DUTY CYCLE CONTROL IN WIRELESS SENSOR NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MİNE YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2007

DUTY CYCLE CONTROL IN WIRELESS SENSOR NETWORKS

submitted by MİNE YILMAZ in partial fulfillment of the requirement for the degree of Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University by,

Prof. Dr. Canan Özgen ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ________________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Semih Bilgen ________________
Supervisor, **Electrical and Electronics Engineering**

Examining Committee Members:

Prof. Dr. Uğur Halıcı ________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Semih Bilgen ________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Özgür Barış Akan ________________
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. Cüneyt Bazlamaccı ________________
Electrical and Electronics Engineering Dept., METU

Dr. Altan Koçyiğit ________________
Informatics Institute, METU

Date: (METU, II) **07.09.2007**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Mine YILMAZ

Signature :

ABSTRACT

DUTY CYCLE CONTROL IN WIRELESS SENSOR NETWORKS

YILMAZ, Mine

M. Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Semih BİLGEN

September 2007, 63 pages

Recent advances in wireless communication and micro-electro-mechanical systems (MEMS) have led to the development of implementation of low-cost, low power, multifunctional sensor nodes. These sensor node are small in size and communicate untethered in short distances. The nodes in sensor networks have limited battery power and it is not feasible or possible to recharge or replace the batteries, therefore power consumption should be minimized so that overall network lifetime will be increased. In order to minimize power consumed during idle listening, some nodes, which can be considered redundant, can be put to sleep. In this thesis study, basic routing algorithms and duty cycle control algorithms for WSNs in the literature are studied. One of the duty cycle control algorithms, Role Alternating, Coverage Preserving, and Coordinated Sleep algorithm (RACP) is examined and simulated using the ns2 simulation environment. A novel duty cycle control algorithm, Sink Initiated Path Formation (SIPF) is proposed and compared to RACP in terms of sleep sensor ratio and time averaged coverage.

Keywords: Wireless Sensor Networks, duty cycle control, energy efficiency.

ÖZ

KABLOSUZ SENSOR AĞLARI İÇİN DOLULUK BOŞLUK ORANI KONTROLÜ

YILMAZ, Mine

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Semih BİLGEN

Eylül 2007, 63 Sayfa

Kablosuz iletişim ve mikroelektromekanik sistemlerdeki gelişmeler, düşük maliyetli, düşük enerjili, çok fonksiyonlu algılayıcı düğümlerinin geliştirilmesine ve gerçekleştirilmesine yol açmıştır. Bu algılayıcı düğümleri küçük boyutlu olup kısa mesafelerde sorunsuz iletişimi mümkün kılmıştır. Algılayıcı ağlarındaki bu düğümler sınırlı pil gücüne sahiptir ve pillerin değiştirilmesi ya da yeniden doldurulması mümkün değildir. Boş dinleme zamanında harcanan enerjinin azaltılması için, gereksiz kabul edilebilen bazı düğümler uyutulabilir. Bu tez çalışmasında, temel rota tespit yöntemleri ve doluluk boşluk algoritmaları incelenmiştir. Doluluk boşluk algoritmalarından biri olan RACP algoritması incelenmiş ve ns2 kullanılarak simule edilmiştir. Yeni bir doluluk boşluk algoritması, SIPF, önerilmiş ve simülasyon yoluyla RACP ile uyuyan algılayıcı oranı ve kapsama alanının zaman üzerinden ortalaması bakımlarından karşılaştırılmıştır.

Anahtar Kelimeler: Kablosuz Algılayıcı Ağları, doluluk boşluk oranı, enerji verimliliği

To My Parents and To My Brother

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. Semih Bilgen for his realistic, encouraging, and constructive approach throughout my masters study and his efforts during supervision of my thesis.

I would like to express my appreciation to ASELSAN Inc. and my colleagues for understanding and support during my academic studies.

Finally, I would like to express my thanks to my parents, Melike and Erdem Yılmaz, my brother Erdi Yılmaz, for their love, trust, understanding, and every kind of support not only throughout my thesis but also throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	Acknowledgement Packet
ADV	Advertisement Packet
APTEEN	AdaPtive Threshold sensitive Energy Efficient Sensor Network Protocol
ACQUIRE	ACtive QUery forwarding In sensor nEtworks
CADR	Constrained Anisotropic Diffusion Routing
CBQ	Class Based Queuing
CBR	Constraint Based Routing
COR	Coordination Packet
CSMA/CA	Carrier Sense Medium Access/Collision Avoidance
EA-ALPL	Energy Aware Adaptive Low Power Listening
FTP	File Transfer Protocol
GAF	Geographic Adaptive Fidelity
GEAR	Geographical and Energy-Aware Routing
IDSQ	Information Driven Sensor Querying
LAN	Local Area Network
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
MECN	Minimum Energy Communication Network
ns-2	Network Simulator ver. 2
OTcl	Object Tool Command Language
PEAS	Probing Environment and Adaptive Sleeping
PAM	Position Advertisement Message
PEGASIS	Power Efficient Gathering in Sensor Information Systems
QoS	Quality of Service

RACP	Role-Alternating, Coverage-Preserving, and Coordinated Sleep Algorithm
REQ	Request Packet
SAR	Sequential Assignment Routing
SIPF	Sink Initiated Path Formation
SMECN	Small Minimum Energy Communication Network
SPIN	Sensor Protocol for Information via Negotiation
Tcl	Tool Command Language
TCP	Transmission Control Protocol
TEEN	Threshold sensitive Energy Efficient Sensor Network Protocol
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

Low-cost, low power, multifunctional sensor nodes that are small in size and communicate untethered in short distances have been developed due to the recent advances in micro-electro-mechanical systems (MEMS) and wireless communication [1]. These tiny sensors have the ability of sensing, data processing, and communicating with each other. Wireless Sensor Networks (WSN) which rely on collaborative work of large number of sensors are realized.

Sensor nodes can be used within many deployment scenarios such as continuous sensing, event detection, event identification, location sensing, and local control of actuators for a wide range of applications such as military, environment, health, space exploration, and disaster relief [32]. Although a large volume of research has been performed and some algorithms are proposed, there is ongoing research on this subject in recent years.

One of the challenging subjects and design constraints in WSNs is efficient energy consumption. Since a sensor node is a microelectronic device, it can only be equipped with a limited power source (<0.5 Ah, 1.2 V). In most application scenarios, replenishment of power resources might be impossible or infeasible [32]. Moreover, each node plays the dual role of data originator and data router, in multi-hop sensor networks, therefore disfunction of nodes can cause serious problems in the sensor network [32]. Furthermore, most of the application based on long time monitoring directly affects the network efficacy and usefulness.

Main sources of power dissipation are used during data processing, data transmission, data reception and idle listening. The power consumed during transmission is the greatest portion of energy consumption of any node [7]. Considering the limited capabilities and vulnerable nature of an individual sensor, a wireless sensor network has a large number of sensors deployed in high density (high up to 20nodes/m3 [30]). Since the nodes are deployed densely and in an *ad hoc* fashion, many nodes stay inactive for long periods and idle listening power dissipation becomes significant. Therefore these nodes can be considered as redundant and can be put to sleep. The main idea will be scheduling sensors to work alternatively and the system lifetime will be prolonged correspondingly.

In this work, we present a novel node scheduling scheme, Sink Initiated Path Formation (SIPF), which is used to configure node work status and schedule the sensor on-duty time in large sensor networks. The algorithm that we proposed is self-configured, fully distributed, and localized. It operates in the network layer as the sensors that are sleeping and awake are essentially differentiated in their routing functionalitites. Since the working environments for WSNs are hostile and remote working environments, it would not be convenient or possible to configure network manually after deployment. For this reason self-configuration is necessary. In order to erase the need for a global synchronization overhead, the proposed algorithm has to be distributed and localized. This favors also scalability of the network.

In the proposed approach, a decision algorithm is proposed in order to decide whether each node in the network sleeps or not periodically. The decision is based on the local neighbor information. Each node gathers local neighbor information from each of its neighbors and the information is updated periodically.

Within the scope of this study, first a literature survey has been performed to review the current work on WSN duty cycle control algorithms, then simulation based performance evaluation of such algorithms has been exercised, with special focus on a selected study, repeating the results reported in the literature, and finally a novel duty cycle control algorithm has been proposed and its performance has been evaluated, comparing the achievement with that selected study.

The rest of this work is organized as follows: Chapter Two presents a review of the relevant literature, Chapter Three is devoted to the prepared simulation infrastructure and repetition of the results reported in the literature, Chapter Four presents the proposed duty cycle control algorithm and compares its performance against the algorithm selected from published literature, and Chapter Five concludes the study.

CHAPTER 2

LITERATURE SURVEY

2.1 ROUTING IN SENSOR NETWORKS

Wireless Sensor Networks (WSN) have a wide range of applications such as environmental monitoring, biomedical research, human imaging and tracking, and military applications [1]. Sensor network design is influenced by many factors, which include *fault tolerance*; *scalability*; *production costs*; *operating environment*; *sensor network topology*; *hardware constraints*; *transmission media*; and *power consumption* [1].

2.1.1 DESIGN ISSUES

The routing protocols designed for WSN should consider the goal, application area, and architecture of the network. The design of routing protocols is influenced by many challenging factors caused by the nature of the WSNs. These factors must be overcome before efficient communication can be achieved in WSNs. Some of these factors will be reviewed in this part [7].

Node Deployment: Node deployment can be random, deterministic or self-organizing. For deterministic deployed networks the routes are pre-determined, however for random deployed networks and self-organizing networks route designation have been a challenging subject.

Energy consideration: Since the life-time of the WSN depends on energy resources and their consumption by sensors, the energy consideration has a great influence on route design. The power consumed during transmission is the greatest portion of energy consumption of any node. Direct communication consumes more power than multi-hop communication; however the multi-hop communication introduces extra topology management and medium access control.

Data Delivery Models: Data delivery model depends on the application and can be continuous, event-driven, query-driven, or hybrid. In continuous model of delivery, each sensor sends the data periodically. In event-driven and query driven models of the data delivery, the transmission is triggered by an event or a query generated by the sink. Hybrid model is a combination of continuous, event-driven and query-driven data delivery models.

Node Capabilities: In the earlier works the nodes were usually assumed to have equal capacity of computation, power, and communication. However, it is possible for nodes to have different functionalities, such as relaying, sensing, and aggregation. Some nodes having these three functions together may consume more energy. In some network topologies there exists cluster heads which have more power and consumes more energy than the other nodes of the network. Heterogeneity is common for much of the networks and introduces complexity to route determination.

Data Aggregation: Since the sensors are densely deployed by definition, the data gathered from each node are correlated. Therefore data aggregation or in other words data fusion decreases the size of the data transmitted.

Fault Tolerance: WSNs are prone to failures; some of the nodes may fail or be blocked by physical interference, physical damage, or lack of power. The routing protocol has to be dynamic; failures of specific nodes should not affect network operation.

Scalability: WSNs may consist of hundreds, thousands or more nodes. Any protocol including routing protocols should manage this huge number of nodes.

Network Dynamics: Most of the proposed networks are considered to be stationary; however for some application areas WSNs in which some or all nodes are mobile are required. Routing protocols for such networks must cater for mobility requirements.

Quality of Service: Some applications require QoS; especially there exist some time-critical applications. The relevance of the data expires within some period. For such applications the routing protocols should be designed according to the requirements. However, the general trend is to attribute more importance to energy awareness than QoS requirements.

2.1.2 ROUTING PROTOCOLS

The routing protocols proposed for WSNs are classified considering several architectural factors [7]. Taxonomy of routing protocols is helpful while designing the network protocol.

2.1.2.1 DATA CENTRIC PROTOCOLS

It is not appropriate to use global identifiers for this huge number of randomly deployed nodes, in most of the WSN applications. However this introduces complexity to query data from a specific set of nodes. Therefore the data is collected from the deployed region. Since the collected data is correlated and mostly redundant; collected data is aggregated in some nodes resulting decrease in the amount of transmitted data so transmission power. The following routing algorithms' main consideration is data and its properties.

- *Flooding:* This is a classical and old routing mechanism [8]. The data gathered is broadcasted unless the specified maximum number of hops per packet is reached, or the packet delivered to the destination. This protocol brings implosion, overlap, and resource blindness problems.

- *Gossiping:* This is also a classical and old method, resembling flooding [8]. The gathered data is not broadcasted but sent to randomly chosen neighbor node until the specified maximum number of hops per packet is reached or the packet delivered to the destination. The delivery of the data takes so much time.

- *SPIN, Sensor Protocol for Information via Negotiation:* The basic idea is using a meta-data or high level descriptors [9]. There are three types of messages, ADV, REQ, and DATA. As shown in Fig.1, the source node broadcast an ADV message to its neighbors, ADV message indeed is meta-data. The interested nodes send REQ, and then the source node sends the DATA to interested nodes. Using same procedure the data can reach interested nodes in the whole network from one end to the other end. Data aggregation is employed.

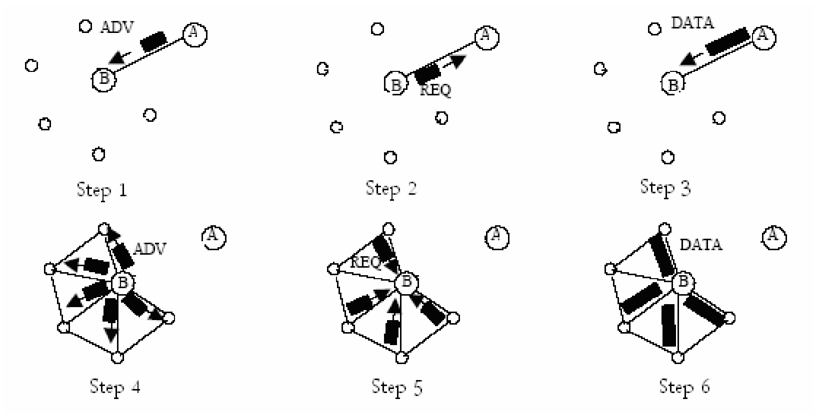

Figure 2.1 – The SPIN Protocol [9]

- *Directed Diffusion* [10]: The sink broadcasts the "interest" message, namely the task descriptor to all nodes, as shown in Fig.2.2. The interest is stored to cache of every node, until timestamp of time specified messages expires. The message contains several gradient fields. The gradient to the sink is set up as the interest propagated through network. When the source node gets the interest it sends the data through the gradient path of the interest. The directed diffusion algorithm solves problems of node addressing or maintaining a global network topology, data caching also reduces energy consumption.

- *Energy-aware Routing* [11]*:* In order to increase network life time a set of sub-optimal routes are proposed to use. The paths are chosen according to energy consumption of the path. Using the path that is consuming minimum energy frequently deplete energy source of specific nodes. Since one of the certain paths are chosen with equal probability, the network life time increases.

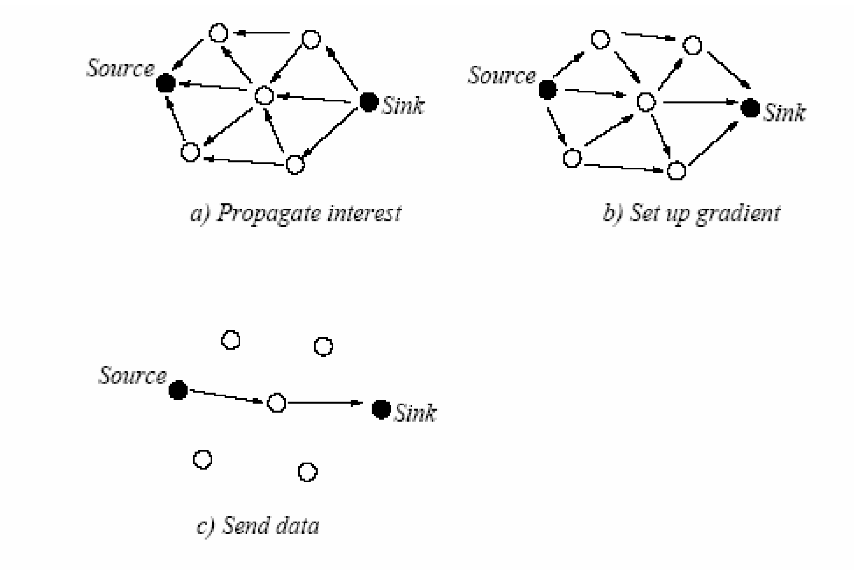

Figure 2.2 – An Example of Directed Diffusion [10]

- *Rumor Routing* [12]: Rumor routing can be considered as a derivation of "directed diffusion". If the number of queries is large, but number of events is small, directed diffusion becomes inefficient. Considering this shortcoming of the directed diffusion, flooding the events not the queries is proposed. Rumor Routing is another solution to this problem, it is in-between flooding query and flooding event. The main idea is to route the queries to specific nodes that have observed specific events. When a node detects an event, it adds it to its event table and generates an agent in order to flood through network and propagate the detected information to the distant nodes. The sink queries then the query transmitted to the related node easily and efficiently.

- *Gradient Based Routing* [13]: This algorithm is to some extent different than "Directed Diffusion". The interest packet is diffused through network in order to query the data. During the query process, the distance to the sink in terms of number of hops is recorded in the interest packet. Each node can discover the minimum distance from itself to the sink. The gradient is calculated as the difference between the node's gradient and its neighbor's gradient. The node decides to forward the packet to the link with the largest gradient. Three different spreading techniques have been proposed: Stochastic Scheme (when two or more next nodes have the same gradient the node selects one randomly), Energy-Based Scheme (when a node has scarce energy, it increases its gradient), and Stream-Based Scheme (to divert streams away from nodes relaying traffic).

- *CADR, Constrained Anisotropic Diffusion Routing* [14]*:* The objective of the algorithm is to maximize information gain, however this causes a reduction in latency and bandwidth. There are two techniques; CADR and IDSQ (Information-Driven Sensor Querying). In *CADR*, each node calculates information, cost objective. According to this and end-user requirements, the node specifies a route. In *IDSQ*, querying node determine the node that can provide the most useful information also considering energy cost.

- *COUGAR* [15]*:* The network considered as a huge distributed database system. A leader node is selected to aggregate data and transmits the aggregated data to gateway. The gateway is responsible for generating query plan which specifies the necessary information about data flow and in-network computation for incoming query. Then the gateway sends it to the relevant nodes. Query plan also describes how to select a leader.

- *ACQUIRE, Active Query forwarding In sensoR nEtworks* [16]*:* The algorithm is designed for one-shot, complex queries for replicated data. Although flooding based mechanisms like flooding, gossiping, SPIN, and directed diffusion algorithms are well-suited for continuous, aggregate queries because cost of initial flooding can be compensated and can become negligible during continuous data flow from source(s) to the sink, these algorithm are not energy efficient for one-shot, complex queries for replicated data. The algorithm sees the network as a distributed database. Sending sink the query, each node receiving the query responds using its pre-cached information, and forward it to the other nodes. The pre-cached information is continuously updated. The complex queries are resolved to simple sub queries while the query is forwarded through a path in the network. After resolving the query completely, the information send back to the sink reverse or the shortest path is used. The algorithm is efficient if complex queries are common for the network.

2.1.2.2 HIERARCHICAL PROTOCOLS

Hierarchical protocols have been proposed in order to meet the energy efficiency and scalability requirement of the WSNs. The main issue is forming sub network clusters, encouraging multi hop transmission and enabling data fusion.

- *LEACH, Low Energy Adaptive Clustering Hierarchy* [17]*:* The algorithm is based on clustering. Clusters of sensor nodes are formed according to the received signal rate of the nodes. Local cluster heads act as a router to the sink. The number of the cluster heads is limited, approximately 5% of the all nodes. The cluster head selection is performed randomly, in order to balance the energy of the network. Every node picks a number between 0 and 1 randomly if the number is greater than the calculated value for following equation, where p is the desired percentage of the cluster heads, r is the current round, and G is the set of nodes that have not been cluster heads in the last 1/p rounds.

$$T(n) = \begin{cases} \dfrac{P}{1-P*\left[r.\mathrm{mod}(1/P)\right]} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

LEACH is completely distributed; no global knowledge is applied

- *PEGASIS, Power Efficient Gathering in Sensor Information Systems* [18]*:* It is an improved version of LEACH. Instead of forming clusters, it is based on forming chains of sensor nodes. One node is responsible for routing the aggregated data to the sink. Each node aggregates the collected data with its own data, and then passes the aggregated data to the next ring. The difference from LEACH is to employ multi hop transmission and selecting only one node to transmit to the sink or base station. Since the overhead caused by dynamic cluster formation is eliminated, multi hop transmission and data aggregation is employed, PEGASIS outperforms the LEACH. However excessive delay is introduced for distant nodes, especially for large networks and single leader can be a bottleneck. Figure 2.3 illustrates PEGASIS algorithm. Node c0 and c4 forwards the obtained data to c1 and c3, respectively. Node c1 and c3 aggregates the data and forward it to c2. c2 is responsible for sending the gathered data to the base station.

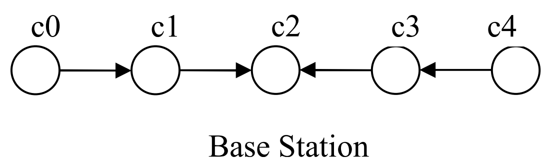

Figure 2.3 – Chaining in PEGASIS [7]

- *Hierarchical-PEGASIS* [19]*:* The protocol is an extension of PEGASIS aiming to decrease the delay. Simultaneous transmissions are employed. In order to avoid collisions of simultaneous transmissions, two different solutions are proposed: The first possible solution can be employing signal coding namely CDMA; the other is allowing simultaneous transmission only for spatially separated nodes. Fig.4 illustrates the algorithm. Nodes c0, c2, c4, and c6 forward their obtained data to c1, c3, c5, and c7, respectively. Doing the data aggregation, c1 and c5 forwards the data to c3 and c7. Then c7 sends the aggregated data to c3. Node c3 is responsible for sending the gathered data to the base station. If PEGASIS is employed for this network it will take 4 unit times to transmit all the data to c3, whereas it takes 3 unit times. The difference will increase with increasing network size.

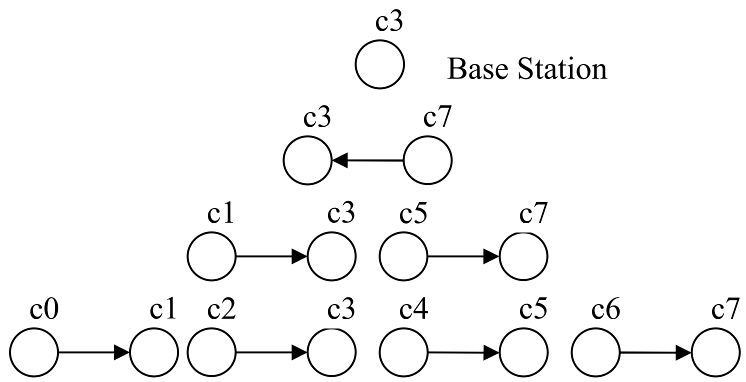

Figure 2.4 – Hierarchical PEGASIS for Chain Based Binary Scheme [7]

- *TEEN, Threshold sensitive Energy Efficient sensor Network Protocol* [20]*:* Closer nodes form clusters, with a cluster heads to transmit the collected data to one upper layer. Forming the clusters, cluster heads broadcast two threshold values. First one is hard threshold; it is minimum possible value of an attribute to trigger a sensor node. Hard threshold allows nodes transmit the event, if the event occurs in the range of interest. Therefore a significant reduction of the transmission delay occurs. Unless a change of minimum soft threshold occurs, the nodes don't send a new data packet. Employing soft threshold prevents from the redundant data transmission. Since the protocol is to be responsive to the sudden changes in the sensed attribute, it is suitable for time-critical applications. In the LEACH protocol, every cluster-head directly communicate with sink, however in TEEN there are three kinds of nodes and a base station (sink) as illustrated in Figure 2.5. Simple nodes gather the data from environment and forward it to the "1st Level Cluster Head"s. Each 1st Level Cluster Head aggregates the data gathered from the simple nodes connected to its cluster, then forward it to "2nd Level Cluster Head"s. 2nd Level Cluster Head can directly forward the data gathered from its cluster to base station. Some of simple nodes may be belong to cluster of 2nd Level Cluster Heads when they are close to these nodes.

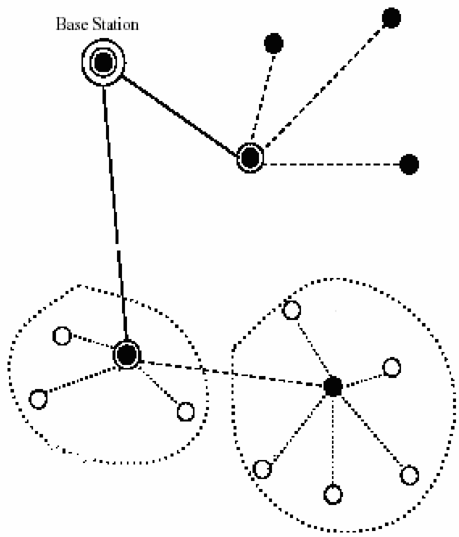

Figure 2.5 – Hierarchical Clustering in TEEN and APTEEN [20]

- *APTEEN, AdaPtive Threshold sensitive Energy Efficient sensor Network Protocol* [21]: The protocol is an extension of TEEN aiming to capture both time-critical events and periodic data collections. The network architecture is same as TEEN. After forming clusters the cluster heads broadcast attributes, the threshold values, and the transmission schedule to all nodes. Cluster heads are also responsible for data aggregation in order to decrease the size data transmitted so energy consumed. According to energy dissipation and network lifetime TEEN gives better performance than LEACH and APTEEN because of the decreased number of transmissions. The main drawbacks of TEEN and APTEEN are overhead and complexity of forming clusters in multiple levels, implementing threshold-based functions and dealing with attribute-based naming of queries.

- *Energy-aware Routing for Cluster-based Sensor Networks* [22]*:* Sensors are grouped into clusters. Cluster heads namely gateways are less energy constrained nodes. Gateways maintain the states of the nodes and sets up multi hop routes. Sink only communicates with the gateway. Gateway informs other nodes about in which slot they should listen others' transmission in which slot they can use for transmission. The sensor can be in four states; sensing only, relaying only, sensing-relaying and inactive. A cost function is defined between any two nodes in terms of energy consumption, delay optimization and other performance metrics. Using this cost function, a least-cost path is found between sensor nodes and the gateway.

- *Self-organizing Protocol* [23]*:* The protocol can be applied to the heterogeneous networks which consist of mobile, stationary nodes. The sensing nodes send the captured data to predetermined set of nodes, namely routers. This stationary router nodes form the backbone of the network and forward the gathered data to a more powerful node, namely sink. Since such a heterogeneous network requires addressing, the address of the node is identified through the router it connected. There are 4 phases to build a routing table. First phase is *Discovery Phase;* each node discovers its neighbors. In the *Organization Phase* groups are formed, each node allocates an address; routing tables are formed for each node. Next phase is *Maintenance Phase.* Routing tables and energy levels are updated in this phase. The last phase is *Self-organization Phase,* in case of node failure or partition, group reorganizations are performed.

2.1.2.3 LOCATION BASED PROTOCOLS

Most of the routing protocols for sensor nodes require location information for sensor nodes. Since addressing like IP-addressing is not employed in WSNs, and the nodes are spatially correlated, routing paths can be maintained easily and efficiently employing location information.

- *MECN, Minimum Energy Communication Network* [24]*:* Each node is expected to know its location (using GPS, deterministic and anchored node deployment etc.). Multihop communication is employed in this algorithm without clustering. The node with a packet to send to the sink, decides whether or not to employ multihop communication by calculating the approximate energy costs from the destinations. The scheme identifies a relay region for each node, to send any node from this region is more efficient in terms of power consumption, and a minimum energy path is formed using local information of each node. Since the protocol is self-reconfiguring, it can dynamically adapt to the node failures or topology changes.

- *SMECN, Small Minimum Energy Communication Network* [25]*:* It is an extension to MECN. The minimum energy network is constructed like MECN, but the relay region is smaller (in terms of number of edges). SMECN uses less energy than MECN.

- *GAF, Geographic Adaptive Fidelity* [26]*:* Although it is designed for mobile networks, it can be applied to stationary networks also. The algorithm based on turning of some nodes without affecting the routing fidelity. The network divided into grids, nodes in the same grid considered equivalent in energy cost. Some of the nodes in the same grid turn power off. Therefore an increase in network lifetime is observed, especially for higher densities. Nodes change their state from sleeping to active in turn so that load is balanced. There are three possible states for nodes: sleeping, active, and discovery (determining the neighbor in the grid).

- *GEAR, Geographical and Energy-Aware Routing* [5]*:* The main idea is to use geographical information while diffusing the query. Each node keeps an estimated cost of transmissions to the destination through their neighbors. The transmission cost depends on residual energy and distance to destination. There is two phases of the algorithm: forwarding the packet to the target region, forwarding the packet within the target region.

2.1.2.1 NETWORK FLOW AND QoS-AWARE PROTOCOLS

- *Maximum Lifetime Energy Routing:* The main aim of the protocol is to maximize the network lifetime. There are two different algorithms defining the link costs differently.

i. $c_{ij}=1/(E_i-e_{ij})$

ii. $c_{ij} = e_{ij}/E_i$

E_i is the residual energy at node i, e_{ij} energy consumed when a packet transferred over the link, c_{ij} link costs.

- *Maximum Lifetime Data Gathering:* The lifetime "T" of the system defined as the number of the rounds or periodic data readings from the sensors until the first sensor dies. There are many algorithms proposed based on maximum life time concept. An algorithm called Maximum Lifetime Data Aggregation (MLDA) is proposed. The algorithm considers data aggregation while setting up maximum lifetime routes. In this case, if a schedule "S" with "T" rounds is considered, it induces a flow network G. The flow network with maximum lifetime subject to the energy constraints of sensor nodes is called an optimal admissible flow network. Then, a schedule is constructed by using this admissible flow network.

- *Minimum Cost Forwarding:* The aim is to find the path with minimum cost in the network. The cost function for the protocol captures the effect of delay, throughput and energy consumption from any node to the sink. There are two phases of the protocol. The set-up phase is every node calculates its cost of transmission to the sink by adding up cost of the link to the cost of the neighbor node (minimum of it). In the second phase, the source broadcasts the data to its neighbors. The nodes receiving the broadcast message, adds its transmission cost (to sink) to the cost of the packet. Then the node checks the remaining cost in the packet. If the remaining cost of the packet is not sufficient to reach the sink, the packet is dropped.

- *SAR, Sequential Assignment Routing:* The SAR algorithm creates multiple trees. The root of each tree is one hop neighbor from the sink. Each tree grows outward from the sink while avoiding nodes with very low QoS (i.e., low throughput/high delay) and energy reserves. One of the paths is selected according to QoS and energy resources.

- *SPEED:* Each node maintains its neighbors' information, and routes the packets using geographical information. The protocol requires calculating the estimated speed of the links and end-to-end delays. The main consideration of the algorithm is the end-to-end delay (not the power consumed). Moreover, it provides congestion control.

Table 2.1 summarizes the basic properties of the algorithms discussed.

Table 2.1 – A Comparison of the Main Routing Algorithms

Routing Protocol	Data-centric	Hierarchical	Location-based	QoS	Network-flow	Data aggregation
SPIN [9]	✓					✓
Directed Diffusion [10]	✓					✓
Rumor Routing [12]	✓					✓
GBR [13]	✓					✓
CADR [14]	✓					
COUGAR [15]	✓					✓
ACQUIRE [16]	✓					
LEACH[17]		✓				✓

cont.

PEGASIS [7]		✓				✓
TEEN[20] & APTEEN [21]	✓	✓				✓
Younis et al. [22]		✓	✓			✓
Subramanian & Katz [23]		✓				✓
MECN [24] & SMECN [25]			✓			
GAF [26]		✓	✓			
GEAR [5]			✓			
Chang et al. (max. lifetime energy)		✓			✓	
Kalpakis et al. (max. lifetime data gathering)			✓		✓	
Akkaya et al. (min. cost forwarding)				✓		
SAR				✓		
SPEED			✓	✓		

2.2 DUTY CYCLE CONTROL IN WIRELESS SENSOR NETWORKS

The key challenge in wireless sensor network protocol designs is to provide energy efficient communication, since most of the nodes in sensor networks have limited battery power and it is not feasible to recharge or replace the batteries. There are several levels of power consumption in sensor networks such as:

a. Idle Listening: The major power consumption source for WSNs,
b. Retransmissions resulting from collisions,
c. Control packet overhead,
d. Unnecessarily high transmitting power,
e. Sub-optimal utilization of the available resources [2].

By definition, sensor nodes are deployed in an *ad hoc* fashion, with individual nodes remaining largely inactive for long periods of time. In order to minimize power consumed during idle listening, some nodes, which can be considered redundant, can be put to sleep. Therefore the energy of the nodes and the energy of the network are conserved. The idea is sensor nodes dynamically create on-off schedules such that the nodes will be awake only when they are needed. This also limits the collisions, therefore the energy consumed during retransmissions. Although, it seems best way to limit consumed energy and the main consideration should be energy efficiency, the other QoS issues have to be considered.

The key design considerations for duty cycle control protocol design are scheduling and routing.

2.2.1 SCHEDULING

In order to maintain a connected network topology, to guarantee the delivery of the packets by scheduling the sleep schedules of the nodes between source and destination, the MAC layer protocols have to be carefully designed.

The S-MAC [3] protocol is proposed as a MAC algorithm in order to coordinate and synchronize the sleep/wakeup duty cycles. S-MAC is basically a CSMA/CA protocol based on 802.11. To maintain the synchronization, each node broadcasts its schedule in a SYNC message periodically, so that the neighbors can update that information in their schedule tables. The problem of neighbors can never see each other, which can be caused by SYNC message corruption, interference, or medium kept busy and SYNC packets can not be sent in time, is overcome by periodically followed neighbor discoveries. The S-MAC does not require all nodes to be synchronized, only the nodes belonging to the same virtually constructed cluster have to be synchronized, however the border nodes have to maintain more than one schedule. The scheme works well with stationary network topologies in which frequent changes are not common.

Most of the MAC protocols have been proposed for stationary networks. The objective of the following MAC protocol is its ability to work energy-efficiently in both stationary scenarios and mobile nodes. MS-MAC [4] would work similarly to S-MAC with stationary nodes. In order to avoid the excess waiting time of mobile nodes in order to join a new cluster, each node discovers the presence of mobility within its neighborhood based on the received signal levels of periodical SYNC messages from its neighbors. If there is a change in a signal received from a neighbor, it presumes that the neighbor or it-self are moving, and predicts the level of the mobile's speed. The SYNC message in MS-MAC also includes information on the estimated speed of its mobile neighbor or mobility information. If there is more than one mobile neighbor, then the SYNC message only includes the maximum estimated speed among all neighbors. This mobility information is used by neighbors to create an active zone around a mobile node when it moves from one cluster to another cluster, so that the mobile node can expedite connection setup with new neighbors before it loses all its neighbors.

Du et al. [39] proposed the algorithm in order to reduce end-to-end latency with duty cycle MAC protocol. The nodes that are forwarding data has to be awake only when they are receiving or transmitting a packet. The protocol sends a small control frame along the data forwarding path in order to inform every node when to be awake in order to receive the packet.
There are three stages of an operational cycle; SYNC, DATA, and SLEEP. Figure 2.6 shows an overview of the RMAC algorithm.

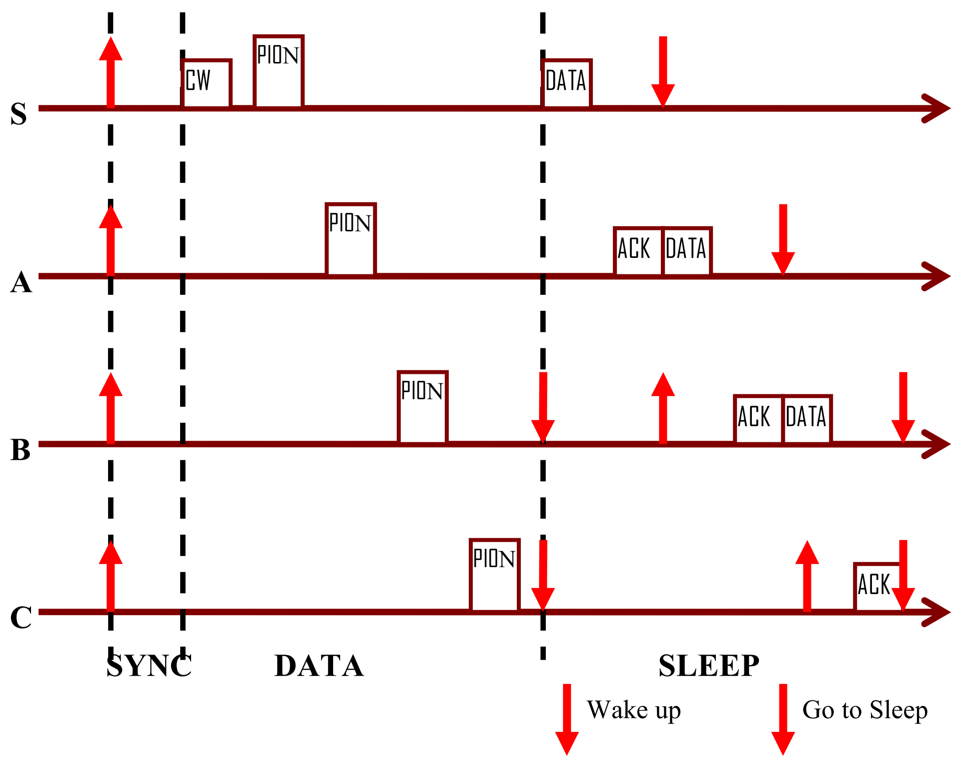

Figure 2.6 – RMAC Overview [39]

In the SYNC stage, RMAC synchronizes the clocks on the sensor nodes. In the DATA stage, firstly a control frame is sent in order to initiate the traffic. PIONs namely a series of Pioner frames are used as control frames like RTS and CTS. A PION is for requesting communication from downstream, like an RTS frame and also used for confirming the communication to upstream like CTS. Using a PION in dual purpose increases the efficiency. During the SLEEP period, nodes go to sleep if they do not have a communication task, that is set by a PION. If they are stimulated with a PION, they must stay awake for a specific time in order to be able to receive and forward the packet. Completing its task, each node goes back to sleep state.

The cross-layer scheduling algorithm for power efficiency [2] is proposed in order to conserve energy by turning off some sensor nodes. The idea is sensor nodes dynamically create on-off schedules such that the nodes will be awake only when they are needed. The scheduling and routing schemes work separately. There are two phases of the algorithm: *The Setup and Reconfiguration Phase and the Steady State Phase.*

The Setup and Reconfiguration Phase: It is initialization of the network to update the network routes and queries. This phase is relatively short; its goal is to set up the schedules that will be used during the steady state phase. The setup and reconfiguration algorithm is *independent* of the underlying routing algorithm. Therefore, many of the algorithms available for routing in ad hoc and sensor networks can be used. Power aware routing algorithms may be preferable, as they have been shown to provide substantial increases in network lifetime.

The Steady State Phase: It is similar to forwarding phase. It utilizes the Schedule established in the setup and reconfiguration phase to forward the data to the base station. Each node stores a schedule table. The scheduling for sleep and active states are calculated according to the packets that the nodes will transfer. Three different actions considered in this paper are: Sample, Transmit, and Receive.

2.2.2 ROUTING

Putting nodes to sleep affects network layer, because the sleeping nodes are no longer the part of the network, so they can not participate in the routing. Moreover there will be topology changes caused by sleep schedules. A link between two nodes will be active if and only if both nodes are active. The path selection has to be carefully engineered, because the algorithm affects the latency and power consumption.

"A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management" [6] algorithm is proposed, in order to construct the approximate topology of the network, using neighborhood information and putting the redundant nodes to sleep. These nodes logically organize the network in the form of clusters comprised of nodes in their neighborhood. *TopDisc* forms a Tree of Clusters *(TreC)* rooted at the monitoring node, which initiates the topology discovery process.

The "topology discovery request" message floods through the network; every active network receives the message. The node receiving the "topology discovery request" may respond this message in two different ways: *Direct Response* (i.e. every active node receives the request, forwards it to one of its neighbors, and immediately sends back a response with its neighbor list along the reverse path) or *Aggregated Response (*i.e. before sending a response, it waits its child nodes' responses in order to aggregate the responses then sends back to its parent).

The *TopDisc* algorithm is a hierarchical tree-based clustering scheme gathers neighborhood information from all sensor nodes. However, this protocol provides only partial link information. The algorithm assigns the nodes their duties and gets the topology of the network. Then, the idle nodes are selected and they put to sleep, and duty cycle assignment is done. The main idea is to employ minimum required number of awake nodes, and the rest of the nodes are sleeping.

The authors of [27] propose the "Adaptive Sleep Discipline for Energy Conservation and Robustness in Dense Sensor Networks" algorithm, a randomized algorithm to provide robustness to the variations in network connectivity. The algorithm does not require nodes to keep any state information about their individual neighbors. Each node independently decides when to sleep and wakeup, based on local observations.

The main constraints considered are latency, capacity (i.e. ability to carry a certain load), employing no global time-slots or coordination with neighbor.
If the estimated activity is too low to satisfy the delay constraint, the node decides to wake up more often. Conversely, if the activity is higher than necessary, the node decides to sleep longer.

Energy Aware Adaptive Low Power Listening (EA-ALPL) [28] better adapts to dynamic sensor network topologies and non-uniform energy consumption. EA-ALPL enables each sensor node to set its own listening mode according to its local state.

Initially nodes are unaware of their neighbors, so all nodes listen at an initial listening mode. Firstly each sensor sends periodic route update messages to declare its presence and state. A routing graph is formed, data flows through a base station in order to learn how many descendants the node has in the routing graph. According to its number of descending nodes, each node calculates it duty cycle. If a node has many child nodes and depleted its energy sources, announces to its child to select another parent in order to decrease the burden of the node.

“Network Coverage Using Low-Duty Cycled Sensors” [29] considers the main problem as network coverage. The node decides to sleep according to its residual energy and obviously its relative location. Each sensor in the network knows its location someway and there three roles to assign the nodes as head sponsor and a regular node.

Initially every node is regular. Each sensor informs its neighbors its location periodically by sending coordinate packet, COR. The sensor decides to enter sleep state after realizing that its sensing area is fully contained by its neighbors. The node deciding to sleep sends REQs to its neighbors. If the neighbors send ACK, they become sponsors for a designated time. The sponsors are not allowed to sleep for the designated time. However, RACP allows only the necessary neighbors are the sponsors, others are free to sleep.

In order to avoid simultaneous sleep requests of neighbor nodes and ACKs, waiting a random back-off time before sending REQs and ACKs is proposed. The back-off time can be proportional to the residual energy in order to make easier to put sleep the nodes with low residual energy.

Tian and Georganas [30] considers that every node knows its own location and knows the size of its sensing area. The algorithm is divided into duty cycles, which begins with a self-scheduling phase followed by a working phase. Before the first phase every node is on-duty. In the first phase, the self-scheduling phase, each node advertises its position with Position Advertisement Message (PAM) and listens its neighbor’s PAMs. Then it decides whether or not to sleep by calculating the sponsored coverage by its neighbors with its own sensing area. Nodes investigate the eligibility rule and decide their operation mode (on-duty or off-duty). If the sensing area of one node is fully embraced by the union set of its neighbors’, this node is eligible to sleep in order to reduce overall energy dissipation of the network. Eligible nodes can turn off its sensing and transmitting unit in order to save energy. Non-eligible nodes perform sensing tasks i.e.

collecting and delivering data to sink node. However the eligible nodes do not turn their sensing units immediately. In order to eliminate blind points, the simultaneous decision making of nodes is prevented by introducing random back off scheme.

PEAS [31] is another algorithm which aims to reduce power consumption by keeping only a essential set of nodes working and putting the rest into sleep mode. The sleeping periods are self-adjusted dynamically, so as to keep the sensors' wakeup rate almost constant, thus adjusting to high node densities. PEAS protocol is suitable for much harsh or even hostile working environment, where node failures is very common, the need for long-last operations with dense networks, and nodes are too constrained in memory and computing resource. PEAS consists of two simple algorithms: Probing Environment and Adaptive Sleeping. In the probing phase, initially all nodes are sleeping and they sleep for an exponentially distributed random time. When a node wakes up, it sends a PROBE message within a certain probing range Rp (different from sensing range). Any working nodes within Rp should send back a REPLY message. The sleeping node starts working if it does not hear any REPLY message. Otherwise, it goes back to sleep again for another random time. The probing range Rp is given by the application depending on the degree of robustness it needs. An application requiring highly robust functioning may choose a small Rp to achieve a greater density, thus higher redundancy of working nodes. Adaptive sleeping is based on keeping the number of wakeups, so the overhead is constant in unit time regardless of the network size. The frequency of wakeups is application dependent.

Jemal et al. [32] focuses on the problem of maximizing sensor network lifetime using grid-based sensor networks with emphases on sensor node sleep scheduling problem. It is assumed that the sensors have the capability of buffering sensed data and there exists a mobile base station which is a single mobile data collector such as an unmanned aerial vehicle (UAV) that flies around the field being monitored, visits the area periodically, and collects the data buffered from the nodes in the area. These properties eliminate the need for lengthy multi-hop routing. The fundamental idea of the proposed algorithm is to schedule the sleep and wakeup rate dynamically, according to the characteristics of the mobile base station movement scheduling. The algorithm combines three approaches: virtual cluster architecture, dynamic cluster head and scheduled wakeup/sleep components. The sensor nodes are grouped into units, called virtual clusters, each has a set of cluster heads. The cluster heads are not fixed, it is determined based on the position of mobile base station. Multihop communication is used in each cluster in order to transmit data to cluster heads. Base station only visits the cluster heads instead of visiting each sensor. Duty- cycle scheduling is determined by the position of the MBS. This set is determined by current cluster node and the next cluster that MBS will visit. The granularity of the sleep interval is also determined by the location of the MBS. The further the MBS is the longer a sensor node could afford to spend in sleep mode.

Table 2.2 summarizes the basic properties of these duty cycle control algorithms.

Table 2.2 – Comparison of Duty Cycle Control Algorithms

Routing Protocols	Scheduling	Sleep Decision
S-MAC [3]	✓	Predefined duty-cycle
MS-MAC [4]	✓	Predefined duty-cycle

cont.

RMAC [39]	✓	Synchronization using PION packets
The Cross-Layer Scheduling Algorithm [2]	✓	Each node adjusts its duty cycle dynamically according to network load.
A Topology Discovery Algorithm [6]	✗	Each node decides to sleep according to the network topology, its location, and the residual energy.
Adaptive Sleep Discipline [27]	✗	Each node decides to sleep based on local observations of latency and load capacity of network dynamically.
EA-ALPL [28]	✗	Each node calculates its duty cycle according to information of its location (number of descending nodes in the tree structure) and residual energy.

cont.

Network Coverage Using Low-Duty Cycled Sensors [29]	✘	Each node decides to sleep according to its location, sensor density of the network, and its residual energy.
Tian and Georganas [30]	✘	Each node decides to sleep the sensing area is fully embraced by the union set of its neighbors'.
PEAS [31]	✘	The algorithm is based on keeping only an essential set of nodes working and putting the rest into sleep mode.
Jemal et al. [32]	✓	Grid based topography and UMV is used in order to decide sleeping schedules.

Up to now, we have investigated routing and duty cycle control algorithms proposed in the literature. In the next chapter, one of these duty cycle control algorithms will be further investigated; its performance will be studied and compared with the performance of the algorithm that we shall propose in Chapter Four.

CHAPTER 3

NETWORK COVERAGE USING LOW DUTY CYCLED SENSORS

3.1 SIMULATION BACKGROUND

In the scope of this thesis, the algorithm proposed by Liu and Hsin [29] is simulated and the results are compared with the results presented in the related paper. The objective of this part of the study is essentially to verify our simulation infrastructure that will be used to evaluate the performance of SIPF in Chapter 4.

Liu and Hsin proposed an algorithm in order to put redundant nodes into the sleep (off) state so the idle listening power dissipation will be eliminated, the overall network power consumption will be decreased, and the network lifetime will be increased.

They performed simulations using Matlab, however in the present study; ns2 is used while confirming the results. The outcomes can be considered as consistent with the result presented in the paper. Later the comparisons will be presented.

The reason for selected this algorithm for comparison is we inspired from this algorithm while designing a new duty cycle control algorithm. Another reason is we inspected energy efficiency by examining sleep sensor ratio like they do so.

Although the algorithm has been briefly introduced in Chapter 2, a detailed description will be given below in Section 3.2, in order to form an opinion about the simulation results.

The simulations are performed using ns2. Ns2 stands for **n**etwork **s**imulator (ver. **2**). Ns2 is an object-oriented, discrete event driven network simulator developed at UC Berkeley [33]. Ns-2 is a commonly used tool to simulate the behavior of wired and wireless networks.

Ns provides significant support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. It also supports applications like web caching. It implements network protocols such as TCP and UDP, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms. NS also implements multicasting and some of the MAC layer protocols for LAN simulations.

Ns Simulator is based on two languages: an object oriented Simulator, written in C++, and an OTcl (an object oriented extension of Tcl (Tool Command Language)) interpreter, used to execute user's command scripts [35].

The simulator is based on two class hierarchies: the compiled C++ hierarchy and the interpreted OTcl one, with one to one correspondence between them [36]. There are two specifications that have to be achieved by the simulator, so these two languages are used by ns. Detailed simulations of protocols need a systems programming language which can competently manipulate bytes, packet headers, and implement algorithms. For these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. However, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important. Ns meets both of these requirements. The compiled C++ hierarchy achieves efficiency in the simulation and faster execution times, and reduces packet and event processing time. In the OTcl script provided by user, the particular network topology, the specific protocols, the applications that will be simulated, and the form of the output can be defined. The OTcl can make use of objects compiled in C++ through an OTcl linkage that creates a matching of OTcl object for each of the C++ [36].

The code to interface with the OTcl interpreter resides in a separate directory, tclcl. The rest of the simulator code resides in the ns-2 directory.
The most important six classes that are used in *ns*: The Class Tcl, containing the methods that C++ code will use to access the interpreter, the class TclObject, the base class for all simulator objects that are also mirrored in the compiled hierarchy. The class TclClass, defining the interpreted class hierarchy, and the methods to permit the user to instantiate TclObjects, the class TclCommand, used to define simple global interpreter commands, the class EmbeddedTcl containing the methods to load higher level built-in commands that make configuring simulations easier, and the class InstVar containing methods to Access C++ member variables as OTcl instance variables.

Ns-2 includes a tool for viewing the simulation results, called nam, the network animator. Nam is a Tcl/Tk based animation tool tovisualize the network simulation traces and real world packet trace data. The design theory behind nam was to create an animator that is able to read large animation data sets and be extensible enough so that it could be used in different network visualization situations. Under this constraint nam was designed to read simple animation event commands from a large trace file. In order to handle large animation data sets a minimum amount of information is kept in memory. Event commands are kept in the file and reread from the file whenever necessary. Unfortunately nam tool of the software is not completely developed and it can not be used for wireless network visualization situations.

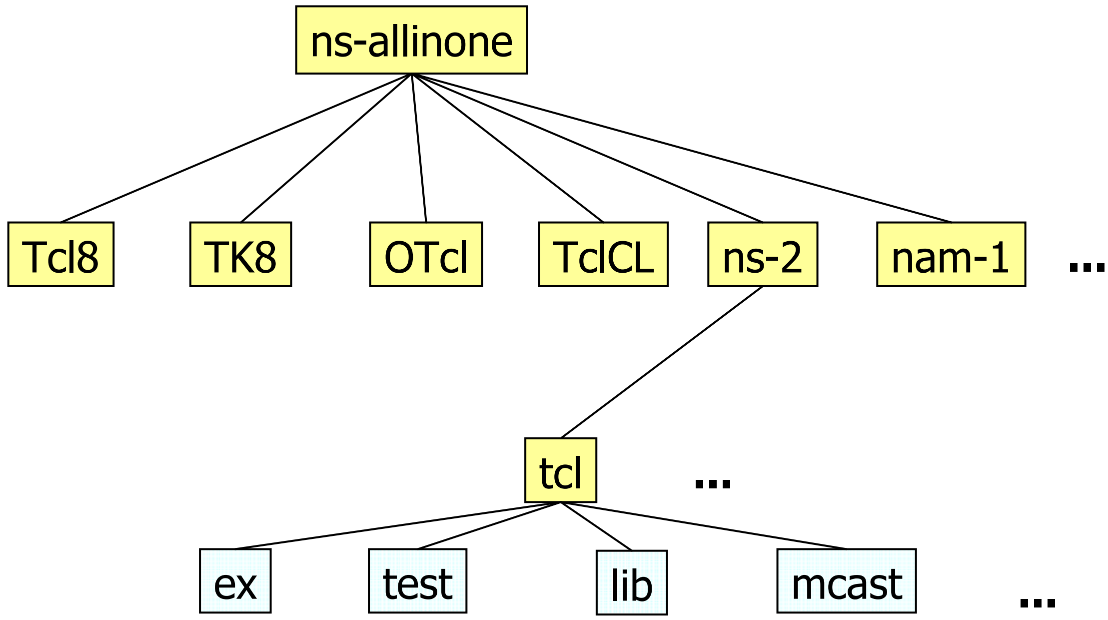

Figure 3.1 – Ns Directory Structure [37]

To sum up ns is an object-oriented, discrete event driven network simulator. Since ns is an extensible and an open source program, it is very suitable for academic and educational purposes. Moreover it is possible to create new algorithms using a rich library of network and protocol objects.

3.2 SIMULATION RESULTS

The algorithm proposed by Liu and Hsin [29] is Role-Alternating, Coverage-Preserving, and Coordinated Sleep Algorithm (RACP). Each sensor in the network knows its location someway and there are three roles to assign the nodes as head, sponsor, and a regular node. Initially every node is regular.

The algorithm can be considered as divided into cycles consisting of two phases. In the first phase each sensor informs its neighbors its location by sending the coordinate packet, COR. The COR packet is sent at the beginning of the each duty cycle periodically. At the same time, nodes listen to their neighbors' COR packet. In the next phase, the sensor decides to enter sleep state after realizing that its sensing area is fully embraced by its neighbors. The node deciding to sleep sends REQs to its neighbors. If the neighbors send ACK, they become sponsors for a designated time. The sponsors are not allowed to sleep for the designated time. The node sending requests decides to sleep if it receives enough ACKs, to fully cover its sensing area.

In order to avoid simultaneous sleep requests of neighbor nodes and ACKs, waiting a random back-off time before sending REQs and ACKs is proposed. The back-off time can be proportional to the residual energy in order to make easier to put sleep the nodes with low residual energy.

Sensors are deployed randomly with node density λ in a square field of dimension L x L. Node density operationally defined as number of sensor nodes in unit area. The communication radius (both reception and transmission) and sensing radius is considered to be the same. The sensing and communication radius is R.

Liu and Hsin take the dimension of the square field, L, as 50m. Total area of the square region which the nodes deployed is $2500m^2$. The sensing and the communication radius is considered to be 1m. The simulations are done for different node densities and the sleep sensor ratio is explored. Sleep sensor ratio is defined as the ratio of the number of sleeping sensors to the number of total sensors averaged over time before the first sensor death.

The simulations performed using ns2 in the scope of this thesis work used L as 4m, R as 1m. Figure 3.3 presents the sleep sensor ratio values obtained in our simulations as well as those reported in [29]. The reason for using smaller dimensions in this simulation work is to match with memory and processor limitation.

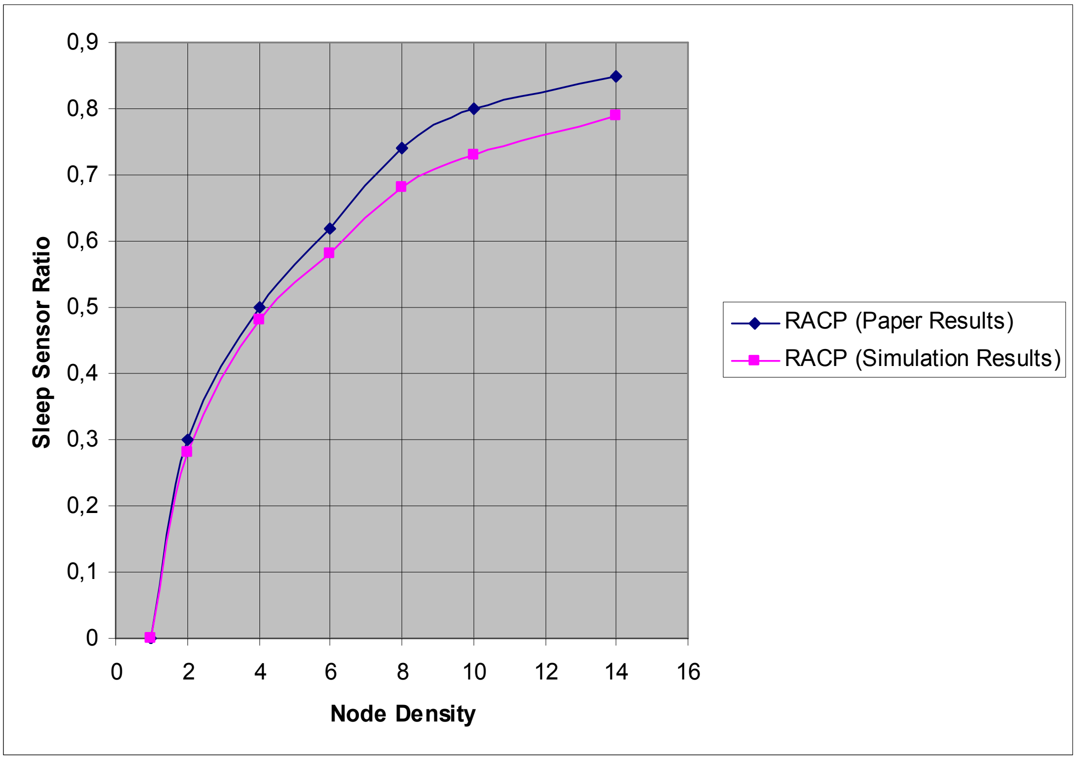

Figure 3.2 – Comparison of Simulation Results

As seen in the graph the obtained sleep sensor ratio is slightly but consistently less than the paper results. The reason is the difference in the number of edge nodes.

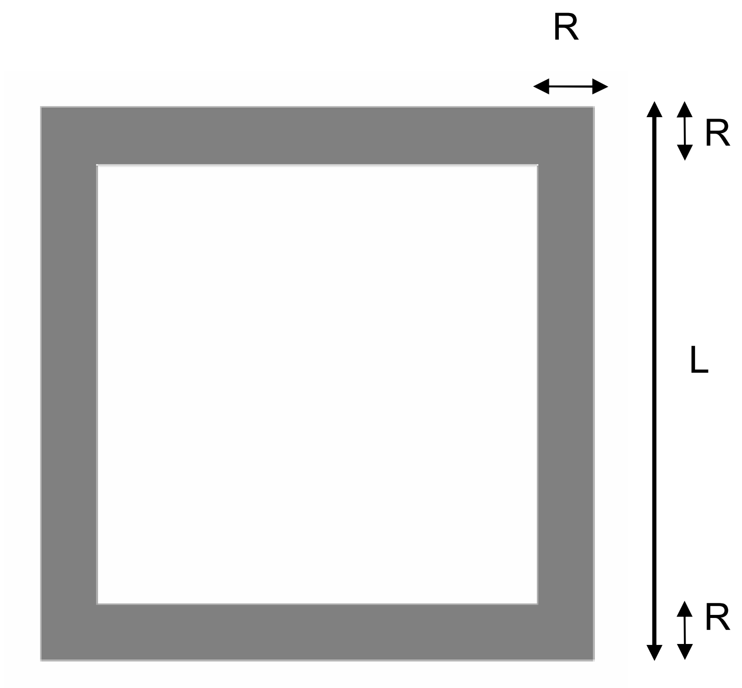

Figure 3.3 – Edge Nodes

Edge nodes are the nodes located at the boundary of the deployed area; shaded area in Figure 3.3 shows the edge nodes. Since a smaller deployment area is used and $1-\frac{(L-R)^2}{L^2}$ is the ratio of edge nodes to number of nodes and a smaller deployment area is used. Namely smaller value for *L,* edge dimension of deployment region is chosen and same value for *R*, sensing and communication range is chosen in order to match with memory and processor limitation match.

$$1-\frac{(L-R)^2}{L^2}=\frac{L^2-(L^2-2LR+R^2)}{L^2}=\frac{2LR-R^2}{L^2}$$

The equation above shows that smaller L causes larger number of edge nodes. Since the nodes are randomly deployed and our deployment area is smaller compared with the investigated simulations' coverage area, the number of edge nodes is larger in our network. The eligibility to be turned off will not be satisfied for edge nodes since their coverage area can not be covered by their neighbor nodes. Intuitively, increasing edge nodes will increase the number of the sponsor nodes.

CHAPTER 4

THE PROPOSED ALGORITHM

In this chapter a new algorithm, Sink Initiated Path Formation (SIPF), for sensor network node sleep scheduling will be proposed in order to further increase the sleep sensor ratio and thus reduce overall energy consumption and increase network lifetime.

4.1 ASSUMPTIONS

A two dimensional square area is used for node deployment. The dimension of one edge is *L*. The network consists of a sink and randomly deployed nodes. The sink has more power and more processing capability than the other nodes. Since the sink aggregates data in order to send to a distant station, the events gathered by nodes will be sent to the sink which is at the one corner of the deployment area. The location of the sink is considered as known by deployed nodes. Each node knows its location coordinates and the dimensions of the deployment region using low-cost, low-power GPS or other localization algorithms. The network uses multihop communication; this also results in a reduction in energy dissipation. For simplicity in analysis, it is assumed that sensing and communication radius are the same and has identical value, *R*, for each node. Therefore, each node is connected if and only if the distance between them is equal or smaller than the sensing and communication range, *R*. Each node considers the connected nodes as its neighbor nodes.

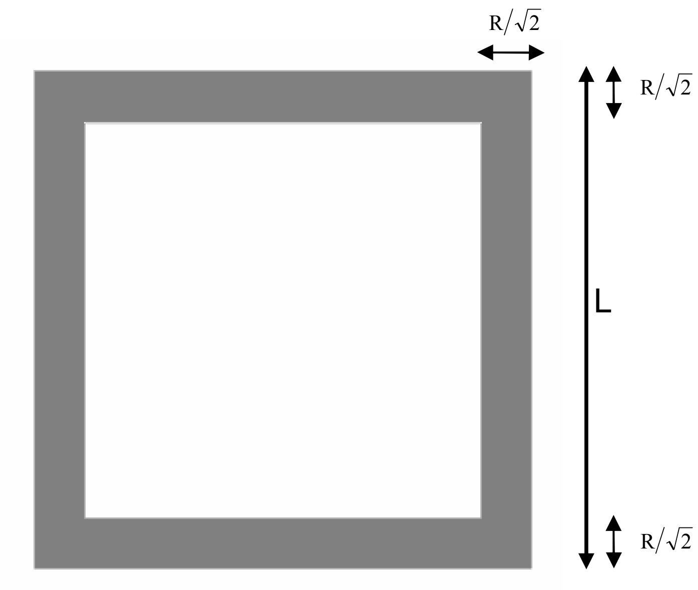

Figure 4.1 – Leaf and Central Nodes

The nodes are divided into two main groups, according to their location. The nodes on edges of the deployment area are considered as leaf nodes, nodes in the shaded area as shown in the Figure 4.1. The thickness of the edge band which the leaf nodes located is $T = R/\sqrt{2}$, where R is sensing radius and T stands for thickness of the edge band.

There are six roles to assign to the nodes; three of them are for leaf nodes and the rest are for central nodes. Three roles for leaf nodes are leaf regular, leaf sponsor, and leaf head. Three roles for regular nodes are regular, sponsor, and head.

In the algorithm the main constraint is considered as power dissipation, and power consumed in the network is considered as correlated with the number of awake nodes. Therefore the purpose is to maximize the sleep sensor ratio of the network.

4.2 DESCRIPTION OF THE ALGORITHM

The algorithm can be considered as divided into duty cycles consisting of two phases. Each cycle mainly consists of a self scheduling phase and a data transfer phase.

In the self scheduling phase, some nodes will be decided to be redundant and they can be put off. In the data transfer phase only necessary nodes will be awaken, and the data transfer is done from event to sink. Since in the self-scheduling phase there is some power dissipation, the data transfer phase should be long enough compared with self scheduling phase in order to compensate energy dissipated in the self scheduling phase.

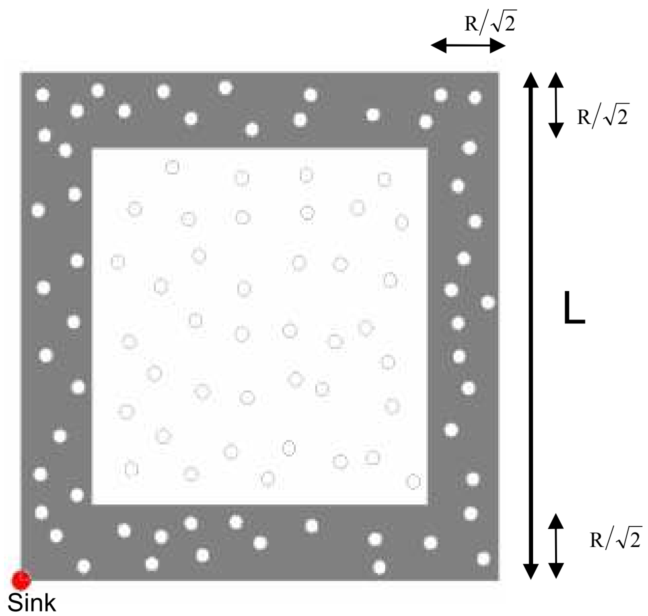

Figure 4.2 – Sink, Leaf Nodes and Central Nodes

The self scheduling phase is divided into three subsections. At the beginning of the first duty cycle, the roles for the deployed nodes are not assigned. Each node considered as knowing its coordinates, dimensions of the deployment region, and the coordinates of the sink. The deployed nodes decide to be a regular node or a leaf regular node according to their coordinates. The nodes on edges of the deployment area are considered as leaf regular nodes, the other nodes are regular nodes. Figure 4.2 shows the edge nodes as in the shaded area.

The thickness of the edge band which the leaf nodes located is $T = R/\sqrt{2}$, where R and T stand for sensing radius and thickness of the edge band, consecutively. After deciding to be leaf regular node or regular node not, each node broadcasts a "Hello" packet to its neighbors including the information of its coordinates with source ID, its residual energy and its role. At the same time every node listens for its neighbors' "Hello" packet for a designated time and records its neighbors to its neighbor list. Coming to the end of the designated hello waiting time, all nodes hear from their neighbors and the second sub phase starts.

The second sub phase is selection of head leaf nodes. The main idea is selecting the possible most distant neighbor node as the next node instead of making a random selection.

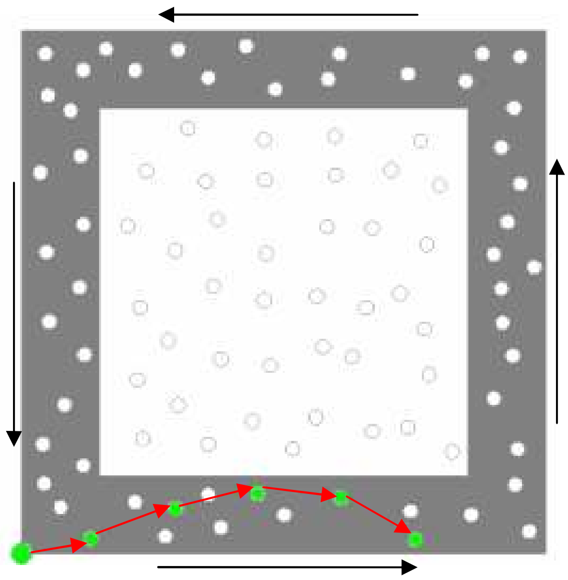

Figure 4.3 – Head Leaf Node Selection

Therefore the number of the head leaf nodes namely awake leaf nodes is minimized. The selection starts with the sink. Sink chooses a leaf node with maximum distance but obviously the selected leaf node is sink's neighbor. Sink node sends a request packet to the selected node in order to inform that it will be the next head leaf node. Receiving the request packet, the selected head leaf node selects another node to be the next head leaf node among its neighbors considering the distance between them then informs by sending a head leaf node request packet. This procedure is repeated through the subsequent selections. The selections are done through a predefined direction, encircling the square deployment region, as shown in the Figure 4.3. The leaf nodes selected as head leaf nodes will be awake and rest of the nodes will be asleep.

If sink or any head leaf node can not find a neighbor through the predefined direction, the algorithm will terminate immediately. This causes no node is put to sleep state. This means that network is not connected so it will not be healthy to run the algorithm and to put some nodes to sleep state. For this reason, for small node densities in which the network has large number of blind points, it is not possible to get acceptable sleep sensor ratio.

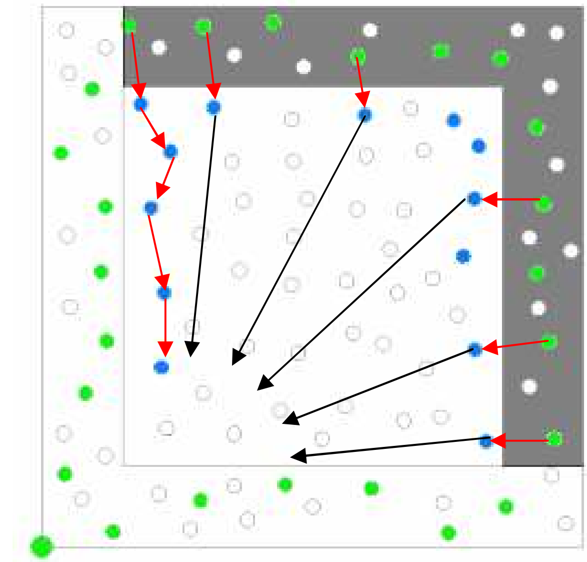

Figure 4.4 – Central Head Node Selection

The last sub-phase is the selection of central head nodes. Similarly, the aim is to select the possible most distant neighbor node directed to the sink as the next node instead of making a random selection. The selection is initiated by the head leaf nodes residing at the edges that are not adjacent to the sink corner, shaded edges in the Figure 4.4. The selection starts with a randomly determined delay in each node, the value of delay has different value for each node. Each leaf node finds the possible most distant neighbor node directed to the sink, informs it by sending head node request packet. The node receiving the request packet becomes a head node. Then the node selected to be head node finds the possible most distant neighbor node directed to the sink. This procedure is repeated through the subsequent selections directed through the sink, as shown in the Figure 4.4.

The selection procedure is not started concurrently by each head leaf node residing at the edges that are not adjacent to the sink corner, because as mentioned before a randomly determined delay is established before starting the head node selection procedure. If a neighbor has been selected as head node by another node and this neighbor is closer than itself to the sink, the node would not select any node. The traffic that will be sent to the sink will be forwarded through the neighbor node that was selected as head, previously. Thus the number of awake nodes will be minimized.

The selected nodes namely nodes receiving requests will be active, rest of the nodes will be turned off, in order to minimize the overall power consumption. Completing the selection of the head nodes, the self scheduling phase will be over and a data transfer phase will start.

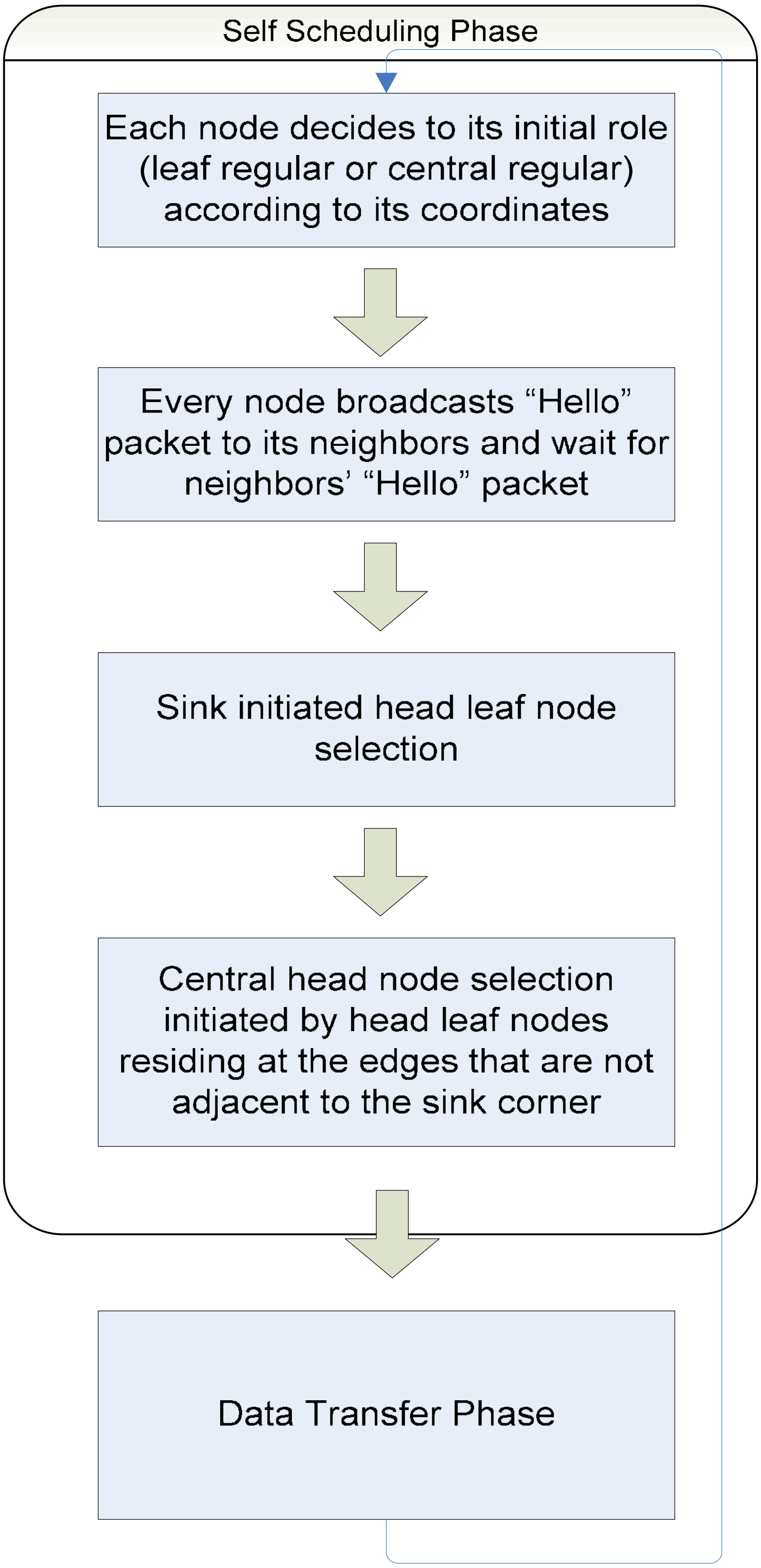

Figure 4.5 – SIPF Algorithm

The flow chart in Figure 4.5 summarizes the algorithm

In order to distribute the work load consequently energy dissipation overhead evenly all over the network, the self scheduling phase will be repeated periodically in order to put other nodes different nodes asleep. The data transfer phase have last long enough to compensate the time and consumed power overhead introduced by the self scheduling phase. In order to avoid selection of same nodes as head or head leaf nodes, residual energy will be considered while selecting active nodes, in consequent self scheduling phases. The hello packet which broadcasted at the beginning of the self scheduling phase will include information about residual energy of the node.

4.3 PERFORMANCE EVALUATION

In this sub-section performance evaluation of the proposed algorithm based on ns2 simulations will be presented. Sensors are deployed randomly with node density λ in a square field of dimension L x L. The nodes are uniformly distributed. Although the communication radius (both reception and transmission) and sensing radius is considered as identical, it is easy to modify their values. The sensing and communication radius is R. As mentioned before, the sensor nodes know their location, using a low-power, low-cost GPS or other localization algorithms.

The edge length, *L, of* the square deployment region is taken as 4 unit lenght. The sensing and the communication range, R is 1 unit lenght. The simulations are performed for different node densities and the sleep sensor ratio is explored.

The simulations performed using ns2 in the scope of this thesis work. Wireless channel is used as channel, 802.11 is used as MAC, Omni Antenna is used as antenna during the simulations.

Figure 4.5 presents the graph for both simulations' results. The reason again for using smaller dimensions than the dimensions used by Liu and Hsin in this simulation work is to match with memory and processor limitation of the simulation computer.

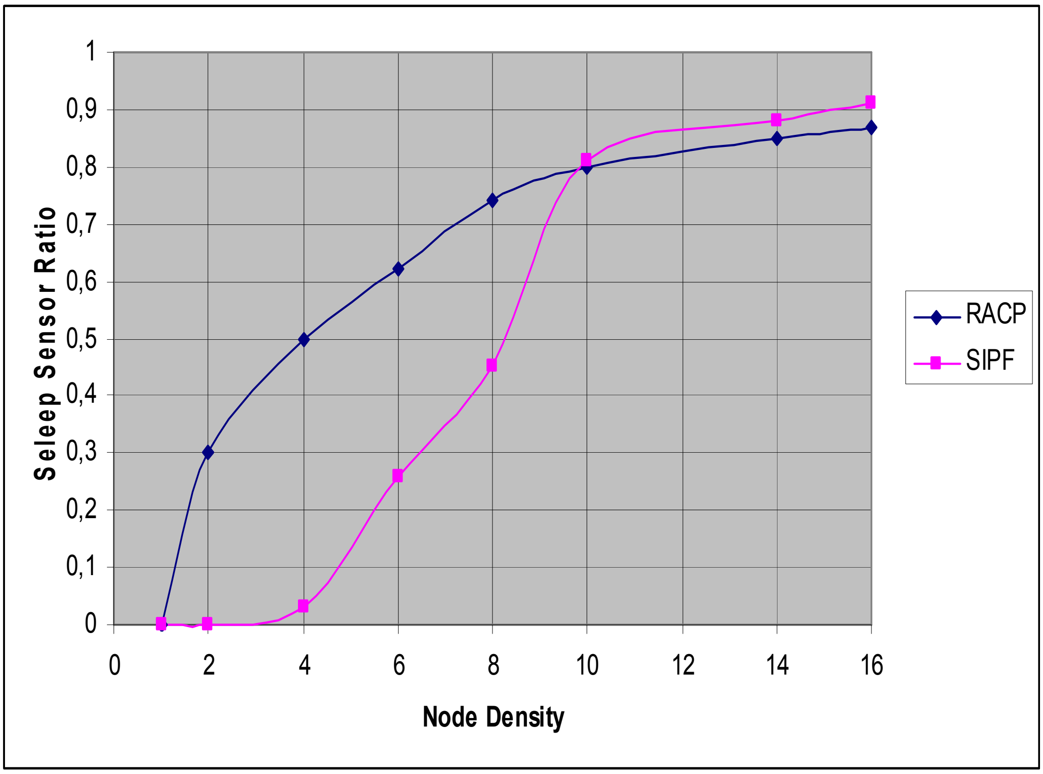

Figure 4.6 – Comparison of Performance

The basic objective of our algorithm is to achieve minimum power dissipation by turning maximum number of nodes off at the same time conserving the connectivity in the network. Therefore sleep sensor ratio is explored and compared with the results acquired by Liu and Hsin. Figure 4.6 illustrates the sleep sensor ratio that can be achieved under our algorithm and under RACP algorithm. As mentioned in Sec. 3.2, sleep sensor ratio is defined as the ratio of the number of sleeping sensors to the number of total sensors averaged over time before the first sensor death.

One important result that has to be mentioned is that the sleep sensor ratio obtained with our proposed algorithm is 0 for node densities smaller than 8. This is due to the fact that with low node densities, the network is not fully connected in the scope of our algorithm. In essence, while the algorithm runs, some nodes do not have any node to select as next head node; this brings an end to the algorithm. For small node densities, no node can be put to sleep. Even for node density of 8, 46% of simulations result with no sleeping nodes. For this reason, for small node densities in which the network has large number of blind points, it is not healthy to apply our algorithm. Trying to decrease power consumption may lead to failures of data transfer.

This comparison clearly shows that for large densities, our algorithm outperforms RACP in that higher sleep sensor ratios are achieved, leading to higher energy conservation.

Table 4.1 – Coverage Evaluation of SIPF

Node Density	**Coverage**
8	83.2%
10	81.5%
14	76,7%

Table 4.1 shows the time averaged coverage obtained by SIPF. It is seen that coverage levels are decreasing with increasing node density. The reason for

decreasing coverage with increasing node density is that the average number of awake nodes decreases with increased node density. This table clearly shows that energy conservation is obtained at the cost of reduced coverage of the sensing area.

As mentioned before, SIPF fails in the networks with low densities. One solution to overcome the shortcoming of the algorithm can be to switch and use another duty cycle control algorithm such as RACP, when SIPF fails. This introduces an overhead of time, packet traffic, data processing, and also extra energy dissipation. However this portion of energy can also be considered as negligible compared with the energy dissipated in the data transfer phase. It is obvious that the time for scheduling will increase resulting in increased latency in the network.

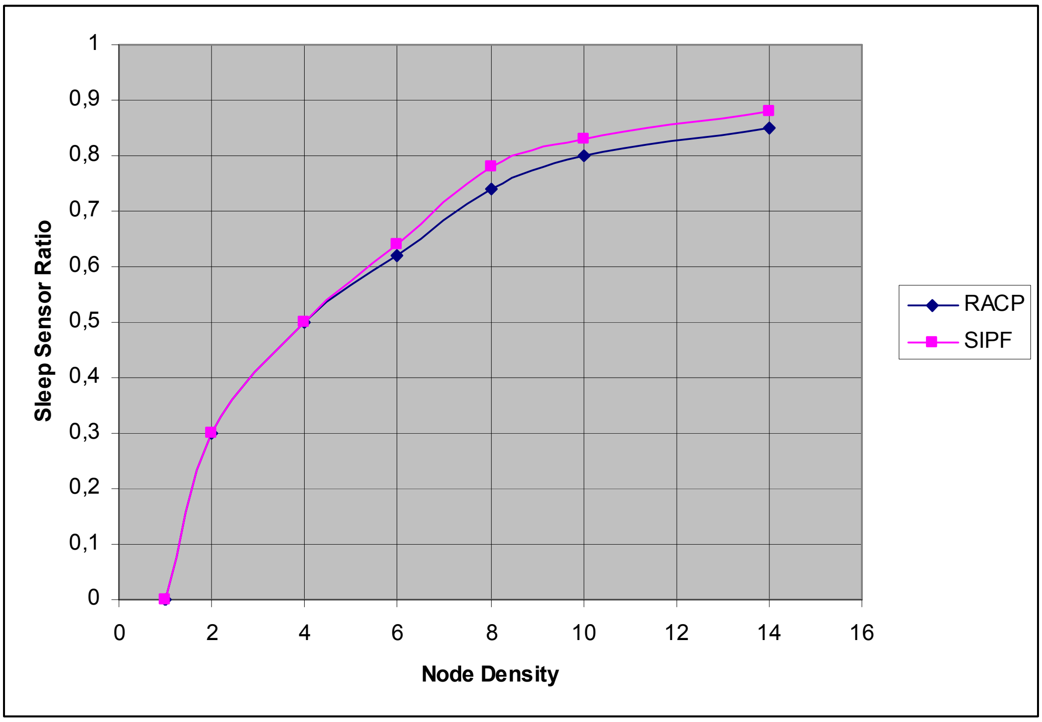

Figure 4.7 – Performance Comparison of RACP with SIPF combined with RACP

The Figure 4.7 shows that the overall sleep sensor ratio is increased when SIPF and RACP are used together, as described. The simulation for this is not performed; the graph is obtained by combining the two separate simulation results.

CHAPTER 5

CONCLUSION

As mentioned before, efficient power consumption is a challenging problem in wireless sensor networks which is battery-powered and used for detection in wide variety of applications. Since each node has limited battery power and it is impossible or infeasible to recharge the batteries, reducing power consumption will increase the network lifetime. The network lifetime directly proportional to the efficient power consumption and disfunction of any node causes serious damage to the network service considering nodes' dual role of data originator and data router. Considering densely deployment property of the WSNs and efficient energy consumption requirement, duty cycle control algorithms are realized.

In this study, a novel algorithm has been proposed and simulated. The sleep sensor ratio is examined and compared with the algorithm, RACP [29].

The algorithm that we proposed is based on increased number of sleeping sensors. The algorithm consists of cycles in which nodes decides to sleep or not. In each cycle, different nodes select to be awake. Therefore periodical sleeping is realized and increasing sleep sensor ratio increases network lifetime.

The simulation results show that our algorithm outperforms RACP, in terms of the ratio of sleeping nodes, with densely deployed sensor networks in that a greater reduction in duty cycle is achieved. This signifies that power consumed in a specific time will be smaller with our algorithm. Therefore with our duty cycle control algorithm, the sensor network will survive longer, in which power consumption is one of the main constraints. This, however, is achieved at the cost of reduced area coverage, as expected.

Redundant nodes at the edges can not be put to sleep with the duty cycle control algorithm proposed by Liu and Hsin. However we can put them to sleep with our algorithm. As explained in the section 3.2, every edge node is on duty every time, regardless of the network node density with the duty cycle control algorithm proposed by Liu and Hsin. This problem is solved with our algorithm.

One of the shortcomings of our algorithm is that if the network is not densely deployed, our algorithm does not put any node to sleep. The sleep rate is 0 for node densities smaller than 8. This is due to the fact that the network is not fully connected in the scope of our algorithm. In essence, while the algorithm runs, some nodes do not have any node to select as next head node; this brings to an end to the algorithm. For small node densities, no node can be put to sleep. Even for node density of 8, 46% of simulations result with no sleeping nodes. For this reason, for small node densities in which the network has large number of blind points, it is not healthy to apply our algorithm. Trying to decrease power consumption may lead to the failures of data transfer.

However, it is worth mentioning that one of the main properties of the WSNs is dense deployment. Since an individual sensor node has limited capability and vulnerable nature, a sensor network usually has to be densely deployed. The WSNs are generally deployed with high densities, e.g. up to 20 nodes/m^3 [30]. Considering that the algorithm that we propose is only unsuccessful for smaller densities, the algorithm can be considered as successful in energy dissipation reduction.

Another solution to overcome this shortcoming of the algorithm for can be to switch and use another duty cycle control algorithm such as RACP, when SIPF fails. This increases the sleep sensor ratio for the networks with small densities. Therefore the consumed energy will also be decreased, increasing the overall network lifetime. However, this introduces an overhead of time, packet traffic, data processing, and also extra energy dissipation, as mentioned before. However this portion of energy can also be considered as negligible compared with the energy dissipated in the data transfer phase. It is obvious that the time for this iterative scheduling will increase resulting in increased latency in the network.

Our duty cycle control algorithm brings in an overhead of time dissipation, and packet traffic, and power dissipation caused by scheduling the sensor on-duty time. Therefore the data transfer phase should be long enough compared with self scheduling phase in order to overcome this overhead caused by scheduling time. The actual threshold levels of data transfer time beyond which our scheduling becomes beneficial may be further studied for precise recommendations.

Another shortcoming of the algorithm that we proposed is that the blind points are not eliminated. When an event cannot be detected by any on-duty node, but is within the range of the original sensing coverage, we call the event source cell a “blind point” [30]. Since the nodes are alternatively put to sleep state, that is to say in each cycle different nodes are selected to sleep, blind points are alternating in each cycle. Although blind points are not eliminated, the coverage obtained is acceptable. Further research may be conducted to define a combined objective function, incorporating energy conservation as well as coverage. Alternatively, the aim may be to maximize sleeping node ratio to minimize energy consumption, under the constraint of a specified minimum acceptable coverage.

As future work, different types of traffic can be applied to the network and how different types traffic structures affect our algorithm can be examined. This would enable us to examine how contention affects our algorithm. Latency can also be inspected because intuitively, having blind points in the network will cause higher end-to-end latency. This can also be examined by creating some events in the network and investigating network response time or data accumulation at the sink.

An important research goal for the future is to determine whether all these routing optimization algorithms can be unified under a single routing architecture that would be suitable for a large set of applications. A cross layer routing protocol enhanced with MAC layer will establish further increase utility and energy efficiency.

REFERENCES

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

[2] ML Sichitiu, "*Cross-layer scheduling for power efficiency in wireless sensor networks*," in INFOCOM, 2004, pp. 1740–. 1750.

[3] W. Ye, J. Heidemann, and D. Estrin, "*An Energy-Efficient MAC Protocol for Wireless sensor Networks,*" in Proc. IEEE. INFOCOM, Jun. 2002

[4] H. Pham and S. Jha, "*An adaptive mobility-aware MAC protocol for sensor Networks (MS-MAC)*", The 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS-2004), October 24-27, 2004, Fort Lauderdale, Florida, USA

[5] Y. Yu, D. Estrin, and R. Govindan, *"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.

[6] B. Deb, S. Bhatnagar, and B. Nath, "*A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management*" Technical Report Technical Report DCS-TR-441, Department of Computer Science, Rutgers University, May 2001.

[7] K. Akkaya, M. Younis, *“A survey on Routing Protocols for Wireless Sensor Networks”,* Computer Networks (Elsevier) Journal, 2004

[8] S. Hedetniemi and A. Liestman, *“A Survey of Gossiping and Broadcasting in Communication Networks”*, Networks, Vol. 18, No. 4, pp. 319-349, 1988.

[9] W. Heinzelman, 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’99), Seattle, WA, August 1999.

[10] C. Intanagonwiwat, R. Govindan and D. Estrin, *"Directed diffusion: A scalable and robust communication paradigm for sensor networks",* in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, August 2000.

[11] R. Shah and J. Rabaey, *"Energy Aware Routing for Low Energy Ad Hoc Sensor Networks",* in the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002.

[12] D. Braginsky and D. Estrin*, "Rumor Routing Algorithm for Sensor Networks,"* in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.

[13] C. Schurgers and M.B. Srivastava, “Energy *efficient routing in wireless sensor networks*,” in the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.

[14] M. Chu, H. Haussecker, and F. Zhao, *"Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks,"* The

International Journal of High Performance Computing Applications, Vol. 16, No. 3, August 2002.

[15] Y. Yao and J. Gehrke, *"The cougar approach to in-network query processing in sensor networks,"* in SIGMOD Record, September 2002.

[16] N. Sadagopan et al., *"The ACQUIRE mechanism for efficient querying in sensor networks,"* in the Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska, May 2003.

[17] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, *"Energy-efficient communication protocol for wireless sensor networks,"* in the Proceeding of the Hawaii International Conference System Sciences, Hawaii, January 2000.

[18] S. Lindsey and C. S. Raghavendra, *"PEGASIS: Power Efficient GAthering in Sensor Information Systems,"* in the Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.

[19] S. Lindsey, C. S. Raghavendra and K. Sivalingam, *"Data Gathering in Sensor Networks using the Energy*Delay Metric"*, in the Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[20] A. Manjeshwar and D. P. Agrawal, *"TEEN : A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks,"* in the Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.

[21] A. Manjeshwar and D. P. Agrawal, *"APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks,"* in the Proceedings of the 2nd International Workshop on Parallel and

Distributed Computing Issues in Wireless Networks and Mobile computing, Ft. Lauderdale, FL, April 2002.

[22] M. Younis, M. Youssef and K. Arisha, *"Energy-Aware Routing in Cluster-Based Sensor Networks",* in the Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), Fort Worth, TX, October 2002.

[23] L. Subramanian and R. H. Katz, *"An Architecture for Building Self Configurable Systems,"* in the Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA, August 2000

[24] V. Rodoplu and T.H. Ming, "*Minimum energy mobile wireless networks,"* IEEE Journal of Selected Areas in Communications, Vol. 17, No. 8, pp. 1333-1344, 1999.

[25] L. Li and J. Y. Halpern, *"Minimum energy mobile wireless networks revisited,"* in the Proceedings of IEEE International Conference on Communications (ICC'01), Helsinki, Finland, June 2001

[26] Y. Xu, J. Heidemann, and D. Estrin, *"Geography-informed energy conservation for ad hoc routing,"* in the Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01), Rome, Italy, July 2001.

[27] J. Greunen, D. Petrovic, A. Bonivento, J. Rabaey, K. Ramchandran, A. Sangiovanni-Vincentelli, "*Adaptive Sleep Discipline for Energy Conservation and Robustness in Dense Sensor Networks*" IEEE International Conference on Communications ICC04, 2004

[28] Jurdak, R., Baldi, P. and Lopes, C.V. "*Energy-Aware Adaptive Low Power Listening for Sensor Networks*" In proceedings of the Second International Workshop on Networked Sensing Systems, San Diego, CA June 2005.

[29] M. Liu and C. F. Hsin, "*Network coverage using low duty-cycled sensors: Random and coordinated sleep algorithms*," in IPSN, 2004.

[30] D. Tian and N. D. Georganas, *"A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks,"* in Wireless Communication Mobile Computing 2003; **3**:271–290

[31] F. Ye, G. Zhong, S. Lu, L. Zhang, *"PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks,"* in Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS'03*)*

[32] J. H. Abawajy, S. Nahavandi and F. Al-Neyadi, *"Sensor Node Activity Scheduling Approach*," in IEEE, 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)

[32] I. F. Akyildiz, M. C. Vuran, O. B. Akan, and W. Su, "*Wireless Sensor Networks: A Survey REVISITED,*'' to appear in Computer Networks Journal (Elsevier), 2006.

[33] G. Berbellini, *"An Introduction to NS-2",* http://esd.sci.univr.it/pages/docs/Scheda_NS2_1.pdf , 2005.

[34] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/

[35] E. Altman, T. Jimenez, "*NS Simulator for Beginners, Lecture Notes 2003, 2004"*, Univ de Los Andes, Merida, Venezuela,and ESSI.

[36] K. Fall, K. Varadhan, "*The ns Manual (formerly ns Notes and Documentation)*", The VINT Project

[37] Ö. B. Akan, "METU - EE644 Lecture Notes"

[38] S. Du, A. K. Saha, D. B. Johnson, "*RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks*" in INFOCOM 2007.