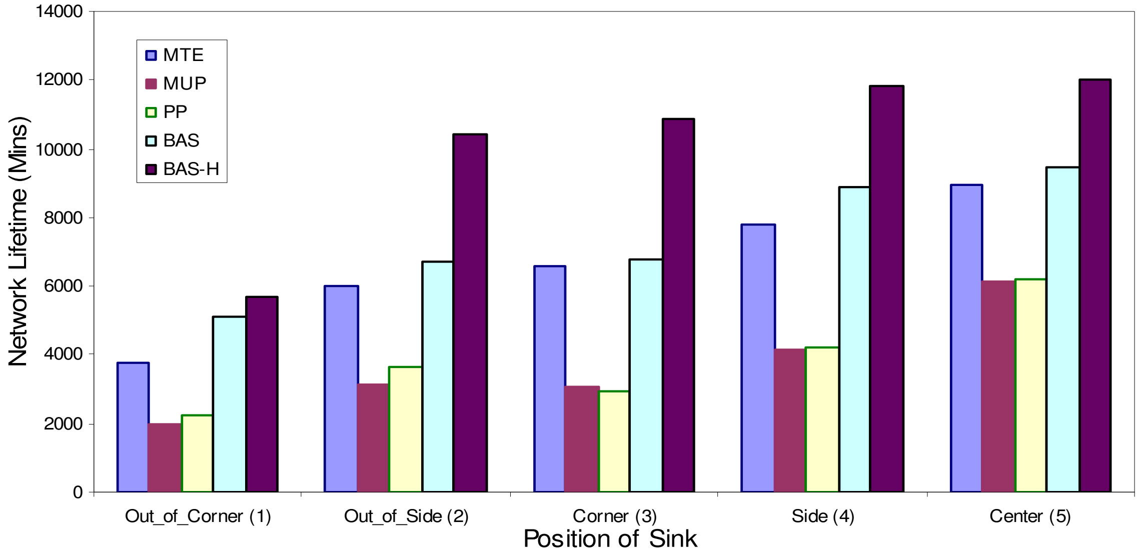

FIGURE 5.5. Network lifetime versus positions of the sink for 90 per cent coverage

Figure 5.5 compares the proposed algorithms' performances normalized to MTE. After 90 per cent coverage, we can expect the MTE to perform better than the other algorithms, because the dynamic algorithms were using more energy than MTE in trying to delay the node deaths.

5.2.3. Network lifetime with different shapes of deployment area

In the following simulations, the effects of different shapes of deployment area to the network lifetime are measured. Different area setups from 40x250 meters rectangle to 100x100 meters square or 57 meters radius circular area are inspected. MUP and PP algorithms are not inspected since their performance is seen to be worse than our competitor MTE algorithm in the previous simulations. Simulations were made with the changed parameter values in Table 5.3.

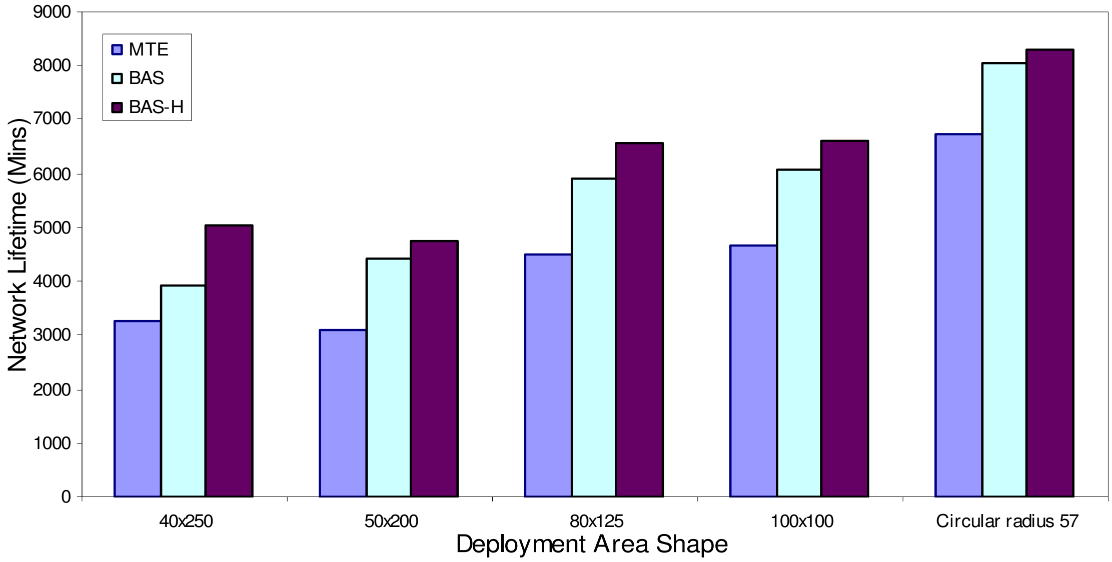

FIGURE 5.6. Network lifetime versus deployment area shape

TABLE 5.3: Different shapes of deployment area in the simulation

PARAMETER	VALUES IN THE SIMULATION
Area Type:	Effect is measured, tested area types are 40x250 meters rectangular, 50x200 meters rectangular, 80x125 meters rectangular, 100x100 meters square, and circular area with 57 meters radius.

The changing parameter is the shape of the deployment area. It is seen that performance increases when the deployment area is changed from long and thin to square or circular shapes. The mean path lengths are longer in long rectangular shapes than in the square shape, and that for the circular shape, path lengths are the shortest. For this reason, if the objective is to monitor maximum meter square surfaces with the same energy expenditure, or if it is to monitor the same meter square surfaces by spending the least energy, circular shapes give better results. If our objective is to monitor some specific square meters of an area and if we have the chance of selecting the area shape, we should choose a circular shape instead of a long and thin shape. Again with the shape of the surface being changed, it is seen that BAS-H and BAS perform better than MTE in all of the tested shapes.

5.2.4. Network lifetime with changing Traffic Load

In all of the previous simulations, the traffic load is fixed; the events are generated at a rate of, on average, one event per minute. The inter-arrival time between the events is between zero and two minutes, and the distribution is uniformly random with the average of one minute. The events happen at a random coordinate in the deployment area. The sensors that are located near the event location within their sensing range can sense the event. All of the sensors that sense the event send their information about the event to the sink. The traffic is generated in this way. Different traffic rates are tested with different event generation rates. Average event rates between 0.1 to 10 events per minute have been tested. The range of differentiation is between zero to 2xAverage Rate and there is uniform distribution in these ranges. The simulations gave network lifetimes as presented in Figure 5.7.

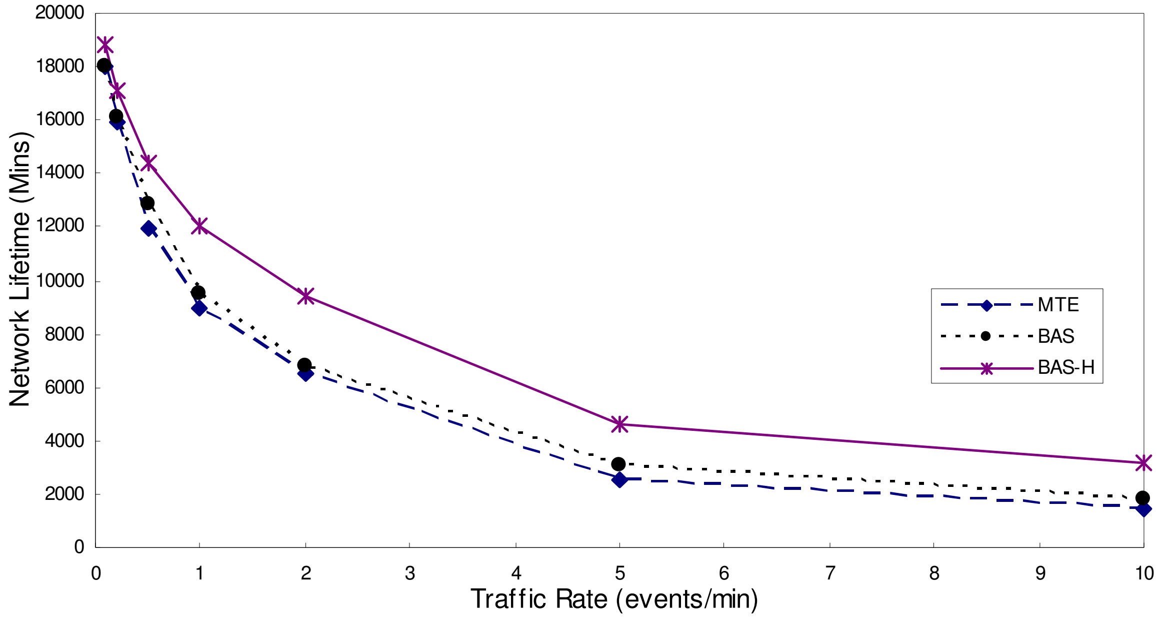

FIGURE 5.7. Network lifetime versus traffic rate

When the traffic rate is very low, that is less than one event per minute; the fixed waiting energy becomes effective and the most important expenditure, therefore the difference in the lifetimes becomes much less obvious. This is because most of the energy is spent waiting, and the difference of using effective routing does not create differences in the network lifetime. In changing the traffic rate as well, BAS-H performs the best; BAS only performs a little bit better than MTE however this small advantage is kept consistently. The advantage of BAS to MTE is seen as smaller because the range of network lifetime is 20,000 minutes, which is bigger than the other graphics, so the difference in results may seem to be smaller in Figure 5.7.

5.2.5. Network lifetime with different Sensing Ranges

In the following simulations, the effects of different sensing ranges to the network lifetime are measured. Simulations were made with the changed parameter values in Table 5.4:

TABLE 5.4: Different sensing ranges in the simulation

PARAMETER	VALUES IN THE SIMULATION
Sensing Range:	Effect is measured, values are 3, 5, 8, 10, 13, 15 meters

When the sensing range is small like 3 meters, the network seems to have the longest lifetime. In fact, in these cases, most of the events are not sensed, and sensors do not spend energy reporting the events to the sink.

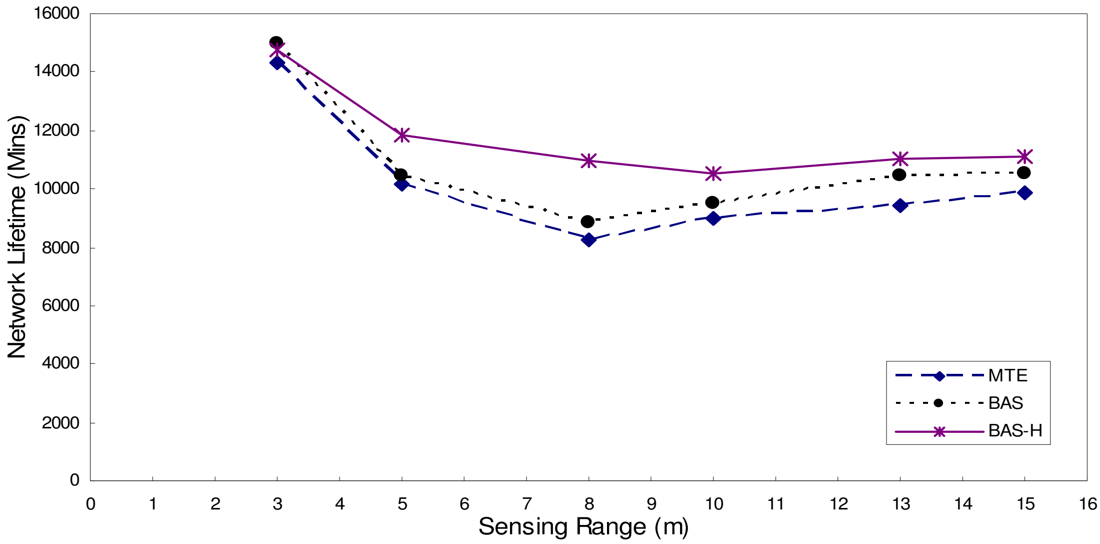

FIGURE 5.8. Network lifetime versus sensing range

The events are happening with at the same rate in a random place in the network. The sensors may detect the events or not, as time passes. If the events are not detected, sensors only spend waiting energy. If the events are detected, then the information is sent to the base station, and sensors spend more of their energy for the transmission of data. For the small sensing range, the sensors generally die spending their energy waiting for the events to happen because the network sensing coverage is less than 27 per cent of the deployment area when the sensing range is 3 meters. When the range is increased, this effect is lost and they start to detect more of the events so the lifetime drops. After the sensing range is more than 10 meters, the lifetime starts to increase, as then, more sensors start to backup each other by sensing the same regions, and for the coverage to decrease to 90 per cent of the original coverage area, more sensors need to deplete their energies. Since this takes more time, the network lifetime is again increased when the sensing range is bigger than 10 meters. Although more sensors may be dead in those cases, their coverage is increased and because the network lifetime depends on the coverage, lifetime is increased. This effect can be seen in Figure 5.8. The BAS-H algorithm performs consistently better than the others and the BAS algorithm performs better than MTE.

5.2.6. Network lifetime with different Communication Ranges

In the following simulations, the effects of different communication ranges to the network lifetime are measured. Simulations were made with the changed parameter values in Table 5.5.

TABLE 5.5: Different communication ranges in the simulation

PARAMETER	VALUES IN THE SIMULATION
Communication Range:	15, 17, 19, 20, 23 ,25, and 30 meters

When the communication range is small like 15 meters, most of the network setups with random sensor deployments have coverage of less than 90 per cent in the first round because about 10 per cent of the sensors are disconnected with the sink. Therefore, the average network lifetime is small in these cases. When the communication range is more than 17 meters, the network is fully connected from the beginning and there is a slight increase in the network lifetime with longer communication ranges, as there are more different alternatives by which to route the packets. In addition, when the communication range is longer, the information can be sent to the sink with fewer hops. Fewer hops or direct communication minimizes the extra energy spent in the fixed cost of transmission and receiving energies $E_{t.elec}$ and $E_{r.elec}$. The total of $E_{t.elec}$ and $E_{r.elec}$ are as big as the amplifier energy E_{amp} spent for 12.5 meters communication, so longer communication ranges are more feasible because by this way, the number of hops required to reach the base station is reduced. These effects can be seen in Figure 5.9. In all of the cases, BAS and BAS-H outperform the MTE algorithm for 90 per cent coverage.

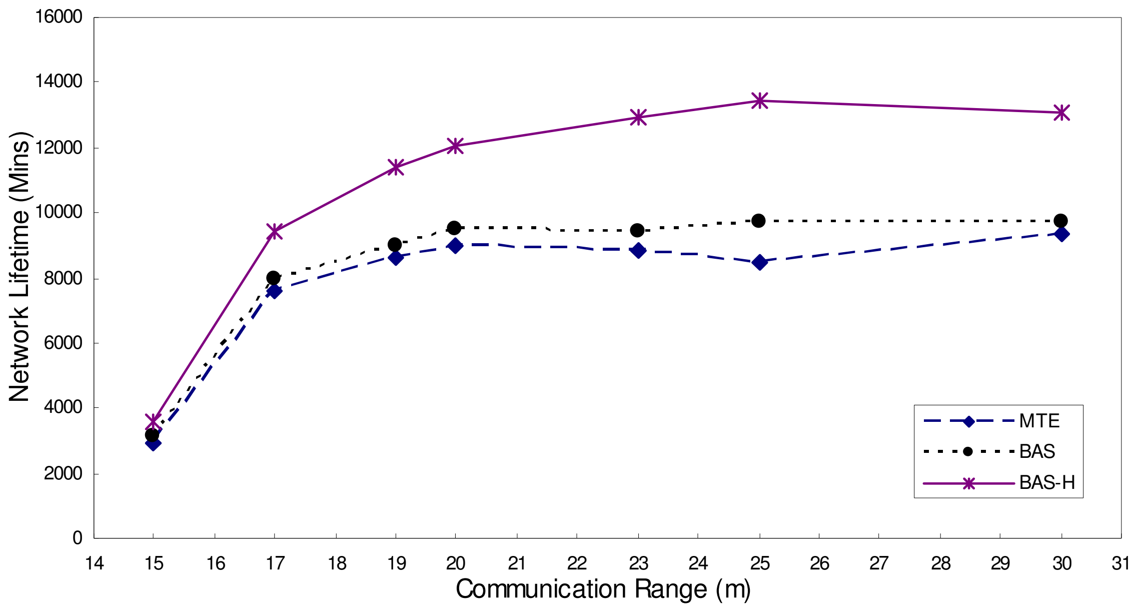

FIGURE 5.9. Network lifetime versus communication range

5.2.7. Network lifetime with different Transmission & Receiving Energy Coefficients ($E_{t.elec}$ and $E_{t.elec}$)

In the following simulations, the effects of different transmission & receiving energy coefficients to the network lifetime are measured. Simulations were made with the changed parameter values in Table 5.6.

TABLE 5.6: Transmission and receiving energy totals in the simulation

PARAMETER	VALUES IN THE SIMULATION
$E_{t.elec} + E_{r.elec}$	10, 100, 150, 200, 400 nJ/bit

As expected, when transmission and receiving energy totals are increasing, the network lifetime decreases. However, even when they are theoretically close to zero, the network lifetime has upper boundaries because of both the amplifier energy and the waiting energy used. When they are small, dynamic algorithms perform better than MTE because smaller fixed costs in energy means more alternative paths to use with small energy expenditure. Nevertheless, for MTE, it uses the same smallest cost path until one of the relaying sensors dies. It can be seen in Figure 5.10 that the BAS algorithm loses its advantage when the transmission and receiving energy expenditures are big. The BAS-H algorithm still keeps its advantage in this case as well; we can say that the traffic load factor in BAS-H that diverts the traffic to the less busy nodes produces this effect.

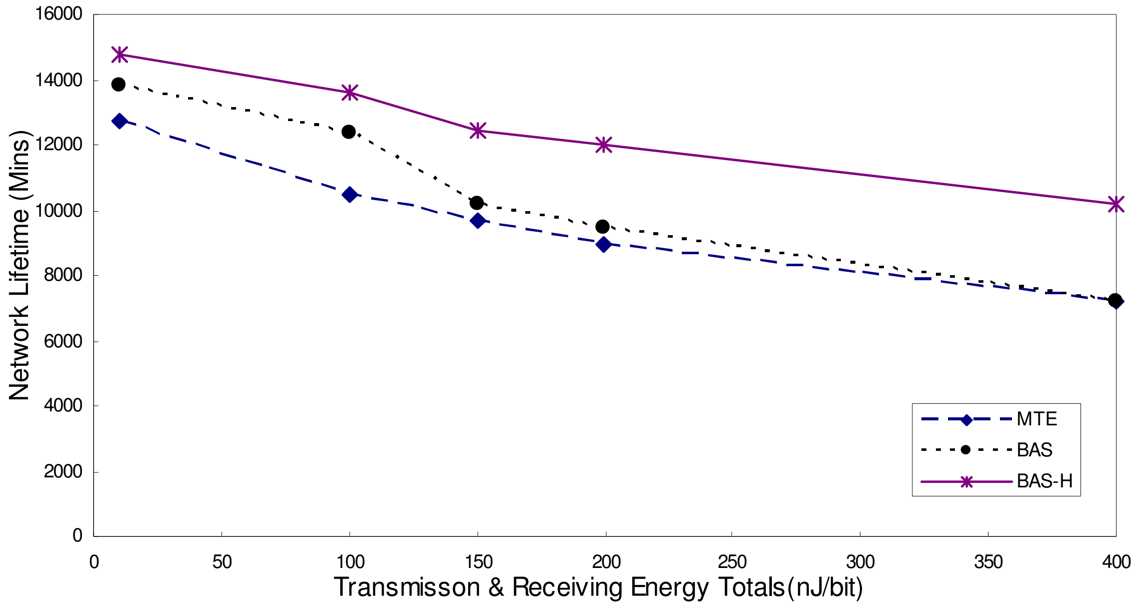

FIGURE 5.10. Network lifetime versus transmission & receiving energy coefficient totals

5.2.8. Network lifetime with different Amplifier Energy Coefficients (E_{amp})

In the following simulations, the effects of different amplifier energy coefficients to the network lifetime are measured. Simulations were made with the changed parameter values in Table 5.7.

TABLE 5.7: Simulation parameters for changing the amplifier energy coefficient

PARAMETER	VALUES IN THE SIMULATION
E_{amp}:	10, 50, 100, 500, 1000 pJ/bit m^3

As expected, when the amplifier energy coefficient is increased, the network lifetime decreases. When this coefficient is close to zero, the network lifetime is similar to the case where transmission and receiving energy is very small; this shows that the amplifier energy coefficient has a similar effect on the network lifetime. This has to do with the similar energy expenditures from these coefficients. Energy spent from the transmission & receiving energy coefficients is 200nJ/bit and energy spent from the amplifier energy is 100nJ/bit for 10 meters communication and 800nJ/bit for 20 meters communication. Their energy spending comparison is from a multiple of half to fourth of each other and their non-existence brings similar results.

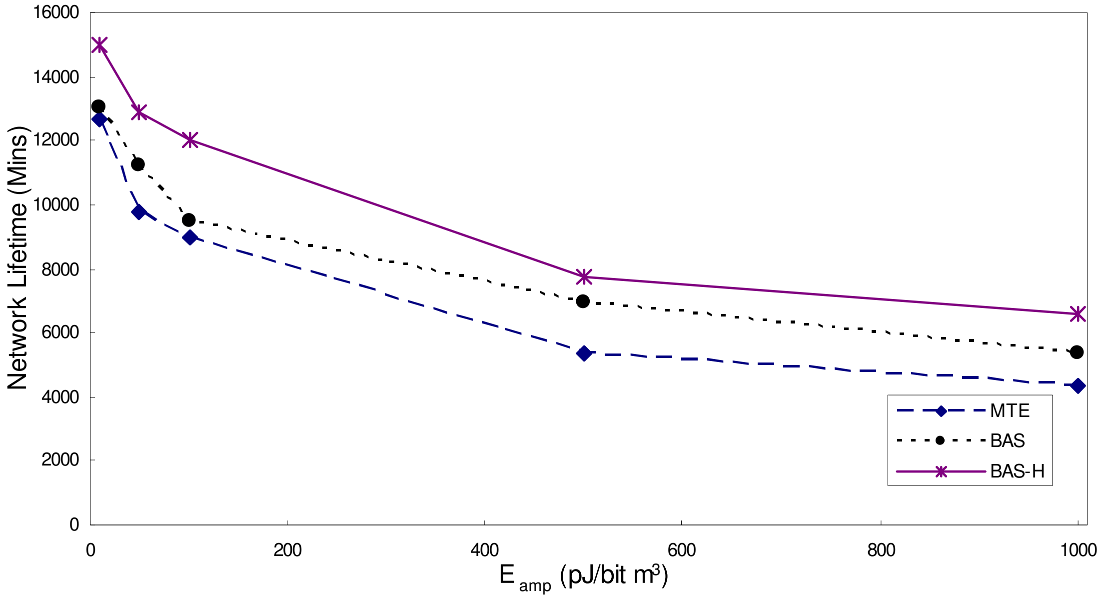

FIGURE 5.11. Network lifetime versus amplifier energy coefficient

When the amplifier energy constant is increased to 10 times the original value of 100pJ/bit, the network lifetimes fall to just half of the original values. This is because the amplifier energy is just one of the factors affecting the network lifetime. There is also the waiting energy spent, and the transmission and receiving energies; these factors spend significant amounts of energy as well. Just increasing the amplifier energy factor did not reduce the relative contributions of the other factors, and did not shorten the network lifetime as much of increasing the amplifier energy factor.

5.2.9. Network lifetime with different waiting energy consumption rates

The previous simulations were made with 50nJ/min fixed waiting energy consumption rate for each sensor. The following simulations were made to observe the effects of different waiting energy consumption rates. The network lifetime decreases with bigger waiting energies for all algorithms. When it is small and near to zero, a very substantial improvement in the network lifetime is observed. This difference shows that the waiting energy has a considerable effect on the network lifetime. The other energy expenditure factors like the amplifier energy and the transmission energy coefficient were effective only on the sensors transmitting or relaying data. However, the waiting energy is effective on every sensor, regardless if it sends data by sensing an event or relaying it from the other sensors.

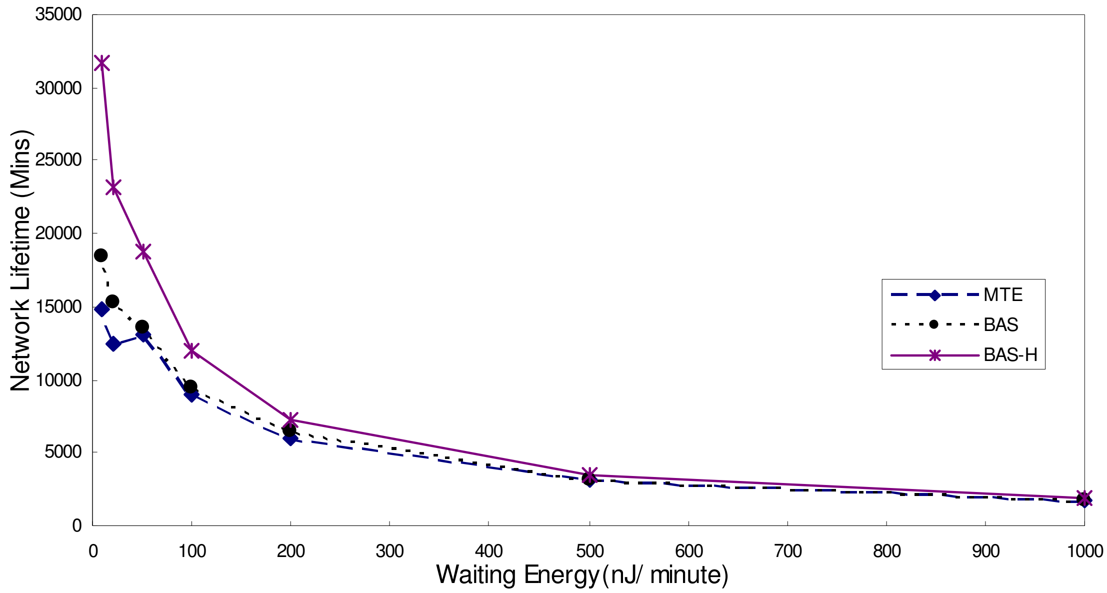

FIGURE 5.12. Network lifetime versus waiting energy consumption

Changes in this energy-spending parameter have a more direct effect on the network lifetime since it is effective on every sensor. When it is half of the original value, the network lifetime can be nearly double of the original lifetime. When the waiting energy is increased to double of initial value, the network lifetime falls to nearly half of the original lifetime. For small values of this variable, the performance differences between the routing algorithms become more obvious, because the waiting energy is the fixed cost in the network, and the routing algorithms can show their real performances when the fixed costs are smaller. It can also be seen that when the waiting energy coefficient is 10 times its original value, the lifetimes of the networks are very similar, and their differences are lost. This is because this cost item more directly affects every sensor and it has a greater impact on the network lifetime. When the waiting energy is very big, most of the battery energy is spent in the waiting task, very little part of it is spent in routing. This result shows the importance of sleeping the sensors when they are not needed. For a routing algorithm as well, sleeping schedules have a big effect on the network lifetime.

6. CONCLUSIONS

In this thesis, our research problem is one of the most important problems of wireless sensor networks: energy saving. Today, sensor networks are well-known; properties and problems, and general solution methods are known by the researchers. Due to this reason, we did not go into too much detail explaining the topic. We have explained the sensor network structure and sensor nodes' architecture, in general terms. Then, we explained why energy saving has become the biggest research issue in the field. We listed the main project teams and their works on the topic.

We have inspected the ways to improve network lifetime in wireless sensor networks. Within the areas to save energy, we have focused on the network level packet routing techniques, and energy saving on multi-hop communication has been investigated. Some of the existing routing protocols/algorithms like SPIN, Directed Diffusion, Rumor Routing, Gradient Based Routing, EAR, LEACH, PEGASIS, TEEN, GEAR were revised. Their general characteristics and their performance gains against their competitors were given. Then we described the network energy consumption models used. In addition, we explained that the definition of the network lifetime can be made in many different ways and the results associated with the same solution can differ according to these definitions. In our work, we decided to use the network lifetime definition as the sensing coverage of the network at a given time compared to the original area. Based on the previous research, we tried to introduce different ideas in the network level routing algorithms. It has been shown that routing decisions can change the network lifetime drastically since one of the main energy consumers is the radio communication. Another method for increasing the network lifetime is implementing a sleep schedule for some of the sensors to allow them to hibernate for a specific time.

Instead of static routing techniques, dynamic routing algorithms that change routing paths according to the battery levels of sensors were offered as a solution to the problem of avoiding the excessive energy used in communication. In addition, the sensors monitoring the same area were found to be a property that could be utilized to improve network lifetime. We have implemented our proposed algorithms and made simulations with them. We were unable to achieve performance improvements in all of our algorithms, but Remaining Battery over Area Sensed (BAS) and Remaining Battery over Area Sensed with History (BAS-H) algorithms performed better than the other proposed algorithms.

The most important improvement that can be made on our algorithms is to implement a sleeping/hibernation schedule on them. In particular, the normalized battery metric in the BAS and BAS-H can be used as a parameter to decide on the frequency of the sleeping period. Some work in the literature aims at splitting sensors into sub-sets that can independently sense the same target area. If not, monitoring some part of the deployment area for some time is acceptable, and if the main criterion is monitoring most of the area, then implementing a schedule related to a sensor's proximity to its neighbors may be feasible. Such a parameter may include the sensor's and its neighbors' battery levels, and the sensors may be hibernated or fully operated probabilistically or by interaction with the neighboring nodes. The probability of hibernation would be related to this parameter. Also, as another improvement, all of the low-level operations in MAC layer communication with neighboring sensors could be designed to work effectively and in harmony with the routing protocol.

It is shown that the average network lifetime for different network setups can be improved. We believe that these kinds of algorithms have application areas over the other hierarchical protocols like LEACH that require direct communication between the cluster-heads and the base stations. If the cluster heads are far from the base station and multi-hop communication is needed, data can arrive at to the cluster head, and then forwarded to the base station with this kind of multi-hop routing schema. We think that there is still some research area within the topic of network layer routing algorithms in Wireless Sensor Networks, and for the implementation of better algorithms to improve lifetimes.

REFERENCES

1 Haenggi, M., “Opportunities and Challenges in Wireless Sensor Networks” in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, Ilyas, M. and I. Mahgoub (editors), pp. 22-24, CRC Press, Boca Raton, 2005.

2 Heinzelman, W. B. “Application-Specific Protocol Architectures for Wireless Networks” *Doctor of Philosophy Thesis, MIT*, June 2000.

3 Al-Karaki, J.N. and A.E. Kamal, “Routing techniques in wireless sensor networks: a survey”, *Wireless Communications, IEEE* , Volume 11, Issue 6, pp. 6- 28, December 2004.

4 Min, R. , M. Bhardwaj, S.H. Cho, A. Sinha, E. Shih, and A. Wang “An Architecture for a Power-Aware Distributed Microsensor Node”, *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS’00) 2000*, pp. 581-590, Lafayette, October 2000.

5 Ye, W. , J. Heidemann, and D. Estrin, “Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks” *IEEE/ACM Transactions on Networking*, Volume 12, Issue 3, pp. 493-506, June 2004.

6 Shih, E., S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, “Physical Layer Driven Protocol and Algorithm Design for Energy Efficient Wireless Sensor Networks”, *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking- MOBICOM 2001*, Rome, pp. 272-287, July 2001.

7 Sohrabi, K., J. Gao, V. Ailawadhi, and G. J. Pottie, “Protocols for Self-Organization of a Wireless Sensor Network”, *IEEE Personal Communications*, Volume 7, Issue 5, pp.16-27, October 2000

8 Woo, A., D. E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks", *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking- MOBICOM 2001*, Rome, pp. 221-235, July 2001.

9 Arisha, K., M. Youssef, and 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 (IEEE IMPACCT) 2002, New York, May 2002.

10 Ye, W., J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM) 2002*, New York, Volume 3, pp. 1567-1576, June 2002.

11 IEEE Standard 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf, October 2003.

12 Bougard, B., F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, "Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor Networks: Modeling and Improvement Perspectives," *Design, Automation, and Test in Europe (DATE)*, Munich, Volume 1, pp. 196-201, March 2005.

13 "IEEE 802.15 Working Group for WPANs", http://www.ieee802.org/15/

14 Callaway, E. H., "*Wireless Sensor Networks—Architectures and Protocols*" Auerbach Publications, Florida, 2004.

15 "Wireless Integrated Network Sensors" http://www.janet.ucla.edu/WINS/

16 "MIT µAMPS Project" http://www-mtl.mit.edu/researchgroups/icsystems/uamps/

17 “Wireless Embedded Systems Group” http://webs.cs.berkeley.edu/

18 Akyildiz, I.F., W. Su, Y. Sankarasubramaniam, E. Cayirci, "A survey on sensor networks", *IEEE Communications Magazine*, Volume 40, Issue 8, pp. 102- 114, August 2002.

19 Akyildiz, I.F., W. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless sensor networks: a survey”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Volume 38 , Issue 4, pp.393-422, March 2002.

20 Demirkol, I., C Ersoy, F Alagöz “MAC Protocols for Wireless Sensor Networks: a Survey”, *Accepted to IEEE Communications Magazine,* 2005

21 Akkaya, K. and M. Younis, "A Survey on Routing Protocols in Wireless Sensor Networks", *Ad Hoc Networks Journal*, Elsevier, Volume 3, pp. 325-349, 2005.

22 Rentala, P., R. Musunuri, S. Gandham, U. Saxena, "Survey on sensor networks", *Technical Report, UTDCS-33-02,* University of Texas at Dallas*, 2002*

23 Xu, N., “A Survey of Sensor Network Applications” *IEEE Communications Magazine*, Volume 40, Issue 8, pp. 102–114, August 2002.

24 Royer, E. M., C.K. Toh, “A review of current routing protocols for ad hoc mobile” *IEEE Personal Communications*, 1999, Volume 6, Issue 2, pp. 46-55, April 1999.

25 Sahni, S. and Xiaochun Xu “Algorithms for Wireless Sensor Networks”, *International Journal of Distributed Sensor Networks,* Taylor & Francis, Volume 1, Number 1, pp 35-56, January 2005.

26 Heinzelman, W., J. Kulik, and H. Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks”, *Proceedings of the 5th annual ACM/IEEE*

international conference on Mobile computing and networking (Mobicom) 1999, Seattle, pp. 174–185, August 1999.

27 Intanagonwiwat, C., R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* (*ACM MobiCom'00)*, Boston , pp. 56–67, August 2000.

28 Akkaya, K. and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks" *Ad Hoc Networks*, Volume 3, Issue 3, pp. 325-349, May 2005.

29 Braginsky, D. and D. Estrin, "Rumor routing algorithm for sensor networks", *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications,* Atlanta, pp. 22-31, October 2002.

30 Schurgers, C. and M.B. Srivastava, "Energy efficient routing in wireless sensor networks", *Proceedings of the IEEE Military Communications Conference (MILCOM), 2001*, Washington, Volume 1 , pp. 357–361, October 2001.

31 Shah, R.C. and J. M. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks*", Proceedings of the IEEE Wireless Communications and Networking Conference 2002*, Orlando, Volume 1, pp. 351-355, March 2002.

32 Ye, F., A. Chen, S. Lu and L. Zhang "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks," *Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN) 2001*, Phoenix, pp. 304–309, October 2001.

33 Rodoplu, V. and T. H. Meng, "Minimum Energy Mobile Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Volume 17, no. 8, pp. 1333–1344, August 1999.

34 L. Li, and J. Y. Halpern, "Minimum-Energy Mobile Wireless Networks Revisited", Proceedings of the IEEE International Conference on Communications (ICC) 2001, Helsinki, Volume 1, pp. 278–283, June 2001.

35 Servetto, S. and G. Barrenechea, "Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks", *Proceedings of the* 1st International Workshop on Wireless Sensor Networks and Applications 2002, Atlanta, pp. 12–21, September 2002.

36 Willig, A., R. C. Shah, J. M. Rabaey, A. Wolisz, "Altruists in the PicoRadio Sensor Network", *Proceedings of the International Workshop on Factory Communication Systems, 2002,* Vasteras, Sweden, pp.175-184, August 2002.

37 Chang, J.H. and L. Tassiulas, "Maximum Lifetime Routing in Wireless Sensor Networks*" IEEE/ACM Transactions on Networking*, Volume 12, No. 4, August 2004.

38 Tan, H. Ö. and İ. Körpeoğlu, "Power efficient data gathering and aggregation in wireless sensor networks", *ACM SIGMOD Record*, Volume 32, Issue 4, pp. 66-71, December 2003.

39 Younis. M, M. Youssef and K. Arisha "Energy-aware management for cluster-based sensor networks", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Volume 43 , Issue 5, pp. 649-668, December 2003.

40 Chiasserini, C.F. and R.R. Rao, "Routing Protocols to Maximize Battery Efficiency", *Proceedings of 21st Century Military Communications Conference (IEEE MILCOM 2000)*, Los Angeles, Volume 1, pp. 496-500, October 2000.

41 Cai, WA., XA Jin, YA Zhang and KA Chen "A load-balanced minimum energy routing algorithm for Wireless Ad Hoc Sensor Networks", *Journal of Zhejiang University-Science A, 2006*, Springer, Volume 7, Number 4, pp. 502- 506, April 2006.

42 Heinzelman, W. R., A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", *Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000*, Maui, Hawaii, Volume 2, pp. 10, January 2000.

43 Oyman, E. I., and C. Ersoy, "Overhead Energy Considerations for Efficient Routing in Wireless Sensor Networks" *Computer Networks: The International Journal of Computer and Telecommunications Networking,* Elsevier, Volume 46, Issue 4, pp. 465-478, November 2004.

44 McKeown, N., "CS244a: An Introduction to Computer Networks, Handout 5: Internetworking and Routing", *CS244a Course Lecture Slides*, Winter 2003, Stanford University, pp. 8-9, http://www.stanford.edu/class/cs244a/ .

45 Heinzelman, W. B., A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communication*, Volume 1, Issue 4, pp. 660–70, October 2002.

46 Heinzelman, W.R., A. Sinha, A. Wang, A.P. Chandrakasan "Energy-scalable algorithms and protocols for wireless microsensor networks", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000 (ICASSP'00)*, Istanbul, Volume 6, pp. 3722-3725, June 2000.

47 Lindsey, S. and C. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems," *Proceedings of the IEEE Aerospace Conference 2002,* Big Sky, Volume 3, 9–16, pp. 1125–1130, March 2002.

48 Lindsey, S., C. Raghavendra, and K. Sivalingam, "Data Gathering in Sensor Networks using the Energy*Delay Metric", Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS 2001) on Issues in Wireless Networks and Mobile Computing 2001, San Francisco, pp.2001-2008, April 2001.

49 Manjeshwar, A. and D. P. Agarwal, “TEEN: a Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks”, Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS 2001) on Issues in Wireless Networks and Mobile Computing 2001, San Francisco, pp.2009-2015, April 2001.

50 Manjeshwar, A. and D. P. Agarwal, “APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks”, *Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS 2002)*, Fort Lauderdale, Florida, pp. 195–202, April 2002.

51 Yu, Y., R. Govindan and D. Estrin , “Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks”, *UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023,* May 2001.

52 Rabaey, J.M., M.J. Ammer, J.L. Silva, D. Patel, S. Roundy, “Pico Radio Supports Ad Hoc Ultra-Low Power Wireless Networking”, *IEEE Computer Magazine*, pp. 42-48, July 2000.

53 Muruganathan, S.D., D.C.F. Ma, R.I. Bhasin, A.O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks", *IEEE Communications Magazine*, Volume 43, Issue 3, pp 8-13, March 2005.

54 Onur, E., C. Ersoy, and H. Delic, “Finding sensing coverage and breach paths in surveillance wireless sensor networks”, *Proceedings of the 15th IEEE International Symposium Personal, Indoor and Mobile Radio Communications*, *2004*, Barcelona, Volume: 2, pp. 984-988, September 2004.

55 Onur, E., C. Ersoy, and H. Delic, “On the quality of deployment in wireless sensor networks”, *Proceedings of the 8th International Conference on Telecommunications*, *ConTEL 2005*, Zagreb, Volume 2, pp. 547- 551, June 2005.

56 Cardei, M. and J. Wu, “Coverage in wireless sensor networks,” in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, Ilyas, M. and I. Mahgoub (editors), pp. 361-369, CRC Press, Boca Raton, 2005.

57 Demirkol, I., F. Alagoz, H. Delic, and C. Ersoy, “Wireless sensor networks for intrusion detection: packet traffic modeling”, *IEEE Communications Letters*, Volume 10, Issue 1, pp. 22-24, January 2006.

58 Stojmenovic, I. and X. Lin “Power Aware Localized Routing in Wireless Networks”, *IEEE Transactions on Parallel and Distributed Systems,* Volume 12, No. 11, pp. 1122-1133, November 2001.

59 Melodia, T., D. Pompoli and I. F. Akyildiz “Optimal local topology knowledge for energy efficient geographical routing in sensor networks”, *Proceedings of the Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, *(INFOCOM 2004),* Hong Kong, Volume 3, pp. 1705-1716, March 2004.

60 Hac, A., *Wireless Sensor Network Designs*, p 108. , John Wiley & Sons Lmt., West Sussex, 2003.

61 Lu, J. and T. Suda, “Coverage-aware self-scheduling in sensor networks”, *Proceedings of IEEE 18th Annual Workshop on Computer Communications*, *(CCW 2003)*, Dana Point, California, pp. 117-123, October 2003.

62 S. Slijepcevic and M. Potkonjak, “Power efficient optimization of wireless sensor networks”, *IEEE International Conference on Communications, 2001*, Helsinki, Volume 2, pp. 472-476, June 2001.

63 Cardei, M. and D. Du, “Improving wireless sensor network lifetime through power aware organization”, *Wireless Networks* , Springer, Volume 11, Number 3, pp. 333-340, May 2005.

---

64 Oyman, E.I., C. Ersoy, “Multiple sink network design problem in large scale wireless sensor networks”, *Proceedings of IEEE International Conference on Communications, 2004*, Paris, Volume 6, pp. 3663- 3667, June 2004.